CONCEPT

Software Testing and Quality Assurance: Test Design, Automation and CI/CD

12 – 23 April 2027
London - Central London four-star



Software Testing and Quality Assurance: Test Design, Automation and CI/CD

Ref: 598_17823 **Date:** 12 – 23 April 2027 **Location:** London - Central London four-star **Fees:** 45540 SAR

Introduction:

Software testing and quality assurance often break down when teams rely on late manual checks, unclear test design and automation that fails without explanation. Defects then escape to users, releases slip and nobody can say how much risk remains. This Core Concept course builds the practical discipline behind reliable delivery: structured test design techniques, risk-based planning, defect reporting, unit and API tests, UI automation, test-driven and behaviour-driven development, and testing inside continuous integration and delivery pipelines. Participants finish with a Test Strategy and Automated Test Suite Design for a case application.

Course Objectives:

- Derive test cases from requirements using equivalence partitioning, boundary value analysis, decision tables, state transitions and exploratory charters
- Plan testing around product risk with clear entry and exit criteria, and report defects that developers can reproduce and fix
- Write unit tests with test doubles and apply test-driven and behaviour-driven development to new features
- Design API, contract and UI automation tests that stay stable and maintainable as the application changes
- Place automated tests, quality gates, performance checks and user acceptance activities within a delivery pipeline
- Produce a Test Strategy and Automated Test Suite Design that a delivery team can adopt

Target Audience:

- Software testers who design and execute functional and regression tests
- Quality assurance engineers responsible for test automation frameworks and suites
- Developers who write unit, integration and API tests for their own code
- Delivery and team leads who decide release readiness and quality gates
- Business analysts and product owners who define acceptance criteria and run user acceptance testing

Course Outline:

Day 1: Testing Principles and the Test Lifecycle

- Seven Testing Principles and the Cost of Late Defect Discovery
- Test Process Activities: Planning, Analysis, Design, Implementation, Execution and Completion
- Test Levels: Unit, Integration, System and Acceptance Scope Definition
- Functional, Non-Functional, Confirmation and Regression Test Types
- Testware Inventory: Test Basis, Test Conditions, Test Cases and Traceability Matrix

Day 2: Static Testing and Black-Box Test Design Techniques

- Static Analysis Checks: Coding Standards, Complexity and Unreachable Paths
- Review Types: Informal Review, Walkthrough, Technical Review and Inspection Roles
- Equivalence Partitioning for Valid and Invalid Input Classes
- Boundary Value Analysis: Two-Value and Three-Value Approaches
- Decision Table Testing for Business Rule Combinations

Day 3: State-Based, Experience-Based and White-Box Techniques

- State Transition Diagrams and Valid and Invalid Transition Coverage
- Exploratory Testing Charters and Session-Based Test Management
- Error Guessing and Checklist-Based Testing from Defect Taxonomies
- Statement and Branch Coverage Measurement for White-Box Testing
- Acceptance Criteria Workshops and Collaboration-Based Test Design

Day 4: Test Planning, Risk-Based Testing and Defect Management

- Test Plan Content: Scope, Approach, Entry and Exit Criteria and Schedule
- Product Risk Register: Likelihood, Impact and Test Priority Mapping
- Defect Report Structure: Reproduction Steps, Severity, Priority and Evidence
- Defect Lifecycle Workflow and Triage Meeting Decisions
- Test Progress Metrics: Execution Status, Defect Density and Requirements Coverage

Day 5: Week-One Case Study: Test Design for a Booking Application

- Case Brief: Online Booking Application Requirements and User Stories
- Risk Analysis and Test Condition Derivation for the Case Application
- Test Case Suite Built with Partitions, Boundaries, Decision Tables and States
- Peer Inspection of the Test Suite Against a Review Checklist
- Test Summary Report: Results, Residual Risk and Release Recommendation

Day 6: Agile Testing, Unit Tests and Test-Driven Development

- Agile Testing Quadrants and the Whole-Team Quality Approach
- Test Pyramid Design: Unit, Service and End-to-End Test Proportions
- Unit Test Structure with Arrange-Act-Assert and Mocks, Stubs and Fakes
- Red-Green-Refactor Cycle in a Test-Driven Development Lab
- Behaviour-Driven Development: Given-When-Then Scenarios and Living Documentation

Day 7: API Testing, UI Automation and Test Data Quality

- API Test Design: Request Validation, Status Codes, Schemas and Negative Cases
- Consumer-Driven Contract Tests Between Services
- UI Automation Principles: Page Object Pattern, Stable Locators and Explicit Waits
- Flaky Test Diagnosis and Suite Maintainability Rules
- Test Data Management: Synthetic Data, Masking and Environment Configuration

Day 8: Testing in CI/CD Pipelines and User Acceptance

- Deployment Pipeline Stages: Commit, Build, Automated Test and Release Gates
- Test Ordering by Speed, Parallel Execution and Fast Feedback Loops
- Quality Gates: Coverage Thresholds, Static Analysis Results and Pass Criteria
- User Acceptance Testing Planning with Business Users and Product Owners
- Acceptance Sign-Off, Defect Communication and Release Readiness Reviews

Day 9: Performance Testing, Debugging and Quality Measurement

- Performance Test Types: Load, Stress, Soak and Spike Profiles
- Workload Modelling, Response Time Percentiles and Throughput Analysis
- Debugging Techniques: Log Analysis, Bisecting Changes and Failure Reproduction
- Root Cause Analysis of Escaped Defects with Five Whys and Fishbone Diagrams
- Quality Metrics Dashboard: Defect Escape Rate, Time to Detect and Automation Coverage

Day 10: Capstone: Test Strategy and Automated Test Suite Design

- Case Application Architecture Review and Quality Risk Profile
- Test Strategy Drafting: Levels, Types, Techniques and Responsibilities
- Automated Suite Layering, Framework Categories and Pipeline Placement
- Test Data, Environment and Maintenance Plan for the Suite
- Test Strategy and Suite Design Presentation and Peer Challenge

Skills You Will Gain:

- Black-Box Test Case Design
- Risk-Based Test Prioritisation
- Defect Triage and Reporting
- Unit Testing with Test Doubles
- API and Contract Testing
- UI Automation Framework Design
- Continuous Integration Test Orchestration
- Performance Test Analysis

Why Attend This Course:

- Return with a Test Strategy and Automated Test Suite Design built for a realistic case application
- Replace ad hoc checking with test cases derived systematically from requirements and business rules
- Turn fragile, slow automation into a layered suite that gives fast feedback in every build
- Exchange testing practices with testers, developers and leads from other organisations and sectors

Conclusion:

Reliable software depends on tests that are designed deliberately, automated at the right layer and run on every change. The first week covers testing principles, the test lifecycle, static testing, test design techniques, risk-based planning, defect management and metrics, closing with a guided case study. The second week adds agile testing, unit tests, TDD, BDD, API and UI automation, test data, pipeline quality gates, user acceptance, performance testing and debugging. The final day produces a Test Strategy and Automated Test Suite Design for a case application.