

Минск. 15 – 16 ноября
2019

Автоматизация тестирования десктоп- приложений

Кюзалова Карина

ХКФ Банк. Москва, Россия

О себе



Раньше

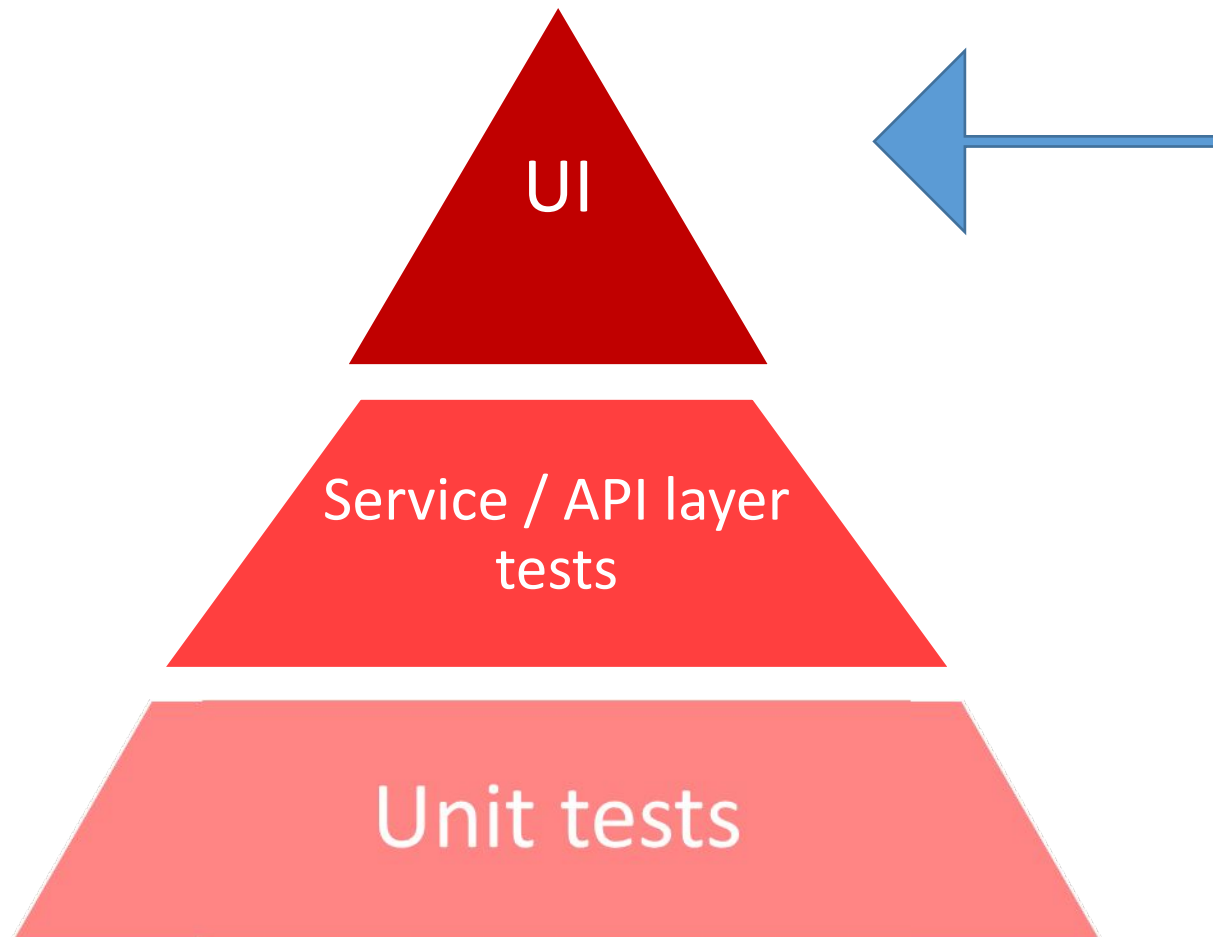


Сейчас



Главный специалист
автоматизированного
тестирования
кредитной системы

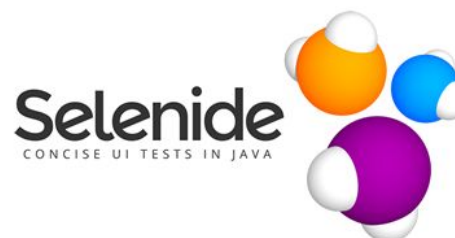
Уровни автоматизации



UI тестирование

- Web-тестирование
- Тестирование мобильных приложений
- Тестирование десктоп-приложений

