

The logo for 'ag;je days' is contained within a dark blue speech bubble shape. The text 'ag;je' is on the top line and 'days' is on the bottom line, both in a white, bold, sans-serif font. The semicolon in 'ag;je' is stylized with a right-pointing arrowhead.

ag;je  
days

# 10 шагов к лучшему качеству кода приемочных тестов

Вагиф Абилов

Miles Oslo

# Коротко обо мне

---

- ▶ Электронная почта: [vagif.abilov@gmail.com](mailto:vagif.abilov@gmail.com)
- ▶ Твиттер: [@ooobject](https://twitter.com/ooobject)
- ▶ GitHub: [object](https://github.com/object)
- ▶ BitBucket: [object](https://bitbucket.org/object)
- ▶ Блог: <http://vagifabilov.wordpress.com/>
- ▶ Статьи: <http://www.codeproject.com>



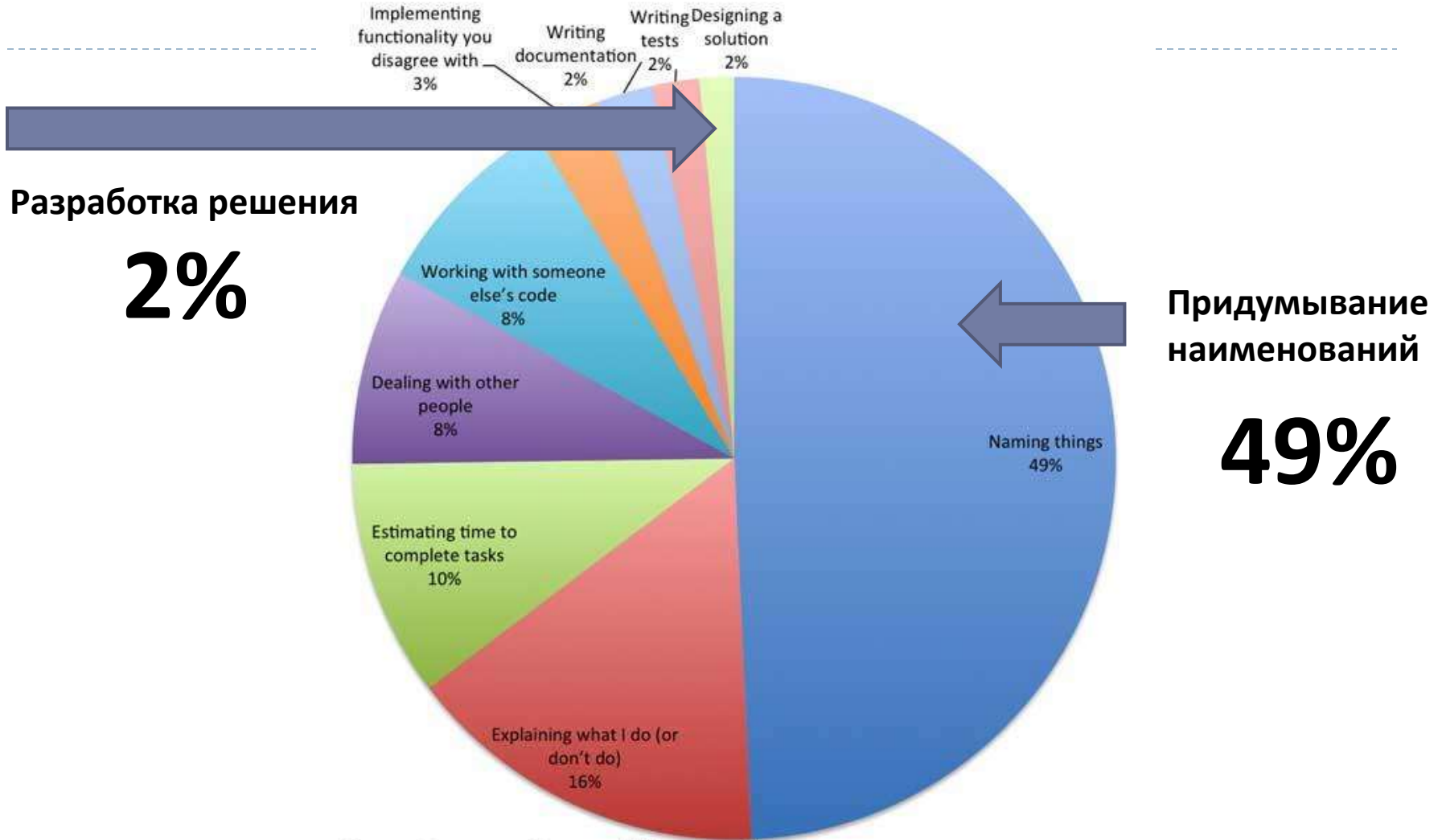
# План презентации

---

- ▶ Насколько хорош код тестов? Или насколько плох?
- ▶ Тестеры как полноправные граждане
- ▶ 10 шагов, ведущих к улучшению кода тестов
- ▶ Примеры и случаи из практики



