

System Testing Course Outline

General Description

The typical tester spends most of his or her working time focusing on issues like test design and execution, test and test result tracking, and bug reporting. This person may be a programmer responsible for unit or component testing, a tester working as part of an independent test team, or a user, customer, or business analyst involving in system or acceptance testing. Based on professional experience and a survey of techniques, Rex Black will review the key skills needed for day-to-day success as a tester. This half-day course may be extended to a full day with hands-on exercises.

Learning Objectives

Through presentation and discussion, and hands-on exercises, attendees will learn to:

- Design and develop white-box and black-box tests, using techniques including:
 - + Requirements and code reviews.
 - + Equivalence classes and boundary value analysis.
 - + Decision tables.
 - + State-transition diagrams.
 - + Orthogonal arrays.
 - + Branch and condition coverage.
 - + Basis paths.
- Execute test cases and track their results.
- Research observed anomalies and write actionable, high quality bug reports.

Course Materials

This course includes the following materials:

<i>Name</i>	<i>Description</i>
Course Outline	A general description of the course along with learning objectives, course materials and an outline of the course topics, including approximate timings for each section.
Noteset	A set of approximately 200 PowerPoint slides covering the topics to be addressed.
Bibliography and resources	A set of further readings, Web sites, tools and other resources to help implement the concepts.

The printed course materials are provided in a binder in a way which makes it convenient for course attendees to remove portions as needed for reference; e.g., during exercises.

Session Plan

Introduction: what is system testing?

Basic effective and efficient black-box testing

- Equivalence partitioning
- Boundary value analysis
- Use cases
- Nouns and verbs
- Customer data and workflows
- Decision tables

Exercise: Equivalence classes and boundary value analysis

Exercise: Internet kiosk decision table tests

Exercise: Decision table, boundary value analysis, and scenarios

Exercise: Building a decision table and tests

Advanced behavioral testing

- State transition diagrams
- Risk-driven tests
- Exploratory tests
- Domain testing
- Orthogonal arrays and all-pairs tables
- Syntax testing

Exercise: Internet kiosk state-transition diagram tests

Exercise: Internet kiosk compatibility test

Effective and efficient white-box (“structural”) testing

- Code coverage

- McCabe complexity, basis paths, basis tests
- Data flow coverage (“set use pairs”)
- Application program interface (API) testing
- Mutation and fault injection
- Other white-box tests
- Drivers and stubs
- Integration techniques
- Integration complexity, basis paths, basis tests
- Other considerations

Exercise: White-box triangle tests

Exercise: Internet kiosk integration tests

Documenting, calibrating, and assessing tests

- What to document
- Test case templates
- Detail and precision of documentation
- Traceability
- Predicting test effectiveness
- Continuous improvement

Exercise: Traceability matrix for Internet kiosk

Exercise: Print server tests in a template

Recommended Readings

The class materials include a bibliography of books related to software testing, project management, quality, and other topics of interest to the test professional.