Web



Mobile



Appium



Robotium



eggplant



UI TESTING FOR ANDROID
espresso

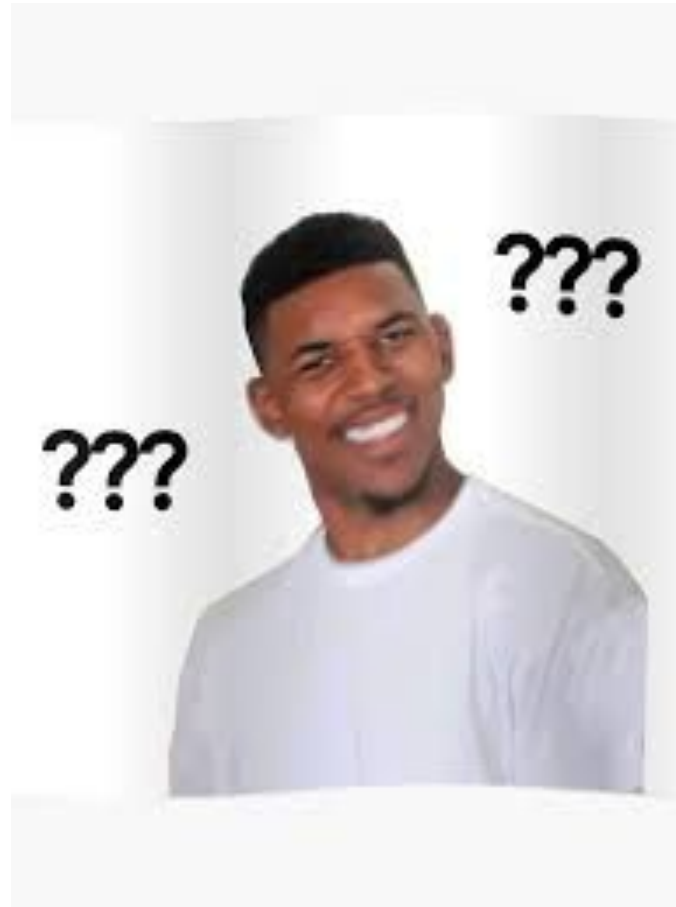


TestComplete
by SMARTBEAR



Ranorex

Desktop



Особенности десктоп-приложений

- Должно быть развернуто и установлено
- Зависят от операционной системы
- Должны быть достаточные аппаратные ресурсы компьютера

Desktop Testing



Инструменты



Поиск элементов



Фреймворк

Поиск элементов

Критерии	UISPY	Inspect	FlauInspect, Testcomplete UI и т.д
.NET Framework	.NET 2	.NET 2 .NET 3	.NET 2 .NET 3
Недостатки	Нет поддержки	Находит не все свойства элементов	Находит не все свойства элементов
	устарел	Постоянно зависает	Работает только со своим fw

