



Mental hacks to master Unit Testing

Agenda

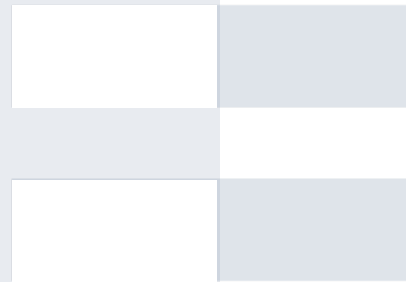
- The vicious circle
- Break the loop
- Getting the time
- Mental Hacks (*Using the time wisely*)



The vicious circle

What today
is...

No time to
unit test



...is
Tomorrow

Crazy regressions
& unexpected bugs

What today
is...


...is
Tomorrow

Lack of
practice

No meaningful
unit tests



Break the loop



Getting the time



Selling points

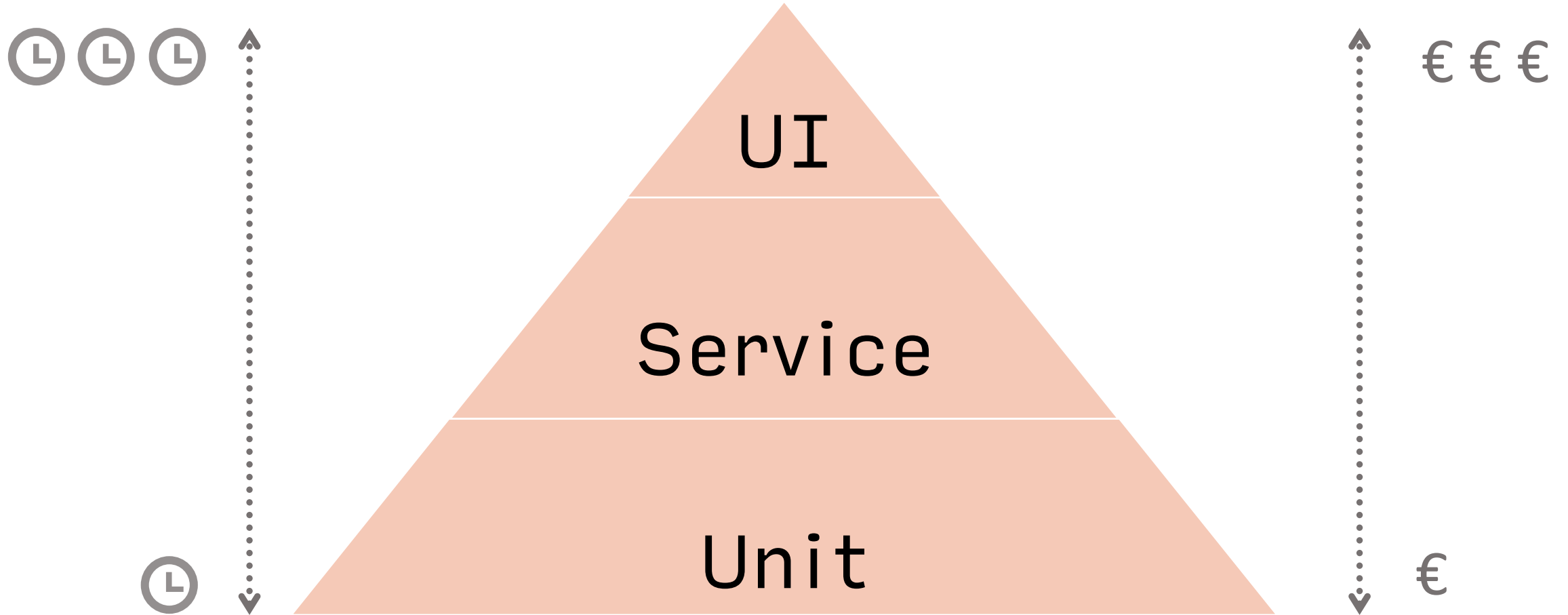
ARGUMENTS to convert
your boss a Unit
Testing fan

Argument to get time

What is unit testing

...in non techies words

Argument to get time



Argument to get time

The goal of unit testing
is to **sustain project growth**
by preventing software entropy

Argument to get time

2
Solid unit tested code
increases time to market
by minimizing regressions

Argument to get time

3

Writing unit tests today
reduces the cost
of the feature of tomorrow



**Once you get
the time...**

you need to be aware of...

