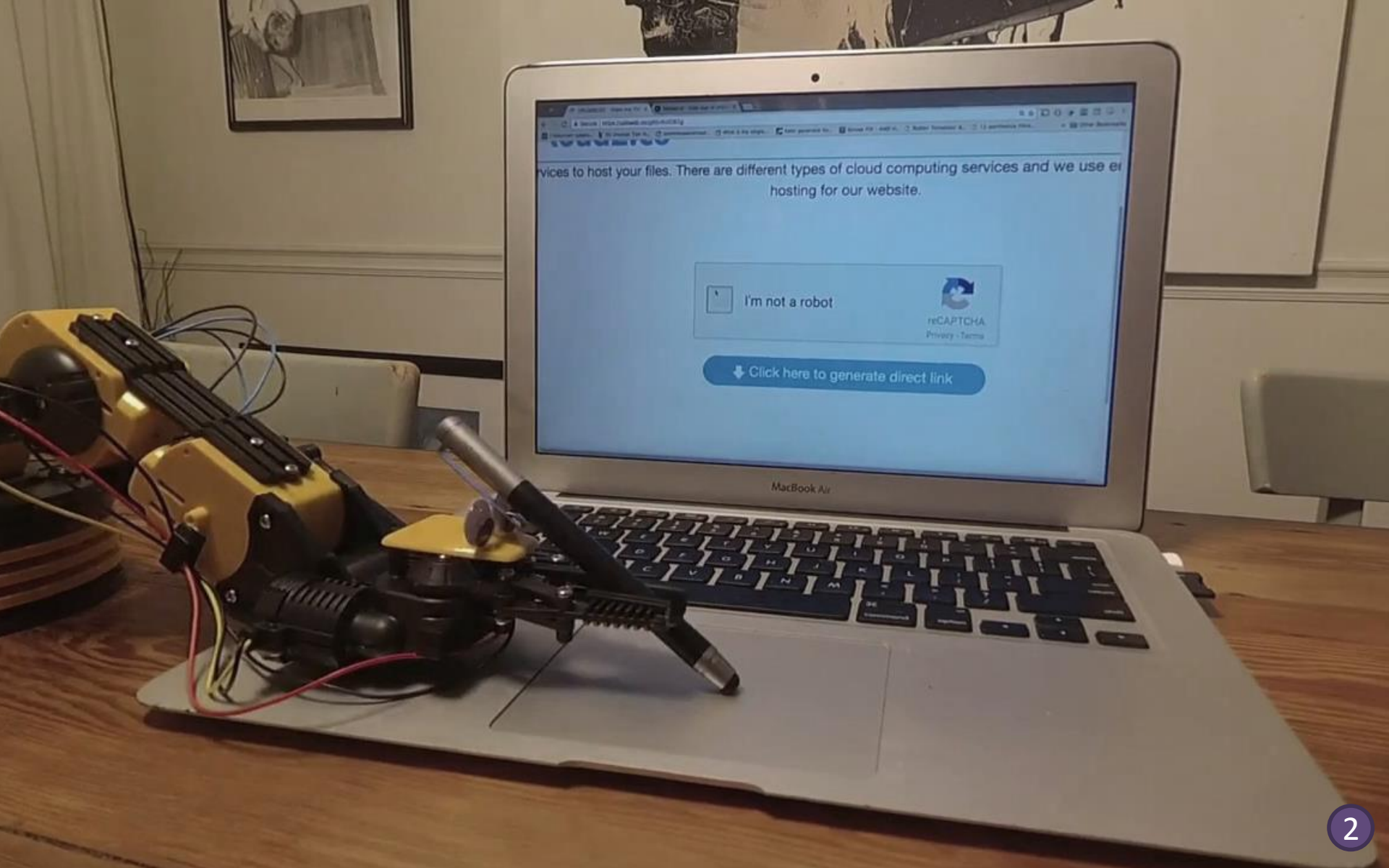




Alfa·Bank



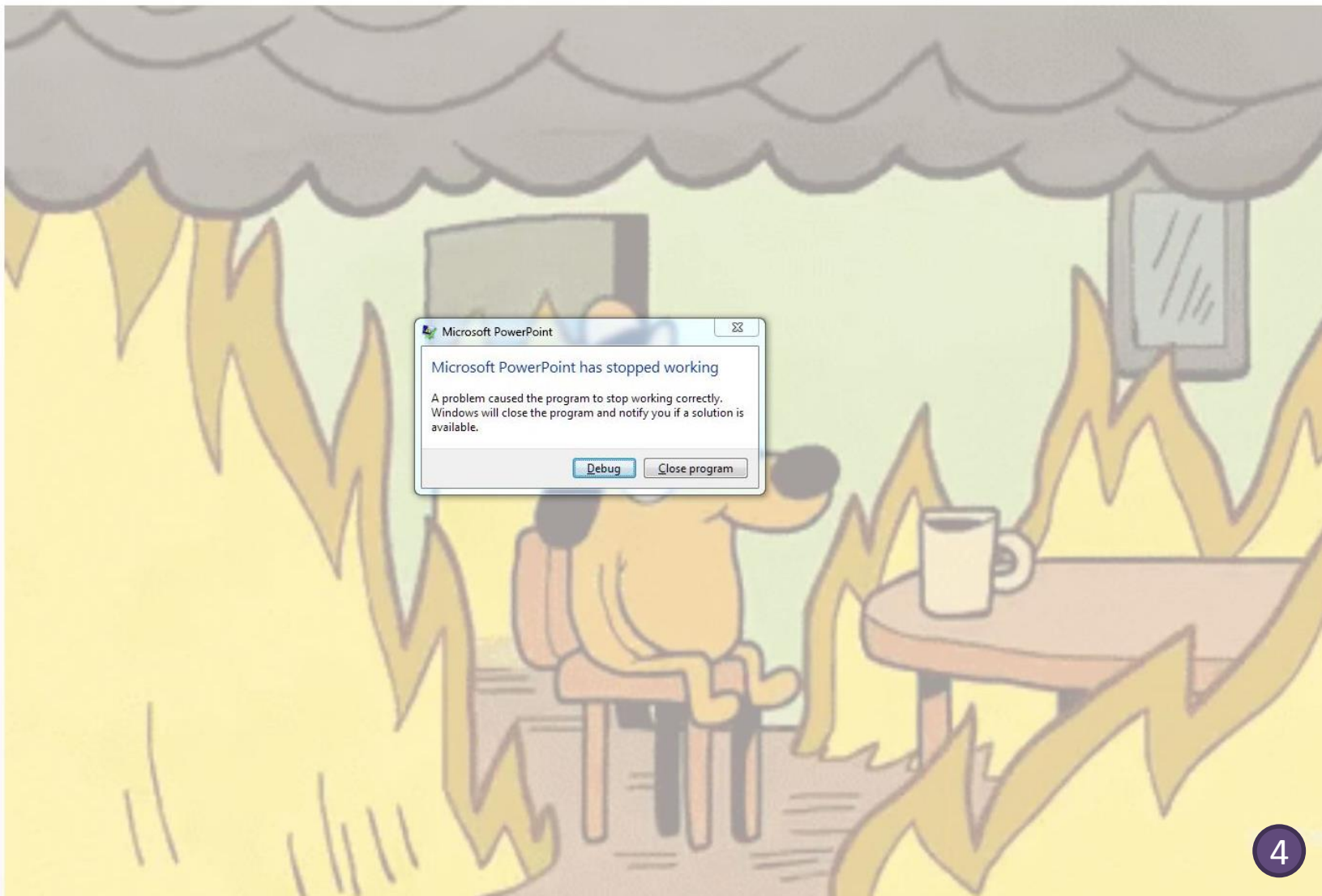
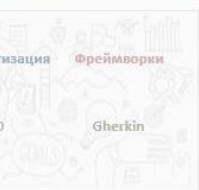
Ганин Андрей

