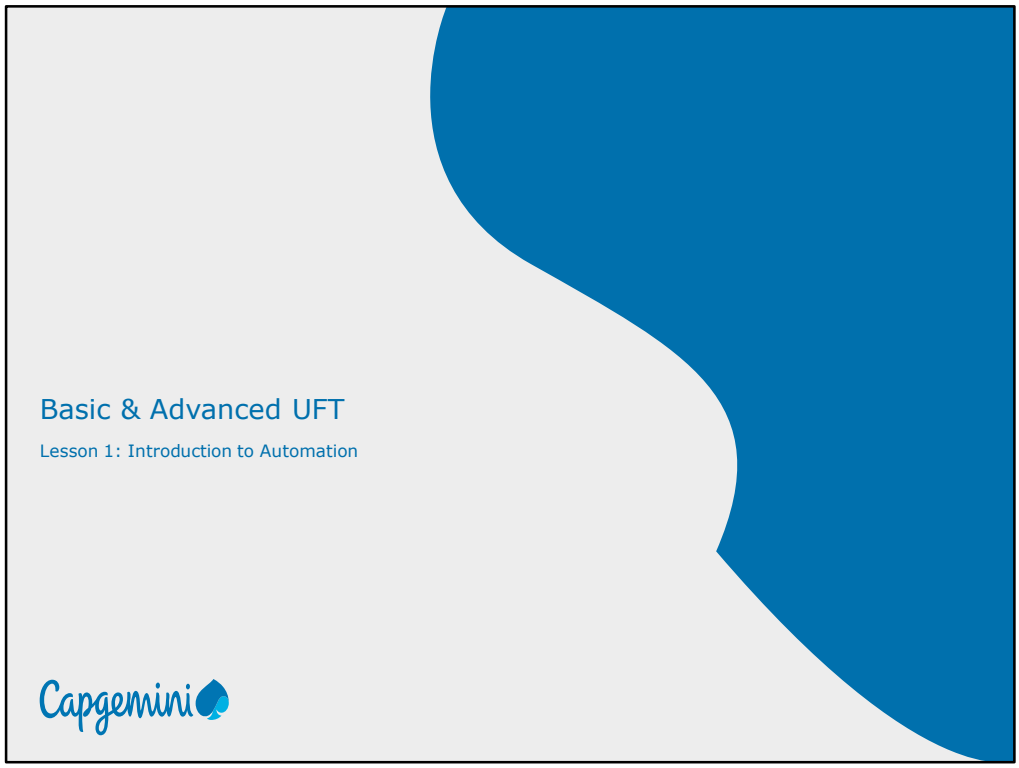

Basic & Advanced UFT
Lesson 1: Introduction to Automation
Capgemini

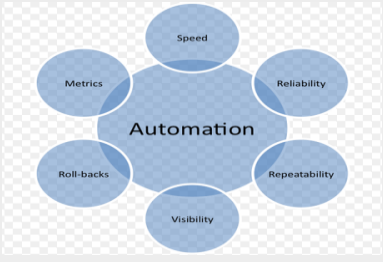
Lesson Objectives

- What is Automation?
- What is Test Automation?
- Why to Automate?
- Manual Testing Vs Automation Testing
- Manual To Automated Testing – The Process
- Advantage of Automation Testing
- What Should Be Automated?
- Automation Testing – Best Practices
- Common Misconceptions About Automated Testing
- Various Automation Testing Tools

1.1: Introduction To Automation

What is Automation?

- The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is automation applied to an inefficient operation will magnify the inefficiency." - Bill Gates



- Automation is the linking of disparate systems and software in such a way that they become self-acting or self-regulating.

1.1: Introduction To Automation

What is Test Automation?

- It is a process in which automation tools run the tests suite, predefined actions on a software application, reporting outcomes, comparative results and generate detailed test reports.
- Automated testing or test automation is a method in software testing that makes use of special software tools to control the execution of tests and then compares actual test results with predicted or expected results.
- Automation is used to add additional or complex validation that may be too difficult to perform manually.
- If the project prospects and results align, your project is behaving as it should, and you are likely bug free. If the two don't align, still, there is a problem that requires to be addressed.
- You'll have to take a look at your code, alter it, and continue to run tests until the actual and expected outcomes align.

