

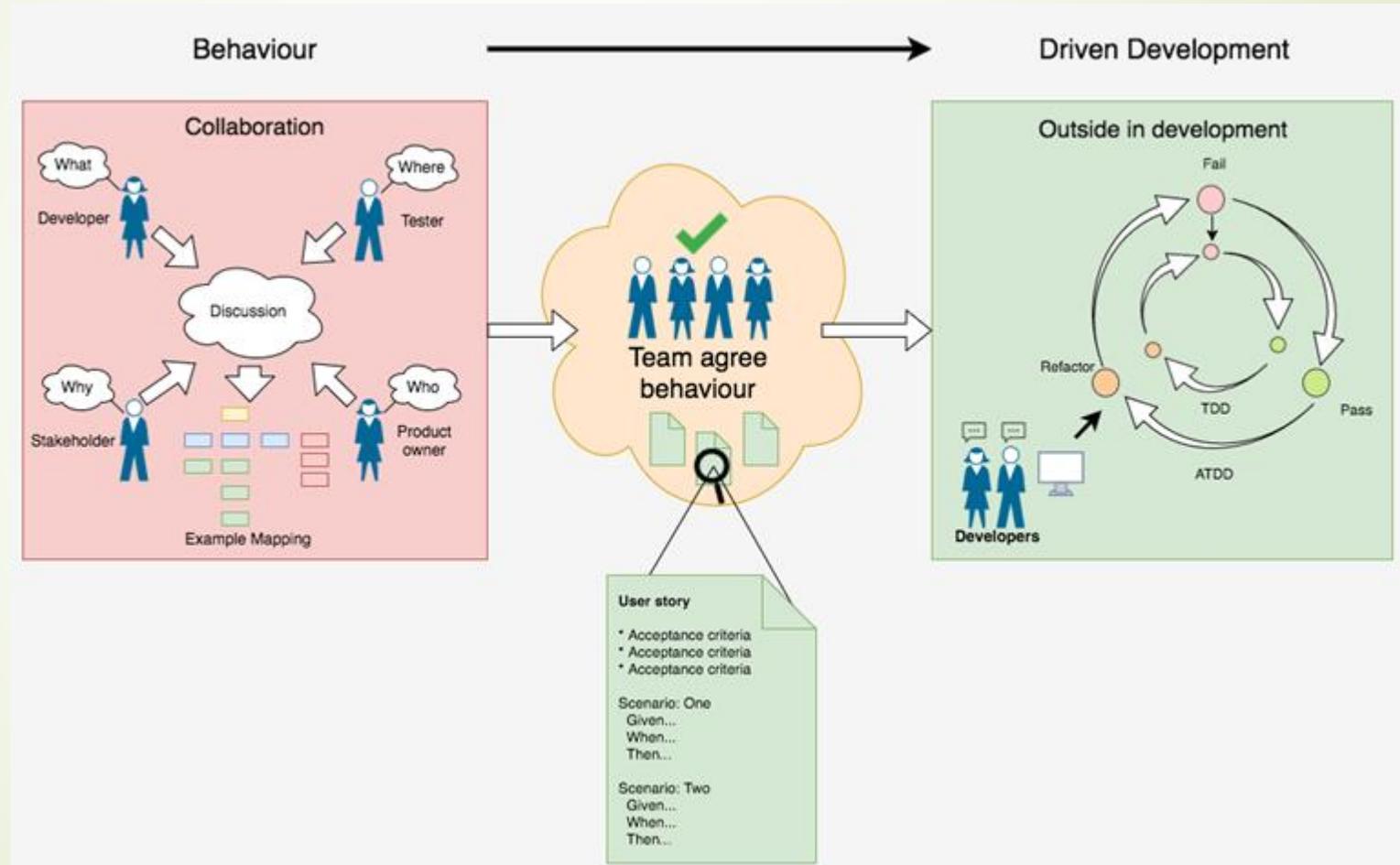


Использование SpecFlow при автоматизации тестирования Web- приложений

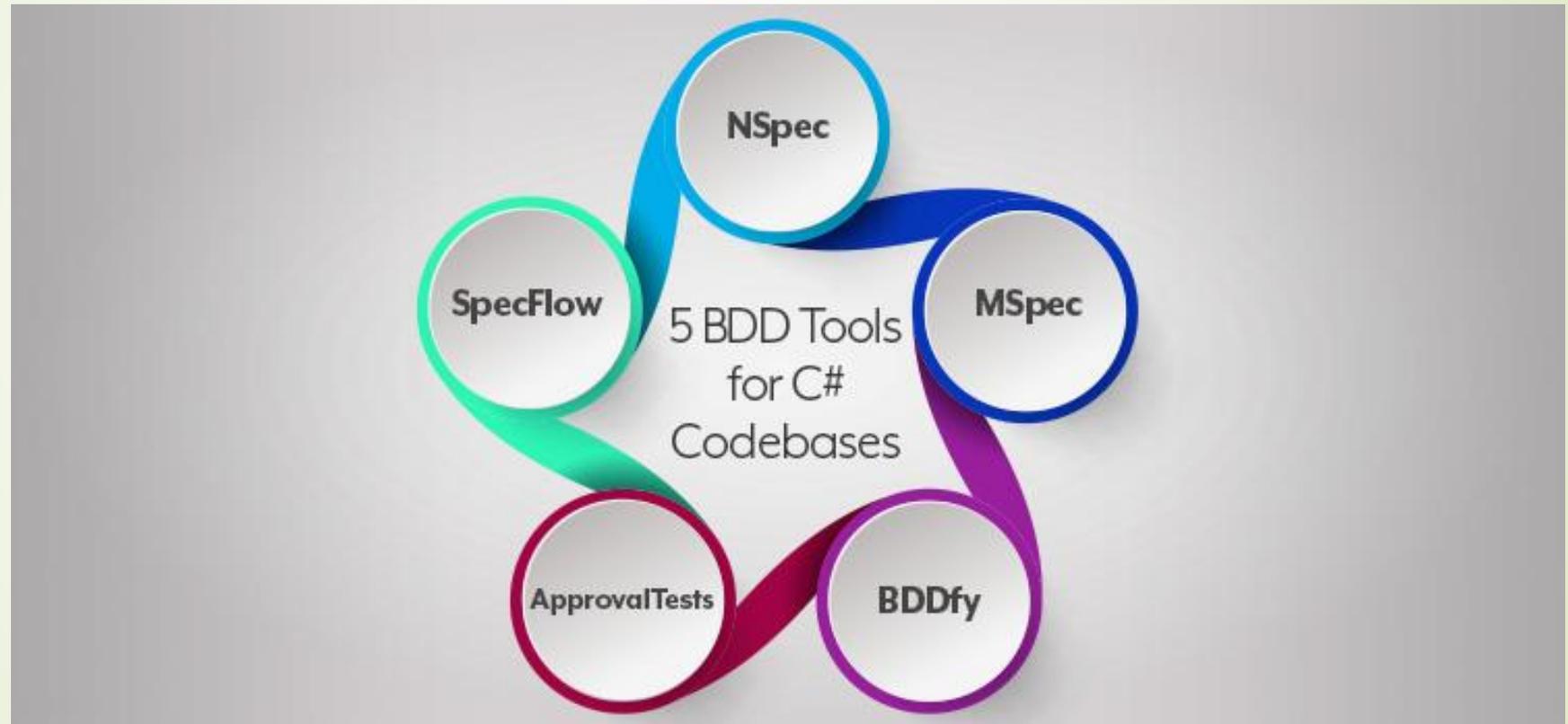
Докладчик: **Кулемеев Антон Григорьевич**

