

TDD

Reddy Tintaya - FullStack Developer at Oof1-Gfrog

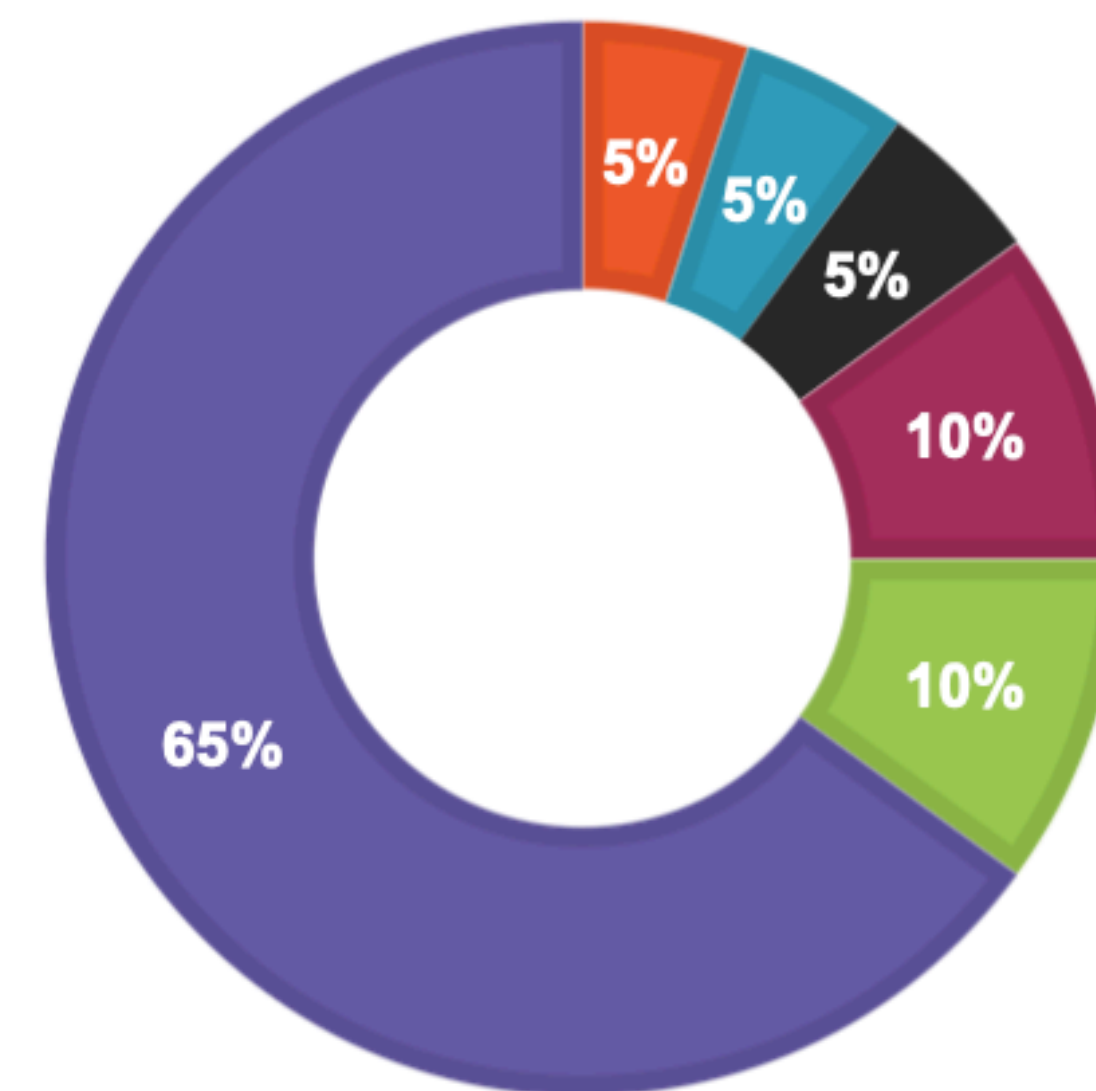
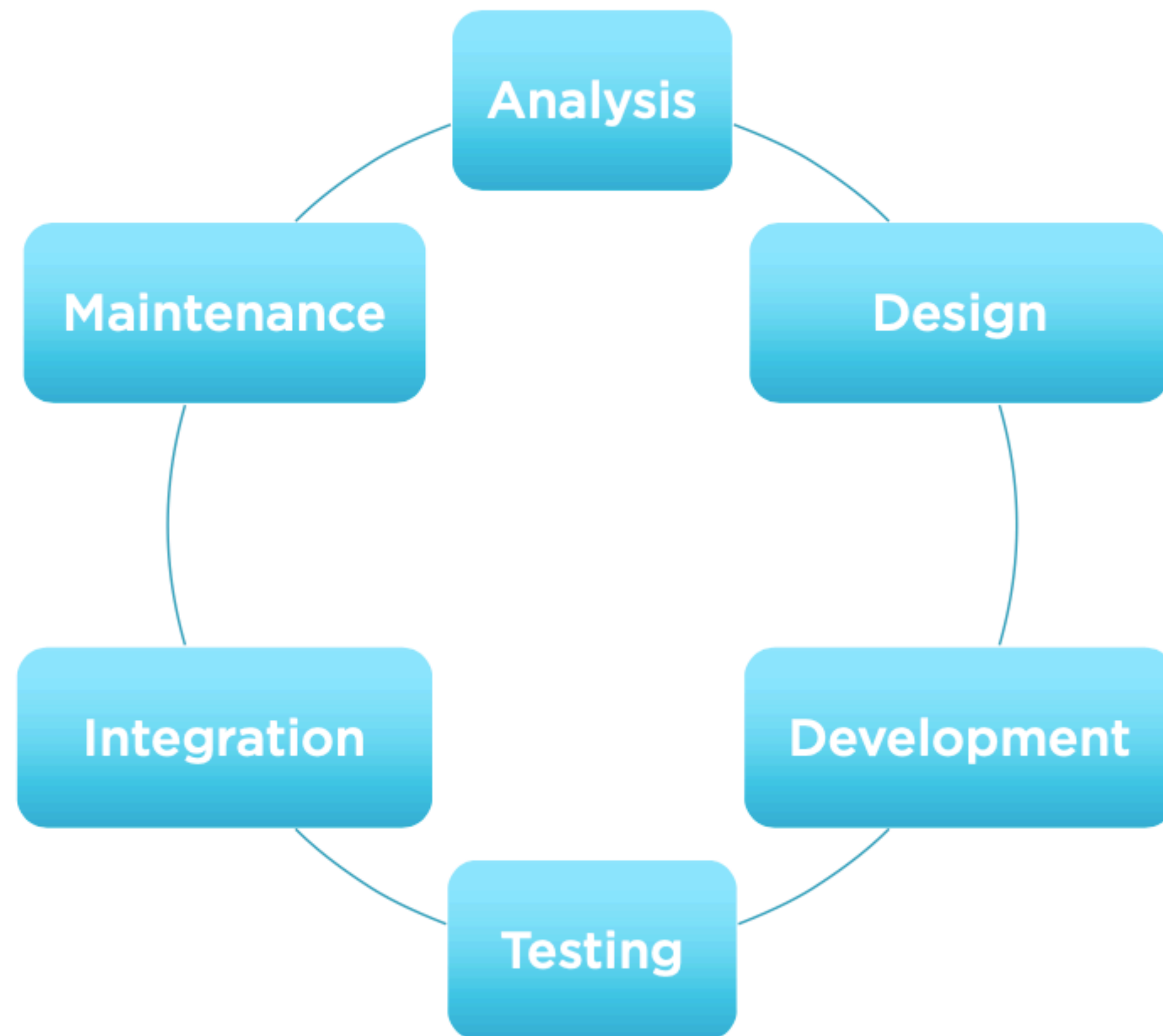
Prerequisites