# Programmers' Hardest Tasks



Разработка решения

2%

Придумывание названий

49%

Data Source: Quora/Ubuntu Forums  
Total Votes: 4,522



# Насколько хорош код тестов? Или насколько плох? Небольшой опрос

---

- ▶ **Какое качество вы ожидаете от кода тестов?**
  - ▶ Код тестов обычно уступает в качестве коду продукта
  - ▶ Код тестов обычно превосходит в качестве код продукта
  - ▶ Качество кода тестов и продукта обычно находятся на одном уровне
- ▶ **Как долго живет код тестов?**
  - ▶ Код тестов живет столько же, сколько код продукта
  - ▶ Код тестов удаляется или переписывается раньше кода продукта
  - ▶ Код тестов живет дольше кода продукта



# Тестеры как полноправные граждане

---

- ▶ Все, что выпускается, публикуется или отсылается заказчику, обычно является предметом более строгих требований, чем программы для внутреннего использования
- ▶ Внутренности продукта подчиняются внешним требованиям, внутренности тестов подчиняются внутренним требованиям
- ▶ Руководство зачастую относится менее строго к выбору тестерами средств и языков программирования
- ▶ Нет причин, по которым тестеры не должны обращать в свою пользу либерализм руководства - тестеры ПО обладают большими возможностями адаптирования новых методов и технологий



# 10 шагов к лучшему качеству кода приемочных тестов

---

- ▶ Преодолевайте барьеры понимания
- ▶ Пользуйтесь исполняемыми спецификациями
- ▶ Дайте актерам действовать в сценариях
- ▶ Спустите КАК на нижние уровни
- ▶ Уменьшайте число уровней трансляции
- ▶ Пользуйтесь исследовательскими скриптами
- ▶ Группируйте код автоматизации тестов в библиотеки
- ▶ От DRY к DRO - следите за сообществами
- ▶ Пробуйте динамические языки
- ▶ Восстанавливайте лишь существенные хранимые состояния



# 1. Преодолевайте барьеры понимания

---

- ▶ «Для метода BDD тестирование не является первичным. Вы получите из него тесты, но обычно потребуется и дополнительное тестирование. Куда важнее в этом разработка и понимание того, что должно стать конечным результатом.» – Лиз Кеог
- ▶ «Обсуждение является гораздо более важным, чем инструментальные средства BDD» – Лиз Кеог





