

# System Testing

# Intended Learning Outcomes

At the end of the lesson, you should be able to

1. Understand the importance of system testing;
2. Differentiate the different types of system testing; and
3. Identify the tools recommended for system testing.

# Course Outline:

- System Testing definition
- Importance and Characteristics of System testing
- Guidelines of System testing
- Key areas and pre-requisites of System testing
- Different types of System testing
- Tips to perform the System test
- Difference between error, defect and failure in Software testing
- System testing's contribution to software development

# What is System Testing?

- A level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems.
- System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.
- A black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

# Examples of System Testing

1. Take a web application as example. Suppose we are doing testing on a web application of a school and in this web application there are many modules like Teacher Module, Staff Module, Parent Module, Student Module, and Admin Module and so on. Now we have to do System Testing on this web application by maintaining certain steps.
2. A car manufacturer does not produce the car as a whole car. Each component of the car is manufactured separately, like seats, steering, mirror, break, cable, engine, car frame, wheels etc. After manufacturing each item, it is tested independently whether it is working the way it is supposed to work and that is called Unit testing. Now, when each part is assembled with another part, that assembled combination is checked if assembling has not produced any side effect to the functionality of each component and whether both components are working together as expected and that is called integration testing.

# Why is System Testing necessary?

1. Software testing is really required to point out the defects and errors that were made during the development phases.
2. It's essential since it makes sure that the customer finds the organization reliable and their satisfaction in the application is maintained.
3. It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence.
4. Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.

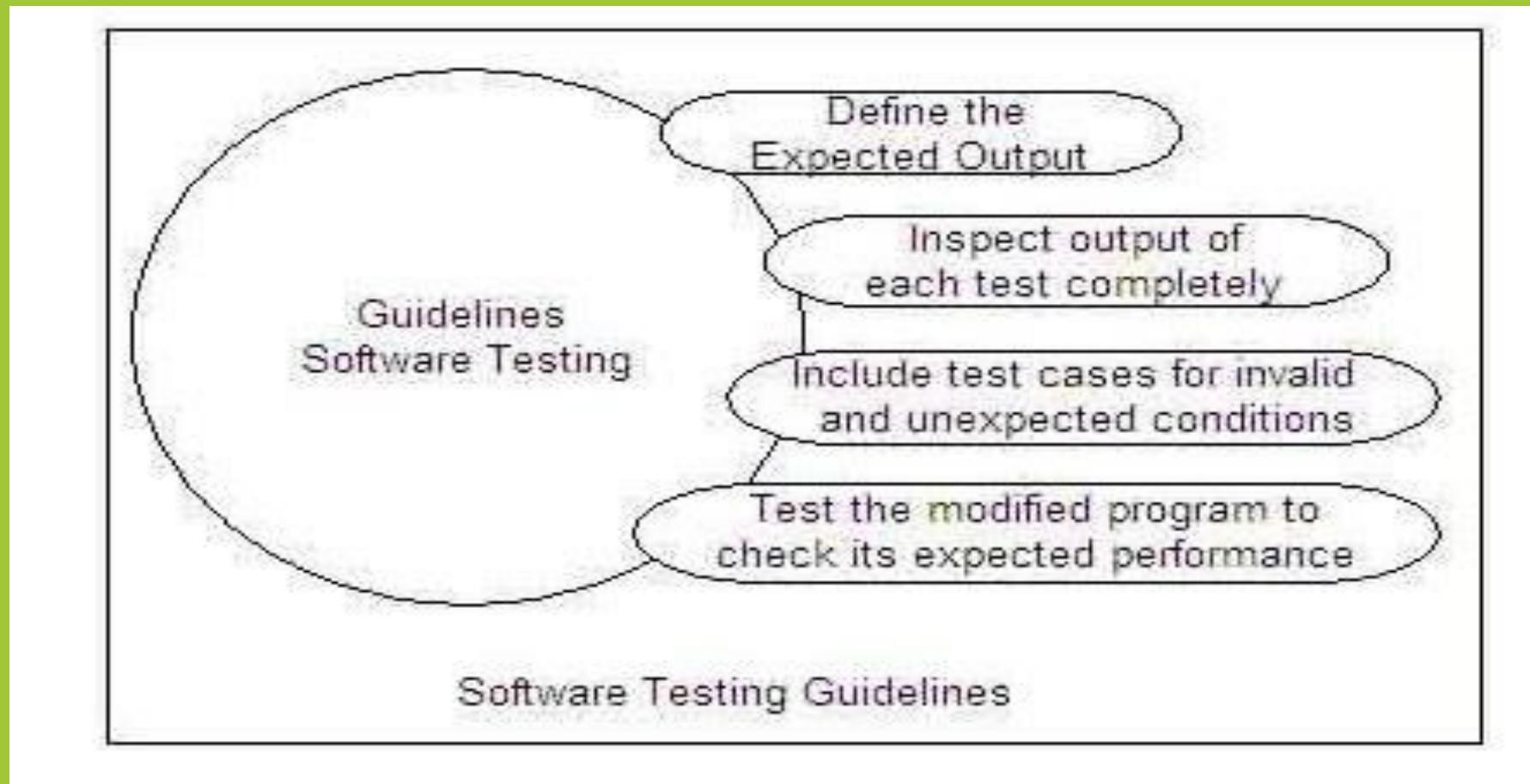
# Why is System Testing necessary?

