BDD

specflow

1) BDD, Specification by Example
2) SpecFlow, TestStack.BDDfy
3) Acceptance testing

Links

https://habrahabr.ru/post/166747/

Specification By Example – BDD для прагматиков

https://habrahabr.ru/post/182032/

Исполняемая спецификация: SpecFlow от А до Я

https://habrahabr.ru/post/178407/

Автоматизация тестирования Web-приложений (BDD-style)

SbE

В основе процесса *Specification By Example* лежат

- agile
- TDD
- BDD
- Continuous Integration

Ключевые элементы *Specification By Example* :

- Выделяйте главное *(deriving scope from goals)*
- Составляйте спецификацию совместно *(specifying collaboratively)*
- Приводите примеры *(illustrating using examples)*
- Очищайте спецификацию *(refining the specification)*
- Внедряйте автоматизацию тестирования без изменения спецификации *(automating validation without changing specification)*
- Встраивайте выполнение тестов в процесс сборки и развивайте документацию *(validating frequently, evolving a documentation system)*

User Stories

User-stories обязательно должны содержать 3 пункта:

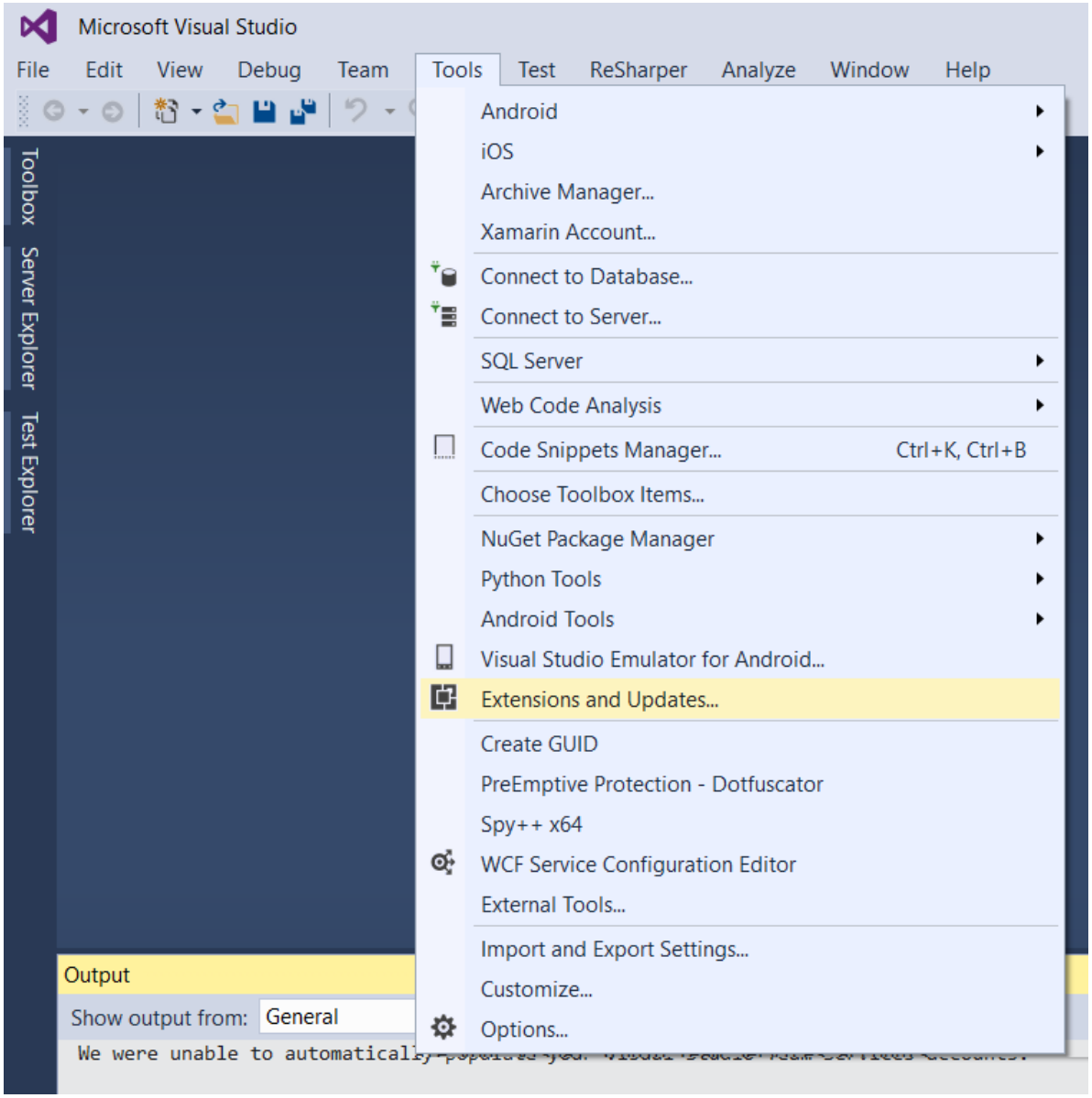
In order to	—	зачем?
As a	—	кто?
I want	—	что?

