ROITRAINING
MAXIMIZE YOUR TRAINING INVESTMENT
1-800-ROI-9877
WWW.TRAININGBYROI.COM

Course 306:
Effective Software Testing Methods:

Testing Against Time: Effective and Efficient Defect Detection in Software

(4 days)

Course Description…

When defects are discovered in a system in production by the user or customer, the blame is often placed on the testers. Testing has always been considered the last line of defense against defective software. New approaches to testing apply validation and verification methods throughout the entire Software Development Life Cycle (SDLC).

This course presents current industry best practices in application testing, inspection, verification, validation, and test automation. We will also discuss defect tracking and removal that reduce the risk of delivering software with defects. Using workshops that involve actual case studies from your organization, participants learn the current methods and strategies that can be applied to assure high quality in delivered software.

Learning objectives…

- Apply the concepts of decision tables, test cases and scenarios to all phases of testing.
- Manage objectives, costs and benefits to improve testing and inspection.
- Write test plans, cases, logs and reports.
- Determine the most effective testing and inspection strategy for projects of varying complexity and size.
- Explore a variety of testing methods that may be applied to improve defect discovery rates
- Implement advanced unit- and integration-testing techniques.
- Execute and manage system, acceptance and regression testing.

Who should attend…

Participants attending this course include test professionals, test managers, project leaders, quality analysts, and developers.

Prerequisites…

No specific prerequisites are assumed. A familiarity with software development concepts is recommended.

See next page for a detailed course outline…

Course Outline...

Unit 1: Software Testing

Unit Objectives

Testing Concepts

- Why do we test?
- The objectives of testing
- Testing approaches
- The goal of testing

Role of Quality in the Life Cycle

- Testing in the System Development Life Cycle (SDLC)
- What is quality
- Why don't we get good quality in our delivered systems?
- Solving the Problem Statement

Testing Process

- Inspections
- Test stages

Testing Planning and Automation

- Automated testing tools

Unit 2: Test Planning

Test Planning Issues

- Five dimensions of test planning
- What constitutes test planning
- Test estimation factors

Test Program Direction

- QA Policy
- Test Policy
- Overview of testing policy
- What is planned

Testing and Risk

- Risk reduction through testing
- Measuring success of testing

Unit Summary and Best Practices

Unit 3: Creating Test Cases

Unit Objectives

What Is a Test Case?

- Definition
- Test case format
- Executing a test case

How Do We Create a Test Case/Scenario?

- Requirements
- State diagrams
- Use Cases
- Scenarios

Workshop: Writing a test plan from requirements

Equivalence Classes

- Equivalence class principles
- Boundary analysis
- Equivalence class best practices

Organizing Test Cases

- Test case organization
- Decision Tables

Reporting on Tests
- Audiences
- Defect Report

Unit Summary and Test Case Best Practices

Unit 4: Test Methods

Unit Objectives

General Approach to Testing

Classic Testing Methods
- Script-Driven testing
- Cause-effect testing

Alternate Test Execution Approaches
- Exploratory testing

Testing in Agile Development
- Test-driven development
- Extreme Programming (XP)

Unit Summary

Unit 5: Inspections

Unit Objectives

Benefits of Inspection
- What is an inspection?
- Benefits of early inspection
- Inspection costs

Inspection Process
- Process Overview
- Choosing reviewers
- Planning the inspection
- Inspection steps

Inspection Criteria
- Software requirements checklist
- Design checklist
- Coding checklist
- Test case checklist
- Test procedures checklist
- Test plan checklist

Workshop: Inspecting our work

Unit 6: Unit Testing

Unit Objectives

Unit Testing Planning and Preparation
- Goals of unit testing
- Pragmatics of unit testing
- Approach to unit testing
- The unit test plan
- How unit testing is done
- The test harness

White Box Testing
- White box testing methods
- What is done in white box testing
- Complexity and measurement

Black Box Testing
- When to do black box testing
- Risk-based testing
- Unit testing in object-oriented development

Automated Unit Testing
- Path testing
- Object-oriented testing tools

Unit Summary

Unit Testing Best Practices

Unit 7: Integration Testing

Unit Objectives

Integration Test Planning and Preparation
- Goals of integration
- Definition of integration
- Integration test plan
- Planning the Integration Test process

The Order of Integration
- General principles
- Alternate orders of integration
- What else is integrated?

Builds
- Smoke test
- Types of builds

The Testbed Database
- Creating a testbed database
- Maintaining the testbed database
- Tips for an efficient applicable testbed

Automating Integration Tests
- Automated builds
- Daily build and smoke test

Workshop: Creating integration test procedures

Unit 8: System Testing

Unit Objectives

System Testing Planning and Preparation
- Goals of system testing
- Repeatable system tests
- System test plans

System Testing Scenarios
- Scripted system test scenario design
- Example system test scenario

Risk-Based Testing
- Test scheduling
- Risk-driven strategy for system test phase
- Test selection criteria

Non-Functional Testing
- System test taxonomy
- Test scenario types
- External interface testing
- Performance testing
- Usability testing

Automating System Tests
- Performance measurements
- Automated scenario testing
- GUI testing tools

Alpha and Beta Testing
- Alpha testing
- Beta testing

Workshop: System test scenarios

Unit 9: Acceptance Test

Acceptance Test Planning and Preparation

- Goals of acceptance testing
- Scope of acceptance testing
- Acceptance test plans
- Structured, unscripted acceptance tests
- Scripted acceptance tests
- Statement of acceptance

Executing Acceptance Tests

- Acceptance test issues
- Release

Alternate Methods

- Parallel running
- Going live: uncontrolled, unscripted

Automating Acceptance Tests

Workshop: Creating the acceptance test scenario

Unit Summary and Acceptance Test Best Practices

Unit 10: Maintenance and Regression Testing

Unit Objectives

Maintenance Test Concepts

- Types of software maintenance
- Determining correct behavior
- The regression test suite

Testing in Corrective Maintenance

- Corrective maintenance
- Release cycles

Testing in Perfective Maintenance

1) Perfective maintenance test strategy

Testing in Adaptive Maintenance

- Adaptive maintenance
- Approach to adaptive maintenance testing

Regression Test Suites

- Identifying regression test suites
- Test reusability

Automated Regression Testing

- GUI regression tools

Workshop: Impact on system requirements

Unit 11: Test Management

Unit Objectives

Test Management Issues

- Software testing model
- The test environment

Test Plans and Documentation

- Sources of test plans
- Typical test documentation

Test Automation

- Real cost of test automation
- Automated test guidelines
- Automated test suites

Unit 12: The Bottom Line

Where to go for more information

Practical ideas in software testing