- No TDD experience required
- Experience writing code

Target Audience

- Software Developers
- Software Testers
- Software Managers

Costs of developing software



- Analysis
- Design
- Development
- Testing
- Integration
- Maintenance

**Maintenance accounts for 65% of
all software development costs!**

Maintenance Challenge

Maintenance Challenge



- Frágil con el tiempo
- Incrementa su rigidez (resistencia al cambio)



Maintenance Challenge



Code Entropy

- Frágil con el tiempo
- Incrementa su rigidez



Isolated Ownership

- Pensamiento sesgado
- Falta de empoderamiento de equipo

Maintenance Challenge



Code Entropy

- Frágil con el tiempo
- Incrementa su rigidez



Isolated Ownership

- Pensamiento sesgado
- Falta de empoderamiento de equipo



Infrequent Validation

- Regression-prone
- Frágil contra cambios de código

Legacy code

- Código heredado por alguien mas
- Código heredado por una version mas antigua del software
- Código sin tests

TDD? Test Driven Development

Test

- Proceso para establecer la calidad, performance (comportamiento) o fiabilidad de algo, especialmente antes de llevarlo a que sea usado masivamente.

Test

- Proceso para establecer la calidad, performance (comportamiento) o fiabilidad de algo, especialmente antes de llevarlo a que sea usado masivamente.



Test

- Proceso para establecer la calidad, performance (comportamiento) o fiabilidad de algo, especialmente antes de llevarlo a que sea usado masivamente.



**Satisfies
requirements**



**Responds correctly
to all input**

Test

- Proceso para establecer la calidad, performance (comportamiento) o fiabilidad de algo, especialmente antes de llevarlo a que sea usado masivamente.



**Satisfies
requirements**



**Responds correctly
to all input**



**Acceptable
performance**

Test-Driven Development

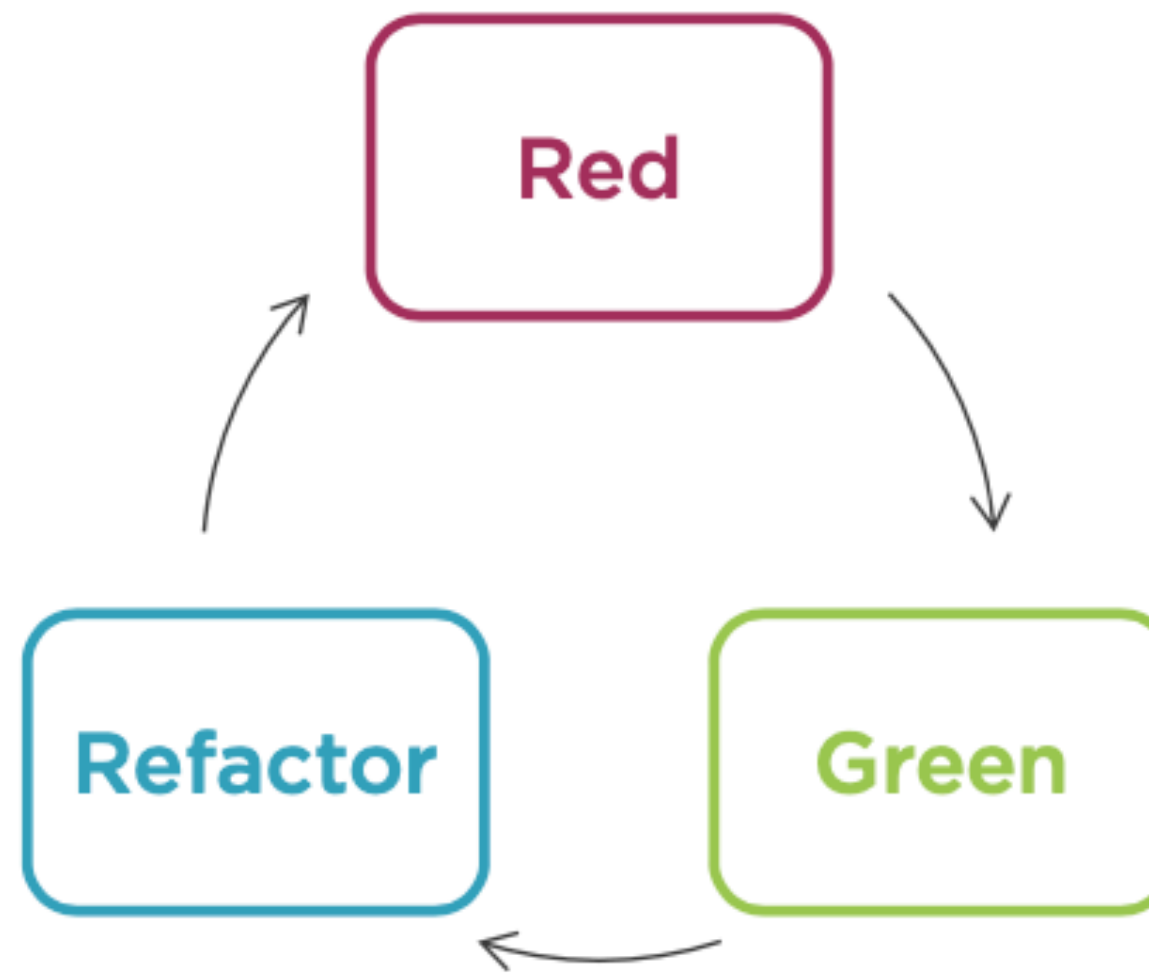
- ***Proceso de Desarrollo de Software*** que se centra en la repetición de un ***pequeño ciclo de desarrollo***: los requerimientos son convertidos a pruebas (tests) muy específicas, luego el software es mejorado para pasar únicamente esos tests. (*Wikipedia-refactorizada*)

Test-Driven Development

- ***Proceso de Desarrollo de Software*** que se centra en la repetición de un ***pequeño ciclo de desarrollo***: los requerimientos son convertidos a pruebas (tests) muy específicas, luego el software es mejorado para pasar únicamente esos tests. (*Wikipedia-refactorizada*)

“Red. Green. Refactor.”

Red. Green. Refactor.