Инженер-программист ОАО «ТомскНИПИнефть»

Behavior-Driven Development



Примеры BDD фреймворков

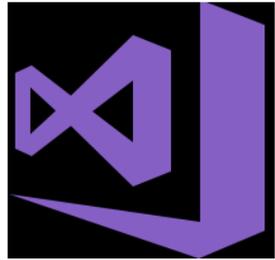


SpecFlow

- ▶ SpecFlow – это .NET средство, позволяющее писать спецификации с использованием языка Gherkin.
- ▶ Является аналогом Cucumber
- ▶ Позволяет использовать различные Unit Test Providers
- ▶ Возможны несколько вариантов интеграции

specflow 

Интеграция SpecFlow



specflow 



specflow 



specflow 

Ключевые слова языка Gherkin

- Feature
- Background
- Scenario
- Given
- When
- Then
- And
- But

```
Feature: SpecFlowFeature1
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

  @mytag
  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 языка Gherkin

Ключевое слово «Feature»

- ▶ Каждый feature-файл должен начинаться с этого ключевого слова.
- ▶ Определяет функциональные возможности теста, которые будут тестироваться в этом файле.
- ▶ Служит хорошей документацией к тесту.

```
Feature: LogIn Action
```

```
In order to check the credentials data
```

```
As a user
```

```
I want to be the authorization was successful
```

Ключевые слова языка Gherkin

Ключевое слово «Background»

- Ключевое слово «Background» используется для определения шагов, которые являются общими для всех сценариев в файле.

```
Feature: Basket on the website
```

```
Background: User is Logged In
```

```
  Given Navigate to Home Page
```

```
  And Click on the LogIn link
```

```
  When Enter UserName and Password
```

```
  And Click on Submit button
```

```
Scenario: Search a product and add the first result/product to the User basket
```

```
  Given User searched for Lenovo Laptop
```

```
  When Add the first laptop that appears in the search result to the basket
```

```
  Then User basket should display with 1 item
```

```
Scenario: Delete product from basket
```

```
  Given User find product
```

```
  When User click delete button
```

```
  Then the result should be
```

Ключевые слова языка Gherkin

Ключевое слово «Scenario»

- ▶ Каждый тест называется сценарием и описывается с помощью ключевого слова **«Scenario»**.
- ▶ Каждый сценарий состоит из трех частей:
 - Предусловие к тесту. Ключевое слово **«Given»**
 - Этап проверки выполнения. Ключевое слово **«When»**
 - Проверка выходных данных с ожидаемым результатом. Ключевое слово **«Then»**

```
Scenario: Successful Login
  Given User is on Home Page https://mail.yandex.ru
  When User enters UserName and Password
  And User clicked button SignIn
  Then User sees your Inbox
```

Ключевые слова языка Gherkin

Ключевое слово «Given»

- ▶ Ключевое слово «Given» определяет предварительное условие к тесту.

```
Given User is on Home Page https://mail.yandex.ru
```

Ключевые слова языка Gherkin

Ключевое слово «When»

- ▶ Ключевое слово «When» определяет тестовое действие, которое будет выполнено.

```
When User enter credentials
```

Key	Value
Username	super.testlogin2018
Password	Test@123

Ключевые слова языка Gherkin

Ключевое слово «Then»

- ▶ Ключевое слово «Then» определяет результат предыдущих шагов.

```
Then User sees your Inbox
```

Ключевые слова языка Gherkin

Ключевое слово «And»

- ▶ Ключевое слово «And» используется, чтобы добавить больше условий для «Given», «When» and «Then».

```
When User enters UserName and Password  
And User clicked button SignIn
```

Ключевые слова языка Gherkin

Ключевое слово «But»

- ▶ Ключевое слово «But» используется для добавления комментариев отрицательного типа.
- ▶ Это не жесткое правило.
- ▶ Имеет смысл использовать «But», когда вы попытаетесь добавить условие, которое противоположно тому, которое пытается установить ваш тест.

```
Scenario: Unsuccessful Login
```

```
Given User is on Home Page
```

```
When User Navigate to LogIn Page
```

```
And User enters UserName and Password
```

```
But The user credentials are wrong
```

```
Then Message displayed Wrong UserName & Password
```

Различия между Scenario Outline и Tables

► Scenario Outline:

- Используется ключевое слово «Examples» для определения тестовых данных.
- Работает для всего сценария.
- SpecFlow автоматически запускает тест полностью столько раз, сколько данных указано в тестовом наборе.

► Tables:

- Ключевое слово не используется для определения тестовых данных
- Работает только для одного шага, ниже которого он определен.
- Отдельный шаг должен принимать тестовый набор, а затем его можно запускать один или несколько раз, но только для этого шага.

Пример оформления Scenario Outline и Tables

