



*Piriorities in development*

*Application Testing*

*TDD principles and guidelines*

*TDD in action*

# Test Driven Development

*with Laravel*



Roberto Gallea

*Manuel*



*Ted*



*Their  
systems*

# *Manuel*



*his  
workflow*

*consequences*

# *Manuel Manual testing*



*his  
workflow*

*consequences*

## *Manuel's workflow*

Write  
code

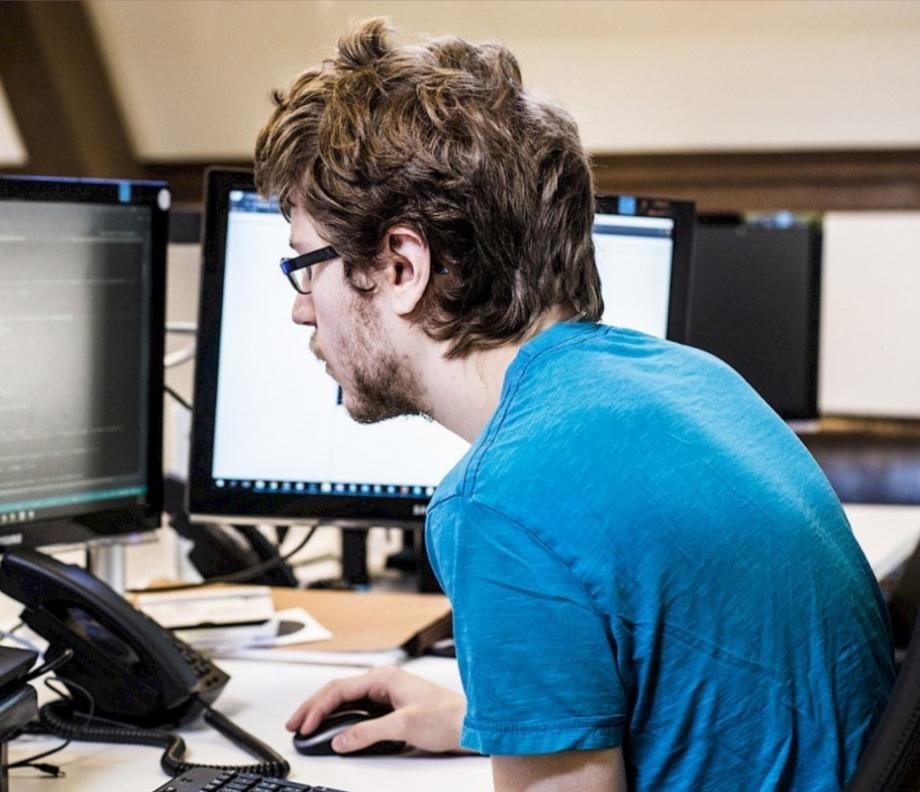
