How to assert

Richard van den Ham
Pieter van den Hombergh
Linda Urselmans

February 7, 2019

Fontys Hogeschool voor Techniek en Logistiek

Topics

Fontys Hogescholen

Introduction

Introduction

Test wise a good test. It broke things.

Fontys Hogescholen

Introduction

Test wise a good test. It broke things.

Engineers use Tools

New, New

- You will learn to work with new tools, techniques and processes.
- We will introduce new tools and concepts almost every week. The pace is high.
- Testing itself, however, is really quite simple.

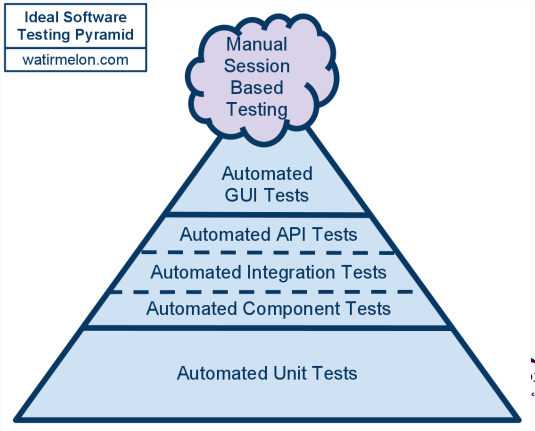
Fontys
Hogescholen

Test levels

OO and Types Of Tests

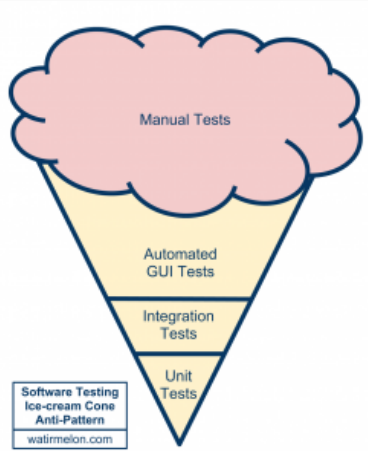
In object oriented development, like in Java, we can distinguish between several kinds of tests. They can be ordered in a layer hierarchy. The top and bottom typically have common names. The three main test types are:

- **End to End Tests** or **UI** tests at the top
- **Unit Tests** at the bottom
- **Integration Tests** in the middle. In the middle layers you see various other names as well



Most testing reality

- In reality there is a triangle, but of a different kind.
- Much nicer  . Sometimes it's a bit messy and may cause slippery floors. 
- Which model do you think is best? The A) ice-cone or the B) Pyramid?



The ice cream cone.

The importance of Unit test in the OO world

- If you answered B) Pyramid, you may stay.

When an application is developed in OO style, the application is developed by breaking the whole problem into smaller sub-problems, like classes and objects, and then the application is build up by assembling it from the classes.

The unit tests tests the classes, the building blocks of the application and with it testing of these building blocks forms the foundation on which you rely.

Proper Unit Testing is the foundation of test automation and continuous building and delivery.

Fontys Hogescholen

Junit asserts

How to Assert

- We will be using JUnit.
- Looking more precisely to the details.

Fontys Hogescholen

to assert or not to assert

- `assert` is a java keyword

```
public class AssertDemo {
    public static void main( String[] args ) {
        assert args.length > 0;
    }
```

- assert tests the expression following the keyword to be true.
  - If not it throws an `AssertionError` exception.
- Such assert tests are turned off by default but can be turned **on** by starting the JVM with `-ea` (for enable assertions). Demo.

Although `assert` has its purpose, it is typically not used very often. In particular not for testing.

Fontys Hogescholen

Order of parameters and message

- Looking at org.junit.Assert.java we see a class with only static methods of the type:

  `static public void assertEquals(String msg, Object expected, Object actual )`. Many methods from org.junit.Assert.java have the same signature.
- Order: (Junit 4)
  1. Message
  2. Expected value expression[1]
  3. Actual value (in case of equals, same)
  4. For doubles and floats, you don't do an exact comparison, but instead allow for some margin. And this margin is the fourth parameter, called delta.
- It is good style to always add a meaningful or identifying message, because it helps (you) to find the exact spot where things went wrong.
- Let us have a look (demo)

[1] expected and actual should evaluate to primitive types or object references

The zoo of Asserts

True expression must be true

False expression must be false

Null expression must be null

NotNull expression must be not null

Equals expected and actual expression must be equal according to `Object.equals( Object )`.

Same **Quiz:** what is the difference with equals?

