Test Automation

SE3010

Software Engineering Process and Quality Management

Agenda

- Introduction
- Discussion
- What was covered so far
- Test automation
    - What is Test Automation?
    - Need for Automation Testing
    - Classifications of Test Automation
    - When/What Tests to Automate
    - Benefits of Test Automation
    - Automation Testing Life Cycle (ATLM)
    - Automation Testing Tools
    - Test Automation Frameworks
    - Barriers to Test Automation, why some fail?
    - Test Cases You Shouldn’t Automate
    - Limitation & Challenges of Automation Testing
    - Test Automation will not…
- Recap
- Next Week…

Introduction

- Why are you following this Module?
- What career paths will benefit from following this Module?

Technical Skills required for a Software Tester

- Agile and DevOps Methodology
- Automation Tests
- Knowledge of SDLC
- Preparing Test
  - ○ Test plan
  - ○ Test Scenarios
  - ○ Test cases
- Latest Technologies and Programming
- Rational and Logical Thinking
- Interpersonal Skills and Good Communication
- Documentation

Software Tester Job Description

- Analyze the system specification
- Contribute to the test plan creation
- Prepare test scenarios
- Document test cases
- Collected test data
- Create test batches
- Perform test cases
- Report defects
- Track defects
- Perform regression testing
- Alter test cases and repeat the process

What was covered so far…

- Specification-Based Test Case Design Techniques
- Code Coverage Analysis
- CC Measure
- WCC Measure
- CFS Measure
- Introduction to Test Automation and Tools
- Functional and Non-functional Testing

What is Test Automation?

- **Software testing technique to test and validate actual test result with expected test result with minimal or no touch.**

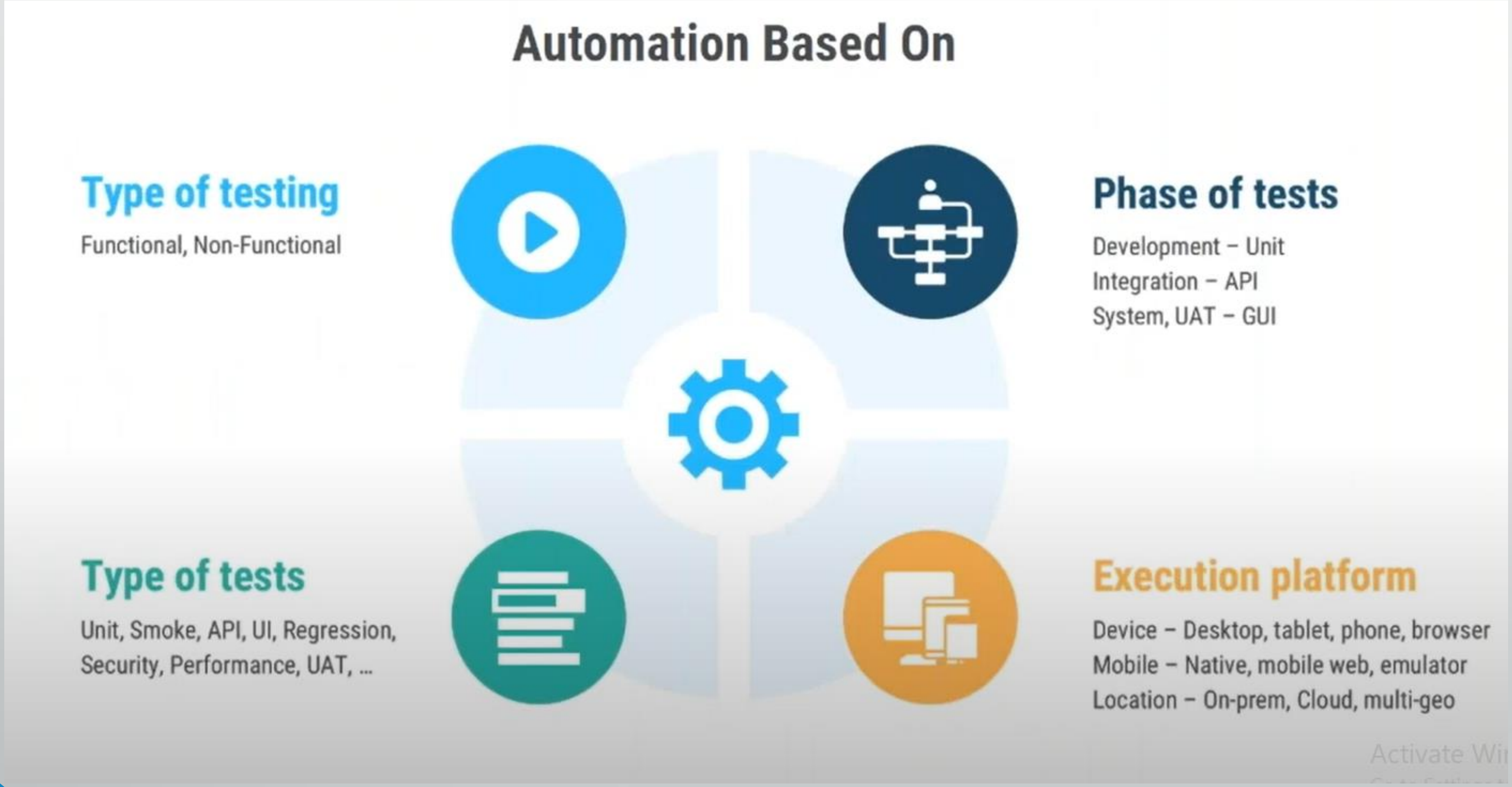
- Automation Testing is use of Tools or Software to perform software testing.
- Developing and executing tests that can run and compare actual to expected results.
- The automation software can also enter test data into the system under test, compare expected and actual results and generate detailed test reports.