## 2. Пользуйтесь исполняемыми спецификациями (executable specifications)

---

**Scenario:** The answer with the highest vote gets to the top

**Given** there is a question «What is your favourite color» with the answers

Answer	Vote
Red	1
Green	1

**When** I upvote the answer «Green»

**Then** the answer «Green» should be on the top

---



## 2. Пользуйтесь исполняемыми спецификациями (executable specifications)

---

**Сценарий:** Ответ с наибольшим количеством голосов выдается первым

**Дано** вопрос «Какой ваш любимый цвет?» с ответами

Ответ	Голосов
Красный	1
Зеленый	1

**Если** я отдам голос за «Зеленый»

**То** ответ «Зеленый» должен стать первым в списке ответов



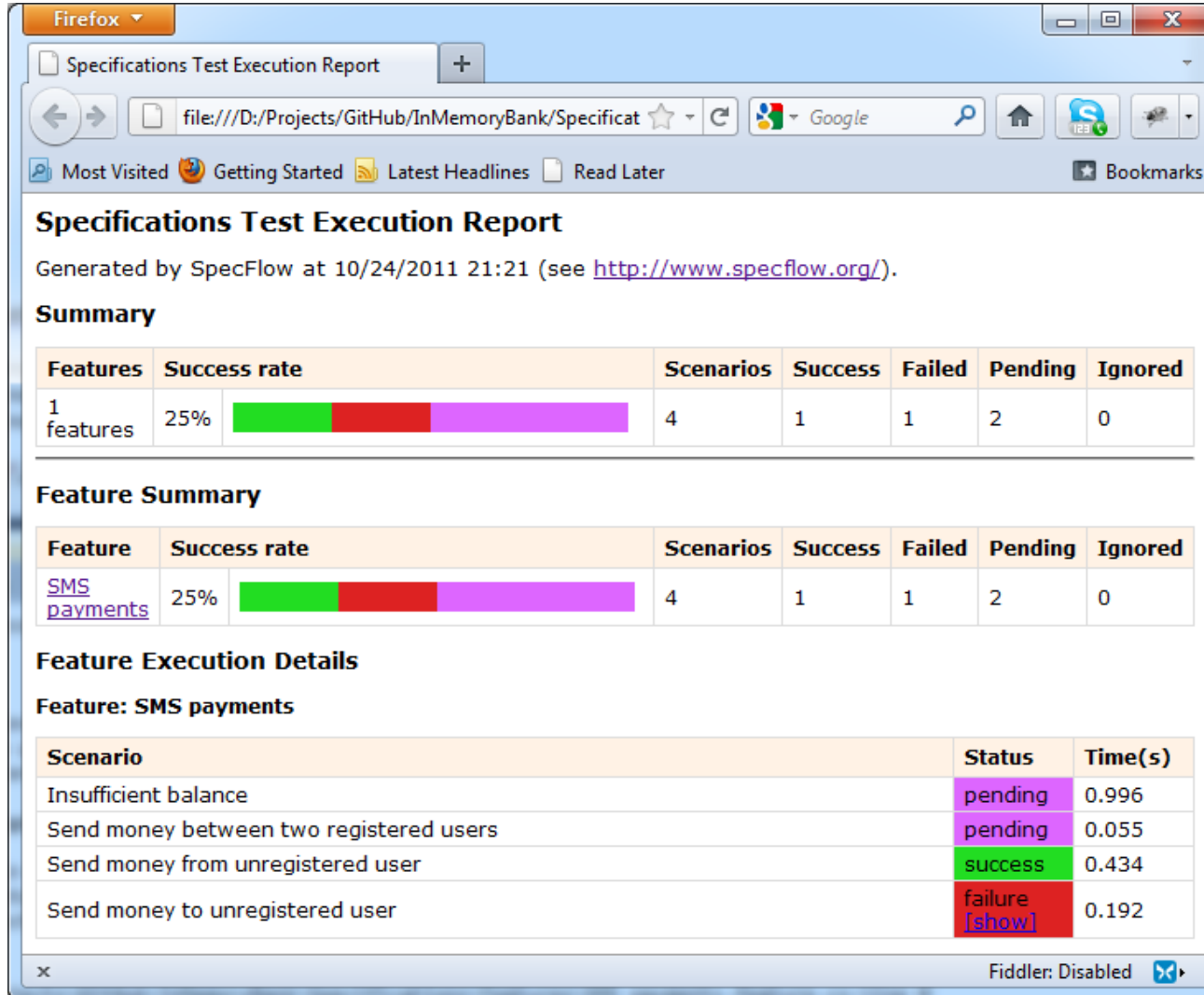
# Преимущества исполняемых спецификаций

---

- ▶ Пишутся на естественном языке (и своем языке)
- ▶ Формализуют требования к продукту
- ▶ Интегрируются в ежедневные билды
- ▶ Живая документация
- ▶ Показывают прогресс разработки
- ▶ Функциональное покрытие



# Функциональное покрытие



Firefox

Specifications Test Execution Report


file:///D:/Projects/GitHub/InMemoryBank/Specificat

Most Visited Getting Started Latest Headlines Read Later Bookmarks


## Specifications Test Execution Report

Generated by SpecFlow at 10/24/2011 21:21 (see <http://www.specflow.org/>).

### Summary

Features	Success rate	Scenarios	Success	Failed	Pending	Ignored
1 features	25% 	4	1	1	2	0

### Feature Summary

Feature	Success rate	Scenarios	Success	Failed	Pending	Ignored
<a href="#">SMS payments</a>	25% 	4	1	1	2	0

### Feature Execution Details

Feature: SMS payments

Scenario	Status	Time(s)
Insufficient balance	pending	0.996
Send money between two registered users	pending	0.055
Send money from unregistered user	success	0.434
Send money to unregistered user	failure <a href="#">[show]</a>	0.192

x Fiddler: Disabled

# Покрытие юнит-тестами

Tests run: 438, Failures: 0, Not run: 2, Time: 54.208 seconds

All Tests Passed

Warning Lfx.Core.Tests.TestAssemblyConfig.AssemblyNamedConfig

Warning Lfx.Core.Tests.TestAssemblyConfig.AssemblyNamedConfig\_InvalidDocument

## Warning Details (2)

Test: Lfx.Core.Tests.TestAssemblyConfig.AssemblyNamedConfig

Type: Warning

Message: TypeMock does not support mocking classes from mscorlib

Test: Lfx.Core.Tests.TestAssemblyConfig.AssemblyNamedConfig\_InvalidDocument

Type: Warning

Message: TypeMock does not support mocking classes from mscorlib

## NCoverExplorer Code Coverage Summary:

Module	Coverage %	Acceptance %	Verdict
Lfx.Core.DLL	98.6%	0%	PASS
Lfx.Data.DLL	100.0%	0%	PASS
Lfx.Global.DLL	95.8%	0%	PASS
Lfx.Msmq.DLL	100.0%	0%	PASS
<b>Total</b>	<b>97.1%</b>	<b>100%</b>	<b>FAIL</b>