What is Test Automation?

Testing is a very important phase in the development process. It ensures that all the bugs are ironed out and that the product or software, is functioning as expected or as close to the target performance as possible.

In the organization, every software development project team tests the software or application, yet one can not ensure that the software is 100% bug free. Test engineers strive to catch these bugs before the product is released but they always creep in and they often reappear, even with the best manual testing processes. Automating testing efforts is the best way to increase the effectiveness, efficiency and coverage of your software testing.

Manual testing requires physical time and effort to ensure the software code does everything it's supposed to do. In addition, manual testers have to make a record of their findings. This involves checking error log files, trace files etc. Automation testing differs from manual testing where a human being is responsible for single handedly testing the functionality of the software in the way a user would. Automated testing is done through an automation tool, less time is needed in exploratory tests and more time is needed in maintaining test scripts while increasing overall test coverage.

1.1: Introduction To Automation

Why to Automate?

- Frequent regression testing
- Virtually unlimited execution of test cases is required
- Rapid feedback to developers
- Reduction in human efforts
- Test same application in multiple environment
- Finding defects missed in manual testing

Why to Automate?

Manual software testing is performed by a human sitting in front of a computer carefully going through application screens, trying various usage and input combinations, comparing the results to the expected behavior and recording their observations. Manual tests are repeated often during development cycles for source code changes and other situations like multiple operating environments and hardware configurations. An automated testing tool is able to playback pre-recorded and predefined actions, compare the results to the expected behavior and report the success or failure of these manual tests to a test engineer. Once automated tests are created they can easily be repeated and they can be extended to perform tasks impossible with manual testing.

1.1: Introduction To Automation

Manual Testing Vs Automation Testing

- Manual Testing
  - Time consuming and tedious
  - Low reliability
  - Human resources
  - Inconsistent
  - Critical bugs escape undetected
  - What happens when multiple platforms involved
  - Delay the ability in thoroughly testing an application
- Automation Testing
  - Speed & Repeatability
  - Programming capabilities
  - Coverage
  - Reliability & Reusability
  - Higher efficiency & product quality
  - Easy to focus on all possible workflows
  - Delivers: Reusability, Consistency and Productivity

Test Automation and its Importance in projects:

Why automate Testing?

In today´s fast moving world, it is a challenge for any company to continuously maintain and improve the quality and efficiency of software systems development. In many software projects, testing is neglected because of time or cost constraints. This leads to a lack of product quality, followed by customer dissatisfaction and ultimately to increased overall quality costs.

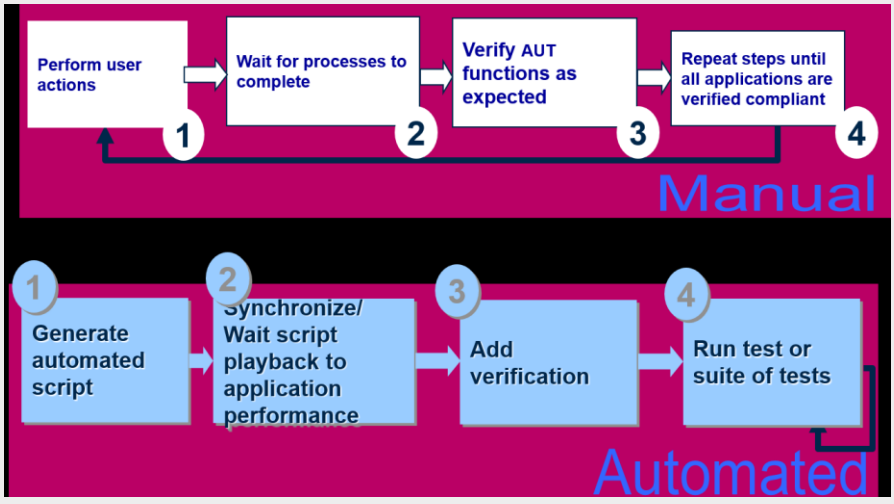
The main reasons for these added costs are primarily:

1. Poor test strategy
2. Underestimated effort of test case generation
3. Delay in testing
4. Subsequent test maintenance

