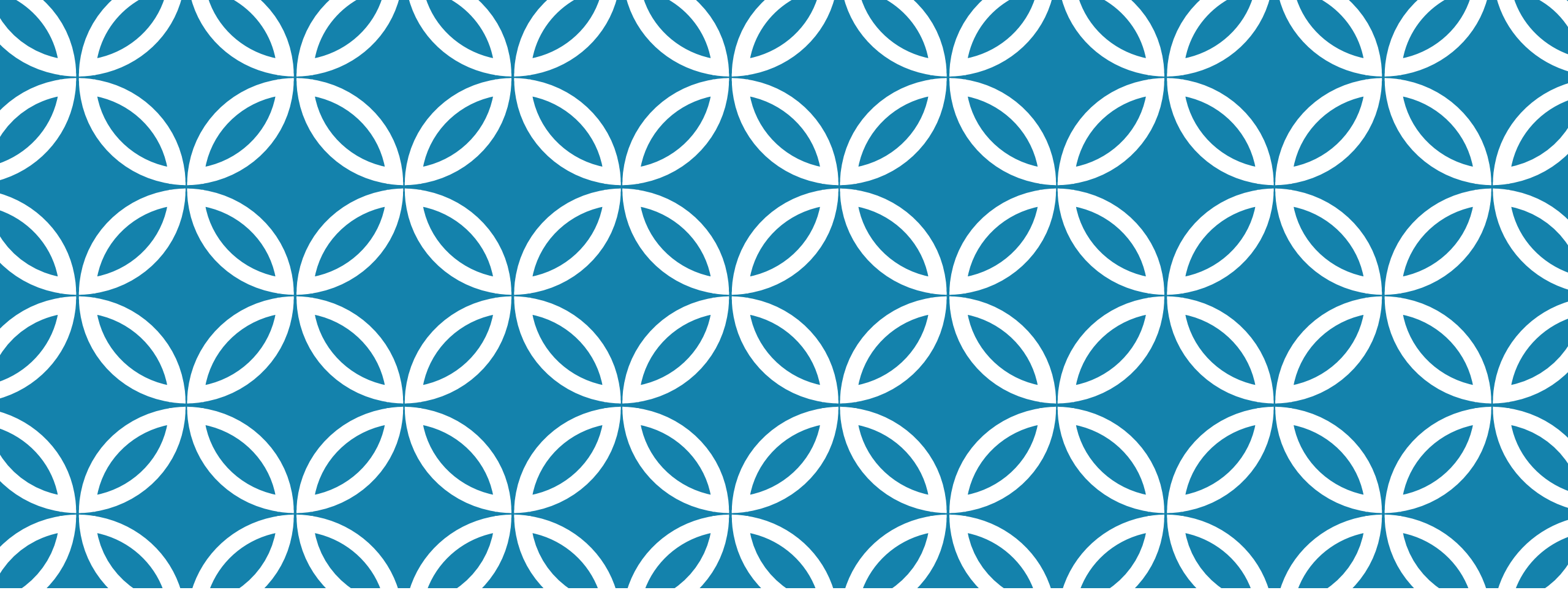


SOFTWARE TESTING (MANUAL)

CHAPTER 04

ELEMENTS OF SOFTWARE QUALITY ASSURANCE (SQA)

Testing: Software testing is a quality control function that has one primary goal—to find errors. The job of SQA is to ensure that testing is properly planned and efficiently conducted for primary goal of software.

Standards: The IEEE, ISO, and other standards organizations have produced a broad array of software engineering standards and related documents. The job of SQA is to ensure that standards that have been adopted are followed and that all work products conform to them.

Reviews and audits: Technical reviews are a quality control activity performed by software engineers for software engineers. Their intent is to uncover errors. Audits are a type of review performed by SQA personnel (people employed in an organization) with the intent of ensuring that quality guidelines are being followed for software engineering work.

Error/defect collection and analysis: SQA collects and analyzes error and defect data to better understand how errors are introduced and what software engineering activities are best suited to eliminating them.