```
Scenario Outline: Add two numbers
  Given I have entered <firstValue> into the calculator
  And I have entered <secondValue> into the calculator
  When I press <action>
  Then the <result> should be on the screen
```

Examples:

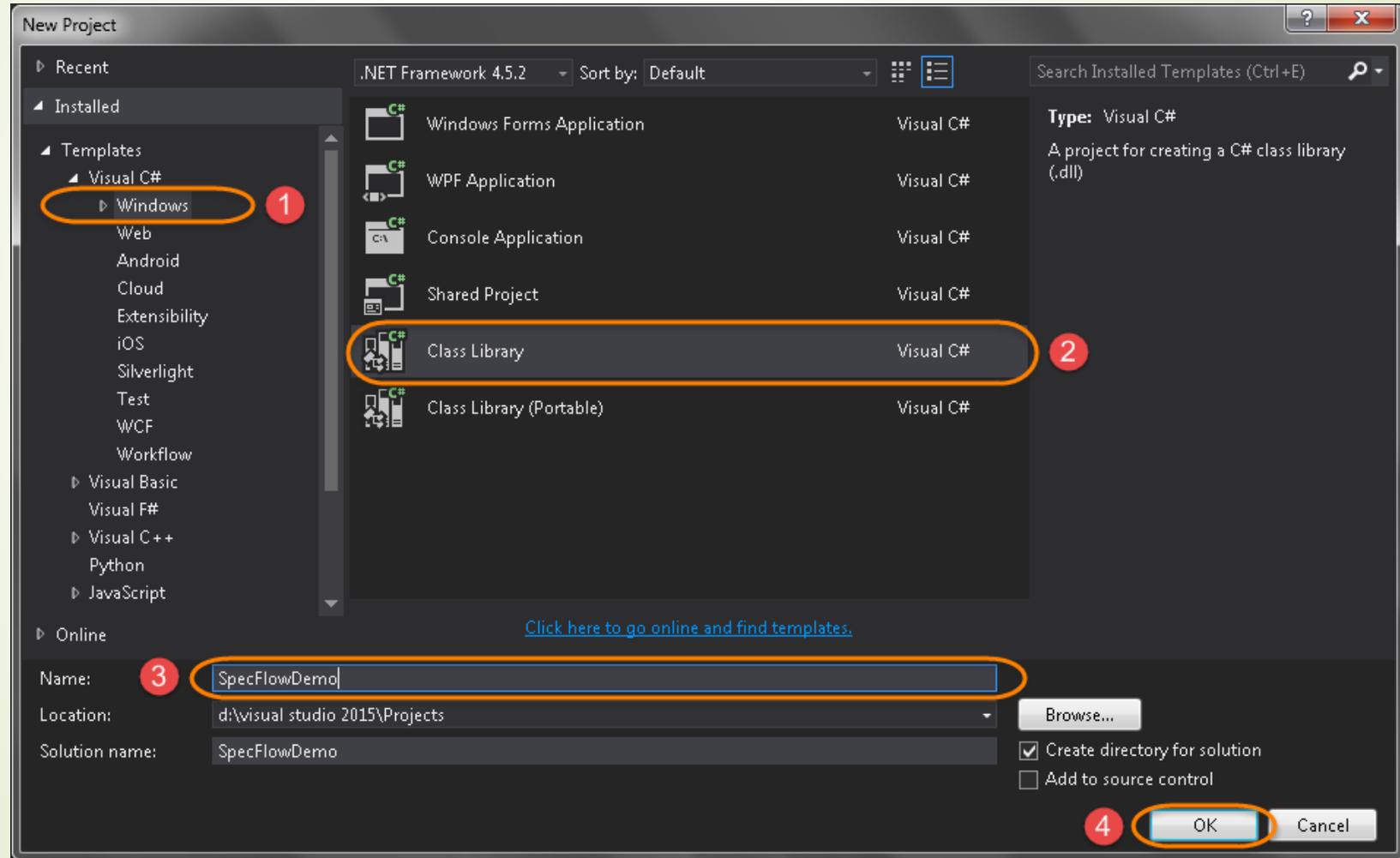
<i>firstValue</i>	<i>secondValue</i>	<i>action</i>	<i>result</i>
1	2	plus	3
2	3	minus	-1
2	2	multiply	4

