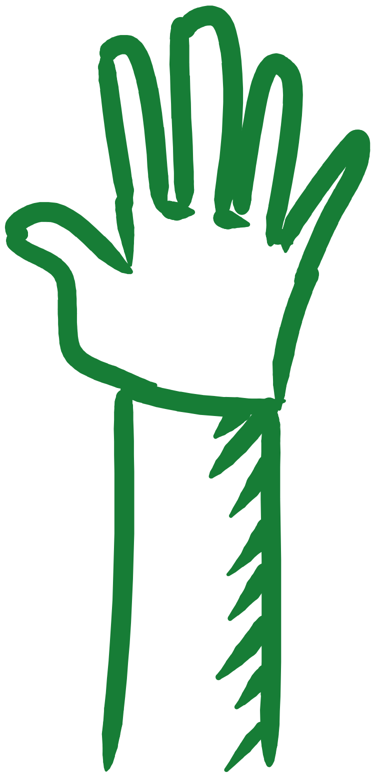
Executable Documentation

for everyone (even you)

Who ...



... has heard of it?

... is using it?

Expect

Why What

1 Executable Documentation

How

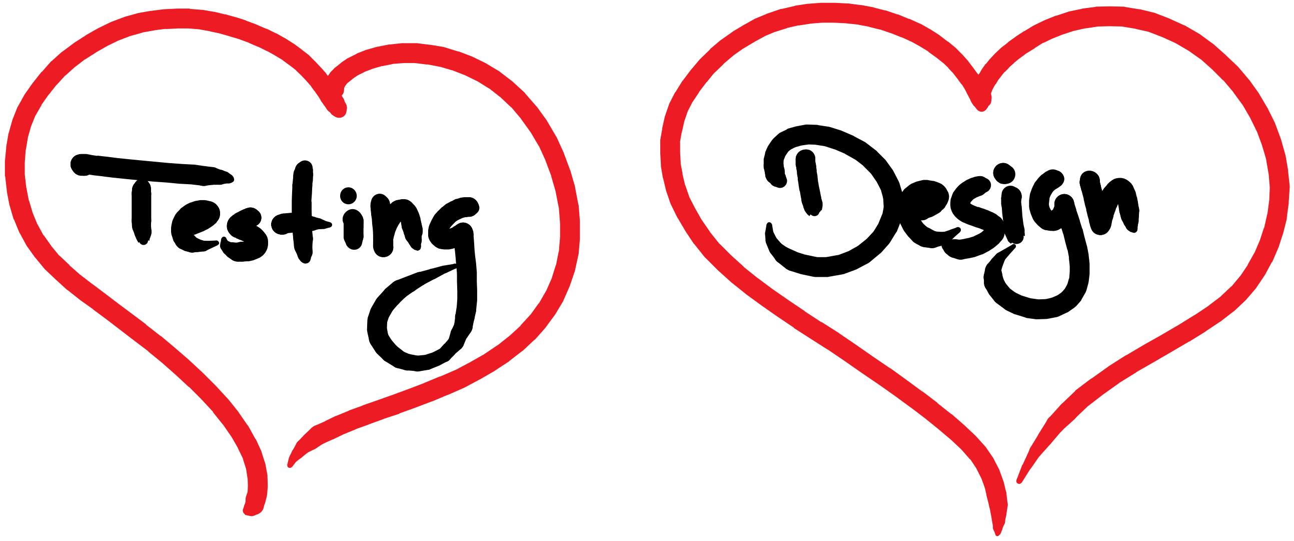
2 dox Publishing tool

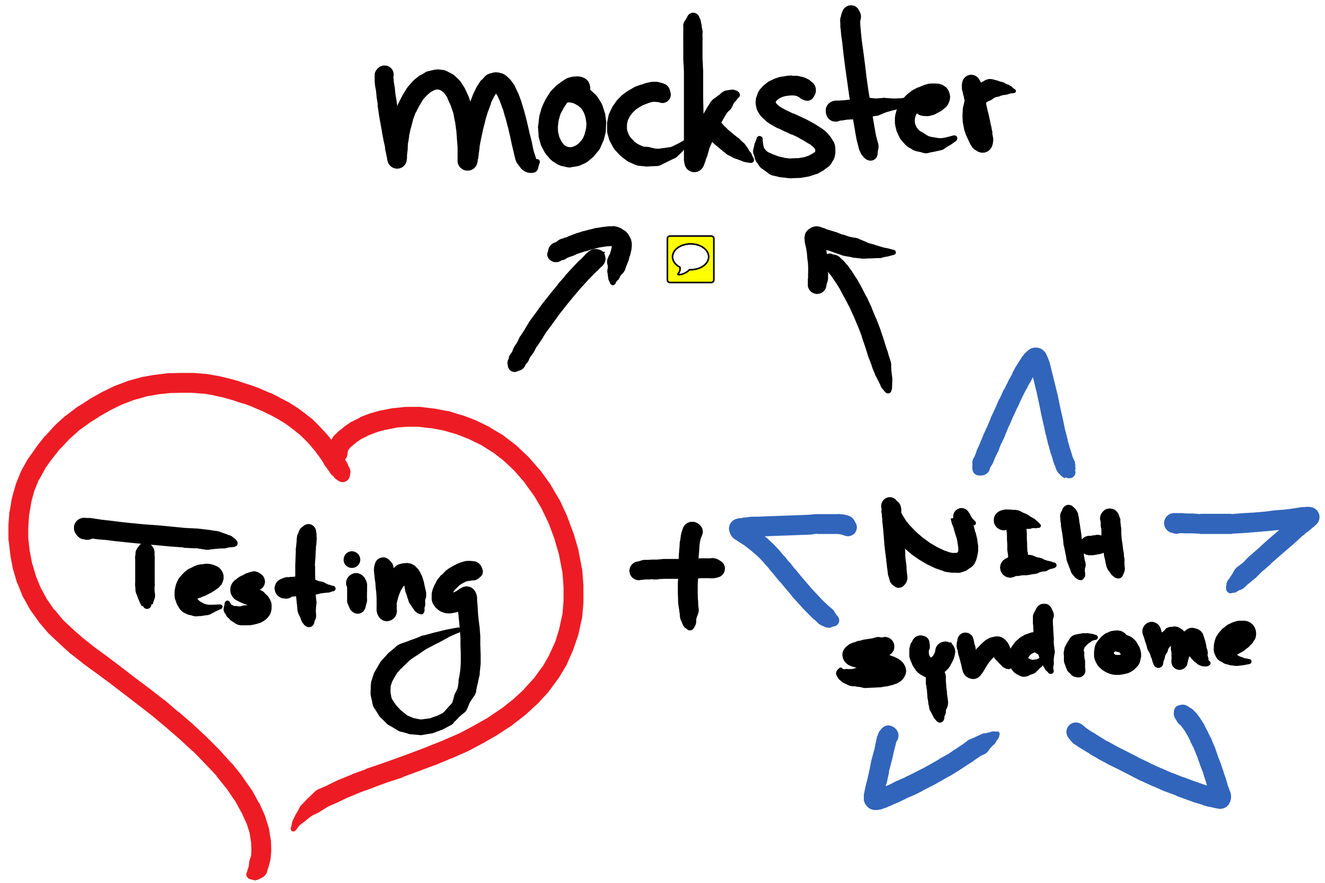
Me:

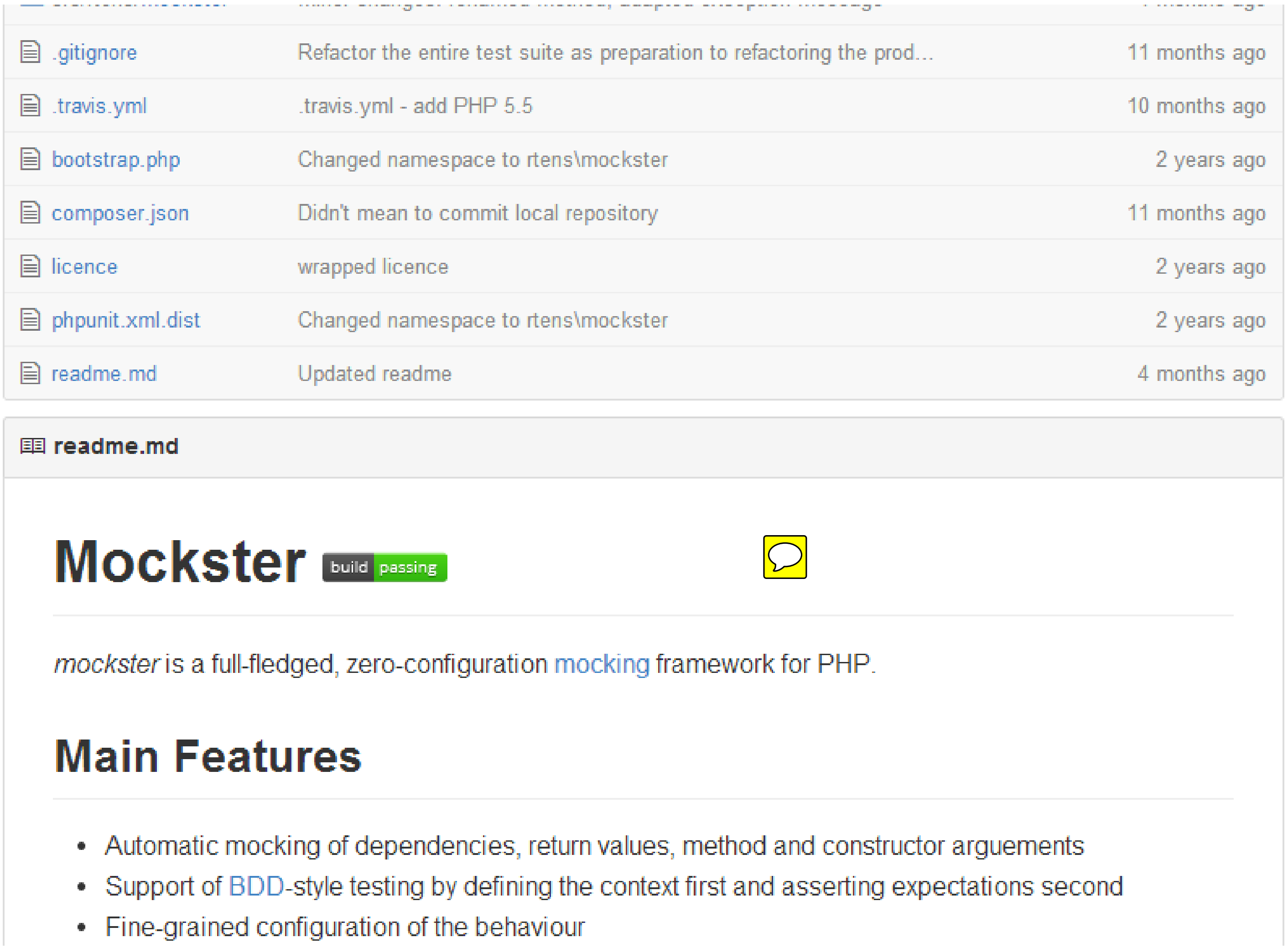
Nikolas Martens

Engineer & Coach

Me:




mockster
Testing
+
NIH syndrome



.gitignore	Refactor the entire test suite as preparation to refactoring the prod…	11 months ago
.travis.yml	.travis.yml - add PHP 5.5	10 months ago
bootstrap.php	Changed namespace to rtens\mockster	2 years ago
composer.json	Didn't mean to commit local repository	11 months ago
licence	wrapped licence	2 years ago
phpunit.xml.dist	Changed namespace to rtens\mockster	2 years ago
readme.md	Updated readme	4 months ago

readme.md

Mockster build passing

mockster is a full-fledged, zero-configuration mocking framework for PHP.