Unvisited Sequence Points: 59 Files: 45

Total Sequence Points: 2069 Classes: 61

Non Comment Lines: 2109 Functions: 479

Module	Acceptance	Unvisited Functions	Function Coverage
Lfx.Core.DLL	0.0 %	4	98.6 %
<b>Namespace / Classes</b>			
Lfx.Core	0	4	98.6 %
CcAppDomain	0	0	100.0 %
CcArgument	0	0	100.0 %
CcArgumentList	0	0	100.0 %
CcAssemblyConfigAttribute	0	0	100.0 %
CcAssemblyResolver	0	0	100.0 %
CcCache	0	0	100.0 %
CcComponentCategory	0	0	100.0 %
CcConfigAction	0	0	100.0 %
CcConfiguration'E	2	2	92.0 %
CcConfigurationBase	0	0	100.0 %
CcConfigurationSectionHandler	0	0	100.0 %
CcDatabaseConfiguration	0	0	100.0 %
CcDatabaseInspection	0	0	100.0 %
CcEventCategory	0	0	100.0 %
CcException	0	0	100.0 %
CcExtensionMethods	0	0	100.0 %
CcInvalidArgumentException	0	0	100.0 %
CcKeywordExpander	0	0	100.0 %
CcLocalizableClass	0	0	100.0 %
CcLog	3	3	93.8 %
CcLogConfigurationBase	0	0	100.0 %
CcLogConfiguration	0	0	100.0 %
CcLogConfigurationBase	0	0	100.0 %
CcMemory	0	0	100.0 %
CcNoCoverageAttribute	0	0	0.0 %
CcQueueConfiguration	0	0	100.0 %
CcQueueConfiguration	0	0	0.0 %
CcSystemInspector	0	0	100.0 %
CcTrace	0	0	100.0 %
CcTrace	0	0	100.0 %
CcType	0	0	100.0 %
CcTypeComponentAttribute	0	0	100.0 %
CcWebResourceSectionHandler	0	0	100.0 %
IcConfigurationProvider	0	0	0.0 %

# Демонстрация исполняемых спецификаций

---

## Relish

Публикация исполняемых спецификаций

Примеры:

GDS/whitehall

brighterplanet/flight



# 3. Дайте актерам действовать в сценариях

---

Feature: Pay bill

Background: Prices

Given the following operations are available:

operation	price
routine check up	10
shots	5

Scenario: Dave pays for Fluffy

Given Dave owns a pet Fluffy

And Dave brings Fluffy into the clinic for the following operations:

routine check up
shots

When the veterinarian charges Dave for the visit

And Dave pays cash

Then Dave is given a receipt which looks like this:

Operations:

\$10 (routine check up)

\$5 (shots)

Total to pay: \$15

(из блога Мэта Уинна)



### 3. Дайте актерам действовать в сценариях

---

**Feature:** Pay bill

**Background:** Prices

**Given** the following operations are available:

operation	price
routine check up	10
shots	5





### 3. Дайте актерам действовать в сценариях

---

**Scenario:** Dave pays for Fluffy

**Given** Dave owns a pet Fluffy

**And** Dave brings Fluffy into the clinic for the following operations:

routine check up	
shots	

**When** the veterinarian charges Dave for the visit

**And** Dave pays cash

**Then** Dave is given a receipt which looks like this:

Operations:

\$10 (routine check up)

\$5 (shots)

Total to pay: \$15

---



# Пример из проекта

---

Добавление актера в сценарий



# Обновление результатов норвежских парламентских выборов

**Scenario:** Received report is saved and notification message is enqueued

**Given** table **RP07** doesn't contain reports for **46010000**

**And** notification queue is empty

**When** receival service receives report **RP07** with data

Element	Value	
<b>dateCreated</b>	<b>2013-05-01T10:01:02</b>	
<b>identifier</b>	<b>46010000</b>	
<b>mandates</b>	<b>Ap=6;H=5;V=4</b>	

**Then** receival service should return **OK**

**And** table **RP07** should contain report for **46010000**

**And** notification queue should contain message

Code	Identifier	Mandates	
<b>RP07</b>	<b>46010000</b>	<b>Ap=6;H=5;V=4</b>	

*(пример 1)*



# Обновление результатов норвежских парламентских выборов

---

**When** receival service receives report **RP07** with data

Element	Value	
<b>dateCreated</b>	<b>2013-05-01T10:01:02</b>	
<b>identifier</b>	<b>46010000</b>	
<b>mandates</b>	<b>Ap=6;H=5;V=4</b>	

