Automation Testing

Overview of Automation Testing

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

A Day in the Life of an Automation Test Engineer

Anna has decided to choose a Test Driven Development for her projects.

To complete her project, she has to learn TDD which comprises a series of frameworks and focuses on delivering quality and value to select the correct tool to complete functional testing for her project.

To achieve the above, she will learn a few concepts in this lesson that can help her to find a solution for the scenario.

Learning Objectives

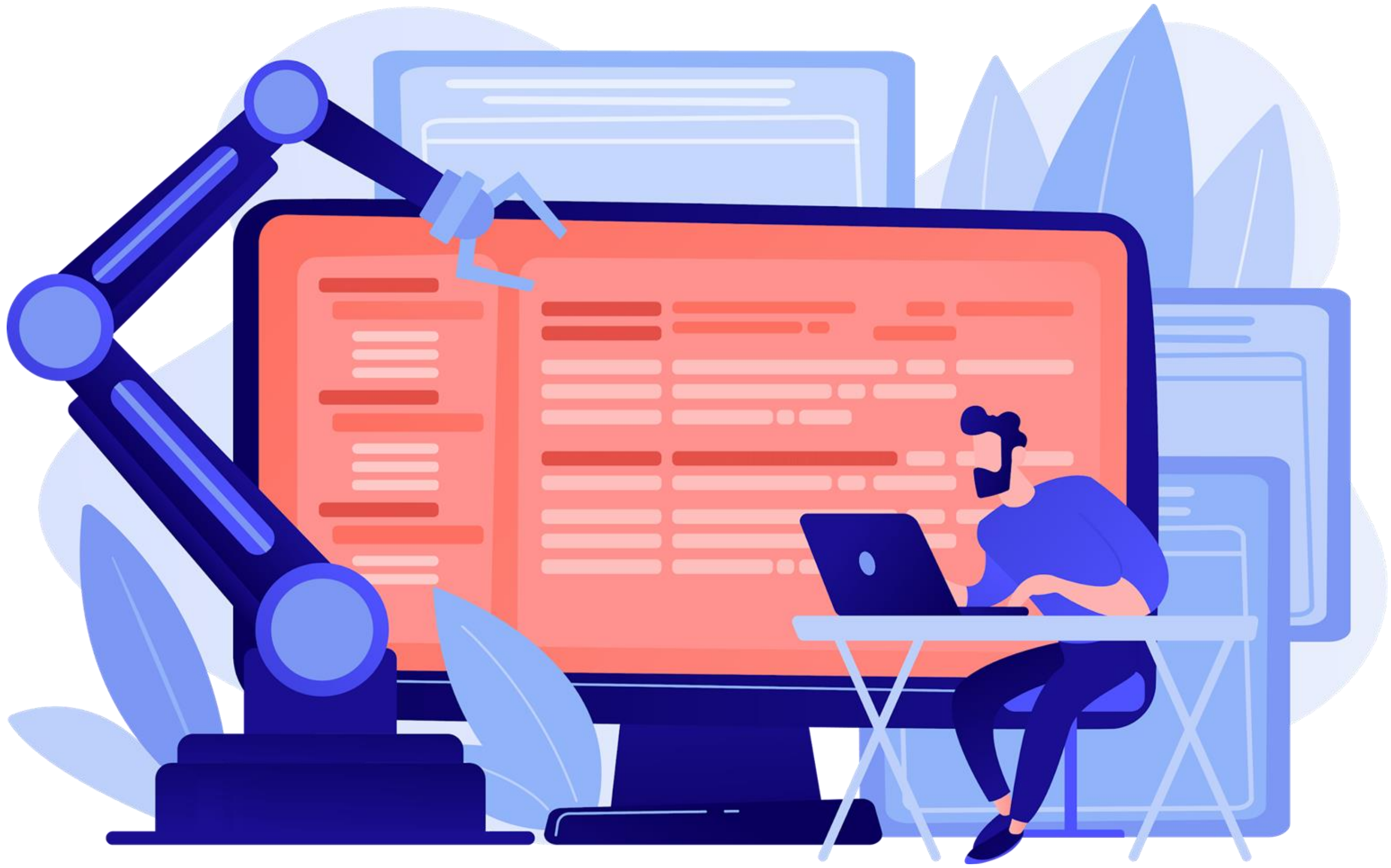
By the end of this lesson, you will be able to:

- Illustrate automation testing
- Understand the basics of TDD
- Configure the frameworks of TDD
- Understand the red-green refactor in depth

What Is Automation Testing?

