Why Tests Don’t Pass (or Fail)

PNSQC October 28, 2009

 Definitions

• Test
  – An exercise designed to surface bugs
  – Includes a validation mechanism
• Pass
  – A test that behaves as expected per validation
• Fail
  – A test where validation fails
  – A test with unexpected behavior
• Oracle
  – Principle or mechanism by which we recognize a problem
  – Answers “why” when unexpected things happen
Pass?

- What results are we checking?
- How do we know the expected outcome?
- Are we checking all of the results?
- What outcomes are we not checking?
- How would we know misbehaviors for the outcomes we are not checking?

A Programming Challenge

Given a program and a set of tests for it, modify the code [to do anything] in a way that the tests don’t detect.

This change is an undetected bug.
Test Result Possibilities

<table>
<thead>
<tr>
<th>Situation</th>
<th>No Error in SUT</th>
<th>Error in SUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>As Expected (Pass)</td>
<td>Correct Pass</td>
<td>Silent Miss</td>
</tr>
<tr>
<td>Unexpected (Fail)</td>
<td>False Alarm</td>
<td>Caught it!</td>
</tr>
</tbody>
</table>

Fail?

- The test finds the bug it’s designed for!
- Something unexpected happens while the test is running.
- Possible false alarms.
- Possible unnoticed failures.

Why Tests Don't Pass
Software Test Execution Model

Influencers
- Test Inputs
  - Precondition Data
  - Precondition Program State
  - Environmental Inputs

System Under Test
- Test Results
  - Post-condition Data
  - Post-condition Program State
  - Environmental Results

Outcomes

Questions to Ask Ourselves

- What values/conditions should influence the SUT? How do we set/cause them?
- Are we controlling or monitoring the best influencers? (What influencers are we not controlling or monitoring?)
- Are we checking the most important outcomes? (What outcomes do we know we are not checking?)
- How do we know the expected outcomes?
- What gives us confidence the test isn’t missing bugs?
So What

- Be open minded about (skeptical of) “Pass”
- Understand that a “Fail” could mask errors
- We aren’t checking all of the results
- We don’t know the outcomes from arbitrary errors
- Pass/Fail metrics don’t really give us interesting information

What Can We Do

- Influencers
  - Set values
  - Monitor
- Outcomes
  - Check what’s practical
  - Compare pre- with post-
  - Real time monitoring
  - Sanity checks
Summary

• The SUT doesn’t really pass or fail
  – “Pass” means we noticed only expected behaviors
  – “Fail” means we noticed something not according to plan

• There are lots of other possible reasons for “passing” or “failing” other than a bug in the SUT