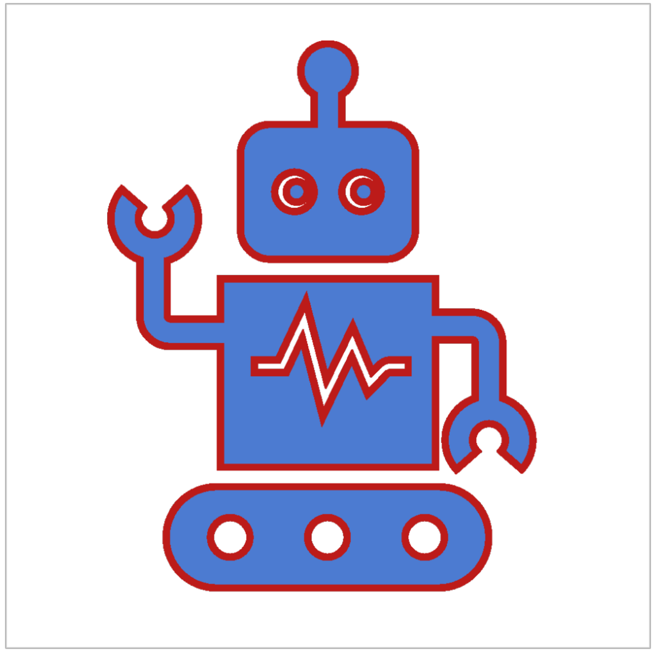


AUTOMATING DEVOPS & QA

COMP 3104 - DEVOPS

SOFTWARE TESTING

- **Automation Testing or Test Automation** is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.
- The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources.
- **Manual Testing** is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application. Manual software testing is the most primitive technique of all testing types and it helps to find critical bugs in the software application.

WHICH TEST CASES TO AUTOMATE?

Test cases to be automated can be selected using the following criterion to increase the automation ROI

- High Risk - Business Critical test cases
  - Test cases that are repeatedly executed
  - Test Cases that are very tedious or difficult to perform manually
  - Test Cases which are time-consuming
- The following category of test cases are not suitable for automation:
  - Test Cases that are newly designed and not executed manually at least once
  - Test Cases for which the requirements are frequently changing
  - Test cases which are executed on an ad-hoc basis.

AUTOMATED TESTING PROCESS

Following steps are followed in an Automation Process

Step 1) Test Tool Selection

Step 2) Define scope of Automation

Step 3) Planning, Design and Development

Step 4) Test Execution

Step 5) Maintenance

QUALITY ASSURANCE IN SOFTWARE TESTING

Quality Assurance in Software Testing is defined as

- ✓ a procedure to ensure the quality of software products or services provided to the customers by an organization.
- ✓ Quality assurance focuses on improving the software development process and making it efficient and effective as per the quality standards defined for software products. Quality Assurance is popularly known as QA Testing.

What is Quality?

- ✓ Quality is extremely hard to define, and it is simply stated: "Fit for use or purpose." It is all about meeting the needs and expectations of customers with respect to functionality, design, reliability, durability, & price of the product.

What is Assurance?

- ✓ Assurance is nothing but a positive declaration on a product or service, which gives confidence.
- ✓ It is certainty of a product or a service, which it will work well.
- ✓ It provides a guarantee that the product will work without any problems as per the expectations or requirements.

QUALITY ASSURANCE FUNCTIONS:

There are 5 primary Quality Assurance Functions:

1. **Technology transfer:** This function involves getting a product design document as well as trial and error data and its evaluation. The documents are distributed, checked and approved
2. **Validation:** Here validation master plan for the entire system is prepared. Approval of test criteria for validating product and process is set. Resource planning for execution of a validation plan is done.
3. **Documentation:** This function controls the distribution and archiving of documents. Any change in a document is made by adopting the proper change control procedure. Approval of all types of documents.
4. **Assuring Quality of products**
5. **Quality improvement plans**

DEFINE THE SCOPE OF AUTOMATION

The scope of automation is the area of your Application Under Test which will be automated. Following points help determine scope:

- The features that are important for the business
- Scenarios which have **a large amount of data**
- **Common functionalities** across applications
- Technical feasibility
- The extent to which business components are reused
- **The complexity** of test cases
- Ability to use the same test cases for cross-browser testing

FRAMEWORK FOR AUTOMATION

- A framework is set of automation guidelines which help in
- Maintaining consistency of Testing
- Improves test structuring
- Minimum usage of code
- Less Maintenance of code
- Improve re-usability
- Non Technical testers can be involved in code
- The training period of using the tool can be reduced
- Involves Data wherever appropriate