Given — первоначальный контекст (предусловие)
When — событие (что является триггером сценария)
Then — результат, который мы хотим получить

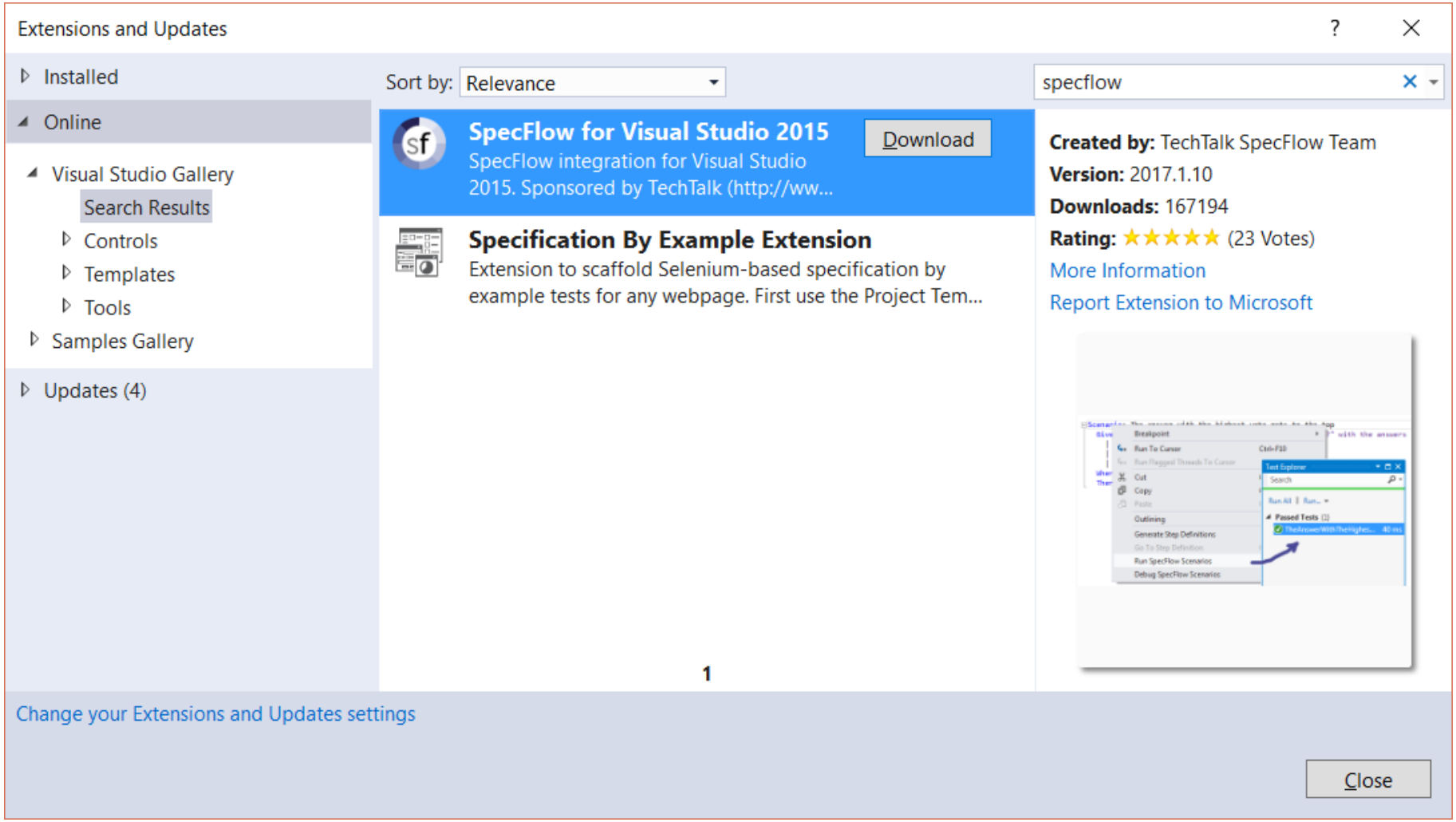
Feature and Scenario

```
Feature: Free delivery
        In order to save money
        As a VIP customer
        I want the system to offer free delivery on certain items to me

Scenario: Free delivery
        Given I am a VIP customer
        And I am on product detail page
        And There are only books in my shopping cart
        And There are <= 10 books in my shopping cart
        And I have added 'ABC Press' book to my shopping cart
        When I press 'Go to checkout' button
        And I have chosen 'Moscow' in 'Ship To' dropdown
        Then I can choose free delivery
```

