FitNesse in Development

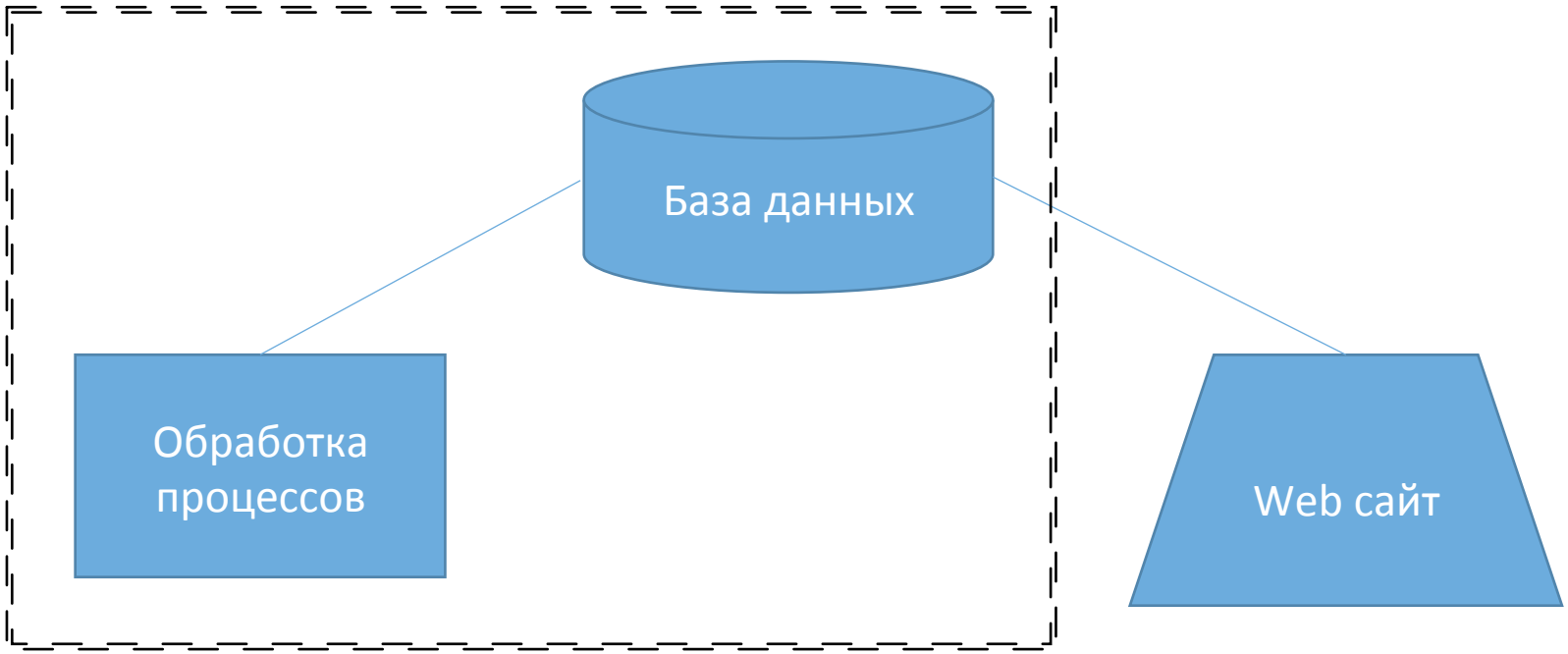
Манушин Игорь, Viacode LLC

.NEXT

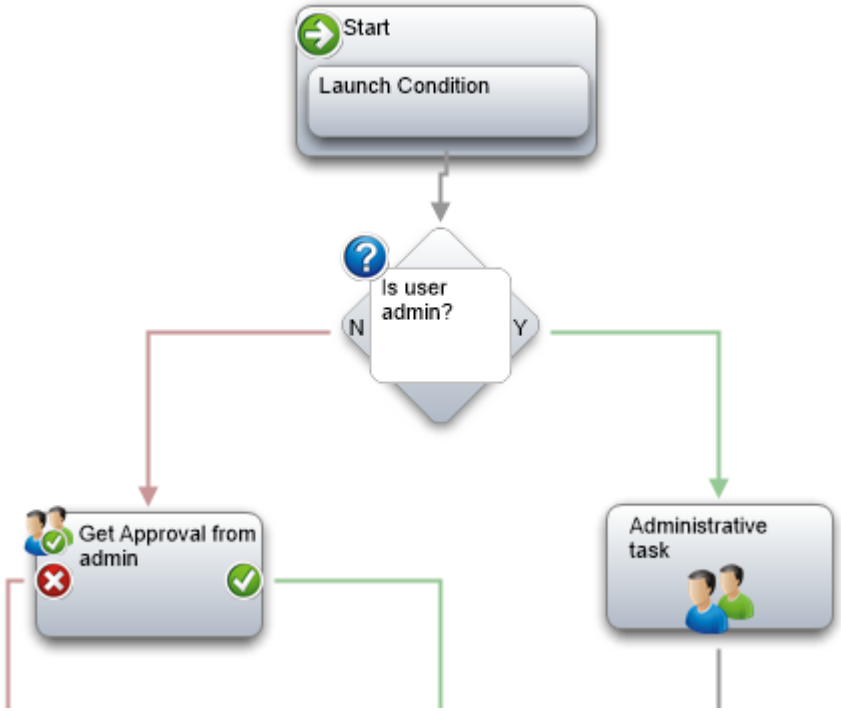
Правила

- Вопросы желательно задавать в конце
- Цифра в конце названия разделяет разные слайды с единой темой

Где применялось? (1)