Mekedi test data add krla values check krnna puluwan actual eka an expected eka . eken report
ekak generate karala gnna puluwan

Need for Automation Testing

- **Speed** : Automation scripts are faster when compared to manual testers effort.
- **Reliable** : Tests perform precisely the same operation each time they are run, there by eliminating human error.
- **Repeatable** : Tests can be repeated n number of times for execution of the same operation.
- **Coverage** : Automated tests increase the coverage.
- **Reusable** : We can reuse tests on different versions of an application, even if the user interface changes.

Classifications of Test Automation



Automation Based On

Type of testing

Functional, Non-Functional

Phase of tests

Development – Unit
Integration – API
System, UAT – GUI

Type of tests

Unit, Smoke, API, UI, Regression, Security, Performance, UAT, ...

Execution platform

Device – Desktop, tablet, phone, browser
Mobile – Native, mobile web, emulator
Location – On-prem, Cloud, multi-geo

When/What Tests to Automate

- **Regression Testing** : When the software application is fairly stable and only regression tests needs to be executed.
- **Smoke Testing** : For getting a high-level assessment of the quality of the build, and making quick go / no-go on further testing.
- **Static & Repetitive Tests** : For automating testing tasks that are repetitive and relatively unchanging from one test cycle to the next.
- **Data Driven Testing** : For testing application functions where the same function needs to be validated with lot of different inputs & large data sets (Ex: Login, & Search).
- **Load & Performance Testing** : No viable manual alternative exists.