SOFTWARE TESTS - DEFINITION

*"Software testing is a **formal** process carried out by a **specialized testing team** in which a software unit, several integrated software units or an entire software package are examined by running the programs on a computer. All the associated tests are performed according to approved test procedures on approved test cases."*

Formal - Software test plans are part of the project's development and quality plans, scheduled in advance and often a central item in the development agreement signed between the customer and the developer.

Specialized testing team - An independent team or external consultants who specialize in testing.

Running the programs - Any form of quality assurance activity that does not involve running the software, for example code inspection, cannot be considered as a test.

Approved test procedures - The testing process performed according to a test plan and testing procedures that have been approved as conforming to the SQA procedures adopted by the developing organization.

Approved test cases - The test cases to be examined are defined in full by the test plan. No omissions or additions are expected to occur during testing.

SOFTWARE TESTING OBJECTIVES

Direct objectives

- To identify and reveal as many errors as possible in the tested software.
- To bring the tested software, after correction of the identified errors and retesting, to an acceptable level of quality.
- To perform the required tests efficiently and effectively, within budgetary

and scheduling limitations.

Indirect objective

- To compile a record of software errors for use in error prevention (by corrective and preventive actions).

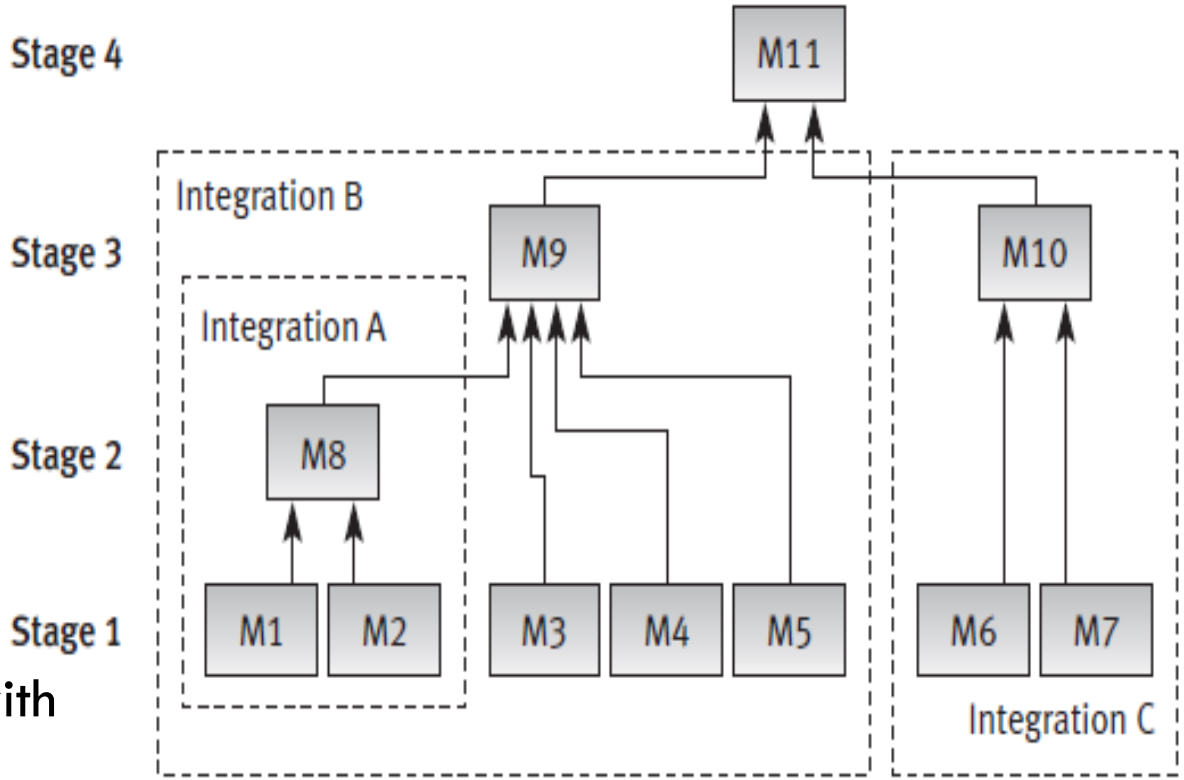
SOFTWARE TESTING STRATEGIES

1. **Big bang testing :** To test the software in its entirety, once the completed package is available.
2. **Incremental testing :** To test the software piecemeal, in modules, as they are completed (unit tests); then to test groups of tested modules integrated with newly completed modules (integration tests). This process continues until all the package modules have been tested. Once this phase is completed, the entire package is tested as a whole (system test). Incremental testing is also performed according to two basic strategies:

   Bottom-up

   Top-down.

INCREMENTAL TESTING : BOTTOM-UP TESTING



Stage 1: Unit tests of modules 1 to 7.

Stage 2: Integration test A of modules 1 and 2, developed and tested in stage 1, and integrated with module 8, developed in the current stage.

Stage 3: Two separate integration tests, B, on modules 3, 4, 5 and 8, integrated with module 9, and C, for modules 6 and 7, integrated with module 10.

Stage 4: System test is performed after B and C have been integrated with module 11, developed in the current stage.

INCREMENTAL TESTING : TOP-DOWN TESTING

Stage 1: Unit tests of module 11.

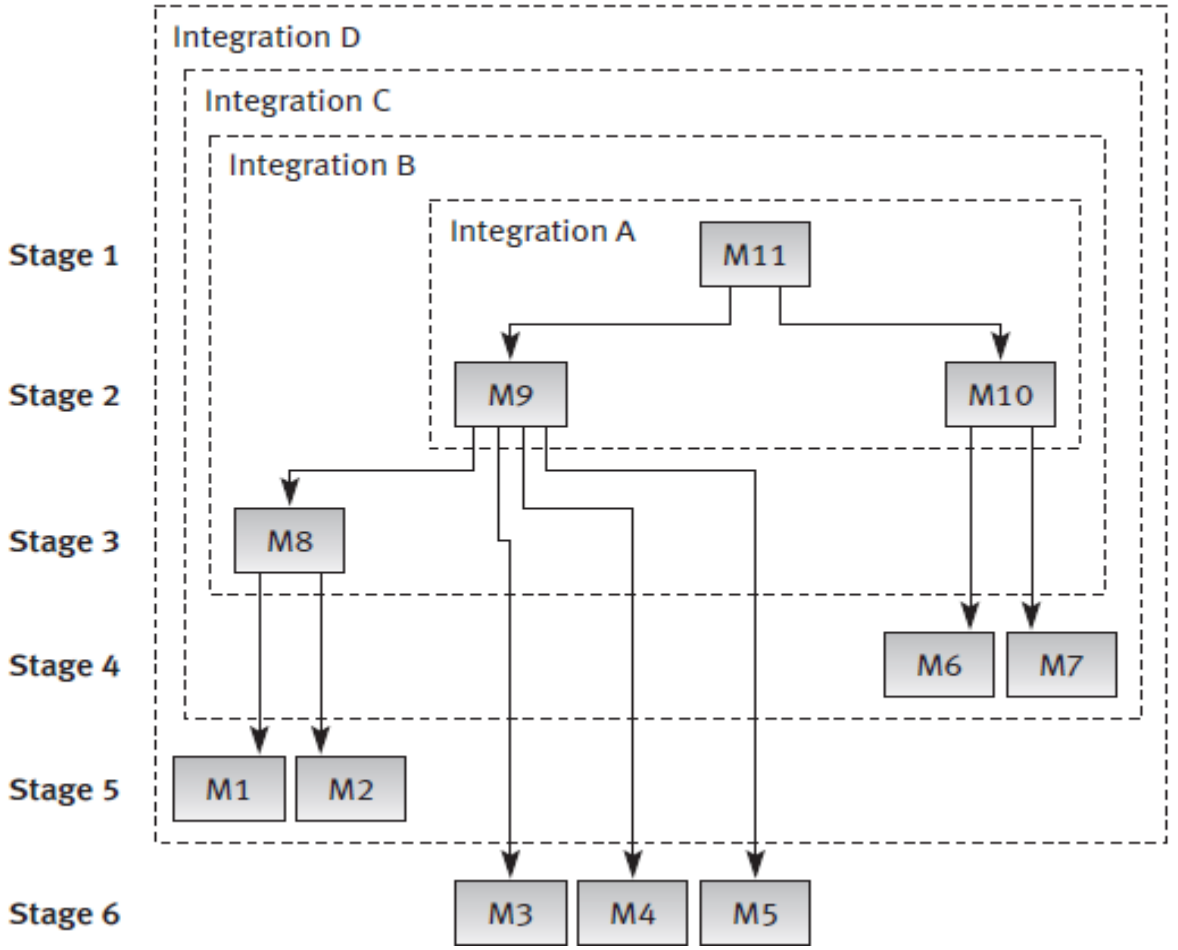
Stage 2: Integration test A of module 11 integrated with modules 9 and 10, developed in the current stage.

