

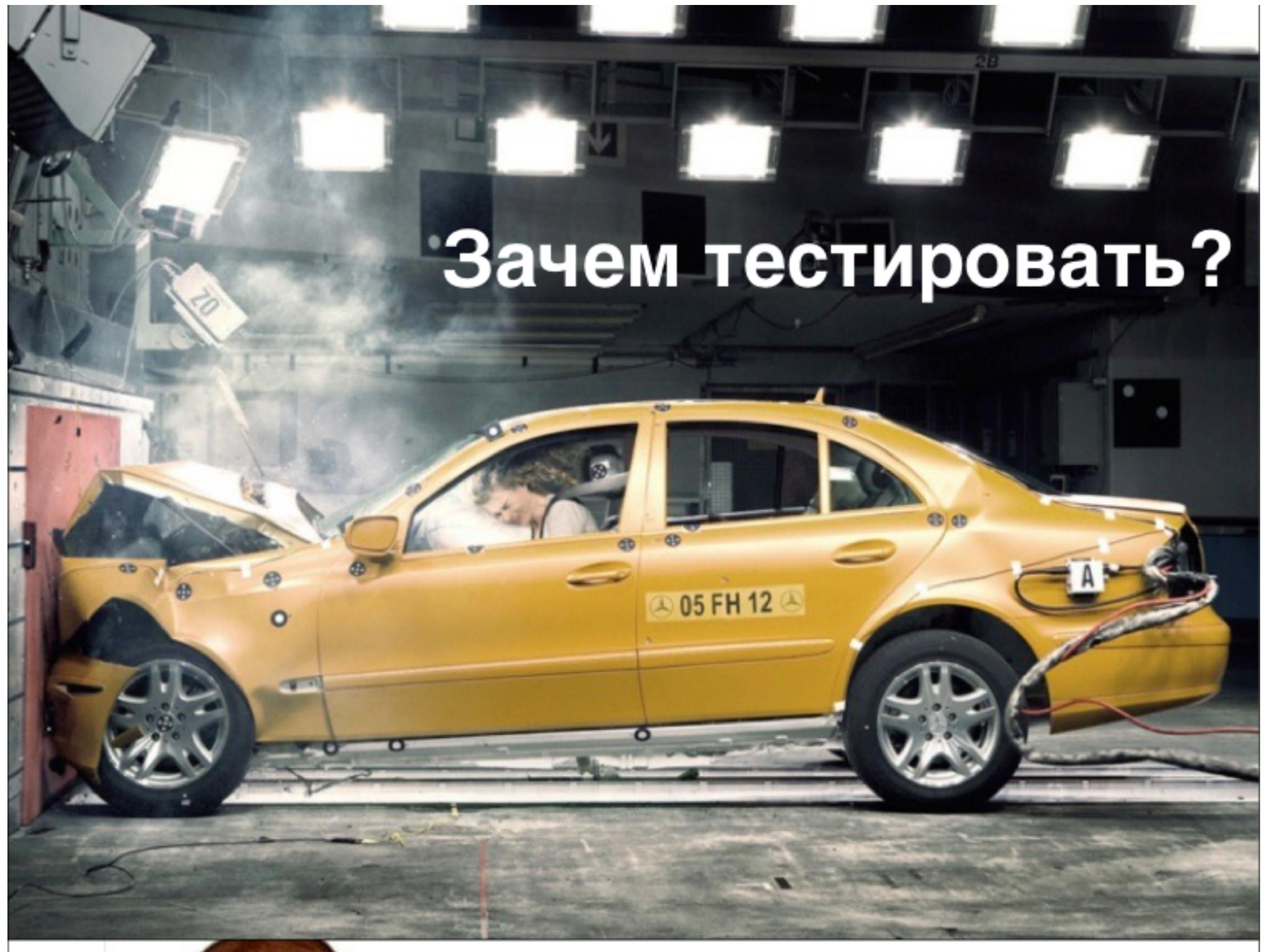
# Модульное тестирование

Антон Бевзюк



ДОДО ПИЦЦА

Зачем тестирувать?





Нельзя просто так  
взять и быстро  
написать хороший  
код!

