Code Based Testing Using Data Flow Analysis

Overview

The IEEE standard for unit testing is to execute, at least once, each statement and conditional branch within a software program. However, experience has shown that this technique is not meticulous enough to reliably test business-critical, safety-critical and mission-critical software.
It is especially weak in testing applications involving high levels of asynchronous interaction.

The one day Code-Based testing seminar introduces techniques designed to increase testing coverage by thirty-five to forty percent over that achieved by the IEEE standard. This is accomplished by ensuring that all data flows (i.e., the set/use pairs of data transformations) are tested. This increased test coverage reveals code defects which only show up when tests are executed in a particular sequence or when a test is in a particular position in the test stream.

Data flow analysis also identifies the source of non-reproducible defects (many of which have their roots in spurious data flows). These order-dependent, position-dependent, and non-reproducible defects, are among the most difficult and expensive defects to find and resolve. Practiced in conjunction with the methods taught in the Requirements-Based Testing seminar, Data Flow Analysis ensures the maximum effectiveness in software testing.

Data Flow Analysis can be applied to evaluate the coverage of existing test libraries. It can also be applied to testing asynchronous interactions, testing maintenance changes, and reverse-engineering logical rules from code. Useful methods for scaffolding code-under-test will be taught, allowing all exception handling to be tested....even those functions invoked only by error conditions within hardware and operating system software.

Intended Audience
Programmers, quality assurance engineers, test engineers and managers.

Prerequisites
- Coding experience in any language

Duration: One day

Exercises:	Over 60% of class
Class Limit:	20 students

Course Outline

Code Analysis and Testing

- Introduction
- Objectives

The Code Analysis Process

- Segmenting Code
- Aliasing
- Data Flow Identification
- Interface Definition
- Higher Orders of Data Flow Coverage

Applying Code Analysis

- Analyzing and Supplementing Test Coverage
- Testing Asynchronous Interactions in Code
- Extracting Functional Requirements from Code
- Determining the Impact of Maintenance Changes
- Implementing Code Analysis
- Planning the Code Analysis Effort
- Automating Code Analysis

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