# Writing Testable Code

Alvaro Videla - Cloud Foundry

## About Me

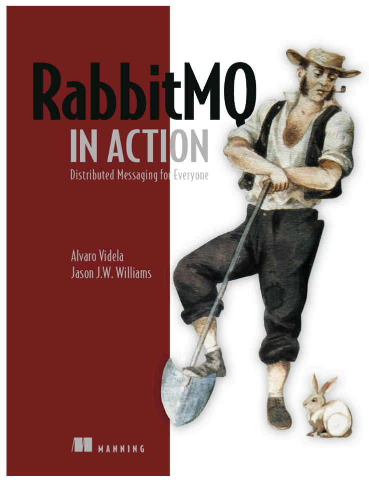
- Cloud Foundry Developer Advocate
- Blog: http://videlalvaro.github.com/
- Twitter: @old_sound

# About Me

Co-author

RabbitMQ in Action

http://bit.ly/rabbitmq



## I’m not a:

## I’m not a:

- Application Testing Guru

# I’m not a:

- Application Testing Guru
- TDD Advocate

## Why is it so hard to write tests?

## Unit Testing

**The goal of unit testing is to isolate each part of the program and show that the individual parts are correct**

http://en.wikipedia.org/wiki/Unit_testing

## Unit Testing

**[...] unit testing by definition only tests the functionality of the units themselves.**

http://en.wikipedia.org/wiki/Unit_testing

## Unit Testing

**[...] Therefore, it will not catch integration errors or broader system-level errors (such as functions performed across multiple units, or non-functional test areas such as performance)**

http://en.wikipedia.org/wiki/Unit_testing

## Dogma
## vs.
## Reality

## A world of Trade Offs

## What should we test?

## How much should we test?

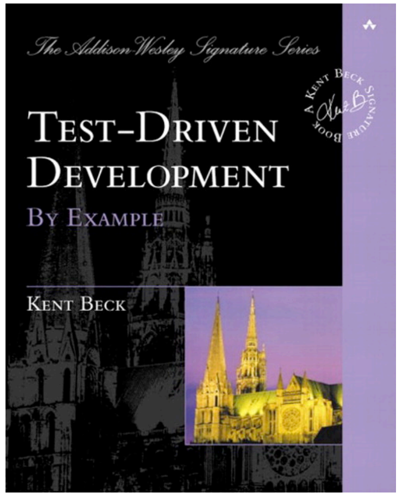
**“I get paid for code that works, not for tests, so my philosophy is to test as little as possible to reach a given level of confidence”**

**– Kent Beck**

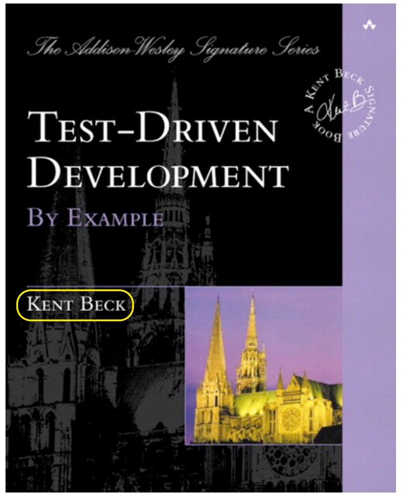
http://stackoverflow.com/questions/153234/how-deep-are-your-unit-tests/153565#153565

## The Hidden
## Secret
## Of TDD

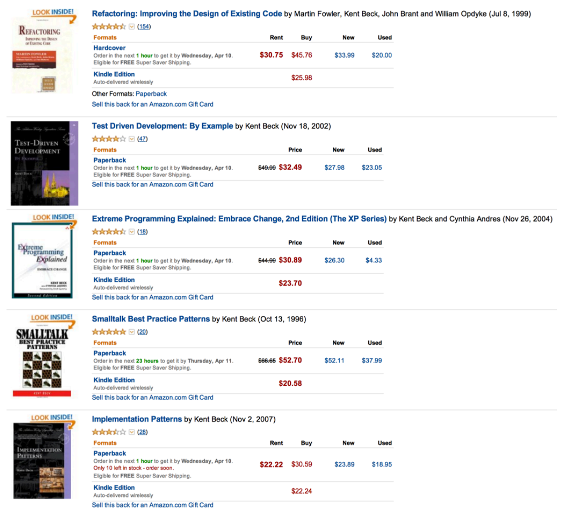
## The Secret of TDD


The Addison Wesley Signature Series
Test-Driven Development
By Example
Kent Beck
A Kent Beck Signature Book