```
When User enter credentials
```

<i>Key</i>	<i>Value</i>
Username	super.testlogin2018
Password	Test@123

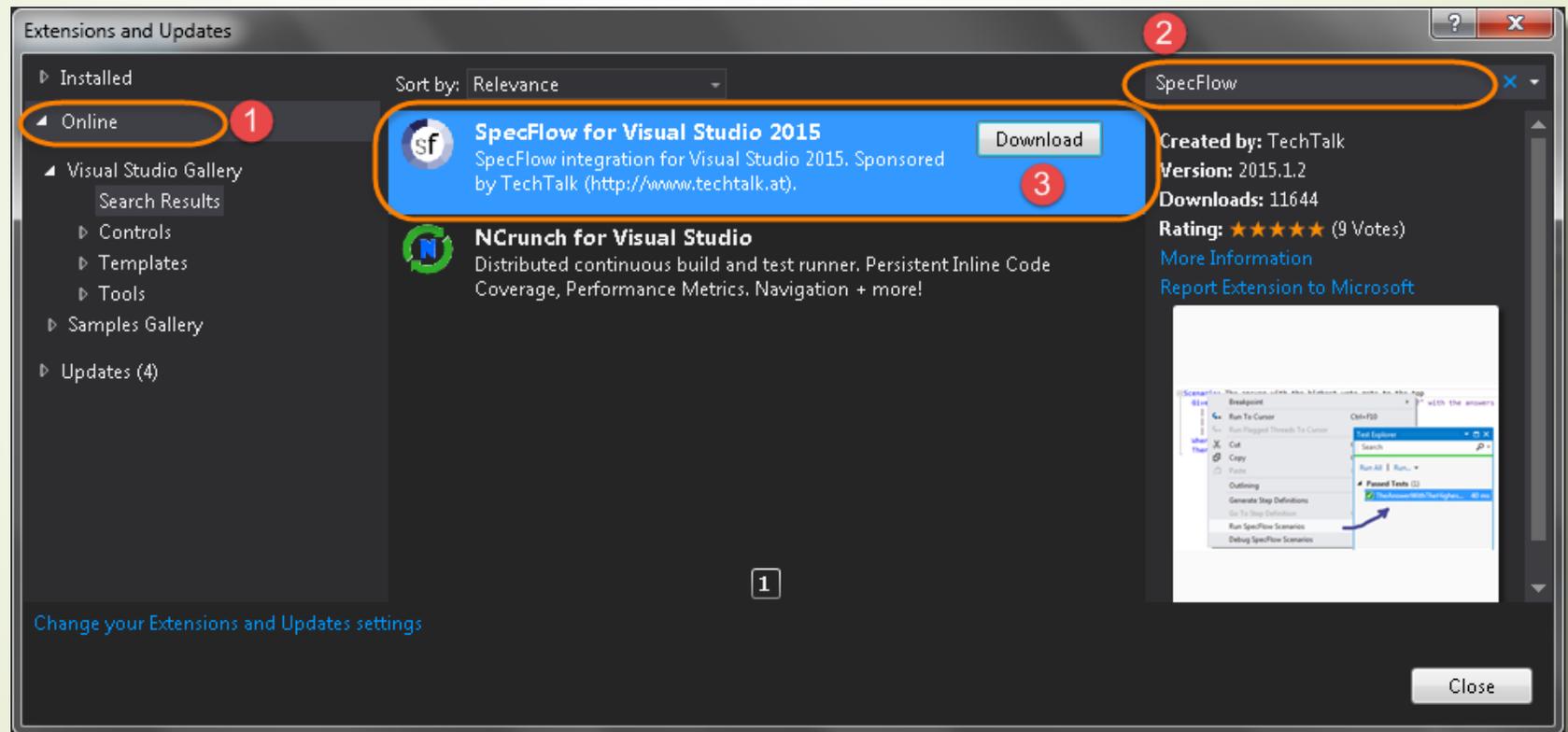
Создание первого проекта SpecFlow

Создание проекта



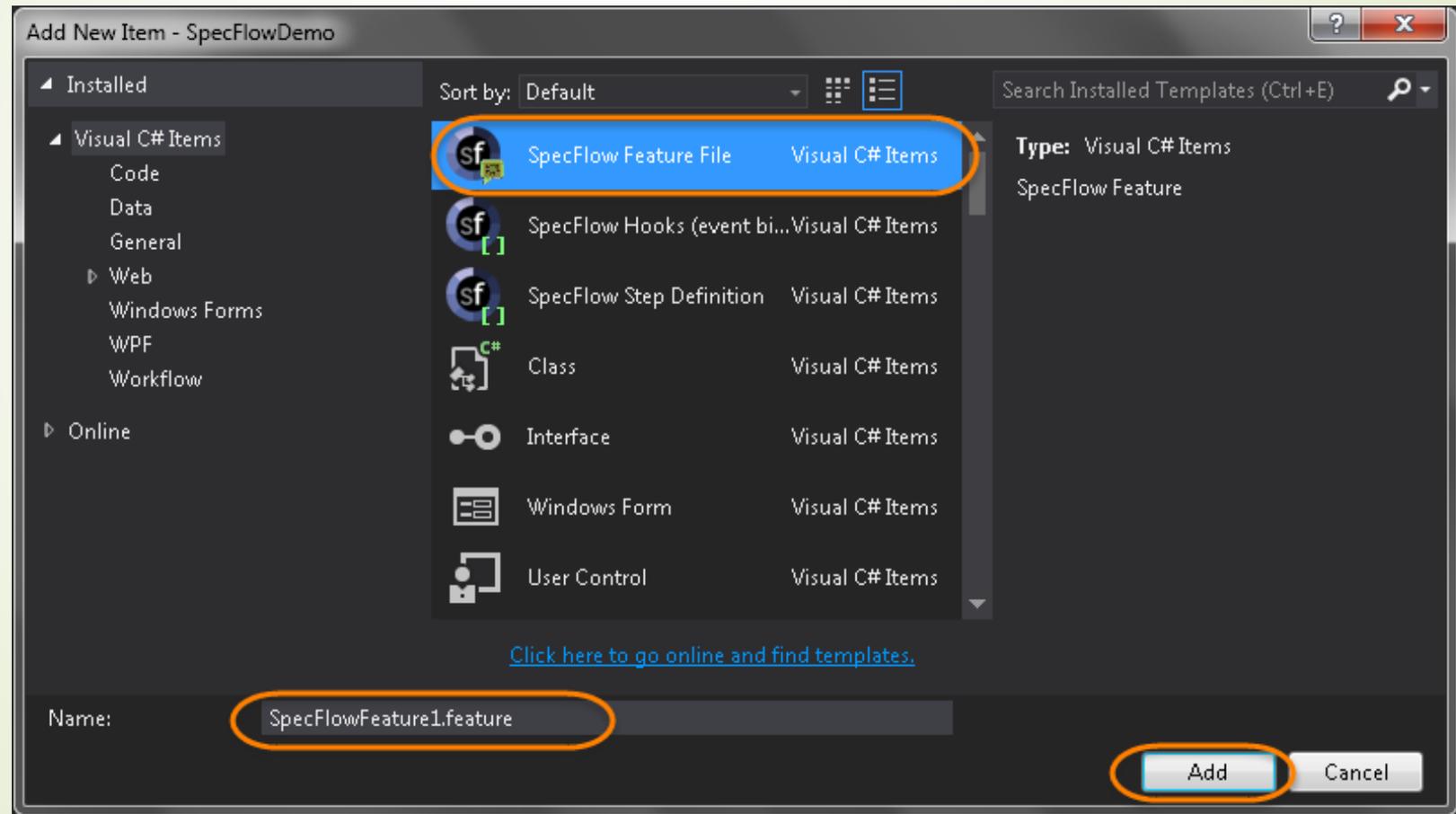
Создание первого проекта SpecFlow

Установка расширения



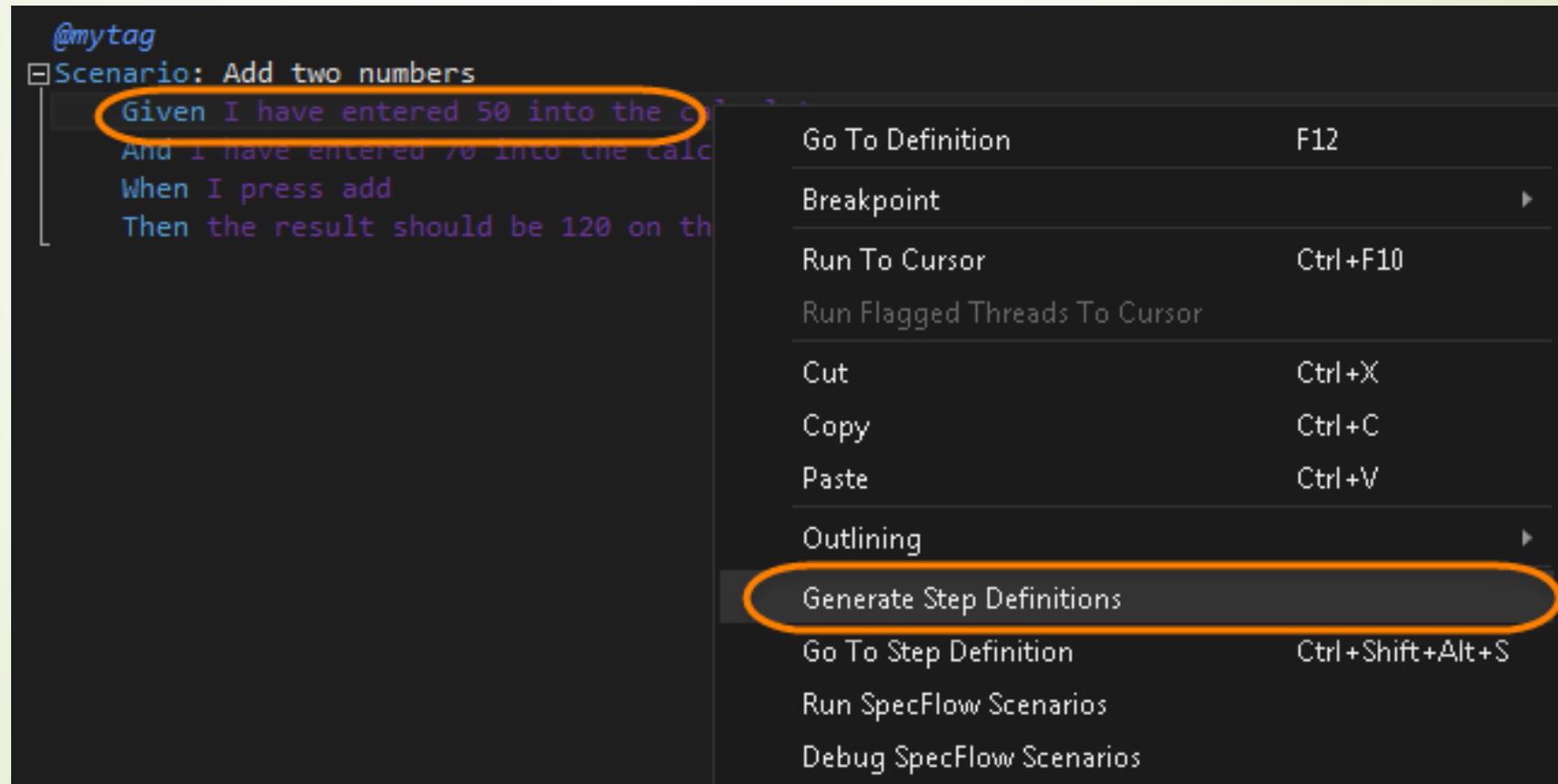
Создание первого проекта SpecFlow

Добавление feature-файла



Создание первого проекта SpecFlow

Генерация шагов тестирования



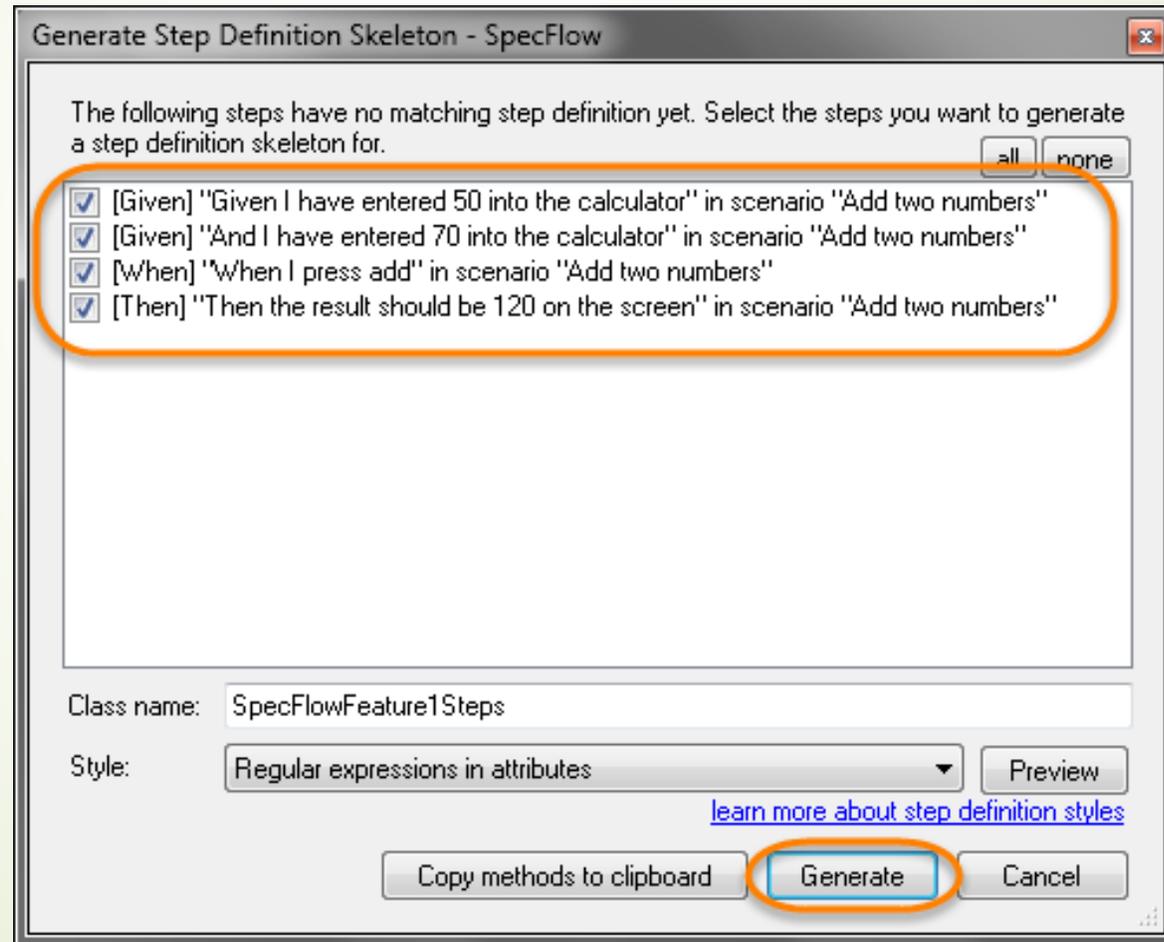
The screenshot shows a code editor with a SpecFlow scenario. The scenario is expanded to show its steps. The first step, "Given I have entered 50 into the calculator", is highlighted with an orange oval. A context menu is open over this step, with the "Generate Step Definitions" option also highlighted by an orange oval. Other options in the menu include "Go To Definition", "Breakpoint", "Run To Cursor", "Run Flagged Threads To Cursor", "Cut", "Copy", "Paste", "Outlining", "Go To Step Definition", "Run SpecFlow Scenarios", and "Debug SpecFlow Scenarios".