QA Architect



Alfa·Bank

t.me/NoLifer







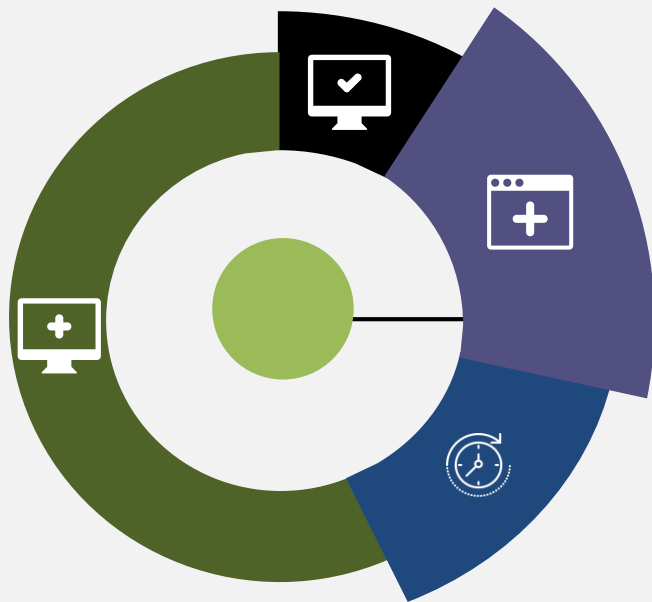
Автоматизация

Фреймворки

BDD

Gherkin

Интернационализации
Регрессионное Стресс Тестирование
Производительности
Дымовое Чёрного Автоматизированное
Конфигурационное Альфа
Статическое Системное Белого
Интерфейса Стабильности Серого
Локализации
Ручное Интеграционное
Негативное Интуитивное Нагрузочное
Функциональное
Приёмочное Формальное
Документации Позитивное Юзабилити
Динамическое
Расширенное Модульное
Бета Безопасности
Взаимодействия



Программный продукт

- Бизнес требования
- Пользовательские требования
- Функциональные требования



Цикл
разработки ПО



Проверка
требований



Доработка ПО



Сокращение
времени





Автоматизация Адаптация Минимизация

Автоматизация

Сложно



Долго



Дорого



Google



Поиск в Google

Мне повезёт!

Google



Поиск в Google

Мне повезёт!



spb dotnet meetup



Все

Видео

Новости

Картинки

Карты

Ещё

Настройки

Инструменты

Результатов: примерно 87 600 (0,34 сек.)

SpbDotNet Community | ВКонтакте

<https://vk.com/spbdotnet> ▼

NET и регулярно проводить встречи в Санкт-Петербурге для обсуждения этих NET-разработчиков на открытый meetup в Санкт-Петербурге.

SpbDotNet Community / TimePad.ru

<https://spbdotnet.timepad.ru/> ▼

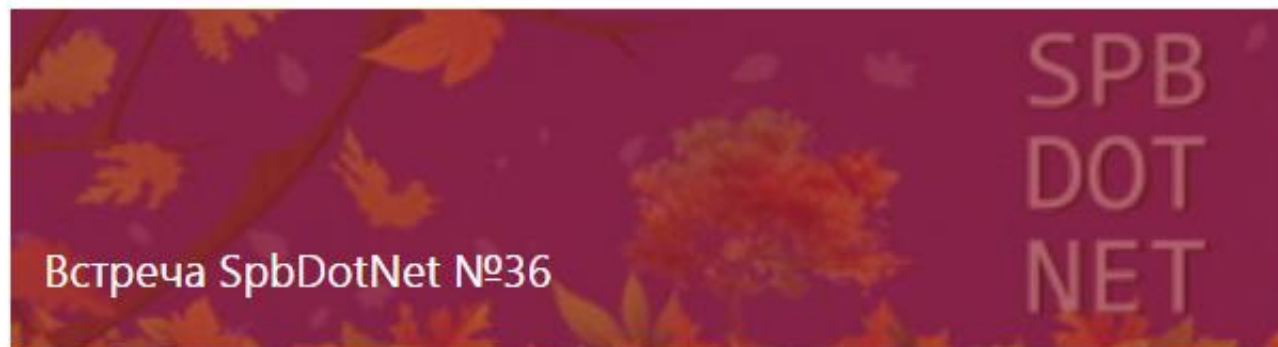
Санкт-Петербург 25 октября с 19:00 до 22:00, через 8 дней ... NET сообществ со всей России. Вас ожидают полезные знакомства, куча подарков и, ... В уютных стенах IT Global Meetup #12 мы соберёмся для обсуждения самых ...

SPB .NET Meetup #12 / События на TimePad.ru

<https://spbdotnet.timepad.ru/event/336766/> ▼

Уже на этой неделе, в четверг, 2 июня мы проведем двенадцатую встречу SPB .NET Community. Встреча будет гиковским pre-party к большой .

События



Санкт-Петербург 🕒 25 октября с 19:00 до 22:00, через 7 дней

Пришла пора побеседовать о диагностике наших приложений и научиться пользоваться одним из самых полезных инструментов разработчика - PerfView. А также обсудим каким образом обеспечивают качество своих продуктов крупнейшие банки. Поговорим о BDD тестировании и правильных инструментах для этого. Вот с такими серьёзными темами пойдём в гости к нашей любимой Аркадии.