Main Features

- Automatic mocking of dependencies, return values, method and constructor arguements
- Support of BDD-style testing by defining the context first and asserting expectations second
- Fine-grained configuration of the behaviour

Basic Usage

First, we need an instance of MockFactory. It extends Factory so it supports singletons and providers (if needed).

```
$factory = new MockFactory();
```

To get a completely empty mock which is but a hollow shell of the given class, use

```
$mock = $factory->getInstance('MyClass');
```

The created instance extends the given class but does not invoke its parent's constructor, nor does any method call actually reach the parent - they are all mocked.

If you want to call the parent's constructor, pass an array with the constructor arguments. If you don't want to pass any arguments, provide an empty array.

```
$mock = $factory->getInstance('MyClass', array('name' => 'Foo'));
```

Basic Usage

First, we need an instance of MockFactory. It extends Factory s
needed).

```
$factory = new MockFactory();
```

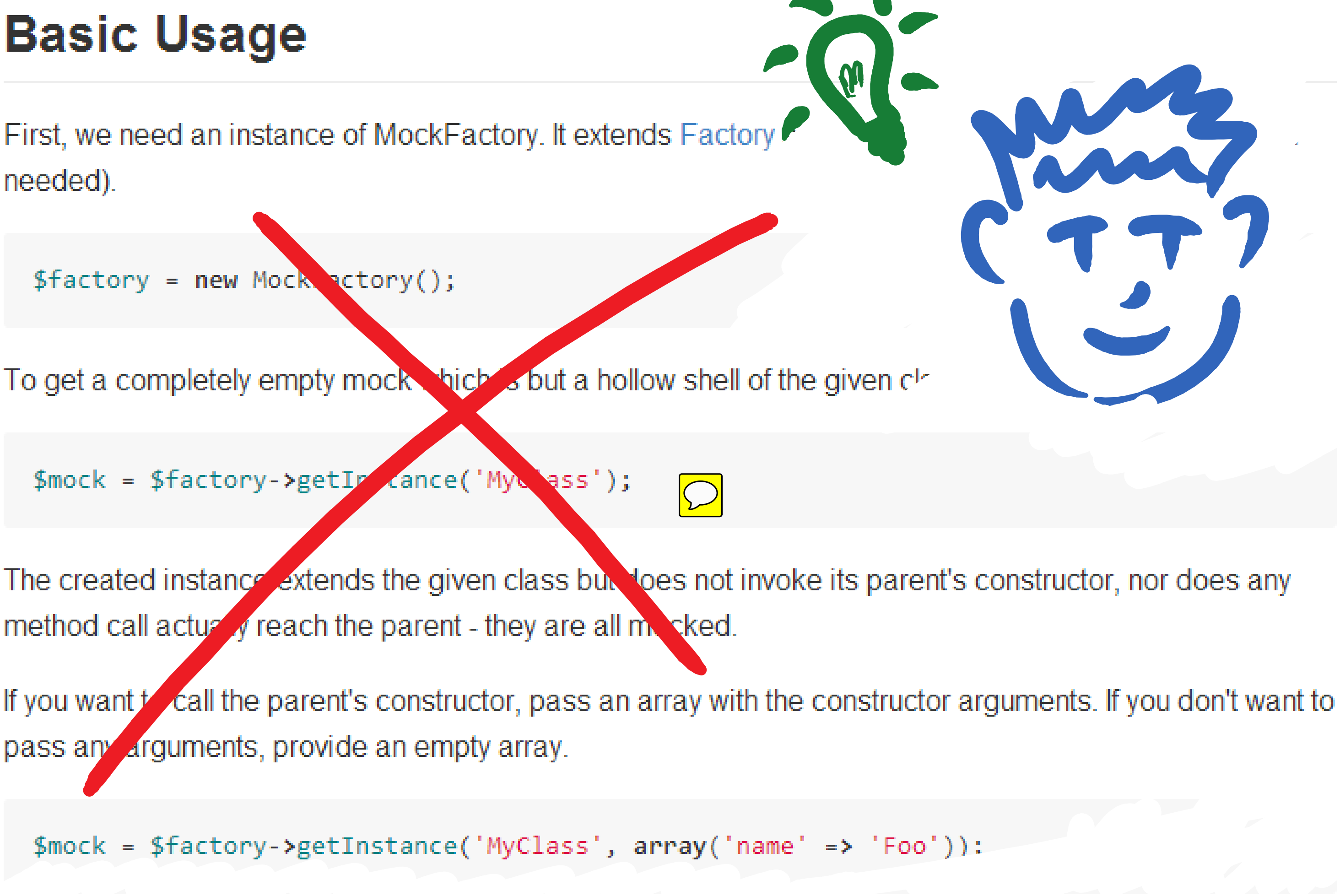
To get a completely empty mock which is but a hollow shell of the given cl
:

```
$mock = $factory->getInstance('MyClass');
```

The created instance extends the given class but does not invoke its parent's constructor, nor does any method call actually reach the parent - they are all mocked.

If you want to call the parent's constructor, pass an array with the constructor arguments. If you don't want to pass any arguments, provide an empty array.

```
$mock = $factory->getInstance('MyClass', array('name' => 'Foo'));
```



Basic Usage

First, we need an instance of MockFactory. It extends Factory

needed).

```
$factory = new MockFactory();
```

To get a completely empty mock which is but a hollow shell of the given cl

```
$mock = $factory->getInstance('MyClass');
```

The created instance extends the given class but does not invoke its parent's constructor, nor does any method call actually reach the parent - they are all mocked.

If you want to call the parent's constructor, pass an array with the constructor arguments. If you don't want to pass any arguments, provide an empty array.

```
$mock = $factory->getInstance('MyClass', array('name' => 'Foo'));
```

Basic Usage

First, we need an instance of Moc

needed).

```
$factory = new MockFactory();
```

To get a completely empty mock which is but a hollo

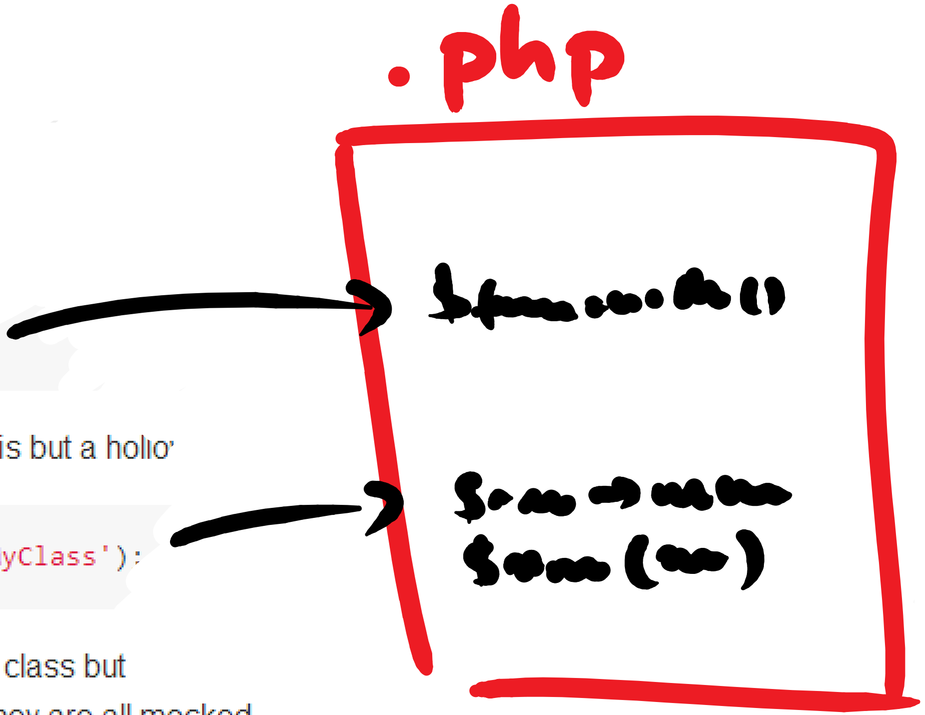
```
$mock = $factory->getInstance('MyClass');
```

The created instance extends the given class but

method call actually reach the parent - they are all mocked.

If you want to call the parent's constructor, pass an array with the constructor arguments. If you don't want to pass any arguments, provide an empty array.

```
$mock = $factory->getInstance('MyClass', array('name' => 'Foo'));
```



Basic Usage

First, we need an instance of Moc
needed).

```
$factory = new MockFactory();
```

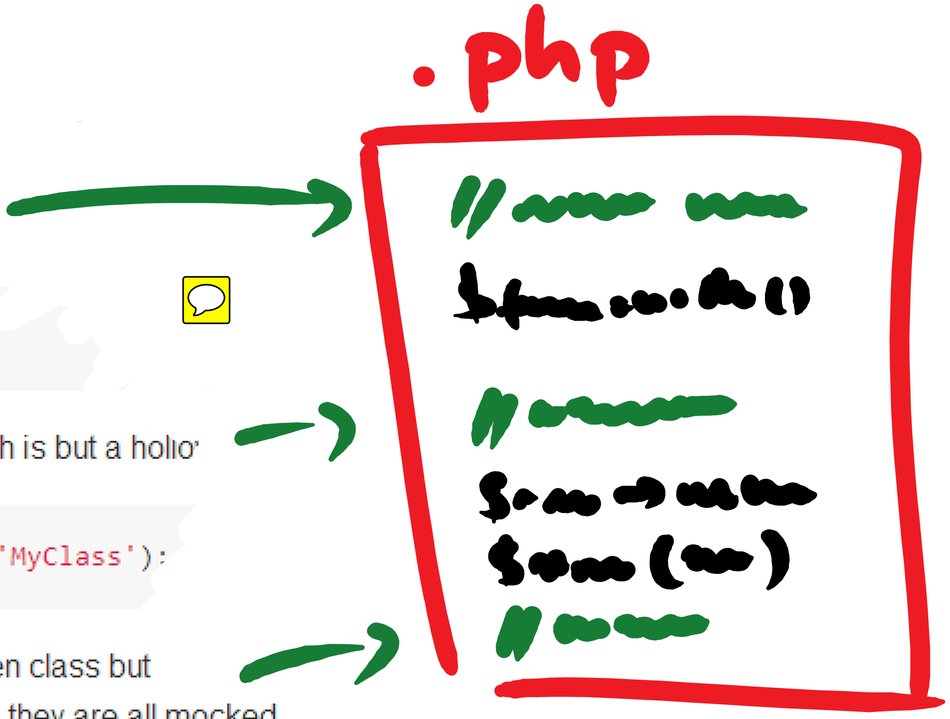
To get a completely empty mock which is but a hollo