Страх



# Кругом идиоты



# Обратная связь

## Test Summary

909 tests      0 failures      0 ignored      5.412s duration

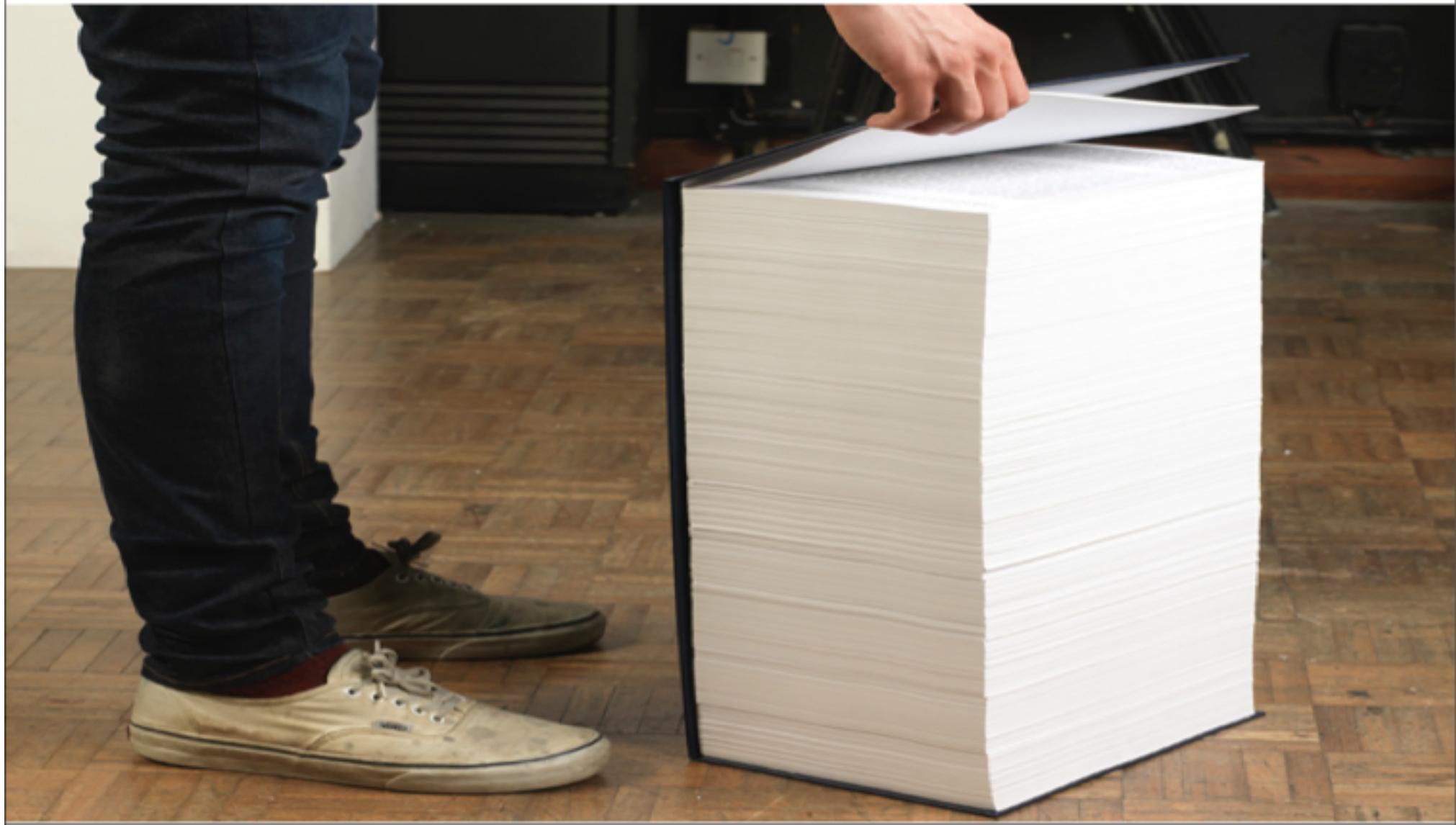
100%  
successful

Packages

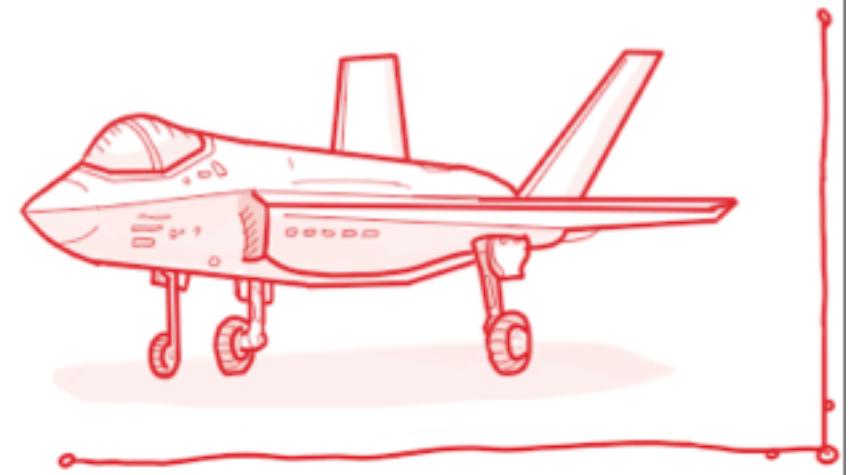
Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">griffon.core.editors.BigDecimalPropertyEditorSpec</a>	15	0	0	0.006s	100%
<a href="#">griffon.core.editors.BigIntegerPropertyEditorSpec</a>	15	0	0	0.006s	100%
<a href="#">griffon.core.editors.BooleanPropertyEditorSpec</a>	17	0	0	0.004s	100%
<a href="#">griffon.core.editors.BytePropertyEditorSpec</a>	13	0	0	0.009s	100%
<a href="#">griffon.core.editors.CalendarPropertyEditorSpec</a>	18	0	0	0.008s	100%
<a href="#">griffon.core.editors.DatePropertyEditorSpec</a>	18	0	0	0.012s	100%

# Документация



# Безопасный рефакторинг



Если люди делают работу,  
которую могут делать  
компьютеры,

То компьютеры  
собираются по ночам  
и смеются...



