1. **Software Testing**

* **Software Testing :**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is an exposure of a system to trial input to see whether software meets correct output. Testing cannot be determined whether software meets user’s needs, only whether it appears to confirm to requirements. Testing can show that a system is free of errors, only that it contains error. Testing finds errors, it does not correct errors. Software success is a quality product, on time and within cost. Through testing can reveal critical mistakes. Testing should therefore,

* Validate Performance
* Detects Errors
* Identify Inconsistencies
* **Test Objective :**
* There is strong evidence that effective requirement management leads to overall project cost savings. The three primary reasons for this are,
* Requirement errors typically cost well over 10 times more to repair than other errors.
* Requirement errors typically comprise over 40% of all errors in a software project.
* Small reduction in the number of requirement errors pays big dividend in avoided rework costs and schedule delays.
* System are not designed as entire systems nor are they tested as single systems the analyst must perform both unit and system testing. For this different level testing are use:
* **Unit Testing :**
* In unit testing Module is tested separately and the programmer simultaneously along with the coding of the module performs it.
* In unit testing the analyst tests the programs making up a system. For this reason, unit testing is sometime called program testing. Unit testing gives stress on modules independently of one another, to find errors. This helps the tester in detecting errors in coding and logic that are contained within that module alone. The errors resulting from the interaction between modules are initially avoided.
* Unit testing can be performed from the bottom up, Starting with smallest and lowest-level modules and proceeding one at a time., for each module in Bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embedded within the larger system.
* **System Testing :**
* This is performed after the system is put together. The system is tested against the system requirement to check if all the requirements are met and if the system performs of specify by the requirements.
* Testing is an important function to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully activated. Another reason for system testing is its utility as a user-oriented vehicle before implementation.
* **System Testing Consists of Following Five Steps :**

1. **Program Testing :**

* A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. It is the responsibility of a programmer to have an error free program. At the time of testing the system, there exist two types of errors that should be checked. These errors are

a) Syntax error

b) Logic error

* A syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted key words are common syntax errors. These errors are shown through error messages generated by the Computer.
* A logic error, on the other hand, deals with incorrect data fields out of range items, and invalid combinations. Since the logical errors are not detected by compile, the programmer must examine the output carefully to detect them.
* When program is tested, the actual output is compared with the expected output. When there is a discrepancy, the sequence of the instructions, must be traced to determine the problem. Breaking the program down into self- contained portion, each of which can be checked at certain key points, facilitates the process.

1. **String Testing :**

* Programs are invariably related to one another and interact in total system. Each program is tested to see whether it confirms to related programs in the system. Each part of the system is tested against the entire module with both test and live data before the whole system is ready to be tested.

1. **System Testing :**

* System testing is designed to uncover weaknesses that were not found in earlier tests. This includes forced system failure and validation of total system, as its user in the operational environment will implement it. Under this testing, generally we take low volumes of transactions based on live data. This volume is increased until the maximum level for each transaction type is related. The total system is also tested for recovery and fall back after various major failures to ensure that no data are lost during the emergency.
* **User Acceptance Testing :**
* An acceptance test has the objective if selling the user on the validity and reliability of the system. It verifies that the system’s procedures operate to system specifications and that the integrity of important data is maintained. Performance of and acceptance of the system. After that a comprehensive test report is prepared. This report shows the system’s tolerance, performance range, error rate and accuracy.
* **Acceptance Testing :**
* Finally the acceptance testing is perform to demonstrate the system to the client on the real life data of the client and on the operating system used by the client testing is an externally critical and time consuming activity it requires proper planning the process starts with a test plant that must be performed and specifies guide line for testing than for different unit the test is case specification document is produce in which list of all different test cases with expected outputs are put together during testing the specified test cases are executed and actual results are compared with expected output the final output of the testing phases is the test report and error report.