Time stealers

SNEAKY things that
can fill your time

Time stealers



Time running unit tests

Time stealers

2

Time spent reading unit tests

Time stealers

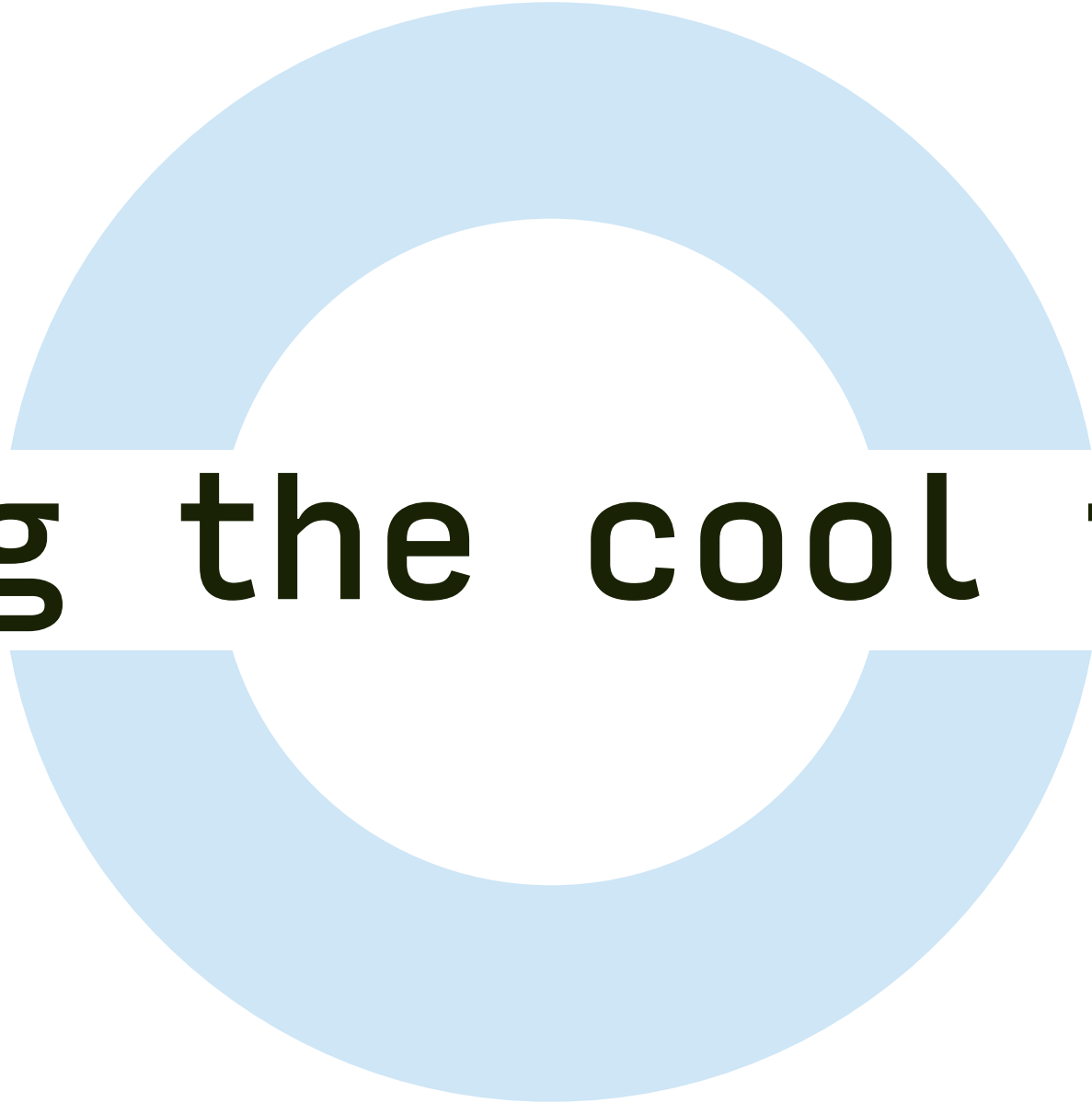
3

Time spent rewriting tests

Time stealers

4

Time dealing with false alarms



Writing the cool test



**Step
back**





