Introduction to Software Testing

simplilearn

A Day in the Life of a MEAN Stack Developer

In this sprint, Joe has test the application with different tools. As he is new to testing, he is trained on the basics of testing with different aspects of a test engineer.

In this lesson, we will learn how to solve this real-world scenario to help Joe complete his training on testing basics.

Learning Objectives

By the end of this lesson, you will be able to:

- Explain the software testing life-cycle
- Explain different types of testing
- Explain bug reporting
- Efficiently do test planning

Overview of Testing

Software Testing: Overview

Software testing is the process of evaluating a system or its components to ensure that the software system is defect-free.

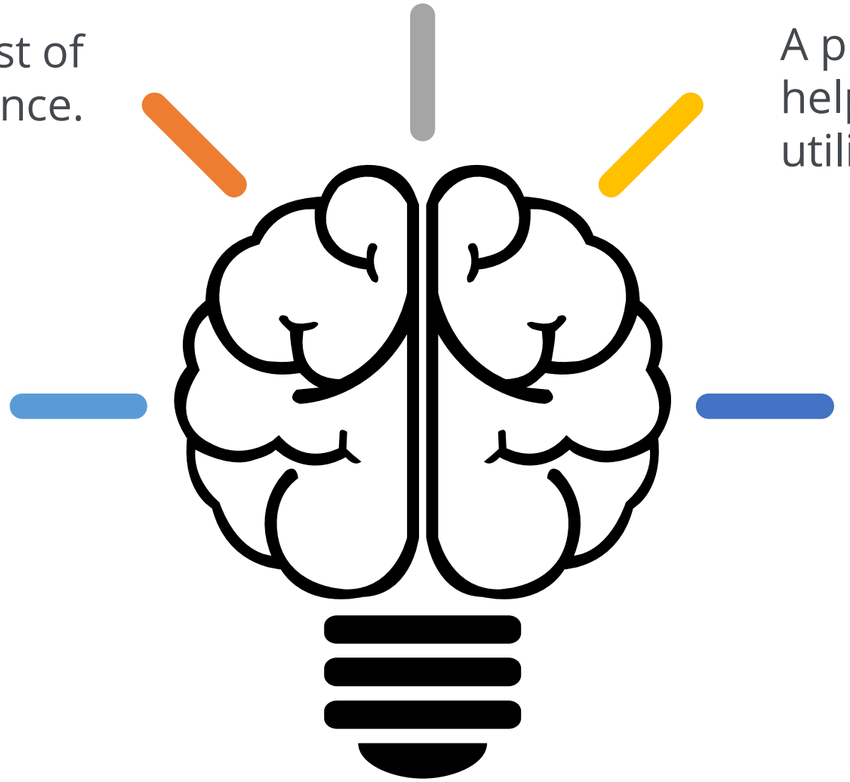
- Testing: Checking if the tangible result matches the projected or the expected output
- High analytical skills required to test an application for all possible use cases with minimum test cases.

Need for Software Testing

It is required to prevent the failure of critical functionalities like, medicine prescription failure of Therac 25.

It helps us to save the cost of maintenance.

A properly tested software also helps in optimal resource utilization.



It helps us to fail first and deliver bug-free products to customers.

A thoroughly tested software improves the performance of the product.

Product Failure Examples

1. Software bug in the airbag detectors led to the recall of over 1 million cars from the market by Nissan.
2. Starbucks also had to close almost 60 percent of its stores in U.S and Canada due to a software glitch in their P.O.S system and had to serve coffee for free as they were unable to print bills.
3. Bloomberg terminal crashed due to a software bug. This affected more than 300,000 traders on financial market. As a result, the government had to postpone a 3bn pound debt sale.
4. Due to a software bug in 2015, CareFusion's Alaris pump has to be recalled as the pump that autmatically delivers medicine to patients, delayed the infusion.

Software Quality

Quality in software testing refers to the amount of conformance for all the external and internal expectations.

Definition by IEEE:

- The degree to which a system, component, or process meets the specified requirements.

- The degree to which a system, component, or process meets customer or user needs or expectations.

Definition by ISTQB:

- **quality:** The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations.

