8 : Test Automation

IT6206-Software Quality Assurance

Level III - Semester 6

Overview

- This chapter is an introduction to test automation. Here we discusses what is test automation, why we need to automate software testing and how to implement it.

- Then we discusses about how to choose the right test automation tool.

- We also discuss about test environments and test environment management.

- Finally we discuses risks and benefits of test automation and introduce few widely used test automation tools.

Intended Learning Outcomes

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

- Explain what is test automation and how and why we should consider it.
- Choose a test automation tool for a given software application.
- Identify requirements for creating a test environment for a given software application.
- Identify risks and benefits of test automation.

List of sub topics

8.1 Introduction to Test Automation

8.2 Choosing the Right Tool

8.3 Effective Use of Tools

8.4 Test Automation Environments

8.5 Manual Testing vs Test Automation

8.6 Benefits and Risks of Test Automation

8.7 Test Tool Considerations

Introduction to Test Automation

8.1 Introduction to Test Automation

What is Automation?

- Automation Testing is the use of software tools to develop and execute tests.
- Tools used in test automation can enter test data into the applications, compare expected and actual results and generate reports.

Why to Automate?

- To improves accuracy, quality and coverage.
- To make test execution faster.

8.1 Introduction to Test Automation

When to Automate

- Regression Testing: Regression testing is used to test whether an application still functions as expected after code changes, or updates. When the software application is fairly stable automation can be used for regression testing.

- Smoke Testing: Automation can be used for smoke testing which is used for getting a quick high-level assessment on the quality of a software build and making go/no-go decision on further testing

- Static & Repetitive Tests: For automating testing tasks that are repetitive and relatively unchanging from one test cycle to the next

8.1 Introduction to Test Automation

- Data Driven Testing: Applications whit functions which needs to be validated with large number of different data sets (ex: login, search)

- Load & Performance Testing: No viable manual alternative exists

8.1 Introduction to Test Automation

Which Tests to Automate?

- Business Critical test cases
  - Business Scenarios and all critical flows which needs to be perfectly tested and function .
- Smoke Tests
  - Smoke Testing is a testing process that determines whether the deployed software build is stable or not. Smoke Tests are good candidates for Automation
- Test Cases that are very difficult to perform manually
  - Task with complex calculation, tedious steps are good candidates for automation
- Test Cases or Modules which are stable enough
  - Automating an application or module is not worth if it is not sufficiently stable.

8.1 Introduction to Test Automation

Types of Automation Tests

- Functional Automated Tests
  - Functional testing assures that software solution fulfills the given business requirements.

  (Functional Automation Tools: Selenium, UFT, RFT, VSTS)

- Non Functional Automated Tests
  - The non Functional Testing is carried out against the non functional requirements (Performance, security, compliance etc.)

  (Non-Functional Automation Tools: Jmeter, Load Runner)

Choosing the Right Tool

8.2 Choosing the Right Tool

Choosing the Right Tool

- Choosing the right automated testing tool is crucial for successful test automation. It's important to consider your specific requirements when selecting a tool.

- Following are some key factors to consider when choosing a testing tool:
  - Compatibility with your platforms and technologies. This includes considering the operating systems, application types, and mobile platforms you need to test.
  - User-friendliness and flexibility for testers of all skill levels. This includes assessing whether the tool supports keyword testing or if scripting expertise is required.

8.2 Choosing the Right Tool

- Look for tools that offer a range of features such as record-and-playback and checkpoint verification to make test creation and execution easier.
- Ability to create maintainable and reusable tests that can adapt to changes in UI. Ensure that the tests you create will remain effective even if there are changes in the application's user interface.
- Integration with your existing ecosystem. Ensure the tool integrates seamlessly with your CI/CD pipeline, test management framework, defect-management system, and source control.
- Ability to test enterprise applications. Consider whether the tool provides support for testing packaged applications such as SAP, Oracle, Salesforce, and Workday.

8.2 Choosing the Right Tool

Follow the below steps when choosing a test automation tool,

- Step 1: Identify the Requirements
  - Clearly define the requirements – what is the project, what is its scope, what kind of testing needed to be done, what are the functionalities you're looking for in the automation tool

- Step 2: Evaluate the Tools and Vendors
  - Consider the information gathered in the previous step to evaluate tools and create a list of tools the best fits for the requirements.

8.2 Choosing the Right Tool

- Step 3: Estimate Cost and Benefit
  - Preform a cost-benefit analysis for shortlisted tools.

