

SpecFlow: Начало

Дмитрий Верескун



Дмитрий Верескун



Старший разработчик (ROKO Labs)



Руководитель учебного центра (там же)



Telegram: d_vereskun
dmitry.vereskun@rokolabs.com



ЧТО ТАКОЕ BDD?

- $BDD = TDD + DDD + OOAD$
 - BDD – behavior-driven development
 - TDD – test-driven development
 - DDD – domain-driven design
 - OOAD – object-oriented analysis and design

КЛЮЧЕВЫЕ АСПЕКТЫ BDD

- Расширение TDD;
- DSL, основанный на естественном языке;
- Понятен не только программистам (тестировщик, аналитик, заказчик, ...);
- Оформляется в виде набора user stories.

Сценарий Пусть будет хорошо

Допустим есть кнопка "Сделать *хорошо*"

Когда я нажимаю на кнопку "Сделать *хорошо*"

Тогда становится *хорошо*

РЕАЛИЗАЦИИ

- **Cucumber** — популярный BDD-фреймворк для Ruby, Java, JavaScript;
- **Gherkin** — язык, на котором пишутся сценарии для Cucumber;
- **SpecFlow** — реализация аналога Cucumber для платформы .NET.

ПРИМЕР USER STORY (FEATURE) НА GHERKIN

```
Feature: Calc summation verification  
  In order to avoid silly mistakes  
  As a math idiot  
  I want to be told the sum of two numbers
```

```
@positive
```

```
Scenario: Add two numbers  
  Given I have entered 50 into the calculator  
  And I have entered 70 into the calculator  
  When I press add  
  Then the result should be 120 on the screen
```

- Ключевые слова
- Заголовок
- Тег
- Принцип AAA

КЛЮЧЕВЫЕ СЛОВА — ОСНОВНАЯ СТРУКТУРА

Feature: CalculatorFeature

In order to avoid silly mistakes

As a math idiot

I want to be told the sum of two numbers

Background:

Я просто комментарий. Добавь меня куда-нибудь)

Given I have a calculator

And I turn it on

- **Feature** — наименование user story;
- Описание user story (несколько строк, не парсится);
- **Background** — инициализация user story.

КЛЮЧЕВЫЕ СЛОВА — СЦЕНАРИЙ

- **@tagName** — тег (можно несколько)
- **Scenario** — заголовок сценария
- **Given** — блок Arrange
- **When** — блок Act
- **Then** — блок Assert
- **And, But** — заменяют любое из Given/When/Then

@positive

Scenario: Add two numbers

Given I have entered *50* into the calculator

And I have entered *70* into the calculator

When I press add

Then the result should be *120* on the screen

КЛЮЧЕВЫЕ СЛОВА – AND, BUT

Given I have entered *50* into the calculator
Given I have entered *70* into the calculator
When I press add
Then the result should be *120* on the screen
Then the button "*add*" is unavailable
Then the button "*clear*" is available



Given I have entered *50* into the calculator
And I have entered *70* into the calculator
When I press add
Then the result should be *120* on the screen
And the button "*add*" is unavailable
But the button "*clear*" is available

РЕАЛИЗАЦИЯ ШАГОВ

[Binding]

1 reference

```
public class CalcSteps
```

```
{
```

```
    private readonly CalculatorContext _calcContext;
```

0 references

```
    public CalcSteps(CalculatorContext calcContext)
```

```
    {
```

```
        _calcContext = calcContext;
```

```
    }
```

```
    [Given(@"I have entered (.*) into the calculator")]
```

0 references

```
    public void EnterNumber(int number)
```

```
    {
```

```
        _calcContext.Numbers.Add(number);
```

```
    }
```


РЕАЛИЗАЦИЯ ШАГОВ

- Атрибут [Binding];
- Регулярные выражения для связи с Gherkin;
- Хранение состояния между шагами в контексте;
- Служебные контексты: ScenarioContext, FeatureContext, ...;
- Встроенное внедрение контекста;
- Не пройденный assert приводит к завершению сценария.