- **software quality:** The totality of functionality and features of a software product that bears on its ability to satisfy stated or implied needs.

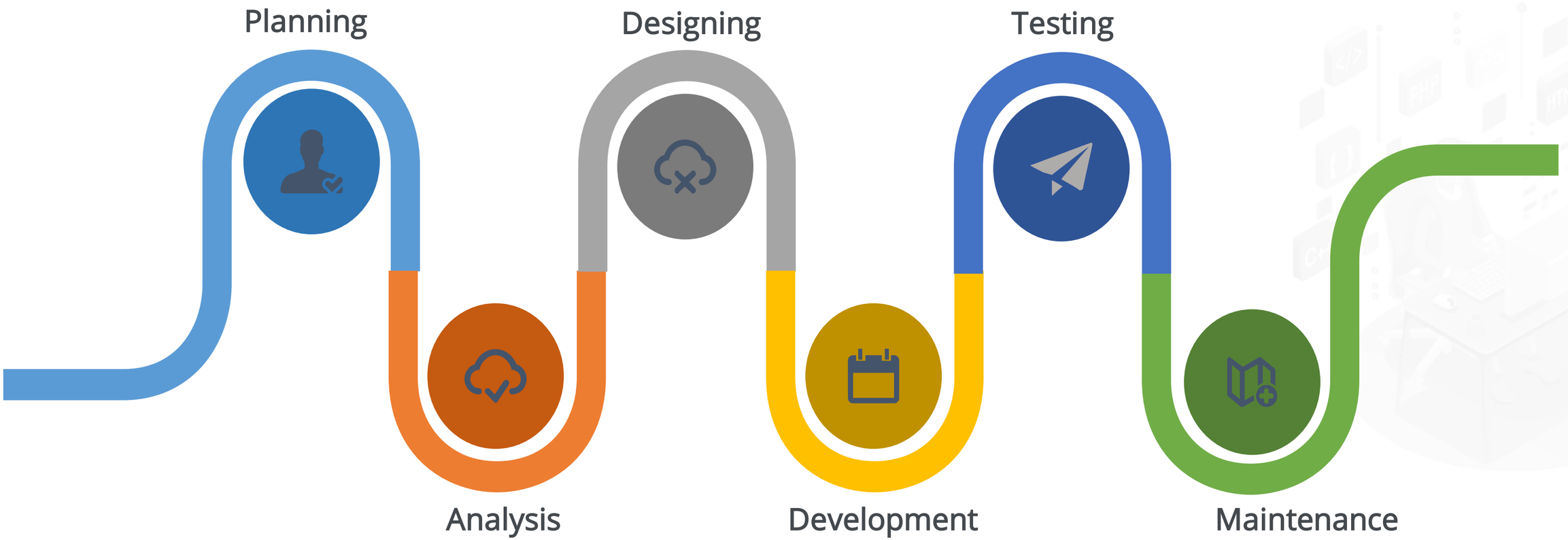
Role of a Test Engineer

- Reads the spec document and understands what needs to be tested.
- Determines the methods to test the different components of a software.
- Lists down the resources required for testing the software and informs the leads about the same.
- Develops and prioritizes test cases.
- Executes the test cases and generates a report, that includes defects, their severity, and priority.
- For every change made, regression tests are run by the test engineer.

