

Advanced Technical Test Analyst Course Outline

General 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.

By the end of this course, an attendee should be able to:

- Structure the tasks defined in the test strategy in terms of technical requirements
- Analyze the internal structure of the system in sufficient detail to meet the expected quality level
- Evaluate the system in terms of non-functional attributes such as performance, reliability, etc.
- Prepare and execute the appropriate activities, and report on their progress
- Conduct technical testing activities
- Provide the necessary evidence to support evaluations
- Implement the necessary tools and techniques to achieve the defined goals

This course covers the International Software Testing Qualifications Board Advanced Syllabus (v. 2007) from the view of Advanced Technical Test Analyst, and is in the process of accreditation by an ISTQB-recognized National Board.

Learning Objectives

Through presentation, discussion, and hands-on exercises, attendees will learn to:

- Explain the stages in an application's lifecycle where non-functional tests and architecture-based tests may be applied
- Explain the causes of certain non-functional testing taking place only in specific stages of an application's lifecycle
- Explain what criteria may influence the structure and level of test condition development
- Describe how test analysis and design activities are also static testing techniques that can be used to discover defects
- Explain by giving examples the concept of test oracles and how a test oracle can be used in test specifications
- Describe the pre-conditions for test execution, including: testware, test environment, configuration management, and defect management
- Outline the activities of a risk based approach for planning and executing technical testing
- List examples of typical defects to be identified by each specific specification-based technique
- Write test cases from given software model in real-life using the following test design techniques (the tests shall achieve a given model coverage)
 - Equivalence partitioning
 - Boundary value analysis
 - Decision tables
 - State transition testing
- Analyze a system, or its requirement specification in order to determine which specification-based techniques to apply for specific objectives, and outline a test specification based on IEEE 829, focusing on component and non-functional test cases and test procedures
- List examples of typical defects to be identified by each specific specification-based techniques
- Write test cases in real-life using the following test design techniques (The tests shall achieve a given model coverage)
 - Statement testing
 - Decision testing

- Condition testing
- Decision/Condition testing
- Modified condition/decision (MC/DC) testing
- Multiple condition testing
- Loop coverage testing
- Linear code sequence and jump (LCSAJ) testing
- Analyze a system in order to determine which structure-based technique to apply for specific test objectives
- Analyze a system to determine its complexity (cyclomatic complexity testing)
- Understand each structure-based technique and its corresponding coverage criteria and when to use it
- Be able to compare and analyze which structure-based technique to use in different situations
- Describe the principle and reasons for defect-based techniques and differentiate its use from specification- and structure-based techniques
- Explain by examples defect taxonomies and their use
- Understand the principle of and reasons for experienced-based techniques and when to use them
- Specify, execute and report tests using exploratory testing
- Specify tests using the different types of software fault attacks according to the defects they target
- Analyze a system in order to determine which specification-based, defect-based or experienced-based techniques to apply for specific goals
- Use the algorithms “Control flow analysis” and “Data flow analysis” to verify if code has not any control or data flow anomaly
- Interpret the control and data flow results delivered from a tool in order assess if code has any control or data flow anomaly
- Explain the use of call graphs for the evaluation of the quality of architecture. This shall include the defects to be identified, the use for test design and test planning, limitations of results
- Explain how dynamic analysis for code can be executed and summarize the defects that can be identified using that technique, and its limitations

- Characterize non-functional test types for domain testing by typical defects to be targeted (attacked), its typical application within the application life-cycle, and test techniques suited to be used for test design
- Specify test cases for particular types of non-functional test types and covering given test objectives and defects to be targeted
- Characterize non-functional test types for technical testing by typical defects to be targeted (attacked), its typical application within the application life-cycle, and test techniques suited to used for test design
- Understand and explain the stages in an application's lifecycle where security, reliability and efficiency tests may be applied (including their corresponding ISO9126 sub-attributes)
- Distinguish between the types of faults found by security, reliability and efficiency tests, (including their corresponding ISO9126 sub-attributes)
- Characterize testing approaches for security, reliability and efficiency quality attributes and their corresponding ISO9126 sub-attributes
- Specify test cases for security, reliability and efficiency quality attributes and their corresponding ISO9126 sub-attributes
- Understand and explain the reasons for including maintainability, portability and accessibility tests in a testing strategy
- Specify test cases for maintainability and portability types of non-functional test
- Outline a review checklist in order to find typical defects to be found with code and architecture review
- Compare review types with each other and show their relative strengths, weaknesses and fields of use
- Analyze, classify and describe functional and non-functional defects in understandable defect reports
- Compare the elements and aspects within each of the test tool concepts "Benefits & Risks", "Test Tool Strategies", "Tool Integration", "Automation Languages", "Test Oracles", "Tool Deployment", "Open Source Tools", "Tool Development", and "Tool Classification"
- Summarize the test tool categories by objectives, intended use, strengths, risks and examples
- Map the tools of the tool categories to different levels and types of testing
- Create keyword / action word tables using the key-word selection algorithm to be used by a test-execution tool