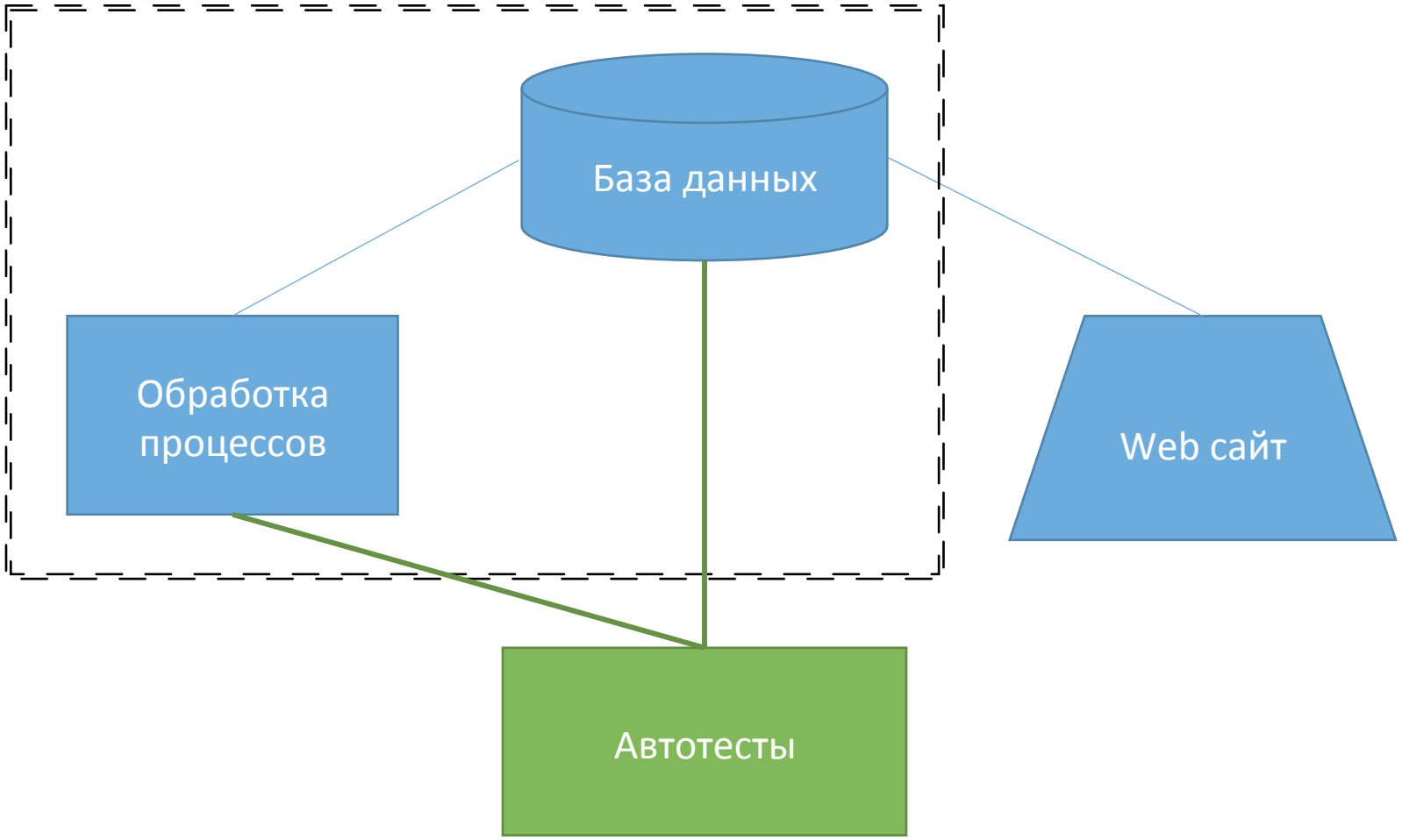


Где применялось? (2)



