

Software Testing and Types

Presented By:

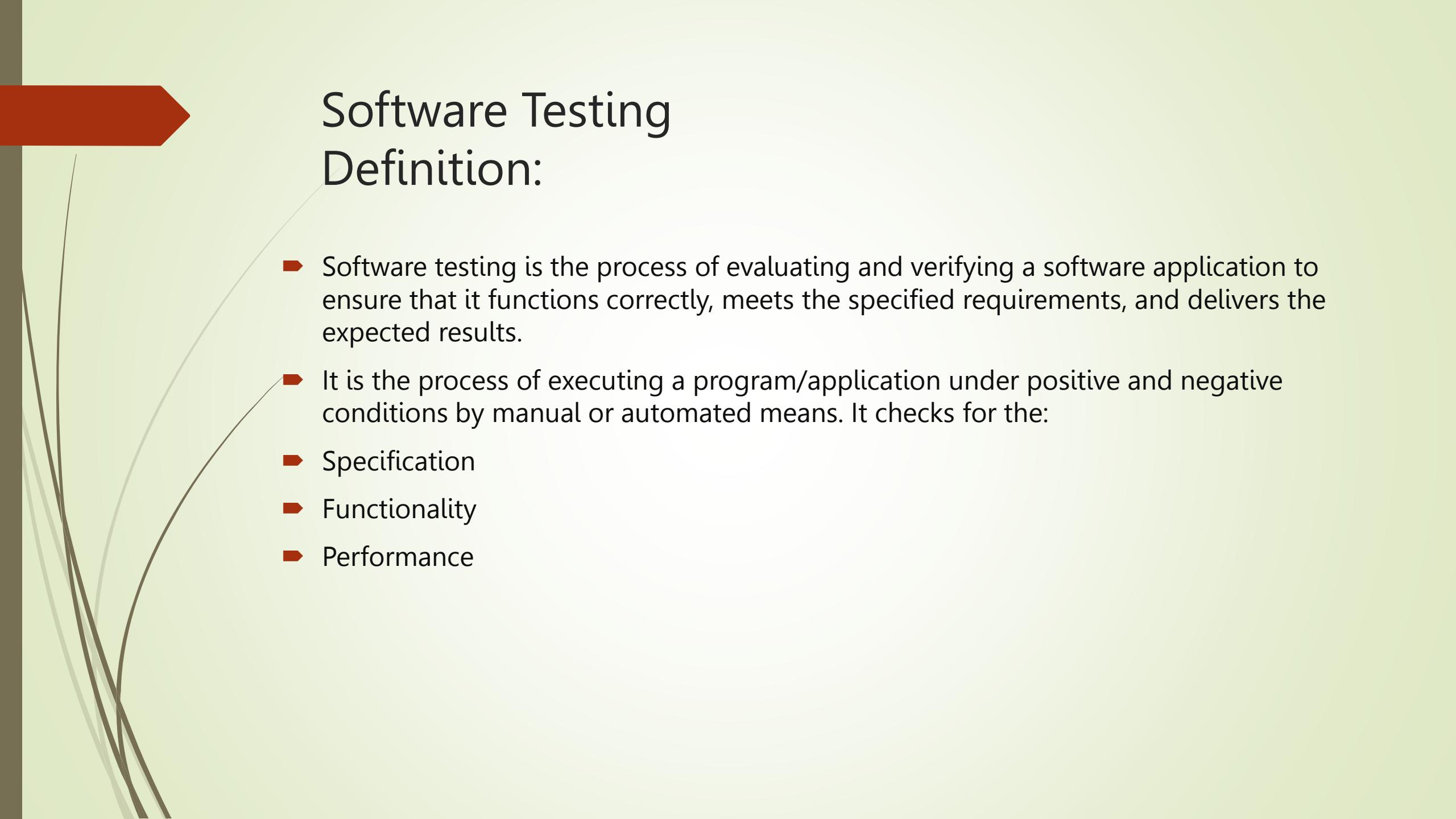
Akash Bhaskar

Amith

Gurukiran L Korabu

Mofiz T

Nandeesh B H



Software Testing

Definition:

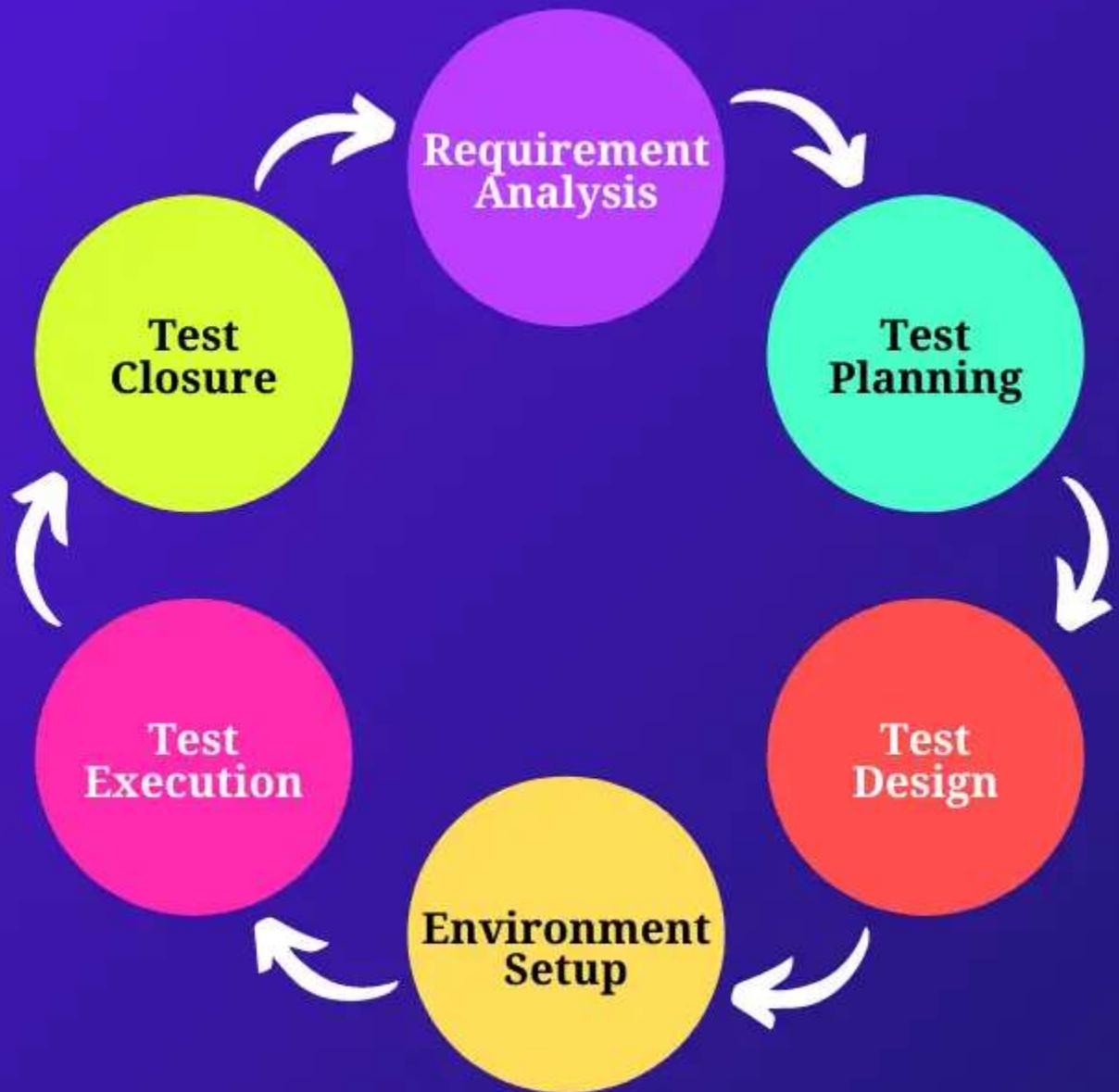
- ▶ Software testing is the process of evaluating and verifying a software application to ensure that it functions correctly, meets the specified requirements, and delivers the expected results.
- ▶ It is the process of executing a program/application under positive and negative conditions by manual or automated means. It checks for the:
 - ▶ Specification
 - ▶ Functionality
 - ▶ Performance

Objectives

- ▶ Uncover as many errors as possible in a given product.
- ▶ Demonstrate a given software product matching its requirements specifications.
- ▶ Validate the quality of a software testing using the minimum cost and efforts.
- ▶ Generate a high quality test cases, perform effective tests, and issue correct and helpful reports.