Stage 3: Integration test B of A integrated with module 8, developed in the current stage.

Stage 4: Integration test C of B integrated with modules 6 and 7, developed in the current stage.

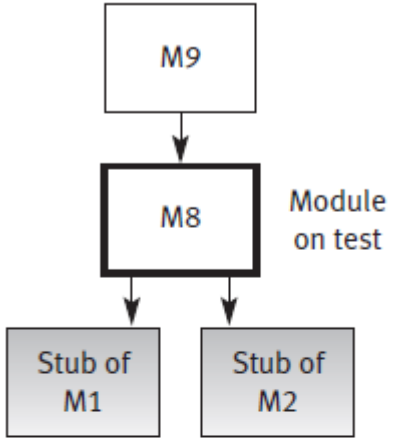
Stage 5: Integration test D of C integrated with modules 1 and 2, developed in the current stage.

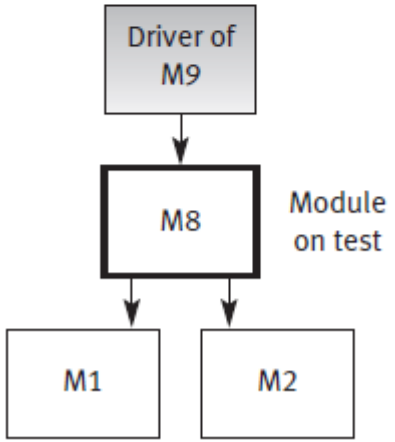
Stage 6: System test of D integrated with modules 3, 4 and 5, developed in the current stage.



INCREMENTAL TESTING : STUBS AND DRIVERS

- Stubs and drivers are software replacement simulators required for modules not available when performing a unit or an integration test.
- A stub (often termed a “dummy module”) replaces an unavailable lower level module.
- A driver is a substitute module but of the upper level module that activates the module tested. The driver is passing the test data on to the tested module and accepting the results calculated by it.
- Substantial savings of resources can be achieved by maintaining a stubs and drivers’ library for future reuse.





SOFTWARE TESTING TYPES

- **Manual testing** is a technique to test the software that is carried out using the functions and features of an application. In manual software testing, a tester tests the software by following a set of predefined test cases. In this testing, testers make test cases for the codes, test the software, and give the final report about that software. Manual testing is time-consuming because it is done by humans, and there is a chance of human errors.

- **Automated Testing** means using special software for tasks that people usually do when checking and testing a software product. Nowadays, many software projects use automation testing from start to end, especially in agile and DevOps methods. This means the engineering team runs tests automatically with the help of software tools.

DIFFERENCES BETWEEN MANUAL TESTING AND AUTOMATION TESTING

Parameters	Manual Testing	Automation Testing
Definition	It is the process of manually testing the software.	It involves the usage of automation tools for executing the test cases.
Test case execution	Test cases are executed manually by the QA Testers.	Test cases are executed automatically using automation tools and scripts.
Efficiency	Manual testing is less-efficient and time-consuming.	Automation testing is more efficient and less time-consuming.
Test coverage	It is difficult to ensure 100% test coverage.	More test coverage can be ensured in automation testing.
Programming knowledge	There is no need to have programming knowledge, only product knowledge is required.	It is important to have programming knowledge to write test scripts.