Платные фреймворки



TestComplete
by SMARTBEAR



TestArchitect™



Telerik

Test Studio

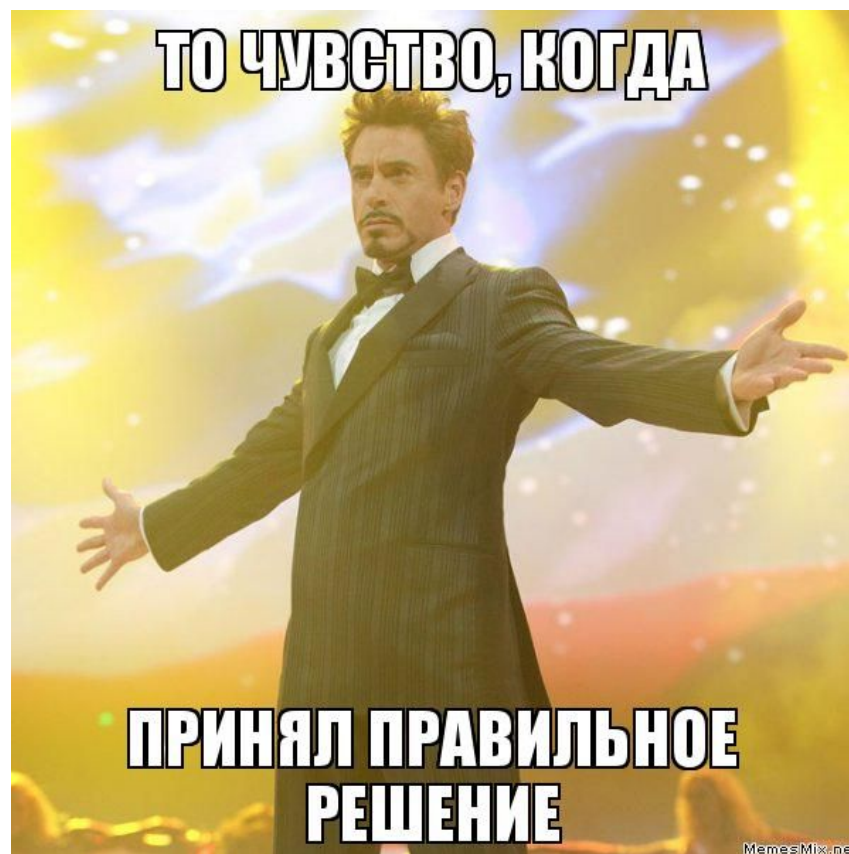
Плюсы

- быстрый старт
- обучение, хорошая поддержка
- широкий спектр платформ

Минусы

- Стоимость
- Невозможность кастомизировать (нет исходного кода)