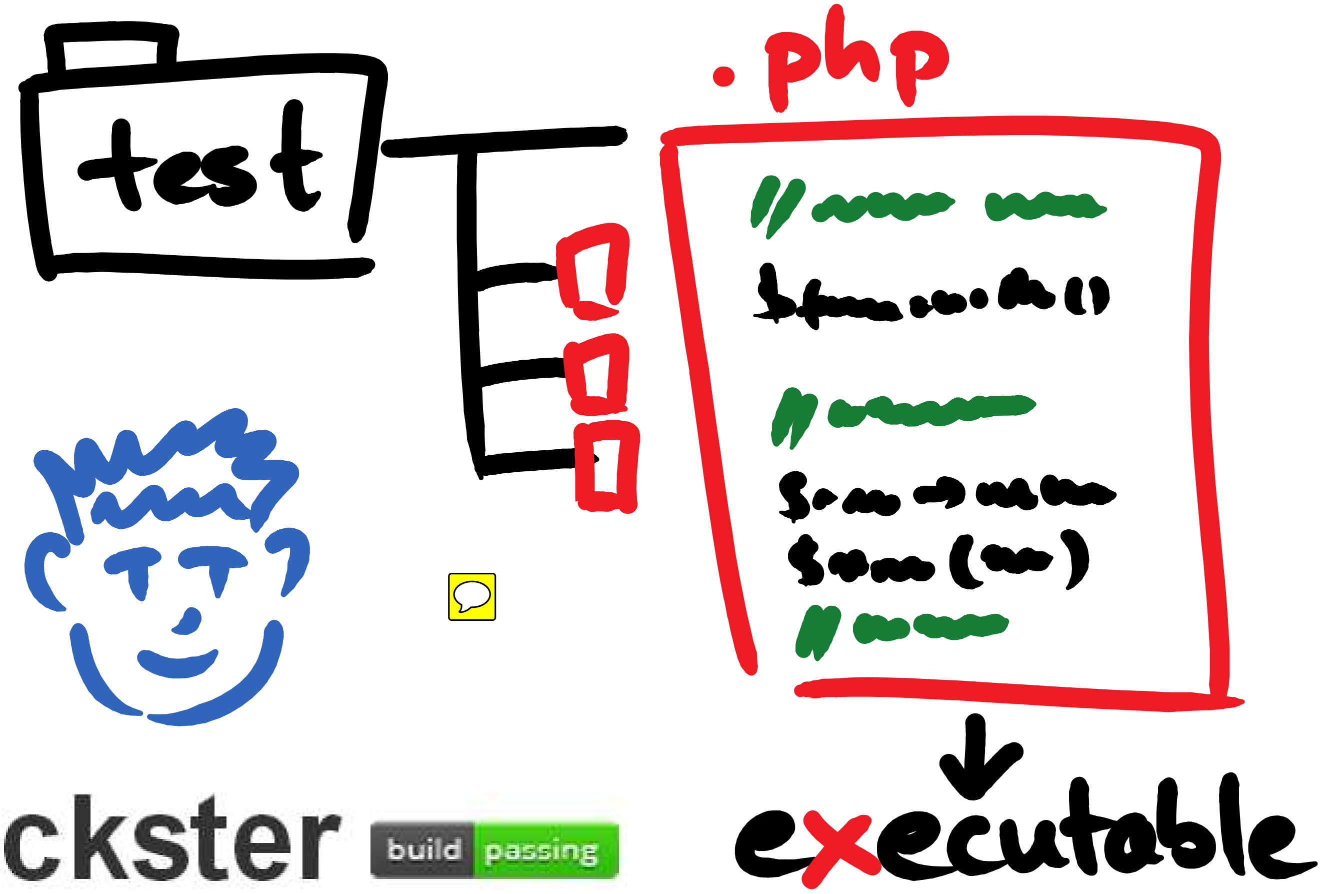
```
$mock = $factory->getInstance('MyClass');
```

The created instance extends the given class but
method call actually reach the parent - they are all mocked.

If you want to call the parent's constructor, pass an array with the constructor arguments. If you don't want to pass any arguments, provide an empty array.

```
$mock = $factory->getInstance('MyClass', array('name' => 'Foo'));
```





Mockster build passing

Validate syntax

```
$mock = $factory->getInstance('MyClass');
$foo = $mock->__mock()->method('foo');
```

But wait!

There is more!

```
$foo->getHistory()->wasCalledWith(['bar']);
```

Validate functionality

```
$mock = $factory->getInstance('MyClass');
$foo = $mock->__mock()->method('foo');

$mock->foo('bar');


$this->assertTrue($foo->getHistory()->wasCalledWith(['bar']));
$this->assertFalse($foo->getHistory()->wasCalledWith(['baz']));
```

Hey! That's a test!

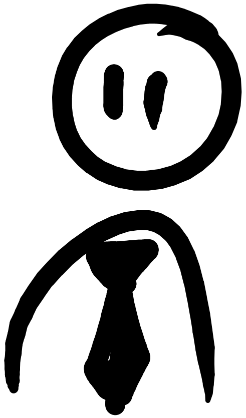
Test $\leftarrow$ Documentation

👍 APIs

👍 Libraries

APIs

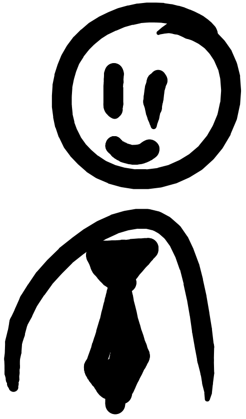
```
$response = $router->respond('/foo/bar', '{"some":"query"}');

$this->assertEquals($response, '{"some":"data"}');
```



APIs

```
$this->whenIRequest_From('{some:"query"}', '/foo/bar');

$this->thenTheResponseShouldBe('{"some":"data"}');
```