5. Testing is required for an effective performance of software application or product.
6. It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.
7. It's required to stay in the business.

# Characteristics of System Test

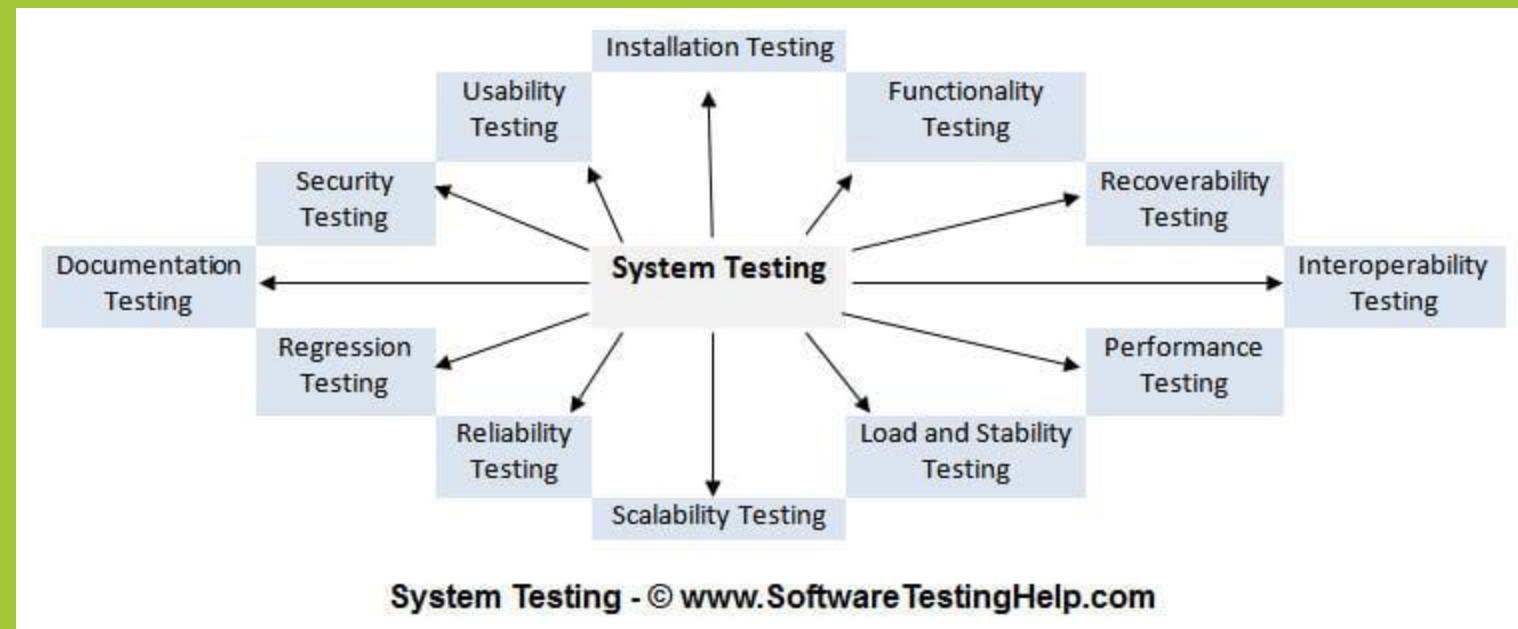
1. High probability of detecting errors.
2. No redundancy.
3. Choose the most appropriate test.
4. Moderate.