Test automation can improve the development process of a software product in many cases. The automation of tests is initially associated with increased effort, but the related benefits will quickly pay off. Automated tests can run fast and frequently, which is cost-effective for software products with a long maintenance life. When testing in an agile environment, the ability to quickly react to ever-changing software systems and requirements is necessary. New test cases are generated continuously and can be added to existing automation in parallel to the development of the software itself. In both manual and automated testing environments test cases need to be modified for extended periods of time as the software project progresses. It is important to be aware that complete coverage of all tests using test automation is unrealistic. When deciding what tests to automate first, their value vs. the effort to create them needs to be considered. Test cases with high value and low effort should be automated first. Subsequently test cases with frequent use, changes, and past errors; as well as test cases with low to moderate effort in setting up the test environment and developing the automation project are best suited for automation.

1.1: Introduction To Automation

Manual To Automated Testing – The Process



1.1: Introduction To Automation

Advantage of Automation Testing

- Optimization of Efficiency & Quality
  - Quick Return on investment (ROI) of Test Automation.
- Advances a Tester's Motivation and Efficiency
  - More efficient Assignments of QA Tasks.
- Increase of Test Coverage
  - Different types of testing to increase test coverage.
- Testing Improves Accuracy
  - Automated tests perform the test steps precisely every time and each time and never forget to record detailed results.
- Tests can run fast and frequently
  - Cost-effective for software products with a long maintenance life
- Useful in agile environment
  - Helps in providing quick feedbacks to the development team & facilitates "shift left" approach in agile projects.

Advantages of Automation Testing

Optimization of Speed, Efficiency, Quality and the Decrease of Costs: The main goal in software development processes is a timely release. Automated tests run fast and frequently, due to reused modules within different tests. Automated regression tests which ensure the continuous system stability and functionality after changes to the software were made lead to shorter development cycles combined with better quality software and thus the benefits of automated testing quickly outgain the initial costs.

Advances Tester's Motivation and Efficiency: Manual testing can be mundane, error-prone and therefore become exasperating. Test automation alleviates tester's frustrations and allows the test execution without user interaction while guaranteeing repeatability and accuracy. Instead testers can now concentrate on more difficult test scenarios.

Increase of Test Coverage: Sufficient test coverage of software projects is often achieved only with great effort. Frequent repetition of the same or similar test cases is laborious and time consuming to perform manually.

Testing Improves Accuracy: Even the most conscientious tester will make mistakes during monotonous manual testing. Automated tests perform the same steps precisely every time they are executed and never forget to record detailed results. Testers freed from repetitive manual tests have more time to create new automated software tests and deal with complex features.

1.1: Introduction To Automation

What Should Be Automated?

- Good Candidates
  - Repetitive tests that run for multiple builds (Sanity check, regression tests).
  - Tests that tend to cause human error.
  - Tests that require multiple data sets (Data driven tests).
  - Frequently used functionality that introduces high risk conditions.
  - Tests that are complex & time consuming to perform manually.
  - Tests that run on several different hardware or software platforms and configurations.
  - Tests requiring a great deal of precision.
- Bad Candidates
  - Tests that are newly designed and not executed manually at least once.
  - Tests for which the requirements are changing frequently.
  - Tests which are executed on ad-hoc basis.
  - Usability Testing.
  - Ad-hoc/Random Testing – based on intuition and knowledge of application.

1.1: Introduction To Automation

Automation Testing – Best Practices

- Decide which test cases to automate
- Design tests before automating
- Test early and regularly
- Select the right automation testing tool
- Proper division of testing efforts
- Create good quality test data
- Don’t automate unstable functionality
- Don’t rely solely on automation – Beware of Passing Tests
- Use test techniques in test automation
- Don’t automate chaos
- Create automated tests that are resistant to changes in the UI

Automation Testing – Best Practices

Decide which test cases to automate: It is impossible to automate all testing, so it is important to determine what test cases should be automated first. There are certain tests that are better done manually. Before going for automated testing, check that the test is suitable for automated testing. A risk-based approach can be used to identify the most important business cases and scenarios. The benefit of automated testing is linked to how many times a given test can be repeated. Tests that are only performed a few times are better left for manual testing. Good test cases for automation are ones that are run frequently and require large amounts of data to perform the same action.

