

EFFECTIVE FUNCTIONAL TEST CASE DESIGN

Two-day workshop

Overview

Effective testing depends on the adequacy and quality of test cases - *How do we know that the test cases generated are complete /sufficient? How do we know that the designed test cases are "good"?*

Designing test cases is a challenging activity wherein the business domain knowledge is used extensively. Depending only on domain knowledge however does pose a risk, as all the test engineers in a project may not all be experts. Is there a technique/method that helps in formally designing the test cases?

Testing is a combinatorial game, wherein the test case can be as a combination of data values. Choosing the minimal and effective set is the key to successful testing. Techniques such as equivalence partitioning, boundary value analysis help in pruning the test values, the key issue that they pose though is - how do I scale up, to use in my problem that is large?

Workshop Objective

This workshop discusses a structured approach to test design using the functional and structural properties of a system-under-test; the intent is answer to the important question - "Are my test cases sufficient?" The workshop also addresses the methods to choose the smallest-but-most-effective scenarios/cases, how to design the execution order for efficient testing and suggest a pragmatic method to document the scenarios/cases.

Topics of discussion

Test design fundamentals

- Objective - Conformance vs. Error-detection
- What are the broad methods to designing test cases?
- What types of defects should we look for?
- Limitations of testing, and how static V&V techniques augment testing
- What are "good" test cases?
- Goal oriented approach to test design
- How to effectively combine test data to design minimal test cases
- Error-failure-fault model
- Understanding that a requirement is composed of functional and non-functional components
- How, an erroneous input triggers a latent fault resulting a failure

Defect taxonomy

- What are the various types of defects?
- How to "trigger" off potential defects?
- What are the various "triggers" that help in detecting defects?
- What types of defects may be discovered in the various phases of SDLC

Black-box test design techniques

- Design using Equivalence classes
- Design using Boundary value
- Design Special value test cases
- Design based on past history-Error based
- Catalog based test design
- Output domain analysis
- Decision table based test design

White-box test design techniques

- Control flow coverage based
- Statement, Branch, Path (Modified-Path) coverage
- How much coverage should we strive for?
- How to judge test adequacy using coverage metrics
- Using data flow to design test cases
- Designing test cases for memory leaks, error handling and concurrency issues

Other aspects to test design

- What information should a test case document contain?
- How to document test scenarios/cases
- Aspects to consider for test data for test execution automation

Workshop benefits

At the end of the workshop, the participants will have good theoretical knowledge of test design methods that they can apply in their current work.

Who should attend?

Developers, test engineers and test managers

Prerequisites - Participants are expected to have good knowledge of programming with preferably knowledge of testing concepts

Delivery method

The workshop consists of classroom discussions and a problem solving/case-study session. It adopts our unique learning model. This enables participants listen to lecture on a topic, introspect using a structured questionnaire, explore the topic - hands-on session, discuss the learning after exploration and finally read the supplemental course notes.