[Далее](#)



SpbDotNet Community

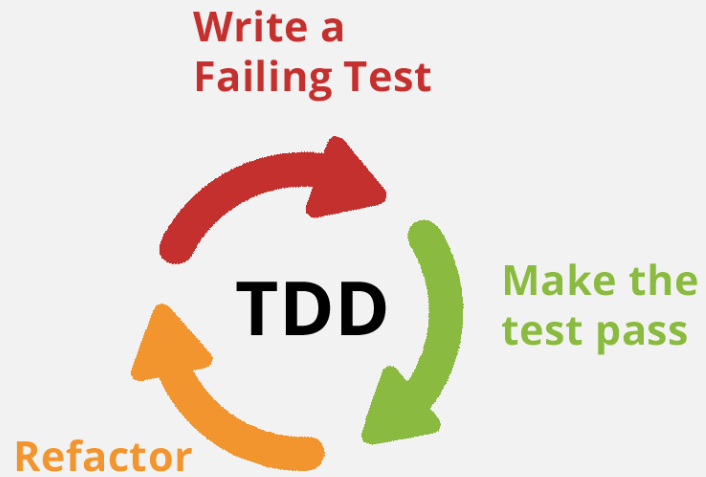
Подписаться на рассылку **14**

```

[ NUnit.Framework.TestAttribute() ]
[ NUnit.Framework.DescriptionAttribute("Google - SpbDotNet") ]

public virtual void Google_SpbDotNet()
{
    TechTalk.SpecFlow.ScenarioInfo scenarioInfo = new TechTalk.SpecFlow.ScenarioInfo("Google - SpbDotNet", ((string[])(null)));
    this.ScenarioSetup(scenarioInfo);
    this.FeatureBackground();
    testRunner.Given("я инициализирую браузер \'GoogleChrome\' без UI", ((string)(null)),
        ((TechTalk.SpecFlow.Table)(null)), "Given ");
    testRunner.Given("я открываю веб-страницу \'google.com\'", ((string)(null)),
        ((TechTalk.SpecFlow.Table)(null)), "Given ");
    testRunner.And("я развернул веб-страницу на весь экран", ((string)(null)),
        ((TechTalk.SpecFlow.Table)(null)), "And ");
    testRunner.And("я ввожу в поле \'Поиск\' веб-страницы значение \'spb dotnet meetup\'", ((string)(null)),
        ((TechTalk.SpecFlow.Table)(null)), "And ");
    testRunner.When("выполнено нажатие на элемент \'КнопкаПоиск\' на веб-странице", ((string)(null)),
        ((TechTalk.SpecFlow.Table)(null)), "When ");
    testRunner.Then("элемент \'SpbDotnet\' отображается на веб-странице", ((string)(null)),
        ((TechTalk.SpecFlow.Table)(null)), "Then ");
    testRunner.Then("выполнен переход на веб-страницу \'spbdotnet.timepad.ru\' после нажатия на элемент " +
        "\'SpbDotnet\'", ((string)(null)), ((TechTalk.SpecFlow.Table)(null)), "Then ");
    this.ScenarioCleanup();
}

