infogain

Understanding Unit Testing

April 22, 2019

Copyright © 2013 Infogain Corporation. All rights reserved.

Objective

Understanding Unit Testing

- ❑ The Role of Testing in Software Development
- ❑ Developing Around Dependencies
- ❑ Organizing and Running Unit Tests
- ❑ Introduction to TDD

Copyright © 2013 Infogain Corporation. All rights reserved.

Unit Test : A Definition

A unit test is a piece of a code (usually a method) that invokes another piece of code and checks the correctness of some assumptions afterward.

If the assumptions turn out to be wrong, the unit test has failed.

A "unit" is a method or function.

Note : Unit testing will be performed against a system under test (SUT).

Copyright © 2013 Infogain Corporation. All rights reserved.

Unit Test : An Introduction

❑ A procedure to validate individual units of source code
Example: A procedure, method or class

❑Validating each individual piece reduces errors when integrating the pieces together later

A unit test is a test related to a single responsibility of a single class, often referred to as the System Under Test (SUT).

❑ A test is an assessment of our knowledge, a proof of concept, or an examination of data.

❑ Unit testing code means validation or performing the sanity check of code.

❑ Sanity check is a basic test to quickly evaluate whether the result of a calculation can possibly be true.

Copyright © 2013 Infogain Corporation. All rights reserved.

Unit Test : Properties of a Good Unit Test

A good unit test *should* have the following properties:

❑ It should be automated and repeatable.

❑ It should be easy to implement.

❑ Once it’s written, it should remain for future use.

❑ Anyone should be able to run it.

❑ It should run at the push of a button.

❑ It should run quickly.

Copyright © 2013 Infogain Corporation. All rights reserved.

Is it really a Unit Test ?

Ask yourself the following questions about the tests you've written:

- *Can I run and get results from a unit test I wrote two weeks or months or years ago?*

  Answer: *If you can't do that, how would you know whether you broke a feature that you created two weeks ago?*

- *Can any member of my team run & get the results from unit tests I wrote 2 months ago?*

  Answer: *We want to make sure that you don't break someone else's code when you change something.*

- *Can I run all the unit tests I've written in no more than a few minutes?*

  Answer: *When you change code, you want to get feedback as early as possible to see if you broke something.*

- *Can I run all the unit tests I've written at the push of a button?*

  Answer: *If you can't fully automate your unit tests, you'll probably avoid running them repeatedly, as will everyone else on your team.*

- *Can I write a basic unit test in no more than a few minutes?*

  Answer: *One of the easiest ways to spot an integration test is that it takes time to prepare correctly and to implement, not just to execute*

How to Write a Unit Test ?

❑ Using a Framework

❑ Without using a Framework

Copyright © 2013 Infogain Corporation. All rights reserved.

Unit Testing Frameworks

Unit-testing frameworks help developers

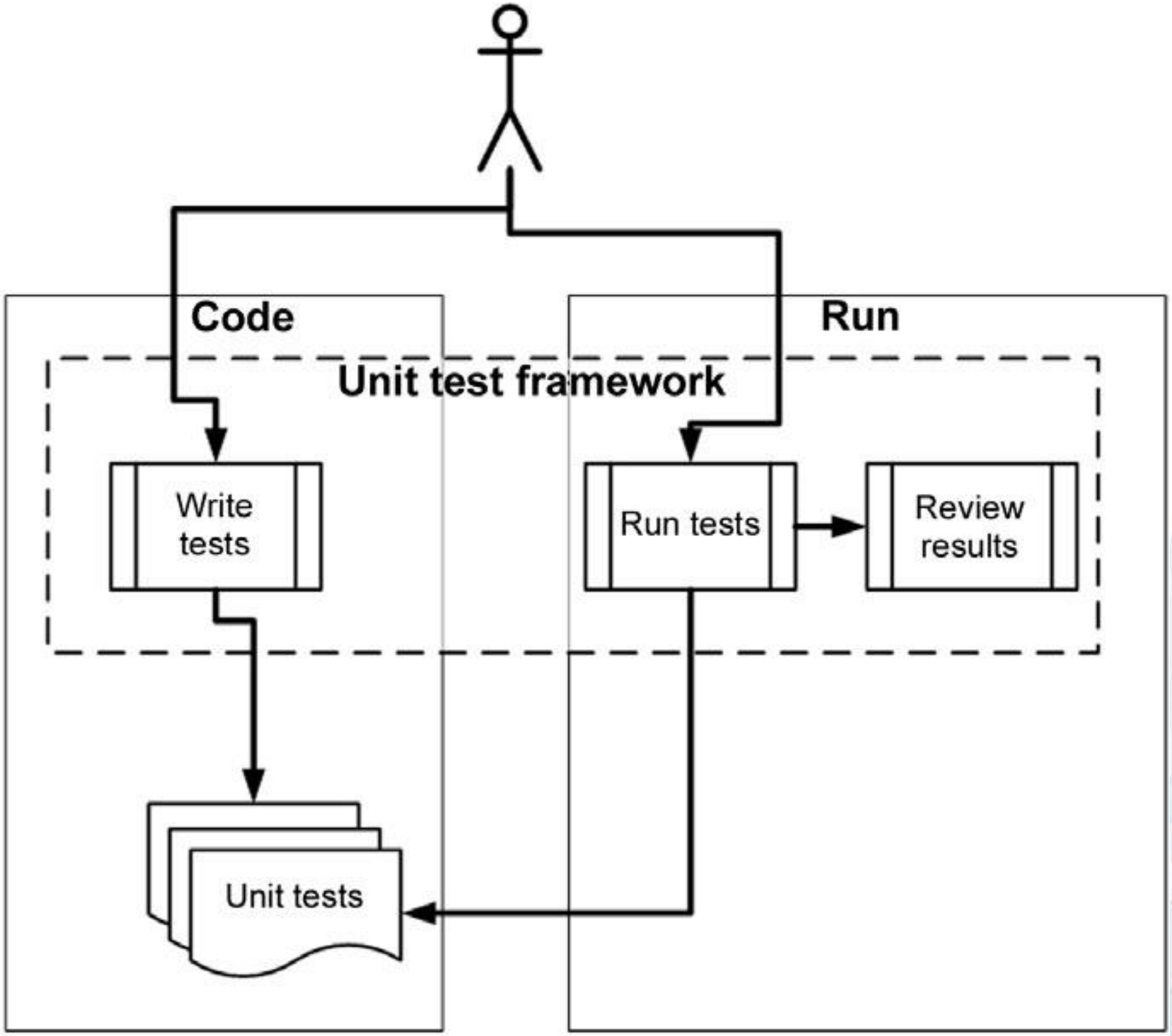
- Write tests more quickly with a set of known APIs,
- Execute those tests automatically,
- Review the results of those tests easily.