**Then** receival service should return **OK**

**And** table **RP07** should contain report for **46010000**

**And** notification queue should contain message

Code	Identifier	Mandates	
<b>RP07</b>	<b>46010000</b>	<b>Ap=6;H=5;V=4</b>	

(пример 1)

---



# Переработанный сценарий

**Scenario:** Parliament overview is updated shortly upon the report receipt

**Given** I see following Parliament party mandates overview

Party	Mandates
Ap	3
H	2
V	1

**When** receipt service receives report **RP07** with data

Party	Mandates
Ap	6
H	5
V	4

**And** I wait 10 seconds

**Then** I should see following Parliament party mandates overview

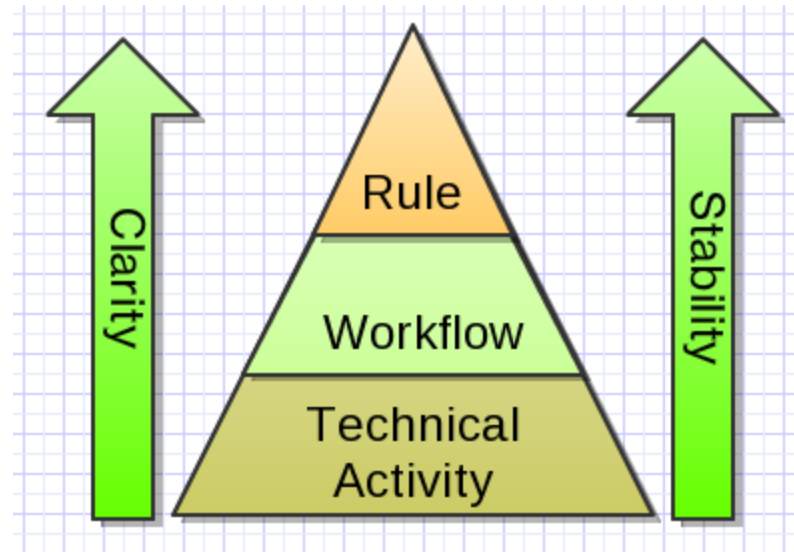
Party	Mandates
Ap	6
H	5
V	4



## 4. «Спустите КАК на нижние уровни»

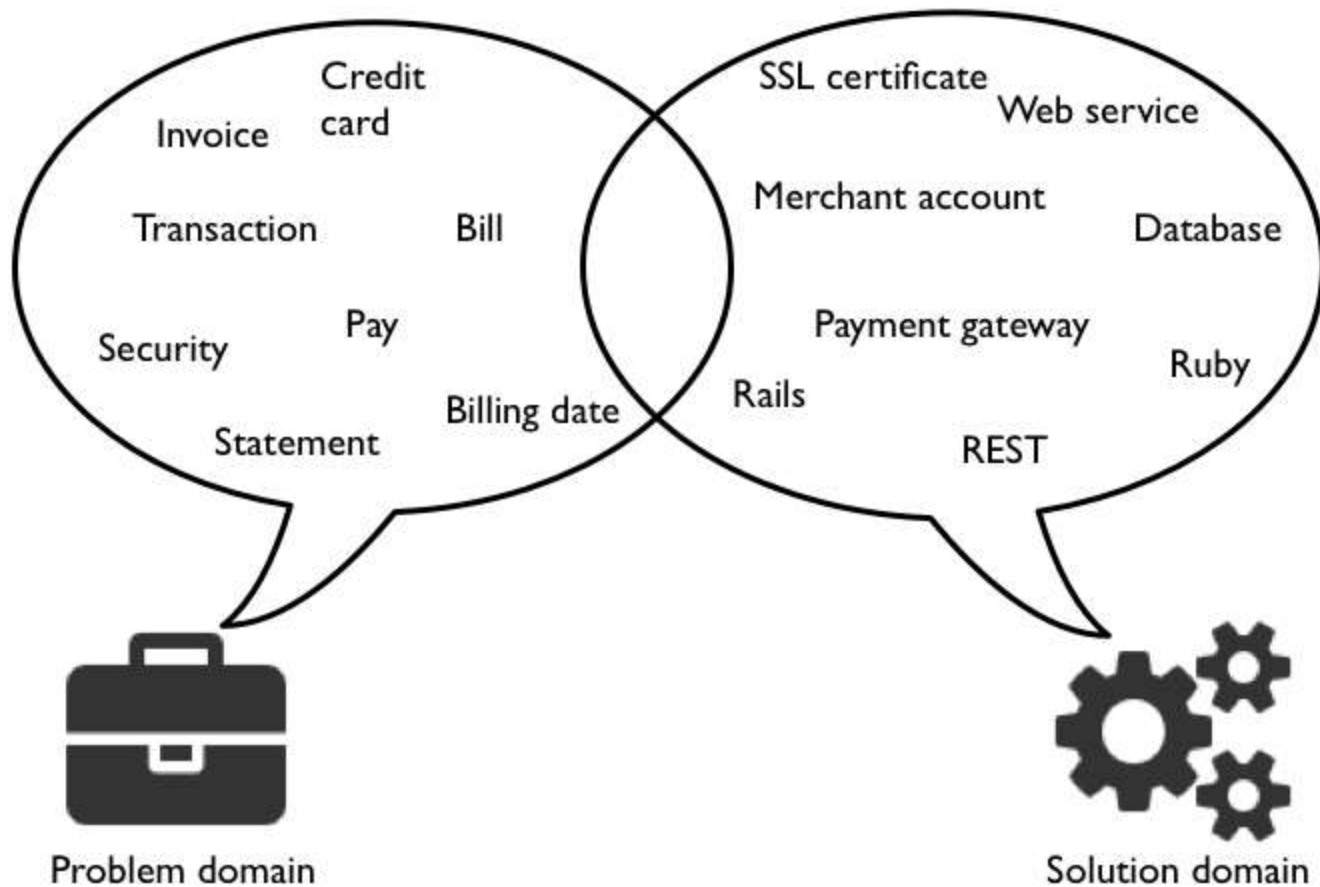
---

- ▶ Цитата из доклада Себа Роуза и Мэтта Уинна
- ▶ Гойко Аджич определил три уровня автоматизации тестов интерфейса пользователя:
  - ▶ Бизнес-правила
  - ▶ Интерфейс пользователя / рабочий процесс
  - ▶ Технический



# Спецификации отражают сущности домена

---



(Из блога Мэтта Уинна)

---



## 5. Уменьшайте число уровней трансляции

---

«Сценарии BDD транслируют пользовательские  
сущности в системные.

Чем больше работы  
производится в этих сценариях,  
тем больше вносится искажений в систему.»

@richarddalton





# В поисках наиболее эффективного языка для сценариев

---

**Scenario:** Refunded items should be returned to stock

**Given** a customer buys a black jumper

**And** I have **3** black jumpers left in stock

**When** he returns the jumper for a refund

**Then** I should have **4** black jumpers in stock



# Примеры реализации кода шагов сценария

---

1. C#/SpecFlow
2. Ruby
3. F#/TickSpec

*(Примеры 2-4)*



## 6. Пользуйтесь исследовательскими скриптами

---

- ▶ Не бросайтесь писать библиотеку автоматизации тестов, пока не исследуете систему
- ▶ Поищите инструменты REPL для языка ваших тестов
- ▶ Исследуйте систему из скриптов
- ▶ Воспользуйтесь результатами исследований для написания автоматизированных тестов
- ▶ Помните, что у ручных скриптов и автоматизированных тестов различные контексты, и обычно они не взаимозаменяемы



