Software Testing Mentor

www.softwaretestingmentor.com

ISTQB Foundation Level and Software Testing Training

Module 4
Test Design Techniques

Session 4 – Structure-based or white-box techniques

Structure Based or White box Techniques

In this session we will learn about:

What is code coverage?

White Box Testing Techniques

- Statement Coverage
- Decision Coverage
- Some other structured based testing techniques

What is Test Coverage?

Test coverage is the degree expressed as percentage to which a specified coverage item has been exercised by test suite

Coverage = (Number of coverage items exercised/Total number of coverage items)*100

100% coverage does not mean 100% tested because coverage techniques measure only one dimension of multiple dimensions

Types of Coverage

Coverage can be measured at component testing , integration testing, system testing or acceptance testing levels

At component testing level coverage items may be the statement, decision coverage etc.

At integration level coverage items are the interfaces or interactions with other software

At acceptance testing level coverage items may be requirements, menu items, business transactions etc.

Types of Coverage Cont.

Coverage of each specification based technique can also be measured

- EP: Percentage of Equivalence partitions exercised
- BVA: Percentage of boundaries exercised
- Decision tables: Percentage of business rules exercised
- State transition table: Percentage of states visited, percentage of valid and invalid transitions exercised

How to measure coverage

1. Decide on the coverage item to be counted

2. Count the structural elements or items

3. Instrument the code

4. Run the tests for which coverage measurement is required

5. Using the outputs from the instrumentation, determine the percentage of items exercised

Statement Coverage and Statement Testing

Statement coverage is also known as line coverage. The formula to calculate statement coverage is:

Statement Coverage=(Number of statements exercised/Total number of statements)*100

Studies in the software industry have shown that black-box testing may actually achieve only 60% to 75% statement coverage, this leaves around 25% to 40% of the statements untested.

Statement Coverage Example

To illustrate the principles of code coverage lets take one pseudo-code

```
READ X
READ Y
IF X>Y
PRINT "X is greater than Y"
ENDIF
```

Let us see how can we achieve 100% statement coverage for this pseudo-code, we can have 100% coverage by just one test set in which variable X is always greater than variable Y.

TEST SET 1: X=10, Y=5

Decision Coverage and Decision Testing

Whenever there are two or more possible exits from the statement like an IF statement, a DO-WHILE or a CASE statement it is known as decision because in all these statements there are two outcomes, either TRUE or FALSE.

Decision coverage ensures that each outcome(i.e TRUE and FALSE) of control statement has been executed at least once.

The formula to calculate decision coverage is: **Decision Coverage**=(Number of decision outcomes executed/Total number of decision outcomes)*100%

Research in the industries have shown that even if through functional testing has been done it only achieves 40% to 60% decision coverage.

Decision coverage is stronger that statement coverage and it requires more test cases to achieve 100% decision coverage.

Decision Coverage and Decision Testing Example

Let us take one example to explain decision coverage:

```
READ X
READ Y
IF “X > Y”
PRINT X is greater that Y
ENDIF
```

To get 100% statement coverage only one test case is sufficient for this pseudo-code.
TEST CASE 1: X=10 Y=5

However this test case won’t give you 100% decision coverage as the FALSE condition of the IF statement is not exercised.

In order to achieve 100% decision coverage we need to exercise the FALSE condition of the IF statement which will be covered when X is less than Y.

So the final TEST SET for 100% decision coverage will be:
TEST CASE 1: X=10, Y=5
TEST CASE 2: X=2, Y=10

Decision Coverage and Decision Testing Cont.

Decision Coverage is also known as Branch Coverage

100% decision coverage guarantees 100% statement coverage but 100% statement coverage does not guarantee 100% decision coverage

Decision Coverage can not test 100% of the software code

Decision Coverage is not suitable for complex decision items like

"If A > 10 or (A < 10 and A+B = 20) then"

Other Structured based Techniques

Statement and Decision Coverage are 2 of the most basic coverage methods

Others more complex Structured based techniques are:

- Condition Coverage
  - http://www.softwaretestingmentor.com/test-design-techniques/condition-coverage/
- Multiple Condition Coverage
  - www.softwaretestingmentor.com/test-design-techniques/multiple-condition-coverage/
- Multiple Condition Decision Coverage
  - http://www.softwaretestingmentor.com/test-design-techniques/multiple-condition-decision-coverage
- Linear Code Sequence and Jump Testing (LCSAJ)

Conclusion

In this session we will learned about:

What is code coverage?

White Box Testing Techniques

- Statement Coverage
- Decision Coverage
- Some other structured based testing techniques

THANK YOU!!!