Copyright © 2013 Infogain Corporation. All rights reserved.

Unit Testing Framework : Role in Developer’s Life

Method	Description
Write tests easily and in a structured manner.	Framework supplies the class library that holds ❑ Base classes or interfaces to inherit. ❑ Attributes to place in your code to note your tests to run. ❑ Assert classes that have special assert methods you invoke to verify your code.
Execute one or all of the unit tests.	Framework provides a test runner (a console or GUI tool) that ❑ Identifies tests in your code. ❑ Runs tests automatically. ❑ Indicates status while running.
Review the results of the test runs.	The test-runners will usually provide information such as ❑ How many tests ran. ❑ How many tests didn’t run. ❑ How many tests failed. ❑ Which tests failed. ❑ The reason tests failed. ❑ The assert message you wrote. ❑ The code location that failed.

Unit Testing Framework : A Graphical View



Copyright © 2013 Infogain Corporation. All rights reserved.

List of Unit Testing Framework Available for Java

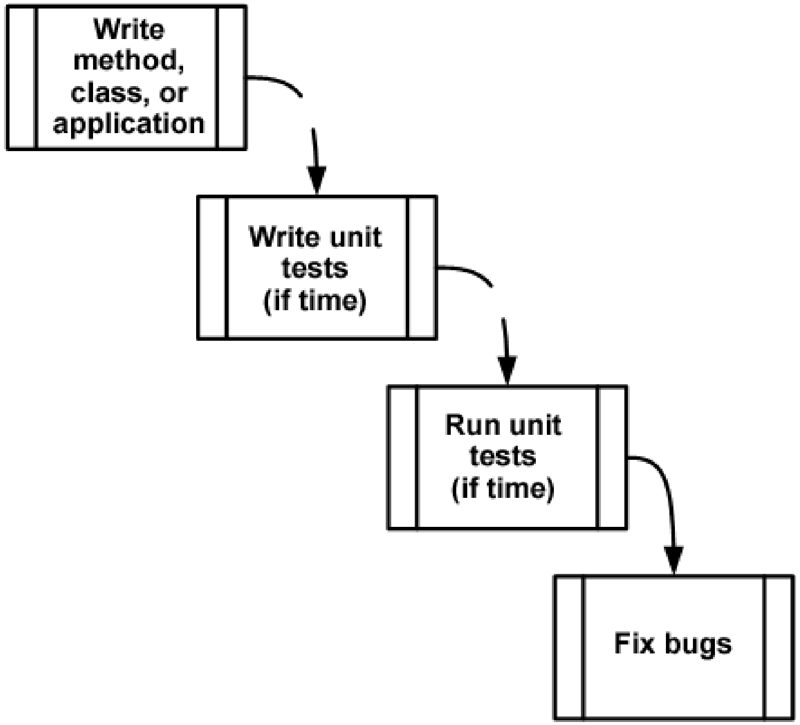
- ❑ SpryTest
- ❑ Jtest
- ❑ JUnit
- ❑ TestNG

Copyright © 2013 Infogain Corporation. All rights reserved.