# Guidelines of System Testing



# Key Areas of System Testing

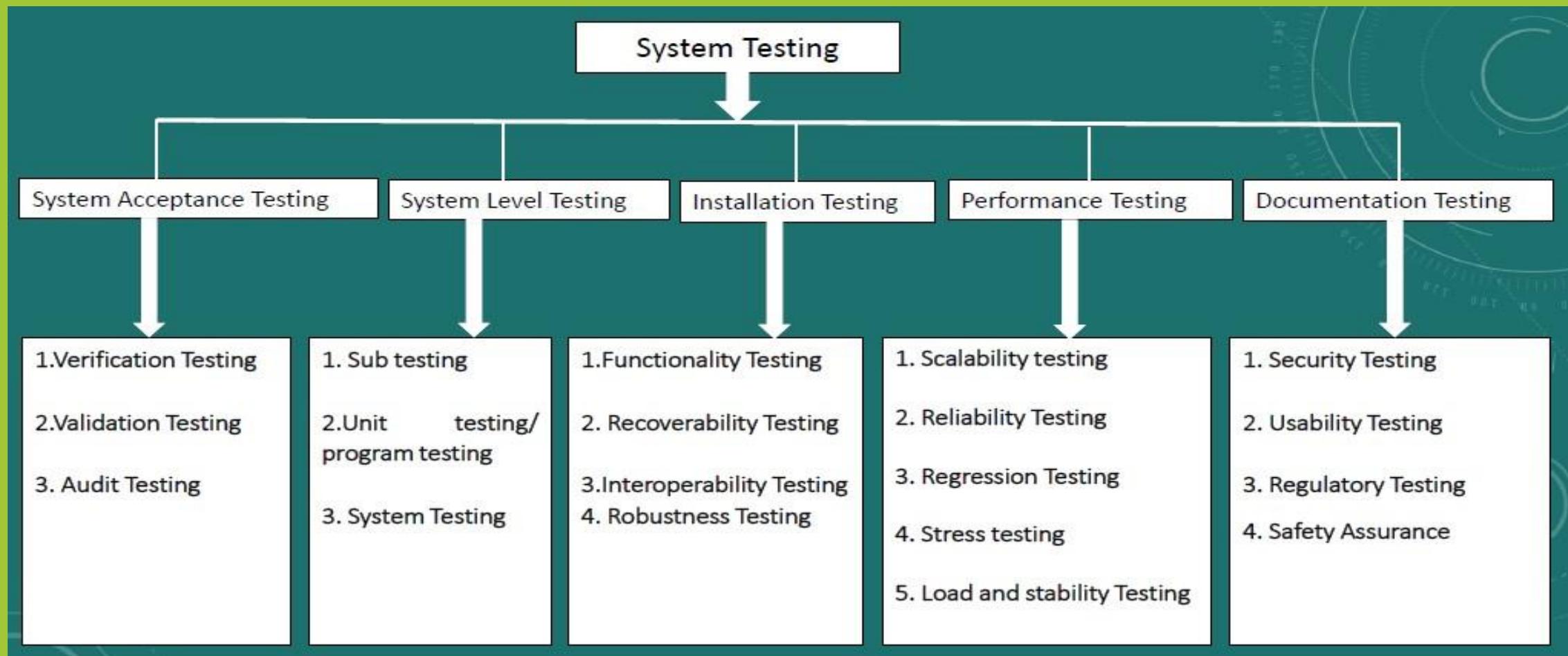
- Performance
- Security
- Recovery
- Interface
- Install-ability
- Usability
- Documentation
- Load/Stress