Где применялось (3)

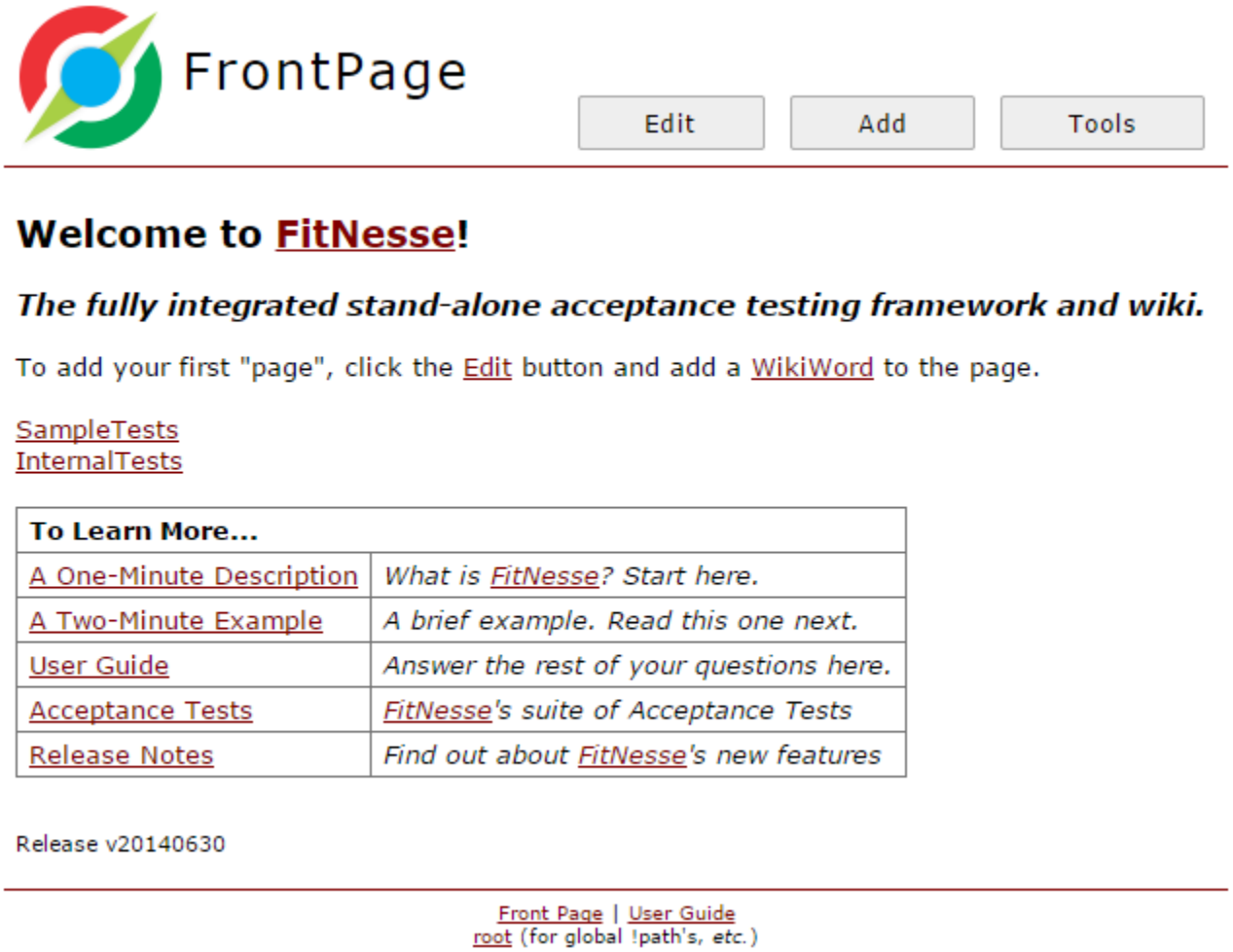
Где применялось? (4)