ubiquitous language



👍 APIs

👍 Libraries

☆ Anything

Example-Driven Development

Acceptance Test Driven

Behaviour Driven Development

Anything

Agile Testing

Example-Driven Development

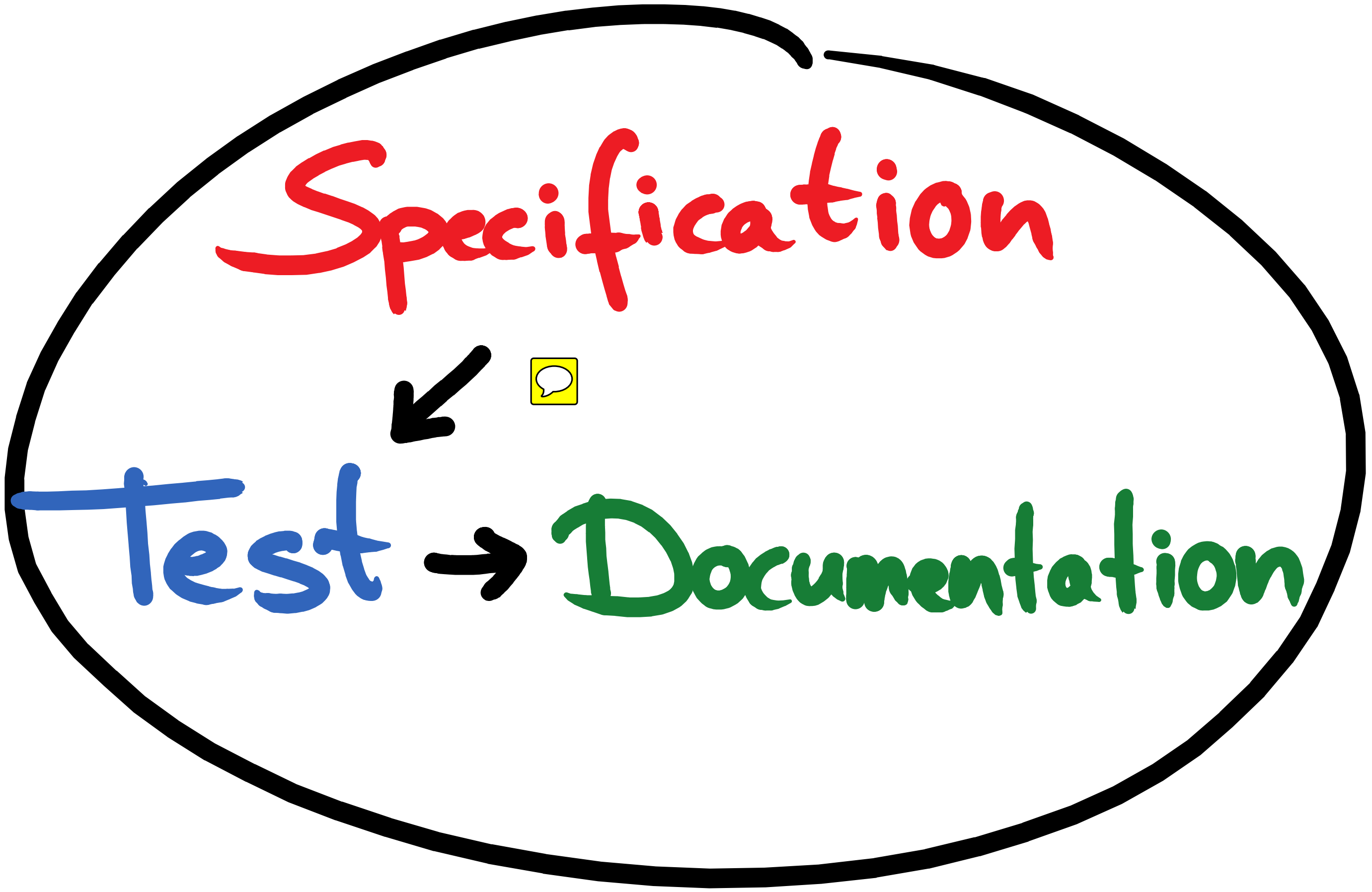
Acceptance Test Driven

Behaviour Driven Development

Specification by Example

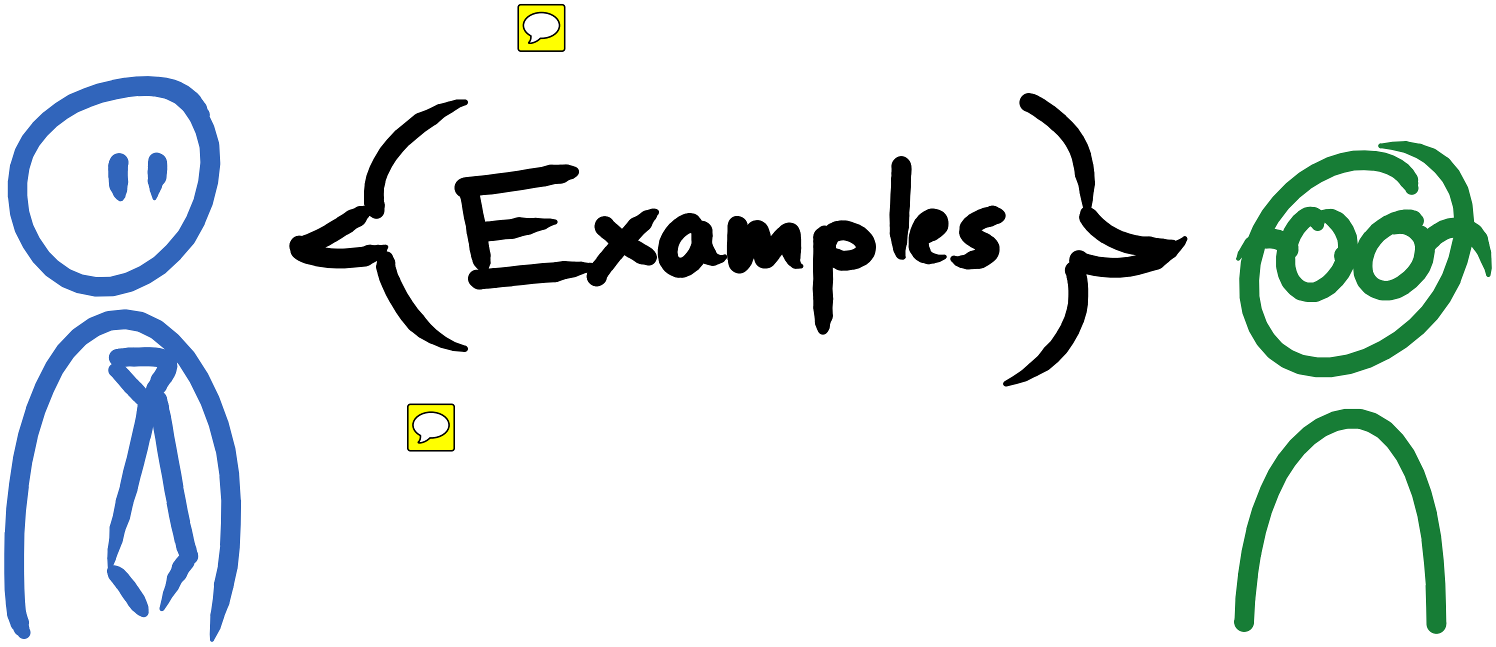
Agile Testing

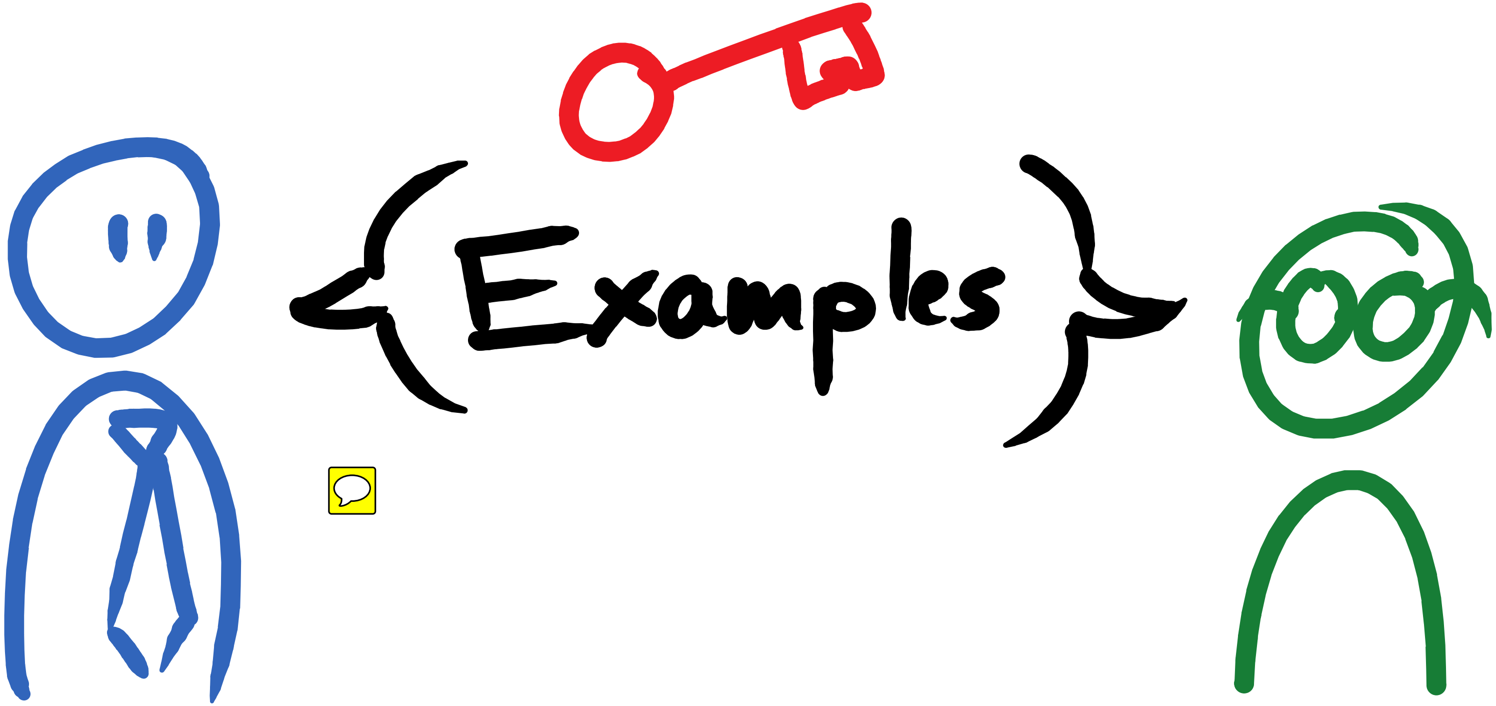
$$\begin{tikzcd}
& \text{Specification} \arrow[dl, leftrightarrow] \arrow[dr, leftrightarrow] & \\
\text{Test} \arrow[rr, leftrightarrow] & & \text{Documentation}
\end{tikzcd}$$

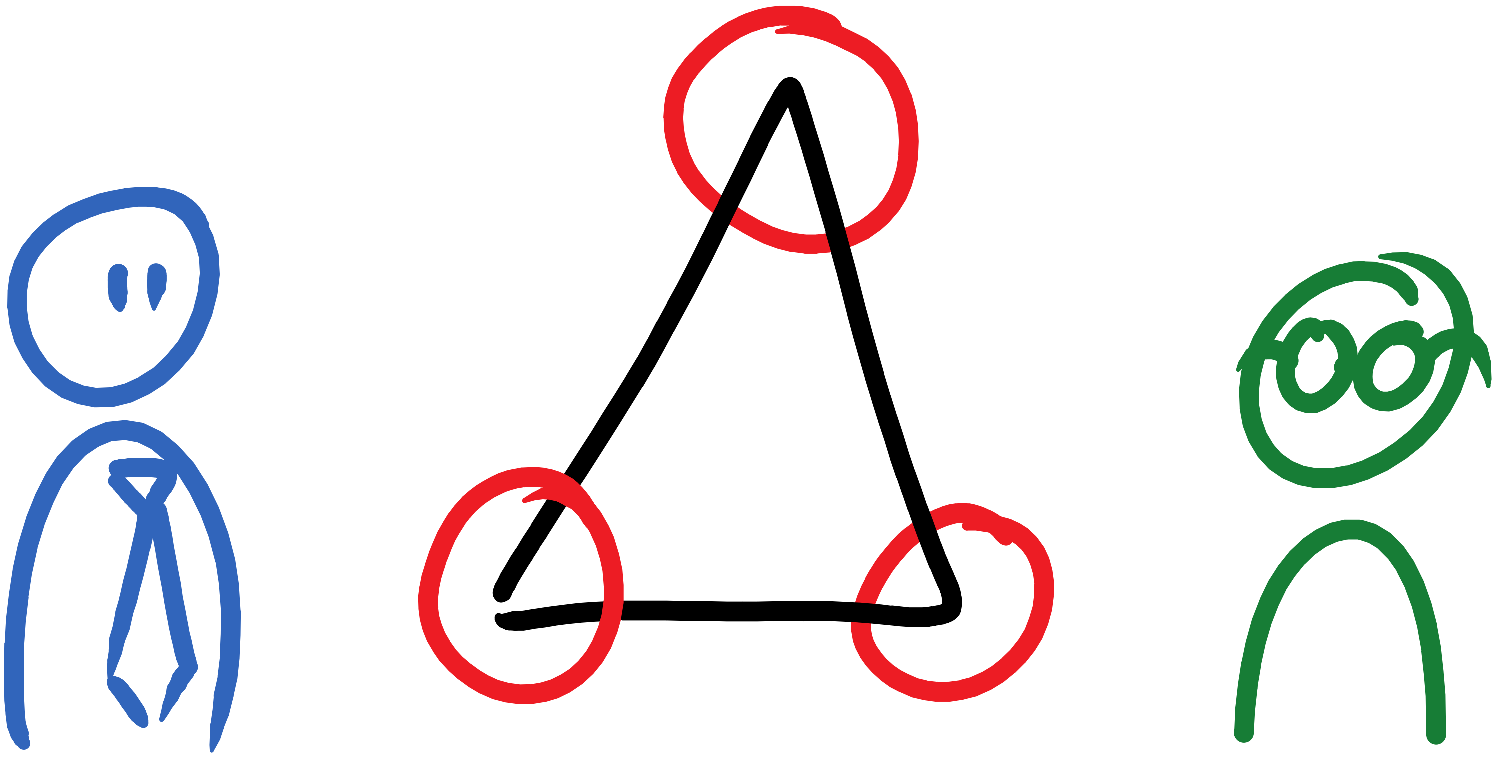

Specification
Test
Documentation

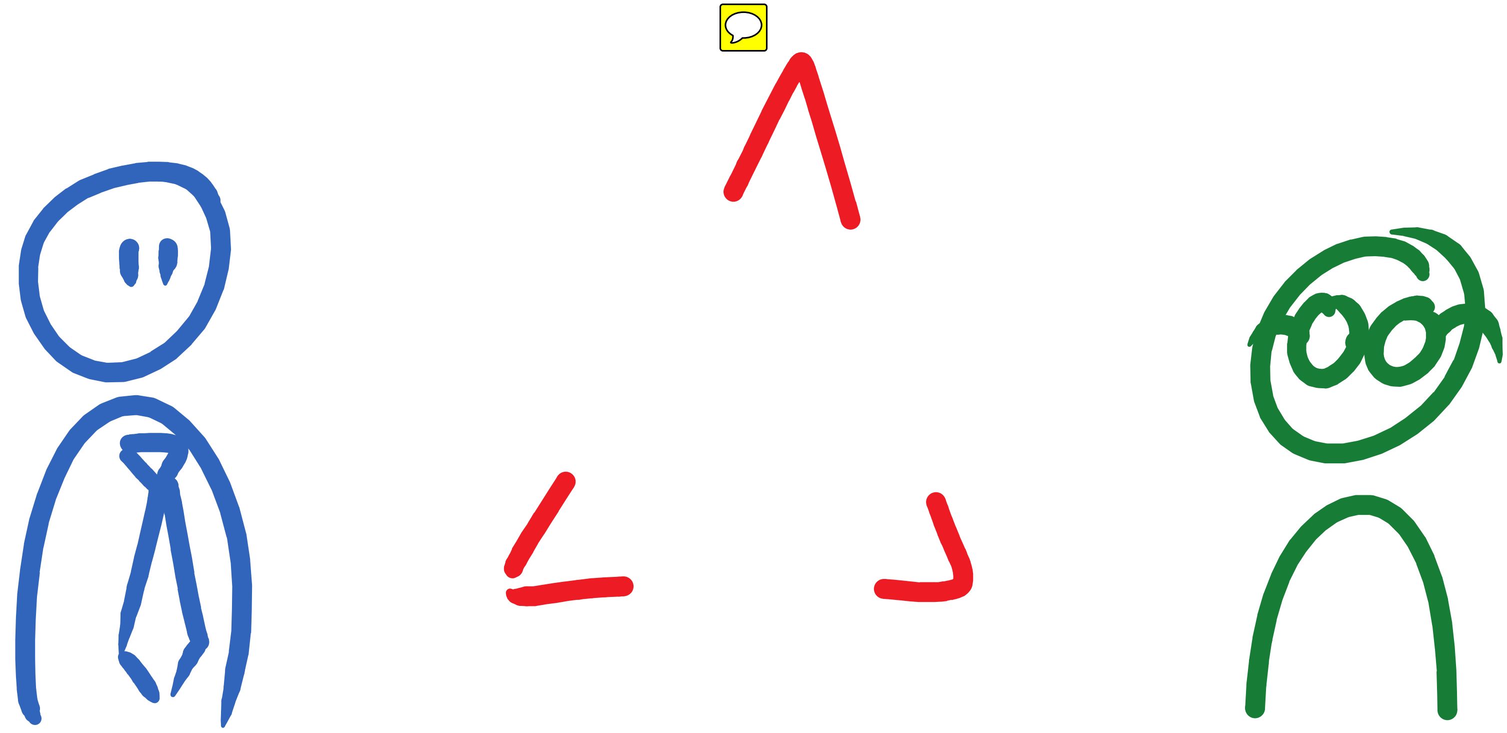
Examples

```
$this->whenIRequest_From('{some:"query"}', '/foo/bar');

$this->thenTheResponseShouldBe('{"some":"data"}');
```