# Prerequisites of System Testing

- Team should make sure the software is unit tested.
- Integration testing should already be implemented on the product.
- The software should be developed completely.
- Before implementing the process of system testing the team should ensure that the testing environment is ready.

# Types of System Testing



# Types of System Testing

## ***System Acceptance Testing***

- This is conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.

### ***1. Verification Testing***

Procedures that attempt to determine that the product of each phase of the development process is an implementation of a previous phase. Each verification activity is a phase of the testing life cycle.

### ***2. Validation Testing***

It is the process of evaluating a system or component during or at the development process to determine whether it satisfies specified requirements. This testing the system runs in a live environment using real data.

# Types of System Testing

During this **validation** a number of items are testing, like-

- **System performance:** Is the throughput and response time for processing adequate to meet a normal processing workload?
- **Peak workload processing performance:** Can the system handle the workload during peak processing period?
- **Methods and procedures testing:** During conversion, methods and procedures for the new system will be put to their first real test.
- **Backup and recovery testing:** We have full sized computer files and databases with real data, we should test all backup and recovery. We should simulate a data lost disaster and test the time required to recover from that disaster.

# Types of System Testing

## **3. Audit Testing**

It certifies that the system is free of errors and is ready to be placed operation.

## ***System Level Testing***

There are three levels of testing to be performed. Such as –

### **1. Sub testing:**

It is the test performed on individual events or modules of a program. In other words, it is the testing of an isolated subset of program.

### **2. Unit Testing/ program testing:**

Unit testing is performed at an extremely low level of system development. As a result it is rarely observed by clients and so doesn't get appropriate recognition of its importance.

# Types of System Testing

## 3. *System Testing*

It ensures that application programs written and tested in isolation work when they are integrated into the total system. The process of testing a hardware and software system to verify that the system meets its specified requirements. This testing process is conducted by the testing teams in both development and target environment.

# Types of System Testing

## *Installation Testing*

To make sure that product / software can be installed on specific or support defined system, can be configured and can be brought into an operational mode.

### **1. *Functionality Testing***

It provide comprehensive testing over the full range of the requirements, within the capabilities of the system. To make sure that functionality of product are working as per the requirements defined, within the capabilities of the system.

### **2. *Recoverability Testing***

To make sure how well the system recovers from various input errors and other failure situations.

# Types of System Testing

## 3. *Interoperability Testing*

To make sure whether the system can operate well with third party products or another kind of inter-operability tests is called (backward) compatibility tests.

- **Compatibility tests** verify that the system works the same way across different platforms, operating systems, data base management.
- **Backward compatibility** tests verify that the current software build flawlessly works with older version of platforms.

## 4. *Robustness Testing*

It is designed to verify how sensitive a system towards an error input or how the system behave in error situations in a changed operational environment.

# Types of System Testing

## ***Performance Testing***

- To measure the performance characteristics of the system, e.g. throughput and response time, under various conditions.

### ***1. Scalability Testing***

To make sure system's scaling abilities in various terms like user scaling, geographic scaling and resource scaling.

- Tests are designed to verify that the system can scale up to its engineering limits.
- Scaling tests are conducted to ensure that the system response time remains the same, or increases by a small amount, as the number of users are increased.