Manual  
test

Modify  
code





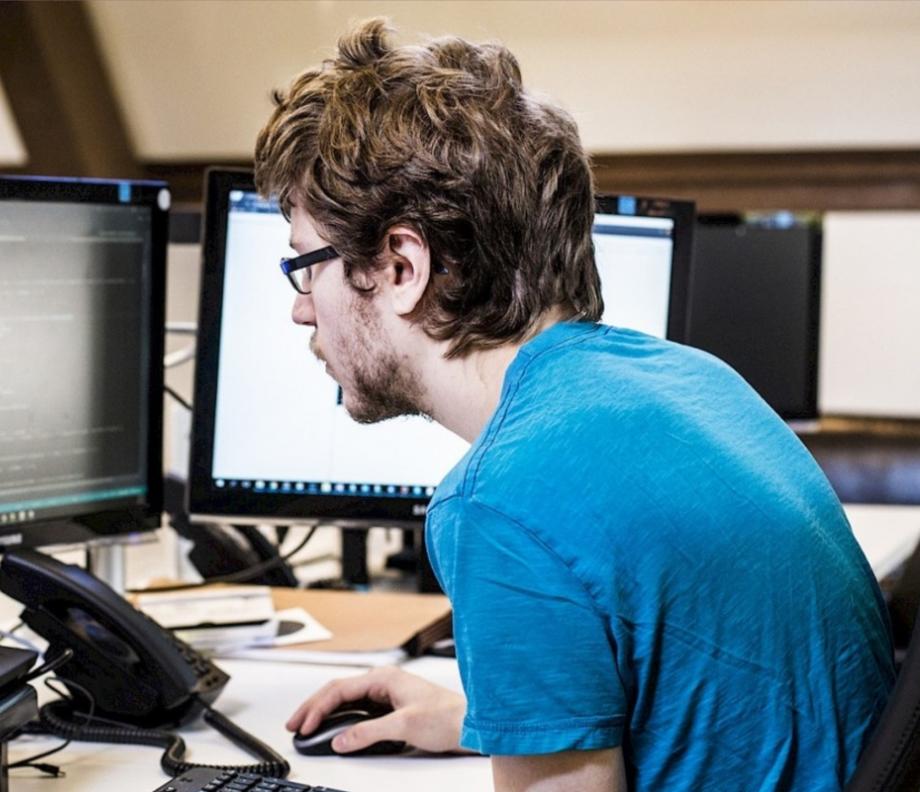
*ted*



*his  
workflow*

*consequences*

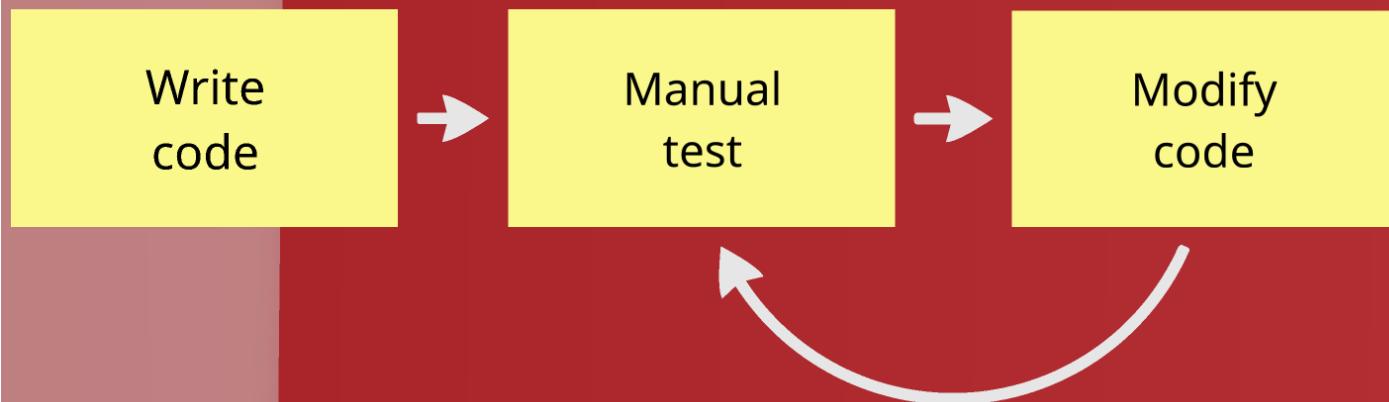
## *Automated testing*



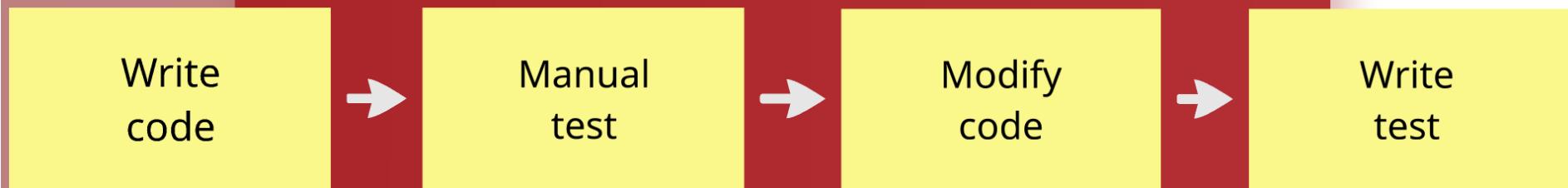
*his  
workflow*

*consequences*

## *Ted's workflow*



## *Ted's workflow*



Modify  
code



Write  
test

Modify  
code



Write  
test



Untestable code!