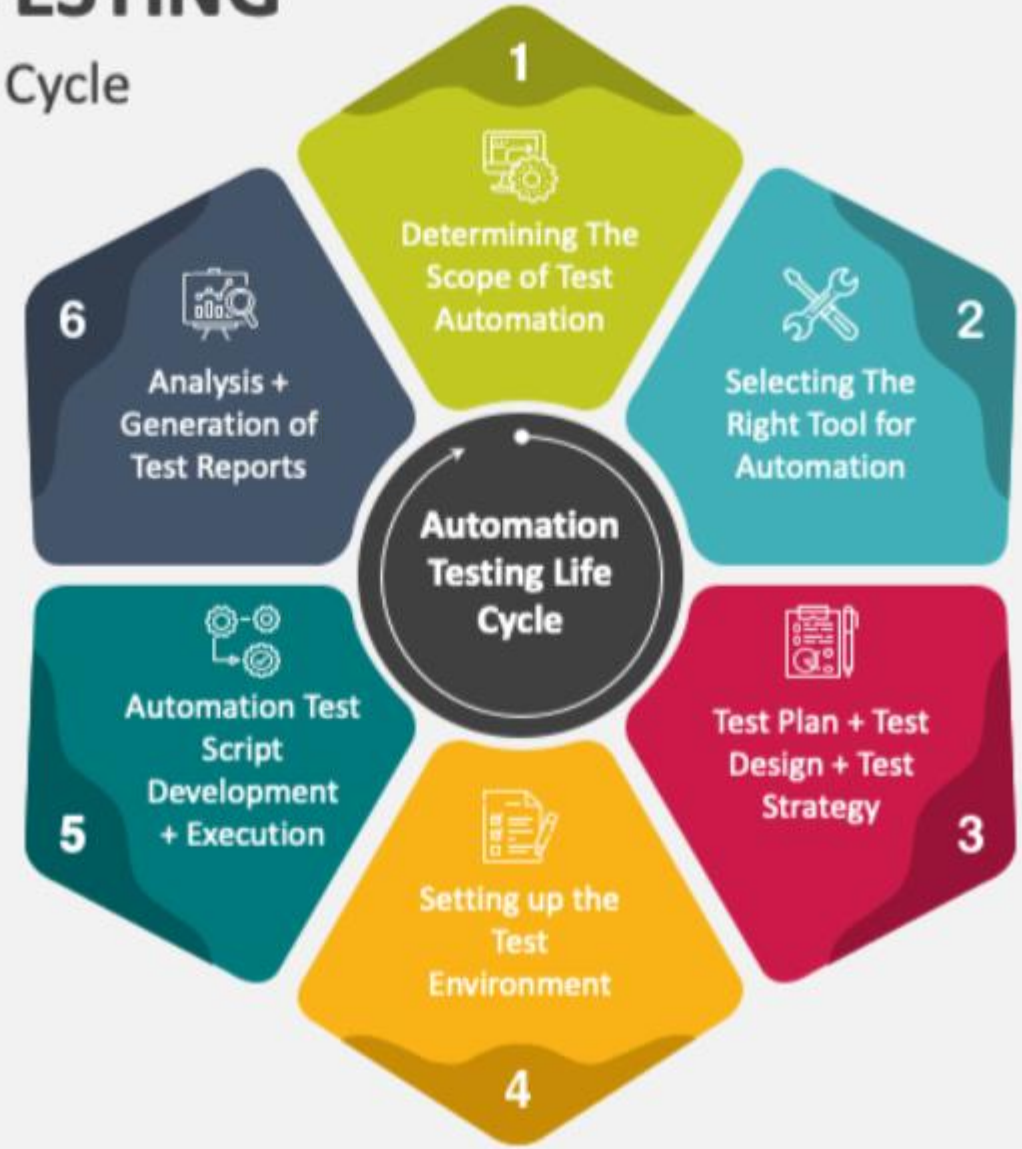
Benefits of Test Automation

- Faster Feedback
- Accelerated results
- Reduced business expenses
- Testing efficiency Improvements
- Reusability of Automated Tests
- Earlier detection of defects
- Thoroughness in Testing
- Faster time to market

AUTOMATION TESTING

Automation Testing Life Cycle

1. scope of test case
2. select right tool
3. test plan
4. setting test environment
5. test developement and execution
6. generation of test report

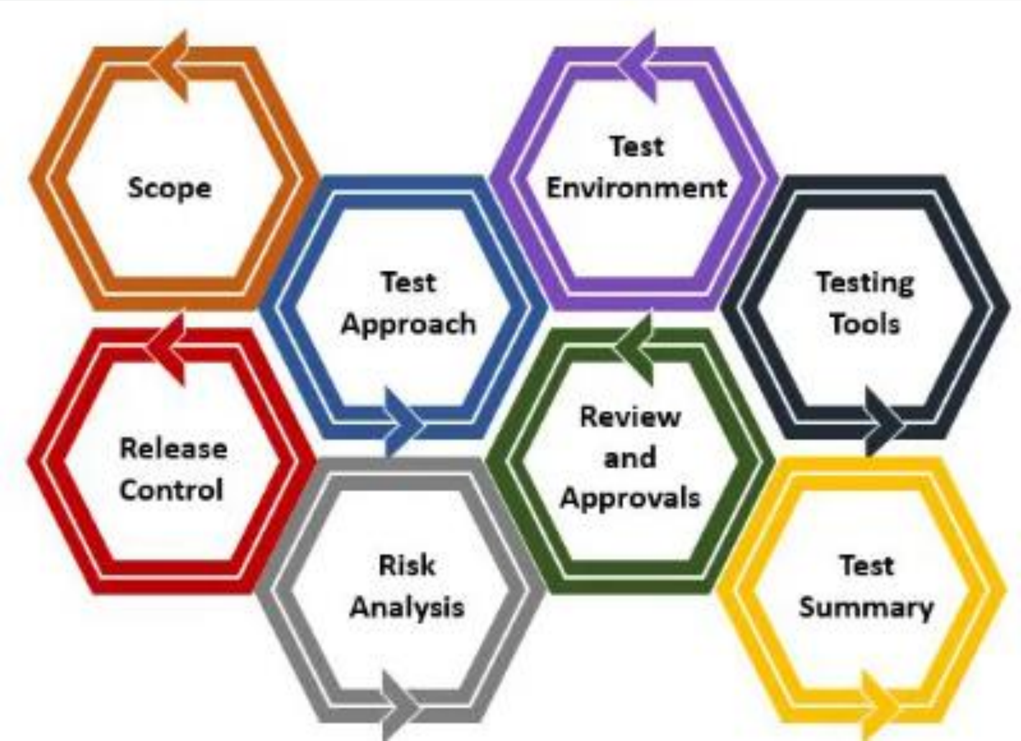

Automation Testing Life Cycle
1
Determining The Scope of Test Automation
2
Selecting The Right Tool for Automation
3
Test Plan + Test Design + Test Strategy
4
Setting up the Test Environment
5
Automation Test Script Development + Execution
6
Analysis + Generation of Test Reports

Stages of Automation Testing Life Cycle (ATLM)

1. **Determining the Scope of Test Automation** : Feasibility study of the automation testing process.
2. **Selection of the Appropriate Automation Tool for Test Automation** : Identifying the right automation testing tool is critical for the automation testing life cycle since automation testing is highly dependent on the tool used.
3. **Developing the Test Plan, Test Design, and Test Strategy** : Test automation framework & Test Management Strategy.
4. **The Test Environment should be Set Up** : Set up and install test environment software as well as link network resources and hardware.
5. **Developing the Automation Test Script and its Execution**
6. **Test Analysis and Generation of Test Results and Reports**