# Types of System Testing

## 2. *Reliability Testing*

To make sure system can be operated for longer duration without developing failures. Reliability tests are designed to measure the ability of the system to remain operational for long periods of time.

## 3. *Regression Testing*

To make sure system's stability as it passes through integration of different sub systems and maintenance tasks. The main idea in regression testing is to verify no defect has been introduced into the unchanged portion of a system due to changes made elsewhere in the system.

# Types of System Testing

## 4. *Stress Testing*

The goal of stress testing is to evaluate and determine the behavior of a component while the offered load is in excess of its designed capacity.

## 5. *Load and Stability Testing*

- Tests are designed to ensure that the system remains stable for a long period time under full load.
- Load and stability testing typically involves exercising the system with virtual users and measuring the performance to verify whether the system can handle the anticipated load.
- This kind of testing helps one to understand the ways the system will fare in live situations.

# Types of System Testing

## *Documentation Testing*

- It means verifying the technical accuracy and readability of the user manuals, tutorials and other help related documents.
1. ***Security Testing:*** To make sure system does not allow unauthorized access to data and resources.
  2. ***Usability Testing:*** To make sure that system is easy to use, learn and operate.
  3. ***Regulatory Testing:*** In this category, the final system is shipped to the regulatory bodies in those countries where the product is expected to be marketed

# Types of System Testing

## 4. Safety Assurance

There are two basic tasks performed by a safety assurance engineering team:

- Provide methods for identifying, tracking, evaluating, and eliminating hazards associated with a system.
- Ensure that safety is embedded into the design and implementation in a timely and cost effective manner, such that the risk created by the user/operator error is minimized.

# System Testing Complete Process

**1. Create a Test Plan:** The initial step of the process involves test plan creation, where in the lead or test manager define the scope and objective of testing, determines the strategies, decides between manual and automated testing, define the exit and entry criteria, assigns roles and responsibilities, among other things.

To test the system completely, the initial step is to define the expectations of the test cases. Also, understand the system's Real-Time usage. The system Test plan may vary from organization to organization as well as based on the project plan, test strategy & main test plan. Standard points considered while creating the Test Plan -

- Goals & Objective
- Scope
- Critical areas Area to focus
- Test Deliverable
- Testing Strategy
- Testing Schedule
- Entry and exit criteria
- Suspension & resumption criteria for system testing
- Test Environment
- Roles and Responsibilities

# System Testing Complete Process

2. Test Case Creation: It is from this step that the process of testing is initiated by the team.

Creating a different type of test cases, such as functional, regression, load, GUI, scalability, installation, etc., is the next step. Test cases must cover all functional, non-functional, technical, and user-interface requirements. It is very much related functional test case writing. In the test case writing, write the test scenarios and use cases.

## Sample Test Case Format

Test Case ID	Test Suite Name	How to Test?	Test Data	Expected Result	Actual Result	Pass/Fail
--------------	-----------------	--------------	-----------	-----------------	---------------	-----------

# System Testing Complete Process

- 3. Select Test Data:** Once the test cases are developed by the team, they work together to select or create the required test data, which plays a critical role in test execution.
- 4. Test Case Execution:** Finally, the test cases made earlier executed by the team, who continually monitor the process and record any errors or issues found by them during the process. Additionally, the output of the testing also recorded.
- 5. Bug Reporting and Fixation:** It is in this stage of the process that the team reports all the recorded bugs and issues to the concerned member of the team.
- 6. Repeat the Test Cycle (If Required):** After all the issues and bugs are resolved and fixed, the team repeats the test cycle to get the expected results.

# Reasons for Performing System Testing

1. It ensures that the product meets the quality standards.
2. Verifies that the software system meets the functional, technical and business requirements requested by the customer/client.
3. Performs end to end testing of the software product, which prevents system failures and crashes during its implementation to the live environment.
4. It is performed in an environment that is similar to the production environment, which enables the developers as well as the concerned stakeholders to get an idea of the user's reaction to the product.

