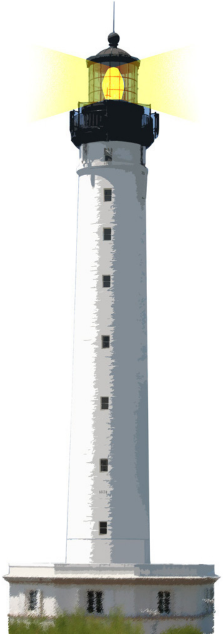
Why Testing in a Nutshell

Damien Cassou, Stéphane Ducasse and Luc Fabresse

WXSYY

http://www.pharo.org



Goal of the lecture

- How can you trust that the changes did not destroy something?
- What is my confidence in the system?
- How do I write tests?
- What is unit testing?

Tests in a nutshell

- Tests represent your trust in the system
- Build them incrementally
    - Do not need to focus on everything
    - When a new bug shows up, write a test
- Even better write them before the code
    - Act as your first client, better interface
- Active documentation always in sync
- It has a cost, but pay off is big
    - writing them, maintain them

Testing Style

" The style here is to write a few lines of code, then a test that should run, or even better, to write a test that won’t run, then write the code that will make it run. "

- write unit tests that thoroughly test a single class
- write tests as you develop (even before you implement)
- write tests for every new piece of functionality

Developers should spend 25-50% of their time developing tests.

But I can’t cover everything!

- Sure! Nobody can but
- When someone discovers a defect in your code, first write a test that demonstrates the defect.
- Then debug until the test succeeds.

Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead. Martin Fowler

Unit Tests

- Unit tests ensure that you get the specified behavior of the public interface of a class
- Normally tests a single class

A Test

A test

- Create a context
- Send a stimulus
- Check the results

A Test

A test

- Create a context: Create an empty set
- Send a stimulus: Add twice the same element
- Check the results: Check that the set contains only one element

Set TestCase

```
TestCase subclass: # SetTestCase
   ...
```

```
SetTestCase >> testAdd
  | empty |
  empty := Set new.                "Context"
  empty add: 5.                   "Stimulus"
  empty add: 5.
  self assert: empty size = 1.      "Check"

SetTestCase run: #testAdd
```

Good Tests

- Repeatable
- No human intervention
- "self-described"
- Change less often than the system
- Tells a story

A course by

inria informatiques mathématiques

and

UTOP Université de Technologie Ouverte Pluripartenaire

in collaboration with

uni sciel

Fondation unit Université Numérique Ingénierie et Technologie

Université de Lille

INSTITUT Mines-Télécom

MINES Douai