Демонстрация

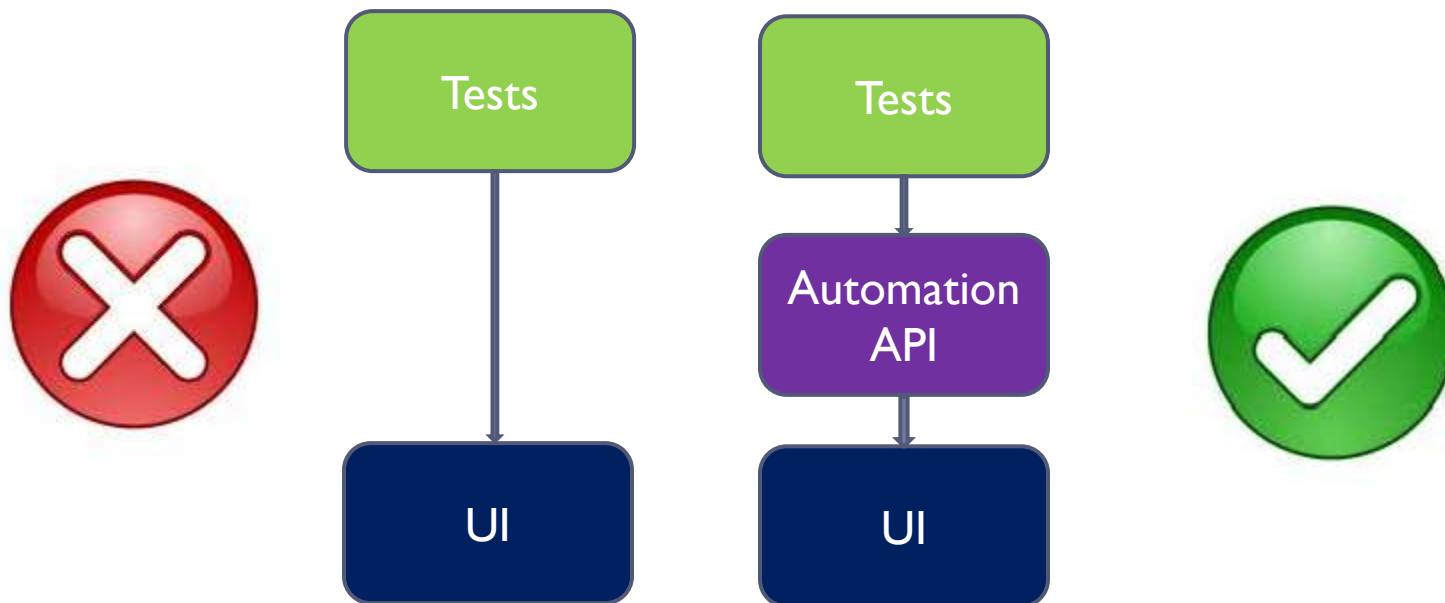
---

Скрипт для интерфейса  
пользователя  
на основе  
scriptcs и FluentAutomation



## 7. Группируйте код автоматизации тестов в библиотеки

- ▶ Если вы напрямую вызываете интерфейс WebDriver в ваших тестах, ваши тесты написаны неверно (Саймон Стюарт)



# «Page» в PageObject может спутать

---

- ▶ Несмотря на термин "объект страницы", эти объекты как правило определяются не для каждой страницы, а для существенного подмножества элементов страницы (Мартин Фаулер)
- ▶ Фаулер также вводит альтернативное определение: "панель"
- ▶ Названия классов объектов страниц могут не быть идентичны именам страниц HTML
- ▶ Вместо отображения отдельных страниц библиотека автоматизации тестов может моделироваться на основе действий пользователя

(Примеры 5-8)

---



# Пример: автоматизация поиска Google

---

**Feature:** Search for professional events

In order to keep updated about professional events

As a service subscriber

I want to be able to search for conferences

**Scenario:** Search for professional conference Web sites

**Given** I am a service subscriber

**When** I search for "Agile Testing Days"

**Then** I should receive results starting with  
"www.agiletestingdays.com"

---



# Паттерн страницы или паттерн поведения?

---

- ▶ Паттерн страницы хорошо подходит для тестов регрессии и внешних сайтов
- ▶ В основе паттерна поведения - действия пользователя, он может дать преимущества, когда проектируемый сайт находится в ранней стадии разработки и может быть подвержен существенным структурным изменениям

*(Примеры 5-8)*

---





## 8. От DRY к DRO – следите за сообществами

---

- ▶ DRY – Don't repeat yourself
- ▶ DRO – Don't repeat others
- ▶ Различные (и бесплатные!) библиотеки уже удовлетворили большинство ваших нужд в форматировании, конвертировании и передачи данных
  - ▶ Генерация псевдослучайных данных
  - ▶ Конвертация таблиц сценариев
  - ▶ Сравнение содержимого таблиц с содержимым баз данных
  - ▶ Перевод DTO в динамические объекты



# Пример: проверка данных ВВОДА ПОЛЬЗОВАТЕЛЯ

---

**Scenario:** Validate users

**Given** the following users exists in the database:

Name	Birth date	Length in meters
John	1940-10-09	1.80
Paul	1942-06-18	1.80
George	1943-02-25	1.77
Ringo	1940-07-07	1.68

**Then** the following users must exist in the database:

John	1940-10-09	1.80
Paul	1942-06-18	1.80
George	1943-02-25	1.77
Ringo	1940-07-07	1.68

(Примеры 9-12)

---



## 9. Пробуйте динамические языки

---

- ▶ Меньше церемоний – больше действий
- ▶ Доступ к базам данных без необходимости создания промежуточных классов
- ▶ Доступ к веб-сервисам без необходимости импорта файлов WSDL
- ▶ Нет нужды в объектах передачи данных (DTO)

*(Примеры 13-15)*



# 10. Восстанавливайте лишь существенные хранимые состояния

---

- ▶ Тест не должен рассчитывать на определенное хранимое состояние
- ▶ После завершения теста не нужно восстанавливать первоначальное хранимое состояние
- ▶ Не тратьте время на сложные процедуры удаления взаимосвязанных данных
- ▶ Просто отодвиньте в сторону данные, которые мешают исполнению теста