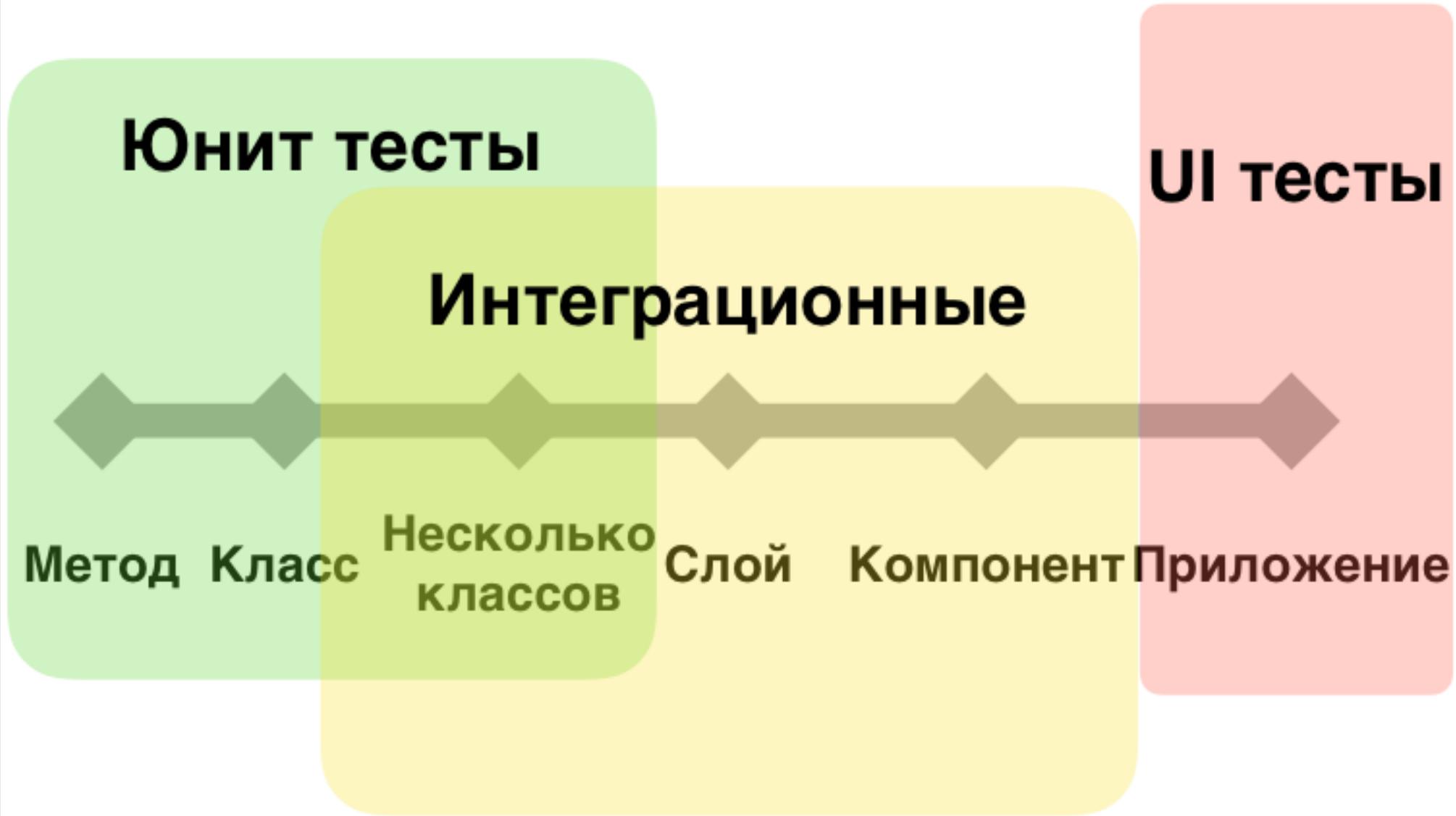
**Monkey Testing**

# **Debuggers