- Step 4: Make the Final Decision
  - Despite reviewing the software manual and vendor information, it may be necessary to test the tool in your real-world work setting prior to purchasing a license.

  - To gain a more comprehensive understanding of the tool, it is advisable to schedule a meeting with the project team and consultants.

Effective Use of Tools

8.3 Effective Use of Tools

Training the Testing Team

- Once you have selected the appropriate tools and hired the necessary resources, the next step is to provide them with training.

- If manual testers are being transitioned to automation engineers, they will require training on automation concepts and terminology. If an automation architect is brought in from outside, they will need to familiarize themselves with the product being tested, the manual testing process, and the organization's expectations.

- It is important to allow team members time to experiment with different approaches and find the most effective automation strategy.

8.3 Effective Use of Tools

- Additionally, they should receive training on any existing tools being used by the organization, such as bug tracking or requirements management software.

- Effective training and clear communication between manual testers, developers, and the automation team are essential for success.

8.3 Effective Use of Tools

Creating the Test Automation Framework

- The primary responsibility of the automation architect is to develop a long-term automation framework to support automated testing.

- The automation framework is essentially a set of guidelines and strategies for creating test scripts that require minimal maintenance if there are any changes in the application. This is the key benefit of having an automation framework.

- There are different types of automation frameworks, including linear, modular, data-driven, keyword-driven, and hybrid.

8.3 Effective Use of Tools

Developing an Execution Plan

- The execution plan involves choosing the appropriate environments in which the scripts will be executed, including the operating system, browser, and various hardware configurations.

- For instance, if a test case requires checking a website on three different browsers - Chrome, Firefox, and IE - the automation team will write the script in a manner that allows it to run on each browser.

- It is crucial to communicate this information prior to scripting so that the automation team can address it appropriately in the scripts.

8.3 Effective Use of Tools

- The execution plan should also specify who will execute the scripts. Typically, the automation team performs the script execution for each build, but this can vary among companies. Some managers require developers to execute these scripts before release, and some companies appoint a dedicated resource specifically for execution. Some companies even run scripts in an unattended mode, which requires no additional resources.

8.3 Effective Use of Tools

Writing Test Scripts

- Once the automation framework is designed, the execution plan is established, and resources are trained on the chosen tool, it's time to begin script writing.

- The scripts should be written in an orderly fashion, adhering to proper naming conventions.

- The source code should be kept in source control to prevent code loss, and version control and history should be maintained.

- Test automation is akin to software development, and therefore all best programming practices should be followed when writing scripts.

8.3 Effective Use of Tools

Reporting

- The automated testing tool usually provides a reporting feature, but it's also possible to develop custom reporting mechanisms, such as automatically emailing the results to management.

- Reports can be generated at the conclusion of each execution, in the form of charts and tables, if management requires them.

- It's important to keep management informed about the test case coverage, including which manual test cases have been automated and which ones still need to be automated.

8.3 Effective Use of Tools

Maintenance of Test Scripts

- When following best programming practices and using a good framework, maintenance is usually not a problem. However, maintenance may be required if there is a change request in the application. In this case, scripts should be updated immediately to ensure that they continue to run without any issues.

- For example, if a textbox in the application is changed to a drop-down list, the script should be updated accordingly. Similarly, if the application needs to support a different language, the scripts should be updated to reflect this change.

- If the framework is not good and best practices are not followed, maintenance can become a nightmare, and this is often the reason why many automation projects fail.

8.3 Effective Use of Tools

- Automation architects are typically expensive because they possess the knowledge and experience to create a robust and easy-to-maintain automation framework.

Test Automation Environments

8.4 Test Automation Environments

- Test Automation Environments

- A test environment or test bed refers to a combination of software and hardware that testing teams utilize for conducting tests.

- Sometimes, a test bed can include both a test environment and test data.

- Establishing the appropriate test environment is crucial for achieving success in software testing. Any mistakes made during this stage could result in additional expenses and work for the client.

8.4 Test Automation Environments

- A test environment can consist of various components, such as,
  - Systems and applications,
  - Test data,
  - Database servers,
  - Client operating systems,
  - Browsers,
  - Server operating systems,
  - Hardware,
  - Networks,
  - Necessary documentation like reference documents, configuration guides, installation guides, and user manuals.

8.4 Test Automation Environments

Process of Software Test environment setup

- Tests are limited to what can be tested and what not should be tested.