Software Life Cycle

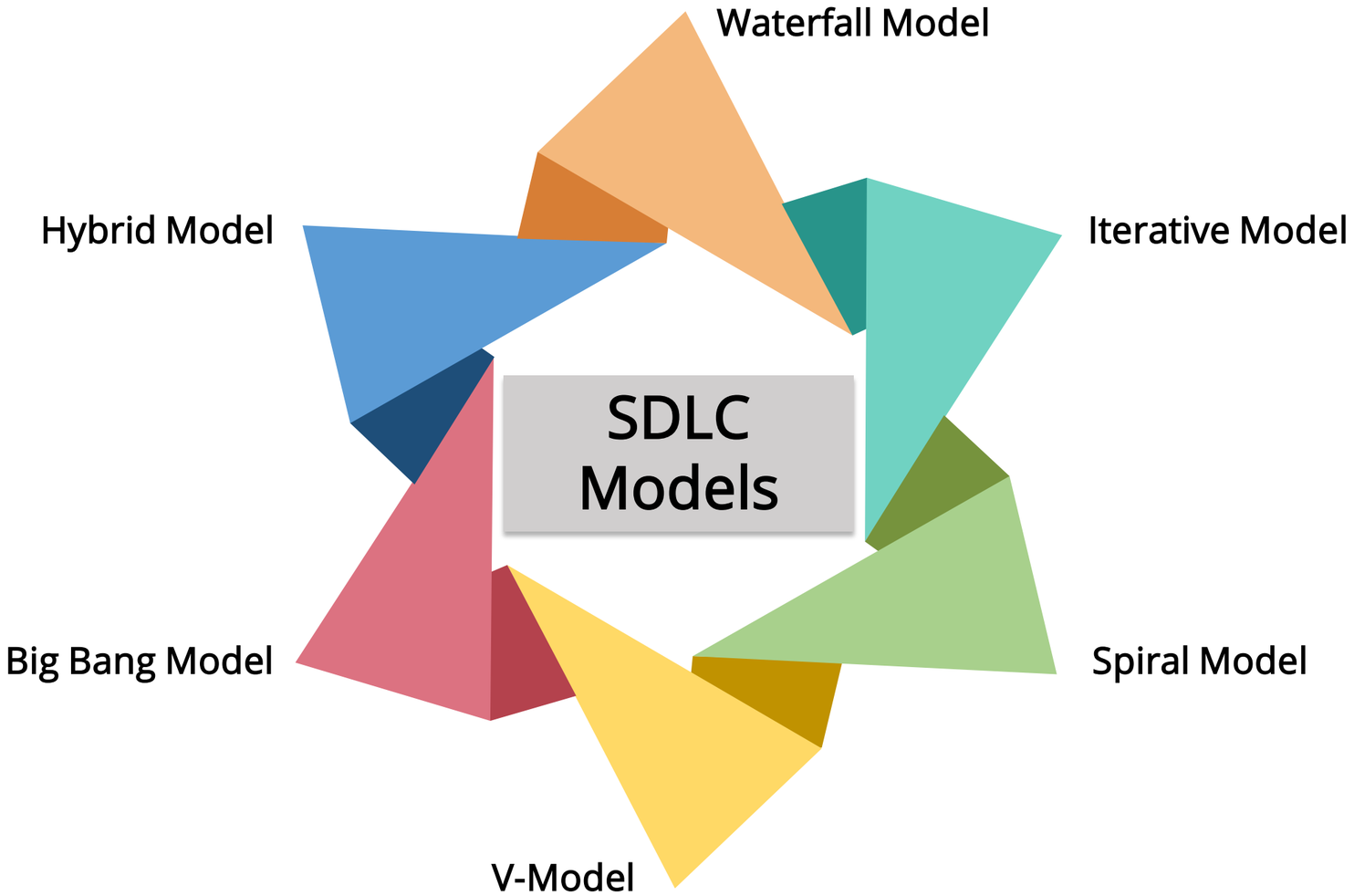
Software Development Life Cycle

Software development life cycle (SDLC) is the methodology of developing a working software and dividing it into different stages of development.



SDLC Models

Below are the 6 major SDLC models that industries have been following:



Components Tested in a Software

Though the test engineer tests the entire software end-to-end, below are some of the major aspects of software that are tested by a test engineer:

2 Accessibility

1 User Interface

3 Design

5 Efficiency

7 User-friendliness

4 Modularity

8 Security

6 Code Optimization

Software Testing Life Cycle

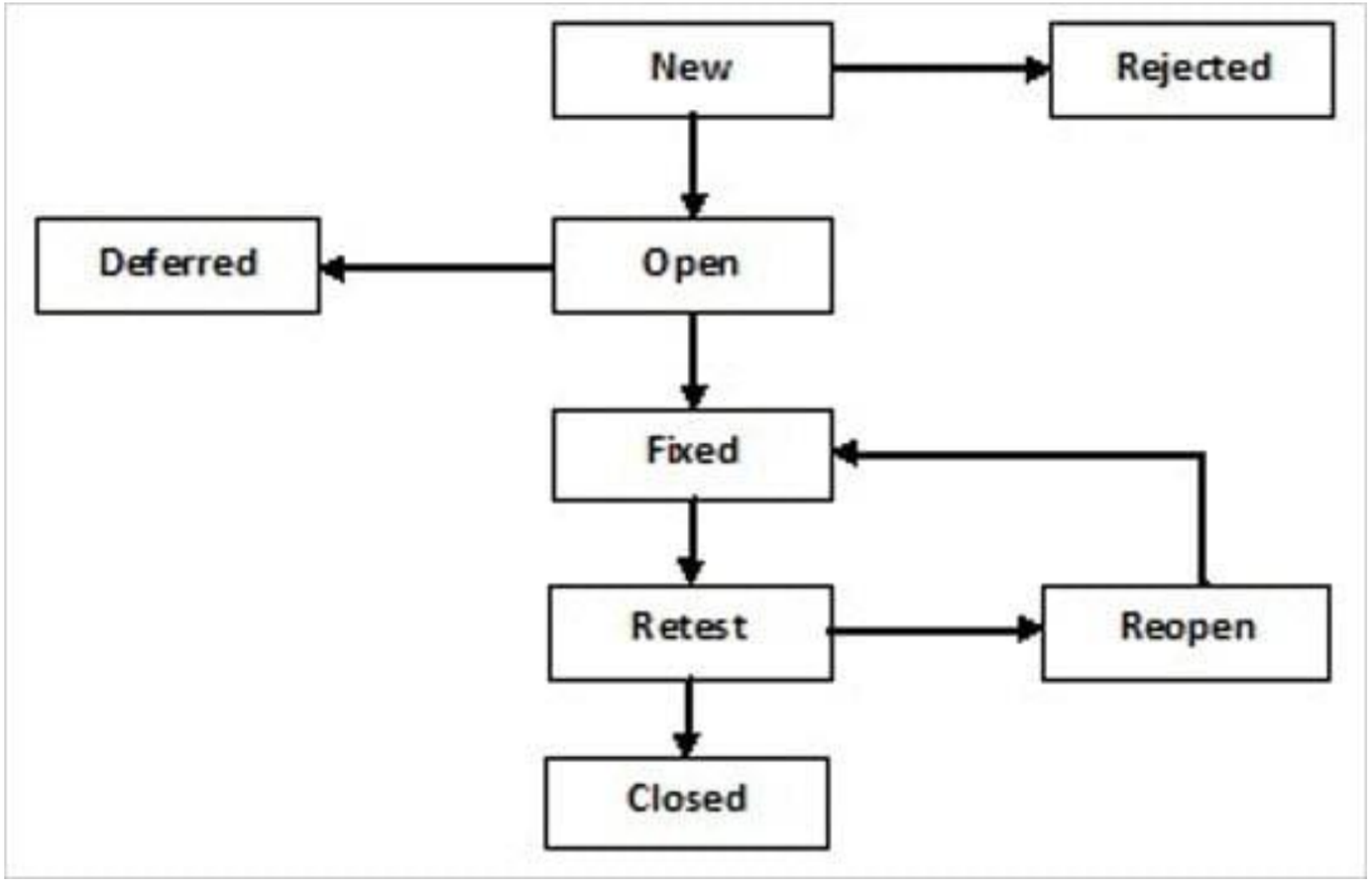
Below are the different stages of software testing:

1. Requirement study
2. Test plan
3. Test case design
4. Traceability matrix
5. Test case execution
6. Bug report
7. Regression test after bug fixes
8. Test reports

Bugs and Errors in Software

Bug Life Cycle

A software bug is an error, flaw, failure, or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. Below is a typical bug life cycle



Bug Life Cycle

Below is the description of all the stages of a bug or defect in software testing:

New: It is the status of the bug when it is first detected by the tester.