# Reasons for Performing System Testing

5. It plays a significant role in delivering a quality product to the end users.
6. It is during this stage of Software Testing Life Cycle (STLC) that the Application Architecture and Business Requirements are tested.
7. Ensures that the input provided to the system offers expected output/result.

# Difference Between System and Acceptance Testing

<b>System Testing</b>	<b>Acceptance Testing</b>
1 System testing is the testing of a system as a whole. End to end testing is performed to verify that all the scenarios are working as expected.	Acceptance testing is done to verify if the product meets customer requirement.
2 System testing includes functional & non-functional testing and is performed by the testers.	Acceptance testing is functional testing and is performed by testers as well as a customer.
3 Testing is performed using test data created by the testers.	Real/Production data is used while performing acceptance testing.
4 A system as a whole is tested to check the functionality & Performance of the product.	Acceptance testing is done to verify that business requirement i.e. it solves the purpose what customer is looking for.
5 Defects found in the testing can be fixed.	Any defects found while acceptance testing is considered as a failure of the Product.
6 System and system integration testing are types for System testing.	Alpha and Beta testing come under acceptance testing.

# Tips to Perform the System Test

1. Replicate real-time scenarios rather than doing ideal testing as the system is going to be used by an end-user and not by the trained tester.
2. Verify the system's response in various terms as the human does not like to wait or to see the wrong data.
3. Install and configure the system as per the documentation because that is what the end-user is going to do.
4. Involving people from different areas like business analysts, developers, testers, customers can send in a better system.
5. Regular testing is the only way to make sure that the littlest change in the code to fix the bug has not inserted another critical bug into the system.

# What is a failure in Software testing?

Failures can also be caused because of the other reasons also like:

- Because of the environmental conditions as well like a radiation burst, a strong magnetic field, electronic field or pollution could cause faults in hardware or firmware. Those faults might prevent or change the execution of software.
- Failures may also arise because of human error in interacting with the software, perhaps a wrong input value being entered or an output being misinterpreted.
- Finally, failures may also be caused by someone deliberately trying to cause a failure in the system.

# Difference between Error, Defect and Failure in software testing:

**Error:** The mistakes made by programmer is known as an ‘Error’. This could happen because of the following reasons:

- Because of some confusion in understanding the functionality of the software
- Because of some miscalculation of the values
- Because of misinterpretation of any value, etc.

