**CHAPTER 6**

**TESTING**

**6.1 SOFTWARE TESTING**

Software testing is a process used to identify the correctness, and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end user. In simple words, software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free.

Software testing is an investigation conducted to provide stakeholder with information about the quality of the product or service under test, Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risk of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing can be stated as the process of validation and verifying that a software program/application product:

1. Meets the requirement that guided its design and development;
2. Works as expected and
3. Can be implemented with the same characteristics.

Software testing depends on the testing method employed, can be implemented at any time in the development process. However, most of the test effort traditionally occurs after the requirement have been defined and the coding process has been completed having been shown that fixing a bug is less expensive when found earlier in the development process. Although in the agile approaches most of the test effort is, conversely on-going. As such, the methodology of the test is governed by the software development methodology adopted.

Different software development models will focus the test effort at different points in the development process. Newer development models, such as agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional models most of the test execution occurs after the requirement have been defined and the coding process has been completed.

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product function properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code; does it do what it is supposed to do and what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing is developed.

**6.2 TESTING PROCESS**

**6.2.1 Levels Of Testing**

* **Unit Testing**

Unit is smallest testable piece of software can be compiled, linked, loaded. e.g, functions/procedure calls, classes, interfaces normally done by the programmer. Test cases written after coding.

* **Integration Testing**

Test for correct interaction between the system units. Systems- built by merging existing libraries, modules coded by different people. Mainly tests the interfaces among the units.

* **System testing**

Test the overall interaction of components. Find disparities between implementation and specification. It involves load, performance reliability and security testing.

* **Acceptance Testing**

It demonstrates satisfaction of user. Users are essential part of process. Usually merged with system testing and done by the test team and customer and done in simulated environment or real environment.

* **Regression Testing**

It is an ongoing process throughout the lifecycle. New bug-fix breaks previously tested Units. Perform regression test whenever program changes.

**6.3 TEST ENVIRONMENT**

The hardware and software environment in which tests will be run, and any other software with which the software under test interacts when under test including stubs and test drivers. The four modules were installed as follows :

* A client machine running printer status service, readxml service and client user interface.
* An administrator system running the two services and the administrator user interface.
* A network printer connected to the network via a LAN cable.

**6.4 UNIT TESTING OF MAIN MODULES**

Unit testing focuses verification efforts on the smallest unit of software design the module. All the important control parts are tested to uncover errors within the boundary of the module. Unit testing is always white box oriented and the step can be conducted in parallel for multiple modules.

This testing involves the test carried out on module program with individual functions and procedures, which make up modules on the system. This is also called as program testing. This program should be tested for correctness of logic applied and should detect errors in coding. All the modules have been individually tested to determine whether they are coded correctly so that they satisfy the requirements in the specification and execute efficiently an individual unit. A program unit denotes a routing or a collection of routines implemented by an individual programmer.

This situation is illustrated as follows

Code & debugging Unit testing Integration testing

Unit testing has a goal of discovering errors in the individual module of the system. Modules are tested in isolation from one another in an artificial environment known as “test harness” which consists of the driver programs and data necessary to exercise the modules.

The two type of test that a source code must satisfy are

* Function test: Specify typical operation condition, typical input values and typical expected results.
* Performance test: in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload.

**6.4.1 Unit Testing of Administrator Module**

This module is concerned with testing all the administrative tasks works properly. The administrator checks for printers and systems connected in a LAN and adds them to the database. If some of the printers and systems already had been added then it sends an alert saying that particular printer or system exists in the database. If any of the error related to the database the software alert the admin to check and correct those errors.

**6.4.2 Unit Testing of Client Module**

The client module authenticates a user, if the user is not a registered user, it sends an alert saying that this particular user is not registered with the Administrator. If the user is authorized user, he has to enter user name and password. Once the user logged in he/she can check his/her quota left, if the quota got over the software will not allow the user to print. User can also request the administrator to grant him/her an additional quota to print.

**6.4.3 Unit Testing of Printer Status Module**

This module is tested using the jobs given by a user to print documents. The service will check the user quota, if the quota left is zero, then this will not allow printing. If sufficient quota is available then it records all information with respect to that job and writes into the xml document.

**6.4.4 Unit Testing of ReadXML Module**

The module is tested using the jobs given by the user. Once the windows service 1 (printer status) writes the necessary information to xml document, the windows service 2 (readxml) puts this information to the server database and deletes the related information from the client machine. Also it updates the quota left of the user into the database along with cartridge level of the printer.

**6.5 Integration Testing Of Modules**

Testing in which software components, hardware components, or both together are combined and tested to evaluate interactions between them. Integration testing usually go through several real word business scenarios to see whether the system can successfully complete workflow tasks. Integration plan specifies the order of combining the modules into partial systems. This part of testing includes combining all the modules discussed above to check whether the modules are works fine according to the constraints.

**6.6 TEST RESULTS**

All the modules were tested rigorously and any error that incurred was rectified. All the error conditions are handled and appropriate messages are displayed if any error occurs.

**6.7 SUMMARY**

The software modules were tested for the functionality and were found to perform as per specifications. The various tests like unit testing and integration testing are performed the various testing platform and the level of testing is also mentioned.