- Following people are involved in test environment setup
  - System Admins,
  - Developers
  - Testers
  - Sometimes users or techies with an affinity for testing.

- The test environment requires setting up of various number of distinct areas like,

8.4 Test Automation Environments

- Step 1) Setup of Test Server
  - Not all tests can be performed on a local machine, and in some cases, a test server may need to be set up to accommodate applications. (ex: CentOS set up for PHP, Java-based applications with or without mail servers, cron set up, Java-based applications, etc.)

- Step 2) Network
  - Network set up as per the test requirement including,
    - Internet
    - LAN/Wifi
    - Private network setup
  - This ensures that other team members are not impacted by congestion that may occur during testing. (Developers, designers, content writers, etc.)

8.4 Test Automation Environments

- Step 3) Test PC setup
  - Test PCs are set up in accordance with the applications being tested.
  - Example:
    - For desktop application testing, you may need various types of OS for different testers PCs.
    - For web application testing, you may need to set up different browsers for different testers.

- Step 4) Bug Reporting
  - Testers should be provided with bug reporting tools.

8.4 Test Automation Environments

- Step 5) Creating Test Data for the Test Environment
  - Numerous companies employ a distinct test environment to evaluate their software products. Typically, production data is duplicated for testing purposes, allowing testers to identify identical issues as those that may arise on a live production server, without causing any damage to the production data.

  - The procedure for replicating production data to test data involves several steps, such as configuring production jobs to transfer data to a shared test environment, modifying any Personally Identifiable Information (PII) and other sensitive data, replacing PII with non-personal data that is logically accurate, and removing any data that is not pertinent to the testing process.

8.4 Test Automation Environments

- The primary concern with copying production data is privacy. To address this issue, one could explore the use of obfuscated and anonymized test data.

- For Anonymization of data two approaches can be used,
  - BlackList: All the data fields are left unchanged except the fields specified by the users.
  - WhiteList: All data fields are anonymized except for a list of fields which are allowed to be copied.

- Moreover, when utilizing production data, it is important to be strategic about how to obtain the data. A useful method is to query the database using SQL scripts.

8.4 Test Automation Environments

Test Environment Management

- Test Environment Management involves the management and maintenance of the test environment.

- The Test Environment Management function comprises several activities, such as,
  - Maintaining an up-to-date central repository of test environments,
  - Managing test environments based on the test team's needs,
  - Establishing new environments to meet new requirements, monitoring the environments,
  - Removing outdated test environments,
  - Investigating any issues with the environments,
  - Coordinating until a resolution is reached.

8.4 Test Automation Environments

Challenges in setting up Test Environment Management

- Proper planning on resource usage:
  - Inadequate resource usage planning can have a negative impact on the final output, and it may also result in conflicts between teams.

- Remote environment:
  - In certain scenarios, a Test environment may be situated in a different geographic location, which requires the testing team to depend on the support team for various test assets. (Software, hardware, and other issues).

- Elaborate setup time:
  - Integration testing may result in complex test setups at times.

8.4 Test Automation Environments

- Shared usage by teams:
  - When the development and testing teams use the testing environment concurrently, the test results may become invalid or corrupted.

- Complex test configuration:
  - Some tests demand intricate test environment configurations, which can present a challenge to the testing team.

8.4 Test Automation Environments

Best practices for Test Environment Management

- Thoroughly comprehend the test requirements and educate the test team members accordingly.
- Verify connectivity before beginning testing.
- Ensure that the necessary hardware, software, and licenses are available, as well as the required browsers and versions.
- Plan the scheduled use of the test environment, as well as the automation tools and their configurations.

Manual Testing vs Test Automation

8.5 Manual Testing vs Test Automation

Criteria	Manual Testing	Automation Testing
Accuracy	The likelihood of human errors in manual testing results in reduced accuracy.	Computer-based automation testing eliminates the possibility of human errors, resulting in increased accuracy.
Testing at Scale	Time increases significantly when testing is at a large.	Large scale testing is less time consuming.
Turnaround time	Due to the longer time required to complete a testing cycle, manual testing results in increased turnaround time.	Automation testing can complete a testing cycle at a rapid pace, resulting in a significantly reduced turnaround time.
Cost Efficiency	More costly since it involves the hiring of experts to perform testing.	Less costly since once the software infrastructure is in place, it works for a long time.

