**SYSTEM TESTING**

**TESTING METHEDOLOGIES**

**System Testing**

System testing includes code testing which examines the logic of the program. Each and every part of the program is checked or executed individually to find out the errors. Once the errors in the program are found out, they are debugged. If wrong data is entered, an error message is displayed on the screen so that the user can correct the data at that time itself.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. A series of testing are performed for the proposed system before the system is ready for the user acceptance test. A candidate system is subject to variety of tests – volume, stress, recovery, security and usability tests.

The steps in the system testing can be categorized as follows:

* Unit Testing
* Requirement Testing
* User Testing
* Validation Testing
* Integration Testing
* User Acceptance Testing

**Unit Testing**

Unit testing focuses on verification efforts on the smallest unit of software design i.e., the module. The unit testing is always white box oriented and the step can be conducted in parallel for modules. The module interface is tested to ensure that information properly flows in and out of the program unit under test. The ‘local data structures ‘are examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm execution.” Boundary Conditions” are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All ‘independent paths ‘through the control structures are exercised to ensure that all statement in a module have been executed at least once. Finally all “Error-Handling “ are tested.

**Requirement Testing**

The main aim of this test plan is to see whether the outputs created and inputs were given according to the user requirements and specifications that have been established. This was done in the security department by having the developer as a secondary person and another employee who conducted the actual test. Some Suggestion was made while requirement testing was done that has been incorporated.

**User Testing**

During the testing the tester places the role of the individual who desires to penetrate the system. The tester may attempt to acquire passwords through external clerical means and may attack the system with the custom software design to break down any defenses that have been constructed. The tester may also overwhelm the system thereby denying service to other s and may purposely cause system errors to penetrate during recovery and may browse through insecure data, hoping to find key to system entry.

**Validation Testing**

At the end of user testing, software is completely assembled as a package, interfacing errors have now being uncovered and correcting test begins. Software testing and validation are achieved through a serious black box tests that demonstrate conformity with the requirement.

A plan outlines the classes of tests to be conducted and test procedure defines specific cases that will be used to demonstrate conformity with requirements. Both the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met. After the validation test, one of the conditions exists.

They are,

The function or performance characteristics confirm to specification and are accepted. A deviation from the specification is uncovered and a deficiency list is created. The deviation or error discovered at this stage in a project can rarely be corrected prior to scheduled completion. It is necessary to negotiate with the customer to establish methods.

**Integration testing**

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit testing modules and build a program structure that has been dictated by design.

There is often a tendency to attempt non incremental integration; that is to construct the program using “big-bang” approach. All modules are combined in advance. The entire program is tested as a whole. When a set of errors is encountered, correction is difficult because isolation of causes is complicated by the vast expanse of the entire program. Once these errors are corrected new ones appear and the process continues in a seemingly endless loop.

Incremental integration is the antithesis of big-bang approach. The program is constructed and tested in small sequence, where errors are easier to isolate and correct; interfaces are more likely to be tested completely; and a systematic test approach may be applied.

**User acceptance testing**

Acceptance testing involves planning and execution of functional tests, performance test, and stress tests to verify that the implemented system satisfies its requirements. Acceptance tests are typically performed by quality assurance and/or customer organizations. Functional and performance tests are performed to determine the limitations of the system. Typically , acceptance typically, acceptance test will incorporate test cases developed during unit testing and integration testing. Additional test cases are added to achieve desired level of functional , performance, and stress testing of the entire system. Tools of Special importance during acceptance testing include a test Coverage analyzer, a timing analyzer, and a coding standards checker. A test coverage analyzer records the control paths followed for each test case. Timing analyzer reports the time spent in various regions of the source code and different test cases.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.