Test Strategy

- **Test strategy** is the document that describes the testing approach of the software product. The test strategy is created to inform project managers, developers, and testers about key issues of the testing process.
- Generally, a test strategy document contains below sections:

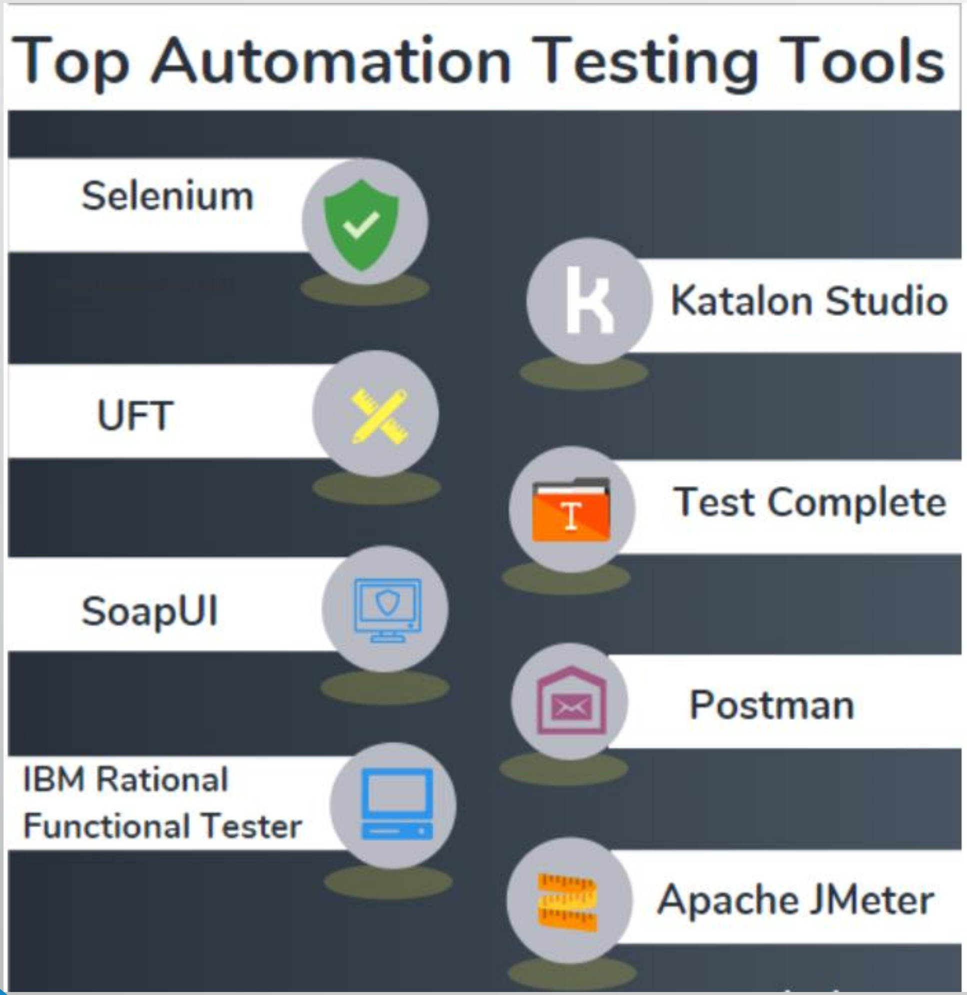


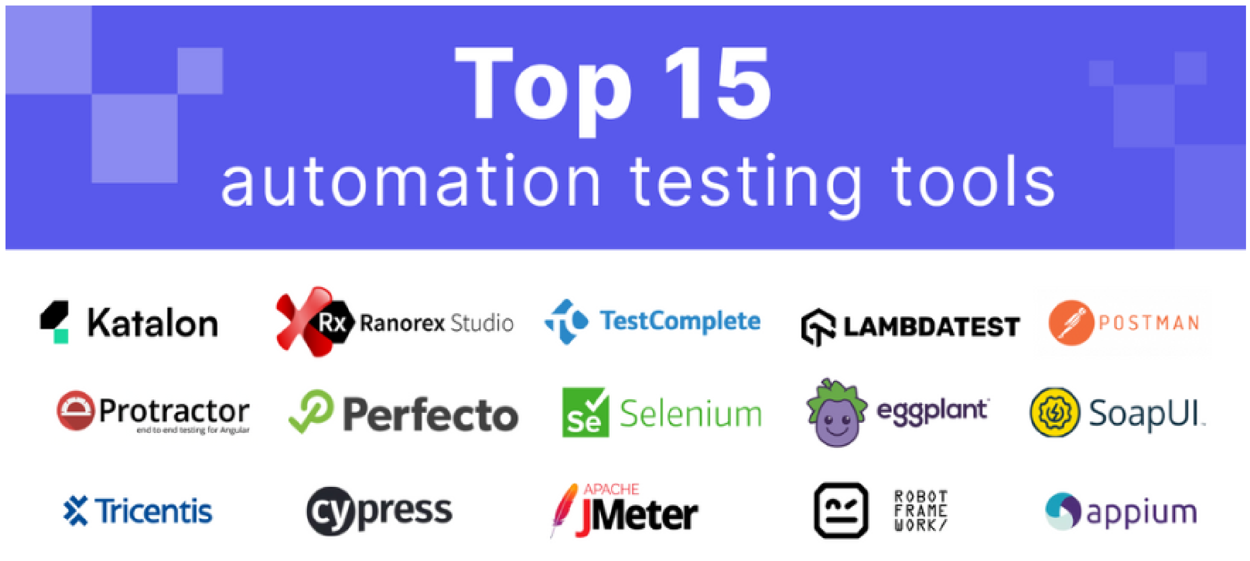
Important:

The automation testing cycle must be implemented stepwise to achieve the best results. Without sufficient planning, scheduling and a sequence of actions, large scripts are created, which often fail and require extensive manual intervention, putting a strain on existing human resources and increase production budgets and timelines.

Automation Testing Tools - I

1. Selenium
2. Postman
3. UFT
4. SoapUI
5. Apache Jmeter





Automation Testing Tools - II

1. **Selenium [https://www.selenium.dev/]** : Selenium is the most ubiquitous open-source automation testing tool, which might be used by almost 8 out of 10 test engineers. It is widely regarded as the industry standard for testing the user interface of web applications.

2. **Katalon Studio [https://katalon.com/]**: This tool addresses the disadvantages of Selenium and comes with much more advanced features. This tool is a potent and comprehensive tool for testing REST APIs, SOAP, web, and mobile applications. More advanced than selenium

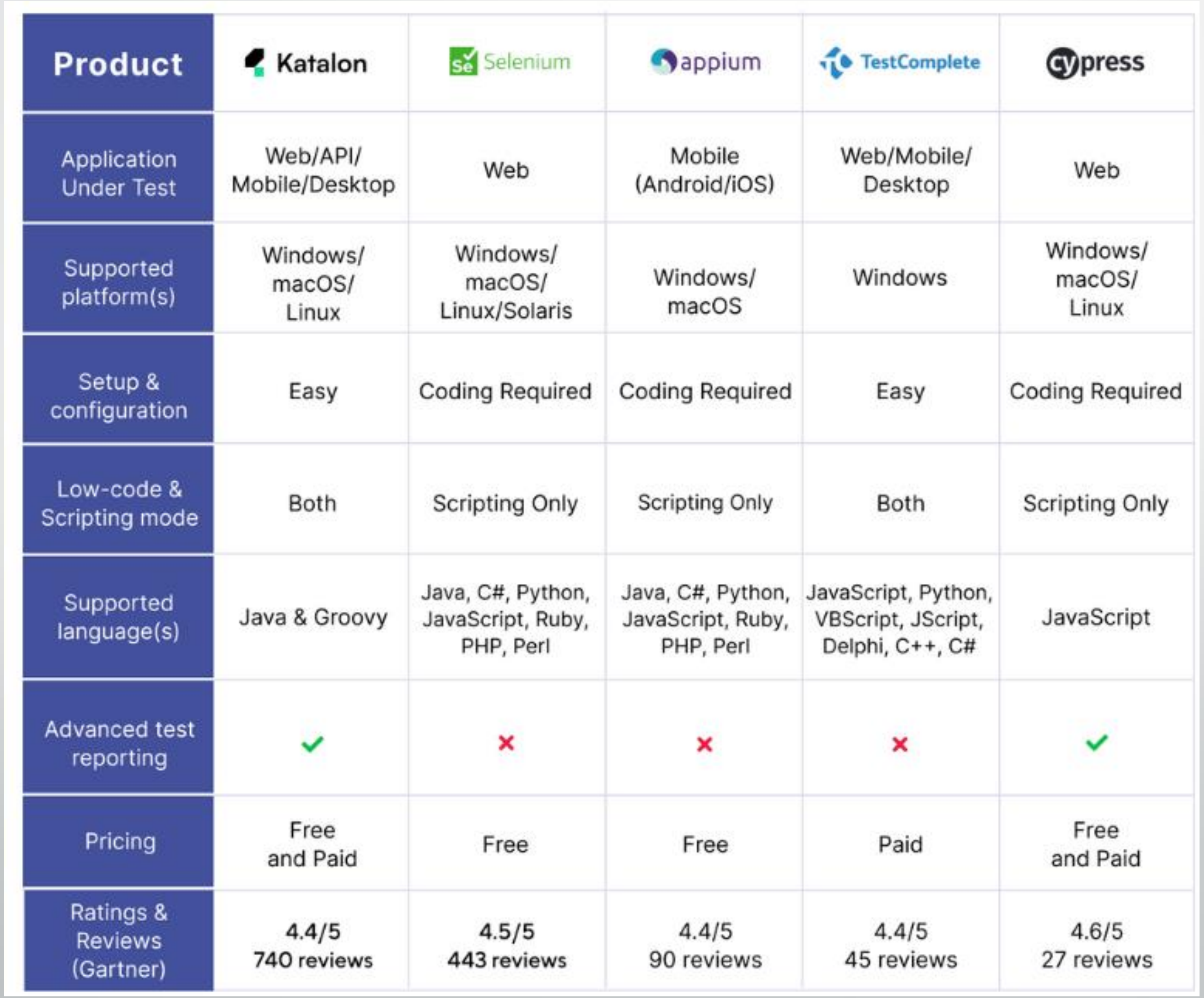
3. **UFT - Unified Functional Testing Tool [https://www.microfocus.com/en-us/products/uft-one/overview]** : Software developers extensively use it for function, regression, and service testing.