When to Write Test Case

- ❑ Once we know how to write structured, maintainable, and solid tests with a unit-testing framework, the next question is when to write the tests.

- ❑ Many people feel that the best time to write unit tests for software is after the software has been written.

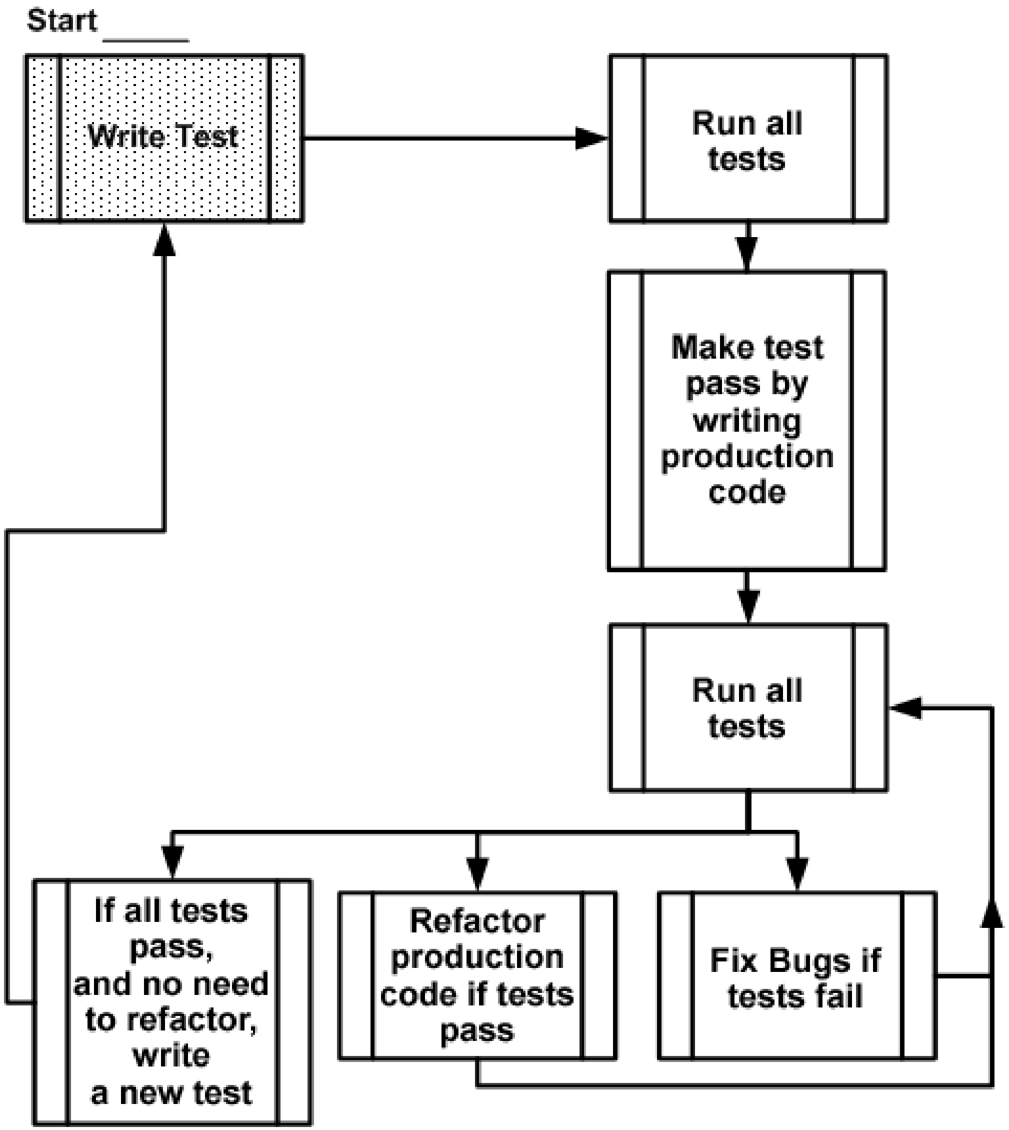

Write method, class, or application
Write unit tests (if time)
Run unit tests (if time)
Fix bugs

Copyright © 2013 Infogain Corporation. All rights reserved.

When to Write Test Case Contd.

A growing number of people prefer writing unit tests before the production code is written.

This approach is called test-first or test-driven development (TDD).


Start
Write Test
Run all tests
Make test pass by writing production code
Run all tests
If all tests pass, and no need to refactor, write a new test
Refactor production code if tests pass
Fix Bugs if tests fail

Copyright © 2013 Infogain Corporation. All rights reserved.

TDD : Approach

The technique of test-driven development is quite simple:

- ❑ Write a failing test to prove code or functionality is missing from the end product.
- ❑ Make the test pass by writing production code that meets the expectations of your test.
- ❑ Refactor your code.

Copyright © 2013 Infogain Corporation. All rights reserved.

infogain

Thank You

April 22, 2019

Copyright © 2013 Infogain Corporation. All rights reserved.