▶ Error, Bug, Fault and Failure

- ▶ Error: It is a human action that produces the incorrect result leading to a fault.
- ▶ Bug: The presence of error at the time of execution of the software.
- ▶ Fault: State of software caused by an error.
- ▶ Failure: Deviation of the software from its expected result.



SOFTWARE TESTING LIFE CYCLE



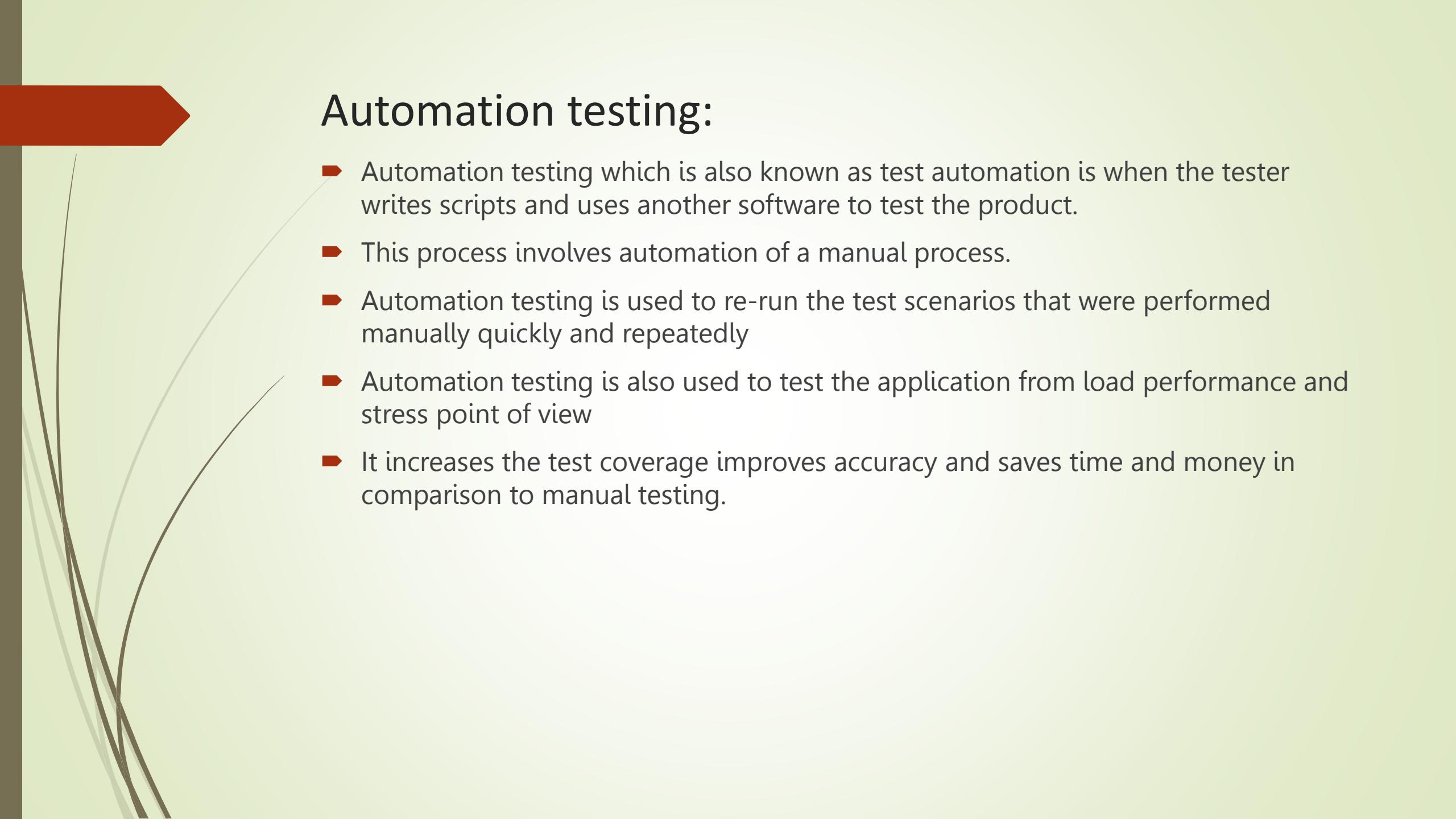
Test Plan: It is a systematic approach to test a system i.e. software. The plan typically contains a detailed understanding of what the eventual testing workflow will be.

