Software Testing

CS-370: Software Design and Development

Robert Bruce

What is software testing?

- Software testing is used to find defects in a software project.
- Software testing is also used to verify that the software meets the customer's requirements.

Source: page 329, *Beginning Software Engineering* (2nd edition) by Rod Stephens

Levels of software testing

- **Unit testing**: “verifies the correctness of specific pieces of code”.
- **Integration testing**: verifies the effectiveness of new code in an existing software project.
- **Regression testing**: “testing the program’s entire functionality” looking for cases where the code fails.
- **Automated testing**: A form of testing in which a program conducts a series of tests then compares the results with the expected test results.
- **Component interface testing**: “studies the interactions between components.”
- **System testing**: “an end-to-end run-through of the whole system.”
- **Acceptance testing**: “determines whether the finished application meets the customers’ requirements.”

Source: pages 335 - 341, *Beginning Software Engineering* (2nd edition) by Rod Stephens

Testing techniques

- **Exhaustive testing**: “[You] consider every possible combination of inputs.”
- **Black-box testing**: You have no idea how the software works. Consequently, you provide random inputs and observe the output for failures.
- **White-box testing**: Given your knowledge of how the software works, you devise test cases which will likely cause the software to fail.
- **Gray-box testing**: You have some knowledge of how the software works. Given this limited knowledge, you devise test cases which will likely cause the software to fail.

Source: pages 342- 344, *Beginning Software Engineering* (2nd edition) by Rod Stephens

The Lincoln index

- The Lincoln index estimates the number of errors present in a software application.
- The general formula is: $L = (E_1 * E_2) / S$

Where:

- L = Lincoln index
- E_1 = number of errors found by Software Tester One (individual on the development team).
- E_2 = number of errors found by Software Tester Two (individual on development team).
- S = number of errors found by both testers (i.e. the intersection of E_1 and E_2).

Source: pages 342- 344, *Beginning Software Engineering* (2nd edition) by Rod Stephens

Calculating the Lincoln index

Example:

- Laura found 15 errors (E_1).
- James found 11 errors (E_2).
- Laura and James found 5 errors in common (S).

Analysis:

- Lincoln index = $E_1 * E_2 / S = 15 * 11 / 5 = 33$
- The Lincoln index predicts the software likely contains a total of 33 errors.
- Laura has found 10 unique errors (and five errors in common with James).
- James has found 6 unique errors (and five errors in common with Laura).
- Total number of undetected errors remaining = Lincoln index – (unique errors from Laura) – (unique errors from James) – (number of common errors both Laura and James found) = 33 – 10 – 6 – 5 = 12