```
@mytag
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
```

- Go To Definition F12
- Breakpoint ▶
- Run To Cursor Ctrl+F10
- Run Flagged Threads To Cursor
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Outlining ▶
- Generate Step Definitions**
- Go To Step Definition Ctrl+Shift+Alt+S
- Run SpecFlow Scenarios
- Debug SpecFlow Scenarios

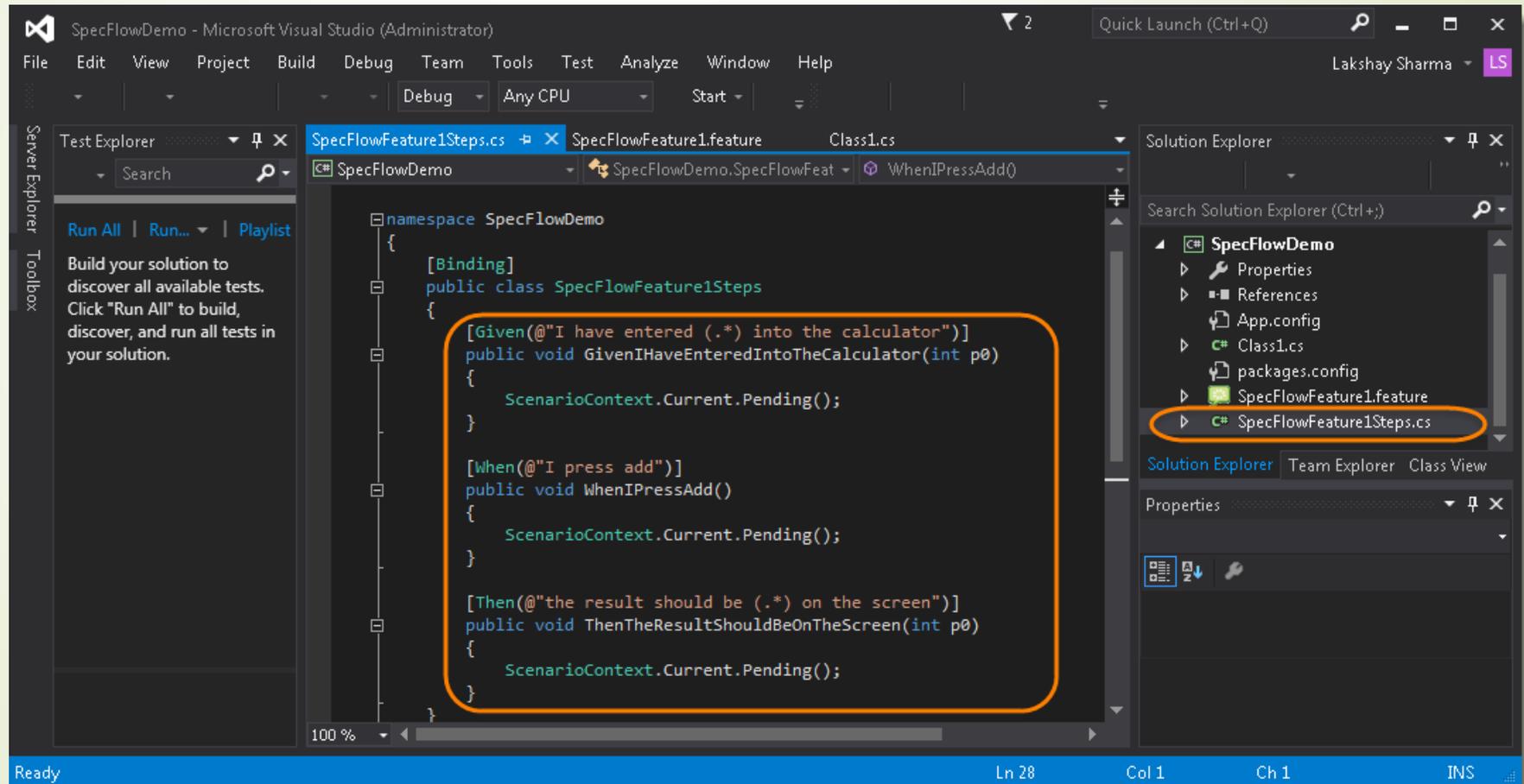
Создание первого проекта SpecFlow

Генерация шагов тестирования



Создание первого проекта SpecFlow

Генерация шагов тестирования



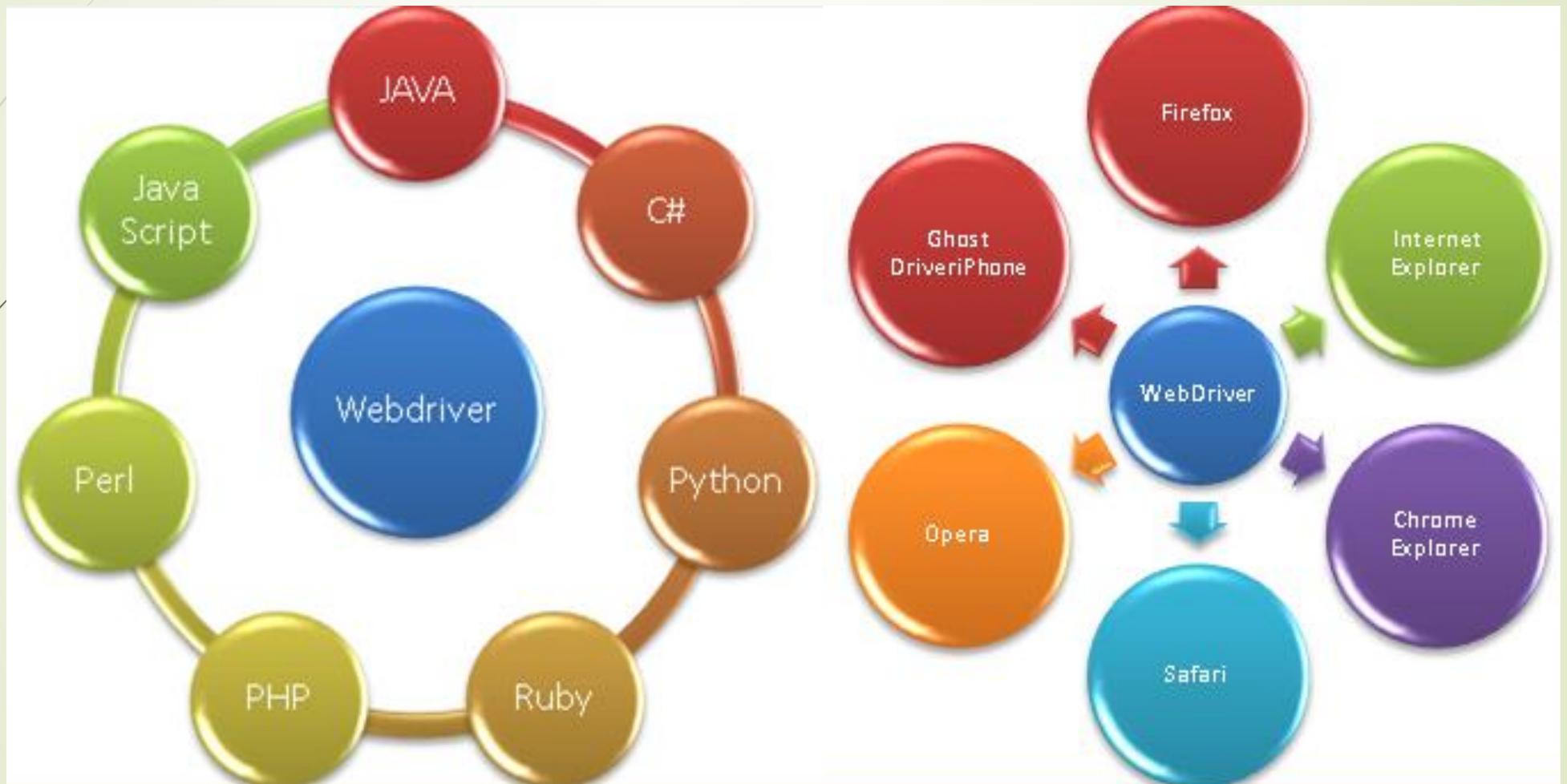
Создание первого проекта SpecFlow

Запуск теста



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="specFlow" type="TechTalk.SpecFlow.Configuration.Configuration" />
  </configSections>
  <specFlow>
    <unitTestProvider name="NUnit" />
  </specFlow>
</configuration>
```