Modify  
code



Write  
test

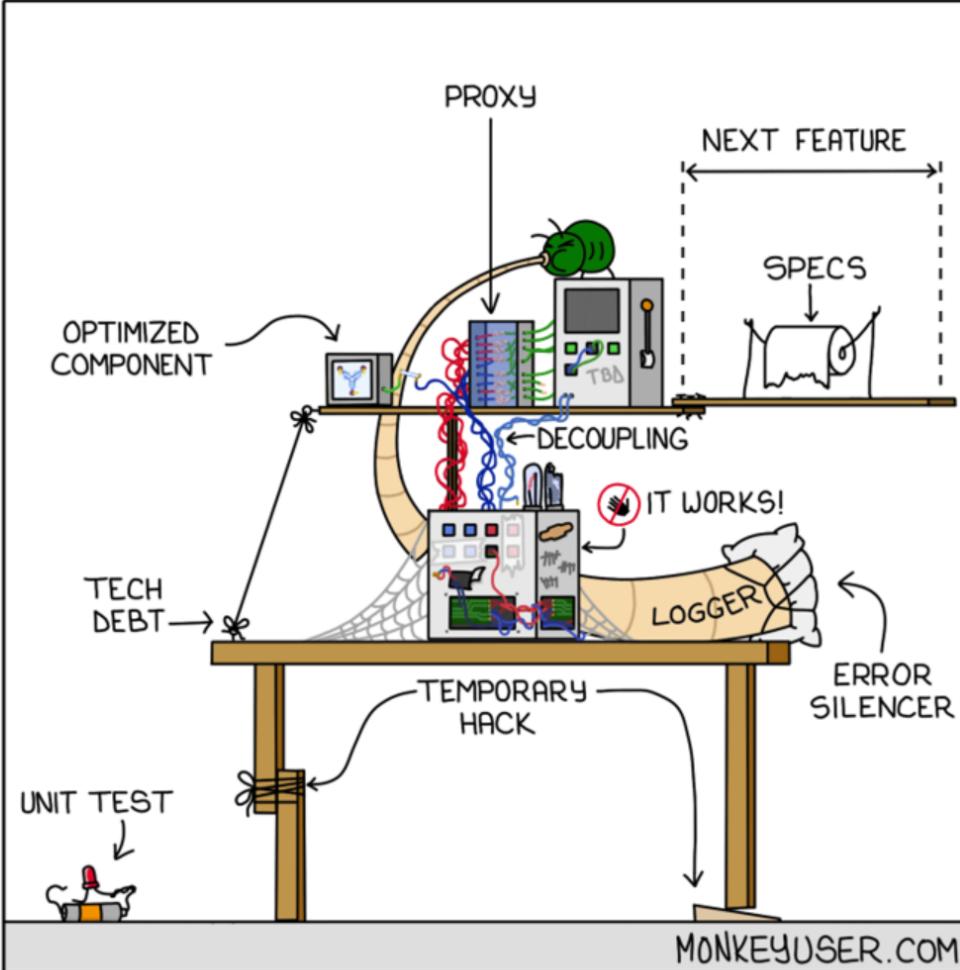


Untestable code!



