

MODULE DESCRIPTION

ISTQB

Basics of software testing

- > You learn what is understood by software testing and why testing is necessary.
- > The principles of software testing are explained and the five main activities of the fundamental testing process are introduced to the participants.

Testing in the software life cycle

- > The relationships between development and testing activities are explained.
- > You get to know the different test levels and test types and learn about special features that are important when testing in-use software as well.

Static test

- > You learn about static inspection techniques and get to know the review process in detail.
- > This section is rounded off with information on tool-based static analysis.

Test case design procedure

- > Both specification-oriented (black box) and structure-oriented (white box) procedures are discussed in detail and enriched with suitable procedures for the optimal derivation of test cases.
- > Likewise, you get to know reasons why test cases are also derived on the basis of intuition, experience and knowledge.

Test management

- > In software testing projects, the management of the testing process, including all parties involved, is a comprehensive project. Setting up a test organization, planning and estimating activities and monitoring progress are just a few examples of the necessary activities that are explained in this section.

Testing tools

- > There are also suitable, i.e. supporting tools for the different roles and activities in software testing; using these can be very useful for the project, but also carries risks. The introduction of tools into a company must therefore be well considered. Information can be found in this concluding chapter.

JIRA

This module provides initial practical experience from everyday software testing. A test project is being run "live" - including test case determination, testing and error recording or reporting. Since JIRA is one of the current tools for this, the above steps are carried out using JIRA.

TA Basics

TA Basics teaches the basics of test automation. The module starts with the question of what test automation is and explains its benefits. Furthermore, the concept of a framework on the basis of the ANECON A2A framework for test automation is explained. Examples from practice are explained and small examples are worked out.

Compact test cases

The module Compact Test Cases explains concepts such as Data Driven Testing or Acceptance Test Approach, the way from manual to automated test cases, as well as the retraceability of test cases to requirements and conditions. Furthermore, the meaning of refactoring is emphasized and the question of where test automation makes sense at all is addressed. In this module, too, the participants themselves work out examples.

Workload and performance tests

How does a system behave if 10 users access it simultaneously? How at 100? At 1000? This module explains and demonstrates non-functional testing of the system under stress conditions. It shows how the data is monitored and recorded during the stress test, how the execution of tests of distributed agents works, which influencing factors are to be considered and which recommendations can be derived from the results.

Selenium

The basics of the testing framework Selenium are explained here. The basic question "How does web application testing work?" is taken as a starting point and built upon systematically. In examples, the participants can write their own web tests in Java and learn how the testing framework can interact with common browsers, how websites can be opened automatically and how the test framework can use the DOM structure to automatically operate elements on the website.

- | | |
|--|---|
| <ul style="list-style-type: none"> - Tosca (taught by Tricentis trainers) - Automation Specialist 1 - Automation of an HTML application - Identification of HTML controls - Create test cases - Verify results - Basics of software development and programming - Software Development Process - Theory - Specification sheet, functional specifications - Requirements engineering - Project management techniques - Process models: process-oriented, iterative processes, agile models (Scrum) - Practical exercises - Requirement Engineering: Defining functional and non-functional functions of a program. Definition of the scope of a first prototype. Creation of a specification sheet. - Programming: Implementation of the requirements from the practical exercise on requirement engineering. Free choice of programming language. | <p>Automation Specialist 2</p> <ul style="list-style-type: none"> > Create templates for test cases > Use of data sets > API test cases - without interface (web services and XML) |
|--|---|