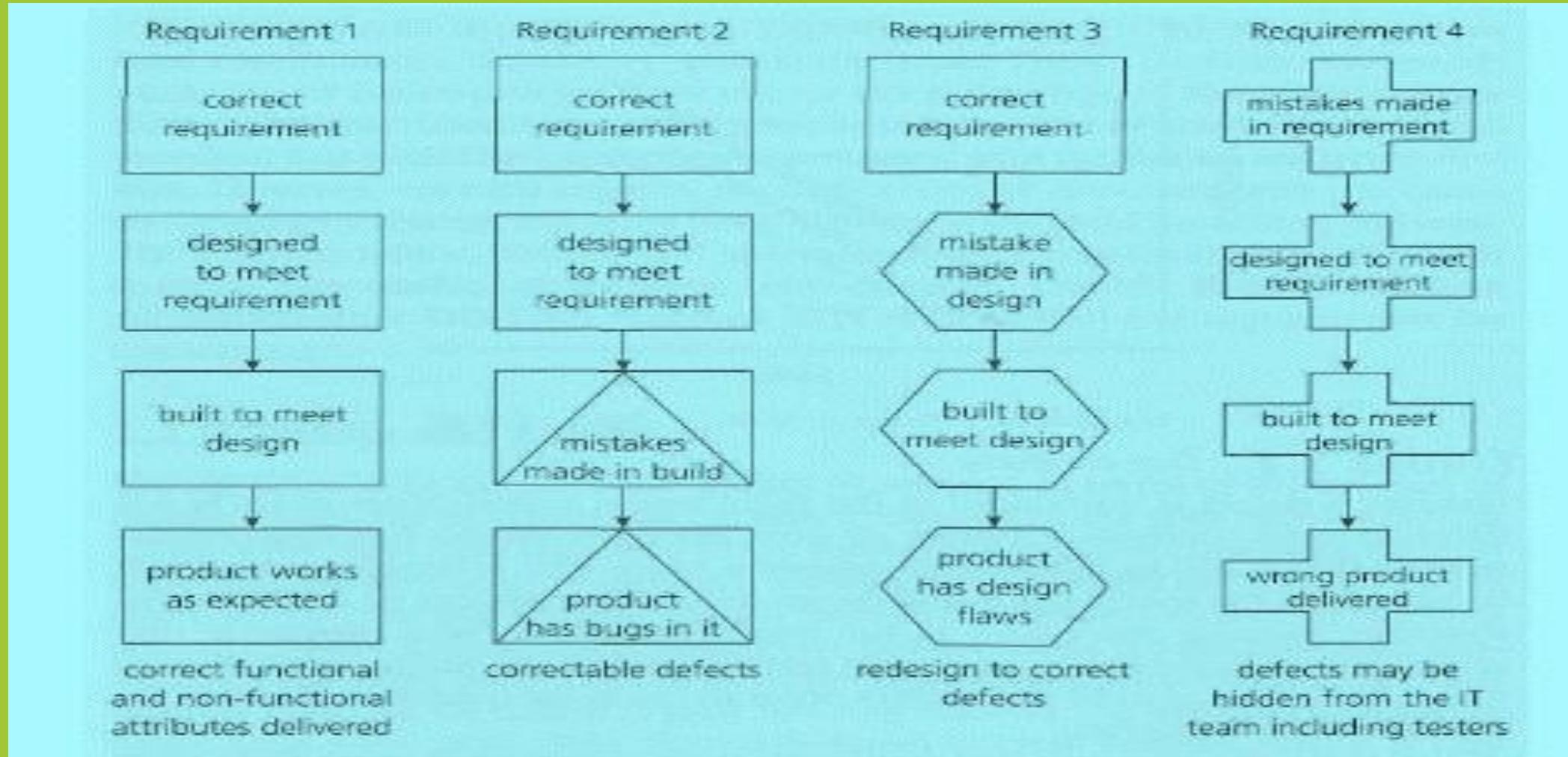
**Defect:** The bugs introduced by programmer inside the code are known as a defect. This can happen because of some programmatic mistakes.

**Failure:** If under certain circumstances these defects get executed by the tester during the testing then it results into the failure which is known as software failure.

# When do defects in software testing arise?

1. The person using the software application or product may not have enough knowledge of the product.
2. Maybe the software is used in the wrong way which leads to the defects or failures.
3. The developers may have coded incorrectly and there can be defects present in the design.
4. Incorrect setup of the testing environments.