- ▶ **Test case:** It is a specific procedure of testing a particular requirement.
- ▶ **It will include:**
 - ▶ Identification of specific requirement tested
 - ▶ Test case success / failure criteria
 - ▶ Specific steps to execute test
 - ▶ Test data



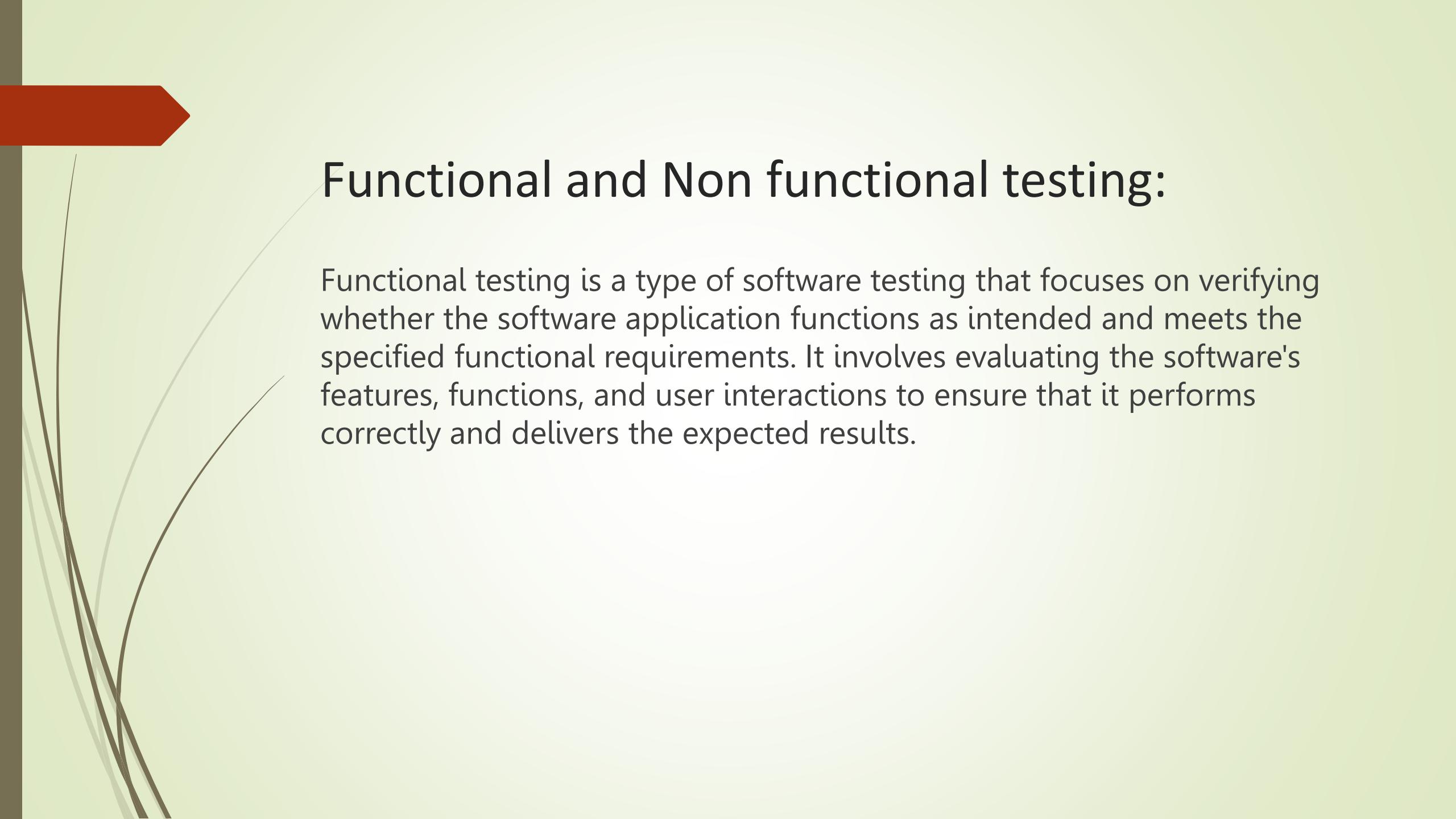
Types of Testing: Manual Testing:-

- ▶ Manual testing includes testing a software manually i.e. without using any automated tool or any script.
- ▶ In this type, the tester takes over the role of an end user and tests the software to identify any unexpected behaviour or bug.
- ▶ There are different stages for manual testing
- ▶ Unit testing
- ▶ Integration testing
- ▶ System testing
- ▶ User acceptance testing



Automation testing:

- ▶ Automation testing which is also known as test automation is when the tester writes scripts and uses another software to test the product.
- ▶ This process involves automation of a manual process.
- ▶ Automation testing is used to re-run the test scenarios that were performed manually quickly and repeatedly
- ▶ Automation testing is also used to test the application from load performance and stress point of view
- ▶ It increases the test coverage improves accuracy and saves time and money in comparison to manual testing.



Functional and Non functional testing:

Functional testing is a type of software testing that focuses on verifying whether the software application functions as intended and meets the specified functional requirements. It involves evaluating the software's features, functions, and user interactions to ensure that it performs correctly and delivers the expected results.

Non functional Testing:

- ▶ Testing an application from its non functional attributes
- ▶ Non functional testing involves testing a software from the requirements which are non functional in nature.

Types of non functional testing:

- ▶ Performance testing
- ▶ Load testing
- ▶ Stress testing
- ▶ Usability testing

Levels of testing

Unit testing: Unit testing is a fundamental level of testing in software development where individual components, modules, or units of code are tested in isolation to verify their correctness.

Purpose:

- ▶ Detecting defects
- ▶ Isolating faults
- ▶ Ensuring reliability
- ▶ Facilitating Refactoring
- ▶ Supporting documentation
- ▶ Promoting collaboration