FitNesse (1)

- Интеграционное тестирование
- Первая версия: 2003 год, Роберт Мартин (на основе проекта fit)
- Wiki разметка для тестировщиков
- Свободный выбор языка программирования для разработчиков

FitNesse (2)


FrontPage
Edit
Add
Tools
Welcome to FitNesse!
The fully integrated stand-alone acceptance testing framework and wiki.
To add your first "page", click the Edit button and add a WikiWord to the page.
SampleTests
InternalTests
To Learn More...
A One-Minute Description
What is FitNesse? Start here.
A Two-Minute Example
A brief example. Read this one next.
User Guide
Answer the rest of your questions here.
Acceptance Tests
FitNesse's suite of Acceptance Tests
Release Notes
Find out about FitNesse's new features
Release v20140630
Front Page | User Guide
root (for global !path's, etc.)

FitNesse (3)

InternalTests > NetRunnerRegression

CollectionsTest

Test | Edit | Add | Tools

Included page: .InternalTests.NetRunnerRegression.SetUp (edit) | Expand | Collapse

variable defined: testName=".InternalTests.TestScenarios.CollectionsTests"

execute page	.InternalTests.TestScenarios.CollectionsTests

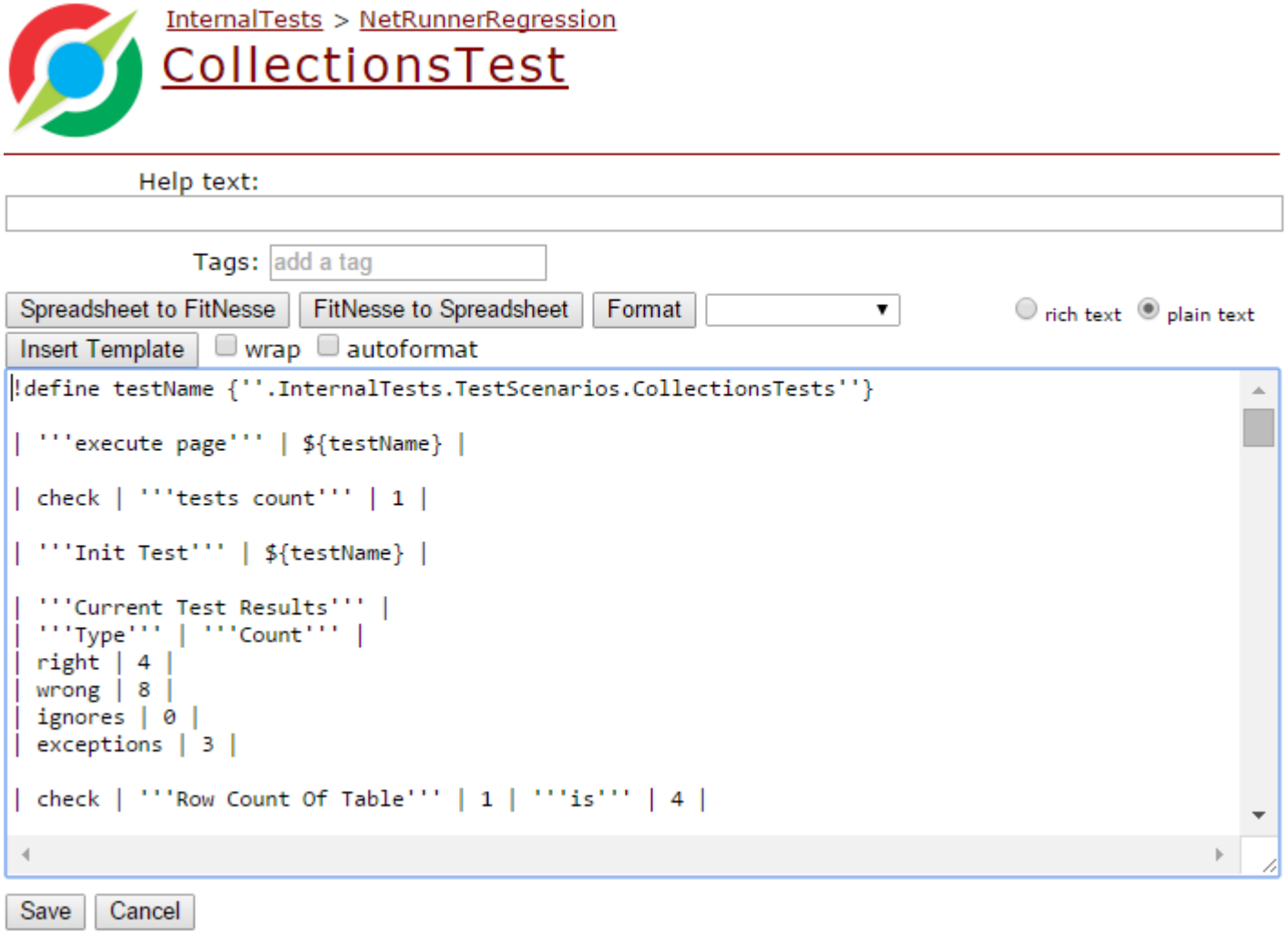
check	**tests count**	1