Red

Write test that fails



Green

Make test pass



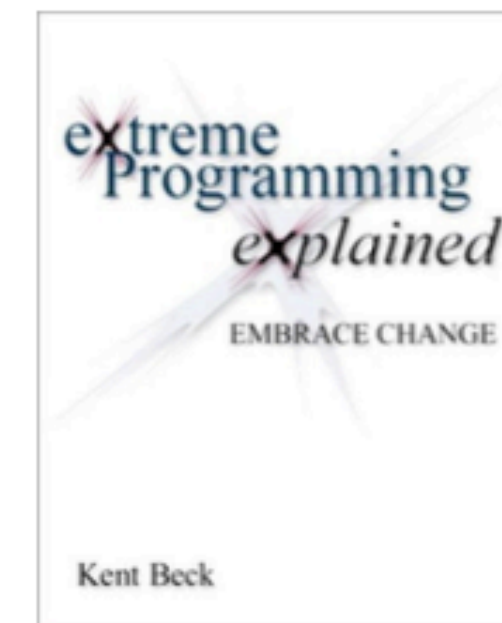
Refactor

Refactor/cleanup code

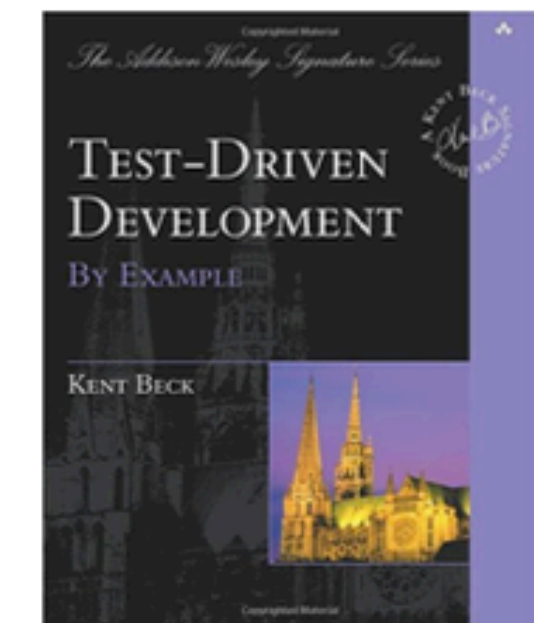
TDD History

JUnit

Kent Beck
Erich Gamma



eXtreme Programming
Kent Beck



TDD: By Example
Kent Beck



"Simple Smalltalk Testing"
SUnit (1994)
Kent Beck



1960s

1989

1999

2002

Por que usar TDD