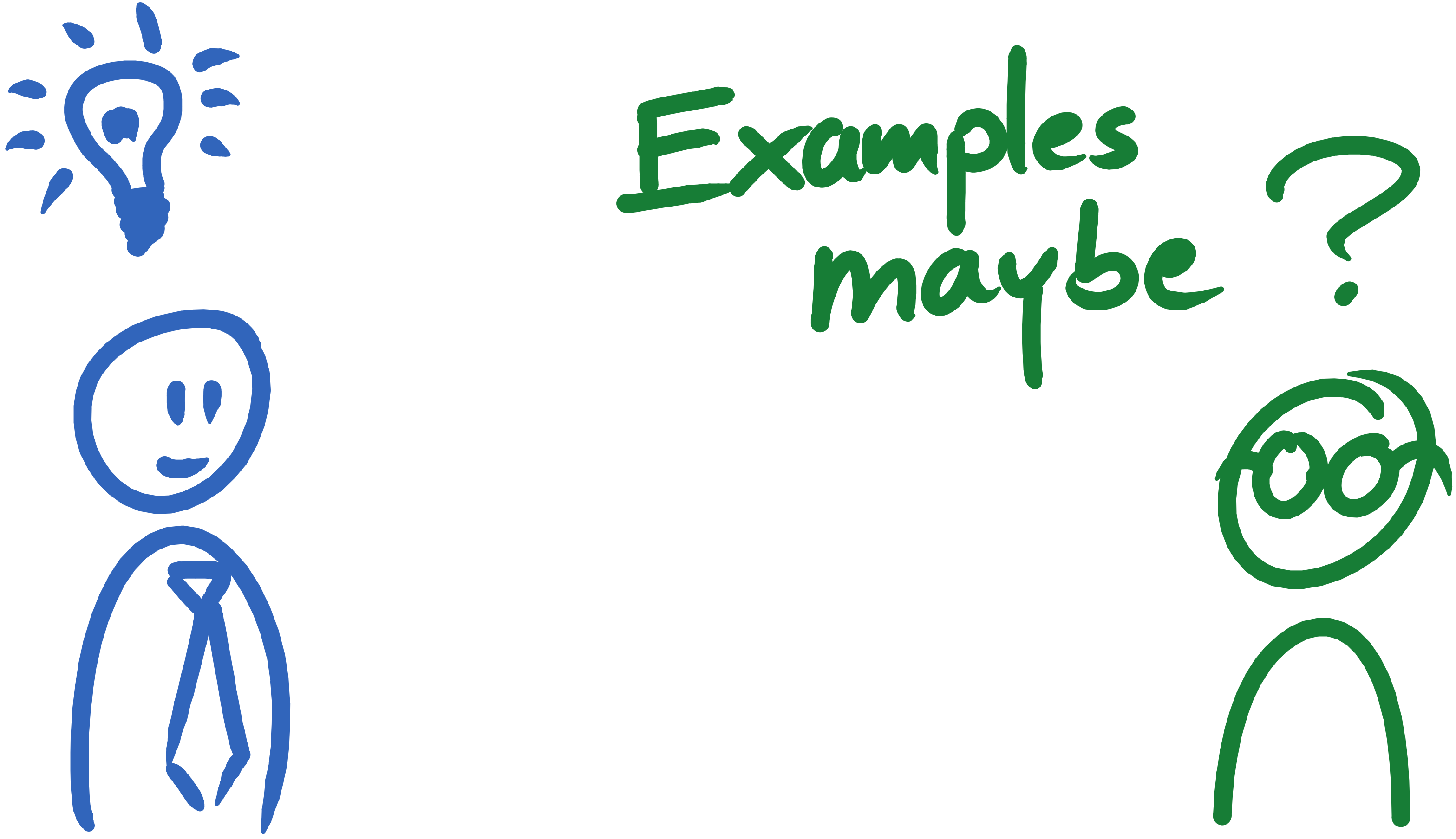


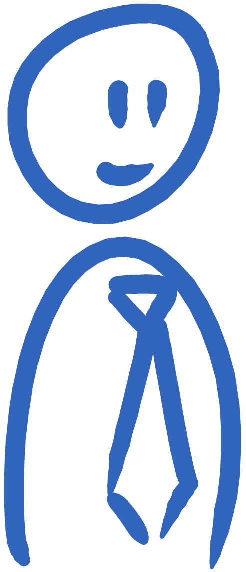

Examples



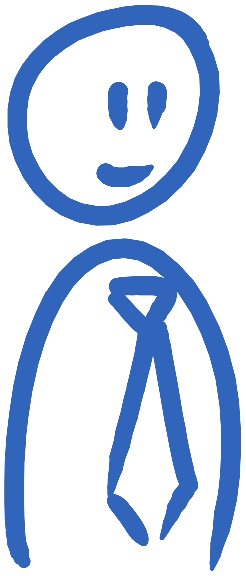


Free delivery
for ViP customers
more than 5 articles
only books


Examples maybe ?



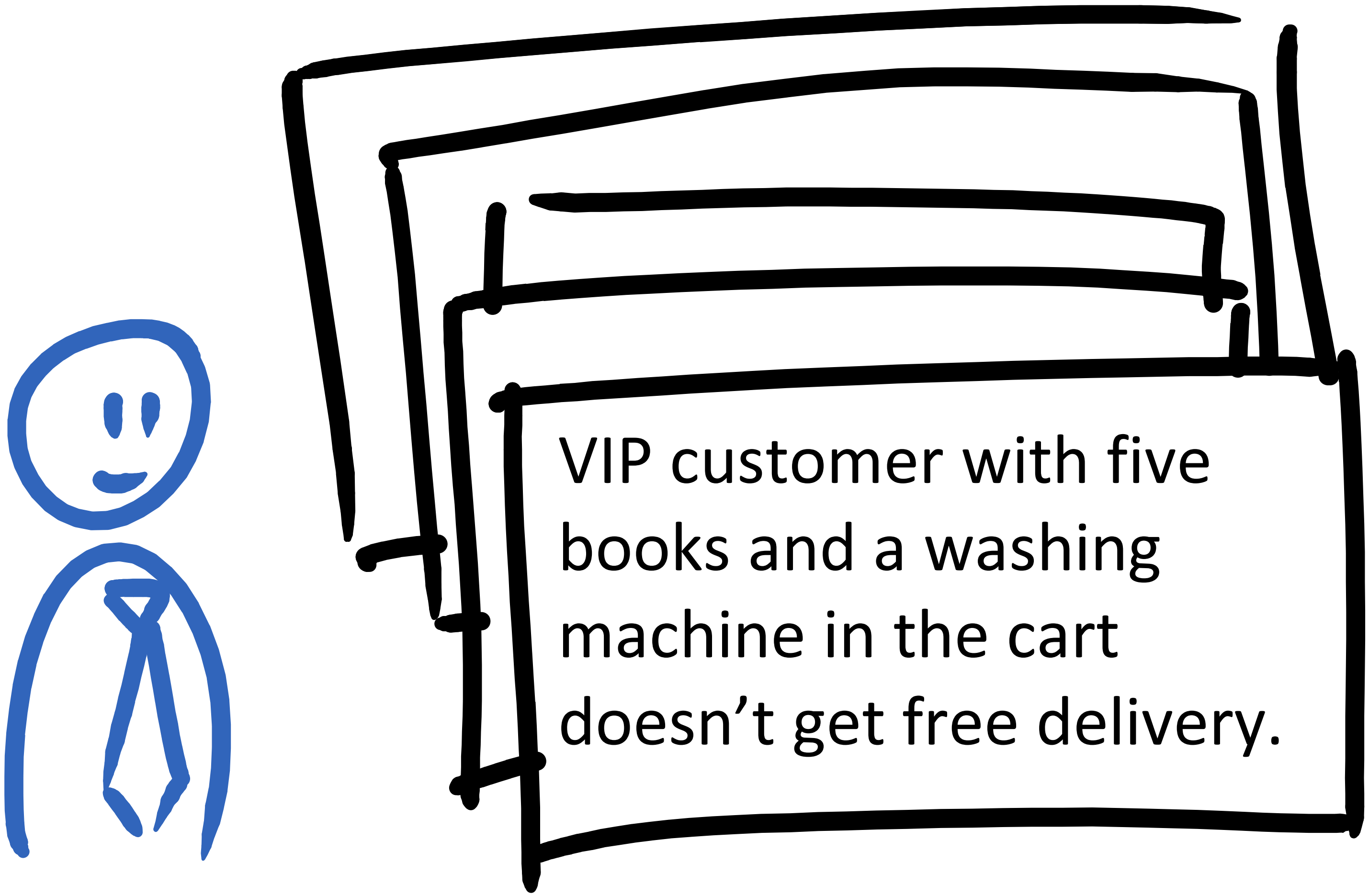
VIP customer with five books in the cart gets free delivery.

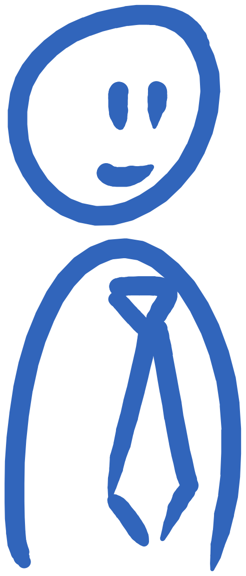


VIP customer with four books in the cart doesn’t get free delivery.

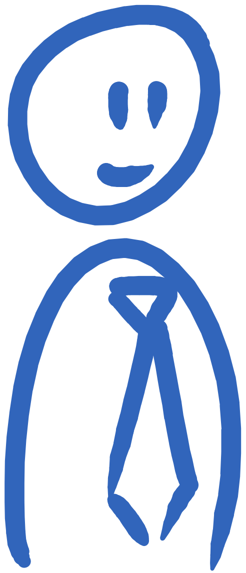
Regular customer with five books in the cart doesn’t get free delivery.

VIP customer with a five washing machines in the cart doesn’t get free delivery.


VIP customer with five books and a washing machine in the cart doesn’t get free delivery.