Selenium WebDriver



Selenium WebDriver Navigation

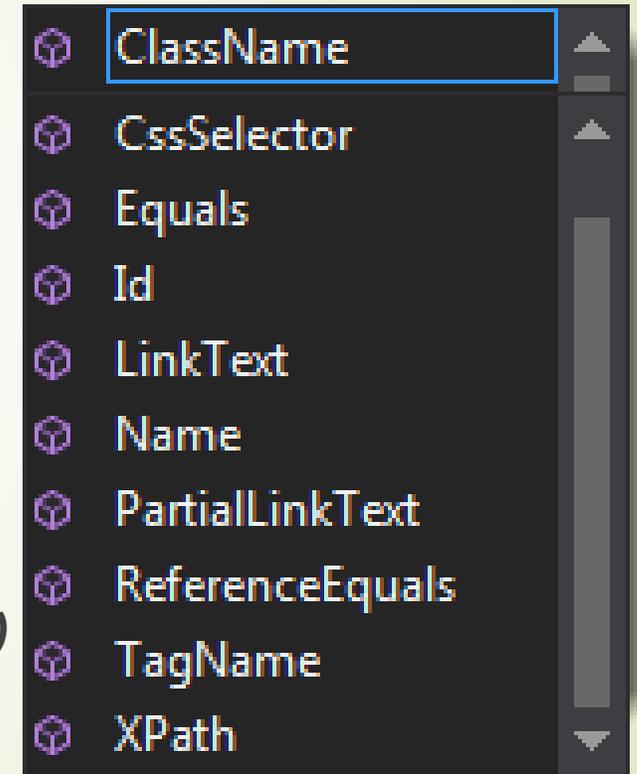
```
References
static void Main(string[] args)
{
    IWebDriver driver = new InternetExplorerDriver(@"C:\Users\abc\Desktop\Server");

    driver.Navigate().
}
}
```

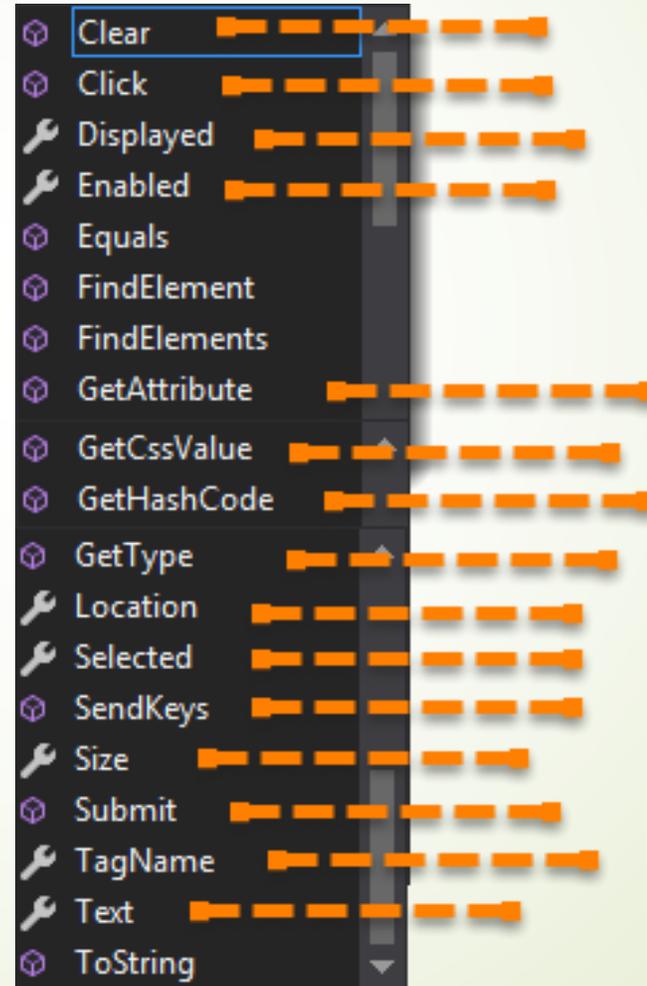
- Back void INavigation.Back()
Move back a single entry in the browser's history.
- Equals
- Forward
- GetHashCode
- GetType
- GoToUrl
- Refresh
- ToString

Selenium WebDriver FindElement и FindElements

- `driver.FindElements(By.ClassName("button"))`
- `driver.FindElement(By.CssSelector("div > p"))`
- `driver.FindElement(By.Id("SubmitButton"))`
- `driver.FindElement(By.LinkText("Partial Link Test"))`
- `driver.FindElement(By.Name("username"))`
- `driver.FindElement(By.PartialLinkText("Partial"))`
- `driver.FindElements(By.TagName("span"))`
- `driver.FindElements(By.XPath("//*[@id="page"]"))`



Selenium WebDriver WebElement



Selenium WebDriver

WebDriver Waits

Факторы влияющие на непредсказуемость веб-приложений при автоматизации тестирования:

- ▶ Медленное подключение к интернету
- ▶ Вызовы Ajax, выполняемые элементами
- ▶ Вызовы базы данных, сделанные веб-сайтом
- ▶ Медленные браузеры
- ▶ Элементы с задержкой загрузки, используемые для оптимизации использования полосы пропускания

Selenium WebDriver

Неявные ожидания(Implicit wait)

```
driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(50));
```

Замечания:

- **Значение по умолчанию 0.**
- Время опроса по умолчанию 250 миллисекунд.
- Задаются на всю сессию
- Можно менять любое количество раз.
- Если элемент не найден выбрасывается исключение **NoSuchElementException**