Init Test	.InternalTests.TestScenarios.CollectionsTests

Current Test Results	
Type	**Count**
right	4
wrong	8
ignores	0
exceptions	3

check	**Row Count Of Table**	1	**is**	4

FitNesse (4)


InternalTests > NetRunnerRegression
CollectionsTest
Help text:
Tags: add a tag
Spreadsheet to FitNesse
FitNesse to Spreadsheet
Format
rich text
plain text
Insert Template
wrap
autoformat
!define testName {''.InternalTests.TestScenarios.CollectionsTests''}
'''execute page'''	${testName}
check	'''tests count'''	1
'''Init Test'''	${testName}
'''Current Test Results'''
'''Type'''	'''Count'''
right	4
wrong	8
ignores	0
exceptions	3
check	'''Row Count Of Table'''	1	'''is'''	4
Save
Cancel

FitNesse (5)

InternalTests > NetRunnerRegression

CollectionsTest

Output Captured | Test | Edit | Add | Tools

Test Pages: 1 right, 0 wrong, 0 ignored, 0 exceptions **Assertions:** 35 right, 0 wrong, 0 ignored, 0 exceptions (17,784 seconds)

Test System: fit:..\binary\currentBuild\NetRunner.Executable.exe

▶ *Engine information*

▶ *Execution plan*

▶ *Included page: .InternalTests.NetRunnerRegression.SetUp (edit)* Expand | Collapse

variable defined: testName=".InternalTests.TestScenarios.CollectionsTests"