are a wasteful Timesink**

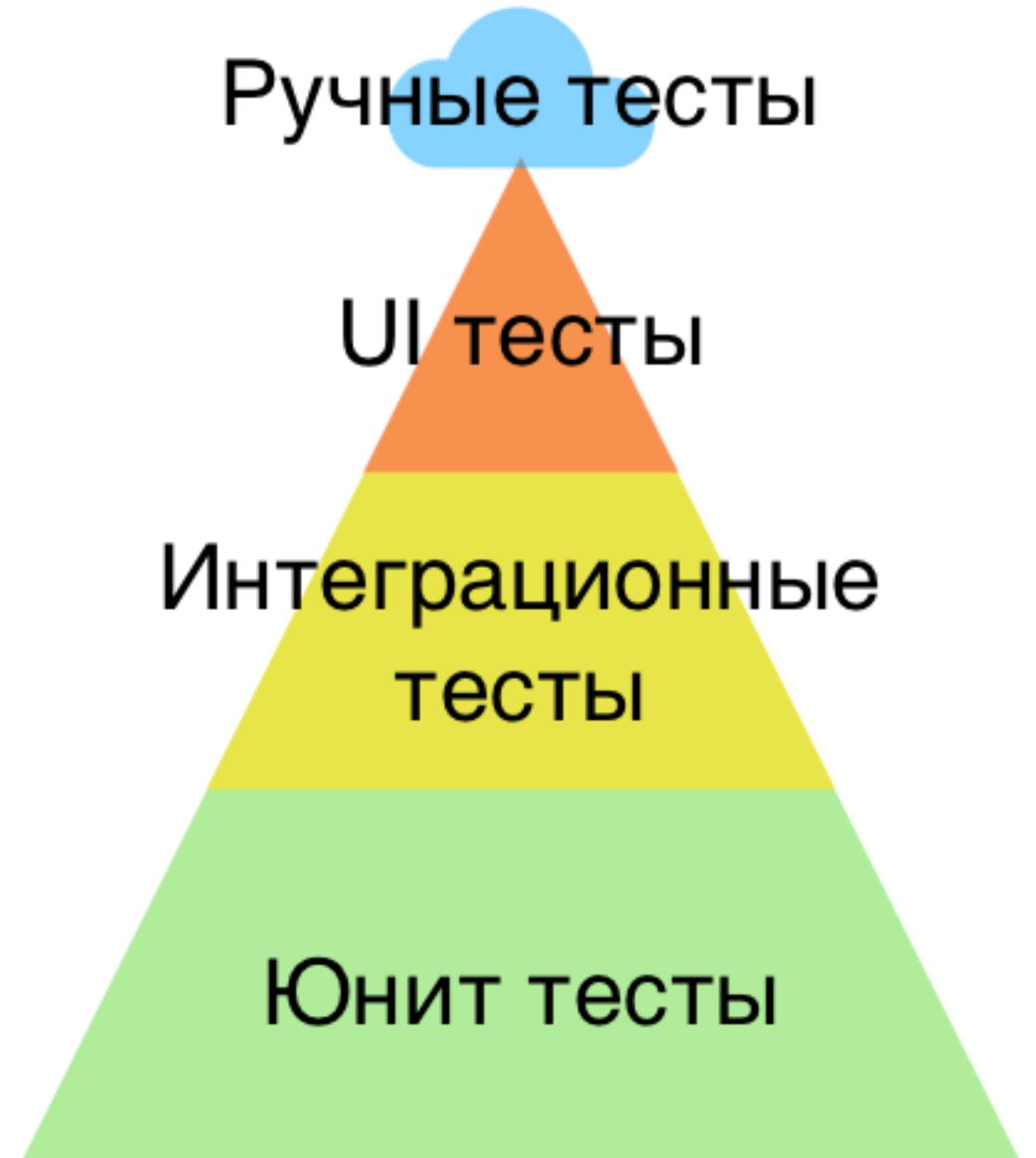
Robert C.  
