

Unit Testing Roundtable

By: Mike Shoemaker

What is a Unit Test?

A unit test examines the behavior of a distinct *unit of work*. Within a Java application, the “distinct unit of work” is often (but not always) a single method. By contrast, *integration tests* and *acceptance tests* examine how various components interact. A *unit of work* is a task that is not directly dependent on the completion of any other task.

Another way to look at a Unit Test?

The relationship a unit test has to a java program is similar to the relationship a use case has to a given scenario. Both are a single unambiguous path through a given problem domain. Just as use cases have alternate paths, unit tests should exercise negative behavior as well.

i.e. divide by zero errors

Excuses for not Unit Testing your code

- It takes too much time to write the tests
- It takes too long to run the tests
- It's not my job to test my code
- I don't really know how the code is supposed to behave so I can't test it.
- But it compiles!
- I'm being paid to write code, not to write tests.
- I fell guilty about putting testers and QA staff out of work
- My company won't let me run unit tests on the live system.

6 Steps to strengthen your testing skills

Remember “Right BICEP”

-

Right – Are the results right?

-

B – Are the boundary conditions correct?

-

I – Can you check the inverse relationship?

-

C – Can you cross check results using other means?

Boundary Conditions

-

Remember “CORRECT”

-

Conformance - does the value conform to an expected format?

-

Ordering - is the set of values ordered or unordered as appropriate?

-

Range - is the value within a reasonable min and max value?

Quiz

If you have 12 feet of lawn that you want to fence, and each section of fencing is 3 feet wide, how many fence posts do you need?

**What are the
consequences of not
Unit Testing?**

Collateral Damage!

This is what happens when a new feature or a bug fix in one part of the system causes a bug(damage) to another, possibly unrelated part of the system.

i.e. Whac-a-Mole

Resurfacing Bugs

Bugs that have been fixed come back

Greater Debugging

- Unit testing helps to limit the amount of debugging since you will have more confidence that the code works.
- Unit testing will also help keep class interfaces cleaner resulting in less spaghetti code

**What benefits come
from unit testing**

Enabling Effective Teamwork

- Unit test show other developers how a client would use a module they wrote
- This saves time, budget, and the frustration level of the client developer

Allows refactoring

- Refactoring how something works without unit test is a disaster waiting to happen
- With unit tests you can verify that you did not break anything that was previously working.

Developer Documentation

- Documentation on a shared network drive is rarely up to date
- Compiled unit test that pass are always up to date.
- Immediate feedback if a code change broke something

Greater test coverage

- Over time, more and more unit tests give you greater test coverage
- If you write a unit test to expose all bugs before fixing them, you assure that a previously untested path is not tested and guaranteed not break in the same manner again

Having Fun

- Writing tests becomes addictive
- Writing tests allows us to do more of what developers enjoy doing, writing code!

Properties of good tests - “A-TRIP”

- Automatic - running tests & checking results
- Thorough - code coverage
- Repeatable - tests should be independent of the environment
- Independent - tests should be independent of one another
- Professional - tests should be written using standards set for production code.

Test Frequency

- Write a new method
- Fix a bug
- Successful compile after changes are made
- Before each check in !!!!!!!
- Continuously, since the longer you wait to integrate the more costly it is

References

- Junit in Action – Vincent Massol, 2004
- Pragmatic Unit Testing – Andy Hunt, Dave Thomas 2003
- <http://www.martinfowler.com/articles/mocksArentStubs.html> - Martin Fowler
- <http://www.mockobjects.com>

Group Exercise

- Create a Shopping Cart Application in a that meets the following requirements
 - Ability to retrieve the item count
 - Ability to add items to the cart
 - Ability to retrieve the item
 - Ability to remove items into the cart
 - Ability to clear items from the cart
 - Ability to override an item in the cart, i.e. to update quantities
 - Ability to retrieve Collection of items in cart
 - Ability to retrieve total cost of shopping cart
 - item price * quantity
 - tax is not in scope