Introduction to Automation Testing

Automation testing is a software testing technique in which a test case suite is executed using particular automated testing software tools.



Why Automation Testing?

Why Automation Testing?

Automation testing is used whenever there are multiple releases or regression cycles on the application or software.

The Significance of Automation Testing

The following are some of the reasons why automation testing is important:

- Cost Saving
- Increased Productivity
- Reusability
- Foundation for CI and DevOps
- Bugs
- Load Testing

Plan, Design, and Develop

Test Planning

A test plan is a detailed document that outlines the testing strategy, objectives, timetable, estimation, deliverables, and resources needed to test a product.

The test plan acts as a blueprint for carrying out software testing tasks in a systematic manner.

Importance of Test Plan

It helps others outside the test team understand the details of tests, such as developers, business managers, and customers.

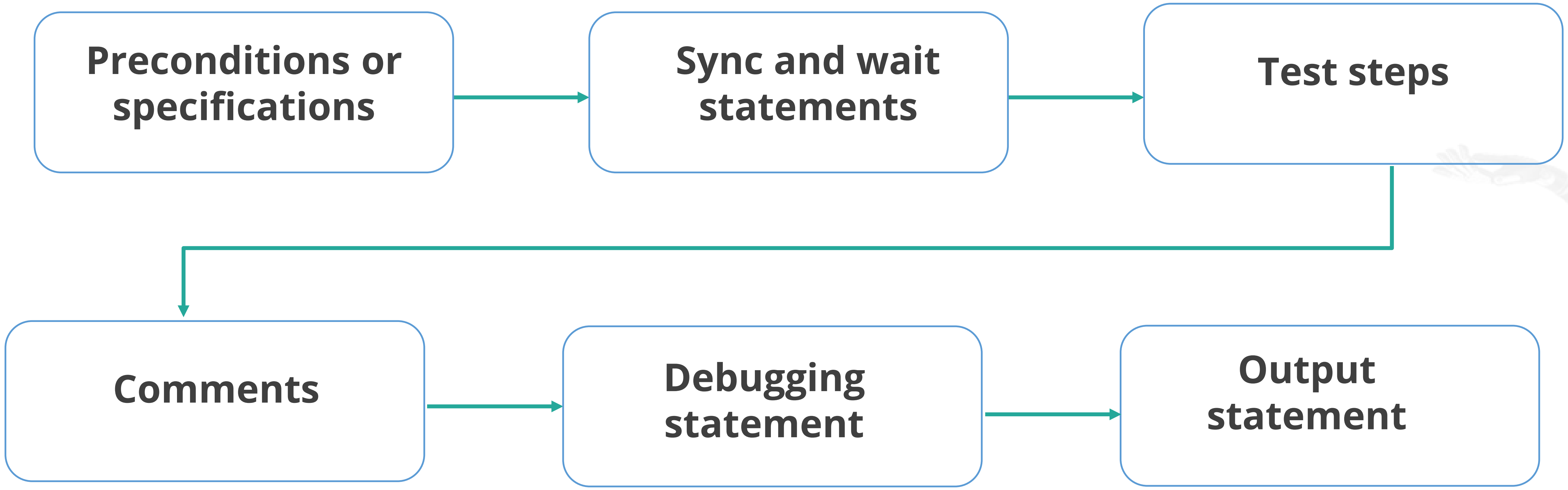
It gives ideas to be guided by the test plan. It's like a set of guidelines that must be followed.

Test Cases

Automation test cases concentrate on application state transitions and data changes, making it even more important that each test case contains only one test.