```



Write a failing
feature test

BDD

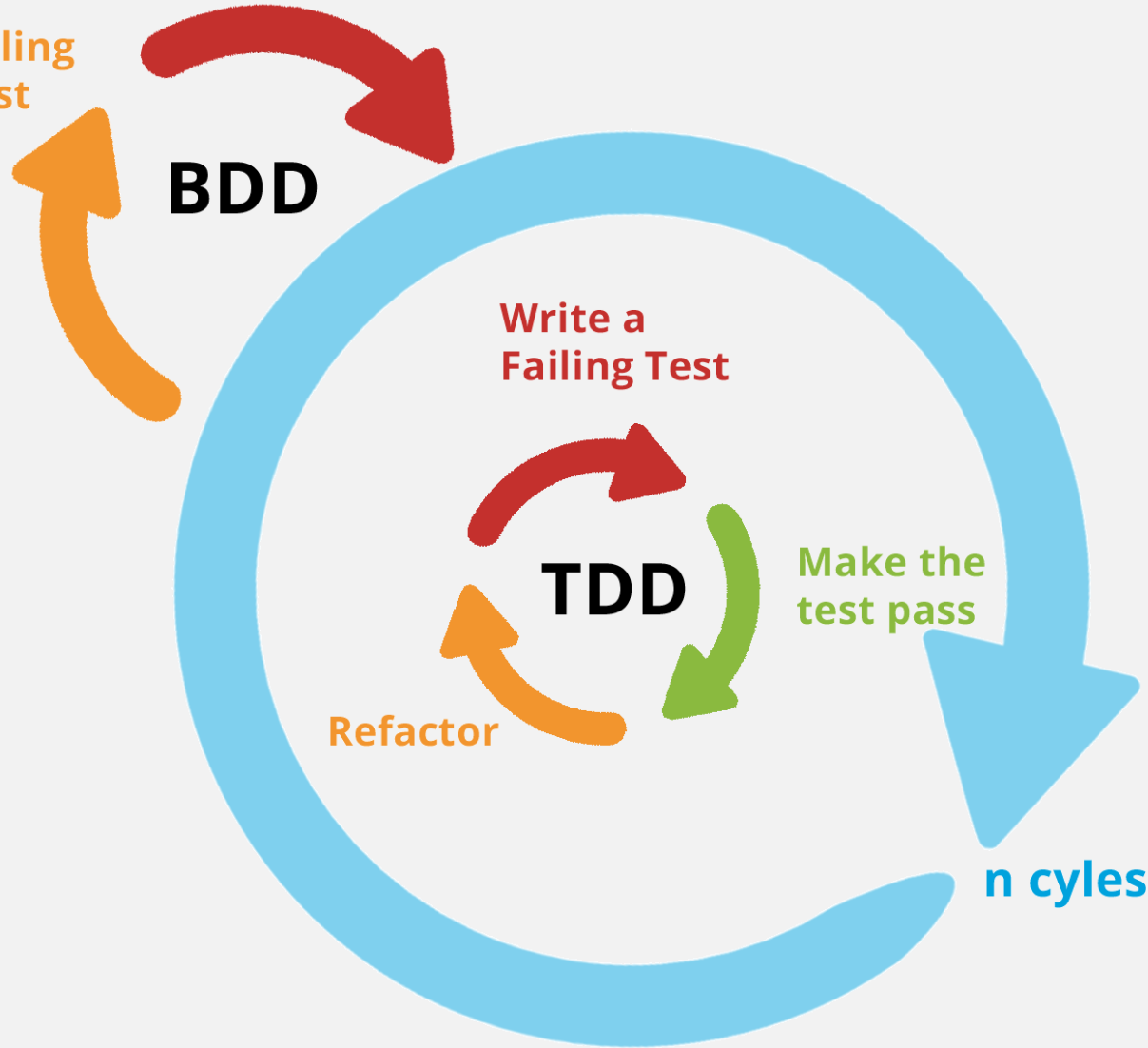
Write a
Failing Test

TDD

Make the
test pass

Refactor

n cycles



[TestMethod]

```
public void ConvertObject_GenericType()
{
    Int16 obj1 = Int16.MaxValue;
    object obj2 = ReflectionHelper.ConvertObject< Int32 >( obj1 );
    Assert.IsInstanceOfType( obj2, typeof( Int32 ) );
    Assert.AreEqual( obj1.ToString(), obj2.ToString() );

    obj1 = default( Int16 );
    obj2 = ReflectionHelper.ConvertObject< Int16 >( null );
    Assert.AreEqual( obj1, obj2 );

    var obj3 = (FGlobals.TestEnum)Enum.GetValues( typeof( FGlobals.TestEnum ) ).GetValue( 0 );
    obj2 = ReflectionHelper.ConvertObject< FGlobals.TestEnum >( obj3.ToString() );
    Assert.AreEqual( obj3, obj2 );

    obj2 = ReflectionHelper.ConvertObject< FGlobals.TestEnum >( obj3 );
    Assert.AreEqual( obj3, obj2 );
}
```

#language: en-US

@WebDriver

Feature: SpbDotNet

Background:

Given у меня есть элементы, присутствующие на веб-странице:

Имя	ID	XPath
Поиск	login__username	
КнопкаПоиск	login__password	//*[@id="tsf"]/div[2]/div[3]/center/input[1]
SpbDotnet	login__submit	//*[@id="rso"]/div/div/div[2]/div/div/div[1]/a/h3

Scenario: Google - SpbDotNet

Given я инициализирую браузер 'GoogleChrome' без UI

Given я открываю веб-страницу 'google.com'

And я развернул веб-страницу на весь экран

And я ввожу в поле 'Поиск' веб-страницы значение 'spb dotnet meetup'

When выполнено нажатие на элемент 'КнопкаПоиск' на веб-странице

Then элемент 'SpbDotnet' отображается на веб-странице

Then выполнен переход на веб-страницу 'spbdotnet.timepad.ru' после нажатия на элемент 'SpbDotnet'

Почему мы должны использовать BDD?

Почему?
Почему?
Почему?
Почему?



Какая часть моего "мяу"
тебе непонятна?

Функция: Веб Сервис - TestService - Метод Get - Проверка пагинации

Для того чтобы получить субъекта

Как пользователь

Я хочу вызывать веб-сервис TestService с методом Get

Предыстория:

Дано параметры вызова метода веб-сервиса на КСШ:

Сервер	Мнемоника	Сервис	Метод	Версия	Код системы
test.server	mnemonic	TestService	Get	1	TEST

Допустим я подключен к БД DataBase:

Драйвер	Сервер	База Данных	Библиотеки	Логин	Пароль
iDB2	iDB2Server	IDB2Database	LIBS	LOGIN	PASS

Сценарий: Проверка пагинации

Когда я заполняю параметры метода веб-сервиса:

Поле	Значение
code	Gender
pageManager.pageNumber	1
pageManager.pageLength	100

И я вызываю веб-сервис на КСШ

Тогда я получаю результат из веб-сервиса

И результат содержит поле *pageManager.totalElements* со значением 2


```
[When(@"я\с+заполняю\s+параметры\s+метода\s+веб-сервиса:")]
```

```
public void When_I_SetWebMethodParams( Table dataTable )
{
    SharedDefs.WS_inParamsKV["default"] = new CKeyValues();
    TextUtils.ParseStructDescription< object >(
        dataTable.Rows.ToArray(),
        ( fld, obj, ex ) =>
        {
            CKeyValues.SetValueHierarchically( SharedDefs.WS_inParamsKV.SingleOrDefault().Value, fld, obj );
            ReporterHelper.ReportEvent( "Задание значения параметров", string.Format( @"{0} = {1};", fld, obj ) );
        }
    );
}
```

```
[When(@"я\s+заполняю\s+параметры\s+метода\s+веб-сервиса\s+(\w+)\s*:")]
```

```
public void When_I_SetWebMethodParams(string alias , Table dataTable)
{
    SharedDefs.WS_inParamsKV[alias] = new CKeyValues();
    TextUtils.ParseStructDescription< object >(
        dataTable.Rows.ToArray(),
        ( fld, obj, ex ) =>
        {
            CKeyValues.SetValueHierarchically( SharedDefs.WS_inParamsKV[alias], fld, obj );
            ReporterHelper.ReportEvent( "Задание значения параметров", string.Format( @"{0} = {1};", fld, obj ) );
        }
    );
}
```





