



TestOps[®]
CERTIFIED LEARNING PROGRAMME

Test Engineer



SUMMARY

Training is focused on supporting existing Test Analysts to develop the skills to become Test Automation Engineers. Learners will be taught how software quality has evolved to become a critical component in delivering working, reliable and stable software.

Covering all aspects of contemporary software quality, through the Test Automation lens, from fundamentals on to principles and practices, through to techniques and tooling. Theoretical learning complimented with hands on practical sessions, all focused on an existing Test Automation solution.



EXPECTED OUTCOMES

1. A solid understanding of the fundamental principles and practices of Test Automation.
2. Be a able to discuss and describe the key aspects of Test Automation in a real world context.
3. A working knowledge of a working Test Automation Platform and the ability to create, update, execute, debug and report automated tests without supervision.



CONTENTS

1. Testing & Test Automation Fundamentals
2. Test Automation Principles, Practices, and Processes
3. Introduction to Test Automation in Practice
4. Introduction to Java for Test Automation Engineers
5. Working with a Test Automation Platform
6. Certification



LEARNING LEVEL

'Foundation Level' learning with no specific prerequisites, however experienced as a software testing practitioner is highly recommended.



TARGET AUDIENCE

This course is aimed at Test Analysts looking to broaden their skills across Test Automation.



COURSE FORMAT

Trainer led using a combination of lecture based sessions, group discussions, 1:1 tutoring, Q&A learning checkpoints and practical exercises.



CERTIFIED EXAM

A 1 hour closed book exam, consisting of 40 multiple choice questions. Learners will also apply their knowledge through completion of a practical exercise. Certification requires a minimum pass rate of 65% for the closed book exam and successful completion of the exercise objectives.



1 TESTING & TEST AUTOMATION FUNDAMENTALS

1.1 Software Testing through the Ages

- 1.1.1 The past
- 1.1.2 Present
- 1.1.3 The Future

1.2 An Introduction Test Automation

- 1.2.1 Background and Definition
- 1.2.2 Purpose and Objectives

1.3 Aspects of Test Automation, Approaches and Integration

- 1.3.1 Agile
- 1.3.2 Waterfall
- 1.3.3 Continuous Testing
- 1.3.4 Continuous Integration
- 1.3.5 Continuous Delivery
- 1.3.6 DevOps

2 TEST AUTOMATION PRINCIPLES, PRACTICES AND PROCESSES

2.1 User Stories and Acceptance Criteria

- 2.1.1 User Stories
- 2.1.2 Acceptance Criteria

2.2 Test Case Management

- 2.2.1 Test Organization
- 2.2.2 Test Planning and Estimation
- 2.2.3 Test Monitoring and Control
- 2.2.4 Configuration Management
- 2.2.5 Risks and Testing
- 2.2.6 Defect Management

2.3 Test Case Execution

- 2.3.1 Testing Throughout The Delivery Pipeline
- 2.3.2 Unit Testing
- 2.3.3 Component, API & Contract Testing
- 2.3.4 Integration Testing
- 2.3.5 End-to-End Testing
- 2.3.6 Exploratory Testing
- 2.3.7 Production Validation Testing

2.4 Automation and Integration

- 2.4.1 Continuous Integration
- 2.4.2 Continuous Testing
- 2.4.3 Executable Specifications
- 2.4.4 Version Control and Artefact Management
- 2.4.5 Automated Quality Checks
- 2.4.6 Automated Builds
- 2.4.7 Automated Data Management
- 2.4.8 Automated Releases
- 2.4.9 Automated Monitoring
- 2.4.10 Automated Reporting
- 2.4.11 Environment Management

2.5 Reporting and Business Intelligence

- 2.5.1 Reporting
- 2.5.2 Business Intelligence

2.6 Release Management

2.7 Change Management

2.8 Documentation and Knowledge Sharing



3 TEST AUTOMATION IN PRACTICE

- 3.1 Architecture
- 3.2 Platform & Tooling
- 3.3 Test Case Management
- 3.4 Automation and Integration
 - 3.4.1 Continuous Integration
 - 3.4.2 Continuous Testing
 - 3.4.3 Version Control and Artefact Management
 - 3.4.4 Automated Quality Checks
 - 3.4.5 Automated Builds
 - 3.4.6 Automated Data Management
 - 3.4.7 Automated Monitoring
 - 3.4.8 Automated Reporting
 - 3.4.9 Environment Management
- 3.5 Release Management
- 3.6 Documentation and Knowledge Sharing

4 INTRODUCTION TO JAVA FOR TEST AUTOMATION ENGINEERS

- 4.1 An Introduction to Java for Testers
- 4.2 Object Orientation
 - 4.2.1 Introduction
 - 4.2.2 Object Orientation Design Thinking
- 4.3 Java Essentials
 - 4.3.1 Class and Objects
 - 4.3.2 Inheritance
 - 4.3.3 Constructor
 - 4.3.4 Statement
 - 4.3.5 Comments
 - 4.3.6 Fields
 - 4.3.7 Methods
 - 4.3.8 Packages
 - 4.3.9 Naming Conventions
 - 4.3.10 Importing Classes
 - 4.3.11 Static Imports
 - 4.3.12 Data Types
 - 4.3.13 Operators
 - 4.3.14 Strings
- 4.4 Decision Logic
 - 4.4.1 Ternary Operators
 - 4.4.2 If, Else and nested statements
 - 4.4.3 Switch statement
- 4.5 Data Structures
 - 4.5.1 Working with Arrays
 - 4.5.2 Working with List Collections
- 4.6 Properties and Property Files
 - 4.6.1 Properties Basics
 - 4.6.2 Java System Properties
 - 4.6.3 Working with Property files
- 4.7 File
 - 4.7.1 Writing And Reading Files
- 4.8 Java and Exceptions
 - 4.8.1 What is an exception?
 - 4.8.2 Catching one or more Exceptions
 - 4.8.3 Exceptions object
 - 4.8.4 Throwing an Exception
 - 4.8.5 Finally



5 WORKING WITH AN EXISTING TEST AUTOMATION PLATFORM

5.1 Workstation Set-up

5.2 Test Automation Project Set-up

5.2.1 Introduction to IntelliJ

5.2.2 Shortcut Keys, Hints and Tips

5.2.3 5.2.4 Code Completion

5.2.4 Creating a project

5.2.5 Import project from version control

5.2.6 Building a project

5.3 Test Automation Platform Overview

5.3.1 Key Resources

5.3.2 Pages

5.3.3 Page Handlers

5.3.4 Step Definitions

5.3.5 Feature Files

5.3.6 Application Entry Points

5.3.7 Page Declarations

5.3.8 Data Management

5.3.9 Browser Support & Management

5.3.10 Test Tags

5.3.11 Environments

5.4 Practical Test Automation

5.4.1 Browser Tools

5.4.2 Page Source

5.4.3 Locators

5.4.4 Creating a test

5.4.5 Updating a test

5.4.6 Executing a Test

5.4.7 Debugging a Test

5.4.8 Reporting

6 CERTIFICATION

6.1 Theory based examination

6.2 Practical skills based examination