Set the concepts
STRAIGHT

Concepts to get straight

Test Accuracy

...what meaningful or cool actually stand for

Concepts to get straight

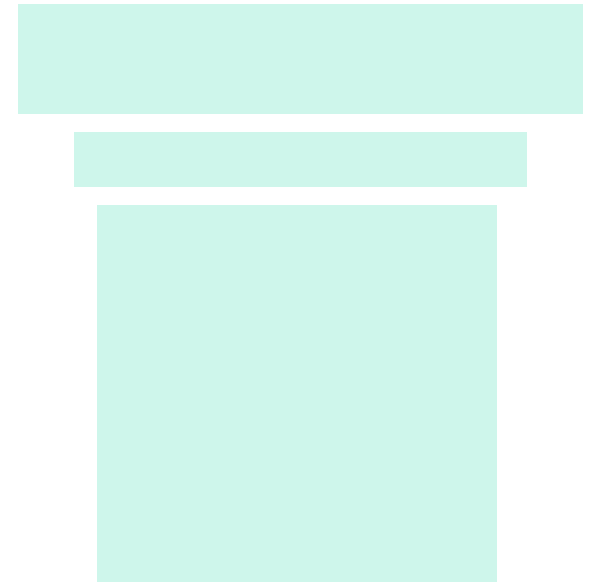
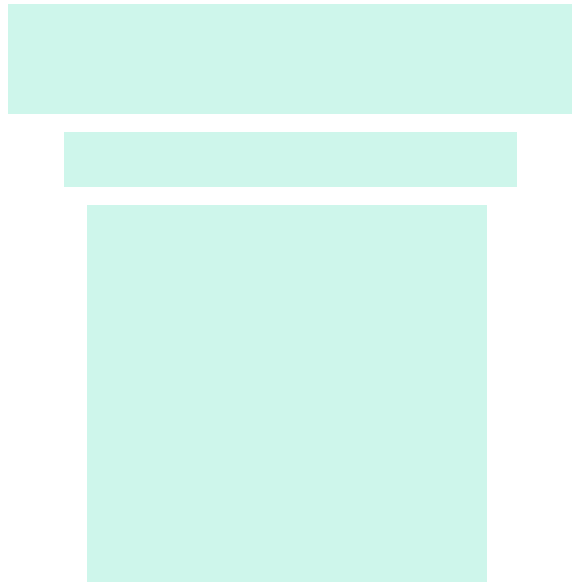
Test Result	Functionality	
	Correct	Broken
Pass		 False Negative
Fail	 False Positive	

