Is Unit Testing Worthwhile?

Nima Seyedtalebi

University of Kentucky

November 7, 2018

Background

- The IEEE Software Engineering Body of Knowledge (SWEBOK) provides a concise definition of software testing: "Software testing consists of the *dynamic* verification that a program provides *expected* behaviors on a *finite* set of test cases, suitably *selected* from the usually infinite execution" [**?**]
- Key points:
    - Dynamic: Input and source code are not always enough to determine behavior
    - Expected: We must be able to define expected behavior to test for it
    - Finite: The set of possible test cases is practically infinite, so we must choose a finite subset
    - Selected: Test cases can vary in usefulness considerably, so the choice is important

Different Kinds of Testing

- Testing can be classified by target or objective
- Classifying by target gives three levels:
  - Unit Testing: Small pieces of software testable in isolation
  - Integration Testing: Interactions between software components
  - System Testing: An entire system
- Classifications by objective:
  - Regression testing
  - Acceptance testing
  - Security testing
  - Performance testing
  - Stress testing

What is Unit Testing?

- From the SWEBOK: "Unit testing verifies the functioning in isolation of software elements that are separately testable." [**?**]
    - What constitutes a unit? It depends on context
    - Developers may have differing ideas about what constitutes a unit
- Usually performed by the developer of the unit or someone with programming skills and access to the source code
- Surveys suggest unit testing is an important testing method that sees widespread use
- Unit testing is sometimes conflated with other kinds of testing
    - E.g. a "unit test" that relies on a database connection is not a unit test under the definition given

Challenges

- Exhaustive testing is impractical at best and impossible at worst
- Consider a program that takes a Unicode string as an argument and writes it to STDOUT:
  - The Unicode 11.0 standard contains 137,374 different characters[**?**],so $(137374)^n$ permutations of length n
  - Such a program depends on the behavior of STDOUT. Truly exhaustive testing would have to account for this
- How to define what a good test case is?
- How do we know if our tests are sufficient?
- Tests must be maintained and run often, both cost time

Software Testing Metrics

- Statement coverage. Use example from 1978 paper about why two cases is not enough to fully test even a simple conditional

How is it Done?

- Include empirical data

Tool Support

- Testing frameworks
- JUnit, Mockito, PowerMock in particular
- Continuous Integration

Arguments For

-