Martin



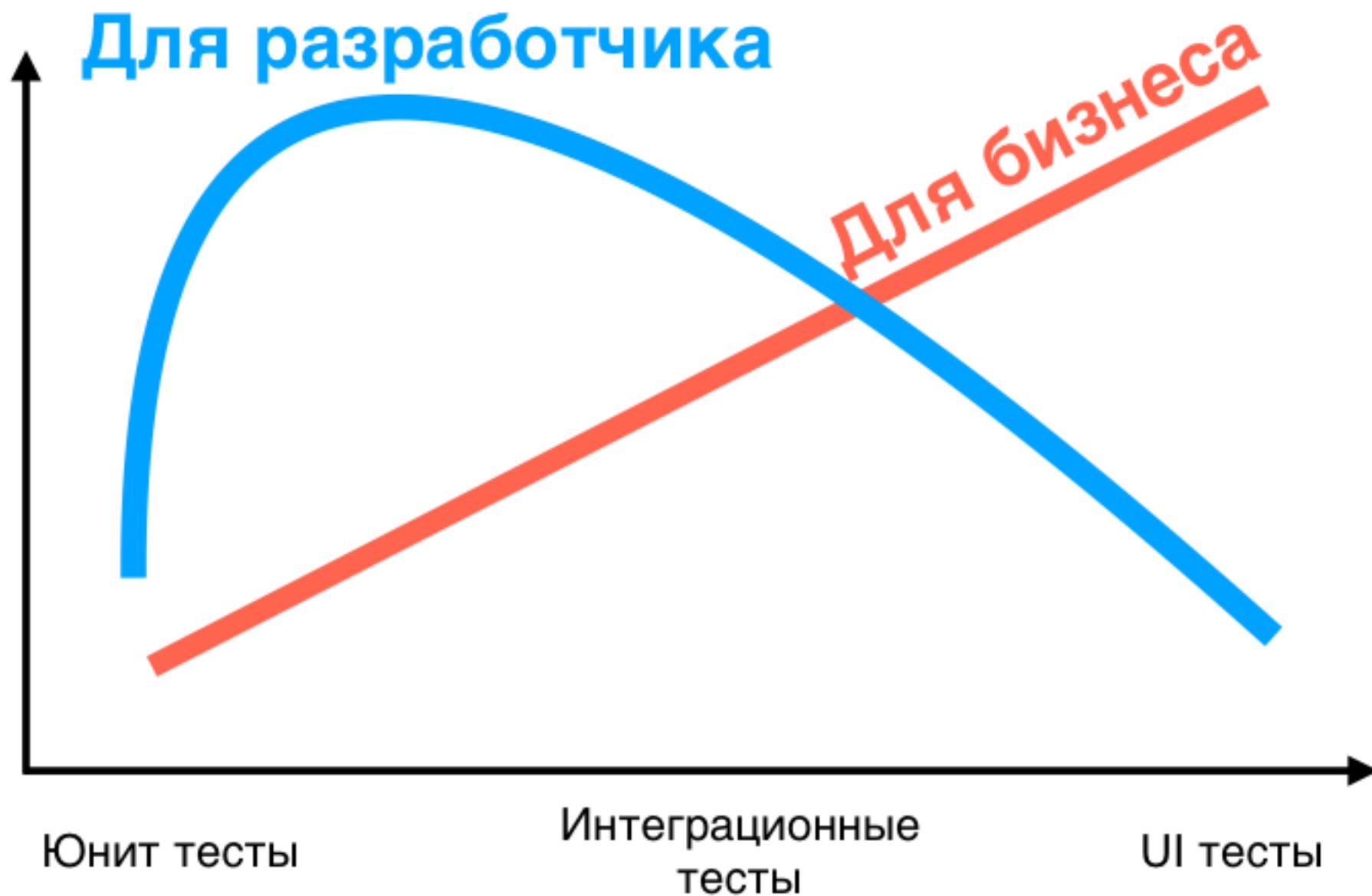
# Что такое Unit?



