

Acceptance Testing - Simply Put!

Identify, Manage, and Use Business-based Tests for Acceptance Test-Driven Development (ATDD)

Instructor-Led, Onsite or Online Duration 2 days

Overview

Acceptance Test-Driven Development (ATDD) and Behavior Driven Design (BDD) are two of the latest improvements in the Information Technology's (IT's) quest to deliver working software quickly, cheaply, and reliably. Test scenarios are business-based situations taken from real life that guide developers in the software creation process. You need to identify and define business tests that will uncover potential errors and increase your confidence in the delivered application's ability to survive the real world of production.

The key to a successful suite of test cases is to use a wide variety of methods to discover and consolidate testing scenarios that the delivered application must pass to be accepted by the business community. To err is human; to test is to find and eliminate the errors before they proliferate in production.

User acceptance testing without a test plan is similar to taking off without a flight plan. If you do not know what needs to be done to responsibly validate that an information technology solution meets the defined requirements, you place the project (and potentially your organization) at risk.

This business software testing course teaches you how to develop a test plan, organize test scenarios, and manage the testing effort for end-user acceptance testing based on requirements, user stories, and use cases. It also defines and uses test-driven design techniques to identify scenarios to be tested, document the setup criteria, script the execution steps, and establish how to quickly evaluate whether the solution passed or failed the test.



Acceptance Testing - Simply Put!

Target

Anyone involved in anyone responsible for defining or executing end-user acceptance tests for IT applications, including (but not limited to):

Audience

Business Analysts	Subject Matter Experts
Agile Product Owners	Project Leaders and Managers
Agile team members	Systems Analysts
Software Testers	Solution Architects
AND anyone wearing the business analysis hat	

Learning Objectives

Upon completion of this training, successful attendees will:

- Defend the need for test-driven development approaches
- Define the purpose of test scenarios, test scripts, test cases, and test plans
- Identify components that need to be tested for acceptance of an IT application
- Determine whether and when to use walkthroughs, reviews, and inspections
- Create an acceptance test plan that is aligned with your business needs
- Recognize the 17 key elements of effective test plans
- Extract test scenarios from business and stakeholder requirements
- Evaluate business events as a source of test scenarios
- Develop test scenarios based on use case paths and descriptions
- Capture the criteria for testing a well-formed user story
- Assess analysis and design models to identify test scenarios
- Create audience-focused test scripts to maximize repeatability
- Optimize tests using equivalence classes, boundary values, and probable errors
- Choose the minimal set of test cases to achieve defined testing goals
- Optimize the use of limited acceptance testing resources
- Manage your test execution process effectively and efficiently
- Record, track, and evaluate errors discovered in acceptance testing
- List the 7 major steps of a well-defined testing methodology
- Plan to incorporate selected techniques to improve your performance on the job

Acceptance Testing - Simply Put!

1 The Truth about Testing

Testing Concepts Defined

- Test-Driven Concepts
- Common Test Documentation
- A Simple Thought Experiment
- A Tale of Two Testing Objectives

Testing Realities

- To Test or Not To Test
- Exercise: Components of IT Solutions
- Worksheet: Components of IT Systems
- What Are You Testing?
- Discussion: Testable Components
- Worksheet: Testable Components
- Testing Reality

2 The Art and Science of Test Planning

Static Testing Methods

- Of People and Methods
- Static versus Dynamic Testing
- Static Testing Methods
- Exercise: Static Coverage
- Rules of Walkthroughs and Inspections
- Roles in Walkthroughs and Inspections
- Static Validation Methods
- Implementing Static Testing
- Exercise: Making It Happen

Dynamic Testing Methods

- Dynamic Testing Approaches
- Testing Phases
- Exercise: Testing Test Phases
- White Box Testing Defined
- Black Box Testing Defined

Test Planning Techniques

- Test Planning Activities
- Criteria for a Testing Objective
- Exercise: Writing Good Test Objectives
- Administrative Components of a Test Plan
- Executive Components of a Test Plan
- Phase Transition Guard Conditions
- Exercise: Phase Transition Guard Conditions
- Scheduling Components of a Test Plan
- Exercise: Defining the Scope of Acceptance Testing
- Roles and Responsibilities
- Planning Required Resources
- Test Plan Validation

3 Identifying Test Scenarios for Test-Driven Development

Identifying Test Scenarios

- Default Test Scenario Identification Methodology
- Exception-Based Test Scenarios
- Given-When-Then Formatted Test
- Expressing Test Scenarios
- From Test Scenarios to Test Cases

From Requirement Statements to Test Scenarios

- Requirement Types a la BABOK®

- Requirement Relationships
- Rules for Effective Requirement Statements
- Exercise: Requirement-based Test Scenarios
- Worksheet: Requirement-based Test Scenarios

From Business Events to Test Scenarios

- Business Events Revisited
- Finding Test Scenarios from Business Events
- Example of Test Scenarios from Business Events
- Exercise: Business Event Test Scenarios
- Worksheet: Business Event Test Scenarios

From Use Cases to Test Scenarios

- Positive and Negative Testing Use Cases
- Revealing Test Scenarios using a Use Case
- Sample Test Scenarios from a Use Case
- Exercise: Use Case Test Scenarios
- Worksheet: Use Case Test Scenarios

From User Stories to Test Scenarios

- Rules for Effective User Stories
- Common User Story Structures
- Exercise: Waist the Waste User Stories
- Worksheet: For Your User Stories or Questions
- User Story-based Test Scenario Identification
- Exercise: Uncovering Test Scenarios in a User Story
- Worksheet: Uncovering Test Scenarios in a User Story

From Diagrams and Models to Test Scenarios

- Diagrams and Testing
- Visible Tools
- Using a Decision Table to Identify Test Scenarios
- Exercise: Test Scenarios from a Decision Table

4 From Test Scenarios to Test Cases

Creating Test Scripts

- Contents of a Test Script
- Audience-Based Scripting
- Exercise: Test Script Creation

Engineering Test Criteria

- Test Engineering
- Testing Document Structure
- Test Case Documentation
- Engineering Test Data
- Exercise: Engineering Set-up Data
- Execution Data Engineering
- Exercise: Applied Test Data Engineering

Validating Test Cases

- Test Case Completeness Check
- Test Case Validation

5 Test Execution, Evaluation, and Reporting

Executing Acceptance Testing

- Test Execution Activities
- 7 Potential Pitfalls
- Test Setup Checklist
- Exercise: Prime Setup Criteria
- Test Reset Checklist
- Test Environment Components (TEC)

Acceptance Testing - Simply Put!

Capabilities Maturity Model

Testing Maturity Model

Reporting Test Results

Evaluating the Results

Exercise: Test Result Evaluation

Defect Reporting

Test Set Grouping Criteria

Sequencing by Flow of Events

Sequencing by Data Creation and Consumption

Exercise: Test Case Grouping

Test Log Content

Incident Report Components

Problem Resolution Components

Defect Isolation versus Debugging

Software Error Categories [1]

Defect Severity Indicator

Test Summary Report

Software Testing Methodology

Exercise: Other Testing Terminology

A “Real” Testing Methodology

Exercise: Testing Activities

6 From Showtime to Go Time!

Personal Improvement Plan

Understanding the Learning Curve

Exercise: My Techniques

My Personal Implementation Plan