Gherkin



Given

When

Then

Given

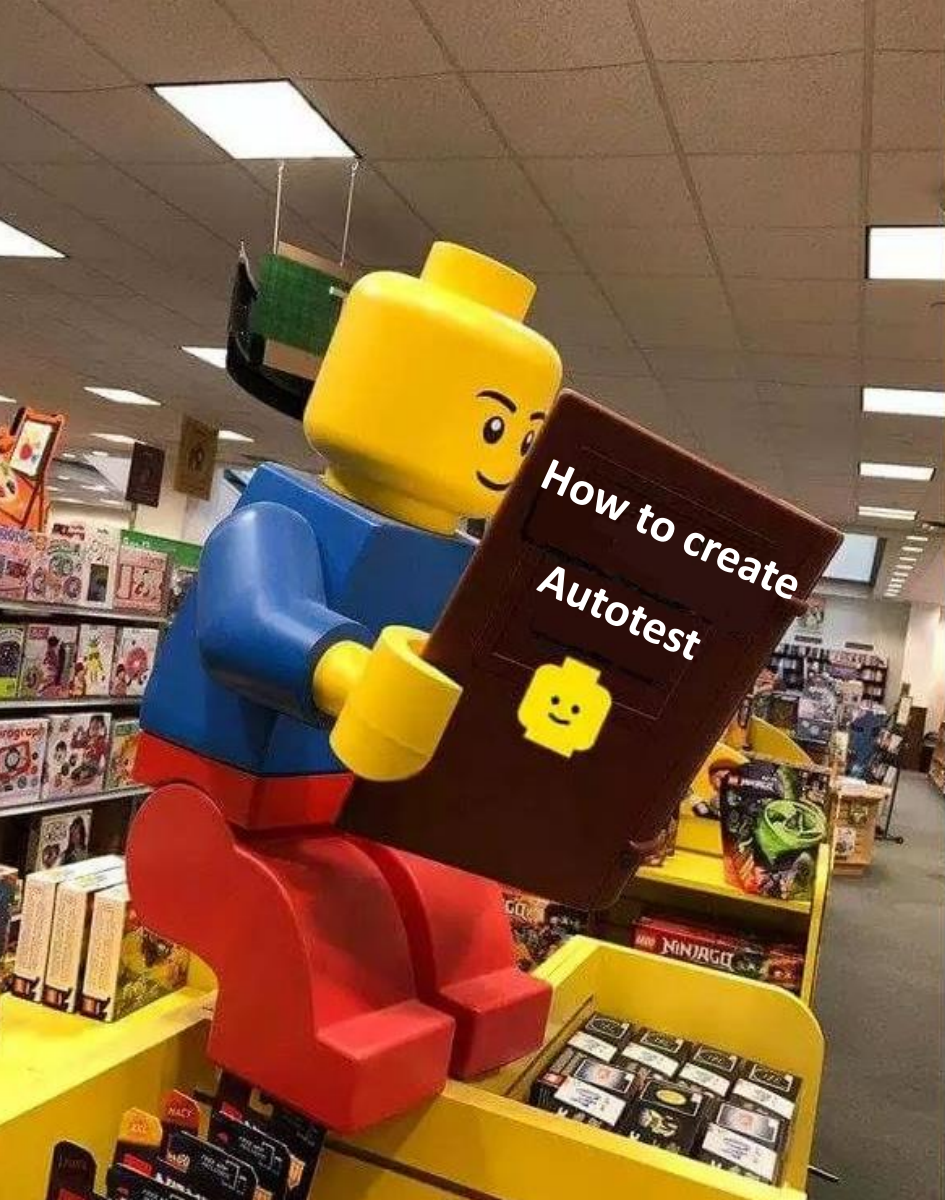
Я открываю
web страницу

When

Перехожу по
ссылке

Then

Отображается
расписание
докладов



```
using OpenQA.Selenium;
```

```
public IWebDriver browser { get; set; }
```

```
[BeforeScenario]
```

```
public void BeforeScenario()
```

```
{
```

```
    browser = new ChromeDriver();
```

```
}
```

```
[AfterScenario]
```

```
public void AfterScenario()
```

```
{
```

```
    browser.Quit();
```

```
}
```

```
using OpenQA.Selenium;
```

```
public IWebDriver browser { get; set; }
```

```
[BeforeScenario]
```

```
public void BeforeScenario()
```

```
{
```

```
    browser = new ChromeDriver();
```

```
}
```

```
[AfterScenario]
```

```
public void AfterScenario()
```

```
{
```

```
    browser.Quit();
```

```
}
```

```
using OpenQA.Selenium;
```

```
public IWebDriver browser { get; set; }
```

```
[BeforeScenario]
```

```
public void BeforeScenario()
```

```
{
```

```
    browser = new ChromeDriver();
```

```
}
```

```
// POST /session
```

```
[AfterScenario]
```

```
public void AfterScenario()
```

```
{
```

```
    browser.Quit();
```

```
}
```

```
using OpenQA.Selenium;
```

```
public IWebDriver browser { get; set; }
```

```
[BeforeScenario]
```

```
public void BeforeScenario()
```

```
{
```

```
    ChromeOptions chromeOptions = new ChromeOptions();
```

```
    chromeOptions.AddArgument( "--headless" );
```

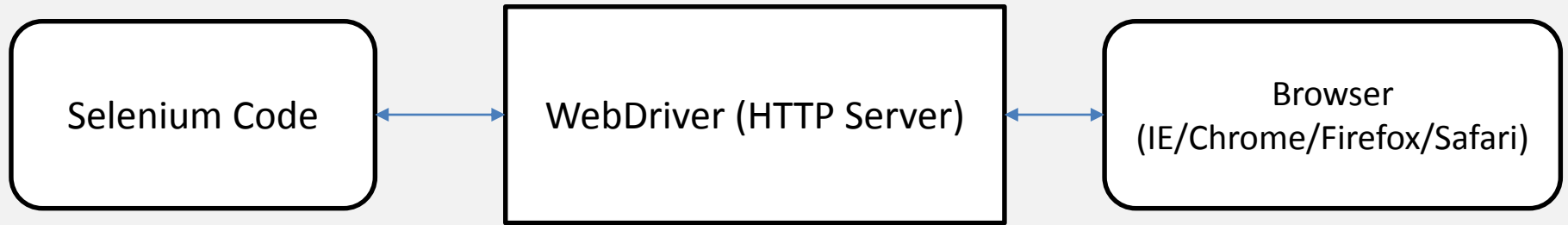
Without UI

```
    chromeOptions.AddArgument( "--disable-gpu" );
```

```
    chromeOptions.AddArgument( "--window-size=1920,1080" );
```

```
    browser = new ChromeDriver(chromeOptions);
```

```
}
```

POST /session

GET /status

DELETE /session /:sessionId

POST /session /:sessionId /timeouts /implicit wait

GET /session /:sessionId /url

```
[StepDefinition(@"я открываю веб-страницу '(.+)'")]  
public void StepDef_I_OpenSite( string site )  
{  
    browser.Navigate().GoToUrl( "http://" + site );  
}
```

```
[StepDefinition(@"я открываю веб-страницу '(.+)'")]  
public void StepDef_I_OpenSite( string site )  
{  
    browser.Navigate().GoToUrl( "http://" + site );  
}
```

```
[StepDefinition(@"я открываю веб-страницу '(.+)'")]  
public void StepDef_I_OpenSite( string site )  
{  
    browser.Navigate().GoToUrl( "http://" + site ); // GET /session/:sessionId/url  
}
```

```
[Given(@"я ввожу в поле '(.+)' веб-страницы значение '(.+)')"]  
public void Given_I_Input_Into_Field_By( string element, string value )  
{  
    var Page_Element = Web.GetPageObjectElement( element );  
    if( Page_Element != null )  
    {  
        Page_Element.Clear();  
        Page_Element.SendKeys( value );  
    }  
    else  
    {  
        Assert.Fail( "Элемент '{0}' не был найден", element );  
    }  
}
```

```
[Then(@"элемент '(.+)' отображается на веб-странице")]
public void Then_I_View_With_Element_By( string element )
{
    var Page_Element = Web.GetPageObjectElement( element );
    if( Page_Element != null )
    {
        Assert.IsTrue( Page_Element.Displayed );
    }
    else
    {
        Assert.Fail( "Элемент '{0}' не был найден", element );
    }
}
```



```

[Scope( Tag = "WebService" )]
[When(@"я вызываю метод \""(.+)\\"" веб-сервиса \""(.+)\\"" с JSON запросом:")]
public void When_I_SendToWS_JSON( string method, string url, string json )
{
    var json = SharedDefs.ReplaceVariables( json, ( val ) =>
        HttpUtility.JavaScriptStringEncode( Reflection.ConvertObject< string >( val ) ), (
val ) => $"{{{val}}}" );

    SharedDefs.WS_lastResponse = new SharedDefs.WebMethodResponse(
        WebServiceRaw.CallWebServiceMethod_Raw(
            url,
            method,
            WebServiceType.REST,
            json
        )
    );
}