# Пирамида тестирования



# ЦЕННОСТЬ ТЕСТОВ



# Структура теста

**Arrange**

**Act**

**Assert**

**Given**

**When**

**Then**

# ЮНИТ ТЕСТ КУРИЛЬЩИКА

```
[Test]
public void TestMethod1()
{
    var calc = new Calculator();
    calc.ValidOperation = Calculator.Operation.Multiply;
    calc.ValidType = typeof(int);
    var result = calc.Multiply(-1, 3);
    Assert.AreEqual(result, -3);
    calc.ValidOperation = Calculator.Operation.Multiply;
    calc.ValidType = typeof(int);
    result = calc.Multiply(1, 3);
    Assert.IsTrue(result == 3);
    if (calc.ValidOperation == Calculator.Operation.Invalid)
    {
        throw new Exception("Operation should be valid");
    }
    calc.ValidOperation = Calculator.Operation.Multiply;
    calc.ValidType = typeof(int);
    result = calc.Multiply(10, 3);
    Assert.AreEqual(result, 30);
}
```

# ЗДОРОВОГО ЧЕЛОВЕКА

```
[TestMethod]
public void CanAuthenticateUser() {
    var page = new TestableLoginPage();

    page.AuthenticateUser("user", "user");

    Assert.AreEqual("user", page.AuthenticatedUser);
}
```

# Имя теста

1. <ClassNameTests>.<Act>\_<Arrange>\_<Assert>
2. When<Action>.<Arrange><Assert>
3. <ClassName>Should.<Assert>\_<Arrange>
4. <Arrange>.<Assert>
5. <Assert>.<Arrange>

# ЮНИТ ТЕСТ курильщика

TestClass1

Test1

BlahBlah

AwesomeTest

# здорового человека

BabyCries.