Fontys Hogescholen

Arrays can be compared too

- Note that **size and order** of elements matter.
- If you want to assert that arrays contain the same members, irrespective of order, sort the arrays, then compare.
- *or* use a smarter Testing framework like AssertJ which has assertions like

Ring Any Bells?

```
assertThat(fellowshipOfTheRing)
    .extracting("name", String.class)
    .contains("Boromir", "Gandalf", "Frodo", "Legolas")
```

In a test it is perfecly fine to dump your array in a List and then do the assert on the list, as long as the test is simple to read and executes quickly.

Fontys Hogescholen

Sometimes simple assert is not enough

- Then you can implement your own 'rules' matching the business rule.
- The order is msg, actual, matcher, and not what you would perhaps expect (msg, expected, actual).
  - The reason (I infer) is that the matcher is typically a anonymous inner class, because you implement it on the spot. Having this anonymous inner class in the middle would give even more awkward code.
- Example on next slide.

Fontys Hogescholen

Cola light;

```
        assertThat( "my taste buds have been violated! ", taste,
                new BaseMatcher() {
            @Override
            public boolean matches( Object item ) {
                Taste t = (Taste) item; // cast to Taste enum
                switch(t) {
                    case COLA: case COGNAC: case WHISKEY:
                        return true;
                        default:
                    case COLALIGHT:
                        return false;
                }
            }

            @Override
            public void describeTo( Description description ) {
                description.
                        appendText( "HOM hates light stuff, "
                                + "it is for sissies or people "
                                + "without discipline");
            }
        });
    }
```

Unit tests

What are unit tests?

- Unit tests test a UNIT, duh.
- The UNIT is ONE class, the **S**ystem **U**nder **T**est (SUT)
- Intended to make your code *fail fast*
  - Meaning it should run fast.
  - It should NOT test other classes, the Dependent On Component (DOC) or collaborators.
- It helps you correct your code and improve your implementation
- Often you see JUnit used as the testing framework. That does not make the tests unit tests. For instance JUnit may very well be used in integration tests and even end to end tests.

Fontys Hogescholen

Problem: How to find test data

- First and foremost they come from your "Business"[2].
- Boundary value analysis
- Equivalence class partitioning

[2]Which does not say it is about business informatics, business in the broader sense. That may very well be creating some utility class

Fontys Hogescholen

Intermezzo: Business *noun, 2*

1. ROLE, FUNCTION
   // how the human mind went about its business of learning
   H. A. Overstreet
2. an immediate task or objective : MISSION
   // what is your business here?
3. a particular field of endeavor
   // the best in the business

From Merriam Webster lemma Business(2).

Fontys Hogescholen

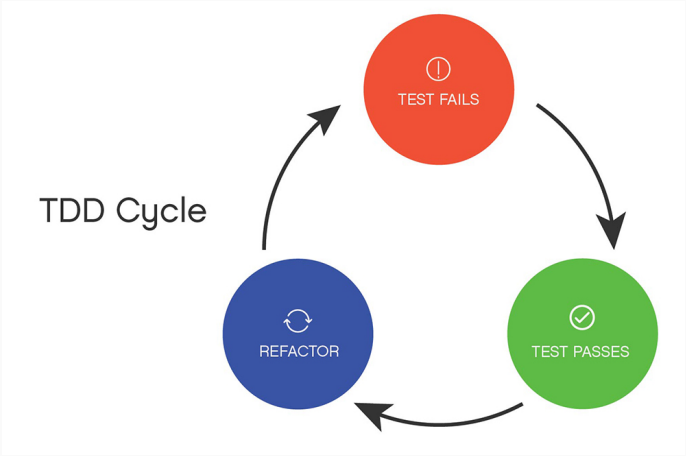
Creating tests is easy

- Using Netbeans IDE.
- Letting netbeans guess the SUT code is easy too.
- let's TDD.

Fontys Hogescholen

TDD

Test driven development



from **IPPON**

Fontys Hogescholen

The promises

- Your write **less code!**!?
- The little code you write is of a better quality. Rationale: The less you write, the less you can err.

From the teachers: Actual personal experience too.

Fontys Hogescholen

The Fine Print, A.K.A. The Dos and Don'ts.

DO Develop in small steps:

- Write a test (method) for the next open requirement.
- See it fail.
- Write the simplest implementation passing the test.
- See it pass.
- Refactor the implementation and retest
- Move to the next requirement

- DO NOT Write all tests up front.
- DO NOT Write the implementation first. We need the test to be RED.
- DO NOT Write tests that never fail, such as having no assert or equivalent.

Remember: A good test is a **RED** test.

Fontys Hogescholen