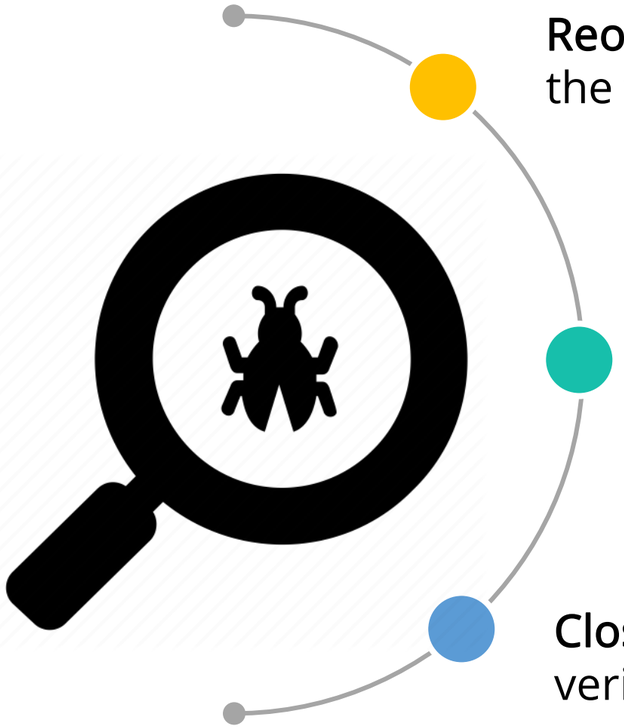
Assigned and Open: It is the status of the bug when the tester assigns the bug to the developer team for fixing it.

Fixed: It is the status of the bug when the developer fixes the bug according to him/her.

Retest and Pending Retest: It is the status of the bug when it is transferred back to the tester to again run tests on it to check if it is completely resolved.

Bug Life Cycle

Below is the description of all the stages of a bug or a defect in software testing:



Reopen: It is the status when the tester detects the same bug after applying the tests again on the previously fixed bug.

Verified: If the tester doesn't detect the same bug again after **Retest**, it is said to be verified.

Closed: The issue is closed once the bug is fixed by the developer team and verified by the tester after regression testing.

Finding a Bug

Take the following steps into consideration while finding a bug or defect in an application:

Understand the application working and give stress to functional testing and prepare test cases for it before starting testing.