CHARACTERISTICS OF MANUAL TESTING

1. **Human Performance:** Human testers who use the product the same way as end users conduct manual testing. Manually carrying out test cases, testers provide input and watch the result to find errors.
2. **Investigative Testing:** Exploratory testing, in which testers examine the application without using pre-defined test cases, is a common component of manual testing. This method aids in locating unforeseen problems and evaluating the user experience.
3. **Flexibility:** Throughout the testing process, testers can adjust to changing conditions and requirements. Based on their observations, they can adapt test cases or investigate new ideas.
4. **Examining Complex Situations:** When testing complex scenarios that could be difficult to automate, manual testing works well. Testers can evaluate the interactions between various components and through complex workflows.
5. **Testing User Interfaces (UI):** Manual testing is a good way to assess an application's user interface. From a user's point of view, testers can evaluate the general design, responsiveness appearance, and feel.

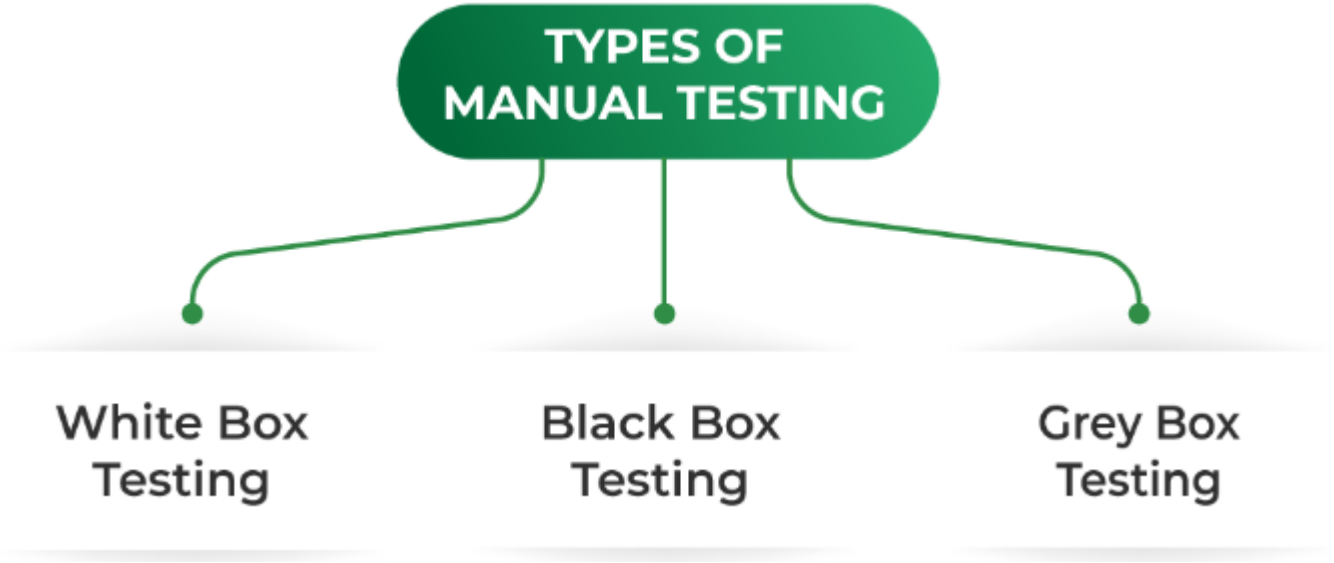
TYPES OF MANUAL TESTING

There are different methods to implement manual testing, but it is broadly classified into three types of manual testing:

1. **Black Box Testing:** In this technique, the tester or the QA analyst will only check the functionality of the particular module or particular method or sometimes the entire application by providing the different test cases manually

2. **White Box Testing:** In this technique, the person will check the internal structure of the system like designs, coding manually. Here, the development team will review the entire coding part line by line to ensure the correctness of the code

Gray Box Testing: This technique is the combination of both white-box testing and black-box testing. Here, the internal structure of the application is partially known by the tester. The tester will check both the internal structure and the functionality of the application manually.