TYPES OF AUTOMATED TESTING

- Smoke Testing
- Unit Testing
- Integration Testing
- Functional Testing
- Keyword Testing
- Regression Testing
- Data Driven Testing
- Black Box Testing

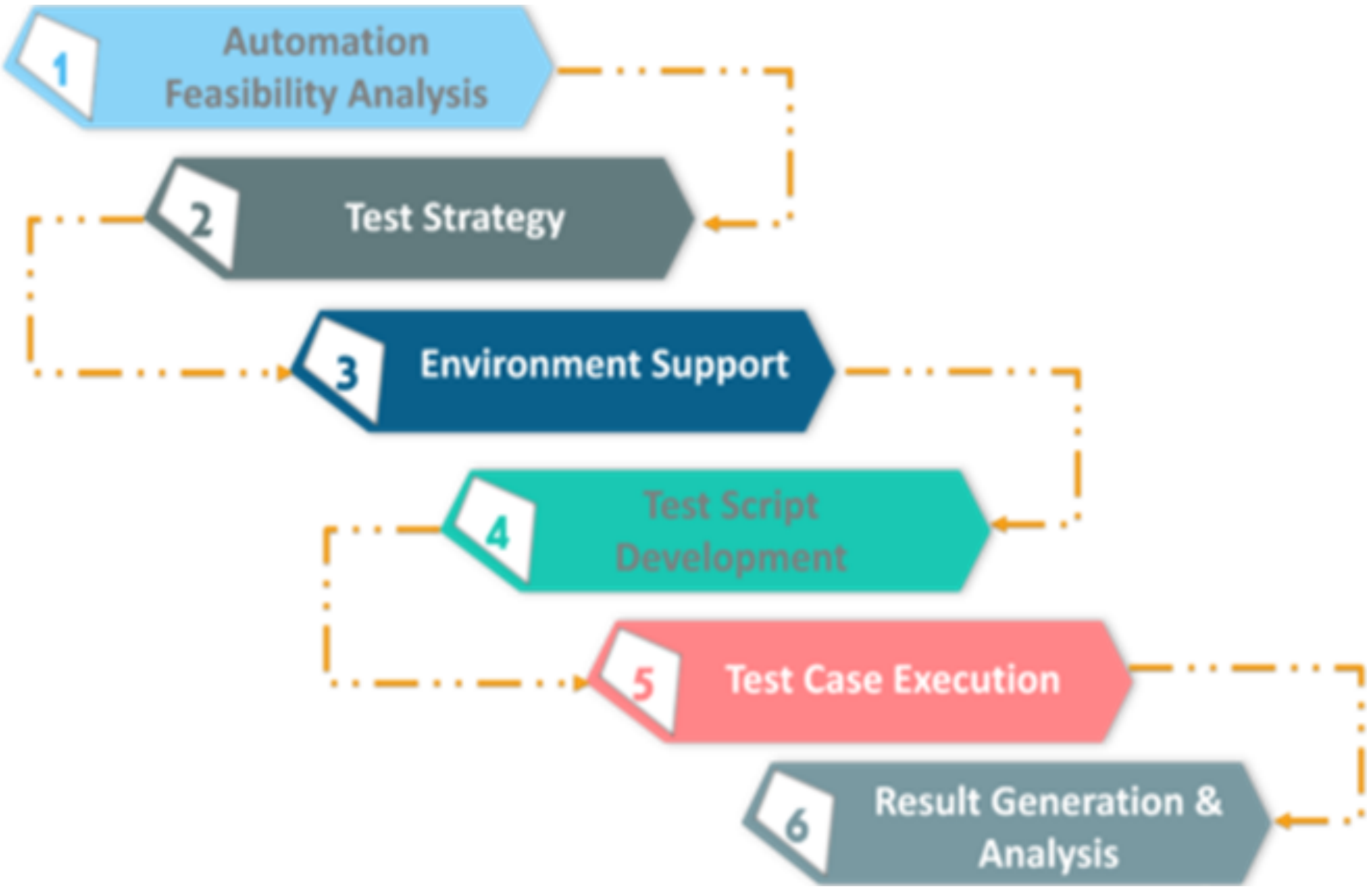
PROS OF AUTOMATED TESTING:

- Automated testing helps you to find more bugs compare to a human tester
- As most of the part of the testing process is automated, you can have a speedy and efficient process
- Automation process can be recorded. This allows you to reuse and execute the same kind of testing operations
- Automated testing is conducted using software tools, so it works without tiring and fatigue unlike humans in manual testing
- It can easily increase productivity because it provides fast & accurate testing result
- Automated testing support various applications
- Testing coverage can be increased because of automation testing tool never forget to check even the smallest unit

CONS OF AUTOMATED TESTING:

- Without human element, it's difficult to get insight into visual aspects of your UI like colors, font, sizes, contrast or button sizes.
- The tools to run automation testing can be expensive, which may increase the cost of the testing project.
- Automation testing tool is not yet foolproof. Every automation tool has their limitations which reduces the scope of automation.
- Debugging the test script is another major issue in the automated testing. Test maintenance is costly.

TEST AUTOMATION STRATEGY



TEST AUTOMATION STRATEGY

Automation Feasibility Analysis

- In this step, you check the feasibility of automation. This includes shortlisting the relevant test cases for automation and selecting the right test tool that fits your requirement.

Test Strategy

- Here, you select the test automation framework. You have multiple options to choose from here. For example, linear test automation framework, data-driven framework, key-word driven framework etc.

Environment Set Up

- In this phase, you set up the testing environment and acquire the required hardware and software to execute the automated test cases.

Test Script Development

- In this step, you start creating the automation test scripts. Make sure that the scripts that you have written are reusable, well-structured and well documented.

Test Case Execution

- In this step, you execute the test cases that you have written previously.

Test Result Generation and Analysis

- In this last phase of test automation life cycle, you analyse the output of test cases and share the reports with stakeholders.

PERFORMANCE TESTING

- **Performance Testing** is a software testing process used for testing the speed, response time, stability, reliability, scalability and resource usage of a software application under particular workload.
- The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application.
- It is a subset of performance engineering and also known as “Perf Testing”.
- The focus of Performance Testing is checking a software program's
  - Speed - Determines whether the application responds quickly
  - Scalability - Determines maximum user load the software application can handle.
  - Stability - Determines if the application is stable under varying loads

TYPES OF PERFORMANCE TESTING

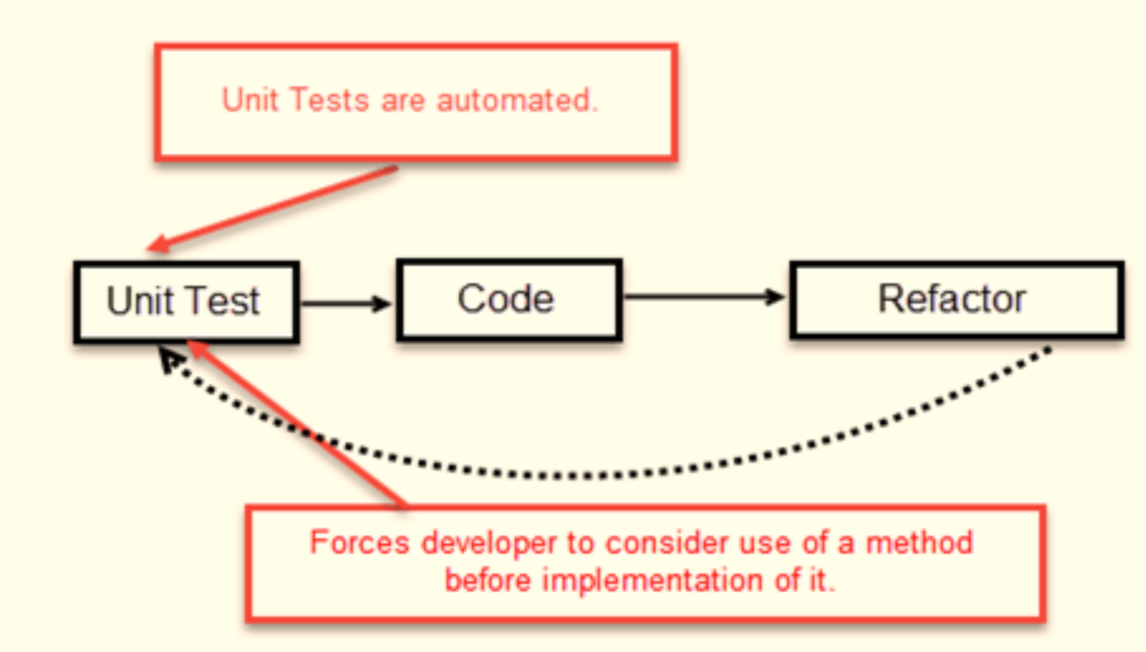
- **Load testing -** checks the application's ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.
- **Stress testing -** involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.
- **Endurance testing -** is done to make sure the software can handle the expected load over a long period of time.
- **Spike testing -** tests the software's reaction to sudden large spikes in the load generated by users.
- **Volume testing** - Under Volume Testing large no. of. Data is populated in a database and the overall software system's behavior is monitored. The objective is to check software application's performance under varying database volumes.
- **Scalability testing** - The objective of scalability testing is to determine the software application's effectiveness in "scaling up" to support an increase in user load. It helps plan capacity addition to your software system.

