TA4Mgr

Test Automation for decision makers:
Getting the most out of test automation

Test Automation for decision makers:
Getting the most out of test automation

TA4Mgr - Version: 1

🕐 1 days Course

Description:

Many laymen believe that starting to use test automation in a software project requires only to learn the relevant tool, and voila! You have test automation. Unfortunately, building a successful and useful test automation, with a real ROI, is much more complicated. Most people eventually learn about the pitfalls only the hard way…
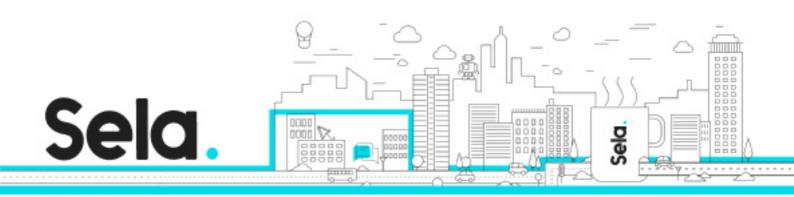
This course, designed primarily for decision makers, will give you an overview on the topic of test automation in order to help you lead a successful test automation project that will improve the entire application development lifecycle and its quality, and save your organization precious time and money!

Intended audience:

This course is targeted primarily at decision makers in software development projects (mainly dev and QA managers), who either haven't started with test automation yet, or have started but feel that it doesn't provide the value they'd expected. In addition it can be useful for testers thinking about starting to use test automation and for anyone who is interested to learn more about test automation.

Prerequisites:

Objectives:

Understand what it takes to build a successful test automation project

Understand the different considerations between manual tests and automated tests

Understand the shortcomings of test automation and common pitfalls

Understand the true value of test automation and how it can improve the entire development lifecycle, the quality of the product, and save money

Know about the most popular tools for test automation and their main features

Topics:

Types of automated tests

- Functional tests
- Unit tests and integration tests
- Performance tests
- Load and stress tests

The different considerations between manual tests and automated tests

- Reliability
- Maintainability
- Diagnosing failures
- Adopting manual tests to automation
- Shortcomings of test automation

Why do we need test automation anyway?

- Enabling agility and rapid releases
- Enable refactoring to improve inner quality and long-term maintainability of the software product
- Continuous Improvement

Organizational patterns for managing the automation project

- Promote manual testers to become automation developers
- Mixing manual testers and few automation developers (the "Building Blocks" pattern/KDT)
- Dedicated automation team
- Automation developers spread across development teams
- A developer centric approach

Test automation and architecture

- The connection between the application's architecture and the tests reliability
- Designing the test automation to fit the application architecture
- Deciding which components of the application to test for best ROI
- Increasing reliability using isolation techniques

Using the test automation for best ROI

- When to run the tests?
    - º Running the tests nightly
    - º Running the tests in Continuous Integration
    - º Running the tests in production
- Handling bugs found by automation (important!)
- When to write the tests?
    - º After development
    - º With development
    - º Before development (ATDD/BDD)

Architecture, business structure and test automation

- Conway's law
- Layered architecture/organization vs a feature based architecture/organization
- Self-organizing teams

Test automation and business culture

- Collaboration vs. blame culture

Popular tools overview

- Tools for UI Automation: Selenium, Coded UI, Ranorex, UFT/LeanFT
- Tools for Rest API: SoapUI, Rest Assured, hand-written
- BDD style tools: SpecFlow/Cucamber, Robot Framework
- ALM Tools: TFS/MTM, Jenkins
- Load testing: VS Load Testing, JMeter