```

#language: en-US

@WebDriver

Feature: Alfabank

Background:

Scenario: Alfa

Given я открываю веб-страницу '*google.com*'

Given я ввожу в поле '*Поиск*' веб-страницы значение '*Альфабанк*'

Then элемент '*alfabank.ru*' отображается на веб-странице

#language: en-US

@WebDriver

Feature: AlfaBank

Background:

Scenario: Alfa

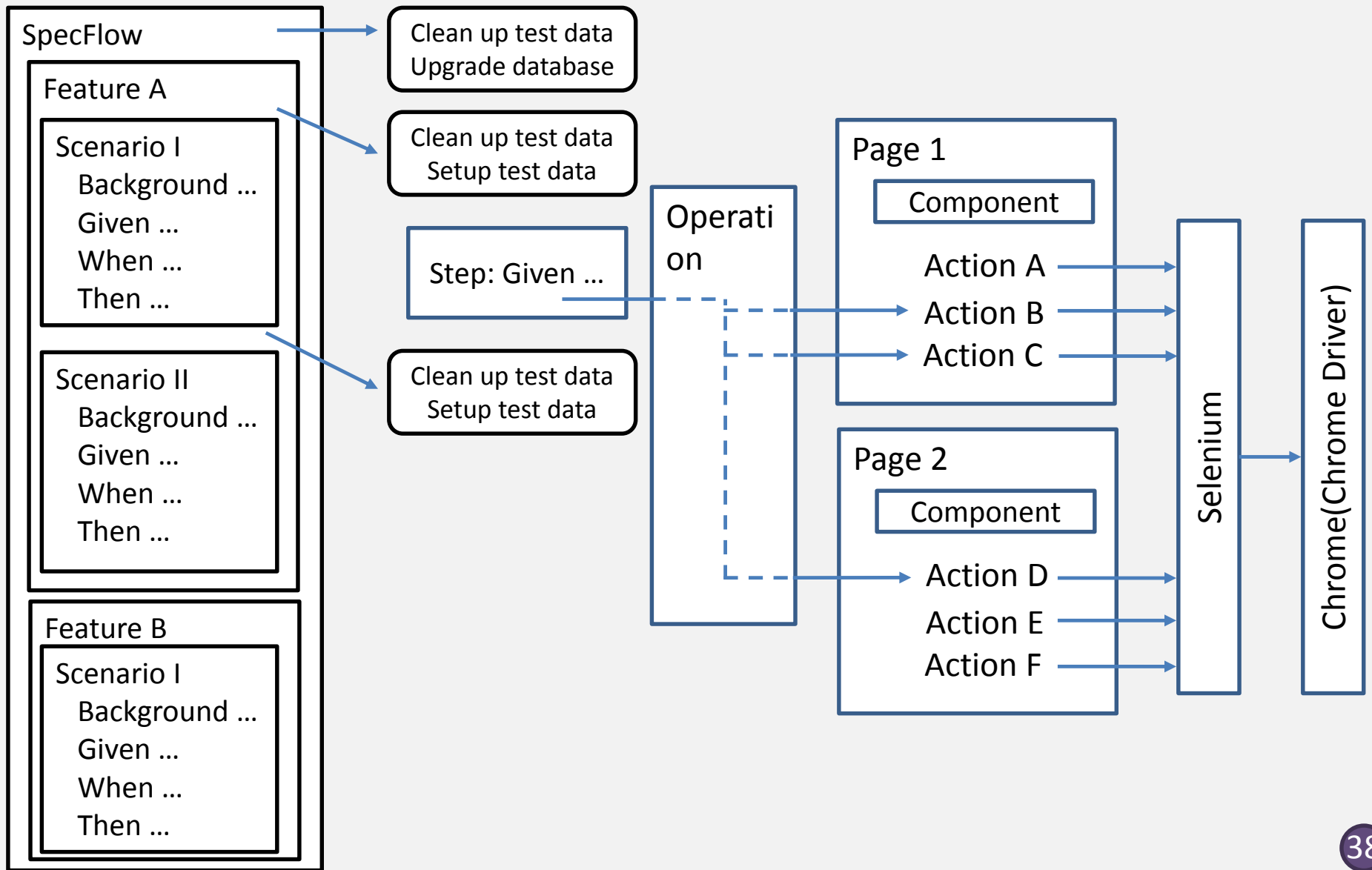
Given я открываю веб-страницу 'google.com'