PERFORMANCE TEST TOOLS

- LoadNinja – This cloud-based load testing tool empowers teams to record & instantly playback comprehensive load tests, without complex dynamic correlation & run these load tests in real browsers at scale.
- NeoLoad - is the performance testing platform designed for DevOps that seamlessly integrates into your existing Continuous Delivery pipeline.
- HP LoadRunner - This tool is capable of simulating hundreds of thousands of users, putting applications under real-life loads to determine their behaviour under expected loads.
- Loadrunner features a virtual user generator which simulates the actions of live human users.
- Jmeter - one of the leading tools used for load testing of web and application servers.

TEST DRIVEN DEVELOPMENT

- **Test Driven Development (TDD)** is software development approach in which test cases are developed to specify and validate what the code will do.
- In simple terms, test cases for each functionality are created and tested first and if the test fails then the new code is written in order to pass the test and making code simple and bug-free.
- Test-Driven Development starts with designing and developing tests for every small functionality of an application.
- TDD instructs developers to write new code only if an automated test has failed.
- This avoids duplication of code.



Unit Tests are automated.
Unit Test
Code
Refactor
Forces developer to consider use of a method before implementation of it.

ADVANTAGES OF TDD

Early bug notification

- Using TDD you build up, over time, a suite of automated tests that you and any other developer can rerun at will.

Better Designed, cleaner and more extensible code

- It helps to understand how the code will be used and how it interacts with other modules.
- It results in better design decision and more maintainable code.
- TDD allows writing smaller code having single responsibility rather than monolithic procedures with multiple responsibilities. This makes the code simpler to understand.
- TDD also forces to write only production code to pass tests based on user requirements.

Confidence to Refactor

- If you refactor code, there can be possibilities of breaks in the code. So having a set of automated tests you can fix those breaks before release. Proper warning will be given if breaks found when automated tests are used.
- Using TDD, should results in faster, more extensible code with fewer bugs that can be updated with minimal risks.

WHAT IS BDD TESTING?

- **BDD (Behavior-driven development) Testing** is a
  - technique of agile software development and is as an extension of TDD i.e., Test Driven Development.
  - In BDD, test cases are written in a natural language that even non-programmers can read.
- BDD provides a path that acts as a bridge to overcome the gap between the technical and the non-technical teams because the test cases are commonly written in simple text, i.e. English.
- The main advantage of BDD is the low jargon and clearer approach which is easier to understand.

PROS OF BEHAVIOUR DRIVEN DEVELOPMENT

- **Test Reuse**
  - With the right approach and tools, it's easy to reuse the test steps in different automation scenarios.
- **Data-driven Testing**
  - At its heart, BDD has parameterisation and data tables built in. You can leverage this to increase test coverage by driving each scenario with various data sets.
- **Targeted Tests**
  - If tags are used effectively in the scenarios and specifications, then there is another benefit to be had. This gives you the capability to target test runs on specific functional areas.

CONS OF BDD

- **Time Overhead**
  - Creating and maintaining the feature files and scenarios requires an overhead investment of time and effort.
- **Retrofitting can be time-consuming**
  - Retrospectively fitting BDD to an existing project can be a time-consuming and difficult process.
- **Structuring files**
  - Structuring all your feature files, scenarios and executable specifications requires some careful planning.
- **Communication**
  - There needs to be a good amount of communication between the person writing the feature files and the person developing the automation code.

SONARQUBE

SonarQube is the leading tool for continuously inspecting the Code Quality and Security of your codebases and guiding development teams during Code Reviews.