MCCALL'S CLASSIC MODEL FOR CLASSIFICATION OF SOFTWARE QUALITY REQUIREMENTS.

- McCall's classic model for classification of software quality requirements. This model has been extended here to the classification of the tests carried out to ensure full coverage of the respective requirements.
- Application of white box and black box testing in the performance of requirements tests has revealed the advantages and disadvantages of each testing concept.

Factor category	Quality requirement factor	Quality requirement sub-factor	Test classification according to requirements
Operation	1. Correctness	1.1 Accuracy and completeness of outputs, accuracy and completeness of data	1.1 Output correctness tests
		1.2 Accuracy and completeness of documentation	1.2 Documentation tests
		1.3 Availability (reaction time)	1.3 Availability (reaction time) tests
		1.4 Data processing and calculations correctness	1.4 Data processing and calculations correctness tests
		1.5 Coding and documentation standards	1.5 Software qualification tests
	2. Reliability		2. Reliability tests
	3. Efficiency		3. Stress tests (load tests, durability tests)
	4. Integrity		4. Software system security tests
	5. Usability	5.1 Training usability 5.2 Operational usability	5.1 Training usability tests 5.2 Operational usability tests
Revision	6. Maintainability 7. Flexibility 8. Testability		6. Maintainability tests 7. Flexibility tests 8. Testability tests
Transition	9. Portability 10. Reusability 11. Interoperability	11.1 Interoperability with other software 11.2 Interoperability with other equipment	9. Portability tests 10. Reusability tests 11.1 Software interoperability tests 11.2 Equipment interoperability tests

WHITE BOX AND BLACK BOX TESTING FOR THE VARIOUS CLASSES OF TESTS

Test classification according to requirements	White box testing	Black box testing
1.1 Output correctness tests		+
1.2 Documentation tests		+
1.3 Availability (reaction time) tests		+
1.4 Data processing and calculations correctness tests	+	
1.5 Software qualification tests	+	
2. Reliability tests		+
3. Stress tests (load tests and durability tests)		+
4. Software system security tests		+
5.1 Training usability tests		+
5.2 Operational usability tests		+
6. Maintainability tests	+	+
7. Flexibility tests		+
8. Testability tests		+
9. Portability tests		+
10. Reusability tests	+	
11.1 Software interoperability tests		+
11.2 Equipment interoperability tests		+

WHITE BOX TESTING

WHITE BOX TESTING

- White box testing is a software testing technique that involves testing the internal structure and workings of a software application. The tester has access to the source code and uses this knowledge to design test cases that can verify the correctness of the software at the code level.
- White box testing is also known as structural testing or code-based testing, and it is used to test the software's internal logic, flow, and structure. The tester creates test cases to examine the code paths and logic flows to ensure they meet the specified requirements.

WHITE BOX TESTING : FOCUSING ON

- **Path Checking:** Examines the different routes the program can take when it runs. Ensures that all decisions made by the program are correct, necessary, and efficient.
- **Output Validation:** Tests different inputs to see if the function gives the right output each time.
- **Security Testing:** Uses techniques like static code analysis to find and fix potential security issues in the software. Ensures the software is developed using secure practices.
- **Loop Testing:** Checks the loops in the program to make sure they work correctly and efficiently. Ensures that loops handle variables properly within their scope.
- **Data Flow Testing:** Follows the path of variables through the program to ensure they are declared, initialized, used, and manipulated correctly.

WHITE BOX TESTING : DATA PROCESSING AND CALCULATION CORRECTNESS TESTS

Applying the concept of white box testing, which is based on checking the data processing for each test case, immediately raises the question of coverage of a vast number of possible processing paths and the multitudes of lines of code. Two alternative approaches have emerged:

"Path coverage" - to plan the test to cover all the possible paths, where coverage is measured by percentage of paths covered.
Ex : calculate the number of possible paths created by a simple module containing 10 conditional statements, each allowing for only two options (e.g., IF–ELSE).