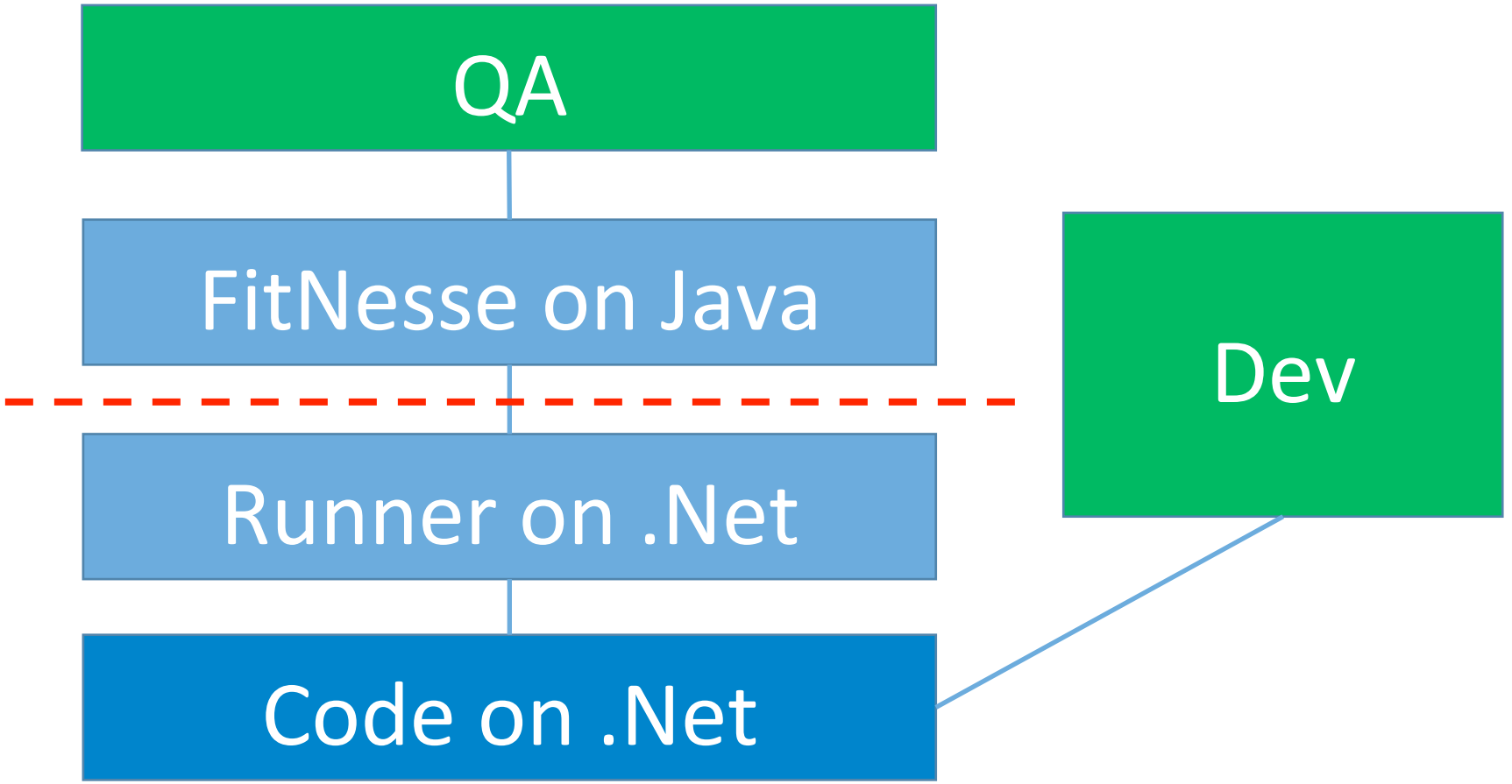
execute page	*.InternalTests.TestScenarios.CollectionsTests*

check	**tests count**	1

Init Test	*.InternalTests.TestScenarios.CollectionsTests*

Current Test Results	
Type	**Count**
right	4
wrong	8
ignores	0
exceptions	3

Как он устроен?



Пример кода

Wiki

```
• | “’ login as “’ | DonJoe |
• | “’ create company “’ | Example |
```

Html

login as	DonJoe

create company	Example

C#

```
public bool LoginAs(User userName)
    {
      /*...*/
    }
```

Начало - без тестов

- Environment настроен и запущен: сайт работает, сервисы работают
- Руками создаем алгоритм
- Система должна создать объекты, дождаться окончания, проверить статус

.NEXT

Почему FitNesse?

- Установка: Java + скопировать jar файл
- Запуск Web сервера: java -jar fitnesse-standalone.jar
- Запуск теста:
  - http://yourSever:8080/SuiteName/TestName?test
  - java -jar fitnesse-standalone.jar –c SuiteName/TestName?test
- Тесты и результаты – текстовые файлы

Первый тест

create matter	test matter

wait until status of	test matter	**will be**	Closed	**with timeout**	60	**seconds**

check	**status of matter**	test matter	**is**	Closed

Wait Until

- Если было хотя бы одно исключение – не ждем, сразу падаем
- Если упал хотя бы один предыдущий wait – не ждем, сразу падаем
- Всегда ждем с timeout
- Раз в секунду проверяем ответ

wait until status of	test matter	**will be**	Closed	**with timeout**	60	**seconds**

Check

- public MatterStatus StatusOfMatterIs(string matterName) { … }
- Возвращаем результат. FitNesse напишет ожидаемый и реальный результаты.

