

Outline
Unit testing
Why should I bother with Unit testing
Sorry excuses
First Unit tests

Methods to Improve Quality

Pieter van den Hombergh


13 september 2005



Pieter van den Hombergh Methods to Improve Quality

Outline
Unit testing
Why should I bother with Unit testing
Sorry excuses
First Unit tests

- 1 Unit testing
- 2 Why should I bother with Unit testing
 - Collateral damage
- 3 Sorry excuses
 - It ...
 - But I already got...
- 4 First Unit tests
 - simple assertions




Pieter van den Hombergh Methods to Improve Quality

Outline
Unit testing
Why should I bother with Unit testing
Sorry excuses
First Unit tests

Motivation for Unit Testing

Or what excuses are there to not do unit testing.




Pieter van den Hombergh Methods to Improve Quality


Outline
Unit testing
Why should I bother with Unit testing
Sorry excuses
First Unit tests


Does my code do what I want?

- Or rather, does the code (under test) fulfil my intent?
- Does it do what I want all of the time?
 - Every program has more than one path of execution, some of them leading to the success scenario, others to failure
Programmers who say they test, very often only test with the **right** input. (By the way: *[h|cr]ackers tend to test with **wrong** input*)
- So test with non existing file, invalid input, (simulated) disk full situations etc.



Pieter van den Hombergh Methods to Improve Quality

| | |
|--|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | Collateral damage |
| <h2>What do I want to accomplish?</h2> | |
| <ul style="list-style-type: none">• Do not get carried away<ul style="list-style-type: none">• Once you get the hang of it, you might think its fun doing unit test. Do not get carried away. E.g. do not test setters and getters if setting a value does not change an object's state. | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |


| | |
|---|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | Collateral damage |
| <h2>Test your exceptions</h2> | |
| <p>Typical code fragment by novices</p> <pre>catch(Exception e) { // <i>silenty ignore</i> }</pre> | |
| <p>Having code like this drops information on the floor. Either really handle the exception or log the information somewhere.</p> | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |

Outline
Unit testing
Why should I bother with Unit testing
Sorry excuses
First Unit tests

Collateral damage

What is collateral damage?

Collateral damage is what happens when a new feature bug fix in one part of the system causes a bug (damage) to another, possibly unrelated part of the system. It's an insidious problem that, if allowed to continue, can quickly render the entire system broken beyond anyone's ability to fix.





Pieter van den Hombergh Methods to Improve Quality

Outline
Unit testing
Why should I bother with Unit testing
Sorry excuses
First Unit tests

Collateral damage

What is collateral damage?

We sometime call this the "Whac-a-Mole" effect.



Pieter van den Hombergh Methods to Improve Quality

| | |
|---|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | Collateral damage |
| <p>In the carnival game of Whac-a-Mole, the player must strike the mechanical mole heads that pop up on the playing field. But they don't keep their heads up for long; as soon as you move to strike one mole, it retreats and another mole pops up on the opposite side of the field. The moles pop up and down fast enough that it can be very frustrating to try to connect with one and score. As a result, players generally flail helplessly at the field as the moles continue to pop up where you least expect them. Widespread collateral damage to a code base can have a similar effect. <i>From Pragmatic unit testing</i></p> | |
| Pieter van den Hombergh | Methods to Improve Quality |



| | |
|--|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | Collateral damage |
| <h2>Can I depend on my code?</h2> <p>Code you cannot depend on is useless. Worse, if you do not know it is (un)reliable, you might be looking for bugs at the wrong places. Testing helps you to assess the validity of the code. So if it blows up, you know what caused it and how to signal it.</p> | |
| Pieter van den Hombergh | Methods to Improve Quality |




Collateral damage

Outline
 Unit testing
 Why should I bother with Unit testing
 Sorry excuses
 First Unit tests

How do I do unit Testing?

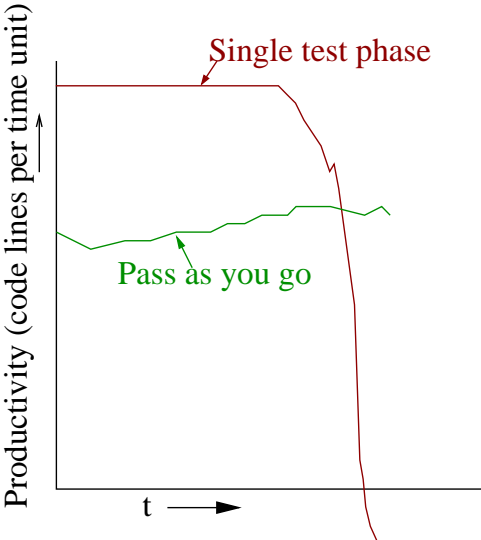
- 1 Write the unit test from the specifications of the code.
- 2 Run it.
- 3 It might not compile or fail because it cannot find the class under test.
- 4 The write the class to test.
- 5 Then cycle through
 - 1 Run the tests
 - 2 Correct the failure
 - 3 Until all test pass.