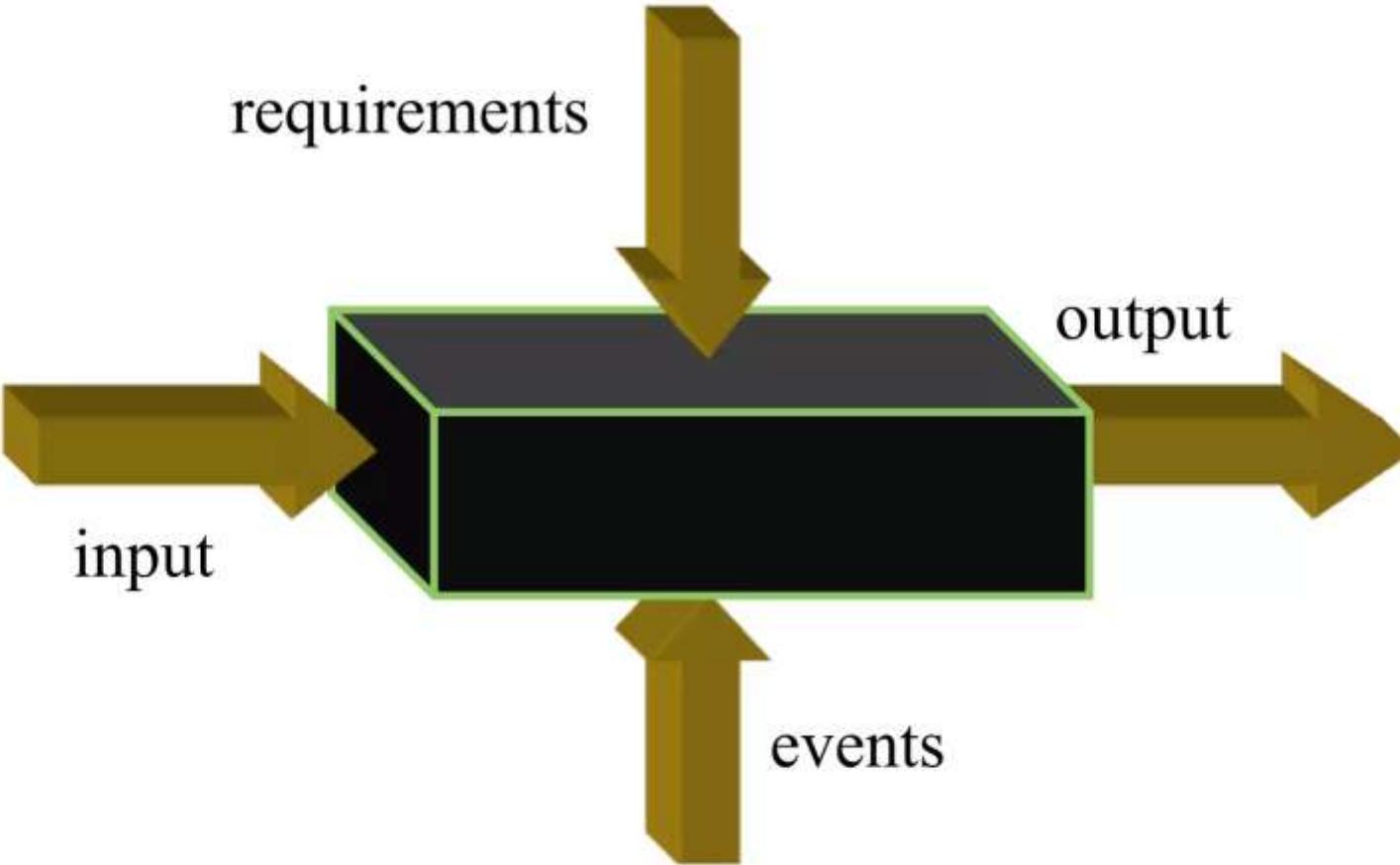
Black box testing

- No knowledge of internal program design or code required.
- Tests are based on requirements and functionality.