check	**status of matter**	test matter	**is**	Closed

Добавляем еще тесты

delete workflow	Test Definition

create definition	Test Definition	**for**	Matter

add single condition	Test IfElse	**with condition**	[Matter].[Status] == Open

add user task	Test User Task	**with following participants**
Name	**Type**	
Don Joe	Individual	
Administrators	Group	

save definition	Test Definition

add trigger on	Matter Created	**for**	Test Definition

Имена

- Переменная, которая хранит имя страницы - ${PAGE_NAME}
- Все названия можно сделать производными от этой переменной:
  - define workflow_name = ${PAGE_NAME} _workflow
  - define matter_name = ${PAGE_NAME}_matter

Удаление

- Начало каждого теста – удаление предыдущих запусков
- Удаление точечное – игнорируем все объекты, которые нам не мешают
- После всех тестов все результаты будут еще в базе

Контекст (1)

delete workflow	Test Definition

create definition	Test Definition	for	Matter

add single condition	Test IfElse	with condition	[Matter].[Status] == Open

add user task	Test User Task	with following participants
Name	**Type**	
Don Joe	Individual	
Administrators	Group	

save definition	Test Definition

add trigger on	Matter Created	for	Test Definition

Контекст (2)

- Все тесты выполняются в одном домене, рабочие классы не пересоздаются
- Для большого количества действий над одним объектом храним его в контексте и не пишем имя

Таблица на вход (1)

add user task	Test User Task	**with following participants**
Name	**Type**	
Don Joe	Individual	
Administrators	Group	

- public **AddParticipantsFixture** AddUserTaskWithFollowingParticipants (string newTaskName)

Таблица на вход (2)

add user task	Test User Task	**with following participants**
Name	**Type**	
Don Joe	Individual	
Administrators	Group	

```
internal sealed class AddParticipantsFixture
{
        public AddPaticipant(string name, MemberType type)
        {
        }
}
```

.NEXT

Таблица на выход

history of	%Task%	**for instance**	Test Definition
Action	**Date**	**Details**	**Name**
Started	{today}	Started on {today}	User Task Activity
Assigned	{today}	Assigned to DJ on {today}	User Task Activity
Completed	{today}	Completed on {today} by DJ	User Task Activity

Список на вывод

```
public IEnumerable HistoryOfForInstance(…)
{
        return dbModel.GetHistory(…).Select( row=> new
            {
                Action = row.GetField<ActivityAction>(…),
                Date = new
                        DateTimeWrapper(row.GetField<string>(…) )
            }
}
```

Анализ строки

- Runner определяет тип:
  - У типа есть зарегистрированный Parser – вызываем его (только NetRunner)
  - У типа есть публичный статический метод Parse – вызываем его
- Сравнение происходит с типами, а не со строками

Подстановка даты (1)

```
public sealed class DateTimeWrapper
{
        private readonly string _line;
        public DateTimeWrapper(string str)
        {
                _line = str;
        }
```

Подстановка даты (2)

```
public static DateTimeWrapper
    Parse(string inputLine)
    {
        var date = DateTime.UtcNow.ToString();
        var replaced = inputLine.Replace(
            “{today}”,
            date);
        return new DateTimeWrapper(replace);
    }
```

Подстановка даты (3)

- Пишем {today}
- Сравниваем строки
- Можем сравнивать и объекты

history of	%Task%	**for instance**	Test Definition
Action	**Date**	**Details**	**Name**
Started	{today}	Started on {today}	User Task Activity
Assigned	{today}	Assigned to DJ on {today}	User Task Activity
Completed	{today}	Completed on {today} by DJ	User Task Activity

Подитог

- Простые команды на вход
- Списки на вход и выход
- Подстановка имени тестов в объекты
- И всё это – при понятных названиях !

Автозапуск

- Шаг в CI
- java.exe -jar fitnesse-standalone.jar –c SuiteName/TestName?test&format=xml
- С помощью xsd создаем html для писем и сайта
- Запуск дочерних процессов:
  - Через coverage tool
  - После остановки – сразу анализируем логи и выводим ошибки

Описание функций

- Doxygen
- Встроенные в NetRunner подсказки

Вопросы ?

- http://fitnesse.org
- NuGet: fitSharp & NetRunner
- Доклад: http://1drv.ms/1Bsgzcf

- igor.manushin@gmail.com