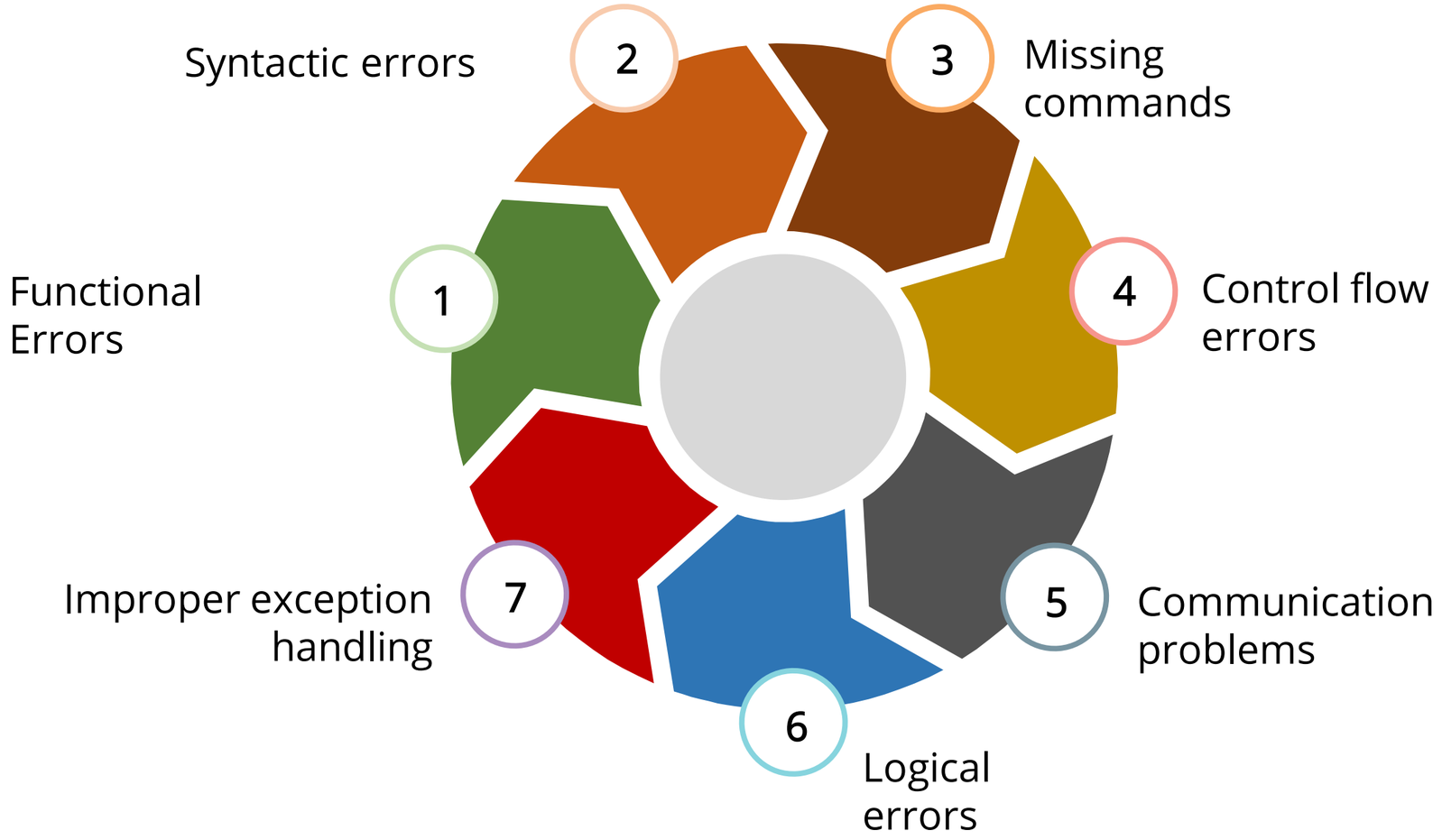
Create a considerable amount of test data, that includes datasets, test conditions, and database records. Also, make sure to the run the tests on different environments.

Fetch the different resulting patterns and compare it with the test results. Also, use your previous test data pattern to analyze the current set of tests.

Make use of the standard test cases which helped you find the bugs or defects in the previously tested application or the previous version of the same application.

Types of Bugs or Errors

Below is the list of the types of error or bugs that a tester may find while testing an application:



Types of Testing

Manual Testing

Definition: Manual Testing is a process of finding out the defects or bugs in a software program. The tester manually executes test cases without using any automation tools.

It is a time consuming process and has the below steps to be followed:

1. Requirement analysis
2. Test plan creation
3. Test cases creation
4. Defect logging
5. Defect fix and re-verification

Automation Testing

Definition: Automation testing is a software testing technique to test and compare the actual outcome with the expected outcome. This can be achieved by writing test scripts or using any automation testing tool.

- Test automation is done to automate the repetitive task and other operations that are difficult to test manually.
- There are many tools for automation testing and some of the popular and widely used are selenium, postman, jmeter, TestNG, and Appium.
- All the tools focus on the different aspects of the software like functional, API, mobile and so on.
- The major difference between manual and automation testing is the introduction of the tools that run the test cases for the tester and generates corresponding reports, which end up in saving time and effort.

Black Box Testing

Definition: Black box testing, also known as behavioral testing, is a software testing method in which the internal structure or design or implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

- It can be for both functional and non-functional aspects of an application.

- The different techniques of black box testing are shown below:

1. **Equivalence partitioning:** In this technique, tester divides the input values into valid and invalid partitions and selects each section as a test data for test cases.

1. **Boundary value analysis:** In this technique, the tester determines the boundary values for input values and selects that for the test cases.

1. **Cause-effect graphing:** In this technique, the tester studies the input conditions and their corresponding output conditions and creates test cases based on it.

White Box Testing

Definition: White box testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure or design or implementation of the item being tested is known to the tester.

- The major difference between black and white box testing is that the testing component is known to the test in white box testing unlike black box testing where it is hidden.

- There are two major techniques used by testers while performing white box testing:

1. **Statement coverage:** In this technique, the tester tests every statement in the code at least once.