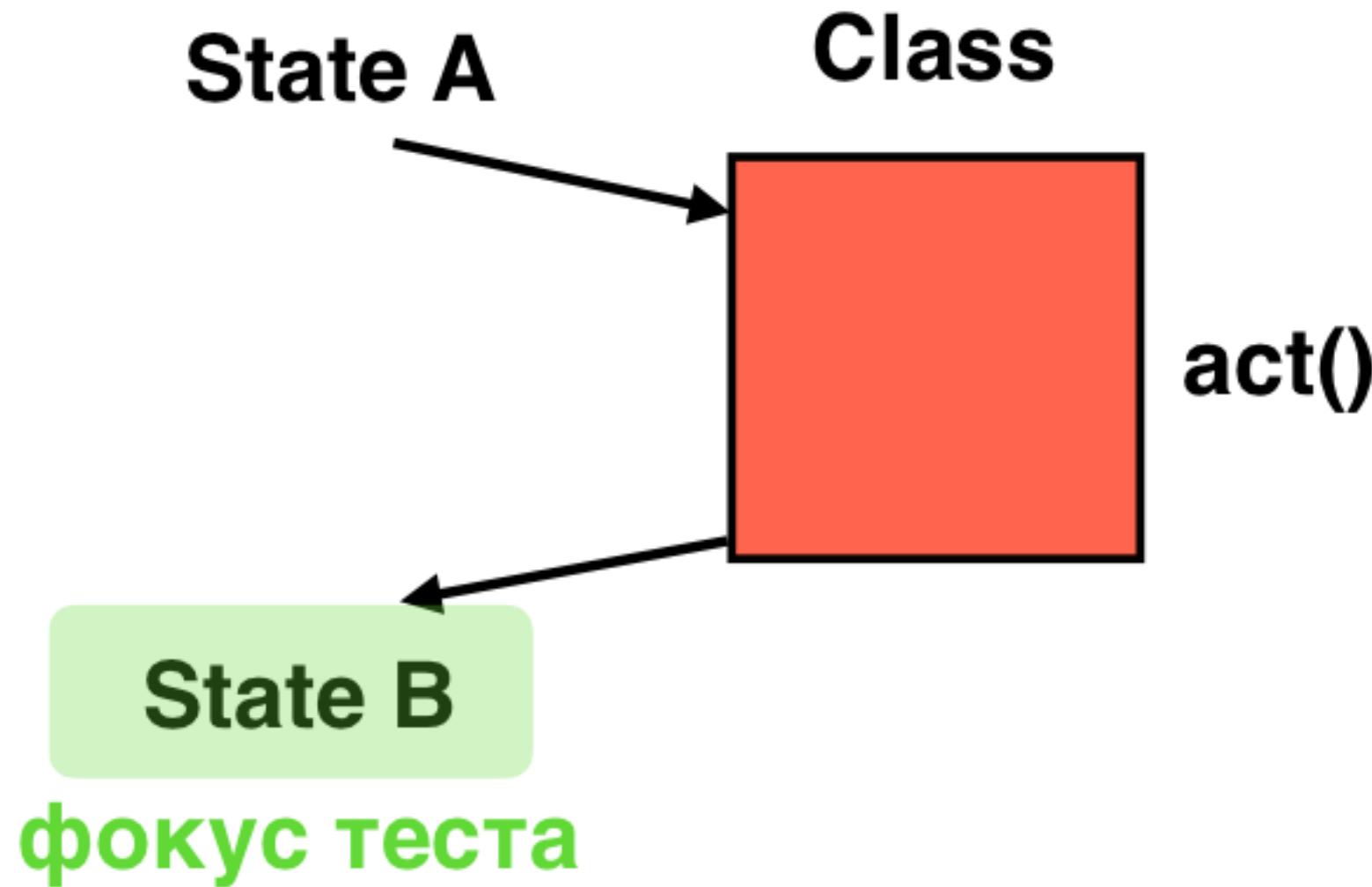
WhenHungry

WhenSick

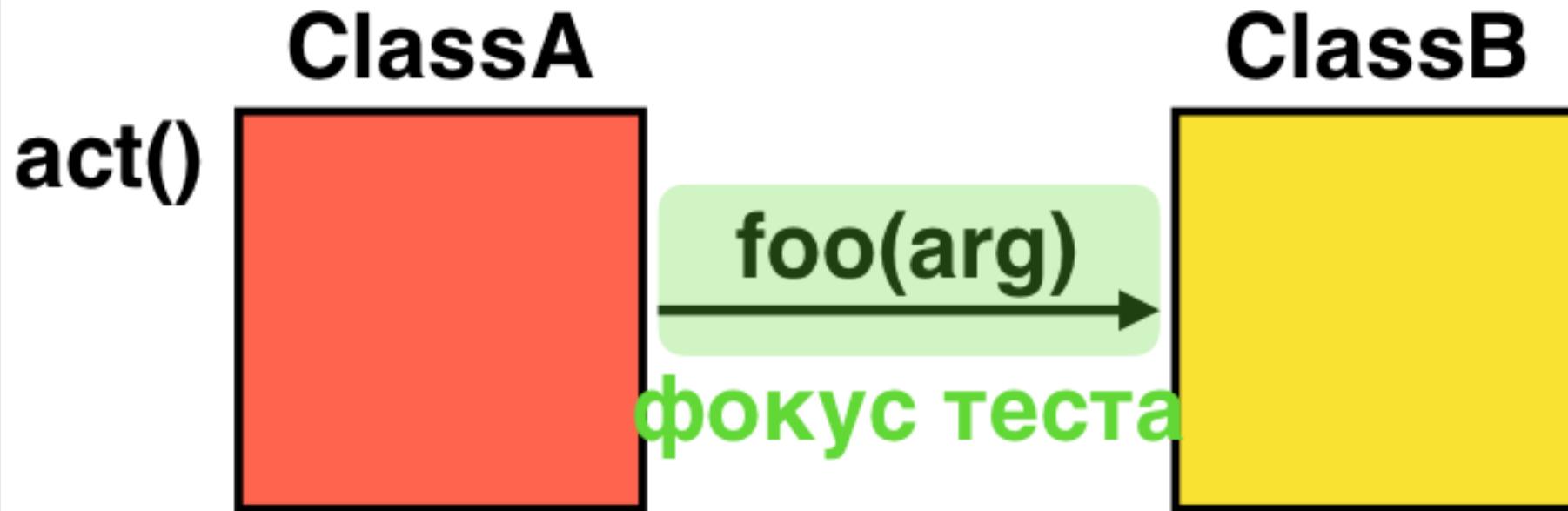
When...