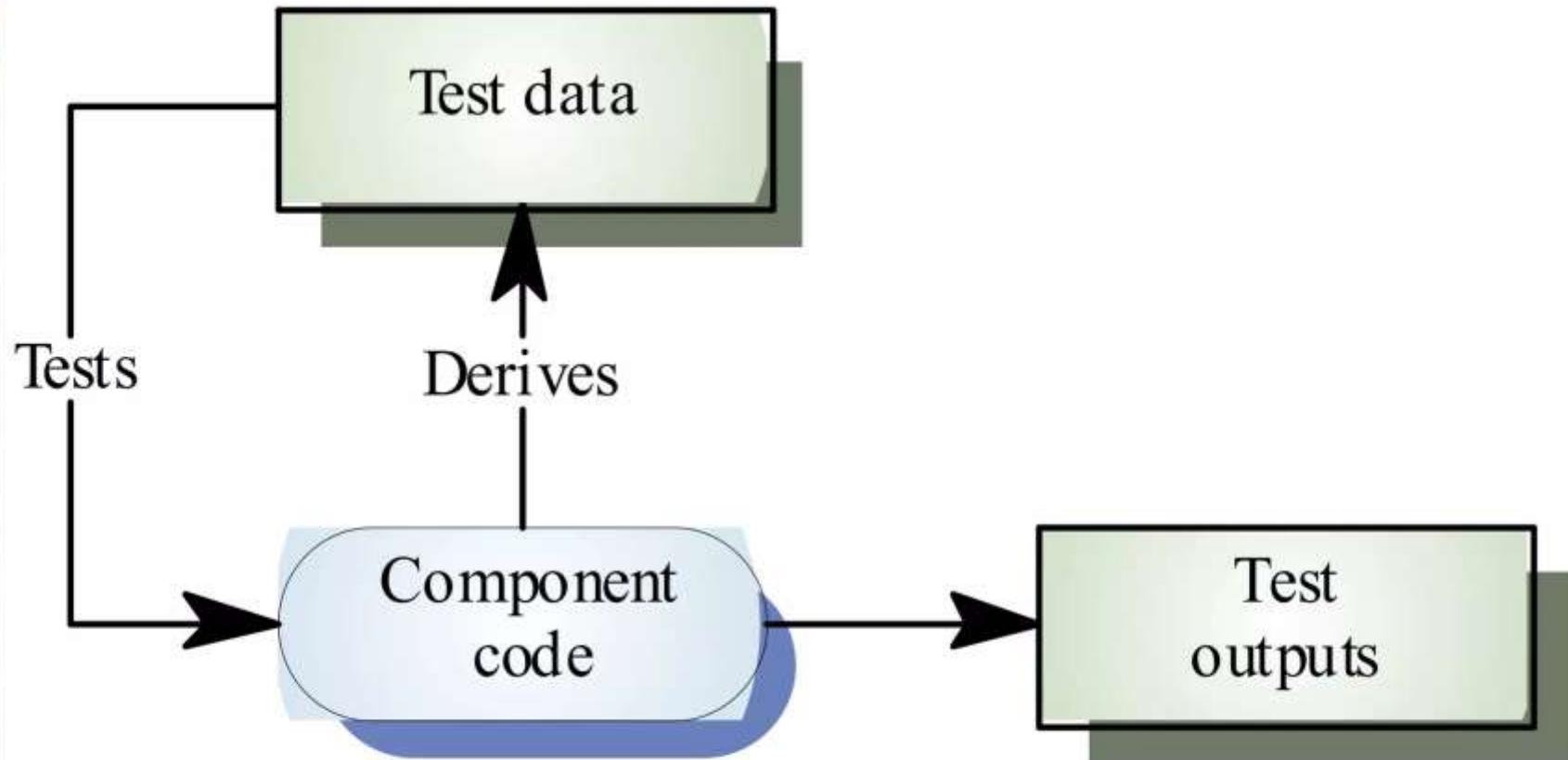
White box testing

- Knowledge of the internal program design and code required.
- Tests are based on coverage of code statements, branches, paths, conditions.

Black box testing



White box testing



Integration testing:

- ▶ Once all the modules have been unit tested, integration testing is performed.
 - ▶ It is a systematic testing.
 - ▶ Produce tests to identify errors associated with interfacing.
-
- ▶ Types:
 - ▶ Big bang integration testing
 - ▶ Top down integration testing
 - ▶ Bottom up integration testing
 - ▶ Mixed integration testing

System Testing:

- ▶ The system as a whole is tested to uncover requirement errors.
- ▶ Verifies that all the system elements work properly and that overall system function and performance has been achieved.