4. **Test Complete [https://smartbear.com/product/testcomplete/]** : one of the most powerful testing tools to test mobile, web, and desktop applications. The test engineer can use VBScript, JavaScript, Python, and C++ to develop their test suite.

5. **SOAP UI [https://www.soapui.org/]** : This is a tool for API testing. This supports both SOAP and REST protocols.

6. **Apache JMeter [https://jmeter.apache.org/]** : This was primarily designed for test loading and performance measurement. Nevertheless, the tool can also be extended to test API performance.

Automation Testing Tools – Comparison



Product	Katalon	Selenium	appium	TestComplete	cypress
Application Under Test	Web/API/ Mobile/Desktop	Web	Mobile (Android/iOS)	Web/Mobile/ Desktop	Web
Supported platform(s)	Windows/ macOS/ Linux	Windows/ macOS/ Linux/Solaris	Windows/ macOS	Windows	Windows/ macOS/ Linux
Setup & configuration	Easy	Coding Required	Coding Required	Easy	Coding Required
Low-code & Scripting mode	Both	Scripting Only	Scripting Only	Both	Scripting Only
Supported language(s)	Java & Groovy	Java, C#, Python, JavaScript, Ruby, PHP, Perl	Java, C#, Python, JavaScript, Ruby, PHP, Perl	JavaScript, Python, VBScript, JScript, Delphi, C++, C#	JavaScript
Advanced test reporting	✓	✗	✗	✗	✓
Pricing	Free and Paid	Free	Free	Paid	Free and Paid
Ratings & Reviews (Gartner)	4.4/5 740 reviews	4.5/5 443 reviews	4.4/5 90 reviews	4.4/5 45 reviews	4.6/5 27 reviews