*(Примеры 16-17)*

---



# Повторим еще раз

---

- ▶ Преодолевайте барьеры понимания
- ▶ Пользуйтесь исполняемыми спецификациями
- ▶ Дайте актерам действовать в сценариях
- ▶ Спустите КАК на нижние уровни
- ▶ Уменьшайте число уровней трансляции
- ▶ Пользуйтесь исследовательскими скриптами
- ▶ Группируйте код автоматизации тестов в библиотеки
- ▶ От DRY к DRO - следите за сообществами
- ▶ Пробуйте динамические языки
- ▶ Восстанавливайте лишь существенные хранимые состояния



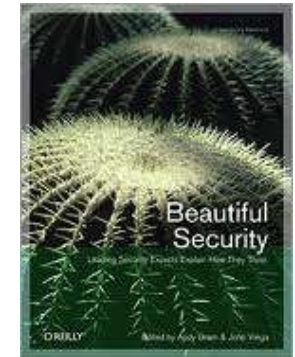
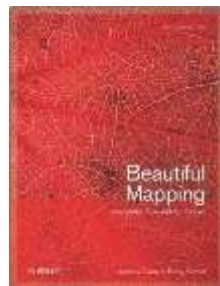
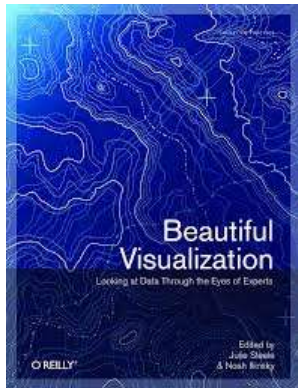
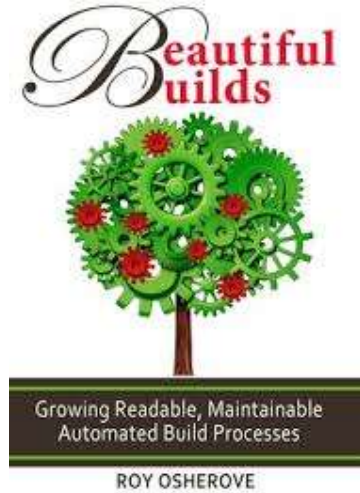
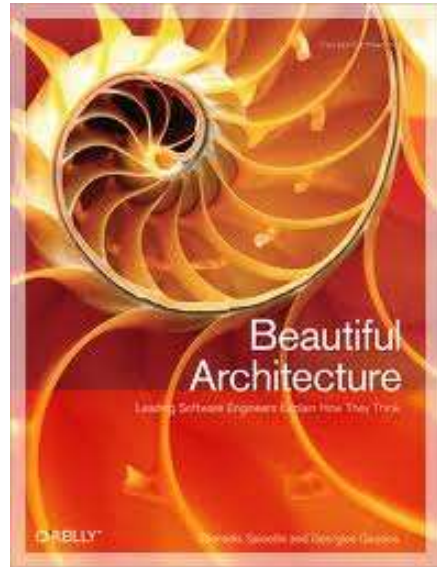
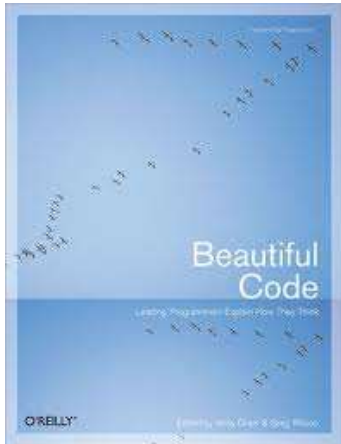
Время для караоке!

---

«Beautiful Code»

Christina Agile-Era





---

every day was so beautiful  
but suddenly i'm full of fears  
they just said we should work in pairs  
what have i done? it's just not fair





---

my code is beautiful  
don't touch it, it is mine!  
bugs won't bring me down  
my code is beautiful  
in every single line  
bugs won't bring me down  
but please don't run my code tonight

---



---

your singletons they are adorable  
they can be reached from anywhere  
your sql calls they are injectable  
to help somebody in despair



---

your code is beautiful  
i won't touch it, it is yours!  
bugs won't bring you down  
your code is beautiful  
in every single clause  
bugs won't bring you down  
but please don't run your code tonight

---



---

don't ask me why...



---

our code is beautiful  
don't touch it, it is ours!  
bugs won't bring us down  
our code is beautiful  
in every single class  
bugs won't bring us down  
but please don't run our code tonight

---



---

don't ask me why...  
it's beautiful



# 10 шагов к лучшему качеству кода приемочных тестов

---

Спасибо!

Вагиф Абилов  
vagif.abilov@gmail.com  
@ooobject  
GitHub: object  
BitBucket: object