Resistance to Refactoring

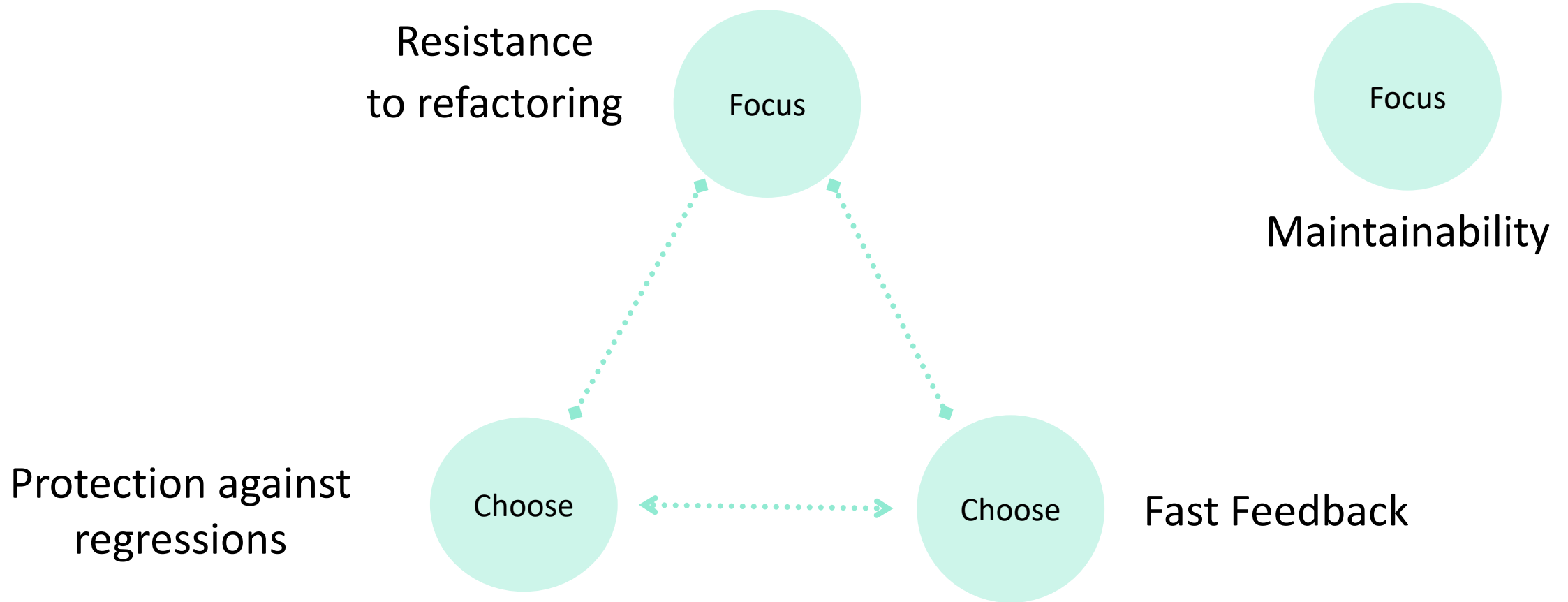
Protection against regressions

Concepts to **get straight**

Pillars of Unit Testing




Concepts to **get straight**





Mental hacks

WRITING the test that
helps



Mental hack: Black box testing



Test **WHAT** happens, not **HOW** it happens

Mental hack: Anti Example

```
41 it("<TodoList /> #completeCalls", () => {
42   ··const todos = ["a", "b", "c"];
43   ··const addTodo = jest.fn();
44   ··const list = mount(
45     ··|··|··<TodoList todos={todos} addTodo={addTodo}/>
46   ··);
47   ··list.find("button").last().simulate("click");
48   ··expect(addTodo.mock.calls.length).toBe(3);
49   });
50
```

Mental hack: Example

```
26 describe('Add Todo', () => {
27   ..it('Should add a new to-do by using keyboard only and pressing ENTER', () => {
28     ..const { queryByText } = render();
29     ..
30     ..const input = queryByText('Add a todo');
31     ..const todoList = queryByText('Todo List');
32     ..
33     ..fireEvent.change(input, { target: { value: 'Buy chocolates for mom' } });
34     ..fireEvent.keyPress(input, { keyCode: keyCodes.ENTER });
35     ..
36     ..expect(todoList).toHaveTextContent('Buy chocolates for mom');
37   });
38 });
```