# **Поведение и состояние**

# Тести на состояние



# Тесты на поведение



# хорошие юнит тесты

Короткие

Быстрые

Независимые

Друг от друга

От порядка

От окружения

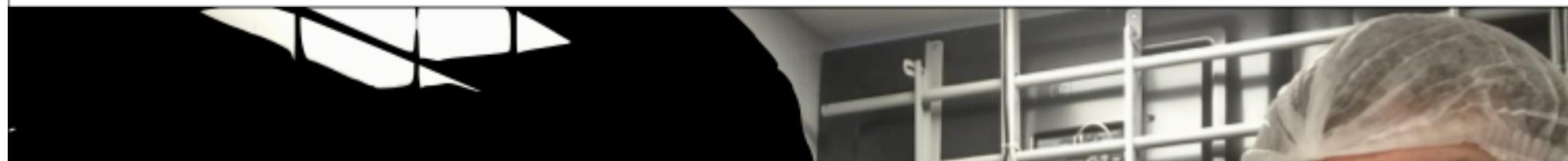
Понятные

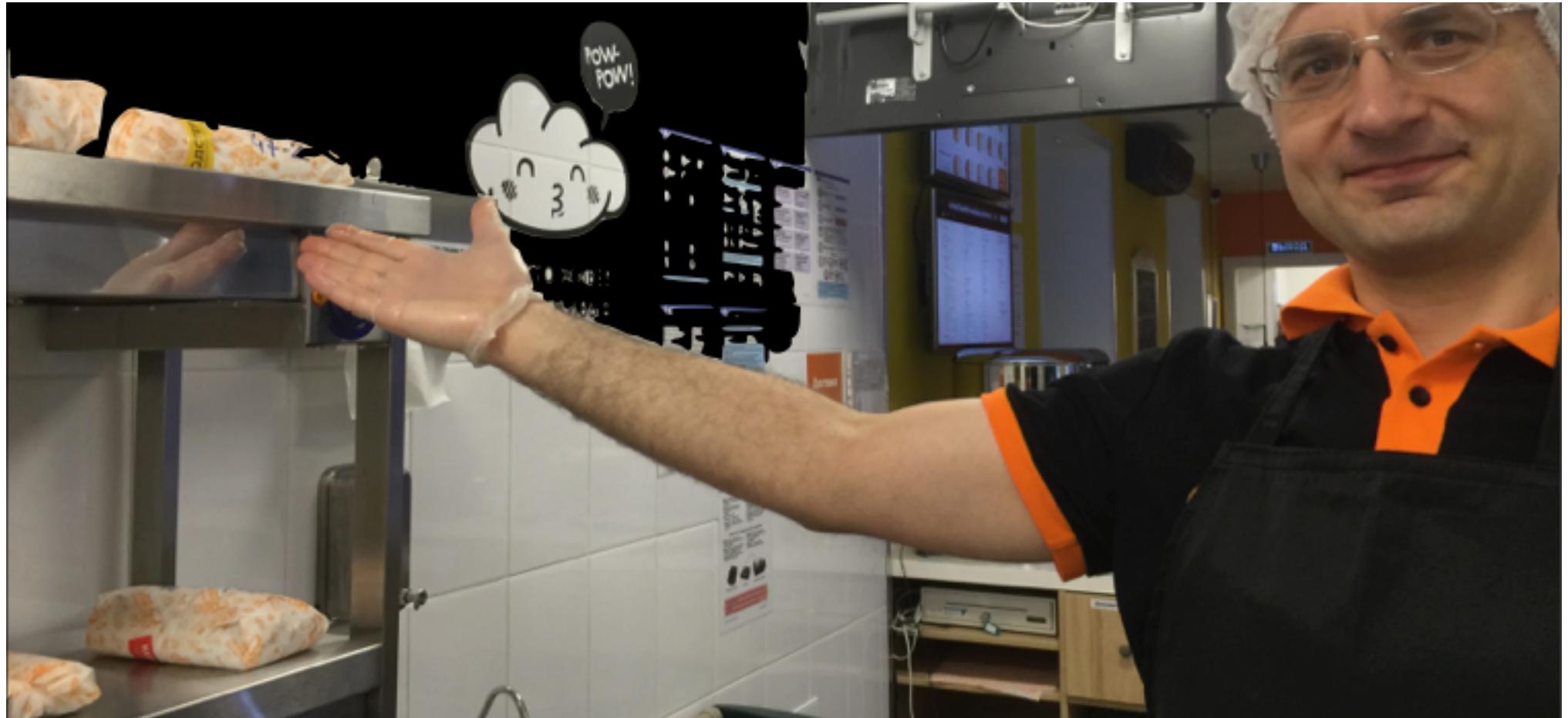
Название

Данные

Сфокусированы

Автоматические





a.bevzuk@**dodopizza.com**

Антон Бевзюк