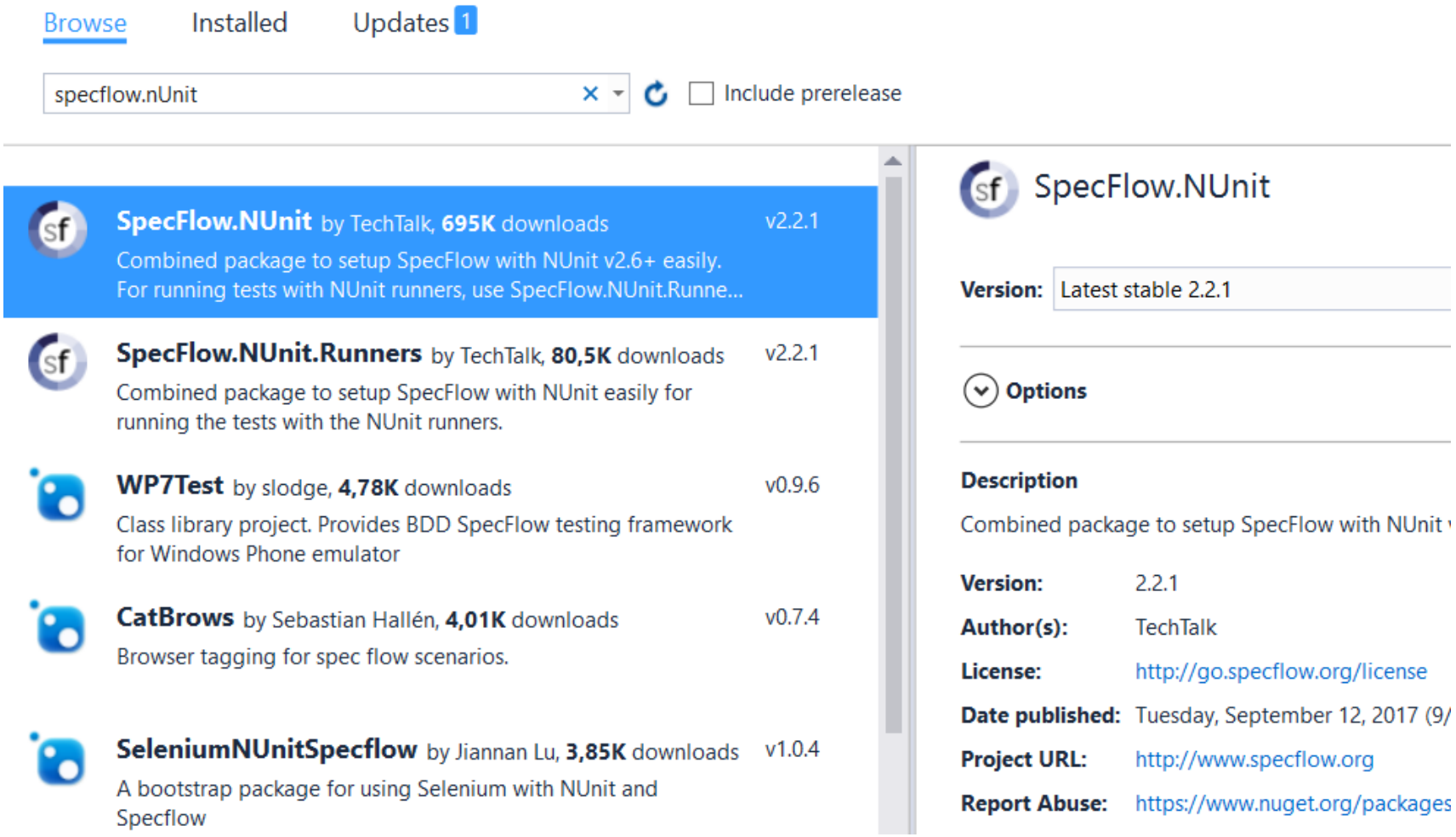
Install SpecFlow



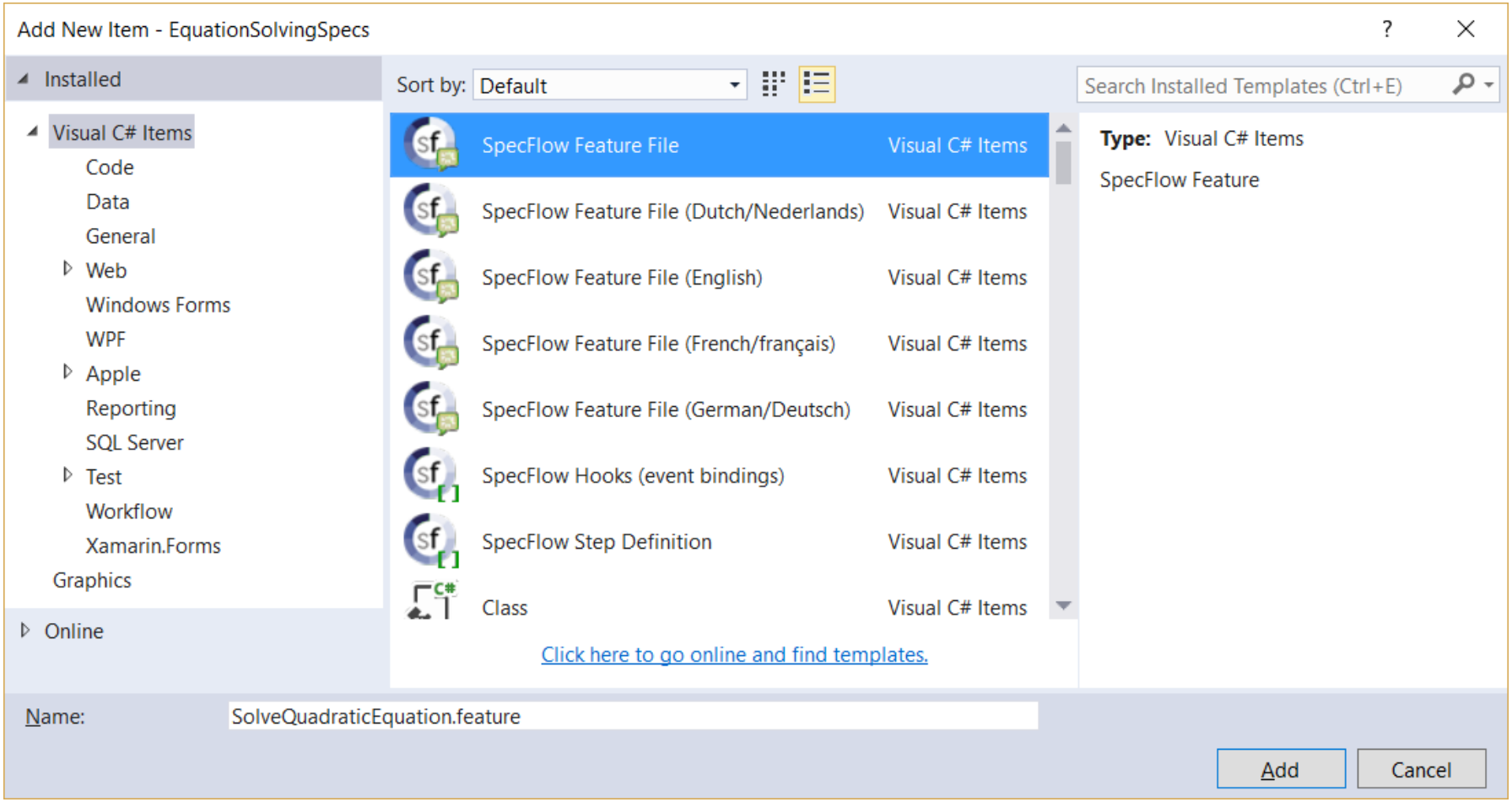
Install SpecFlow



SpecFlow


Browse
Installed
Updates 1
specflow.nUnit
Include prerelease
SpecFlow.NUnit by TechTalk, 695K downloads
v2.2.1
Combined package to setup SpecFlow with NUnit v2.6+ easily.
For running tests with NUnit runners, use SpecFlow.NUnit.Runne...
SpecFlow.NUnit.Runners by TechTalk, 80,5K downloads
v2.2.1
Combined package to setup SpecFlow with NUnit easily for running the tests with the NUnit runners.
WP7Test by slodge, 4,78K downloads
v0.9.6
Class library project. Provides BDD SpecFlow testing framework for Windows Phone emulator
CatBrows by Sebastian Hallén, 4,01K downloads
v0.7.4
Browser tagging for spec flow scenarios.
SeleniumNUnitSpecflow by Jiannan Lu, 3,85K downloads
v1.0.4
A bootstrap package for using Selenium with NUnit and Specflow
SpecFlow.NUnit
Version: Latest stable 2.2.1
Options
Description
Combined package to setup SpecFlow with NUnit
Version: 2.2.1
Author(s): TechTalk
License: http://go.specflow.org/license
Date published: Tuesday, September 12, 2017 (9/
Project URL: http://www.specflow.org
Report Abuse: https://www.nuget.org/packages