# The Secret of TDD


The Addison-Wesley Signature Series
Test-Driven Development
By Example
Kent Beck
A Kent Beck Signature Book

# Some books by Kent Beck



**Refactoring: Improving the Design of Existing Code** by Martin Fowler, Kent Beck, John Brant and William Opdyke (Jul 8, 1999)

(154)

| Formats | Rent | Buy | New | Used |
|---|---|---|---|---|
| **Hardcover** Order in the next **1 hour** to get it by **Wednesday, Apr 10.** Eligible for **FREE** Super Saver Shipping. | $30.75 | $45.76 | $33.99 | $20.00 |
| **Kindle Edition** Auto-delivered wirelessly | | $25.98 | | |

Other Formats: Paperback

Sell this back for an Amazon.com Gift Card

**Test Driven Development: By Example** by Kent Beck (Nov 18, 2002)

(47)

| Formats | Price | New | Used |
|---|---|---|---|
| **Paperback** Order in the next **1 hour** to get it by **Wednesday, Apr 10.** Eligible for **FREE** Super Saver Shipping. | ~~$49.99~~ $32.49 | $27.98 | $23.05 |

Sell this back for an Amazon.com Gift Card

**Extreme Programming Explained: Embrace Change, 2nd Edition (The XP Series)** by Kent Beck and Cynthia Andres (Nov 26, 2004)

(18)

| Formats | Price | New | Used |
|---|---|---|---|
| **Paperback** Order in the next **1 hour** to get it by **Wednesday, Apr 10.** Eligible for **FREE** Super Saver Shipping. | ~~$44.99~~ $30.89 | $26.30 | $4.33 |
| **Kindle Edition** Auto-delivered wirelessly | $23.70 | | |

Sell this back for an Amazon.com Gift Card

**Smalltalk Best Practice Patterns** by Kent Beck (Oct 13, 1996)

(20)

| Formats | Price | New | Used |
|---|---|---|---|
| **Paperback** Order in the next **23 hours** to get it by **Thursday, Apr 11.** Eligible for **FREE** Super Saver Shipping. | ~~$66.65~~ $52.70 | $52.11 | $37.99 |
| **Kindle Edition** Auto-delivered wirelessly | $20.58 | | |

Sell this back for an Amazon.com Gift Card

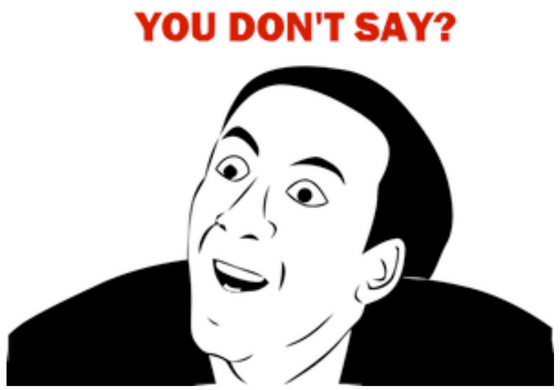
**Implementation Patterns** by Kent Beck (Nov 2, 2007)

(28)

| Formats | Rent | Buy | New | Used |
|---|---|---|---|---|
| **Paperback** Order in the next **1 hour** to get it by **Wednesday, Apr 10.** Only 10 left in stock - order soon. Eligible for **FREE** Super Saver Shipping. | $22.22 | $30.59 | $23.89 | $18.95 |
| **Kindle Edition** Auto-delivered wirelessly | | $22.24 | | |

Sell this back for an Amazon.com Gift Card

**To write good tests first we need to learn how to program**


YOU DON'T SAY?