Фреймворки с ОТКРЫТЫМ КОДОМ



Фреймворки

«Нестандартные» фреймворки

1. SikuliX
2. Robot framework

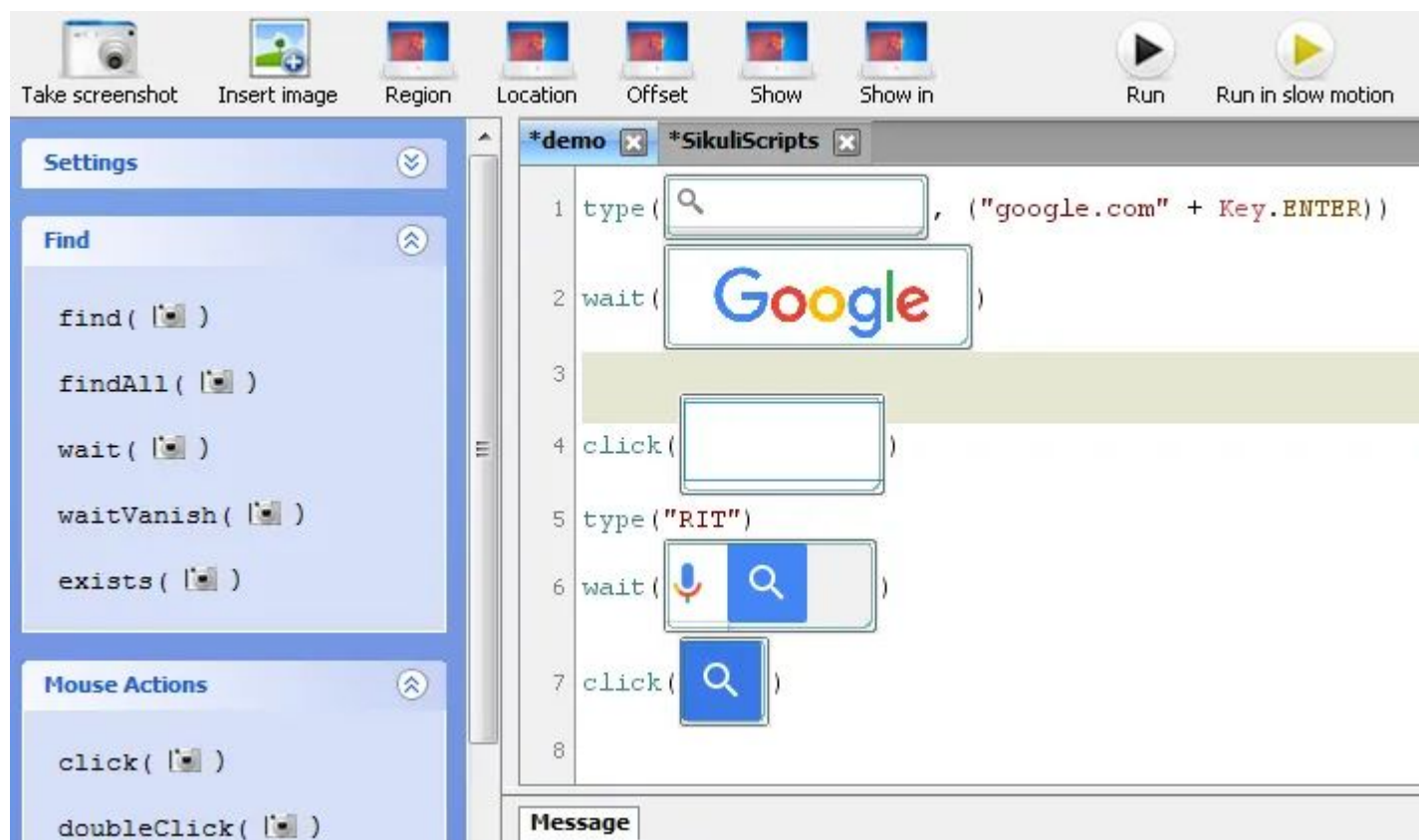
UI library

1. FlaUI

Selenium WebDriver

1. WinApp+Appium
2. Winium

SikuliX



Пример

```
click("1534932430966.png")  
sleep(1)  
type("cmd\n")  
sleep(1)  
type("notepad\n")  
sleep(1)  
type("Hello! From SikuliX")
```

Robot framework

```
PO > SignIn.robot >
SignIn.robot x
*** Settings ***
Library Selenium2Library

*** Keywords ***
Login To Site
    input text    name=username    admin
    input text    name=password    test@123
    click button  id=submit
```

Пример

*** Settings ***

| Library | AutoltLibrary

*** Variables ***

| \${notepad++} | C:\\ProgramFiles\\Notepad++\\notepad++.exe

*** Test Cases ***

| Example |

| Run | \${notepad++}

Flaui



FlaUI.Core 2.0.0-rc1

Library with base elements used in the concrete implementations of FlaUI.

ⓘ This is a prerelease version of FlaUI.Core.

Package Manager

.NET CLI

PackageReference

Paket CLI

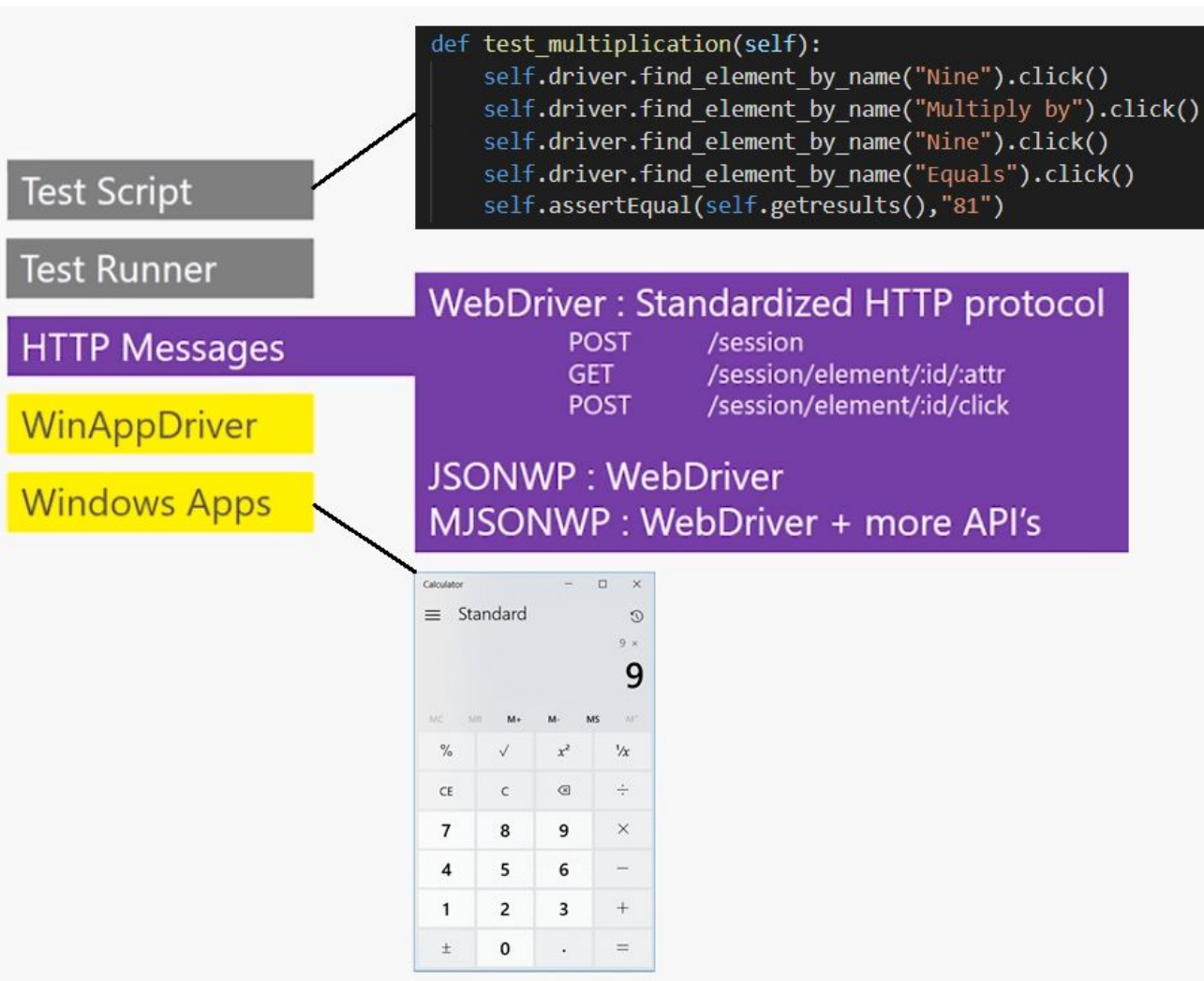
```
PM> Install-Package FlaUI.Core -Version 2.0.0-rc1
```



