**SYSTEM TESTING**

**Project Testing**

Testing is the most important phases in the software development activity. In software development life cycle (SDLC), the main aim of the testing process in the quality, the developed software is tested against attaining the required functionality and performance.

During the testing process the software is worked with some particular test case and the output of the test cases are analyzed whether the software is working according to the expectations or not.

The success of the testing process in determining the error is mostly depends upon the test case criteria, for testing any software we need to have a description of the expected behavior of the system and method of determining whether the observed behavior confirmed to the expected behavior.

**Level of testing**

Since the error in the software can be injured at any stage. so, we have carry out the testing process at different levels during the development. The basic levels of testing are

* + - Unit Testing
    - Integration Testing
    - System Testing
    - Acceptance Testing.

The Unit Testing is carried out on coding.here different modules are tested against the specification produced during design for the modules .In case of integration testing different tested are combined into sub system and tested in case of the System testing the full software is tested and in the next level of testing the system is tested with user requirement document prepared.

**Unit Testing**

Unit testing mainly focused first in the smallest and low level modules, proceeding one at a time. Bottom-up testing was performed on each module. As developing a driver program, the test modules by developed or used. But for the purpose of testing modules themselves were used as stubs, to print verification of the actions performed. After the lower level modules were tested, the modules that the next higher level those make use of the lower modules were tested

Each module was tested against required functionality and test cases were developed to test boundary values.

The unit testing has been tested with sample data and adequate corrections were made as per the user requirement, such that login module was tested with user id and code and appropriate error messages are provided for errors like data entry error, id error,etc.

**Integration Testing**

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover error associated with interfacing. As the system consists of the number the edges of the two modules. The software tested under this incremental bottom-up approach.

Bottom-up approach integration strategy was implemented with the following steps.

* Low modules were combined into clusters that perform specific software sub function
* The clusters were tested

**Testing Process**

A number of activities must be performed for testing software. Testing start with test plan. Test plan identifies all testing related activities that need to be performed along with the schedule and guide lines for testing. The plan also specified the levels of testing that need to be done, by identifying the different testing units. For each unit specified in the plan first the test cases and reports are produced. These report are analyzed.

**Test Plan**

Test plan is a general document for entire project, which defines the scope, approach to be taken and the personal responsible for different activities of testing. The inputs for forming test plan are

* Project Plan
* Requirements Document System Design

There are two basic approaches for testing. They are

* + - Functional Testing
    - Structural Testing

**System testing**

Series of different tests whose primary purpose is to fully exercise the computer-based system. It also tested to find discrepancies between the system and its original objective, current specifications.10011

**Execution Testing**

This program was successfully loaded and executed. Due to programming there were no execution errors as fall as possible

**QUALITY ASSURANCE**

Quality assurance defines the objectives of a project and reviews the overall activities so that error are corrected early in the development process.

**Levels of Quality Assurance**

Quality Assurance comes in three main levels namely

* + - * + Testing
        + Validation
        + Certification

**Testing**

In system testing a common view is to eliminate program errors. This is extremely difficult and time consuming. Since designers cannot prove 100% accuracy. A successful test, then, is one that find errors.

**Validation**

It checks the quality of the software in both simulated and live environments. In the Simulated approach the developers test the product on their workplace to make the products meet its requirements. In the Live Environment phase the product is given to the customer to evaluate the product’s functionality.

Validation refers to the different set of activities that ensure that software correctly implements a specific function and the software that been built is traceable to customer requirements. Verification and validation can be defined in a way like:

* Verification
* Validation

Software validation is achieved through a series of black-box test that demonstrate conformity with requirement. After each validation check a test has been conducted, one of the two possible condition exists

* The function or performance characteristics conform to specification and are expected
* A deviation from specification is uncovered and a deficiency list is created

**Alpha and Beta testing**

The alpha testing is conducted at the developer’s site by the customer. the software is used in the natural setting with the developer”looking over the developer” and recording errors and usages problems. Alpha test is conducted in controlled environment

The beta testing is conducted at one or more customer site by the end user of the software. Unlike software testing the developer is generally not present. Therefore beta test is live application of the software in an environment that cannot be controlled by the developer

**Validation check applied in the project**

* The files entered in the project must only have the doc extension
* The data entered must have only one format
* One should not make a enter into without checking the password
* The qc must enter into the status column only accepted or not accepted
* File which are already be registered should not be registered once again or other time
* Qc can’t enter the file which are not entered by the dc
* Qc should enter the file into corresponding filenames registered by the dc
* Files which are registered and not accepted for the first time should not be registered again but the reentry of date and status should be done only

**Certification**

Certification is to certify that the program or software package is correct and confirms to standards. With growing trend towards purchasing ready to use software, certification has become more important.

**SYSTEM SECURITY**

Software integrity has become increasingly important in the age of hackers and firewalls. This attributes measures a system ability to withstand attacks (both accidental and intentional) to its security. Attacks can be made on all three components of software program, data, and documents

To measure integrity, two additional attributes must be defined

* + Thread
  + Security

**THREAD**

Threat is the probability (which can be derived or estimated from empirical evidence) that an attack of specific type occur with in a specific time.

**Security**

Security is the probability (which can be estimated or derived from empirical evidence) that attack on the specific type will be repelled.

**Security Testing**

Any computer based system that manages sensitive information or causes action that can improperly harm(or benefit) individuals is the target for improper or illegal penetration. Penetration spans a board range of activities; hackers who penetrate system for sport; disgruntled employee who attempt to penetrate for revenge; dishonest individual who penetrate for illicit personnel gains.

Security testing to verify that protection mechanism built into a system will in fact provide proper protection form improper penetration. During system testing, the tester plays the role of the individual who desires to penetrate the system. Anything goes! The tester may attempt to acquire password through external clerical means; may attack the system with custom software designed to break down any defenses that have been constructed may overwhelm the system thereby denying the service to other; may purposely cause system errors.