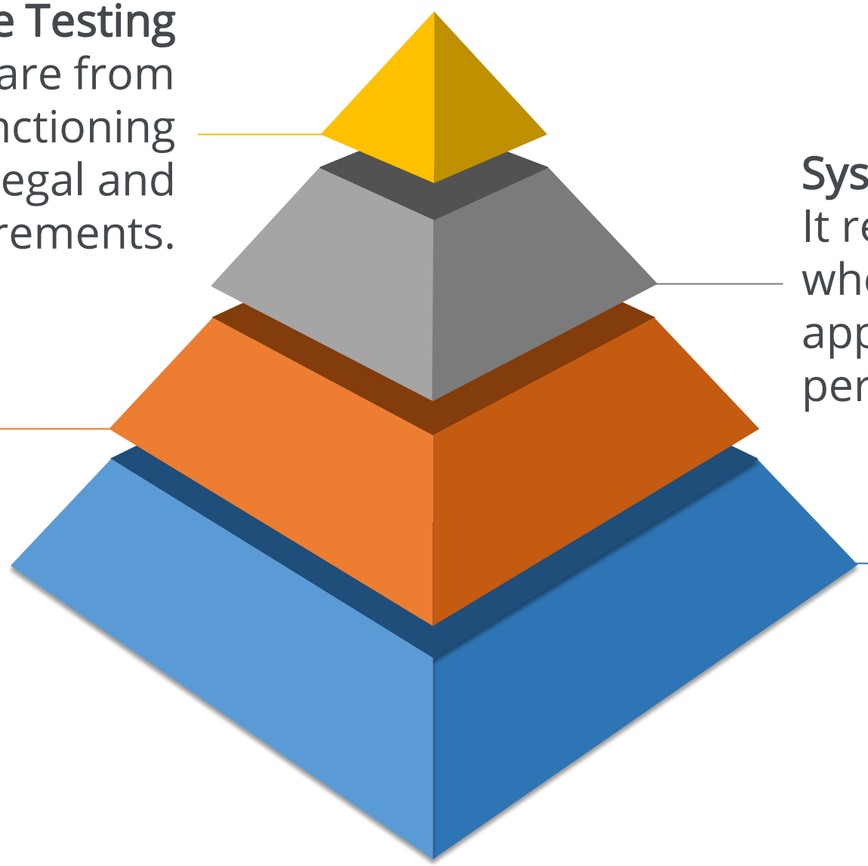
1. **Branch coverage:** In this technique, the tester tests every possible path, every conditional and looping statements, to make sure code is covered in all the possible ways.

Software Testing Levels

Software Testing Levels

Acceptance Testing
It refers to the testing of the software from cosmetic looks to the internal functioning of the application including both legal and contractual requirements.

Integration Testing
It refers to the testing of interaction of one or more unit tests with each other making up a single module of the application.



System Testing
It refers to the testing of the system as a whole where all the components of the application are tested together for performance and also quality assurance.

Unit Testing
It refers to the test of a smallest element of a code that is complete and can be tested.

Test Planning Process

Test Strategy Document

A Test Plan is a document, which describes the scope, objective, method and software testing task. Below are the steps to prepare a comprehensive test strategy document:

Scope: It defines the stakeholders responsible for the review and approval of the document. It also includes the testing activities carried out with timelines.

Test approach: It defines the process of testing, testing levels, roles and responsibilities of team members, tools, and defect management methodologies.

Test environment: It defines the specification of the testing environment. It also describes the test data backup process and restore strategy.

Testing tools: Automation and test tools, which are required along with commercial of them based on the requirements.

Test Strategy Document

Release control: Release management plan contains the version history to make sure that test execution for all modifications is recorded in that release.

Risk analysis: It defines all the risks that a tester can estimate. It also gives a clear plan to mitigate the risks.

Review and approval: All the activities are reviewed and any changes required are made and approval is given to start the testing process.