Beneficios para el negocio



**Requirements
Verification**



**Regression
Catching**



**Lower
Maintenance Costs**

Beneficios para el desarrollador



Design-First



Momentum



Confidence

Ayuda a concentrarse en el Cliente

What Was Spec'd



CC: DanielCD
<http://bit.ly/2jyS1sD>

What Was Delivered



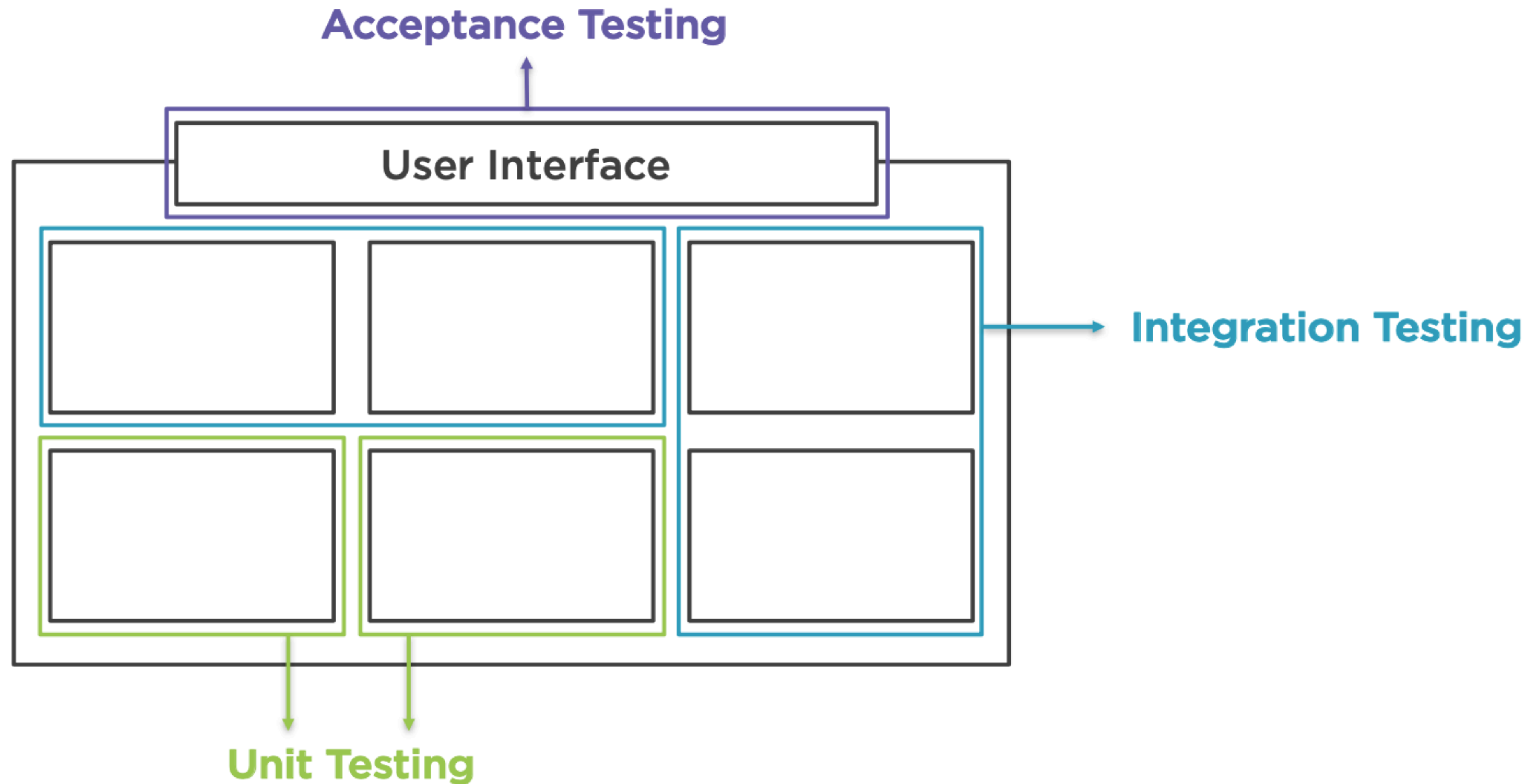
CC: Public Domain