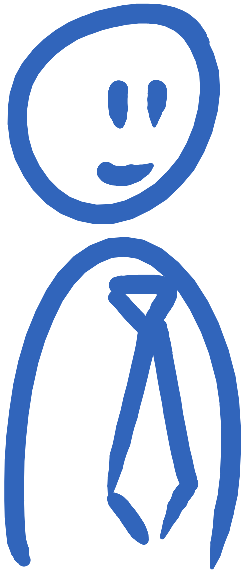


Input		Output
Customer	Order	Delivery



Customer	Order	Delivery
VIP	5 books	free
VIP	4 books	not
Reg	10 books	not
VIP	5 TV	not
VIP	5 books & 1 TV	not

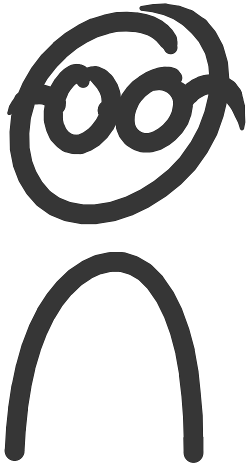
Customer	Order	Delivery
VIP	5 books	free



```
$this->givenIAmAVipCustomer();
$this->givenIHave_BooksInMyBasket(5);

$this->whenICheckMyDeliveryOptions();

$this->thenTheDilveryShouldBeFree();
```

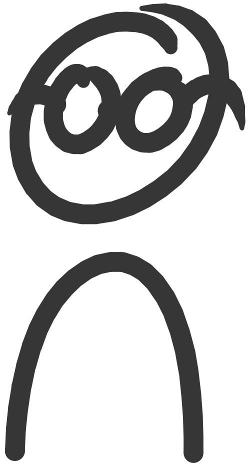
```
function givenIAmAVipCustomer() {
    $this->customer = new Customer();
    $this->customer->setVip(true);
}

function givenIHave_BooksInMyBasket($number) {
    $this->basket = new Basket();
    for ($i=0; $i<$number; $i++) {
        $this->basket->getItems()->put(new Book());
    }
}

function whenICheckMyDeliveryOptions() {


}

function thenTheDeliveryShouldBeFree() {

}
```

Fake it!

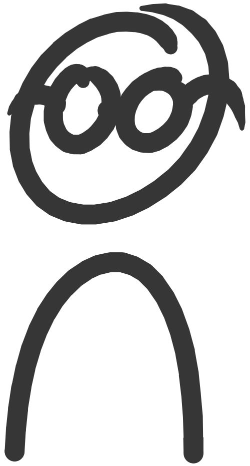


```
function givenIAmAVipCustomer() {
    $this->customer = new Customer();
    $this->customer->setVip(true);
}

function givenIHave_BooksInMyBasket($number) {
    $this->basket = new Basket();
    for ($i=0; $i<$number; $i++) {
        $this->basket->getItems()->put(new Book());
    }
}

function whenICheckMyDeliveryOptions() {
    $delivery = new DeliveryManager($this->customer, $this->basket);
    $this->isFree = $delivery->isDeliveryFree();
}

function thenTheDeliveryShouldBeFree() {

}
```

Do it!



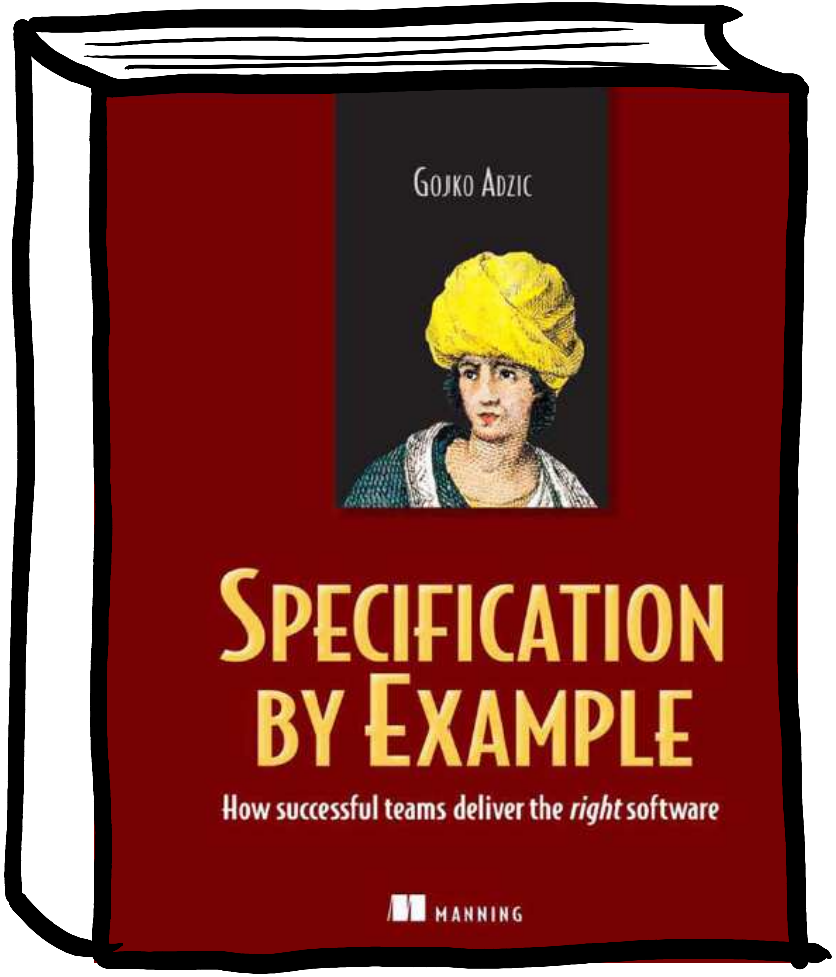
```
function givenIAmAVipCustomer() {
    $this->customer = new Customer();
    $this->customer->setVip(true);
}

function givenIHave_BooksInMyBasket($number) {
    $this->basket = new Basket();
    for ($i=0; $i<$number; $i++) {
        $this->basket->getItems()->put(new Book());
    }
}

function whenICheckMyDeliveryOptions() {
    $delivery = new DeliveryManager($this->customer, $this->basket);
    $this->isFree = $delivery->isDeliveryFree();
}

function thenTheDeliveryShouldBeFree() {
    $this->assertTrue($this->isFree);
}
```

