



## Requirements-Based Testing

**Overview:** The Requirements-Based Testing (RBT) course is a three-day process-oriented class that provides students with a set of practical, yet rigorous techniques for testing the requirements to ensure that they are complete, consistent, accurate and unambiguous. Students will also learn how to design a necessary and sufficient set of test cases to validate that the design and code fully implement all of the functional requirements.

For projects using Agile methodologies RBT is the process to use for Acceptance Test Driven Development (ATDD). When applicable, how RBT is applied on Agile projects is interwoven throughout the course.

**Course Description:** Testing, by definition, is comparing an expected result to the observed result. In software the expected results should be defined in the specifications. Unfortunately, most specifications are not written in sufficient detail to predetermine the results of most tests derived from them. In Requirements Based Testing (RBT) the first task is to ensure that the specifications are correct, complete, unambiguous, and logically consistent.

Once the specifications have been clarified, the second challenge is in defining a necessary and sufficient set of tests that are needed to verify that the design and code fully meet the specifications. In most organizations if you gave the same specification to ten different testers you would wind up with ten totally different sets of tests. The thoroughness of these tests would vary widely, dependent on the experience of the tester. Using RBT the test completion criteria is quantified and test status is measurable.

RBT provides a process for first testing the integrity of the specifications. It then provides the algorithms for designing an optimized set of tests sufficient to verify the system from a black box perspective.

This course is student-paced and students are encouraged to bring samples from their own projects. The course includes an overview of a full test automation environment, including the BenderRBT™ software, which automates much of the requirements-based testing process. However, the focus of the course is on process, not on tools.

### Objectives

- Learn reviewing techniques to identify the majority of ambiguities in the requirement specification.

- Translation of the requirements specification into a cause-effect graph.
- Calculation of the elemental functional variations that must be tested.
- Packaging the functional variations into test cases.

**Prerequisites**

None

**Materials Provided**

Student manual containing the course materials and exercises.

**Intended Audience**

Quality assurance engineers, test engineers, managers

**Duration:** Three days

**Class Limit:** 25 students

**Exercises:** Over 60% of class

## **Course Outline**

### Introduction

- Definition of testable requirements
- Definition of testing, the seven basic steps
- The business case for quality
- Overview of the RBT process

### Initial RBT Steps

- Validating requirements against objectives
- Validating the scope of requirements via scenarios

### Finding Ambiguities in Requirements

- The ambiguity review checklist
- Managing the ambiguity process

### Cause-Effect Graphing

- The basic logical operators
- The five graphing constructs that all functional requirements are composed of

### Environmental Data Constraints

- Boundary condition data constraints
- Processing sequence imposed constraints
- Identifying logical inconsistencies in the processing rules

### Test Case Design

- Defining the test completion criteria
- The concept of fault detection
- Basic strategies for test case design
- Identifying functional variations
- Packaging functional variations into test cases

### Test Case Design Alternatives

- Unsensitized variations
- Path Analysis approaches
- Equivalence class testing with boundary analysis
- Orthogonal pairs / Optimized pairs
- Comparing the strengths and weakness of all the test design approaches

### Additional Points of Integration Into The Development Process

- Moving integration testing up before coding starts
- Moving user acceptance testing up before coding starts

### Introduction To Code Based Testing

- White box test completion criteria
- Data flow based testing
- Integrating black box and white box testing

## Tuning The RBT Process by Project Type

- Rapid prototyping
- Agile methodologies
- New development
- Third party packages
- Maintenance
- Technology conversions
- Rewrites and Re-engineering projects

## Management Considerations

- Planning and estimating guidelines
- Change control
- Impact on staffing
- Test team organization
- Tracking the testing effort
- Contract management

## Test Automation Overview

- Overview of BenderRBT function/features
- Overview of an integrated automated test environment
- Synergy between the RBT process and test automation

## Summary

- Case studies of applying RBT
- Defining the path forward

**Bender RBT Inc.**  
**17 Cardinale Lane**  
**Queensbury, NY 12804**  
**518-743-8755**  
**rbender@BenderRBT.com**