

Designing Interactions with Mocks



Xavier Shay

<http://xavershay.com> | @xshay

Behavior

not

Implementation

Problems



Coupled to implementation

Extraneous details

Fragile

Unit Test

Do our objects do the right thing, and are they convenient to work with?

Integration Test

Does our code work against objects we can't change?

Acceptance Test

Does the whole system work?



Code

drove

Mocks

Mocks

drive

Code

Mocks

are for

Design

Using Test Doubles

Extracting a deck, `instance_double`

Improvements



Code is cleaner

Spec is cleaner

Mocks were a scaffold

Listen

to your

Mocks

Verifying Doubles



Once-off test objects

Protect against interface changes

Working with Mocks



Not mechanical, judgement required
Continual design pressure
Suggests refactorings
Ongoing process

Recap

Expect and allow

Provide alternate implementations for methods.

Verifying Doubles

`instance_double`, `as_null_object`

Mocks for Design

Scaffolding, refactoring.

3rd Party Libraries

can have

Weird APIs

3rd Party Libraries

cannot be

Refactored

Don't mock classes
you don't own

Unit Test

Do our objects do the right thing, and are they convenient to work with?

Integration Test

Does our code work against objects we can't change?

Acceptance Test

Does the whole system work?

Recap

Extracted Banker class

Untangled logic

Moved towards a unit spec

Is our object convenient to work with?

Deep Dive

Mocks under the hood

Mocking is complicated



Replacing existing methods
Dealing with method visibility
... and more!

Isolating specs

Dual-mode doubles and constant mutation

Verifying Double with String

Dependency not loaded

- Don't check interfaces.
- Acts like a non-verifying double.

Dependency loaded

- Check interfaces
- Same as when using non-string variant.

Recap

Dual-mode verifying doubles

`instance_double` or `class_double` with a string parameter

Constant mutation

`as_stubbed_const`

Recap

Designing Interactions with Mocks

Mock Features

Expect and allow

Provide alternate implementations for methods.

Verifying Doubles

`instance_double`, `as_null_object`

Mocks

are for

Design

Design Tension

All-in-one

- Easier to understand quickly.
- Harder to make safe changes.

Abstracted

- More time to initial understanding.
- Easier to make safe changes.

Next Up...

Understanding the RSpec Ecosystem