8.5 Manual Testing vs Test Automation

Criteria	Manual Testing	Automation Testing
User Experience	Manual testing can provide the end user with a superior user experience as it involves human observation and cognitive skills.	As automation testing cannot ensure a good user experience since machines lacks human observation and cognitive abilities.
Areas of Specialization	Manual Testing should be used to perform Exploratory Testing, Usability Testing, and Ad-hoc Testing to exhibit the best results.	Automation Testing should be used to perform Regression Testing, Load Testing, Performance Testing and Repeated Execution for best results.
User Skills	Users must have the ability to mimic user behavior and build test plans to cover all the scenarios.	Users must be skilled at programming and scripting to build test cases and automate scenarios.

Benefits and Risks of Test Automation

8.6 Benefits and Risks of Test Automation

Advantages of Test Automation

- Speed : Test Automation is faster when compared to manual testing.
- Reliable : Automated tests can perform same operation precisely each time.
- Repeatable: Tests can be repeated easily.
- Coverage: Test Automation increase coverage.
- Reusable : Test can be reused in different application versions.

8.6 Benefits and Risks of Test Automation

When is the Test Automation Beneficial.

- Long Run projects
  - Automated tests have the ability to execute quickly and repeatedly, making it a cost-efficient option for software products that require long-term maintenance.
- Loads of Regression Testing
  - Reusability of tests is good for running regressions on constantly changing code.
- When doing manual tests is time consuming and complex
  - When time limitations exist, it may be impractical to perform a comprehensive manual test of all features prior to releasing a software or web application, raising concerns about whether major flaws have been identified or not.
- When application is fairly stable.

8.6 Benefits and Risks of Test Automation

Risks of Automation Testing

- To have overly ambitious or impractical anticipations from the automated testing tools.
- Overlooking the capability of human testers in specific testing scenarios.
- Incorrect evaluation of time and effort required for factors such as:
  - Selection and introduction of automation testing tools
  - Training the software testers
  - Adjustments to the testing procedures within the organization when integrating the automation testing tool
  - Maintainability of test scripts and test results.

8.6 Benefits and Risks of Test Automation

- Incompatibility of the automation testing tools with the test environment.
- Project budget issues.
- Vendor issues:
  - Unavailability of technical support
  - Unavailability of updates for automation tools
  - Liquidation and take over of the vendor organization
  - A free tool is made in to a licensed version

Test Tool Considerations

8.7 Test Tool Considerations

Tools available for Test Automation

Licensed Tools	Open Source
Appium	Testim
Tellurium	Selenium
QTP	Jmeter
Test Complete	Open STA
Win Runner	Soap UI
Rational Functional Tester	
VSTS	
Silk Test	

8.7 Test Tool Considerations

Some of the most commonly used mobile test automation tools are,

- Appium
  - Appium is a freely available automated testing tool that is designed for testing mobile web applications as well as hybrid and native applications. You can create your tests on Appium using various development tools, and it can work with all major programming languages.

- QTP
  - Commonly used for web, mobile, & desktop applications. QTP offers functional & regression testing. It can also be used for API testing.

8.7 Test Tool Considerations

- Selenium
  - Selenium is a highly favored test automation tool widely used in the industry. It permits users to write scripts in diverse programming languages, such as Java, C#, Python, Perl, and Ruby. It also works well with various operating systems and web browsers. Selenium provides different versions tailored to various needs, making it a versatile option for different projects.

- TestComplete
  - TestComplete is a test automation tool that supports testing of web, mobile, and desktop applications. It is commonly preferred by users who use JavaScript, VBScript, Python, or C++Script for writing their tests. TestComplete is especially suitable for testing applications that frequently undergo changes in the user interface.

8.7 Test Tool Considerations

- Testim
  - Testim is an automated testing tool that utilizes machine learning to create, run, and maintain tests. It constantly learns and improves from each execution, resulting in more stable test cases. Additionally, Testim is capable of detecting even the smallest changes in page elements to prevent test failures.

Summary

- Test Automation is to use of Tools or Software to develop and execute tests that can run and compare the actual to expected results.

- Test Automation can be used in Regression Testing, Smoke Testing, Static & Repetitive Tests, Data Driven Testing and, Load & Performance Testing.

- Test environment or Test Bed is a software and hardware arrangement used by testing teams to carry out test cases.

- Test environment includes test server setup, network setup, test pc setup, bug reporting tools and test data.

- Advantages of test automation includes increased speed and reliability, repeatability, increased coverage and reusability.