The Flow of Test Cases

Test case templates for automation differ depending on the automation tools used, however, they should all have the following elements:



Preconditions or specifications

Sync and wait statements

Test steps

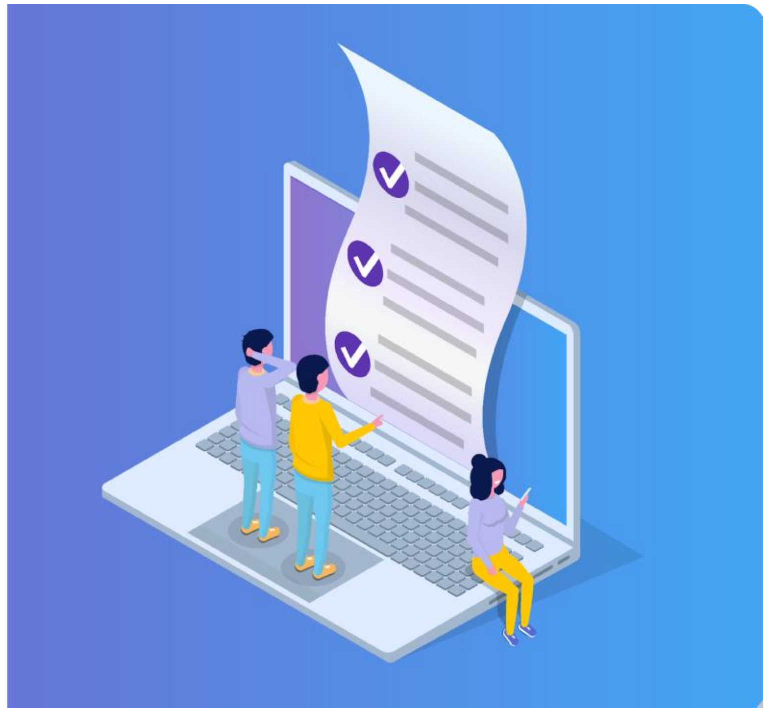
Comments

Debugging statement

Output statement

Test Design

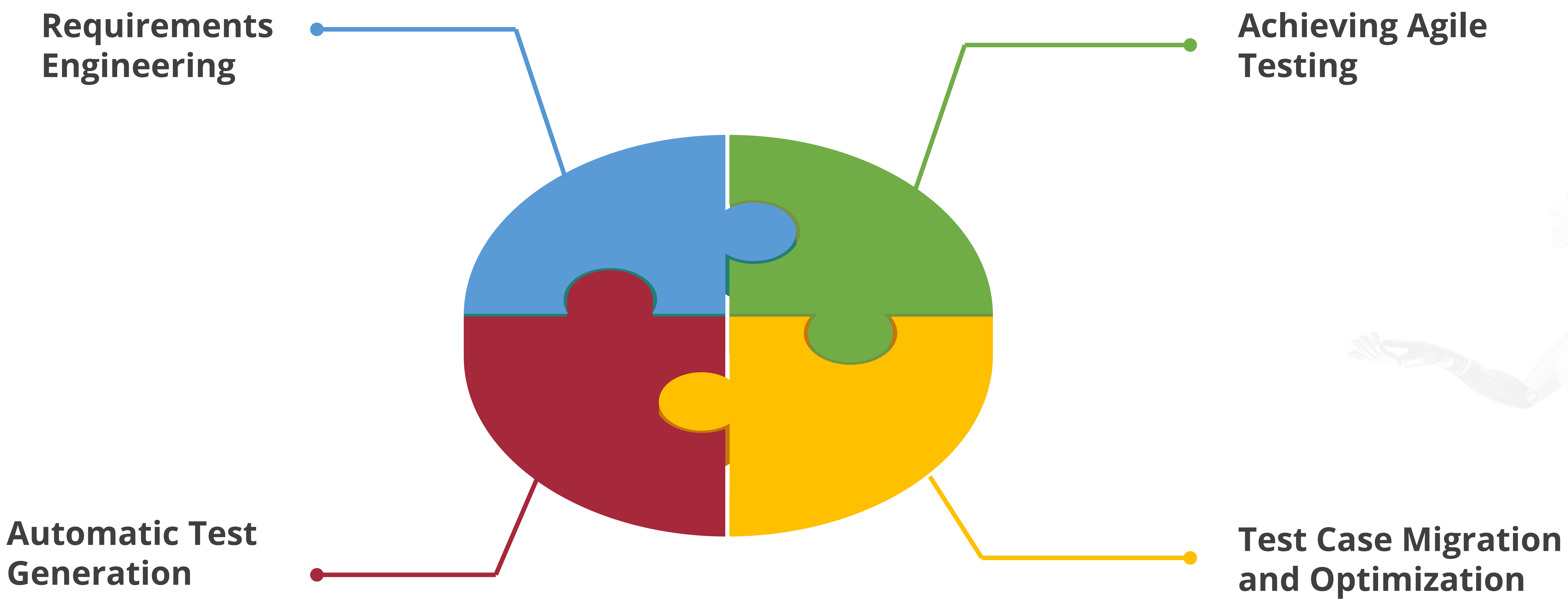
The test design component covers the need to define the number of tests to be done as well as the methods to be used in testing.