"Line coverage" - to plan the tests to cover all the program code lines, where coverage is measured by percentage of lines covered.

WHITE BOX TESTING : SOFTWARE QUALIFICATION TESTS

software that qualifies is coded and documented according to standards, procedures and work instructions. This makes it easier for the team leader to check the software, for the replacement programmer to comprehend the code and continue coding tasks, and for the maintenance programmer to correct bugs and/or update or change the program upon request.

1. Does the code fulfill the code structure instructions and procedures, such as module size, application of reused code, etc.?
2. Does the coding style fulfill coding style procedures?
3. Do the internal program documentation and “help” sections fulfill coding style procedures?

WHITE BOX TESTING : REUSABILITY TESTS

- Software reusability substantially reduces project resources requirements and improves the quality of new software systems.

- In doing so, reusability shortens the development period, which by itself benefits the software development organization.

- Reusability testing supports these functions by determining whether the packaging and documentation of the programs and modules listed for reuse conform to the standards and procedures demanded for inclusion in the reusable software library.

WHITE BOX TESTING PROS AND CONS

The main advantages

1. Direct statement-by-statement checking of code enables determination of software correctness as expressed in the processing paths, including whether the algorithms were correctly defined and coded.
2. It allows performance of line coverage follow-up (applying specialized software packages) that provides the tester with lists of lines of code that have not yet been executed. The tester can then initiate test cases to cover these lines of code.
3. It ascertains quality of coding work and its adherence to coding standards.

The main disadvantages

1. The vast resources utilized, much above those required for black box testing of the same software package.
2. The inability to test software performance in terms of availability (response time), reliability, load durability, and other testing classes related to operation, revision and transition factors.

BLACK BOX TESTING

BLACK BOX TESTING

Black box testing is a testing technique in which the internal workings of the software are not known to the tester. The tester only focuses on the input and output of the software. Whereas, White box testing is a testing technique in which the tester has knowledge of the internal workings of the software, and can test individual code snippets, algorithms and methods.

- **Testing objectives:** Black box testing is mainly focused on testing the functionality of the software, ensuring that it meets the requirements and specifications. White box testing is mainly focused on ensuring that the internal code of the software is correct and efficient.
- **Knowledge level:** Black box testing does not require any knowledge of the internal workings of the software, and can be performed by testers who are not familiar with programming languages. White box testing requires knowledge of programming languages, software architecture and design patterns.
- **Testing methods:** Black box testing uses methods like equivalence partitioning, boundary value analysis, and error guessing to create test cases. Whereas, white box testing uses methods like control flow testing, data flow testing and statement coverage.
- **Scope:** Black box testing is generally used for testing the software at the functional level. White box testing is used for testing the software at the unit level, integration level and system level.

BLACK BOX TESTING : FOCUSING ON

1. **Discovers missing functions, incorrect function & interface errors:** This refers to identifying issues where certain expected functions are not present, functions do not work correctly, or there are errors in the way different parts of the software interact with each other

2. **Discover the errors faced in accessing the database:** This involves finding problems that occur when the software tries to interact with the database. These errors could include connectivity issues, incorrect queries, data retrieval problems, or issues with database transactions.

3. **Discovers the errors that occur while initiating & terminating any functions:** This refers to detecting issues that happen during the startup or shutdown processes of various functions within the software. Problems in these areas can lead to incomplete initialization or improper termination, potentially causing crashes, data corruption, or other malfunctions.

4. **Discovers the errors in performance or behavior of software:** This entails identifying performance-related issues or unexpected behaviors in the software. These could include slow response times, inefficient resource usage, memory leaks, or any other anomalies that affect the overall performance and user experience of the software.