# Their systems

v 2.0.1





*Piriorities in development*

*Application Testing*

*TDD principles and guidelines*

*TDD in action*

# Test Driven Development

*with Laravel*



Roberto Gallea

# *Priorities in development*

*Make it  
work*

*Make it  
maintainable*

*Make it  
testable*

# *Priorities in development*



*Make it  
work*

*Make it  
maintainable*

*Make it  
testable*



*Make it  
testable*

# *Priorities in development*



*Make it work*

*Make it maintainable*

*Make it testable*

# *Priorities in development*

*Make it  
work*

*Make it  
maintainable*

*Make it  
testable*

# *Priorities in development*





*Piriorities in development*

*Application Testing*

*TDD principles and guidelines*

*TDD in action*

# Test Driven Development

*with Laravel*



Roberto Gallea

# *TDD principles and guidelines*

*a step  
further*

*tdd  
workflow*

*principles  
and  
guidelines*

Write  
code

Manual  
test

Modify  
code

Write  
test

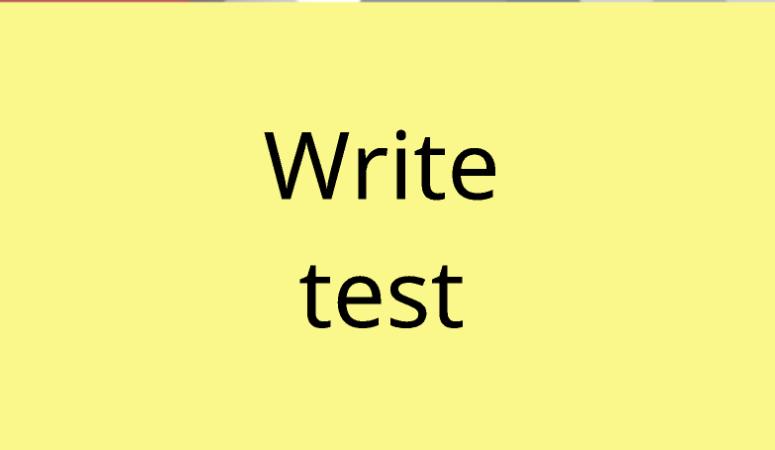


Modify  
code

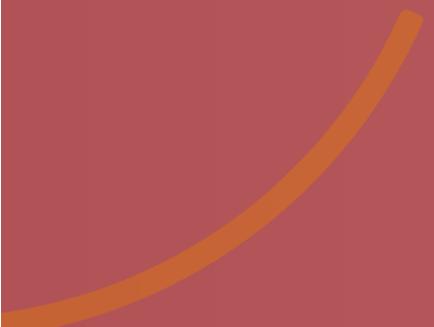


Write  
test





Write  
test



Write or  
Modify  
code

Write  
test



# ***TDD workflow***

## ***TDD workflow***

Write failing  
test

## ***TDD workflow***

Write failing  
test



Make test  
pass

## ***TDD workflow***

Write failing  
test

Make test  
pass

Refactor

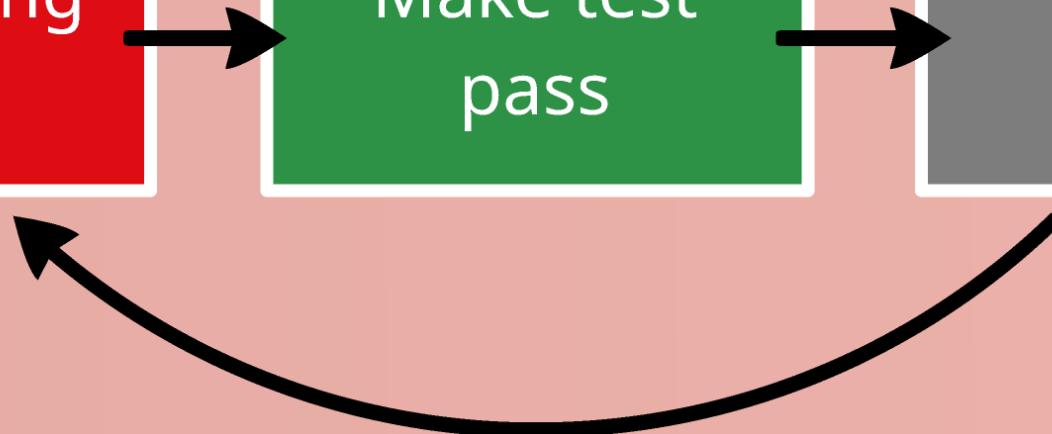


## ***TDD workflow***

Write failing  
test

Make test  
pass

Refactor



# TDD

*principles*

*guidelines*

# *Three simple principles*

1

2

3

# 1

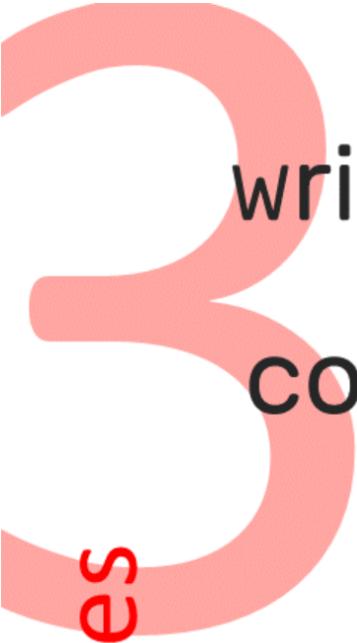
Write production code only to  
make a failing unit test pass.