The automation tool can be used directly or through the test management tool, which will invoke the automation tool.

Use Cases For the Test Design

Following are the test cases:



Requirements Engineering

Achieving Agile Testing

Automatic Test Generation

Test Case Migration and Optimization

Requirements Engineering

The concept of modeling lies in the foundation of requirements engineering, where the initial demand written on a whiteboard is not translated into typical text-based requirements.

The diagrams on the whiteboard convey needs as mathematically accurate **visual flows**, improving requirements engineering **accuracy** and eliminating **ambiguities** while also facilitating greater cooperation and communication among key stakeholders.

Automatic Test Generation

Having the capacity to automatically construct test cases linked to the proper data and expected results right from your requirements models is one method to increase testing speed and velocity.

By automating the automation, you can generate every automated test required to thoroughly test an application.

Test Case Migration and Optimization

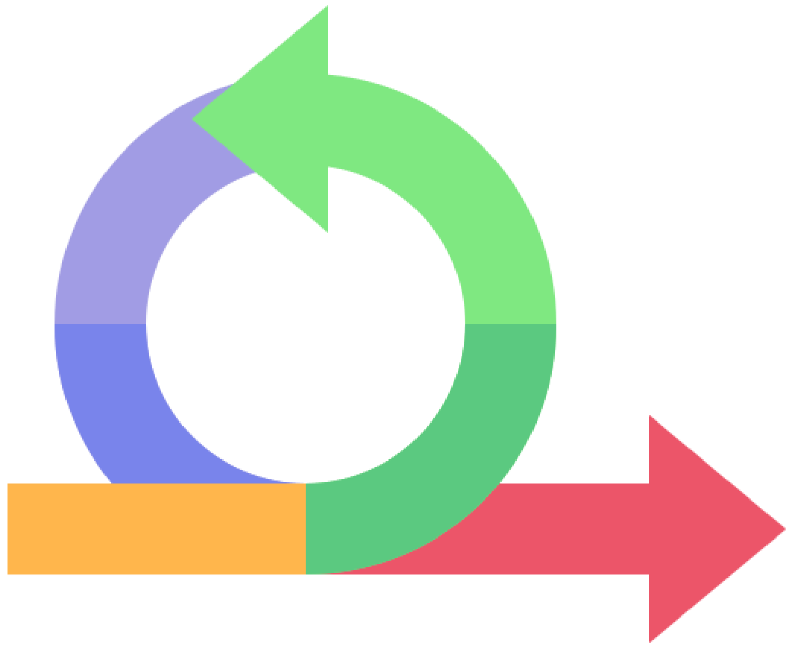
Test case migration and optimization enable you to create the least number of test cases while providing maximum coverage.

ATLASSIAN

Using this connection, test teams can quickly import test cases, remove duplication, and export improved test cases.