Add SpecFlow Feature

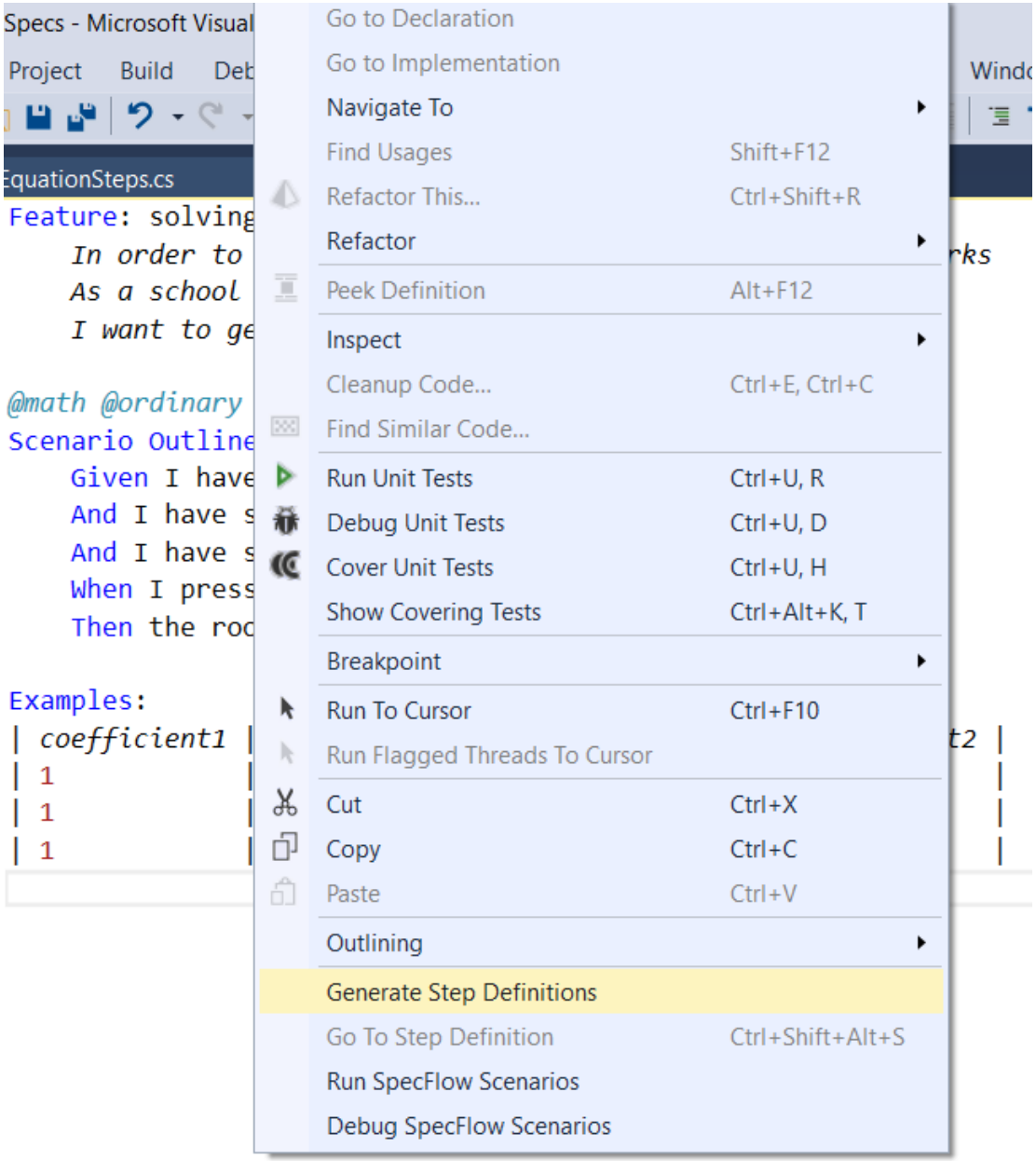


Write SpecFlow Feature

EquationSteps.cs | SolveQuadraticEquation.feature

```
Feature: solving quadratic equation
    In order to demonstrate that Quadratic Formula really works
    As a school math teacher
    I want to get the roots of quadratic equation

@math @ordinary
Scenario Outline: Solve ordinary quadratic equation
    Given I have set <coefficient1>
    And I have set <coefficient2>
    And I have set <coefficient3>
    When I press Solve
    Then the roots should be <root1> and <root2>

Examples:
| coefficient1 | coefficient2 | coefficient3 | root1 | root2 |
| 1            | 1            | -6           | -3    |  2    |
| 1            | 0            | -4           |  2    | -2    |
| 1            | 5            | 0            |  0    | -5    |
```

