18: More testing and coverage

fill out parts form by EOD today!
Coverage

Notion of how completely a piece of code has been tested with a particular set of tests, with respect to a specific metric.

Examples:

- What % of requirements have been tested?
- What % of lines of code have been tested?

100% coverage does not mean 100% tested, but it’s a start to assess testing thoroughness.
White box testing guided by coverage

**Branch** (aka decision) - for every branch (e.g. if-statement), is there at least one test case that evaluates that branch to true and one that evaluates it to false?

**Condition** - like branch coverage, but looking at conditions within branches (e.g. looking at $x > 0$ and $y == 2$ separately rather than just $x > 0 \lor y == 2$)

**Path** - is there a test case that exercises every unique path through the code (as opposed to considering each branch independently)
### Branch coverage

```java
if (x == 3 && y < 0 ) {
    // do something;
} else {
    // do something else
}

q = x + z;

if (q < y) {
    if (x == z) {
        // do another thing
    }
    // do a fourth thing
}
```

<table>
<thead>
<tr>
<th>(x, y, z)</th>
<th>x==3 &amp;&amp; y &lt; 0</th>
<th>x + z &lt; y</th>
<th>x == z</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0, 0)</td>
<td>false</td>
<td>false</td>
<td>n/a</td>
</tr>
<tr>
<td>(1, 1, 3)</td>
<td>false</td>
<td>false</td>
<td>n/a</td>
</tr>
<tr>
<td>(3, -1, 0)</td>
<td>true</td>
<td>false</td>
<td>n/a</td>
</tr>
<tr>
<td>(3, -1, -5)</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>(3, 5000, 3)</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>
if (x == 3 && y < 0 ) {
    // do something;
} else {
    // do something else
}

q = x + z;

if (q < y) {
    if (x == z) {
        // do another thing
    }
}
Path coverage

```java
if (x == 3 && y < 0 ) {
    // do something;
} else {
    // do something else
}

q = x + z;

if (q < y) {
    if (x == z) {
        // do another thing
    }
}
```
Six paths through the flowchart, but one is impossible according to the data.
Overhead of coverage tools

article by Klaus Lambertz at embedded.com:

Measuring code coverage for embedded software
Modified Condition/Decision Coverage (MC/DC)

A more comprehensive coverage metric required by some software safety standards

- Each entry and exit point is invoked
- Each decision takes every possible outcome <- branch coverage
- Each condition in a decision takes every possible outcome <- condition coverage
- Each condition in a decision is shown to independently affect the outcome of the decision
Each condition in a decision is shown to independently affect the outcome of the decision.

Hold all but one condition constant. Does changing that condition affect the outcome of the decision?

\[(x + y) == 3 \land (y < 0) \lor x == 2\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>(x + y == 3)</th>
<th>(y &lt; 0)</th>
<th>(x == 2)</th>
<th>decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
Rest of the V

- Product requirements
- Software requirements
- High level/architecture design
- Integration testing
- (System-level) Software testing
- Acceptance testing

Flow diagram showing the relationship between product requirements, software requirements, high level/architecture design, integration testing, system-level software testing, and acceptance testing.
Integration testing

Use high level design (architecture diagram and sequence diagrams) to test interfaces between modules/components:

- Test every interface (message format, correctness of values)
- Test timing and sequence of messages sent
- Test that unexpected messages are handled

Assume modules are performing individual duties correctly (why?) and just test the communication between them.
Sequence diagram test example

Scenario: check available funds at ATM

User
- insert card
- request PIN
- enter PIN

ATM
- validate PIN
- PIN valid

Bank server
- get available funds
- send available funds

Integration test sends these messages
- display options
- request available funds
- display available funds

But also keeps track of sequence/timing of these