BLACK BOX TESTING : DOCUMENTATION TESTS

- Documentation testing, though neglected in many cases, should be considered as important as code testing or design documents inspections.
- An erroneous user manual or programmer manual can lead to mistakes during program operation and maintenance that may incur damages equivalent in severity to those caused by software bugs. Documents includes,

1. Functional descriptions of the software system.
2. Installation manual.
3. User manual.
4. Programmer manual.

BLACK BOX TESTING : AVAILABILITY TESTS

- Availability is defined as reaction time – the time needed to obtain the requested information or the time required for firmware installed in computerized equipment to react.
- Availability is of highest importance in on-line applications of frequently used information systems. The failure of firmware software to meet availability requirements can make the equipment useless.
- It is relatively difficult to test availability, especially for information systems planned to serve a large population of users, and for real-time systems planned to treat high-frequency events. This difficulty stems from the need to carry out the tests under regular operation load as well as under maximal load conditions as specified in the requirement specifications.

BLACK BOX TESTING : RELIABILITY TESTS

- The software system reliability requirement deals with features that can be translated as events occurring over time, such as average time between failures (e.g., 500 hours), average time for recovery after system failure (e.g., 15 minutes), or average downtime per month (e.g., 30 minutes per month).
- Reliability requirements are to be in effect during regular full-capacity operation of the system. It should be noted that in addition to the software factor, reliability tests also relate to the hardware, the operating system and the data communication system effects.
- Like availability testing, reliability testing is especially difficult as it requires operation of the full range of software applications conducted under regular workload conditions.

BLACK BOX TESTING : STRESS TESTS

- The class of stress tests subsumes two main types of tests: load tests and durability tests.

1. Stress tests: load tests

Load tests relate to the functional performance of the system under maximal operational load: maximal transactions per minute, hits per minute to an Internet site and the like. Load tests, which are usually conducted for loads higher than those indicated in the requirements specification, are of utmost importance for software systems planned to serve simultaneously a large population of users.

2. Stress tests: durability tests

Durability tests are carried out in physically extreme operating conditions such as high temperatures, humidity

BLACK BOX TESTING : SOFTWARE SYSTEM SECURITY TESTS

- Software security components of software systems are aimed at preventing unauthorized access to the system or parts of it, detection of unauthorized access and the activities performed by the penetration, and the recovery of damages caused by unauthorized penetration cases. The main security issues dealt with by these tests are:

1. Access control, where the usual requirement is for control of multi-level access (usually by a password mechanism). Of special importance here are the firewall systems that prevent unauthorized access to Internet sites.
2. Backup of databases and software files and recovery in cases of system failure.
3. Logging of transactions, system usage, access trials, and so forth.

BLACK BOX TESTING : TRAINING USABILITY TESTS

- When large numbers of users are involved in operating a system, training usability requirements are added to the testing agenda.

- The scope of training usability is defined by the resources needed to train a new employee, in other words, how many hours of training are required for a new employee to achieve a defined level of acquaintance with the system or to reach a defined hourly production rate.

- The details of these tests, like any other, are based on system characteristics but, more importantly here, on employee characteristics. The results of the tests should inspire a sophisticated plan of training courses and follow-up as well as improved directions for software system operation.

BLACK BOX TESTING : OPERATIONAL USABILITY TESTS

- The focus of this class of tests is the operator's productivity, that is, those aspects of the system that affect the performance regularly achieved by system operators, or that are applied mainly for information systems that serve many users.

- These tests are of high importance in cases where the workings of the system can affect substantially the productivity of its users. The implementation of this class of tests deals mainly with the productivity, quantitatively and qualitatively.

- Naturally these aspects are highly important for systems that serve as the main vocational tools for a large group of users.

BLACK BOX TESTING PROS AND CONS

Advantages of Black Box Testing

1. The tester does not need to have more functional knowledge or programming skills to implement the Black Box Testing.
2. It is efficient for implementing the tests in the larger system.
3. Tests are executed from the user's or client's point of view.

.

Disadvantages of Black Box Testing

1. There is a possibility of repeating the same tests while implementing the testing process.
2. Without clear functional specifications, test cases are difficult to implement.
3. It is difficult to execute the test cases because of complex inputs at different stages of testing.
4. Sometimes, the reason for the test failure cannot be detected.
5. It does not reveal the errors in the control structure.

END