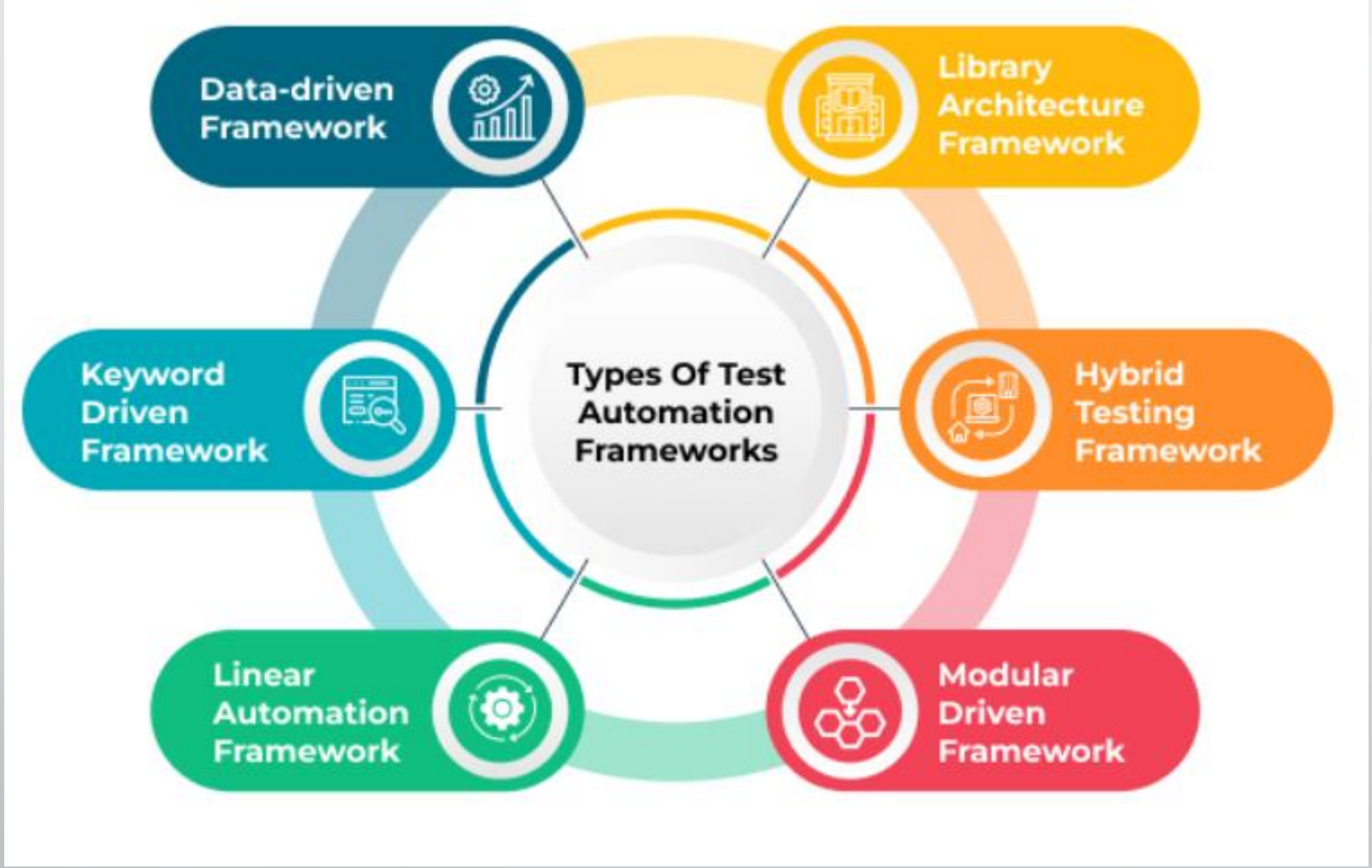
Test Automation Framework

me frameworks wala thyenne test professionalsla test case develop kraddi follow krnna one rules and guidelines dala hadpu tools

- **Test Automation Framework** is a set of tools used to establish rules and guidelines for developing test cases by test professionals. These rules may include coding standards, test data handling, storing test results, and other resources. The framework aids in developing test scripts and recording them.

- Test Automation Framework has several benefits, some of which are as follows:
  - Reduces the cost and lowers risks
  - Improves test efficiency
  - Lowers the cost of maintenance
  - Maximizes test coverage and functionality of the application
  - Defines the reusability of code
  - Avoids duplication of test cases used for automation across platforms
  - Automates testing of independent applications
  - Organizes test suites and enhances test efficiency

Test Automation Frameworks



Test Automation Frameworks

- **Linear Automation Framework** : With a linear test automation framework, also referred to as a record-and-playback framework, testers don't need to write code to create functions and the steps are written in a sequential order.
- **Modular Based Testing Framework** : Implementing a modular framework will require testers to divide the application under test into separate units, functions, or sections, each of which will be tested in isolation.
- **Library Architecture Testing Framework** : The library architecture framework for automated testing is based on the modular framework, but has some additional benefits. In this framework similar tasks within the scripts are identified and later grouped by function, so the application is ultimately broken down by common objectives. These functions are kept in a library which can be called upon by the test scripts whenever needed.
- **Data-Driven Framework** : Using a data-driven framework separates the test data from script logic, meaning testers can store data externally.
- **Keyword-Driven Framework** : In a keyword-driven framework, each function of the application under test is laid out in a table with a series of instructions in consecutive order for each test that needs to be run.
- **Hybrid Test Automation Framework** : As with most testing processes today, automated testing frameworks have started to become integrated and overlap with one another. Hybrid framework is a combination of any of the previously mentioned frameworks set up to leverage the advantages of some and mitigate the weaknesses of others.

Barriers to Test Automation, why some fail - I?

- **Resources & Priorities**
  - ○ Initial Investment (Cost, Time, & Effort)
  - ○ Competing corporate initiatives and priorities
  - ○ Lack of clear mandate and realistic goals
  - ○ Not treated like other software dev projects

- **Tools & Environment**
  - ○ Not using proper tools and framework
  - ○ Legacy or constantly changing code
  - ○ Not having controlled or stable test environment

1. Initial inverstment
2. Tools and environment
3. Culture and skillset
4. Process

Barriers to Test Automation, why some fail - II?

- **Culture & Skillset**
  - Teams attitude approach and resistance to change
  - Lack of experience and false sense of security
  - Relies on programming language only
  - Underestimate the amount of time needed
  - Creating large and end-to-end tests
- **Process**
  - Not reusing automation code
  - Not having a test data strategy in-place
  - Not making you automated tests readable

Test Cases You Shouldn’t Automate

- Tests that will be executed once
- Tests based on visual perception
- Tests without evident pass / fail results
- Anti Automation features, like CAPTCHA
- Raw and unstable functionality
- Features with changing requirements
- Newly designed test cases

Limitation of Automation Testing

- **Need for scripting and programming skills** – Coding and technical skill level of the resource should be good enough to write robust testing code.

- **Need for maintenance of code** – Whenever application code is updated or modified, the code for automated test case must also be updated.