We developers are like those users we like to complain so much about

## Design evolves and matures with time

## Good Code sits in the small details

## TIPS

Separate *pure* code
from *impure* or *stateful*

## Pure Functions

## Pure Functions

- Referential Transparency

## Pure Functions

- Referential Transparency
- Don’t modify external state

## Pure Functions

- Referential Transparency
- Don’t modify external state
- Don’t produce side effects

## What’s wrong with this code?

```
if($player->getScore() > 0) {
  $player->setSwizzle(7);
} else {
  $player->setSwizzle(
    $player->getSwizzle() + 1
  );
}
```

https://dl.dropboxusercontent.com/u/7810909/docs/what-does-fp-mean/what-does-fp-mean/chunk-html/ar01s05.html

## What’s wrong with this code?

```
$newScore = $player->getScore() > 0
            ? 7
            : $player->getSwizzle() + 1;

$player->setSwizzle($newScore);
```

https://dl.dropboxusercontent.com/u/7810909/docs/what-does-fp-mean/what-does-fp-mean/chunk-html/ar01s05.html

**Score calculation can be moved into its own function**

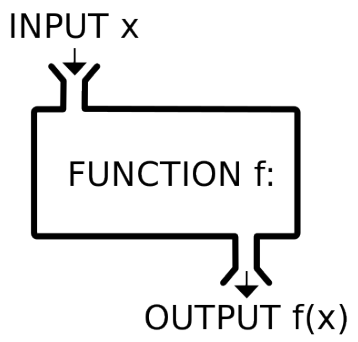
## Score calculation can be tested now

## First write
## Pure Code

## Add impure code step by step when needed

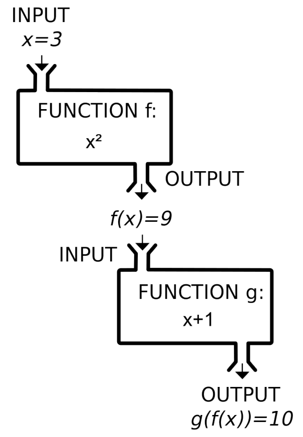
## Write Composable Code

# Function Composition



http://en.wikipedia.org/wiki/Function_(mathematics)

## Function Composition



http://en.wikipedia.org/wiki/Function_(mathematics)

This looks familiar

**“Many UNIX programs do quite trivial tasks in isolation, but, combined with other programs, become general and useful tools.”**

http://math.albany.edu/math/pers/hammond/unixphil.html

## Number of open connections per IP

```
netstat -ntu | awk '{print $5}' | \
cut -d: -f1 | sort | uniq -c | sort -n
```

http://www.commandlinefu.com/commands/view/1767/number-of-open-connections-per-ip.

Why don’t we just code in this style?

This seems familiar again…

# Welcome to Functional Programming

## “Writing unit tests is reinventing functional programming in non-functional languages”

http://noss.github.io/2009/02/25/writing-unit-tests-is-reinventing-functional-programming-in-non-functional-languages.html

## What can we learn from Functional Programming?

## The proper use of Types

What does ‘**null**’ mean?

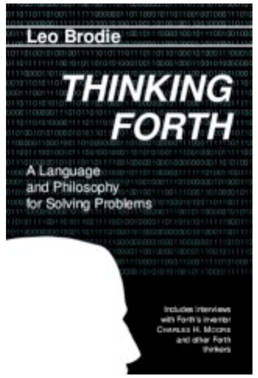
## What does ‘**true|false**’ mean?

## Functions with just one responsibility

Radical separation of pure code from impure code

Let’s see an example

## Food for Thought


Leo Brodie
THINKING FORTH
A Language and Philosophy for Solving Problems
Includes interviews with Forth's inventor Charles H. Moore and other Forth thinkers

http://thinking-forth.sourceforge.net

“Inside every well-written large program is a well-written small program”

http://www.linfo.org/q_programming.html

## Questions?

# Thanks!

http://twitter.com/old_sound

http://github.com/videlalvaro

http://www.slideshare.net/old_sound