Don't write source code until a test requires it

## 2 Write only enough of a unit test to fail.

Aim to have just one thing broken at a time if possible

Reference new classes, methods, variables etc in your test  
first



write only enough production  
code to make the failing unit  
test pass.

Anticipate nothing,  
optimize only after all test passing

*«We are what we  
repeatedly do.  
Excellence then is not an  
act, but a habit.»*

--Aristotle

#1

#2

#3

#4

#5

#6

#7

#8

Things to remember

1

Focus on the requirements

## Things to remember

2 Create a mental map and a list of required tests

Things to remember

3

Make the test pass

Things to remember

4

Write self-explanatory tests

Things to remember

Tests are clients!

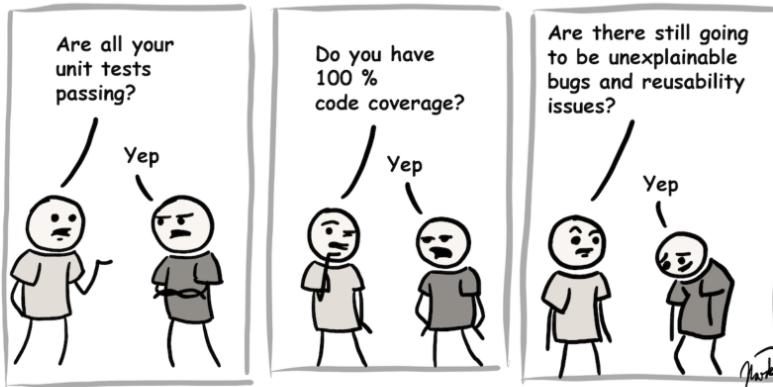
6

# Refactor often

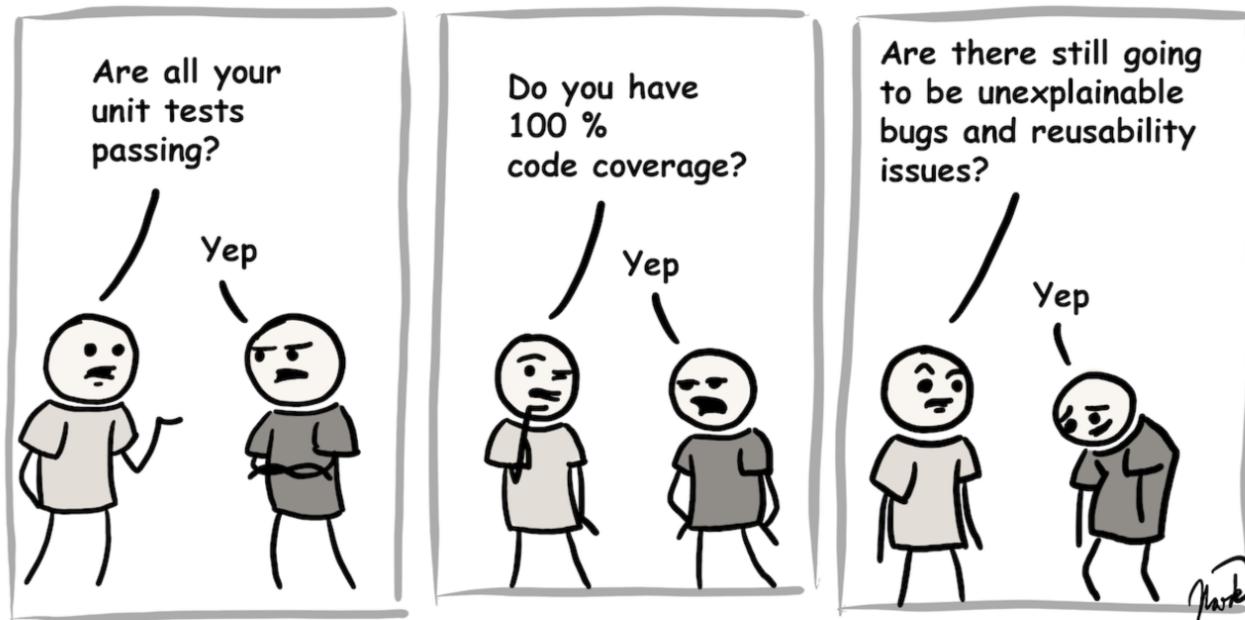
*both production and test code*

# Things to remember

## 7 Tests are not proofs



# Tests are not proofs



Things to remember

Don't drift from discipline



*Piriorities in development*

*Application Testing*

*TDD principles and guidelines*

*TDD in action*

# Test Driven Development

*with Laravel*



Roberto Gallea

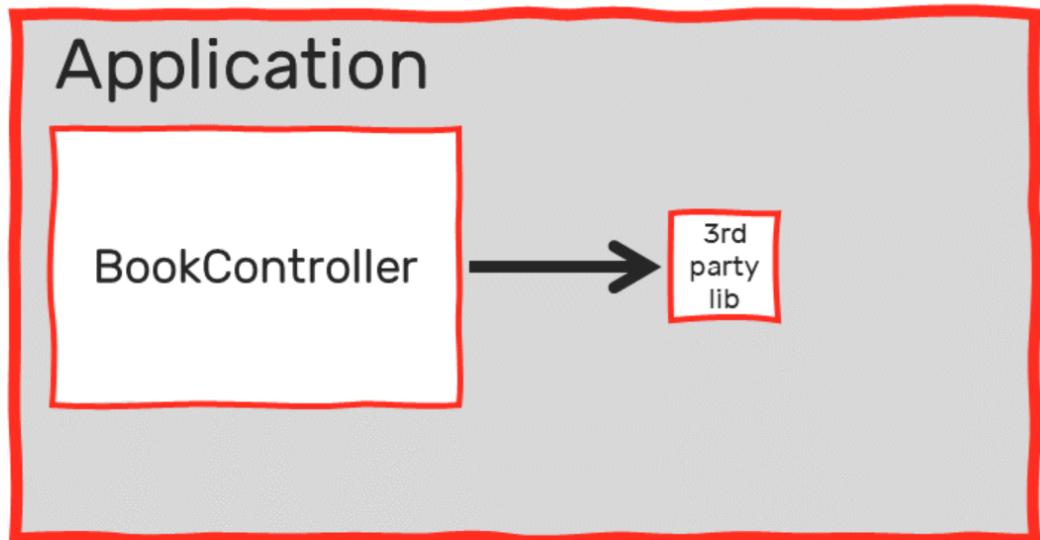