Design tests before automating: Good test design helps in identification of maximum defects and so designing of tests should be done with utmost care and diligence.

Test early and regularly: Start testing as soon as possible. Early testing allows for early identification of bugs, which is both easier and more cost-effective to fix in the long run. Further, in order to have best results, testing needs to be repeated regularly.

Select the right automation testing tool: Selecting an automated testing tool is essential for test automation. There are a lot of automated testing tools on the market, and it is important to choose the automated testing tool that best suits your overall requirements.

1.1: Introduction To Automation

Common Misconceptions About Automated Testing

- Automation will provide you with more free time
  - The misconception that automated testing will give you more free time is both true and false. The time spent on the mundane tasks and repetition a manual tester would go through is instead spent focusing on larger, more important issues involving the software you're developing in automation testing.
- The Cost of Automated Testing is Too High
  - At first, the investment in automated testing might seem cost prohibitive, especially if you're a smaller company. But analysis has shown that, over time, automated testing pays for itself.
- Automated Testing is here to replace Manual Testing
  - Usage of tools will not replace manual testing process. It is here to make repeated work easier, bringing more coverage & efficiency as far as testing applications is concerned.
- Automated Testing is Better Than Manual Testing
  - The reality is, there is no "better" or "worse" in the automated vs. manual debate, there's just "different." Each approach has its own advantages and disadvantages. In the end, both manual and automated testing have their roles, especially if the software you're developing is too large and too complex to rely just on the manual approach.

1.2: Various Automation Testing Tools

Automation Testing Tools

Functional Testing Tool (FTT)	Performance Testing Tool (PTT)	Mobile Automation Testing Tool (MATT)
Google, Selenium (OS)	Apache Jmeter	Appium (OS)
HP UFT (CT)	HP LoadRunner (CT)	HP UFT (CT)
CUIT Coded UI Test (CT)	NeoLoad	Robotium (OS)
Telerik Test Studio	Telerik Test Studio	Monkey Runner
Ranorex Studio	Web Load	Ranorex Studio
Sahi (OS), Sahi Pro (CT)	Load View	UI Automator
SoapUI (OS)	Load Complete (CT)	Selendroid
SmartBear Test Complete (CT)	SmartBear LoadUI Pro, Load Ninja	SmartBear Test Complete (CT)
IBM RFT (CT)	IBM RPT (CT)	MonkeyTalk
Watir (OS)	Testing AnyWhere	Calabash
Borland Silk Test (CT)	Borland Silk Performer (CT)	Experitest
Tosca (CT)	Soasta CloudTest	eggPlant
Sauce Labs (CT)	OpenSTA	Sauce Labs (CT)

OS – Open Source Tool
CT – Commercial Tool/ Licensed Tool

1.2: Various Automation Testing Tools

Management Tools

Test Management Tool	Bug/Defect/Issue Tracking Tool	Configuration Management Tool (CM)	Continuous Integration Tool (CI)
HP ALM (CT)	HP ALM(CT)	Puppet (OS)	Jenkins
Tricentis -qTest	Backlog	CFEngine (OS)	TeamCity
IBM - RQM	IBM - RCQ	Ansible	GitLab
Test Rail	Bugzilla	SaltStack	Travis
TestFLO for Jira	Jira	Bamboo	Codeship
Smartbear - Zephyr	Mantis	Docker	
PractiTest	Redmine	Chef	
Test Collab	Trac		
QA Complete			
Silk Center			

The goals of Software Configuration Management are generally Configuration, Identification, Configuration idioms and baselines, configuration control, implementing a control change process.

Ref : https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/

CI Tools
https://code-maze.com/top-8-continuous-integration-tools/

Summary

In this lesson, you have learnt

- Testing is an extremely creative & intellectually challenging task
- Manual testing is performed by a human sitting in front of a computer carefully executing the test steps
- Automation Testing means using an automation tool to execute your test case suite
- Goal of Automation is to reduce number of test cases to be run manually and not eliminate manual testing all together.

Add the notes here.