Achieving Agile Testing

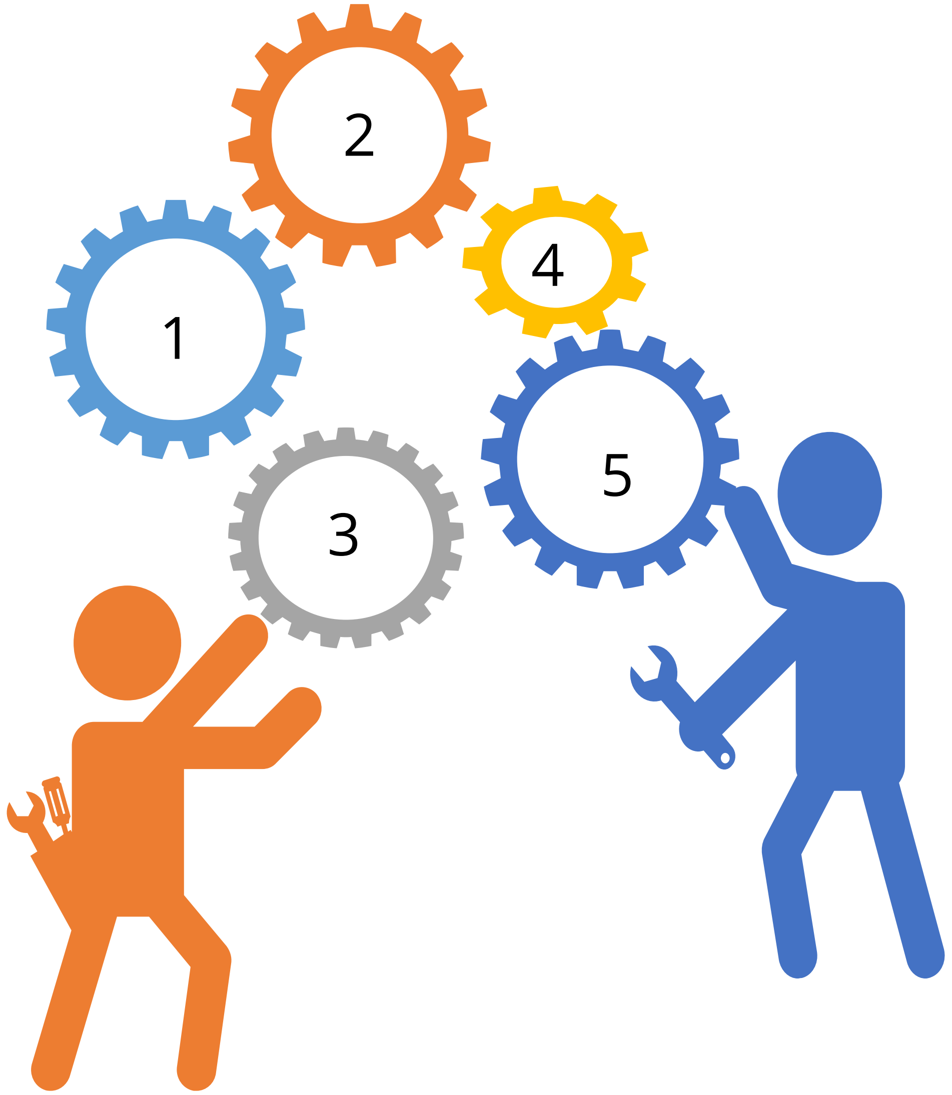
Agile testing is the process of baking quality into a product as it is specified, designed, and delivered.



Test Tool Selection

Test Tool Selection

While choosing the ideal tool, keep the following aspects in mind:



Is it simple to create and maintain scripts for the tool?

Is it compatible with various platforms such as the web, mobile, and desktop?

Is there a test reporting feature in the tool?

How many different sorts of testing can this tool handle?

How many languages is the tool capable of supporting?