Pieter van den Hombergh
Methods to Improve Quality

It ...
But I already got...


Outline
 Unit testing
 Why should I bother with Unit testing
 Sorry excuses
 First Unit tests





- It takes to much time to write the tests
 - *It takes more time to find bugs, even if you intend not to write any. Unit test lets the bugs find you. Unit testing is a method to keep your productivity constant and reliable*





Pieter van den Hombergh
Methods to Improve Quality


| | |
|--|--------------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | It ... But I already got... |
| It takes too much to run the tests | |
| <p>It should n't. Most unit test run quickly. For tests that take a long time to run, run them less often.</p> | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |


| | |
|--|--------------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | It ... But I already got... |
| It is not my job to test my code | |
| <p>Then what is your job? Hand out buggy code?</p> | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |


| | |
|---|--------------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | It ... But I already got... |
| <h2>I don't really know how my code is supposed to behave, so I cannot test it</h2> | |
| <p>Maybe it is not the right time writing your code yet. A (testable) prototype might help clarify the exact requirements. If you do not know what the code is supposed to do, then how do you know it does?</p> | |
|  | |
| Pieter van den Hombergh Methods to Improve Quality | |

| | |
|---|--------------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | It ... But I already got... |
| <h2>But it compiles!</h2> | |
| <p>Your compiler might catch this</p> <pre>public static void main(Strings args []) { // more code left out }</pre> <p>but how about this?</p> <pre>1 public void addit(Object anObject) { 2 ArrayList myList= new ArrayList(); 3 myList.add(anObject); 4 myList.add(anObject); 5 // more code ... 6 }</pre> <p>How should the compiler know?</p> | |
|  | |
| Pieter van den Hombergh Methods to Improve Quality | |

| | |
|--|---------------------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | It ... But I already got... |
| <h2>I'm paid to write code, not to write tests.</h2> | |
| <p><i>By the same logic you are not paid to spend all day in the debugger.</i></p> | |
|  | |
| Pieter van den Hombergh Methods to Improve Quality | |

| | |
|---|---------------------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | It ... But I already got... |
| <h2>Where that 'but...' could lead to</h2> | |
| <p>A consultant recently came to our team asking for help with his project. He pulled out a print-out of his Java source file to reveal the god class to end all god classes, weighing in at a whopping 9700 lines long. I flipped to a random page (one out of 87 printed pages) and saw two squelched exceptions i.e.</p> <pre style="margin-left: 40px;"> catch(SQLException ex) { } </pre> <p>in an if statement nested 6 levels deep. A quick shuffle through the rest of the stack revealed more of the same. I suggested to the others on my team that finding something as bad as this was a special occasion, and we should celebrate. <small>by ChadFowler.</small></p> | |
|  | |
| Pieter van den Hombergh Methods to Improve Quality | |

| | |
|---|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | simple assertions |
| <h2>Test are quite simple</h2> | |
| <p>You can use several rather simple methods, provide by the framework. They typically look much like this:</p> | |
| <pre>public void assertTrue(boolean condition) { if (!condition) { abort(); } }</pre> | |
| <p>As you can see they do nothing if the test-condition is true.</p> | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |

| | |
|---|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | simple assertions |
| <p>Typical use in a unit test would be:</p> | |
| <pre>int a = 2; // some intermediate code // assertTrue(a == 2); // more tests?</pre> | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |

| | |
|---|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | simple assertions |
| Or asserts like this | |
| <pre>public void assertsEquals(int a , int b) { assertTrue(a == b); }</pre> | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |

| | |
|--|----------------------------|
| Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests | simple assertions |
| These kind of assertions exist for the standard java types and then some. You could also create your own: (Even money: whole dollars, euros, not cents) | |
| <pre>class Money { public void assertEvenMoney(String Message , Money amount) { assertEquals(message , amount.asDouble(), Math.floor(amount.asDouble()), 0.001); } public void assertEventMoney(Money amount) { assertEvenMoney("" , amount); } }</pre> | |
|  | |
| Pieter van den Hombergh | Methods to Improve Quality |

| | |
|--|-----------------------------------|
| <p>Outline Unit testing Why should I bother with Unit testing Sorry excuses First Unit tests</p> | <p>simple assertions</p> |
| <p>Ensuring that (the proper) Exceptions are thrown can be tested like this</p> <pre>public void testForException(SomeClass sc) { try { ct.(sc); fail("exception not thrown"); } catch (ExceptionThatWasExpected etwe) { assertTrue(true); // code = documentation } catch (RuntimeException re) { fail("wrong exception thrown" + re.getMessage()); } }</pre> | |
| <p>Pieter van den Hombergh</p> | <p>Methods to Improve Quality</p> |

