

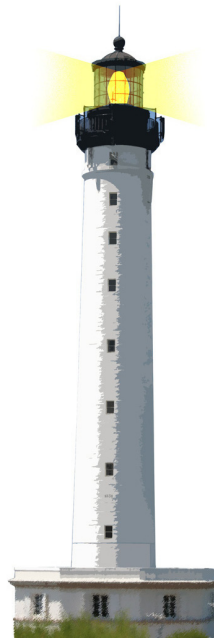
Test-Driven Development

Getting more out of tests

S. Ducasse



<http://www.pharo.org>



Outline

- Example of TDD
- Rethinking it



TDD is about the flow

- Write test first! Yes, yes, yes
- "Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead" - Martin Fowler



TDD: Write your test

Imagine we just created the class Counter

```
CounterTest >> testCount
```

```
| c |
```

```
c := Counter new.
```

```
c count: 10.
```

```
self assert: c count equals: 12
```

TDD: Run your test!

- It is red (the test produces an error)
 - This is normal since we did not define the methods `count` and `count:`.
- So far so good!
- If it is already green this is a bonus and may be your test is not good.



TDD: Make your test pass

- Work until your test gets green!
- [When you are lucky, use XTDD and enjoy coding in the debugger]



TDD: Rerun ****all**** the tests

- You are super smart but ****!**
- Fix the broken tests
 - Either a test is now wrong
 - Or you broke something else and you should fix it

TDD: When all the tests are green

- Commit and take a break
- Then it is a good time to
 - Refactor your code if necessary
 - Revisit your code



Why writing test first?

- You specify what you want to get
- In addition you are your first client: strengthen your APIs
- You get a **clear** context
- You can debug on the spot



Conclusion

- TDD is powerful
- XTDD is even more (see following lectures)



A course by

S. Ducasse, L. Fabresse, G. Polito, and Pablo Tesone



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France

<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>