## *Just a few features*

1. Users can create books
2. Users are notified on book creation
3. Books can be imported by isbn from external sources

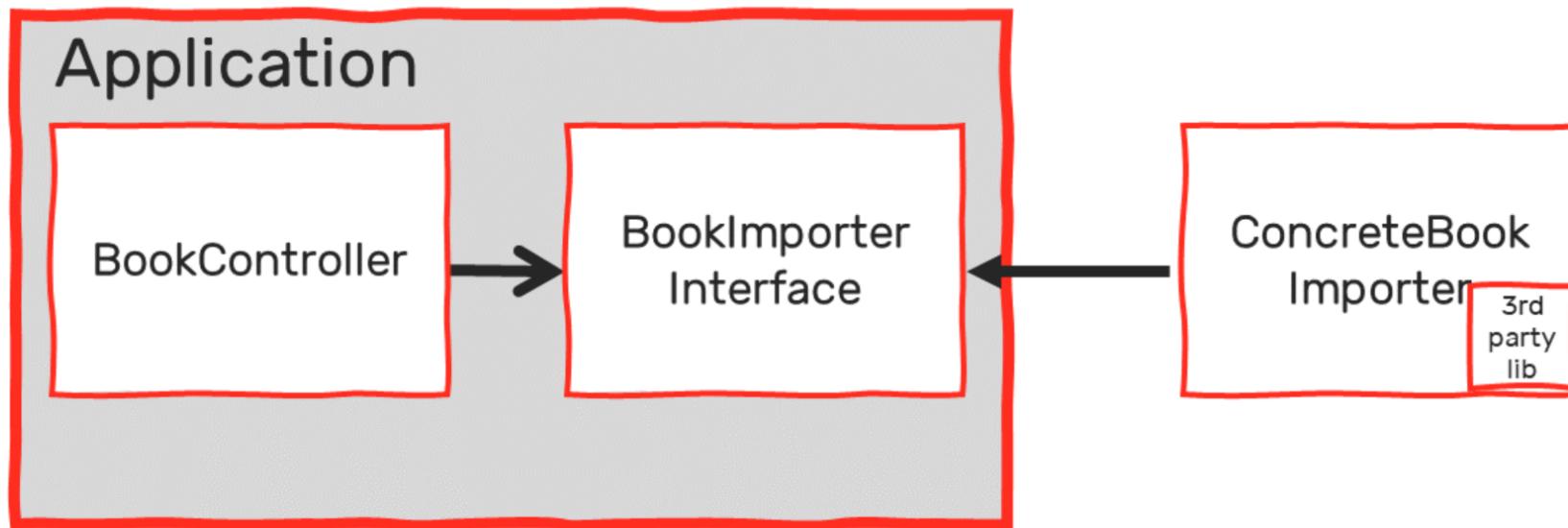
Users are notified on book creation

Books can be imported by isbn from  
external sources

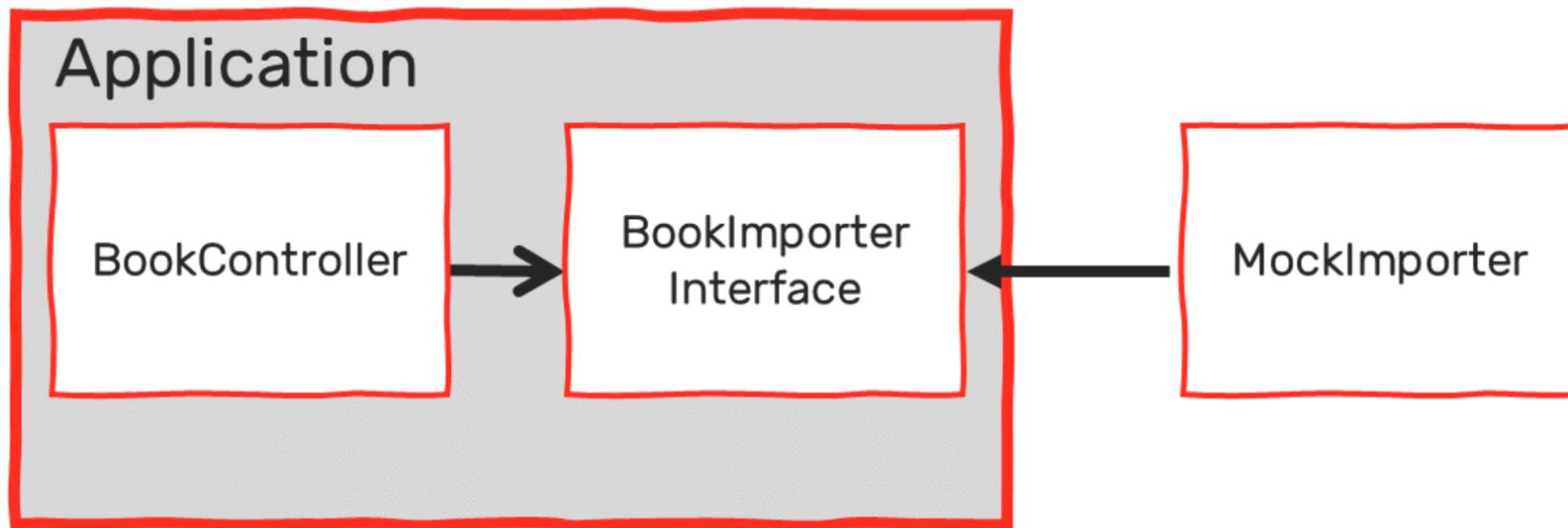
# Messy leads to untestable



# Testable leads to clean code



# Testable leads to clean code





*Piriorities in development*

*Application Testing*

*TDD principles and guidelines*

*TDD in action*

# Test Driven Development

*with Laravel*



Roberto Gallea