Пример

```
using FlaUI.UIA3;
var app = FlaUI.Core.Application.Launch("notepad.exe");
using (var automation = new UIA3Automation()) {
    var window = app.GetMainWindow(automation);
    var button1 = window.FindFirstDescendant(cf =>
cf.ByText("Save"))?.AsButton();
    button1?.Invoke();
}
```

WinAppDriver

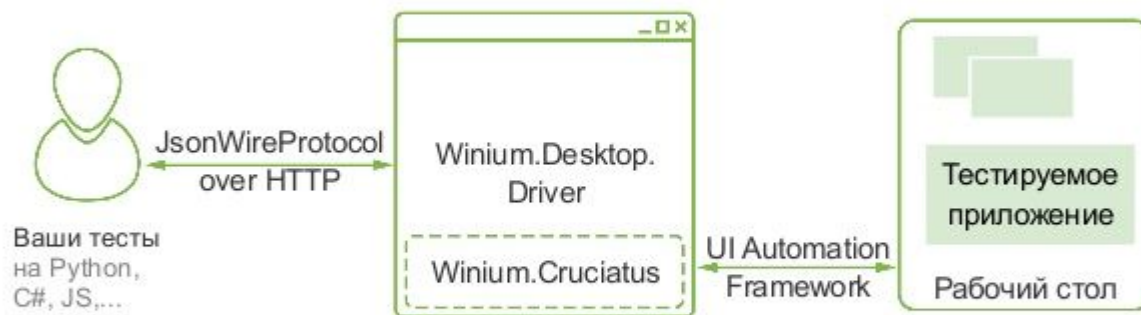


Пример

```
DesiredCapabilities capabilities = new DesiredCapabilities();  
capabilities.setCapability("app", "Microsoft.WindowsNotepad");  
NotepadSession = new WindowsDriver(new  
URL("http://localhost:8080"), capabilities);
```


Winium

Selenium for Windows Phone & Desktop



Пример

```
String apppath = "C:\\Windows\\System32\\notepad.exe";  
String winiumdriverpath = "src\\Winium.Desktop.Driver.exe";  
new DesktopOptions().setApplicationPath(apppath);  
File drivePath = new File(winiumdriverpath);  
WiniumDriverService service = new WiniumDriverService.  
Builder(). usingDriverExecutable(drivePath)  
.usingPort(8080).buildDesktopService();  
service.start();  
WiniumDriver driver = new WiniumDriver(service, options);
```