Automation Test Framework

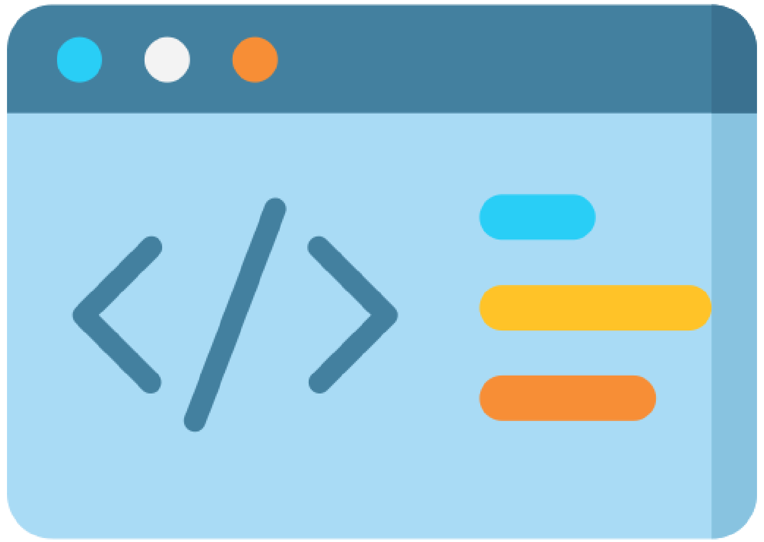
Linear Automation Test Framework

Introduction level testing is included in a linear automation test framework.

Testers write test scripts in order and run them one by one.

Modular-based Test Framework

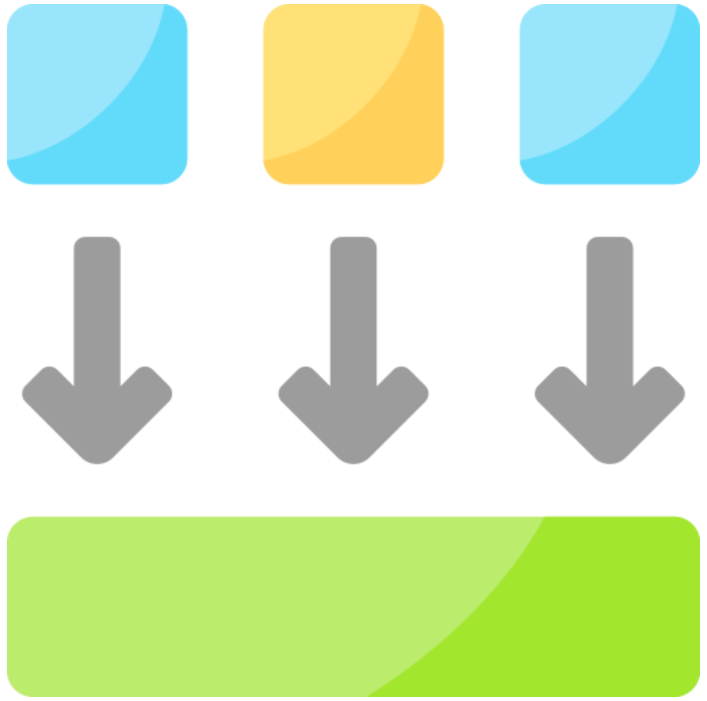
Modular test frameworks split down test cases into small modules.



It takes a method that is both incremental and non-incremental.