Test Plan vs. Test Strategy

There is a lot of confusion between test strategy and test plan as many aspects of them are similar. Thus, below are the differences between them:

Test Plan	Test Strategy
A test plan is derived from Software Requirement Specification (SRS), which puts more stress in providing details of scope of testing and the different activities performed in testing.	A test strategy is a very high-level document, which tells about the testing process as a whole.
It is specific to a particular project.	It is normally for a complete organization. Although, it can also be for a particular project.
Details of testing activities, like the techniques used, schedule, and resources, are mentioned.	It describes the high-level test design techniques to be used and environment specifications.
It is prepared by the test lead or the test manager.	It is generally prepared by the project manager.

Writing Test Cases

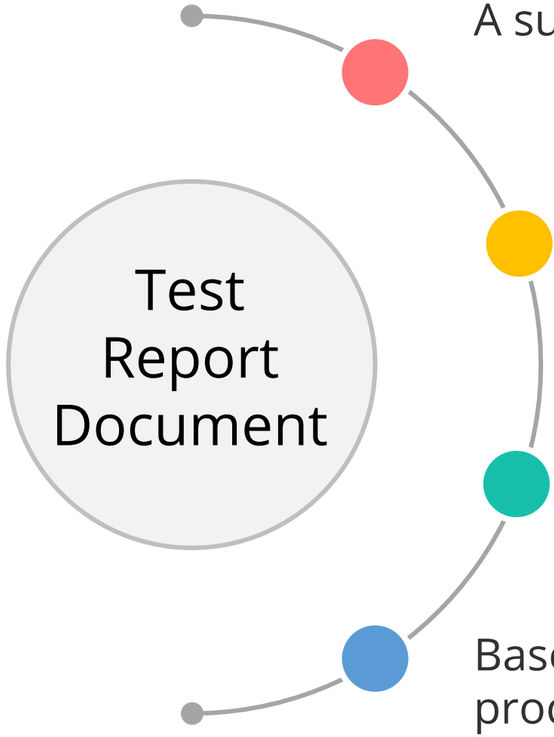
A test case is a set of tasks that will help in completing a test scenario. Below is an example to understand the same:

Test Scenario	Test Case Description	Test Steps	Test Data
Check if the login functionality works.	Check response when valid email and password is entered.	1) Enter the Email Address 2) Enter the Password 3) Click on Sign in	Email: zxcv@qwerty.com Password: simpli@111

Test case, as shown above, includes the steps to perform in order to accomplish the test scenario results and find the defect, if any.

Creating a Test Report

Below are the characteristics that define a test report:



A summary of test activities and final test results.

An assessment of how well the test was executed and what were the impediments.

Evaluation of the quality of the product based on the test results.

Based on the test report, stakeholders should be able to decide the release of the product and the further preventive measures to be implemented.

Creating a Test Report

Below is an example of a test report:

Test Report

Test Cycle **System Test**

EXECUTED	PASSED			130	
	FAILED			0	
	(Total) TESTS EXECUTED (PASSED + FAILED)				**130**
PENDING					0
IN PROGRESS					0
BLOCKED					0
(Sub-Total) TEST PLANNED (PENDING + IN PROGRESS + BLOCKED + TEST EXECUTED)					**130**

Functions	Description	% TCs Executed	% TCs Passed	TCs pending	Priority	Remarks
New Customer	Check new Customer is created	100%	100%	0	*High*	
Edit Customer	Check Customer can be edited	100%	100%	0	*High*	
New Account	Check New account is added	100%	100%	0	*High*	
Edit Account	Check Account is edit	100%	100%	0	*High*	
Delete Account	Verify Account is delete	100%	100%	0	*High*	
Delete customer	Verify Customer is Deleted	100%	100%	0	*High*	
Mini Statement	Verify Ministatement is generated	100%	100%	0	*High*	
Customized Statement	Check Customized Statement is generated	100%	100%	0	*High*	

Selecting the Right Automation Tool

One should keep the below practices in mind while selecting the appropriate automation tool for an application:

Understand the targeted audience

Understand the project requirement thoroughly

List the key criteria suitable for the application

Keep present automation tool as benchmark

Key Takeaways

- Software testing is a subjective task as it requires one’s perspective for an application.
- Test strategy is a document that covers all the aspects of the testing process.
- A good test report helps the stakeholders to make release management decisions.