What Customer Wanted

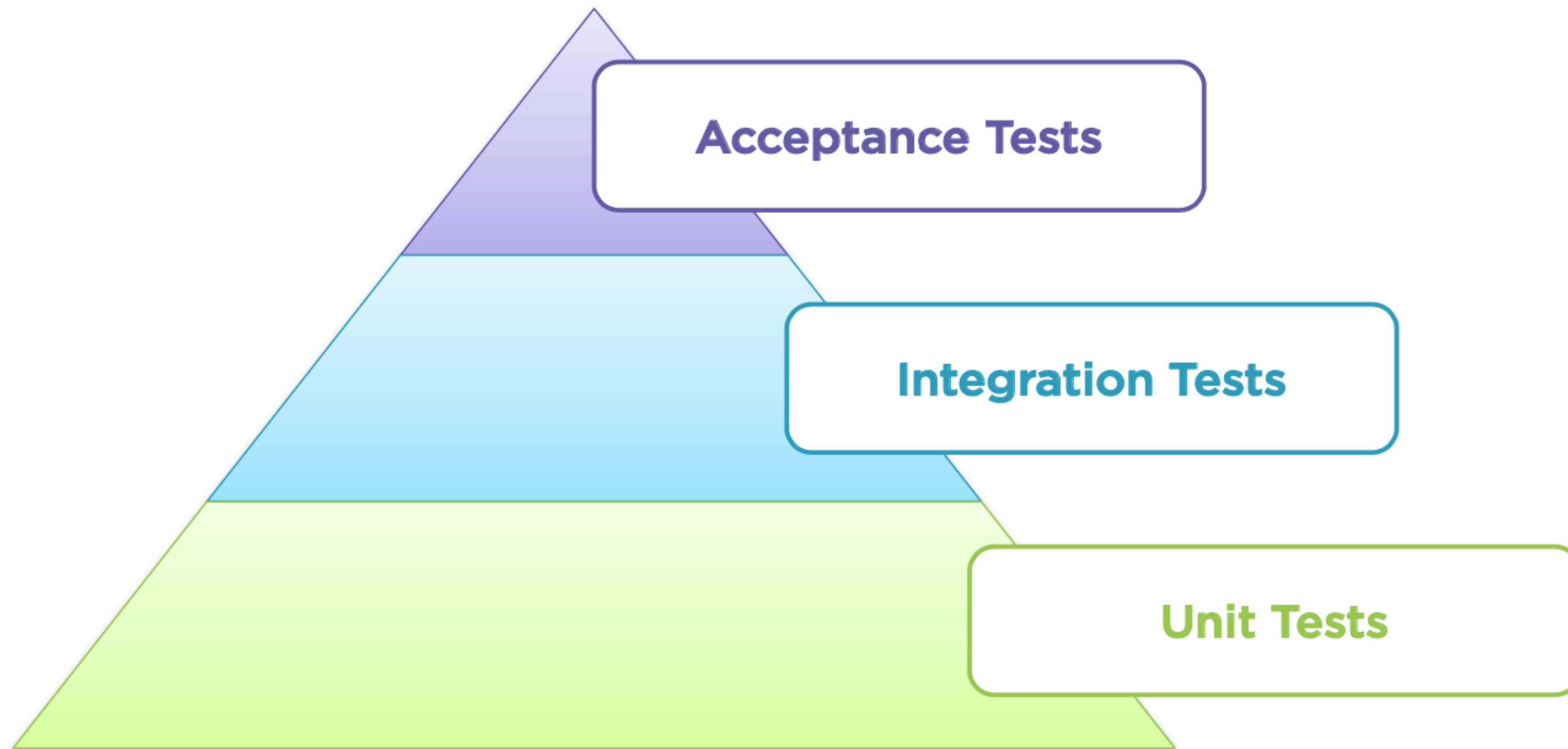


CC: Rudolf Stricker
<http://bit.ly/2jec9AK>

Types of Testing



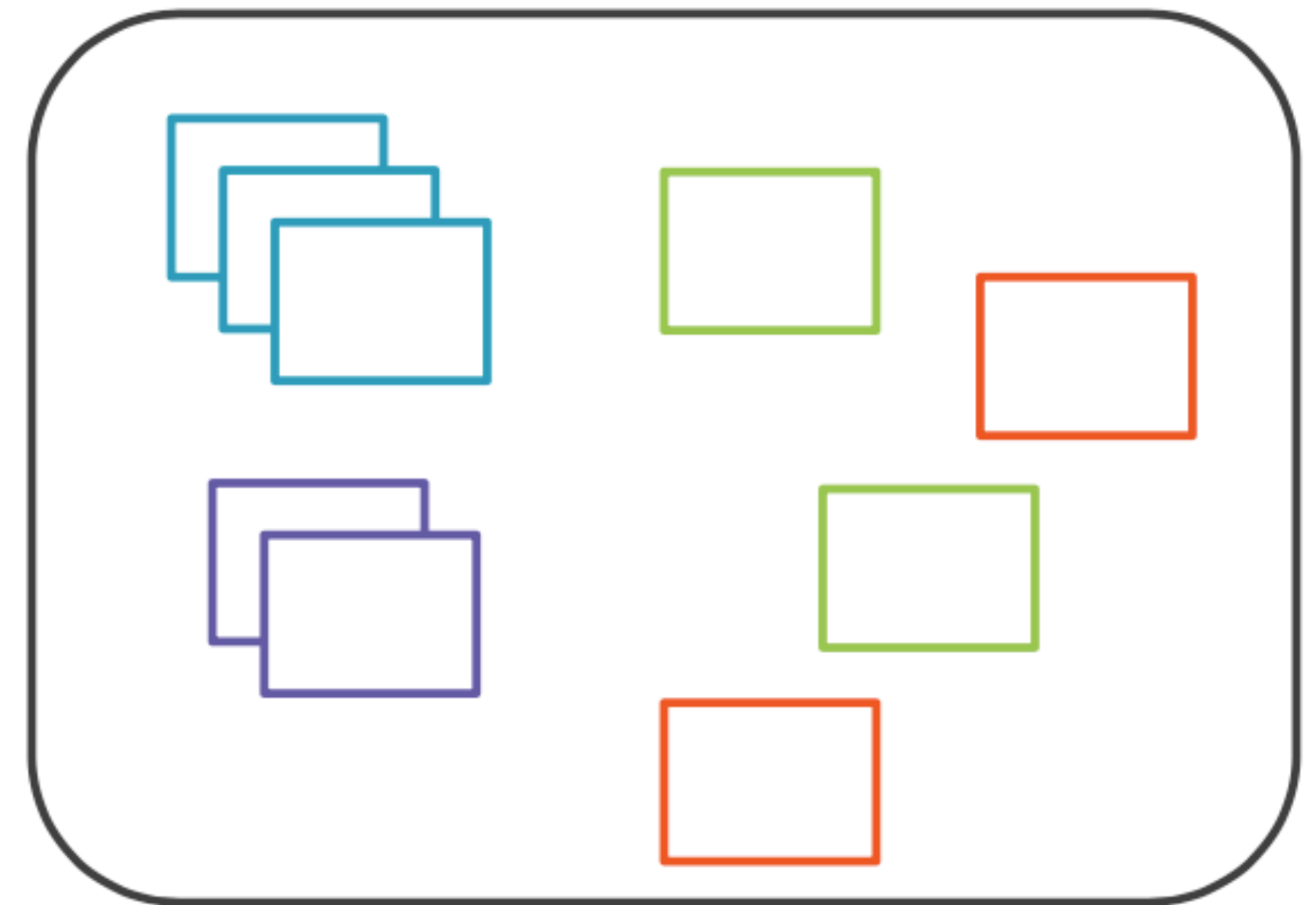
Balanced Testing



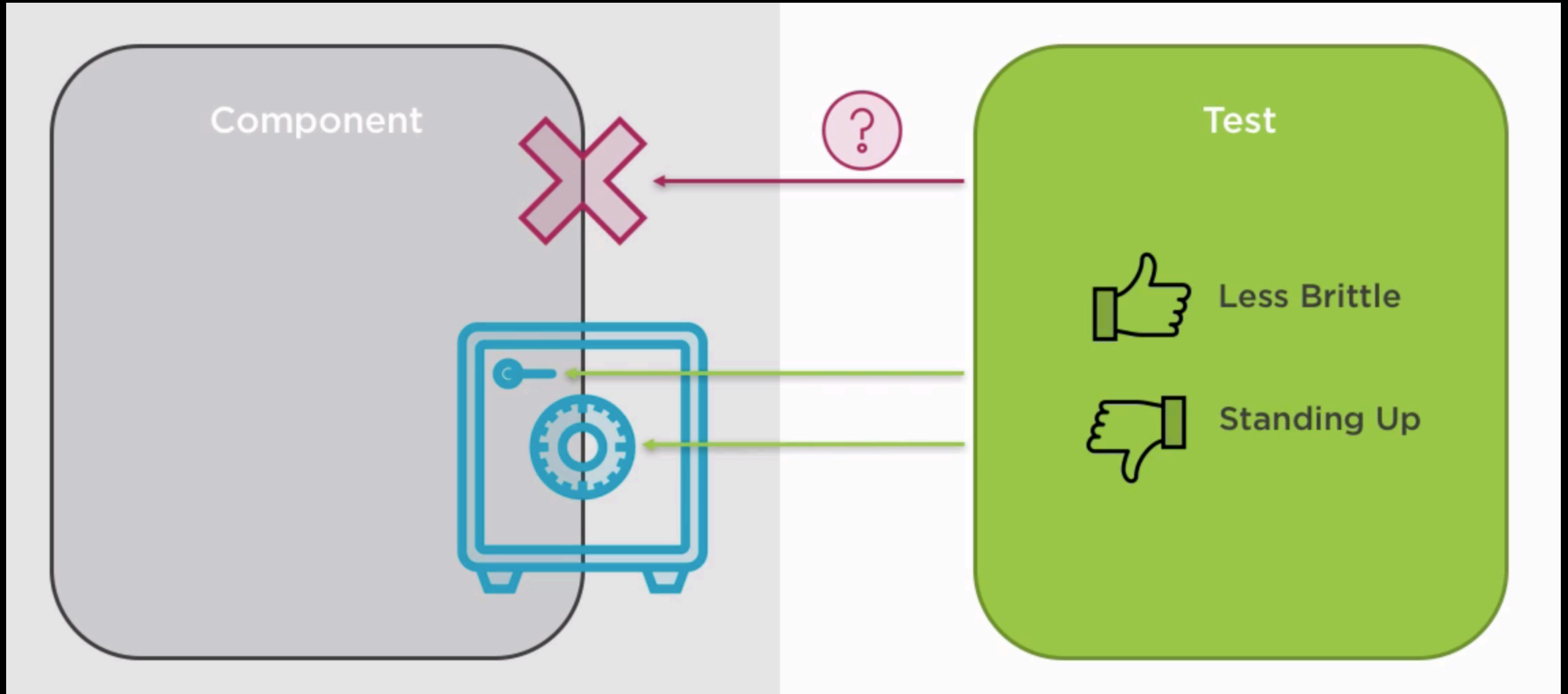
Black Box Testing



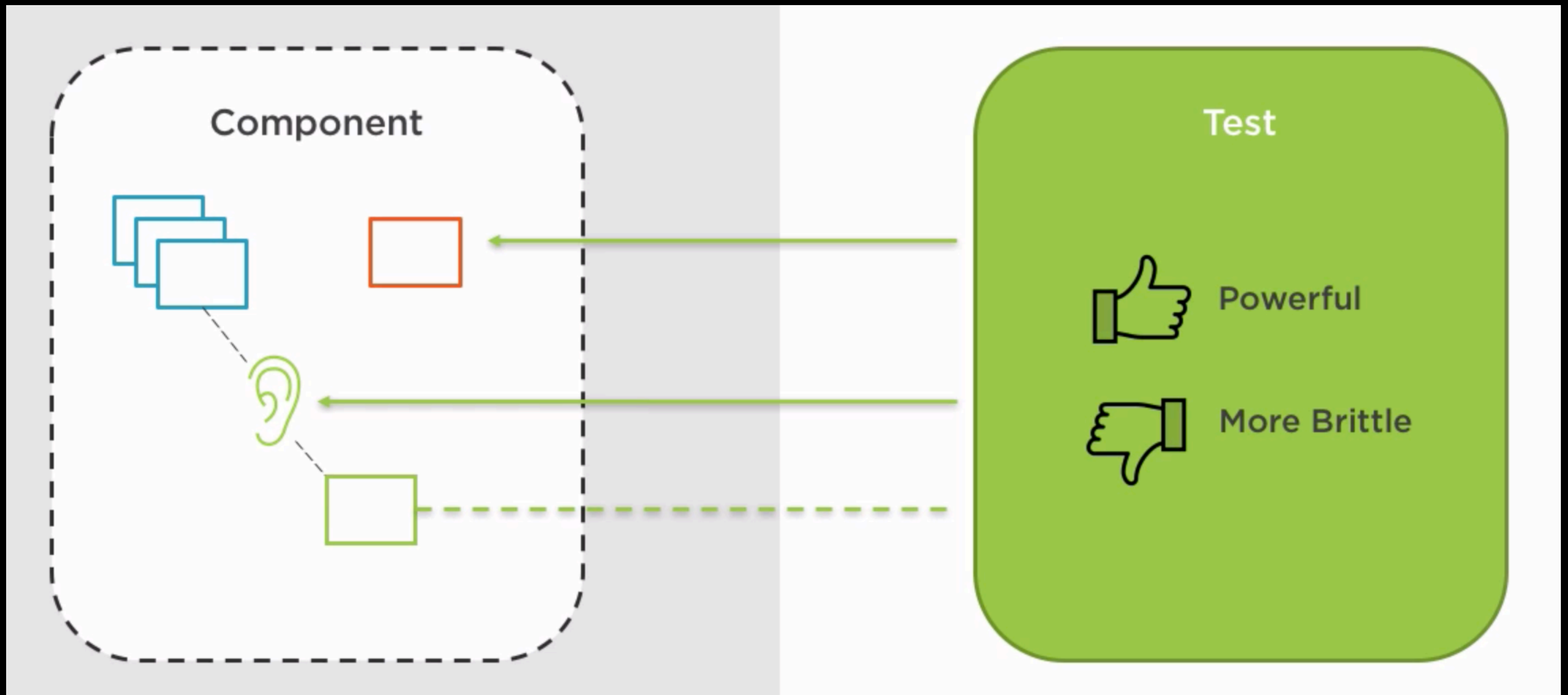
White Box Testing



Black-Box Testing



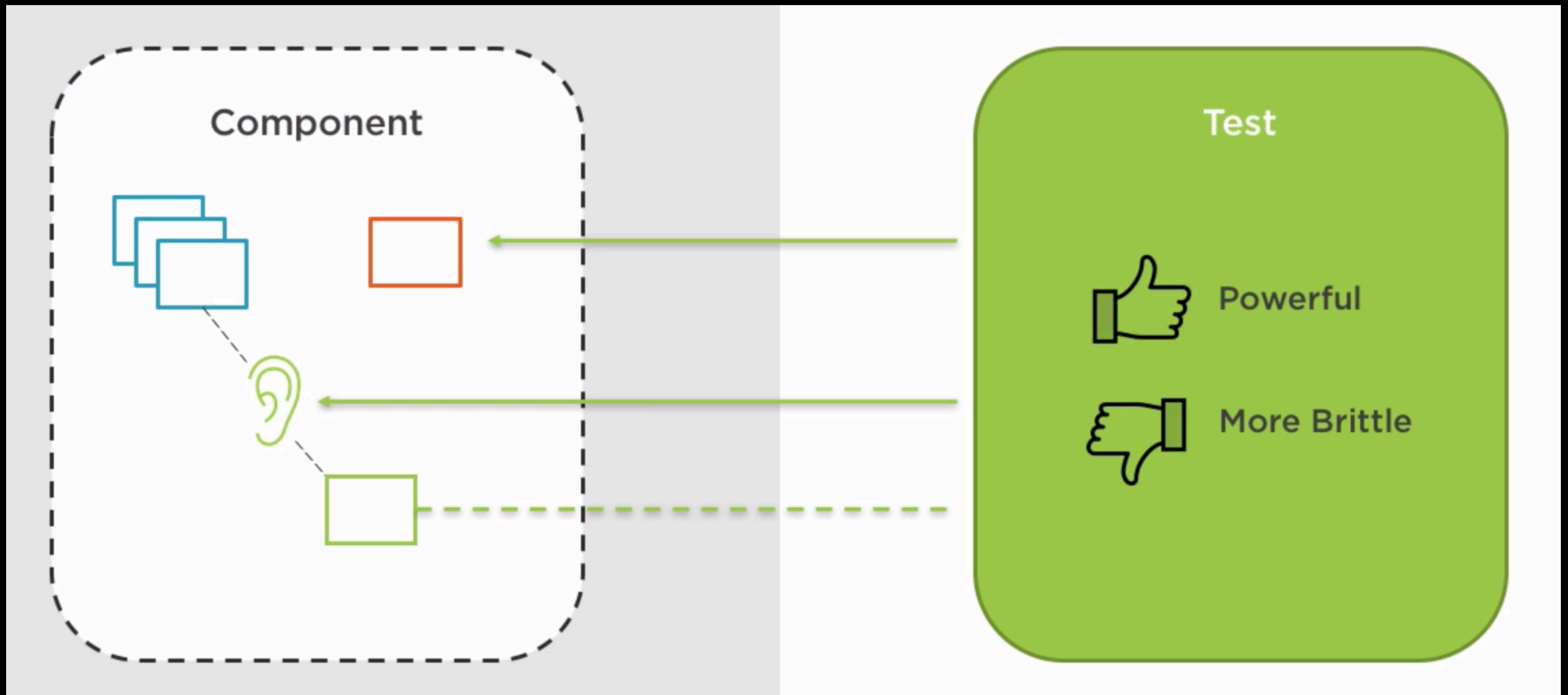