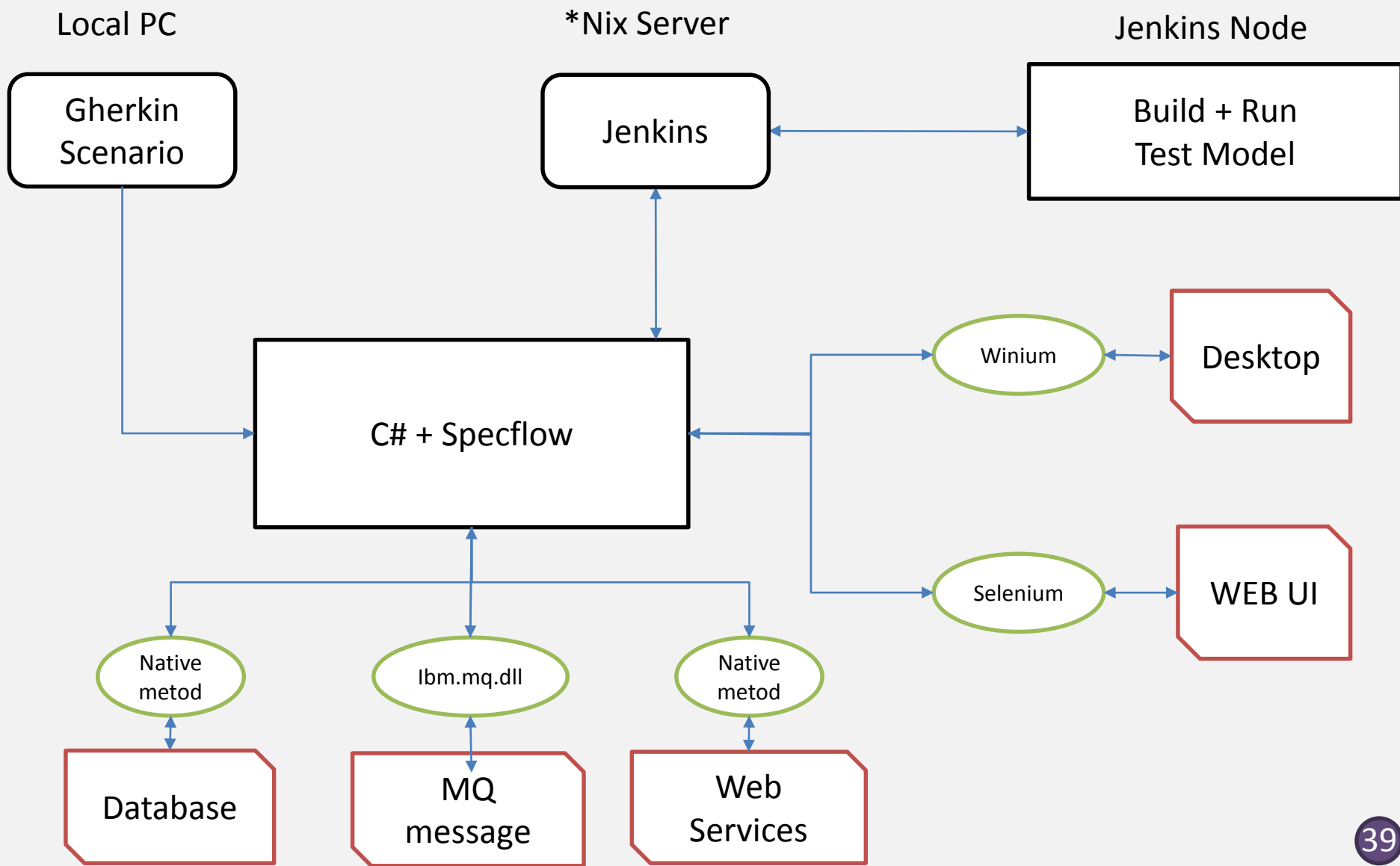
Given я ввожу в поле 'Поиск' веб-страницы значение 'Альфабанк'

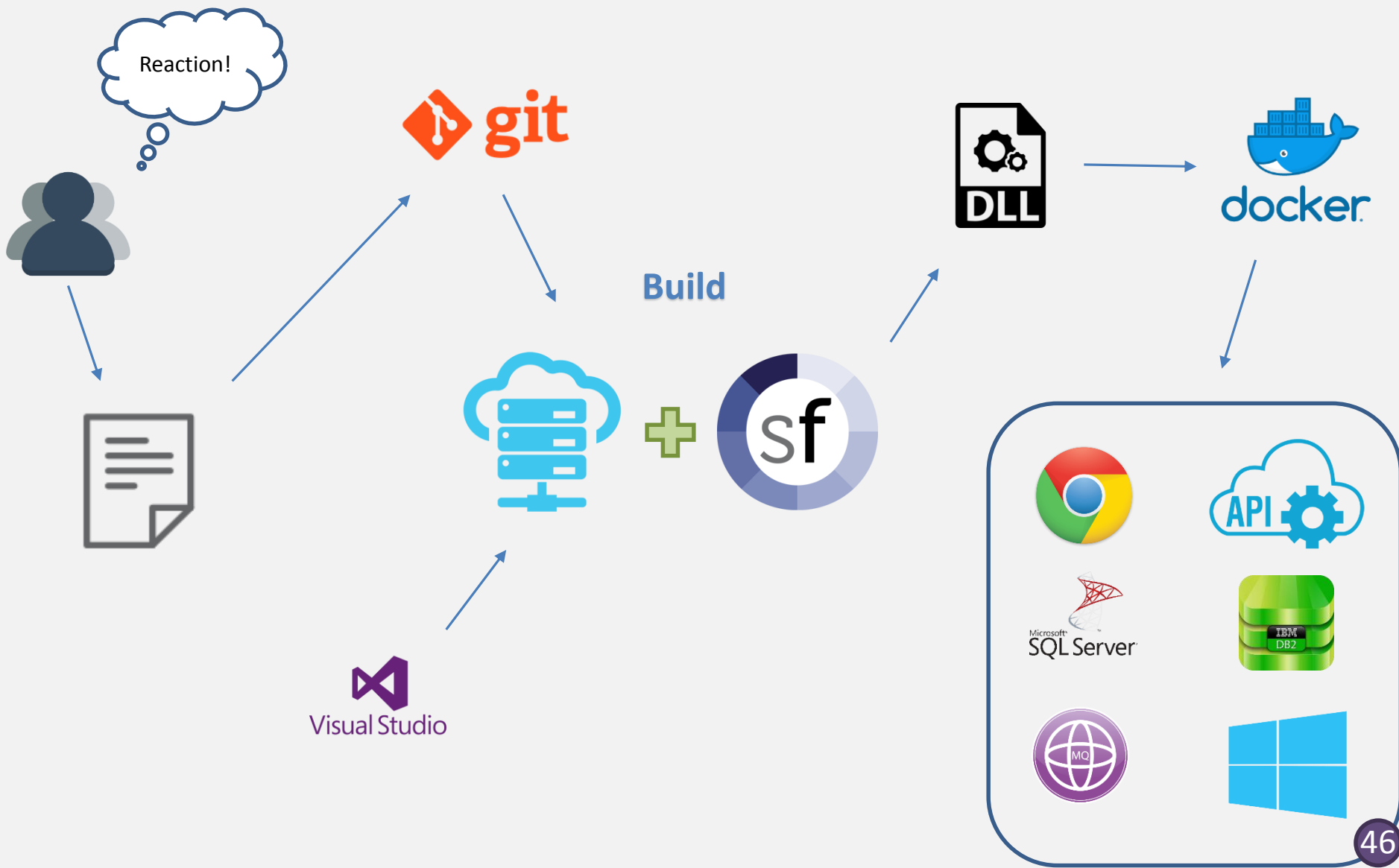
Then элемент 'alfabank.ru' отображается на веб-странице