СТАТИЧЕСКИЙ КОНТЕКСТ СЦЕНАРИЯ

```
public CalcSteps()
{
    ScenarioContext.Current["Numbers"] = new List<int>();
}

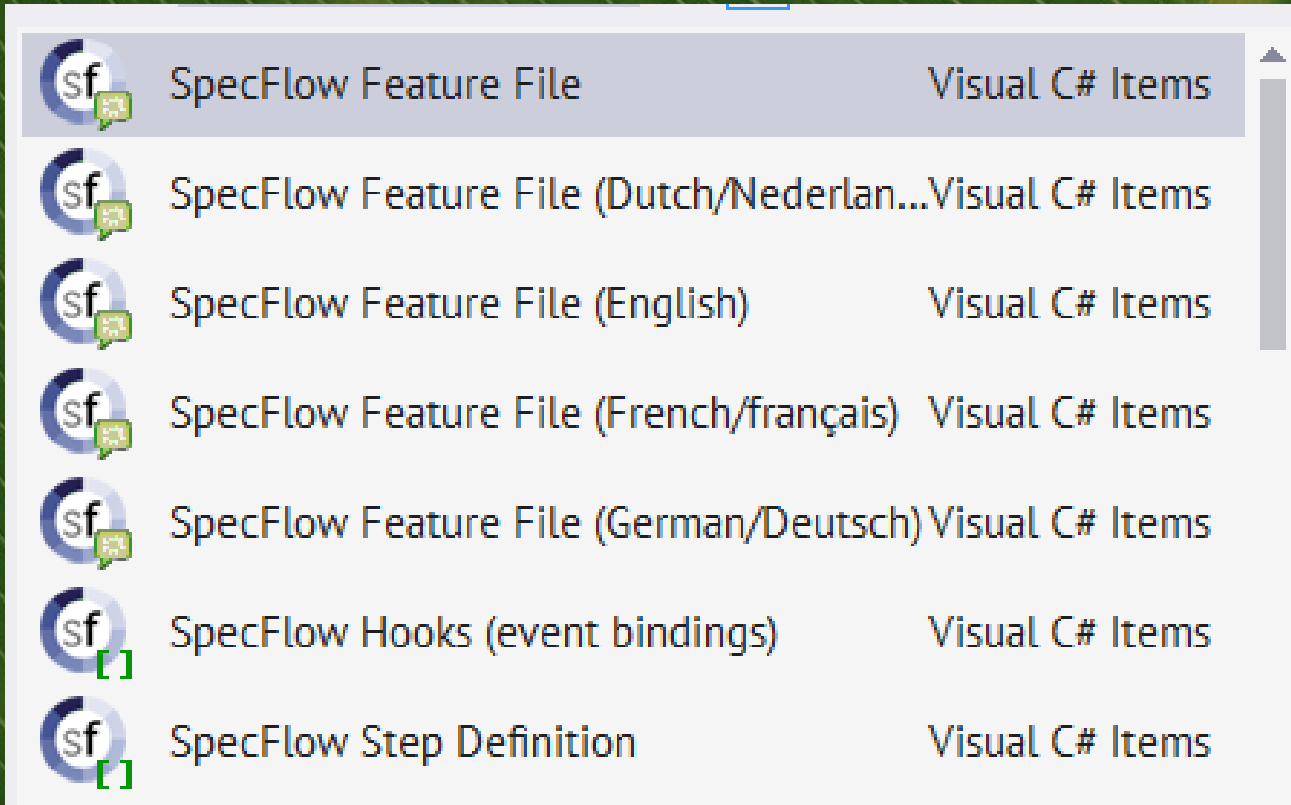
[Given(@"I have entered (.*) into the calculator")]
0 references
public void EnterNumber(int number)
{
    ScenarioContext.Current.Get<List<int>>("Numbers").Add(number);
}

[When(@"I press add")]
0 references
public void PressAdd()
{
    ScenarioContext.Current["Result"] =
        ScenarioContext.Current.Get<List<int>>("Numbers").Sum();
}
```


СПОСОБЫ ХРАНЕНИЯ СОСТОЯНИЯ

- Поля класса;
- Внедрение своего контекста через конструктор;
- Статические экземпляры встроенных контекстов;
- Внедрение встроенных контекстов;
- Наследование от `Steps` (есть свойства `ScenarioContext`, ...).