White-Box Testing

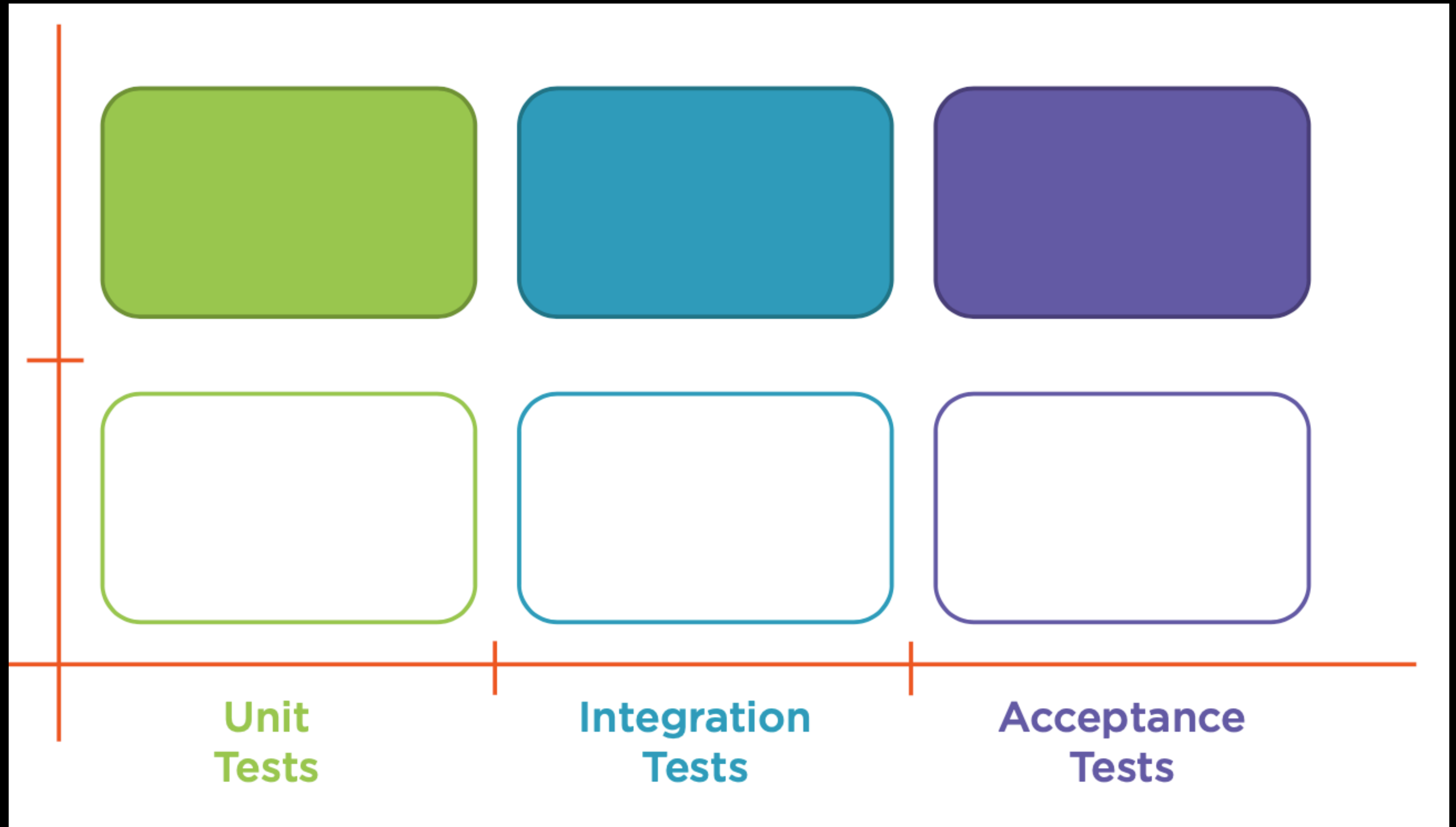


```
class Tiempo:
    ... def get_current_time(self):
    ... | ... return self.current_time
```

```
class ComponenteATestear:
    ... def hola_mundo(self):
    ... | ... current_time = Tiempo().get_current_time()
    ... | ... return "hola, son las {}".format(current_time)
    ... | ...
```