Mental hack: Test what matters

2

Focus on: important or complex code

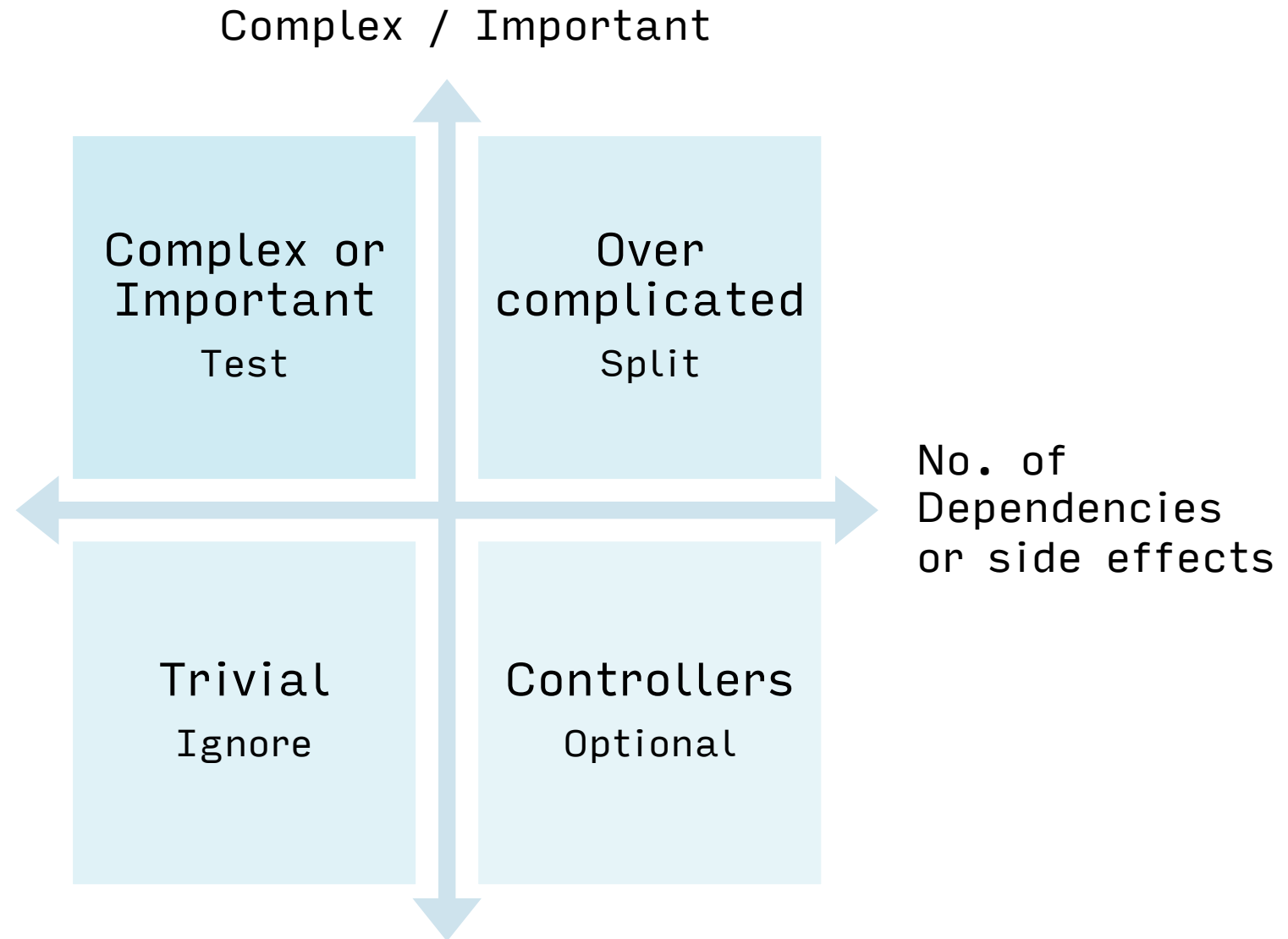
Mental hack: Test what matters

2

Shopping Cart in E-Commerce
Validations in Form Heavy pages

Mental hack: Test what matters

2





Mental hacks

The **STRUCTURE** of the test
that helps

Mental Hack: Test Anatomy

Given — When — Then

Mental Hack: Your test shall tell

2

Function — Scenario —
Expected behavior

Mental Hack: Anti Example

```
41 it("<TodoList /> #completeCalls", () => {
42   ··const todos = ["a", "b", "c"];
43   ··const addTodo = jest.fn();
44   ··const list = mount(
45     ··|··|··<TodoList todos={todos} addTodo={addTodo}/>
46   ··);
47   ··list.find("button").last().simulate("click");
48   ··expect(addTodo.mock.calls.length).toBe(3);
49   });
50
```

Mental Hack: Example

```
26 describe('Add Todo', () => {
27   ···it('Should add a new to-do by using keyboard only and pressing ENTER', () => {
28     ···const { queryByText } = render();
29     ···
30     ···const input = queryByText('Add a todo');
31     ···const todoList = queryByText('Todo List');
32     ···
33     ···fireEvent.change(input, { target: { value: 'Buy chocolates for mom' } });
34     ···fireEvent.keyPress(input, { keyCode: keyCodes.ENTER });
35     ···
36     ···expect(todoList).toHaveTextContent('Buy chocolates for mom');
37   ···});
38   ···});
39 }
```

**Mental Hack: Your test aint'
wrong...**

3

...your code is!

Mental Hack: Read the docs


4

Get familiarized with your
testing framework API



Recap

- The vicious circle
- Break the loop
- Getting the time
- Mental Hacks (*Using the time wisely*)



Spoiler alerts

- **WANT:** a more scalable and flexible codebase
- **PRICE:** the beginner struggle, but keep going
- **NEW YOU:** you'll write more testable code from start



Thank You

Where to find ~~me~~...

DanielaValero.com