# Fundamentals of Test Automation

## Course Summary

### Description

Test automation is a critical part of most of the modern Agile and DevOps methodologies, and while it is true that test automation done correctly can result in better software being created with less effort and time, test automaton done incorrectly can result in a software project failing in a number of ways. This course is a two day introduction to test automation that introduces students to the concepts, techniques and tools of automated testing as well as how to implement automated testing successfully within a software development or production environment.

The course begins with an introduction in the fundamental concepts of testing that are critical and necessary for effective test automation. This includes an overview of how test automation fits into the Agile testing approach and the DevOps continuous testing processes. The three stage process of automated testing are covered (build, verify and deploy) with emphasis on managing risk factors in the testing process as well as a discussion on the core criteria for successful testing: validity, accuracy, reliability and economy.

The basic sorts of tests that are normally subject to automation: functional testing (i.e. acceptance tests, UI use case testing and unit testing), performance, and security testing are examined in turn with a brief discussion of the various tools that are typically for automating that type of test.

Also covered are the kinds of tests that cannot be automated and an examination of the reasons why they cannot be nor should be.

The second main part of the course focuses on integrating automated testing into the development process, with a discussion of how the tools integrate into the development environment, the build environment within the various development process types. Special attention is paid to how test driven development methodologies integrate with the automated testing in both the Agile and DevOps environments. Test Driven Development and Acceptance Test Driven Development are not themselves testing methods but are development methods that rely on specific sorts of automated tests.

The final part of the course covers two main topics: metrics for the effectiveness of an automated testing implementation and the lessons learned, specifically best practices and common pitfalls.

Specific tools and programming languages can be used in the course at the client request.

### Topics

- Fundamental testing concepts including the importance testing correctness (validity, accuracy, reliability and economy).
- Different kinds of testing: UI, unit and component, usability, performance, security, etc and current state of automation for each type.
- How automated component testing with tools like Cucumber drives application development with Acceptance Test Driven Development.
- How automating unit testing with tools like JUnit, TestNG and others (based on what the client uses) drive code development with Test Driven Development.
- Static testing tools such as code analyzers and static code analysis.

# Fundamentals of Test Automation

## Course Summary (cont'd)

- Automating user interface testing with Selenium and similar tools.
- Performance testing tools: load analyzers, stress testing etc.
- Continuous regression testing with automated testing.
- Integrating continuous testing into Agile development projects using ATDD and TDD.
- Automated testing cycles in DevOps for early detection of errors.
- Qualities for automated tests: componentized, repeatable, deterministic, incremental and meaningful.
- Moving from automated testing to continuous testing.
- Survey of tool chains used in DevOps for automated and continuous testing.
- Retrofitting support for legacy systems with automated testing, including mainframe environments.
- Performance metrics for automated testing.
- Best practices and common pitfalls in automated testing.
- Current 2018 challenges: the problem of applying automated testing to AI, analytics and machine learning types of systems (optional).

### Audience

This course is designed for software developers, team leads, those managing software projects, build teams, QA staff, business analysts, and integration specialists.

### Prerequisites

Before taking this course, students should have a basic knowledge of at least Agile development and preferable other methodologies.  The one day Evolution of Software Delivery (PT20307) would be an adequate prerequisite.  Knowledge of basic software development at both the application and code level will be helpful.

### Duration

Two days