Library Architecture Test Framework

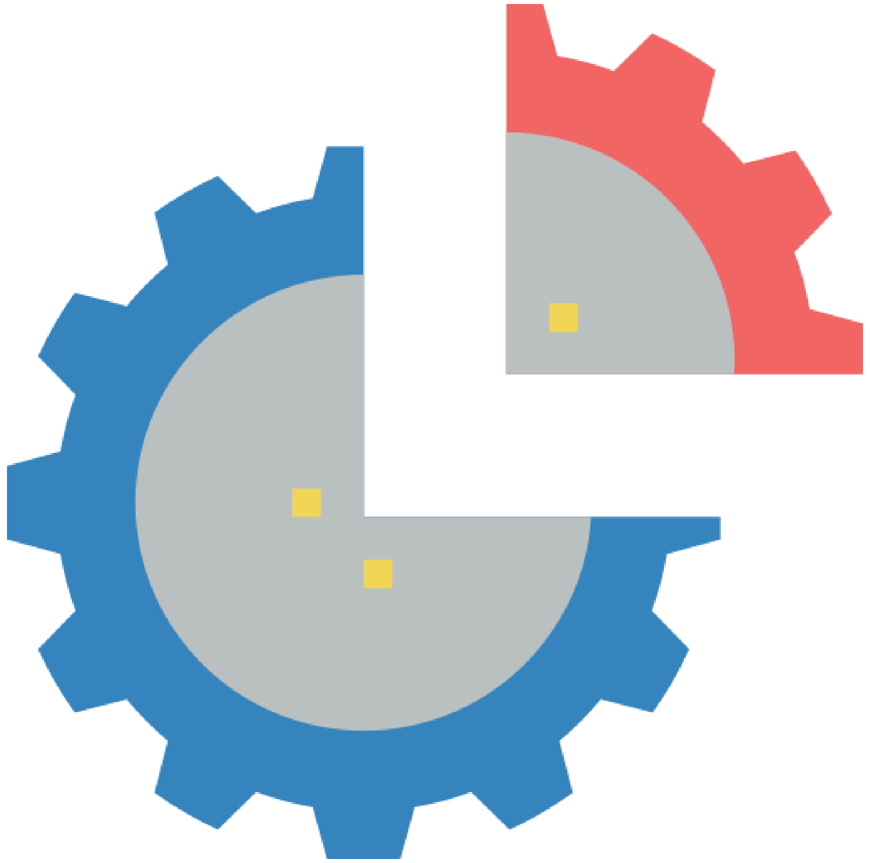
This framework recognizes tasks that share similarities in the test script.



Testers perform the function-based grouping of similar tasks.

Data-driven Test Framework

A data-driven test framework separates test logic from test data.



Hybrid Test Framework

A hybrid test framework gives users more flexibility by mixing elements from other frameworks.

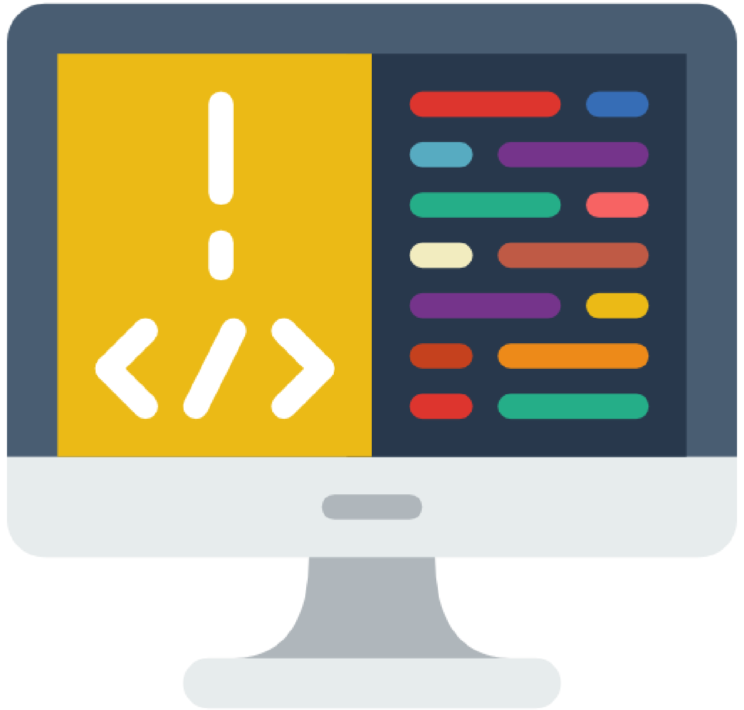
FULL STACK

Types of Automated Testing

simplilearn

Automation Testing Approaches

For automation testing, there are three basic ways to consider:

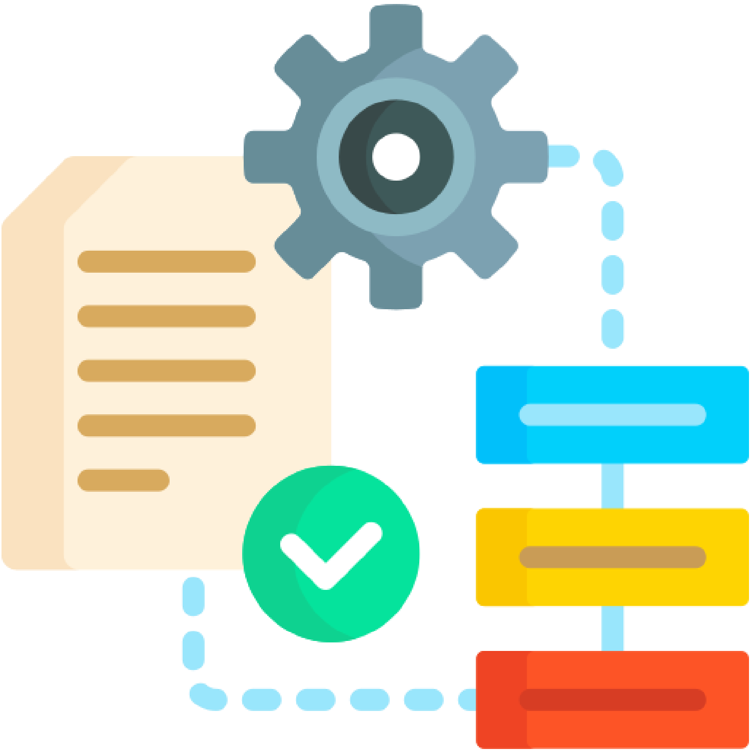


Code-driven:
This method focuses mostly on test case execution to determine whether or not particular code portions are behaving as expected.

Automation Testing Approaches

Graphical user interface (GUI) testing: Users' behaviors can be recorded and analyzed several times by testers.

Automation Testing Approaches



Test automation framework:

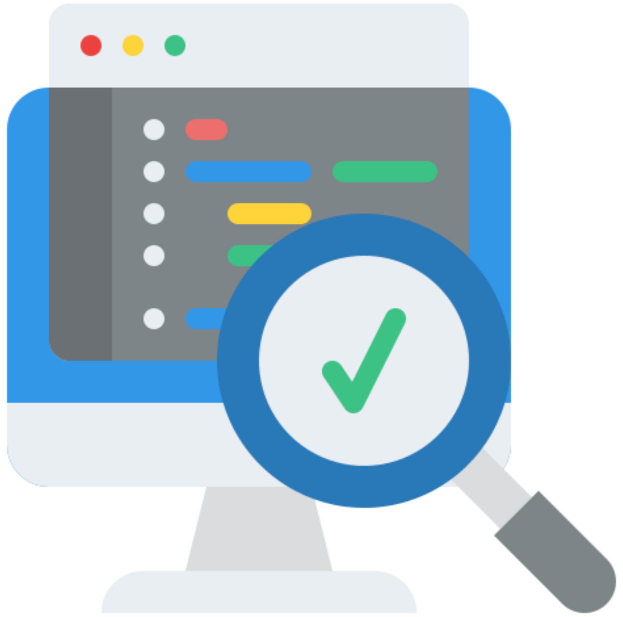
A framework is a set of rules for creating useful results through automated testing.

FULL STACK

Difference Between Manual and Automation Testing

How Does Test Planning Differ for Manual and Automation Projects?

Manual testing is software testing in which tests are run manually by a QA analyst.



It's used to find bugs in the software that is still in the development.

The Difference between Manual and Automated Testing

Automation Testing	Manual Testing
Automated testing is much faster than manual testing.	Manual testing involves more time and resources.
Random testing is not possible with automation.	Manual testing allows for exploratory testing.
Automated testing is a dependable method. It is performed by tools and scripts.	Manual testing is less accurate due to the possibility of human error.
For low-volume regression, it is not cost-effective.	For high-volume regression, it is not cost-effective.

Key Takeaways

- Automation testing is the process of repeating pre-defined actions using tools, scripts, and software to perform test cases.
- Automation scripts can be executed directly by the automation tool or through the test management tool, which will invoke the automation tool.
- For automation testing, three approaches are considered: code-driven, GUI, and automation framework.
- Some of the major components of test automation frameworks are test data management and testing libraries.