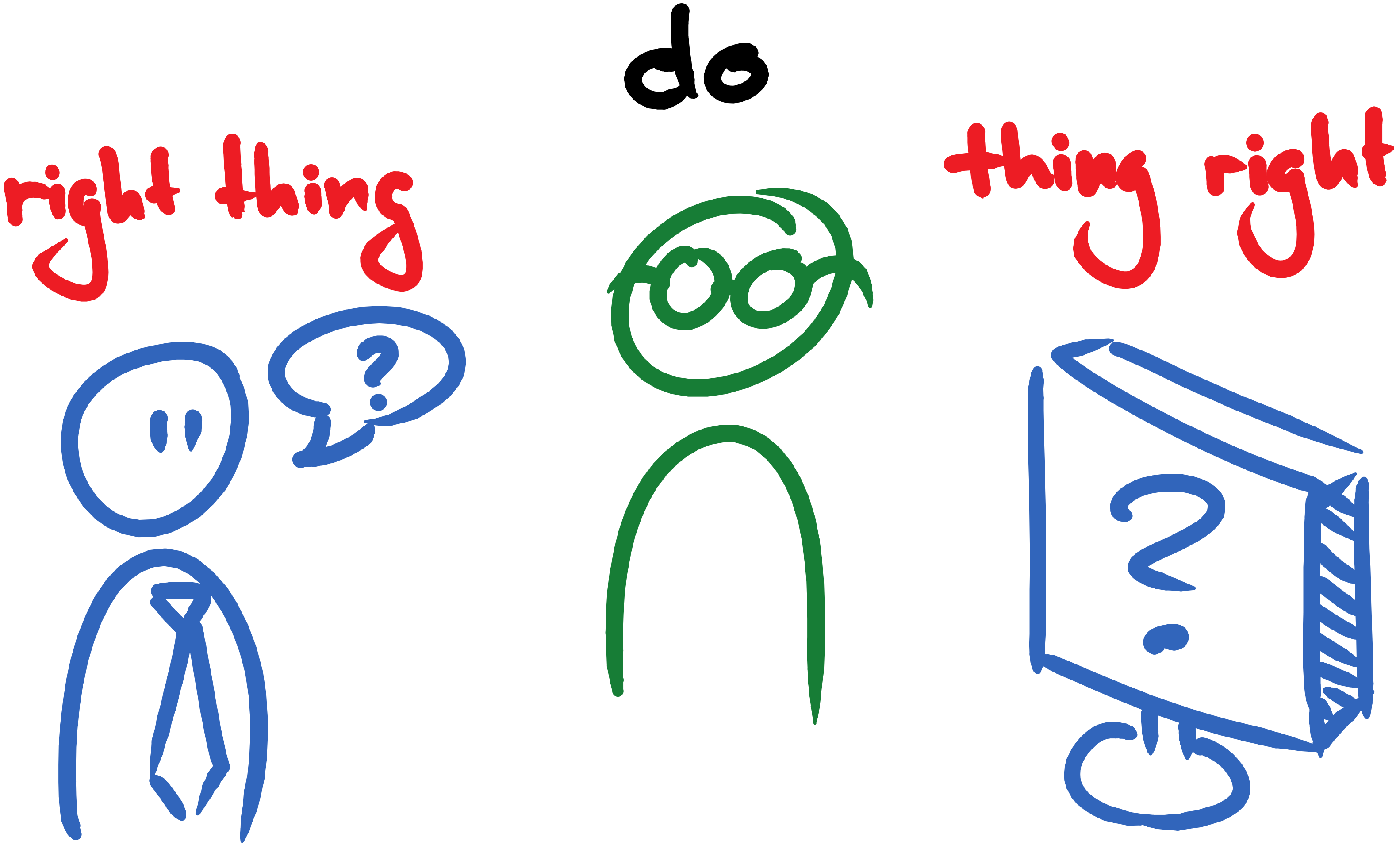


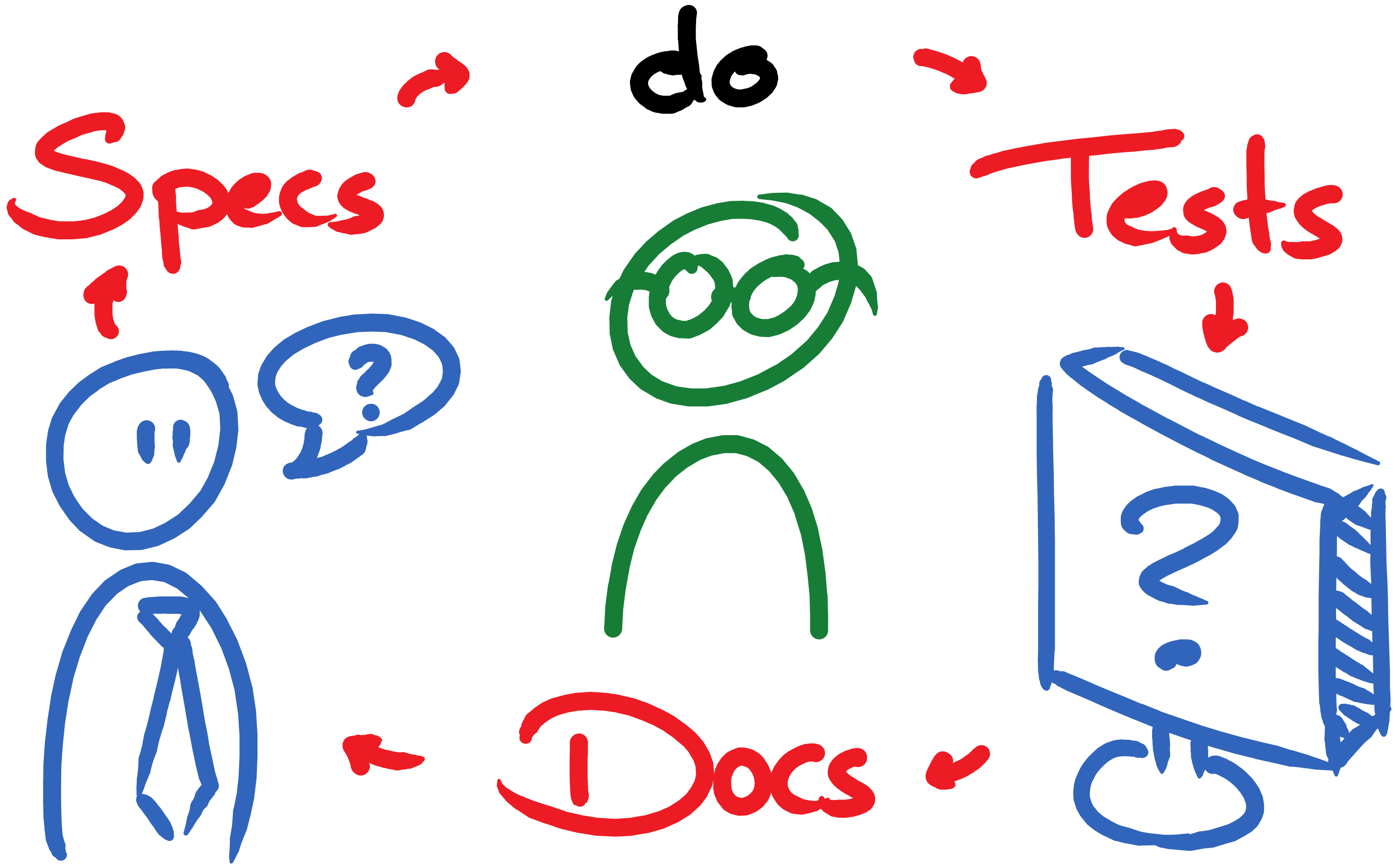
Check it!

