AdvTA

Advanced Test Automation Practices

Advanced Test Automation Practices

AdvTA - Version: 2

🕐 4 days Course

Description:

In the course you will learn how to improve your test automation skills, writing tests that are reliable and easy to maintain. As part of this course you will learn specific hands-on techniques for designing and writing effective automated tests and infrastructure as well as discuss the broader perspective of the role and goals of test automation in the development lifecycle and organization. We'll discuss common pitfalls, tradeoffs, and how test automation can improve the overall performance of the product development team as well as the quality of the developed product.

Intended audience:

Experienced test automation developers.

Prerequisites:

C# or Java (demos will be in C#), with good OOP understanding.
Knowledge of Selenium is also recommended but not required.

Objectives:

Participants will learn a technique for writing reliable and maintainable automated test and infrastructure
Participants will improve their object-oriented design and coding skills
Participants will gain a broad understanding and the role of test automation in the produce development organization, and how it can help the organization work more efficiently and

provide better quality.

Topics:

Test Automation Maturity Levels

- Initial
- QA centric
- Collaboration
- Fusion
- Encompassing
- How to improve

º Manual vs. Automated tests

The ROI of Test Automation

- Getting the most out of Test Automation
- Maintaining a clean product code base
- Continuous Improvement

Architecture and Isolation Techniques

- Determining the inputs and outputs
- Real world examples
- Simulators
- State and Isolation

- º Isolation between environments
- º Isolation between tests
- º Isolation techniques
- Test scopes (e.g., system, unit, integration, etc.)

The Scientific method for writing automated tests

- Recap on MSTest and Selenium (for the demo and lab)
- The scientific method for defining tests
- Writing the tests from top to bottom
- Continuous refactoring to remove duplication

Clean Code and Refactoring Techniques

- Readability
- Modularity and Simplicity
- Poka-Yoke
- Exception Handling
- The SOLID principles
- 4 Rules of Simple Design
- Refactoring techniques and best practices

Test Automation Practices

- Data-Driven Tests
- Failure investigation
- Investigating flaky tests

Test automation and the Organization

- Who should write the tests
- When to write the tests
- When to run the tests
- Business Processes
    - º Handling bugs found by automation
    - º ATDD & BDD
- Reports
- Business structure and Test Automation
- Business Culture and Test Automation