- Understand why record/playback automation is not a viable solution for testing, but how the tools can be used in different frameworks to solve manual test execution limitations
- Design a performance test using performance test tools including planning and measurements on system characteristics
- Describe by example professional, objective and effective communication in a project from the tester perspective

Course Materials

This course includes the following materials:

<i>Name</i>	<i>Description</i>
Course Outline	A general description of the course along with learning objectives, course materials and an outline of the course topics, including approximate timings for each section.
Noteset	A set of approximately 650 PowerPoint slides covering the topics to be addressed.
ISTQB Foundation Syllabus	The Certified Tester Foundation Level Syllabus which forms the basis for the International Software Testing Qualification at the Foundation Level.
Foundation Sample Exam Questions	A set of approximately 150 pages of review materials for the Foundation level covering every learning objective in the ISTQB Foundation Syllabus.
Foundation Mock Exam	A practice exam containing 40 questions and answers to provide a review of the ISTQB Foundation exam.
ISTQB Advanced Syllabus (2007)	The Certified Tester Advanced Level Syllabus which forms the basis for the International Software Testing Qualification at the Advanced Level.
ISTQB Glossary	The latest glossary of terms used in Software Testing produced by member of the ISTQB.
Standard for Software Testing	Standards used in testing which are referenced by the course materials.

<i>Name</i>	<i>Description</i>
Advanced Technical Test Analyst Sample Exam Questions	A complete set of questions for every learning objective in the Technical Test Analyst module of the ISTQB Advanced Syllabus.
Exercise Solutions	A set of approximately 100 pages of detailed solutions for all exercises in the course.
Advanced Technical Test Analyst Mock Exam	A practice exam containing questions and answers to assess your readiness for the ISTQB Advanced exam.
Project Source Documents for Course Exercises	Specifications used in the realistic example project used in exercises for the course.
Bibliography and resources	A set of further readings, Web sites, tools and other resources to help implement the concepts.

The printed course materials are provided in a binder in a way which makes it convenient for course attendees to remove portions as needed for reference; e.g., during exercises.

Session Plan

The course runs for five days, with three hours set aside on the fifth day for the ISTQB Advanced Test Analyst exam if desired. Each day is about 390 minutes of class time, from 9:00 to 5:30. For accredited course offerings, material is covered as described. For custom courses, material may be deleted, added, or expanded upon as needed.

Please note that timings are approximate, depending on attendee interest and discussion. All of the lectures include exercises and/or knowledge-check questions except as noted.

The following shows this session plan in relationship to the chapters and sections of the ISTQB Advanced Syllabus.

- Introduction and Review (60 minutes, no exercises)**
- 1.0 Basic Aspects of Software Testing (30 minutes, no exercises)**
- 2.0 Testing Processes (180 minutes)**
 - 2.3 Test analysis and design (45 minutes, no exercises)
 - 2.5 Test implementation and execution (45 minutes, no exercises)
 - 2.6 Evaluating exit criteria and reporting (90 minutes, 1 exercise)
- 3.0 Test (and Risk) Management (120 minutes, exercise section 9)**
- 4.0 Test Techniques (930 minutes)**
 - 4.2 Specification-based (310 minutes, 4 exercises)
 - 4.3 Structure-based (310 minutes, 4 exercises)
 - 4.4. Defect and experience based (210 minutes, 3 exercises)
 - 4.5 Static analysis (85 minutes, 1 exercise)
 - 4.6 Dynamic analysis (15 minutes, no exercises)
- 5.0 Test of Software Characteristics (240 minutes)**
 - 5.2 Quality Attributes for Domain Testing (90 minutes, 1 exercise)
 - 5.3 Quality Attributes for Technical Testing (150 minutes, 2 exercises)
- 6.0 Reviews (180 minutes, 2 exercise section 5)**
- 7.0 Incident Management (120 minutes, 1 exercise section 4)**
- 8.0 Standards and Test Improvement Process (15 minutes, no exercises)**

[Note: This is an overview only. For a course that focuses on test process improvement, see Advanced Test Manager.]
- 9.0 Test Tools and Automation (210 minutes)**
 - 9.2 Test tool concepts (40 minutes, no exercises)
 - 9.3 Test tool categories (50 minutes, no exercises)
 - 9.4 Keyword-driven test automation (80 minutes, 2 exercises)
 - 9.5 Performance testing tools (40 minutes, 1 exercise)
- 10 People Skills and Team Composition (30 minutes, no exercises)**

[Note: This is an overview only. For a course that focuses on test process improvement, see Advanced Test Manager.]

Recommended Readings

The class materials include a bibliography of books related to software testing, project management, quality, and other topics of interest to the test professional.