Then

```
[T] [ОТЛАДКА: показать тип переменной "{varName}"] -> Steps_Common.StepDef_DEBUG_DisplayVariableType(String)
[T] [ОТЛАДКА: показать значение переменной "{varName}"] -> Steps_Common.StepDef_DEBUG_DisplayVariableValue(String)
[T] [ОТЛАДКА: зашифровать текст с ключем "{salt}":] -> Steps_Common.StepDef_DEBUG_EncodeText(String, String)
[T] [я жду {seconds} сек\.] -> Steps_Common.StepDef_I_WaitForSeconds(Int32)
[T] [я удаляю переменную "{varName}"] -> Steps_Common.StepDef_I_DeleteVariable(String)
[T] [я убеждаюсь, что значение переменной "{varName}" не является NULL] -> Steps_Common.StepDef_I_CheckVariable_IsNotNull(String)
[T] [я убеждаюсь, что значение переменной "{varName}" является NULL] -> Steps_Common.StepDef_I_CheckVariable_IsNULL(String)
[T] [я убеждаюсь, что значение переменной "{varName}" является текстом] -> Steps_Common.StepDef_I_CheckVariable_IsString(String)
[T] [я убеждаюсь, что значение переменной "{varName}" равно "{expected}"] -> Steps_Common.StepDef_I_CheckVariable_Equals(String, Object)
[T] [я убеждаюсь, что значение переменной "{varName}" содержит "{expected}"] -> Steps_Common.StepDef_I_CheckVariable_Contains(String, String)
[T] [я убеждаюсь, что значение переменной "{varName}" начинается с "{expected}"] -> Steps_Common.StepDef_I_CheckVariable_StartsWith(String, String)
[T] [я убеждаюсь, что значение переменной "{varName}" заканчивается с "{expected}"] -> Steps_Common.StepDef_I_CheckVariable_EndsWith(String, String)
[T] [я убеждаюсь, что значение переменной "{varName1}" равно значению переменной "{varName2}"] -> Steps_Common.StepDef_I_CheckVariable_Equals(String, String)
[T] [я убеждаюсь, что значение переменной "{varName1}" идентично значению переменной "{varName2}"] -> Steps_Common.StepDef_I_CheckVariable_IsIdentical(String, String)
[T] [я убеждаюсь, что значение "{value}" json переменной "{varName}" не является NULL] -> Steps_Common.Then_I_CheckJsonValue_IsNotNull(String, String)
[T] [я убеждаюсь, что значение "{value}" json переменной "{varName}" является NULL] -> Steps_Common.Then_I_CheckJsonValue_IsNULL(String, String)
```

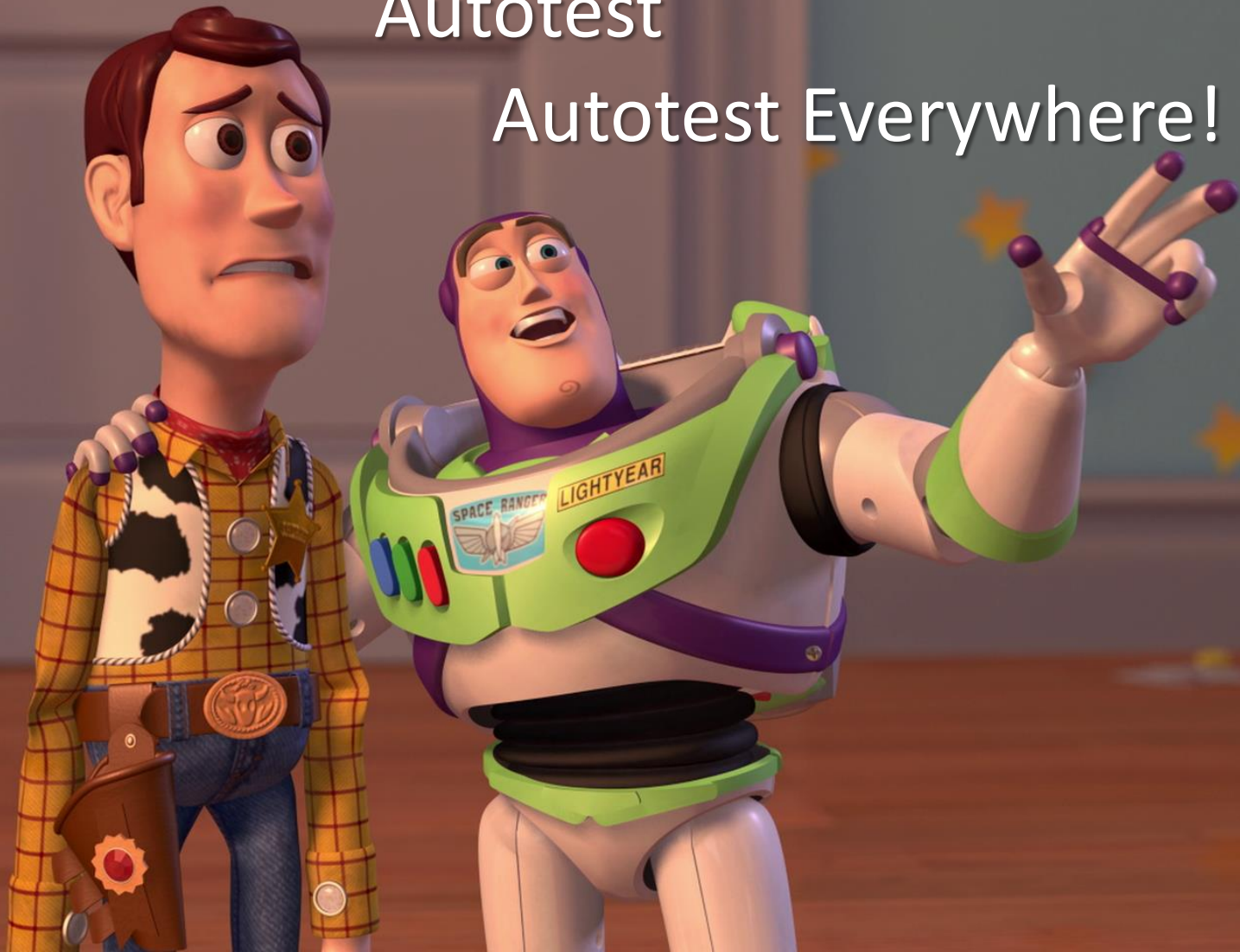






Autotest

Autotest Everywhere!





Alfa·Bank