ШАБЛОНЫ FEATURE-ФАЙЛОВ В VISUAL STUDIO






- Файл user story (feature)
- Инфраструктурные шаги (hooks)
- Файл с реализацией шагов

ИНФРАСТРУКТУРНЫЕ ШАГИ (HOOKS)

- [BeforeTestRun] и [AfterTestRun]
- [BeforeFeature] и [AfterFeature]
- [BeforeScenario] и [AfterScenario]
- [BeforeScenarioBlock] и [AfterScenarioBlock]
- [BeforeStep] и [AfterStep]

АВТОГЕНЕРИРУЕМЫЙ ПРОМЕЖУТОЧНЫЙ КОД

```
#line 19
this.ScenarioSetup(scenarioInfo);
#line 6
this.FeatureBackground();
#line 20
testRunner.Given(string.Format("I have entered {0} into the calculator", a), ((string)
    (null)), ((TechTalk.SpecFlow.Table)(null)), "Given ");
#line 21
testRunner.And(string.Format("I have entered {0} into the calculator", b), ((string)
    (null)), ((TechTalk.SpecFlow.Table)(null)), "And ");
#line 22
testRunner.When("I press add", ((string)(null)), ((TechTalk.SpecFlow.Table)(null)),
    "When ");
#line 23
testRunner.Then(string.Format("the result should be {0} on the screen", sum), ((string)
    (null)), ((TechTalk.SpecFlow.Table)(null)), "Then ");
#line hidden
    this.ScenarioCleanup();
```

- ▶  CalculatorContext.cs
- ◀  CalculatorFeature.feature
- ▶  CalculatorFeature.feature.cs

ТАБЛИЦЫ ЗНАЧЕНИЙ

Scenario: Calculate price

Given I have entered into the calculator:

<i>Item</i>	<i>Price</i>	<i>Count</i>
Bread	25	2
Milk	40	1
Cheese	400	0.5

When I press sum

Then the result should be 290 on the screen

```
[Given(@"I have entered into the calculator:")]
```

0 references

```
public void EnterValues(Table priceTable)
```

```
{
```

```
    IEnumerable<PriceItem> priceItems = priceTable.CreateSet<PriceItem>();
```

```
    string firstItem = priceTable.Rows[0]["Price"];
```

```
}
```


МНОГОСТРОЧНЫЙ ТЕКСТ

```
When I execute SQL:  
...  
SELECT * FROM dbo.Users  
WHERE [Enabled] = 1  
ORDER BY [Name]  
...
```

```
[When(@"I execute SQL:")]  
0 references  
public void WhenIExecuteSql(string sql)  
{  
    // .....  
}
```


ШАБЛОНЫ ТЕСТОВ

- Заголовок Scenario Outline
- Блок Examples
- Можно заменять крупные фрагменты шагов

Scenario Outline: Add numbers

Given I have entered *<a>* into the calculator

And I have entered ** into the calculator

When I press add

Then the result should be *<sum>* on the screen

Examples:

<i>a</i>	<i>b</i>	<i>sum</i>
1	2	3
50	70	120
123	321	444

ЛОКАЛИЗАЦИЯ USER STORY

#language: ru-RU

Функция: Проверка сложения на калькуляторе

Для того, чтобы избежать глупых ошибок

Как рядовой пользователь

Я хочу проверить корректность сложения чисел

@ПОЗИТИВНЫЙ

Сценарий: Сложение двух чисел

Допустим Я набираю 50 на калькуляторе

И Я набираю 70 на калькуляторе

Когда Я нажимаю сложить

Тогда Результат 120 должен появиться на экране

ДОПОЛНИТЕЛЬНАЯ ИНФРАСТРУКТУРА

- Runners

- NUnit
- XUnit
- MSTest
- SpecRun (SpecFlow+)

- Adapters

- Report generators

- Большую часть плагинов можно установить как пакет NuGet или расширение Visual Studio.

The background features several green elements: a large light green circle in the top left, a smaller medium green circle in the top center, a large detailed green leaf with visible veins in the center, a smaller detailed green leaf to its left, and a bright lime green circle in the bottom left.

Спасибо за внимание!



Telegram: d_vereskun
dmitry.vereskun@rokolabs.com