- SonarQube is a web-based open source platform used to measure and analyse the source code quality.
- Code quality analysis makes your code more reliable and more readable.
- SonarQube is written in java but it can analyze and manage code of more than 20 programming languages, including C/C++, PL/SQL, Cobol etc through plugins.
- Plugins extend the functionality of SonarQube. More than 50 plugins are available.
- SonarQube is maintained by SonarSource. Download

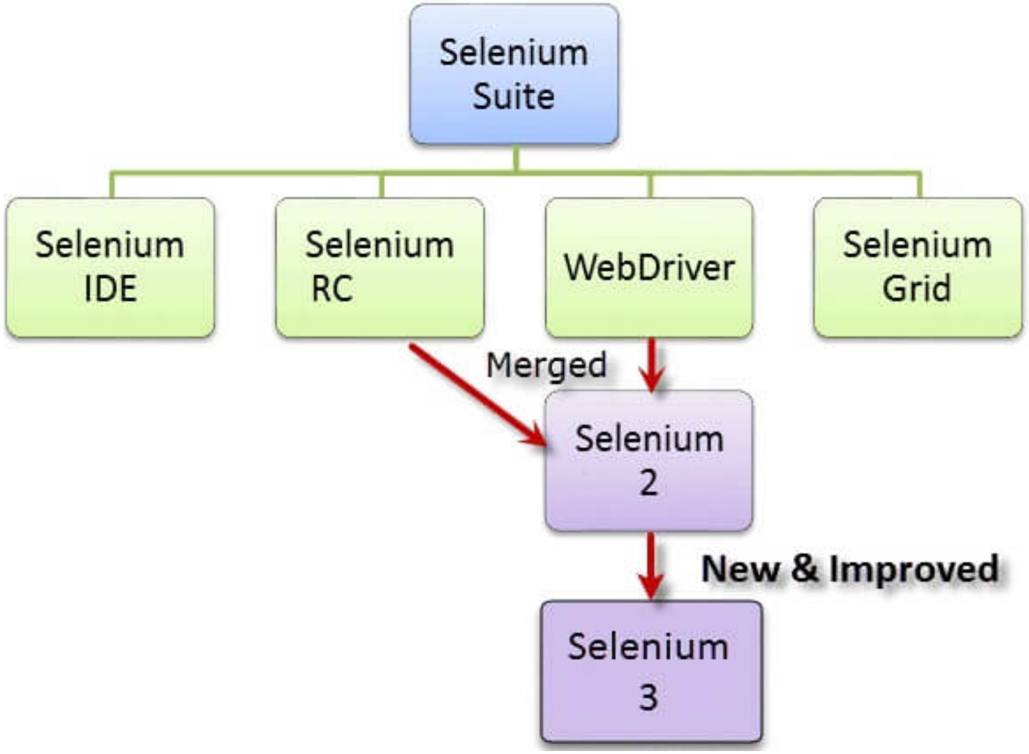
SONAR COVERS THE 7 SECTIONS OF CODE QUALITY

1. Architecture and Design
2. Unit tests
3. Duplicated code
4. Potential bugs
5. Complex code
6. Coding standards
7. Comments

- SonarQube receives files as an input and analyzes them along with barriers.
- Then calculates a set of metrics, stores them in a database and shows them on a dashboard.
- This recursive implementation helps in analysis of code quality and how code improves over time.

WHAT IS SELENIUM ?

- **SELENIUM** is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms.
- You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts.
- Testing done using the Selenium tool is usually referred to as Selenium Testing.
- Selenium Software is not just a single tool but a suite of software, each piece catering to different testing needs of an organization. Here is the list of tools
  1. Selenium Integrated Development Environment (IDE)
  2. Selenium Remote Control (RC)
  3. WebDriver
  4. Selenium Grid



WHAT IS APPIUM ?