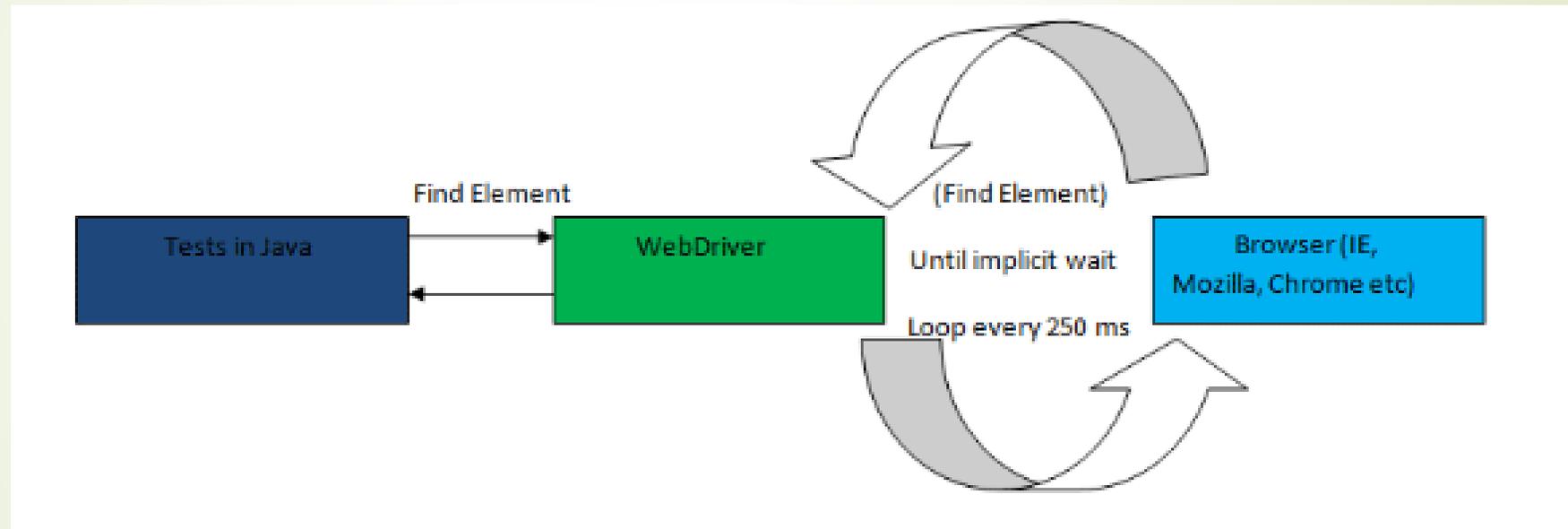
Selenium WebDriver

Явные ожидания(Explicit wait)

```
WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromMinutes(1));  
Func<IWebDriver, bool> waitForElement = new Func<IWebDriver,  
bool>((IWebDriver Web) =>  
{  
    IWebElement element = Web.FindElement(By.Id("target"));  
    return element.GetAttribute("style").Contains("red");  
});  
wait.Until(waitForElement);
```

Selenium WebDriver

Явные ожидания(Explicit wait)



Selenium WebDriver

Пример использования

```
@schedule
Feature: 02 CheckSchedule

@Schedule
Scenario: 01 Browser Schedule Page
  Given goto "http://localhost:52980"
  Then exist link ~ "/schedule/index"
  When goto link href "/schedule/index"
  Then exist link ~ "#dashboard"

Scenario: 02 Create test data
  Given goto "http://localhost:52980/schedule/index#projects"
  When create project "Project"
  Then exist "Project"
  When click "dropdownMenu1"
  And create task "Task1"
  Then exist "Task1"

Scenario: 03 Check Task1 Edit Page
  Given goto test "Task1"
  When "Status" = "В работе"
  When click "save-changes"
  When refresh
  Then "Status" == "В работе"
```

Selenium WebDriver

Пример использования

```
[Given("goto " + ANY_QUOTED_VALUE_REGEX)]
1 reference | 0 changes | 0 authors, 0 changes
public virtual void GivenGoto(string url)
{
    FeatureContext.Current["absPath"] = new Uri(url).GetLeftPart(UriPartial.Path);
    driver.Navigate().GoToUrl(url);
    WriteToLog(message: "[" + DateTime.Now + "]\n" + driver.PageSource);
}
```

```
[When("goto link href " + ANY_QUOTED_VALUE_REGEX)]
5 references | 0 changes | 0 authors, 0 changes
protected void WhenGotoLinkHref(string href)
{
    var xpath = $"//a[contains(@href, '{href}')]";
    driver.WaitUntilElementIsVisible(ExplicitWaitTime, By.XPath(xpath));
    driver.FindElement(By.XPath(xpath)).Click();
    WriteToLog(message: "[" + DateTime.Now + "]\n" + driver.PageSource);
}
```

```
[When("create project " + ANY_QUOTED_VALUE_REGEX)]
0 references | 0 changes | 0 authors, 0 changes
public void WhenCreateProject(string name)
{
    WhenGotoLinkHref("#project/create");
    WhenRefresh();
    When(fieldName: "Name", new WrappedString(name));
    WhenClickButton(buttonId: "save-changes");
    WhenRefresh();
    var projectId = GetParam(driver.Url, paramName: "id");
    var info = new Dictionary<string, string> { { "id", projectId }, { "url", $"#project/edit?id={projectId}" } };
    FeatureContext.Current.Add(name, info);
}
```

Интеграция с TeamCity

- ▶ **Автоматический запуск тестов SpecFlow реализуется как отдельная build конфигурация, которая состоит из следующих шагов:**
 1. NuGet Installer (подготовка репозиториев)
 2. MSBuild (сборка проекта автотестов)
 3. NUnit (выполнение автотестов)
 4. SpecFlow Report Build Step (конвертирование результатов теста в формат html)
- ▶ **Чтобы сохранить созданные выходные файлы, перейдите на вкладку GeneralSettings в конфигурации сборки и добавьте строку SpecFlowTests.html в поле Artifact paths.**
- ▶ **Перейдите к настройкам своего проекта и нажмите вкладку «Report Tabs». Здесь нажмите вкладку «Create new build report tab». В качестве заголовка закладки выберите SpecFlowReport и для начальной страницы введите SpecFlowTests.html.**

Результат работы SpecFlow

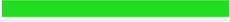
Hydra.Specs Test Execution Report

Generated by SpecFlow at 10/09/2017 15:12 (see <http://www.specflow.org/>).

Summary

Features	Success rate	Scenarios	Success	Failed	Pending	Ignored
8 features	80% 	10	8	2	0	0

Feature Summary

Feature	Success rate	Scenarios	Success	Failed	Pending	Ignored
CheckJobPage	100% 	2	2	0	0	0
CheckListProject	100% 	1	1	0	0	0
CheckSchedule	100% 	1	1	0	0	0
CheckTaskPage	100% 	2	2	0	0	0
EditProjectPage	0% 	1	0	1	0	0
EditSubProject	100% 	1	1	0	0	0
LoadProjectPage	0% 	1	0	1	0	0
LoadSubProject	100% 	1	1	0	0	0

Feature Execution Details

Feature: CheckJobPage

Scenario	Status	Time(s)
Check Job Create Page From Project [show]	success	14.279
Check Job Edit Page [show]	success	7.361

Feature: CheckListProject

Scenario	Status	Time(s)
Check Open List Project [show]	success	4.017

Feature: CheckSchedule

Scenario	Status	Time(s)
Browser Schedule Page [show]	success	6.870

Feature: CheckTaskPage

Scenario	Status	Time(s)
Check Task Create Page From Project [show]	success	3.704
Check Task Edit Page [show]	success	6.657

Feature: EditProjectPage

Scenario	Status	Time(s)
Project Page Edit [show]	failure	6.149

Feature: EditSubProject

Опыт использования SpecFlow

- Используется в 5 проектах.
- Написано более 50 уникальных шагов общего назначения.
- Написано более 20 шагов под конкретный проект.
- Используется для автоматизации рутинного ручного тестирования.
- Используется для «дымового тестирования».
- Был отловлен критический баг незадолго до релиза.
- Было отловлено большое число мелких багов.
- Примерное затраченное время на написание первых шагов составило 2 недели

Заключение



Полезные ссылки

- <https://specflow.org> (Официальный сайт SpecFlow)
- <https://habr.com/ru/post/182032/> (Исполняемая спецификация: SpecFlow от А до Я)
- <https://www.toolsqa.com/> (Tools QA)
- <https://www.browserstack.com/automate/specflow>
(Guide to running Selenium WebDriver tests with SpecFlow on BrowserStack.)
- <https://kreisfahrer.gitbooks.io/selenium-webdriver/>
(Selenium WebDriver)