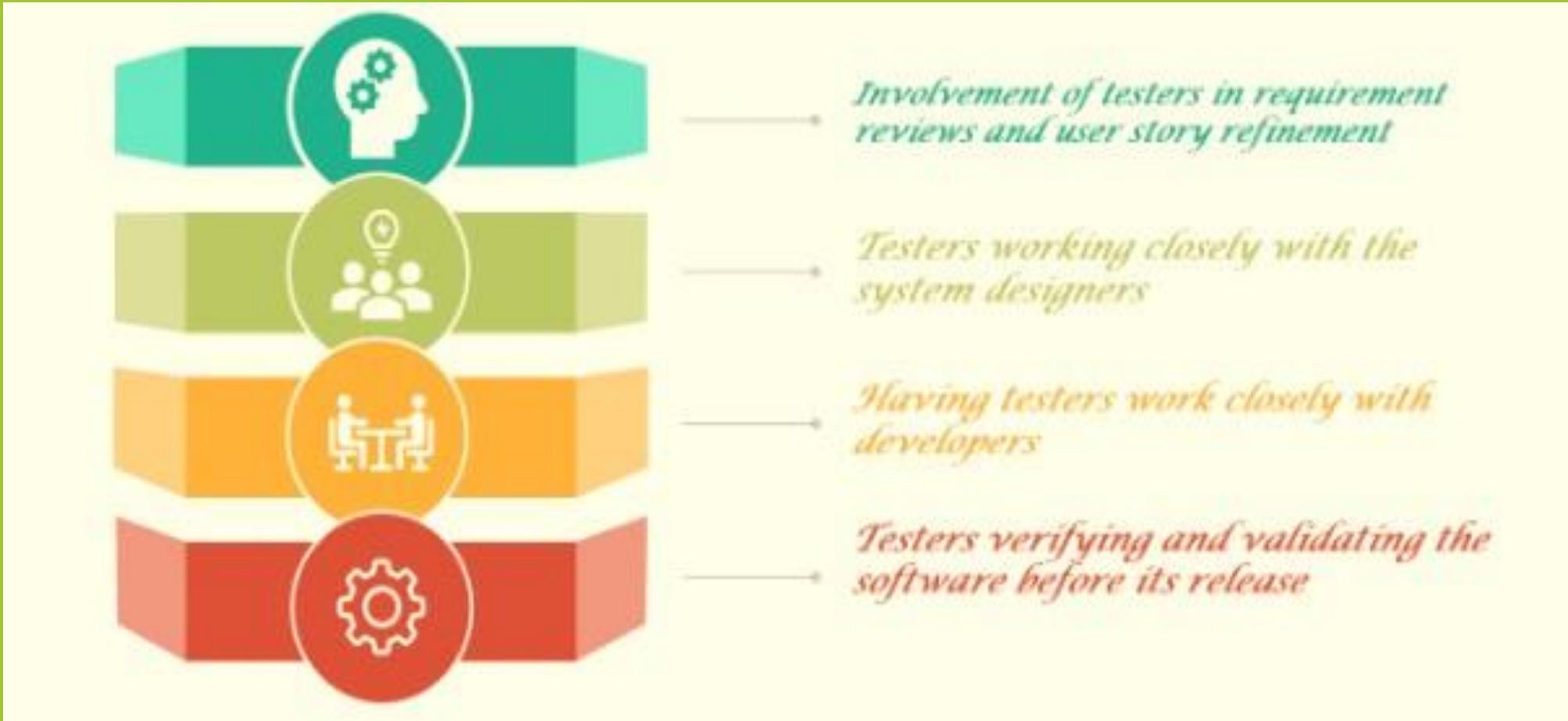
# Examples – Types of Errors and Defects



# From where do defects and failures in software testing arise?

1. Errors in the specification, design and implementation of the software and system
2. Errors in use of the system
3. Environmental conditions
4. Intentional damage
5. Potential consequences of earlier errors

# System testing's contribution to the success of the overall project



# Videos and links

How to write a TEST CASE? Software Testing Tutorial

<https://www.youtube.com/watch?v=BBmA5Qp6Ghk&t=15s>

How to Write Test Cases: Sample Template with Examples By Thomas Hamilton

<https://www.guru99.com/test-case.html>

# **Activity:**

Given below is the list of some common types of Software Testing. Define each and provide an example.

## **Functional Testing types include:**

Unit Testing, Integration Testing, System Testing, Sanity Testing, Smoke Testing, Interface Testing, Regression Testing, Beta/Acceptance Testing

## **Non-functional Testing types include:**

Performance Testing, Load Testing, Stress Testing, Volume Testing, Security Testing, Compatibility Testing, Install Testing, Recovery Testing, Reliability Testing, Usability Testing, Compliance Testing, Localization Testing

Thank you for listening.