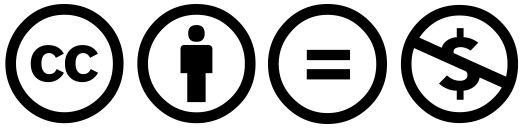
Agile Test Automation Learning Outcomes

ICAgile®
International Consortium for Agile

LICENSING INFORMATION

The work in this document was facilitated by the International Consortium for Agile (ICAgile) and done by the contribution of various Agile Experts and Practitioners. These Learning Outcomes are intended to help the growing Agile community worldwide.



SPECIAL THANKS

ICAgile would like to thank the contributors to the
Agile Test Automation Learning Outcomes:
Janet Gregory • Jeff Payne • Sharon Robson

CONTENTS

HOW TO READ THIS DOCUMENT

This document outlines the Learning Outcomes that must be addressed by accredited training organizations intending to offer ICAgile's Agile Test Automation Certification.

Each LO follows a particular pattern, described below.

0.0.0. Learning Outcome Name

Additional Context, describing why this Learning Outcome is important or what it is intended to impart.

The Learning Outcome purpose, further describing what is expected to be imparted on the learner (e.g. a key point, framework, model, approach, technique, or skill).

LEARNING OUTCOMES

1. TEST AUTOMATION

1.1. TEST AUTOMATION STRATEGY

1.1.1. Automation Pyramid

Automated testing can be performed at various levels within a software application. An automation pyramid or structure describes these various levels and discusses the approach and likelihood of automating tests within each of them.

Explain the various types of testing that can be automated and how decisions get made regarding what to automate during an Agile project.

1.1.2. Planning for Automation

Defining the approach, tools and timings for automation through the project.

Practice planning an Agile test automation effort for each release as part of the test strategy. It also includes how to isolate parts of their system to be able to automate effectively.

1.1.3. Automation Frameworks

Frameworks provide test infrastructure for automating various types and levels of tests.

Identify various types of test automation frameworks so they can effectively choose which frameworks make sense for their particular application based on testing requirements and timelines.

1.1.4. Selecting Tests for Automation

It is typically infeasible because of cost and time to automate all tests that are created and/or run.

Explain how to decide which tests that get created and/or run during an Agile project should be automated vs. tested manually.

1.1.5. Supporting Process

Test automation is performed at various points during Agile project iterations and release cycles. When test automation is performed and for what purpose must be understood.

Explain when it makes sense to automate tests during development iterations and release cycles.

1.2. CONTINUOUS INTEGRATION

1.2.1. Automated Test Cycles (Continuous Testing)

Integrating automated testing into a build environment assures that software changes are tested early and often during the development process.

Discuss tips and techniques for integrating automated tests into an incremental build process such that software is validated during the entire development process.

1.2.2. Code Analysis/Metrics

Code analysis and quality metrics can provide additional insights into an applications quality and release readiness of the software.

Explain code analysis and code metrics for measuring the quality of software applications.

1.3. AUTOMATING STORY AND FEATURE TESTING

1.3.1. Mapping Tests to Automation

When automating tests, the bigger picture is often lost because we concentrate of various levels of testing. We need to take the tests needed for all levels and map them to the overall automation strategy.

Describe how to map feature and story tests to all levels of automation.

1.3.2. ATDD and BDD Testing Frameworks

Frameworks exist that provide support for developing and running automated story and other types of feature tests during software iterations.

Apply at least one commonly used story or feature test framework to further their understanding of how such frameworks are used to support development and testing activities.

1.3.3. UI Testing Frameworks

Tools exist for exercising software through its user interface to test features and combinations of features.

Introduce at least one commonly used UI test tool to further their understanding of how testing can be performed through software's user interface.

1.4. AUTOMATION SUPPORT FOR INTEGRATION AND SYSTEM TESTING

1.4.1. Data Setup and Tear Down

Effective test automation often includes automating the manual processes associated with setting up and resetting test data.

Demonstrate multiple ways to setup and tear-down data (like using a database, using flat files, setting up data within the tests themselves, etc.)."

1.4.2. Data Within Automation

To have stable test automation, you need controlled data.

Discuss different ways of controlling the data so that the automation results are consistent.

1.4.3. Tools to Support Exploratory Testing

While exploratory testing is inherently a manual testing process, tools can be leveraged to assist in the testing process.

Demonstrate tools for recording of results as well as (re)introduce tools to help with exploratory testing such as using Log files to see error messages, automation for data set up.

1.4.4. Tools for Performing Non-Functional Testing

Automation tools exist to support non-functional testing like load testing, performance testing and security testing.

Explain a variety of tools that are available for testing non-functional requirements including load, performance and security.

1.4.5. Virtualization

Virtualization provides a mechanism (often automated) to support effective test environment setup, test execution and test environment tear down during a testing process.

Describe how virtualization can support an automated, effective testing process.