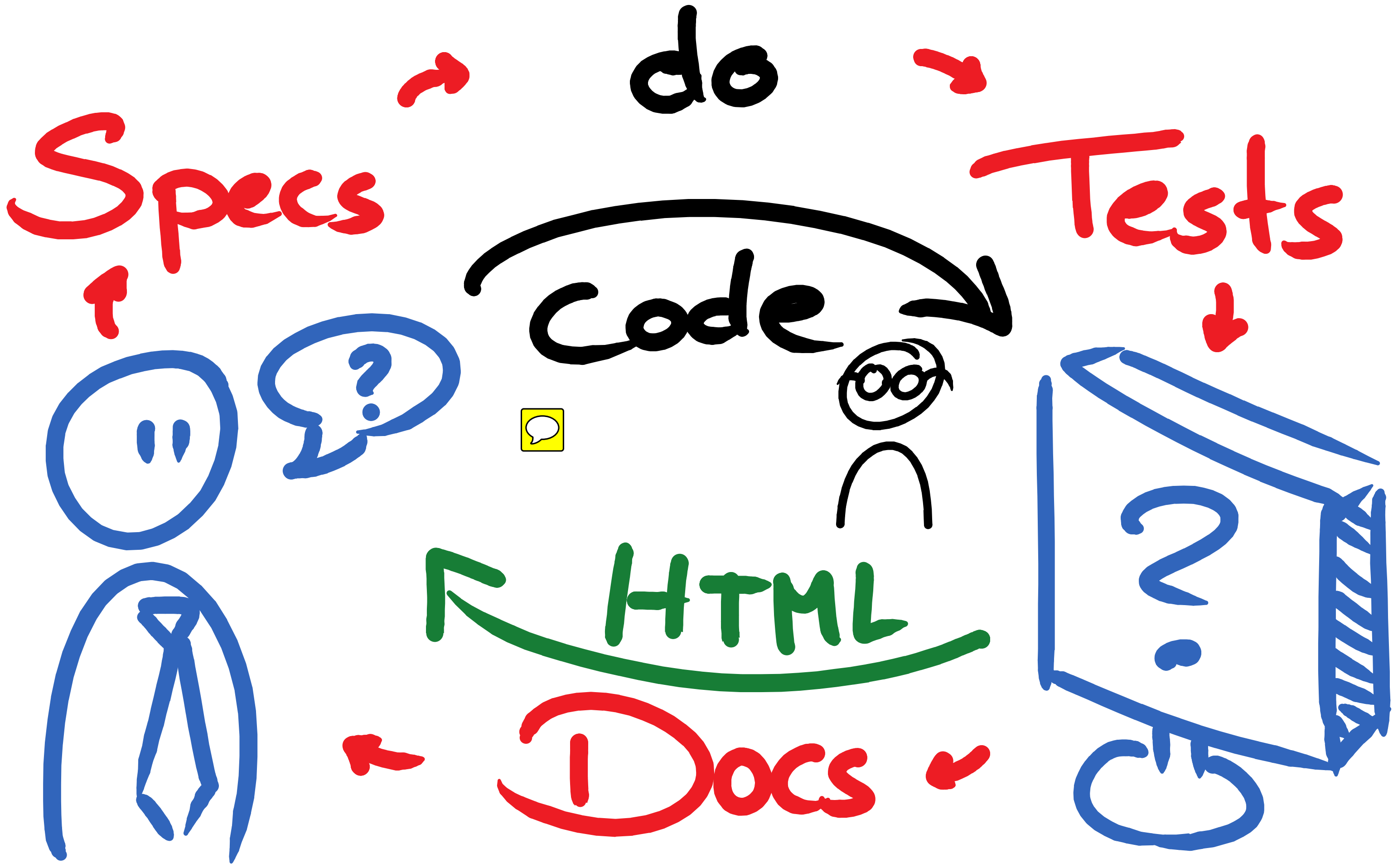
Specification by

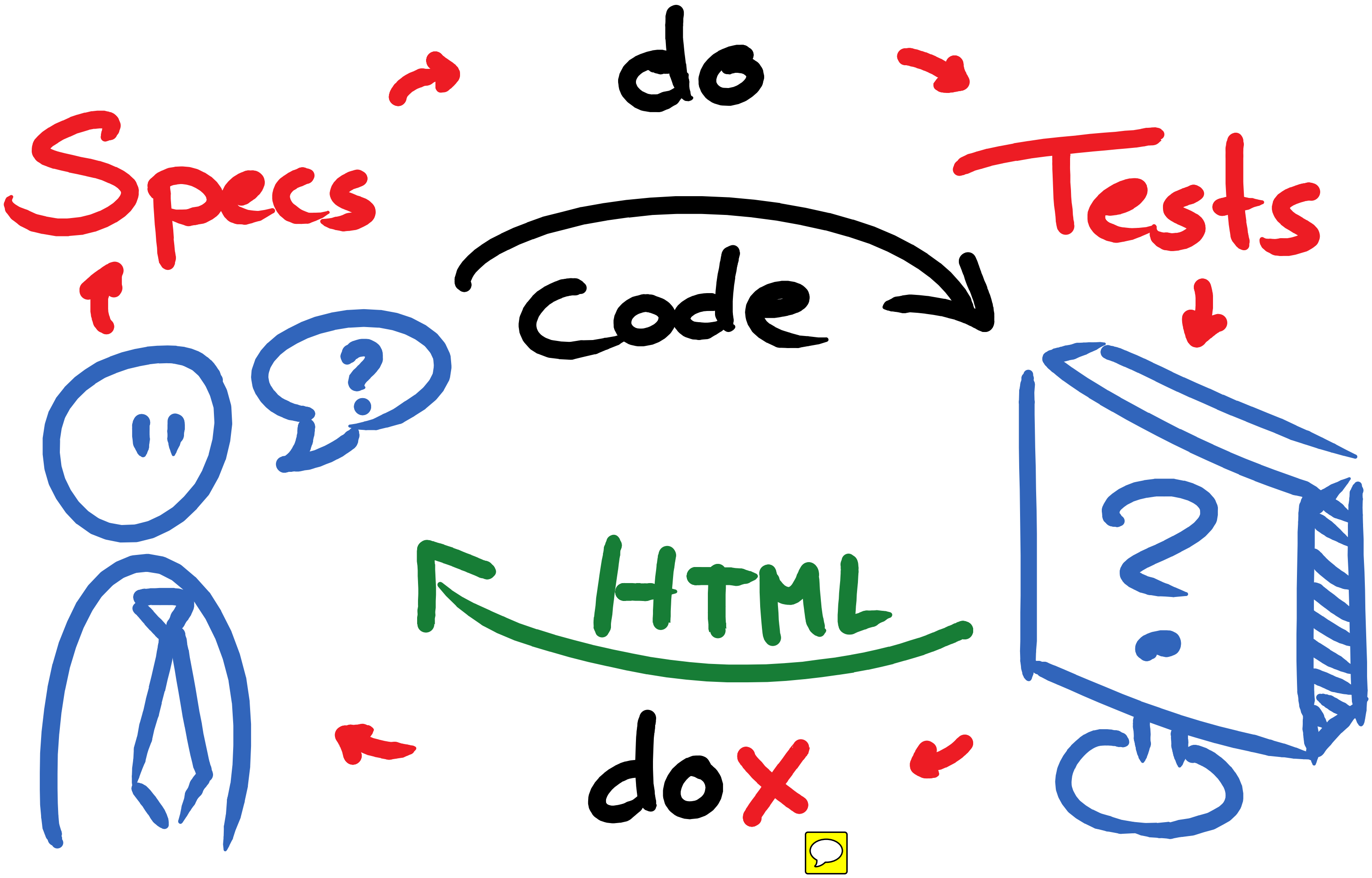
Example

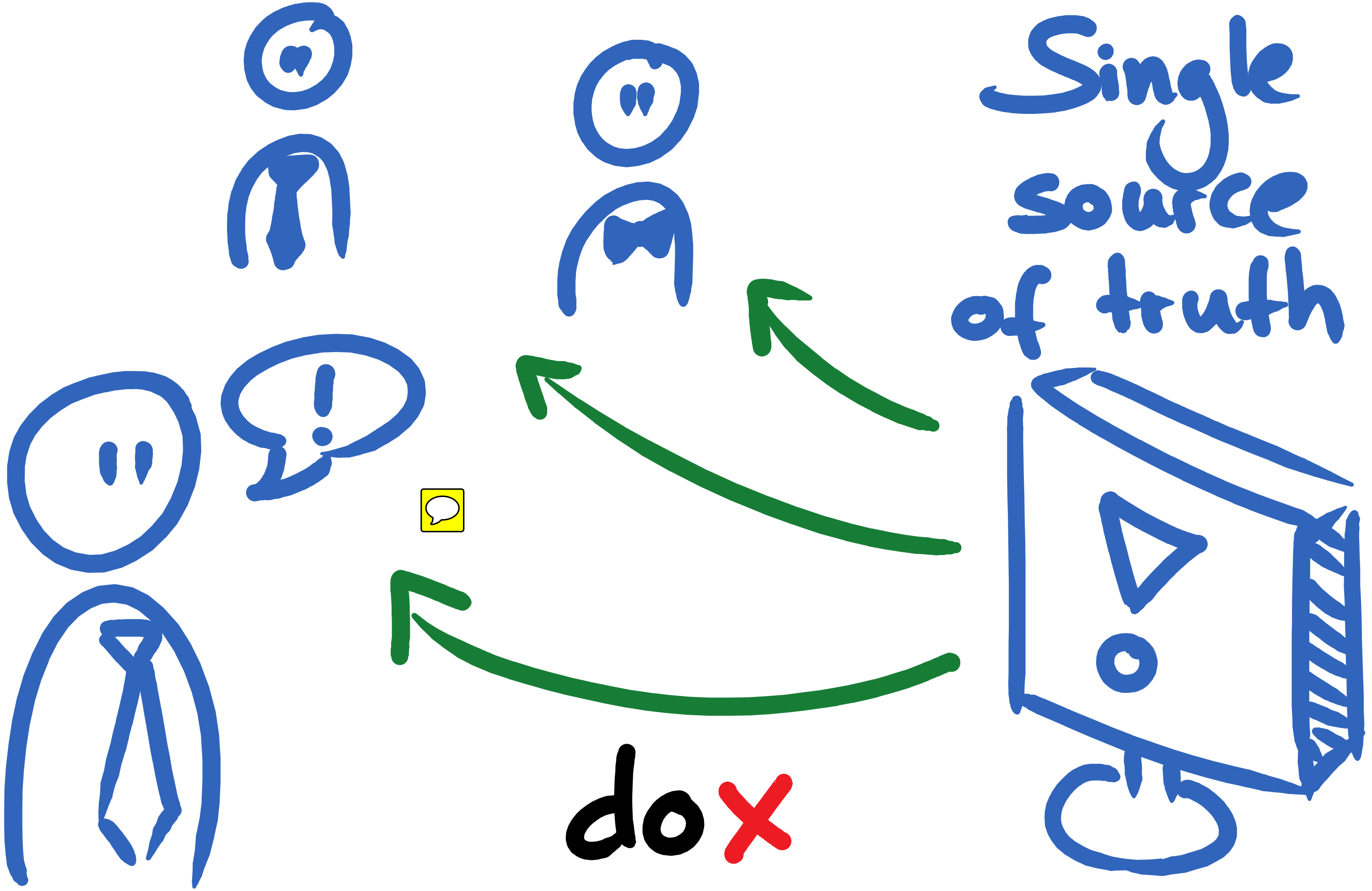

GOJKO ADZIC
SPECIFICATION BY EXAMPLE
How successful teams deliver the *right* software
MANNING

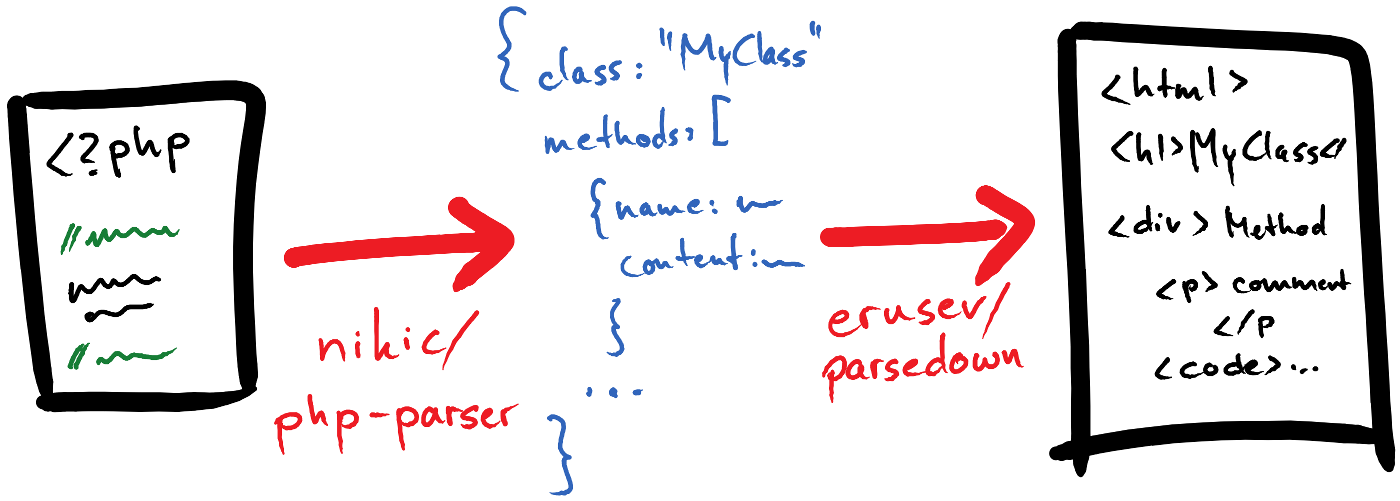

right thing
do
thing right


do
Specs
Tests
Docs
?
?


Specs
do
Tests
Code
HTML
Docs
?
?

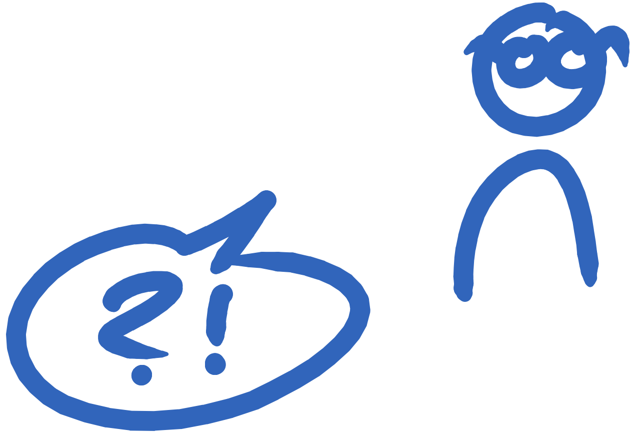

do
Specs
Tests
Code
HTML
dox
?
?


Single source of truth
!
dox

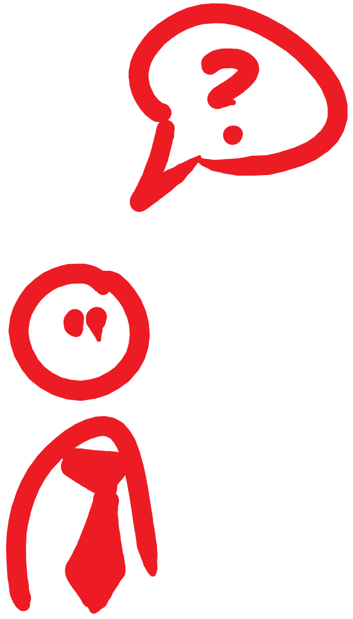


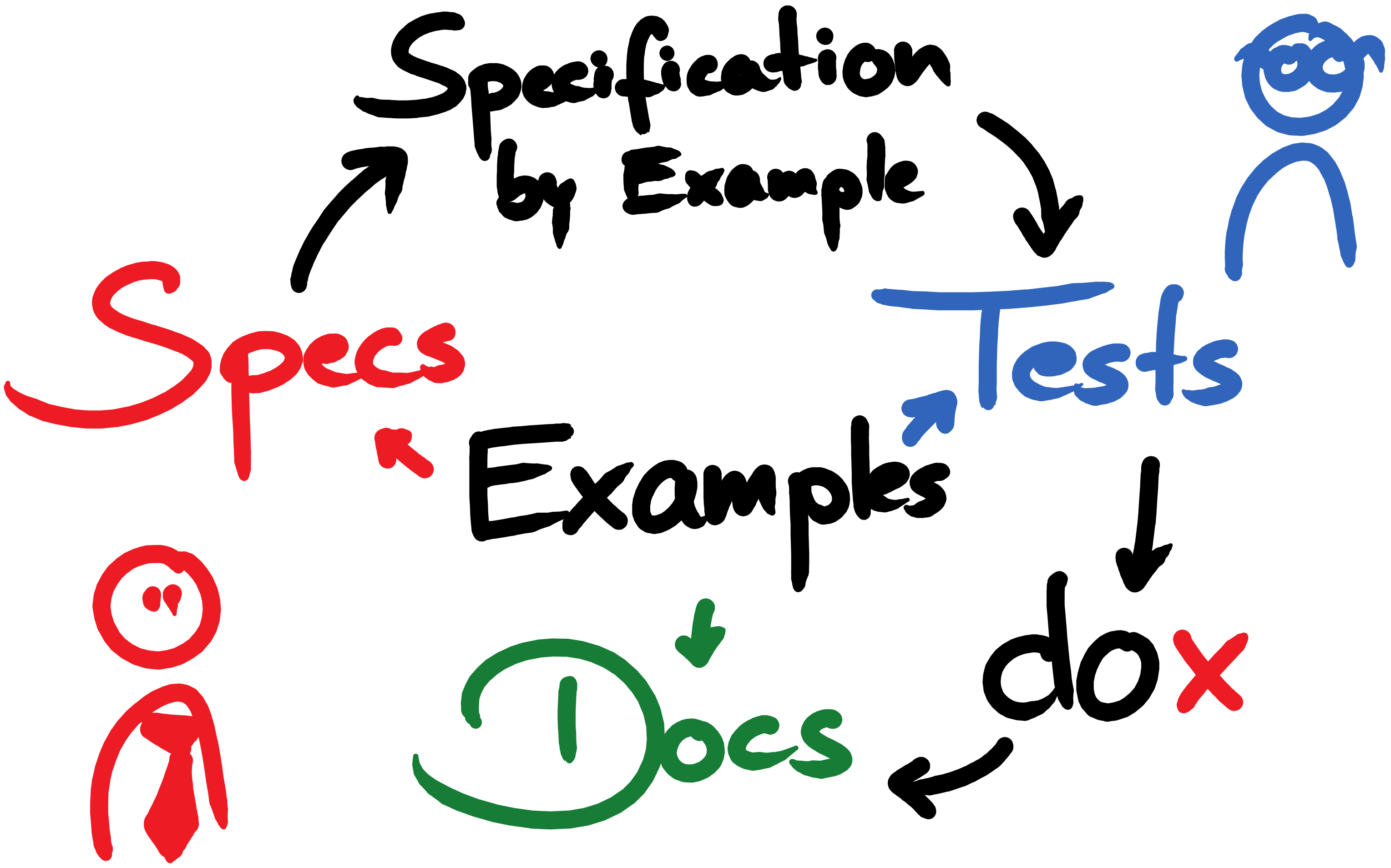
dox pipeline

once again...



Examples




Specification by Example
Specs
Tests
Examples
Docs
dox

more on SbE

http://specificationbyexample.com/ ← the book

http://dannorth.net/introducing-bdd/ ← the beginning

http://skillsmatter.com/podcast/agile-testing/how-to-sell-bdd-to-the-business ← good talk

more good stuff

read the book

http://www.clean-code-developer.de/

root of agile

http://www.extremeprogramming.org/

http://theleanstartup.com/

stop waste

dox.rtens.org