Generate Step Definitions



Generate Step Definitions

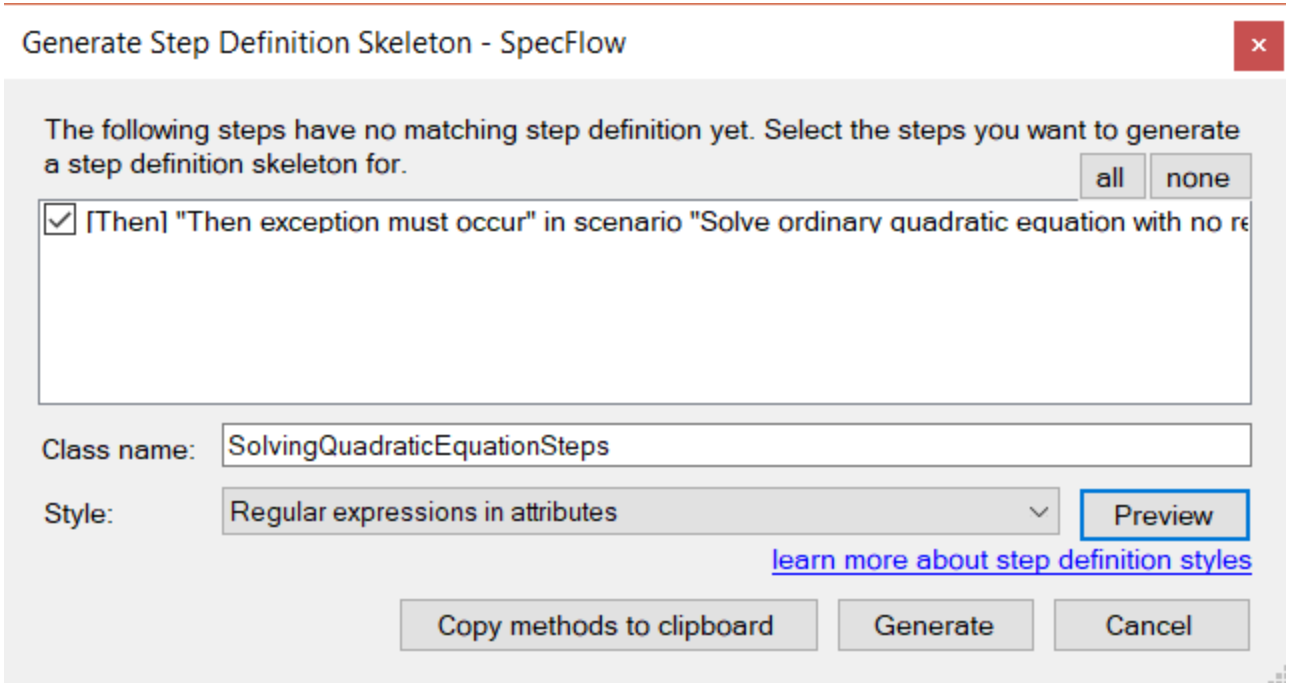
```
using TechTalk.SpecFlow;

namespace EquationSolvingSpecs
{
    [Binding]
    public class SolvingQuadraticEquationSteps
    {
        [Given(@"I have set (.*)")]
        public void GivenIHaveSet(string p0)
        {
            ScenarioContext.Current.Pending();
        }

        [When(@"I press Solve")]
        public void WhenIPressSolve()
        {
            ScenarioContext.Current.Pending();
        }

        [Then(@"the roots should be (.*) and (.*)")]
        public void ThenTheRootsShouldBeAnd(string p0, string p1)
        {
            ScenarioContext.Current.Pending();
        }
    }
}
```

Write Code

```
using TechTalk.SpecFlow;

namespace EquationSolvingSpecs
{
    [Binding]
    public class SolvingQuadraticEquationSteps
    {
        readonly EquationSolver _solver = new EquationSolver(new PolynomialRoots());

        readonly List<double> _coefficients = new List<double>();
        double[] _roots;

        [Given(@"I have set (.*)")]
        public void GivenIHaveSet(double coeff)
        {
            _coefficients.Add(coeff);
        }

        [When(@"I press Solve")]
        public void WhenIPressSolve()
        {
            _roots = _solver.Solve(_coefficients.ToArray());
        }

        [Then(@"the roots should be (.*) and (.*)")]
        public void ThenTheRootsShouldBeAnd(double r1, double r2)
        {
            Assert.That(_roots, Is.EquivalentTo(new [] { r1, r2 }));
        }
    }
}
```

Another scenario

```
@math @exception
Scenario: Solve ordinary quadratic equation with no real roots
    Given I have set 1
    And I have set 1
    And I have set 6
    When I press Solve
    Then exception must occur
```