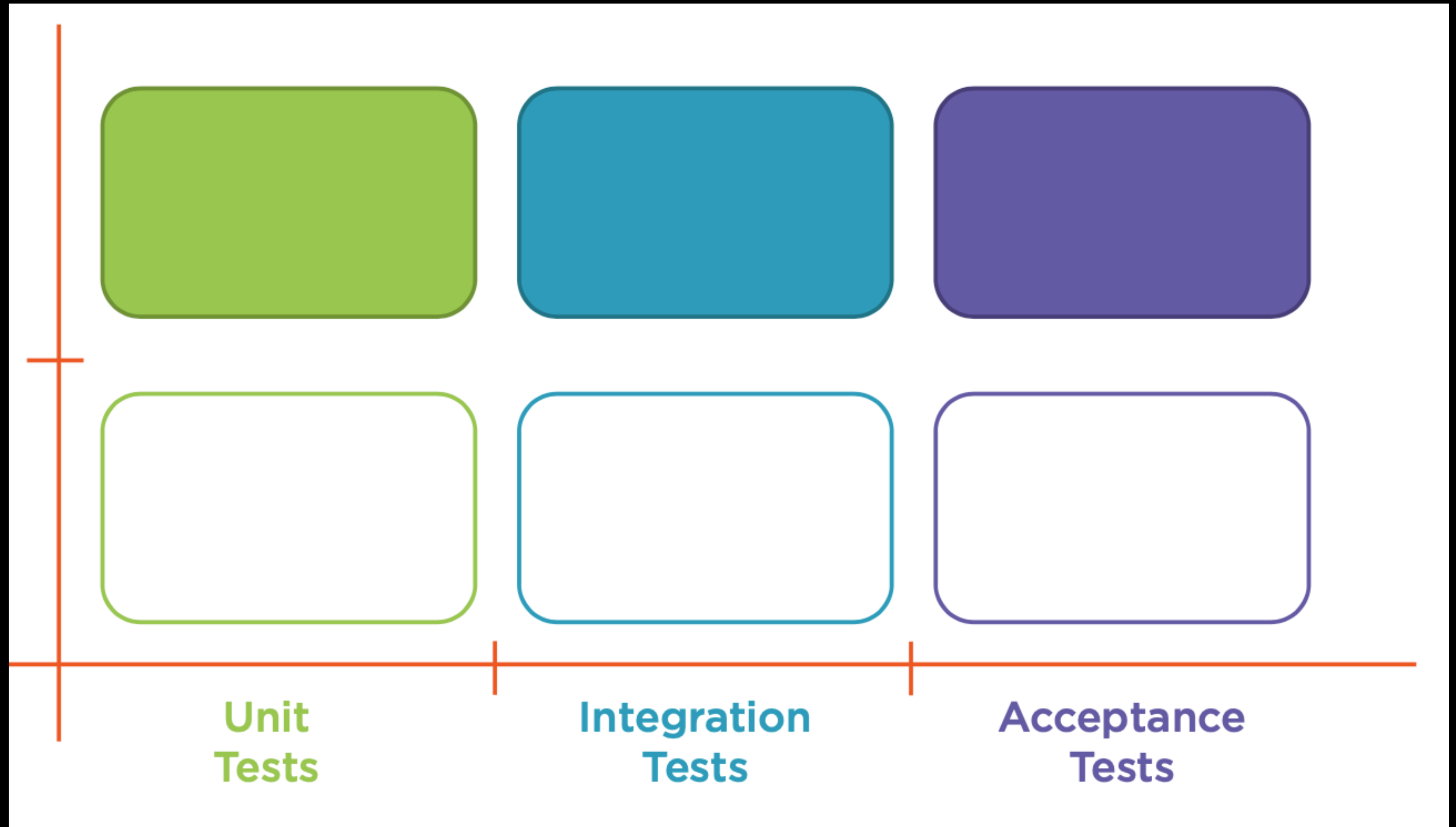
White-Box Testing





Black-Box Testing

White-Box Testing



Otros tipos de testing

- Penetration Testing
- Boundary Testing
- Fuzz Testing
- Smoke Testing
- Stress Testing
- A/B Testing

Test Frameworks

xUnit Frameworks

SUnit (Smalltalk)



JUnit (Java)