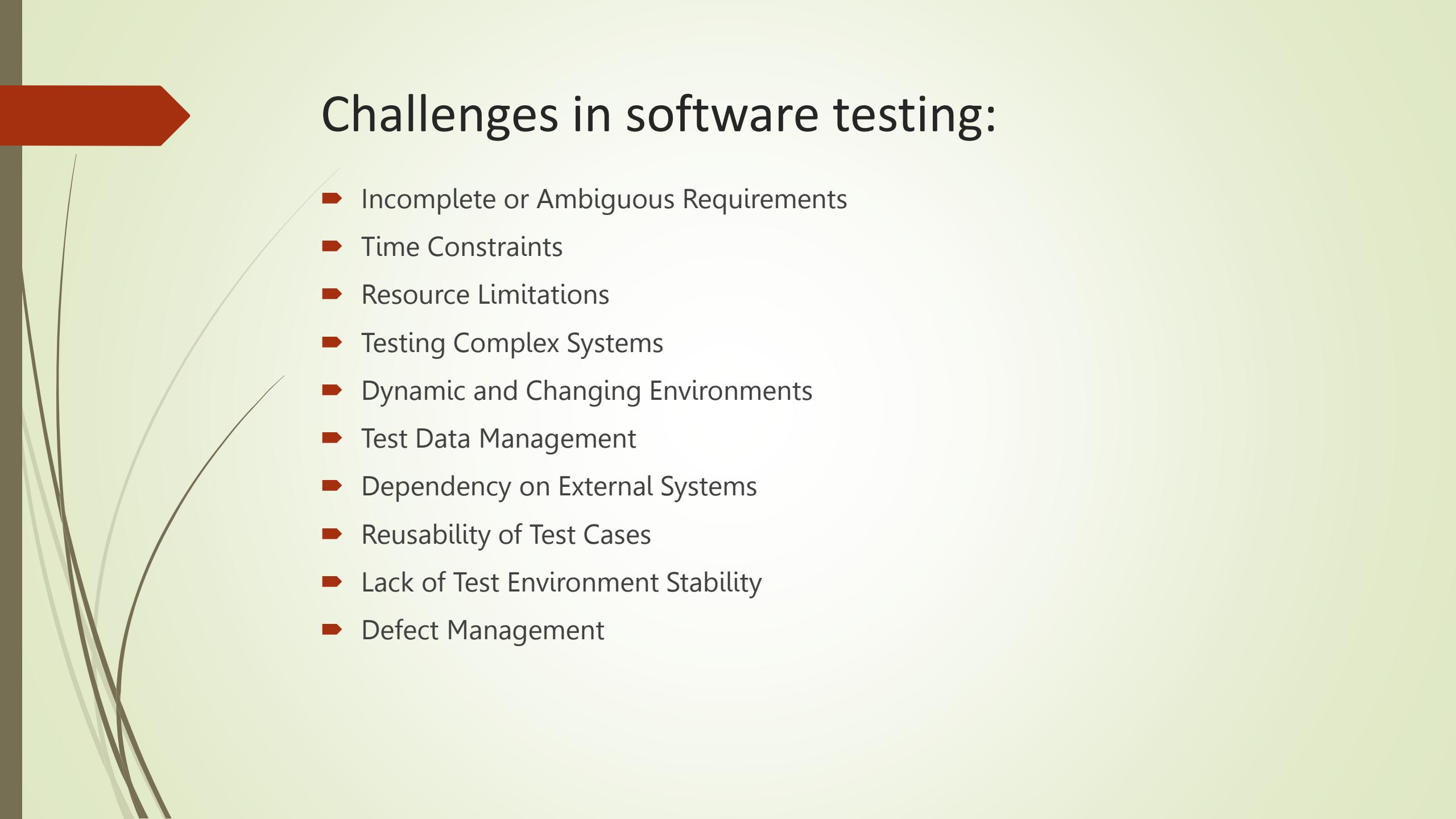
▶ TYPES:

- ▶ Alpha testing
- ▶ Beta testing
- ▶ Acceptance testing
- ▶ Performance testing



Best practices in software testing:

- ▶ Importance of clear requirements and testable specifications
- ▶ Test case design techniques (e.g., Equivalence Partitioning, Boundary Value Analysis)
- ▶ Test data management and test environment setup
- ▶ Collaboration between development and testing teams



Challenges in software testing:

- ▶ Incomplete or Ambiguous Requirements
- ▶ Time Constraints
- ▶ Resource Limitations
- ▶ Testing Complex Systems
- ▶ Dynamic and Changing Environments
- ▶ Test Data Management
- ▶ Dependency on External Systems
- ▶ Reusability of Test Cases
- ▶ Lack of Test Environment Stability
- ▶ Defect Management

Conclusion:

- ▶ Software testing is an indispensable aspect of the software development lifecycle that ensures the delivery of high-quality, reliable, and user-friendly software products.
- ▶ Throughout this presentation, we explored various aspects of software testing, including its definition, types, levels, and techniques.
- ▶ We also discussed the challenges faced by testing teams in today's dynamic and fast-paced development environments.
- ▶ In conclusion, by investing in a robust testing strategy, leveraging the right testing techniques, and addressing the challenges proactively, we can significantly improve the software's quality and meet user expectations.
- ▶ Embracing a comprehensive and iterative testing approach will ultimately lead to enhanced customer satisfaction, reduced maintenance costs, and the successful delivery of innovative and reliable software products.
- ▶ Remember, testing is not just a phase; it is a continuous process that should be integrated into the entire software development lifecycle for optimal results.