

Advanced Level Technical Test Analyst—ISTQB Software Testing Certification Training

Course Summary

Description

High-quality test designs and specifications are the key to precise test cases and an efficient and effective testing process. This hands-on course provides technical test analysts with the ability to define and carry out the tasks required to fulfill the test strategy in terms of technical requirements. This training course presents a comprehensive overview of methods and techniques for deriving and specifying software tests based on the system's implementation and structure ("white box tests"). On completing the course, attendees will be able to select and apply techniques for test case derivation such as control flow or data flow testing as well as static and dynamic analysis. We will look at non-functional testing techniques such as reliability testing, portability testing, performance, load and stress testing. We will also discuss how to succeed in building robust automation architectures and using a variety of tools to reach quality targets.

Objectives

By the end of this course, students will be able to:

- Recognize and classify the typical risks associated with the performance, security, reliability, portability and maintainability of software systems.
- Create test plans which detail the planning, design and execution of tests for mitigating performance, security, reliability, portability and maintainability risks.
- Select and apply appropriate structural design techniques to ensure that tests provide an adequate level of confidence, based on code coverage and design coverage.
- Effectively participate in technical reviews with developers and software architects applying knowledge of typical mistakes made in code and architecture.
- Recognize risks in code and software architecture and create test plan elements to mitigate those risks through dynamic analysis.
- Propose improvements to the security, maintainability and testability of code by applying static analysis.
- Outline the costs and benefits to be expected from introducing particular types of test automation.
- Select appropriate tools to automate technical testing tasks.
- Understand the technical issues and concepts in applying test automation.

Topics

- The Technical Test Analyst's Tasks in Risk-Based Testing
- Structure-Based Testing
- Analytical Techniques
- Quality Characteristics for Technical Testing
- Reviews
- Test Tools and Automation

Advanced Level Technical Test Analyst—ISTQB Software Testing Certification Training

Course Summary (cont.)

Audience

The target audience for this course includes:

- Software testers
- Senior testers
- Test analysts
- Test leads

Prerequisite

- You must have obtained an ISTQB Foundation Level Certification (CTFL) to be eligible for Advanced Level Certification and have 3 years of experience.
- Prior to attending class please download and review the following document: [Advanced Level Technical Test Analyst Syllabus](#)

Duration

Three Days

Advanced Level Technical Test Analyst—ISTQB Software Testing Certification Training

Course Outline

- I. *The Technical Test Analyst's Tasks in Risk-Based Testing*
 - A. Introduction
 - B. Risk Identification
 - C. Risk Assessment
 - D. Risk Mitigation
- II. *Structure-Based Testing*
 - A. Introduction
 - B. Condition Testing
 - C. Decision Condition Testing
 - D. Modified Condition/Decision Coverage (MC/DC) Testing
 - E. Multiple Condition Testing
 - F. Path Testing
 - G. API Testing
 - H. Selecting a Structure-Based Technique
- III. *Analytical Techniques*
 - A. Introduction
 - B. Static Analysis
 - 1. Control Flow Analysis
 - 2. Data Flow Analysis
 - 3. Using Static Analysis for Improving Maintainability
 - 4. Call Graphs
 - C. Dynamic Analysis
 - 1. Overview
 - 2. Detecting Memory Leaks
 - 3. Detecting Wild Pointers
 - 4. Analysis of Performance
- IV. *Quality Characteristics for Technical Testing*
 - A. Introduction
 - B. General Planning Issues
 - 1. Stakeholder Requirements
 - 2. Required Tool Acquisition and Training
 - 3. Test Environment Requirements
 - 4. Organizational Considerations
 - 5. Data Security Considerations
 - C. Security Testing
 - 1. Introduction
 - 2. Security Test Planning
 - 3. Security Test Specification
- D. Reliability Testing
 - 1. Measuring Software Maturity
 - 2. Tests for Fault Tolerance
 - 3. Recoverability Testing
 - 4. Reliability Test Planning
 - 5. Reliability Test Specification
- E. Performance Testing
 - 1. Introduction
 - 2. Types of Performance Testing
 - 3. Performance Test Planning
 - 4. Performance Test Specification
- F. Resource Utilization
- G. Maintainability Testing
 - 1. Analyzability, Changeability, Stability and Testability
- H. Portability Testing
 - 1. Installability Testing
 - 2. Co-existence/Compatibility Testing
 - 3. Adaptability Testing
 - 4. Replaceability Testing
- V. *Reviews*
 - A. Introduction
 - B. Using Checklists in Reviews
 - 1. Architectural Reviews
 - 2. Code Reviews
- VI. *Test Tools and Automation*
 - A. Integration and Information Interchange Between Tools
 - B. Defining the Test Automation Project
 - 1. Selecting the Automation Approach
 - 2. Modeling Business Processes for Automation
 - C. Specific Test Tools
 - 1. Fault Seeding/Fault Injection Tools
 - 2. Performance Testing Tools
 - 3. Tools for Web-Based Testing
 - 4. Tools to Support Model-Based Testing
 - 5. Component Testing and Build Tools