Another scenario

```
[When(@"I press Solve")]
public void WhenIPressSolve()
{
    // 1-ый способ обработки исключения
    try
    {
        _roots = _solver.Solve(_coefficients.ToArray());
    }
    catch (Exception ex)
    {
        _exception = ex;
    }

    // 2-ой способ - Assert.Catch() и использовать Scope в атрибуте When
}

[Then(@"the roots should be (.*) and (.*)")]
public void ThenTheRootsShouldBeAnd(double r1, double r2)
{
    Assert.That(_roots, Is.EquivalentTo(new [] { r1, r2 }));
}

[Then(@"exception must occur")]
public void ThenExceptionMustOccur()
{
    Assert.That(_exception, Is.Not.Null);
    Assert.That(_exception, Is.TypeOf<ArithmeticException>());
}
```

Another feature

```
Feature: solving custom algebraic equation
    In order to demonstrate how the algebraic equations are solved
    As a school math teacher
    I want to get the roots of algebraic equation

@math @ordinary
Scenario: Solve ordinary algebraic equation with real roots
    Given I have specified array of coefficients { 1, 3, -5, -15, 4, 12 }
    When I press Solve
    Then the array of roots should be { -3, -2, -1, 1, 2 }

@math @exception
Scenario: Solve algebraic equation with no unknowns
    Given I have set 0
    And I have set 0
    And I have set 3
    When I press Solve
    Then argument exception must occur
```

Extract Text Context

```
public class EquationSolvingContext
{
    public EquationSolver Solver { get; set; } = new EquationSolver(new FakeRoots());

    public List<double> Coefficients { get; set; } = new List<double>();
    public double[] Roots { get; set; }

    public Exception Exception { get; set; }
}
```

```
public class SolvingCustomAlgebraicEquationSteps
{
    private readonly EquationSolvingContext _context;

    public SolvingCustomAlgebraicEquationSteps(EquationSolvingContext context)
    {
        _context = context;
    }

    [Given(@"I have specified array of coefficients \{ ((?:.,\d+)*(?:.+)) }")]
    public void GivenIHaveSpecifiedArrayOfCoefficients(List<double> coeffs)
    {
        _context.Coefficients = coeffs;
    }

    [Then(@"the array of roots should be \{ ((?:.,\d+)*(?:.+)) }")]
    public void ThenTheArrayOfRootsShouldBe(List<double> roots)
    {
        Assert.That(_context.Roots, Is.EquivalentTo(roots));
    }

    [StepArgumentTransformation(@"((?:.,\d+)*(?:.+))")]
    public static List<double> ListIntTransform(string doubles)
    {
        return doubles.Split(',').Select(double.Parse).ToList();
    }

    [Then(@"argument exception must occur")]
    public void ThenArgumentExceptionMustOccur()
    {
        Assert.That(_context.Exception, Is.Not.Null);
        Assert.That(_context.Exception, Is.TypeOf<ArgumentException>());
    }
```

BDD and UI testing

```
Feature: SearchCSharpFeature
    In order to download WebDriver for C#
    As a Selenium website visitor
    I want to check that corresponding download link exists

Scenario: Search for C# download link
    Given I have opened Selenium search page
    And I have entered C#
    When I press search button
    Then the search results contain Downloads
```

BDD and UI testing

```
[Binding]
public class SearchCSharpFeatureSteps
{
    private IWebDriver _driver;

    private SeleniumSearchPage _searchPage;
    private SearchResultsPage _resultsPage;

    [BeforeScenario]
    public void Init()
    {
        _driver = new ChromeDriver();
        _driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(30);
    }

    [AfterScenario]
    public void Quit()
    {
        _driver.Quit();
    }
```

BDD and UI testing

```
[Given(@"I have opened Selenium search page")]
public void GivenIHaveOpenedSeleniumSearchPage()
{
    _searchPage = new SeleniumSearchPage(_driver);
    _searchPage.Navigate();
}

[Given(@"I have entered (.*)")]
public void GivenIHaveEntered(string query)
{
    _searchPage.EnterSearchQuery(query);
}

[When(@"I press search button")]
public void WhenIPressSearchButton()
{
    _resultsPage = _searchPage.ClickSearch();
}

[Then(@"the search results contain (.*)")]
public void ThenTheSearchResultsContain(string link)
{
    Assert.That(_resultsPage.Links, Does.Contain(link));
}
```

TestStack.BDDfy

http://teststackbddfy.readthedocs.io/en/latest/

Прочитать самостоятельно