Unit Test and JUnit

Hui Chen [a]

[a]CUNY Brooklyn College, Brooklyn, NY, USA

March 15, 2022

Outline

Outline

Types of Tests

- System or Acceptance Test: integrated program meets its specifications
- Integration/system testing: interfaces between units have consistent assumptions, communicate correctly
- Module or Functional Test: within individual units
- Unit testing: single method does what was expected

Outline

Test Automation

Use software

- to control the execution of tests,
- to compare actual outcomes to predicted outcomes, and
- to set up of test preconditions, and other test control and test reporting functions

Benefits of Test Automation

- Reduces testing cost – testing is expensive, and isn't a revenue-creating task
- Reduces human error
- Reduces variance in test quality from different individuals
- Enables regression testing

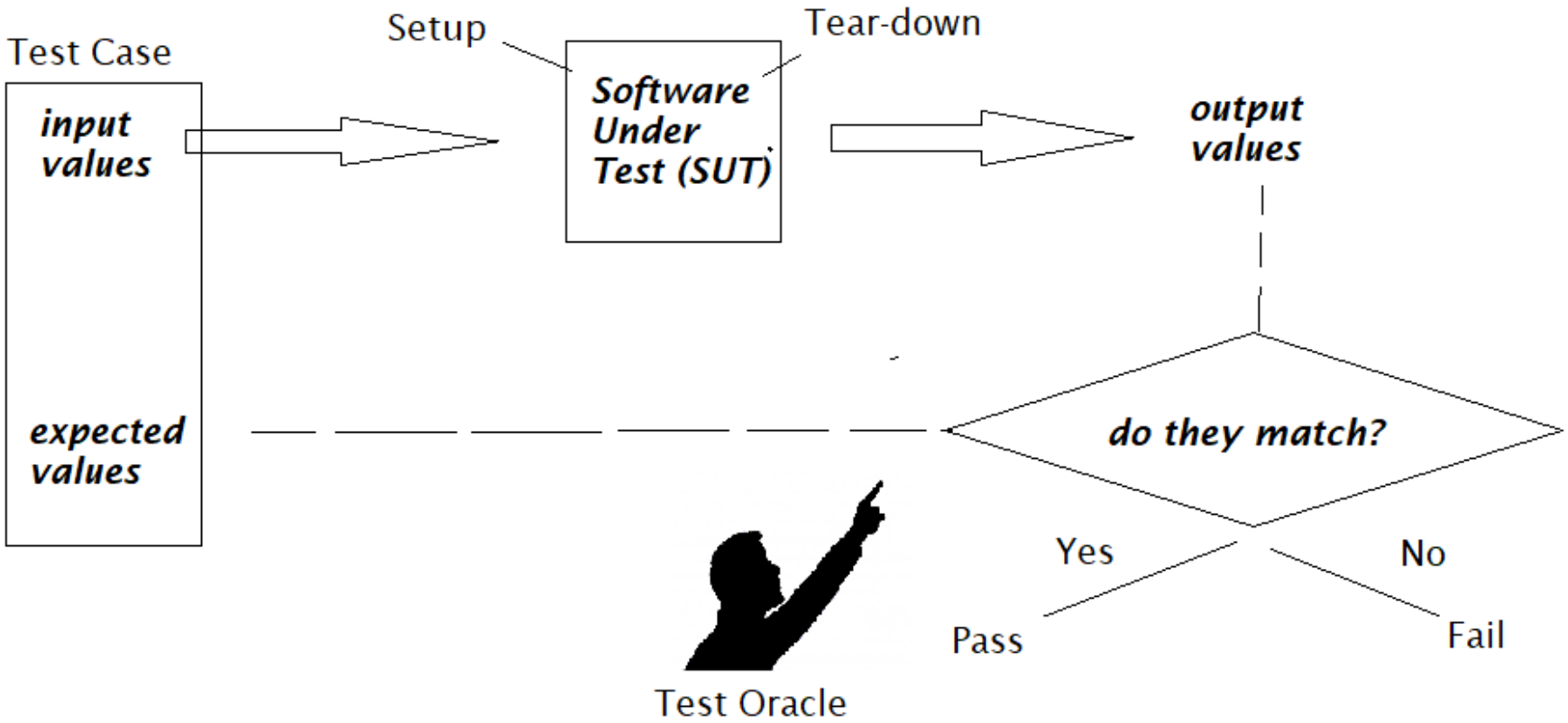
Outline

Software Testability

Degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met

- Testability is dominated by two practical problems
  - How to provide the test values (input and expected result) to the software
  - How to observe the results of test execution

Test Case and Running Test Case



Components of a Test Case

A test case is a multipart artifact with a definite structure

- Test case values
  - input values needed to complete an execution of the software under test
  - output values produced by the software under test (when a test case is executed)
- Expected results: result that will be produced by the test if the software behaves as expected
- A test oracle uses expected results to decide whether a test passed or failed

Putting Tests Together

Test Case

- The test case input and output values, and expected results necessary for a complete execution and evaluation of the software under test

Test Set and test suite

- A set of related test cases

Test Automation Framework

- A set of assumptions, concepts, and tools that support test automation

Outline

What's JUnit?

An open source Java testing framework used to write and run repeatable *automated tests* (https://junit.org whose features include

- Assertions for testing expected results
- Test features for sharing common test data
- Test suites for easily organizing and running tests
- Graphical and textual test runners

JUnit is widely used in industry, and can be used as stand alone Java programs (from the command line) or within an IDE such as Eclipse, Android Studio, and the others

JUnit Tests

- JUnit can be used to test
  - an entire object
  - part of an object – a method or some interacting methods
  - interaction between several objects
- It is primarily intended for unit and integration testing, not system testing
- Each test is embedded into one test method
- A test class contains one or more test methods
- Test classes include :
  - A collection of test methods
  - Methods to set up the state before and update the state after each test and before and after all tests

Write a Simple Test Case

- Each test method checks a condition (assertion) and reports to the test runner whether the test failed or succeeded – comparing expected output and actual output
- All of the test methods must return `void`
- Need to use the methods of the `org.junit.jupiter.api.Assertions` or `org.junit.Assert` class to do assertions
  - API documentation gives a complete description of its capabilities
- The test runner uses the result to report to the user (in command line mode) or update the display (in an IDE)
- A few example methods of org.junit.Assert
  - `assertTrue(boolean)`
  - `assertTrue(String, boolean)`
  - `assertEquals(int, int, String)`
  - `fail(String)`

A demo ...

Let's fire up Android Studio and see how it works ...

Outline

Summary

Unit testing

JUnit

Demo

Outline

References

"Introduction to Software Testing" 2nd Edition. Ammann and Offutt

JUnit 5 User Guide

JUnit 4 Getting Strated Guide