- **Requires more initial developer time** – Any new test automation will require time for development, creation of frameworks etc…

- **Increase tool needs** – Automation testing would increase the need for tools (either licensed or Opensource) and also software required for the same.

Challenges of Automation Testing

- **Unrealistic Expectations** – Generally there is a tendency to be optimistic/have high expectation about what can be achieved by a new test tool.
- **Tool Limitations** – Tools available in the market have one or other limitation, which needs to be addressed.
- **Dependency on 3rd party integration** – Integrations with other applications, plugins, patches, etc… makes automation challenging.
- **Lack of Help & Support required for the tool** – Some tool have extensive support required for the tool, others have to reply on internet and other user forums.
- **Version compatibility for tool and browser** – The browsers are updated very rapidly in the market, but the tool supportability for the version make take considerable time.

Test Automation will not…

- Help a late project
- Pay off on the first release
- Completely replace manual testing
- Eliminate test planning
- Succeed without respect for basic software engineering practices

Next Week…

- How Software Quality Management fits to,
    - Traditional Models
    - Agile
    - SAFe
    - DevOps
    - SRE
    - Etc…

Thank You !!!

Tutorial – 07/04/2024

Q1: How would you differentiate between manual testing and Automation testing?

Q2: Do you expect automation testing to replace manual testing?

Q3: Can we achieve 100% automation possible in any case scenario?

Q4: Who should be responsible for test automation? Developers or the QA?

Q5: What are the different phases in an automation testing life cycle?

Q6: What is CAPTCHA?

Q7: What is continuous delivery?

Q8: What are the differences between open-source tools, vendor tools, & in-house tools in automation testing?

Tutorial – 07/04/2024 [Answers - I]

Q1: How would you differentiate between manual testing and Automation testing?

- Manual Testing Vs Automation Testing
  - Characterized by low accuracy and low reliability - Offers quick, reliable, and more proficient solutions
  - All tasks are performed by humans, so time-consuming - Tools perform the tasks, so comparatively, it takes less time
  - Low investment and lower ROI - Higher investment and higher ROI
  - Ideally suited for individual cases, exploratory, ad-hoc, and usability scenarios - Ideally suited for repetitive tasks, performance testing, load testing, and functional tests
  - The intervention of humans helps in customer experience - No guarantee of customer experience

Q2: Do you expect automation testing to replace manual testing?

- Both automation and manual are part of the overall testing process with their pros and cons. Though automated testing has added benefits, it still can't replace manual testing due to its limitation on several scenarios.

Tutorial – 07/04/2024 [Answers - II]

Q3: Can we achieve 100% automation possible in any case scenario?

- It is almost impossible to implement 100% automation to any testing procedures. Due to the presence of some real-life scenarios such as captcha matching that separates bots from humans or checking the background of the screen that will only take more time for more straightforward tasks. Manual testing is preferred in some cases. Generally, it is not recommended to use 100% automation in any case.

Q4: Who should be responsible for test automation? Developers or the QA?

- The web creation, process execution, and server management of test scripts fall under the team's purview, which is a single entity tasked with delivering a high-quality software system. Therefore, developers and quality assurance specialists should work together and complement one another's abilities to undertake automation testing properly.

Q5: What are the different phases in an automation testing life cycle?

- Determining the Scope of Test Automation --> Selection of the Appropriate Automation Tool for Test Automation --> Developing the Test Plan, Test Design, and Test Strategy --> The Test Environment should be Set Up --> Developing the Automation Test Script and its Execution --> Test Analysis and Generation of Test Results and Reports

Tutorial – 07/04/2024 [Answers - III]

Q6: What is CAPTCHA?

- CAPTCHA is a sort of security that stands for Completely Automated Public Turing Test to Tell Computers and Humans Apart. The CAPTCHA's principal objective is to protect you against spam or denial-of-service attacks by bots/scripts by requiring you to complete a simple test that computers find difficult to understand.

Q7: What is continuous delivery?

- As developers merge new code into the main branch, automation prepares any changes for immediate production, creating seamless transitions into subsequent build stages for better deployment efficiency. Continuous delivery expands on continuous integrations because it takes new code changes into a test environment, so that multiple dimensions of testing can occur beyond the unit testing phase (UI tests, integration tests, load tests, API reliability, etc).

Q8: What are the differences between open-source tools, vendor tools, & in-house tools in automation testing?

- **Open-Source Tools**: They are free tools with source code available on the internet. Example: Selenium
- **Vendor Tools**: These testing tools are developed by companies, and you need to purchase their licenses. Example: Microfocus UFT.
- **In-house Tools**: It is built by companies for their use.

Tips to Face a Successful Interview…

- **Prerequisites : To get called for an Interview,**

- **Preparation for the called Interview,**

- **Interview Day,**
  - ○ Start by answering the First Question Well – "***Tell me about yourself***"
    - ✓ Skills & Qualities,
    - ✓ Experience or Qualifications,
    - ✓ Significant achievements,
    - ✓ Type of person you are and what you're going to do if hired,

Thank You Again!!!