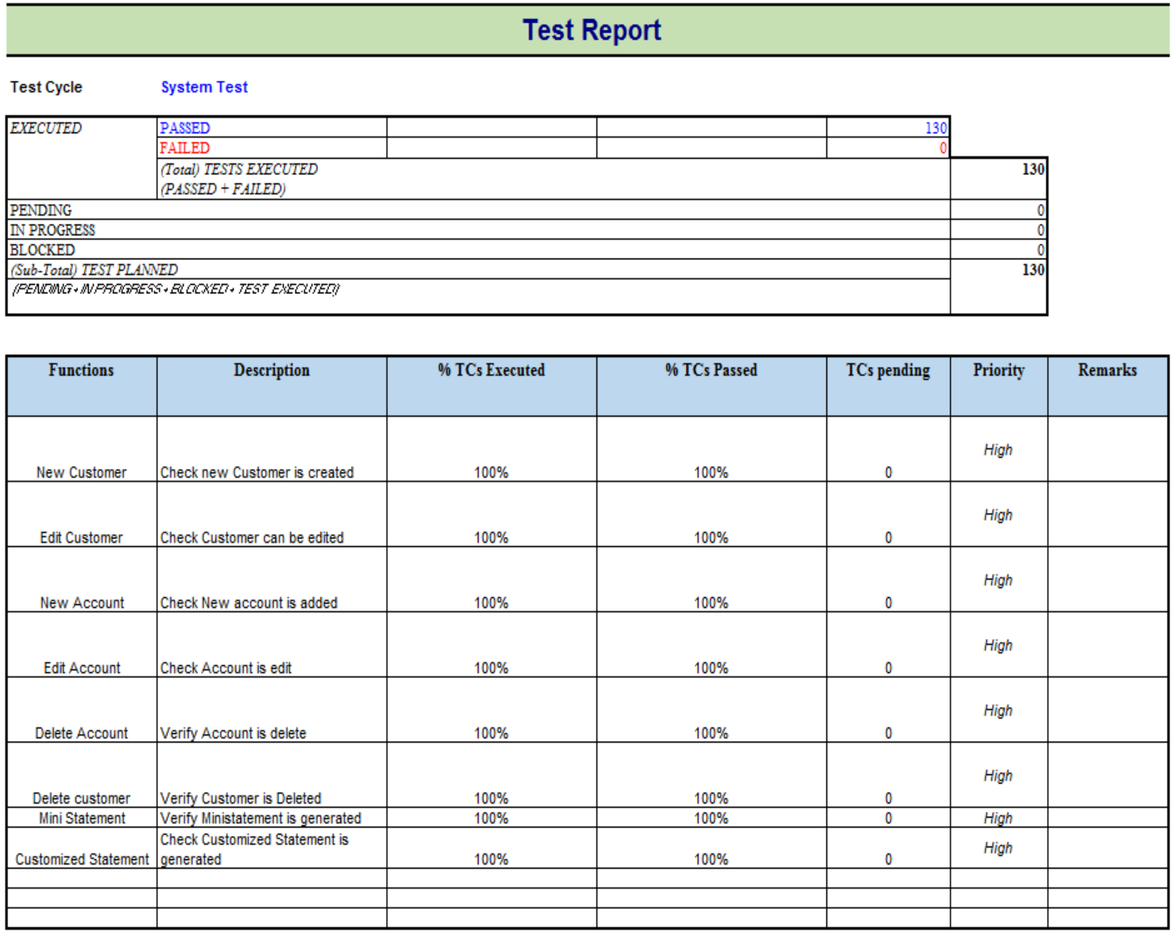
- APPIUM is a freely distributed open source mobile application UI Testing framework.
- Appium allows native, hybrid and web application testing and supports automation test on physical devices as well as an emulator or simulator both.
- It offers cross-platform application testing, i.e. single API works for both Android and iOS platform test scripts.
- It has **NO** dependency on Mobile device OS. Because APPIUM has framework or wrapper that translate Selenium Webdriver commands into UIAutomation (iOS) or UIAutomator (Android) commands depending on the device type, not any OS type.
- Appium supports all languages that have Selenium client libraries like- Java, Objective-C, JavaScript with node.js, PHP, Ruby, Python, C#, etc.

HOW APPIUM WORKS?

- Appium is an 'HTTP Server' written using a Node.js platform and drives iOS and an Android session using Webdriver JSON wire protocol. *Hence, before initializing the Appium Server, Node.js must be pre-installed on the system.*
- When Appium is downloaded and installed, then a server is set up on our machine that exposes a REST API.
- It receives connection and command request from the client and executes that command on mobile devices (Android / iOS).
- It responds back with HTTP responses. Again, to execute this request, it uses the mobile test automation frameworks to drive the user interface of the apps. A framework like:-
    - Apple Instruments for iOS (Instruments are available only in Xcode 3.0 or later with OS X v10.5 and later)
    - Google UIAutomator for Android API level 16 or higher
    - Selendroid for Android API level 15 or less

TEST REPORT

- **Test Report** is a document which contains a summary of all test activities and final test results of a testing project.
- Test report is an assessment of how well the Testing is performed.
- Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.
- For example, if the test report informs that there are many defects remaining in the product, stakeholders can delay the release until all the defects are fixed.



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*	

THANK YOU