NUnit **RUnit** **CppUnit** **EUnit** **PerlUnit** **PHPUnit** **xUnit.net** 

Mocha **AVA** **py.test** **unittest** **FsTest**

https://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

User Interface Frameworks



Selenium

**Visual Studio Coded UI Test
(Microsoft)**

**Test Studio
(Telerik)**



Watir

**Silk Test
(Micro Focus)**

System Frameworks

Simian Army

<https://github.com/Netflix/SimianArmy>



Chaos Monkey



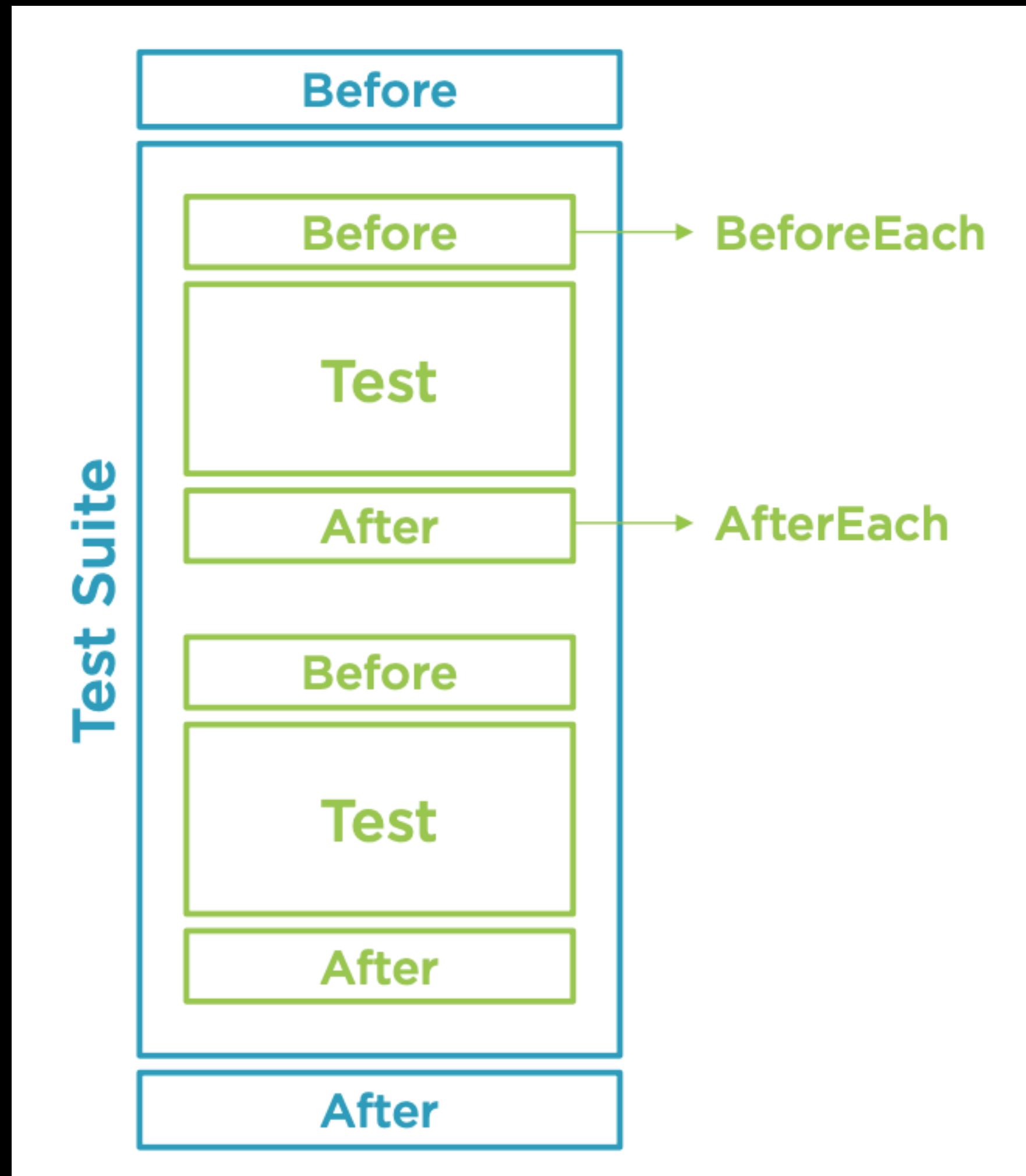
Latency Monkey



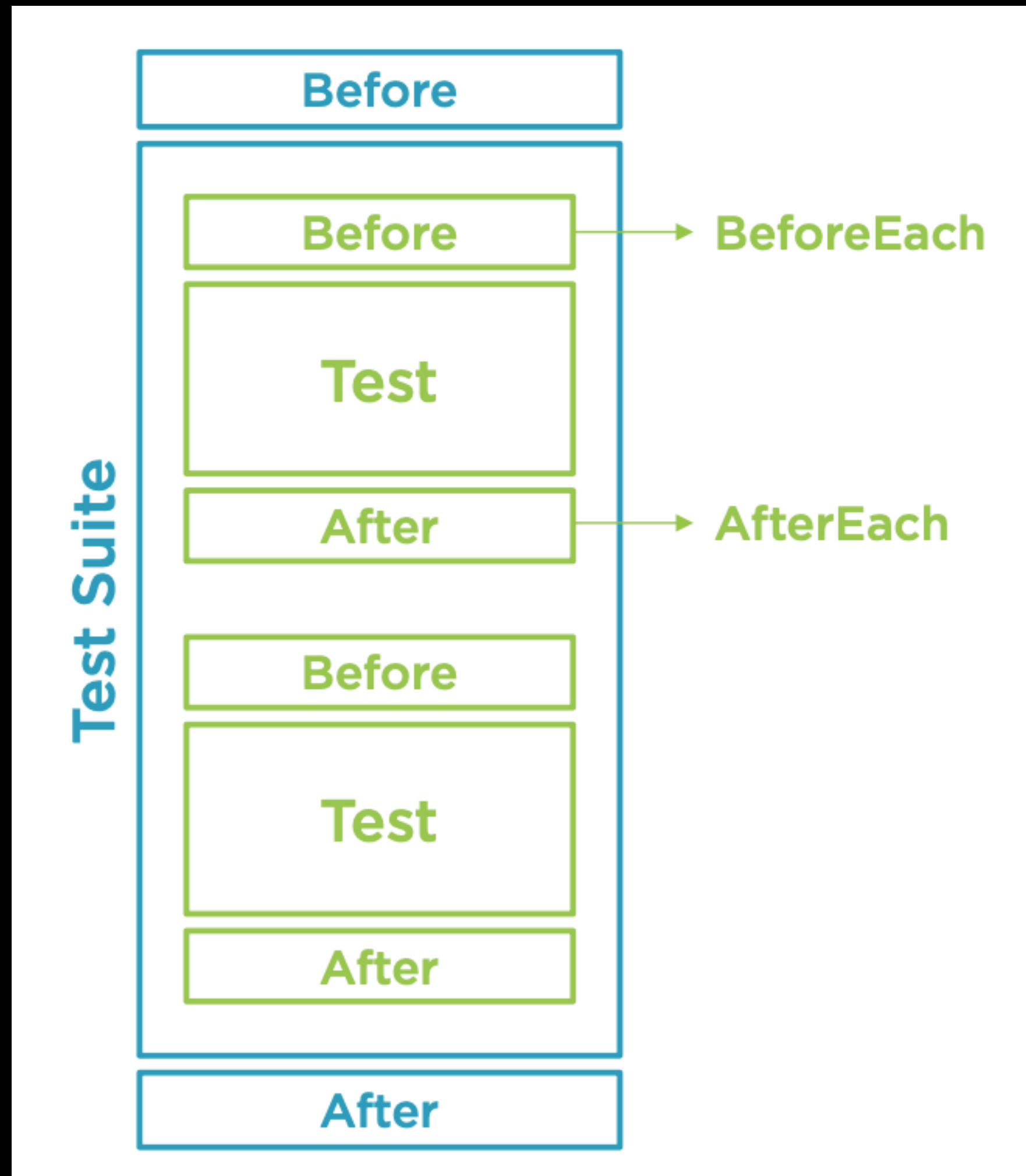
Janitor Monkey

Testing Concepts

Framework Concepts

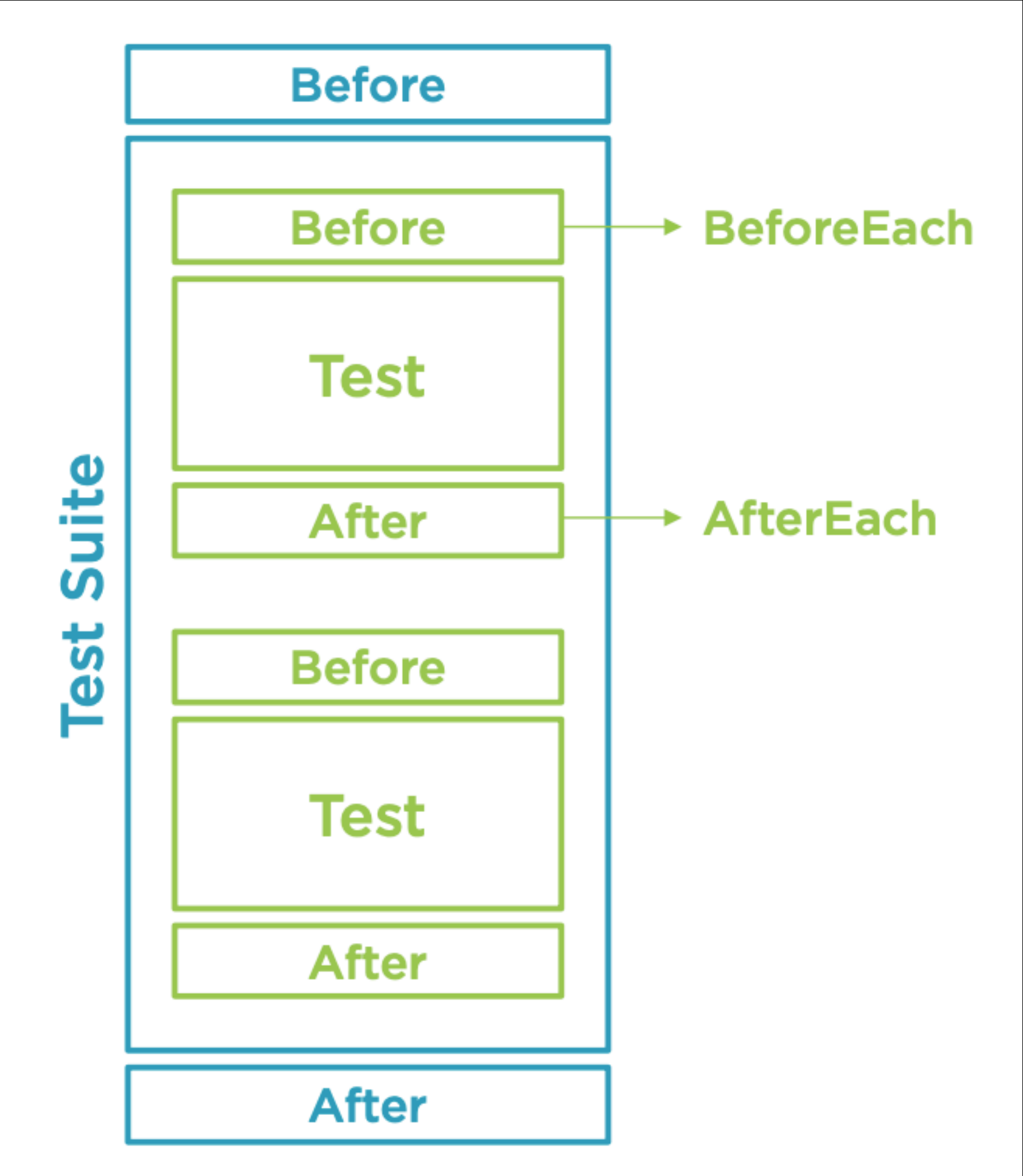


Framework Concepts



Test
Before Test
After Test

Framework Concepts



Test
Before Test
After Test

Test Suite
Before Suite
After Suite

Verification Concepts

Assert

```
Assert.IsTrue(someBoolean);  
Assert.IsFalse(someBoolean);
```

```
Assert.IsNull(someValue);
```

```
Assert.AreEqual(3, someValue);
```

```
Assert.Contains(obj, someCollection);
```

```
Assert.StartsWith("foo", someString);
```

```
someBoolean.Should().BeTrue();
```

```
someList.Should().HaveCount(4);
```

```
someString.Should()  
    .StartWith("Hello")  
    .And  
    .EndWith("World");
```