Сравнение

	Поддержка	Скорость поиска	Передача метаданных	ОС	Стоимость	Инструмент поиска
Test Complete	хорошая	быстро (ок. 1 мин)	да	Windows	платно	есть
Test Architect	хорошая	Быстрый (ок. 1 мин)	да	Windows Linux MacOS	платно	есть
SikuliX	плохая	медленно	-	Windows Linux MacOS	бесплатно	-
Robot Framework	плохая	медленно	-	Windows Linux MacOS	бесплатно	-

Сравнение

	Поддержка	Скорость поиска	Передача метаданных	ОС	Стоимость	Инструмент поиска
FlaUI	хорошая	Быстрый (ок. 1мин)	да	Windows	бесплатно	есть
WinApp	хорошая	Медленно (ок. 3 мин)	нет	Windows	бесплатно	нет
Winium	нет	Медленно (ок. 3 мин)	да	Windows	бесплатно	нет

Трудности с FlaUI

Высокий порог входа в проект – реализация сильно отличается от автотестов для веб-приложений, нужно понимание как работает драйвер

API Webdriver



Итог



Проблемы с кастомным драйвером

Нет реализации некоторых команд

Решение

1. Доработка существующих команд
2. Реализация новых команд

Доработка

Необходимо добавить или изменить реализацию команды
Пример – добавить кастомный поиск элемента по определенному локатору

Решение

1. Находим FindElementExecutor
2. Получаем стратегию поиска

```
var searchStrategy =  
    ExecutedCommand.Parameters["using"].ToString();
```

3. Реализуем логику

```
if (searchStrategy.Equals("name")){  
    //твоя реализация  
}
```

Добавить новую команду

1. Реализация команды – добавить в CommandExecutors
2. Включить в команды драйвера
3. Добавление команды в список команд
4. Добавление команды в словарь – связываем реализацию и API Webdriver

Реализация команды

```
internal class GetElementRectExecutor : CommandExecutorBase
{
    protected override string DoImpl()
    {
        var registeredKey = this.ExecutedCommand.Parameters["ID"].ToString();
        var element=this.Automator.ElementsRegistry.GetRegisteredElement(registeredKey);
        DriverManager.GetActiveWindow();
        var boundingRect = element.FlaUIElement.Properties.BoundingRectangle.Value;
        var response = new Dictionary<string, object>
        {
            { "x", boundingRect.Left},
            { "y", boundingRect.Top},
            { "width", boundingRect.Width },
            { "height", boundingRect.Height }
        };
        return this.JsonResponse(ResponseStatus.Success, response);
    }
}
```

Включить в драйвер

FlaUIDesktopDriver.csproj

```
<ItemGroup>  
  <Compile  
    Include="CommandExecutors\GetElementRectExecutor.cs" />  
</ItemGroup>
```

Добавление в список

DriverCommand.cs

```
public static readonly string GetElementRect = "getElementRect";
```

Добавление в словарь

UriDispatchTables.cs

```
private readonly Dictionary<string, CommandInfo> commandDictionary = new  
Dictionary<string, CommandInfo>();
```

```
public UriDispatchTables(Uri prefix)
```

```
{  
    this.InitializeSeleniumCommandDictionary();  
}
```

```
private void InitializeSeleniumCommandDictionary()
```

```
{  
    this.commandDictionary.Add(DriverCommand.GetElementRect, new  
CommandInfo("GET", "/session/{sessionId}/element/{id}/rect"));  
}
```


ИТОГИ

Поиск элементов

UISpy, Inspect, «свой» инструмент

Фреймворки

1. Нестандартные (SikuliX, Robot Framework)
2. UI Library (**Flauti**)
3. Selenium WebDriver (Winium, WinApp)

Окупаемость

Окупаемость разработки автотестов

Всего 94 кейса

Трудоемкость:

ручного тестирования – 24,82Md/регресс

разработки автотестов – 358Md

$358/24,82 = 14,4$ регресса = 7,2 месяца

Поддержка(актуализация+фиксы)

1 неделя в месяц

Внедрение и доработка драйвера

Внедрение – 3 месяца

Доработка – 27 задач*3Md=81Md

Информация

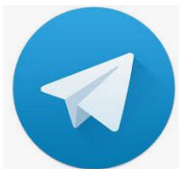
Ссылки:

1. Спецификация WebDriver API -

<https://www.w3.org/TR/webdriver/>

2. Документация UI-Automation -

<https://docs.microsoft.com/en-us/dotnet/framework/ui-automation/ui-automation-fundamentals>



@kyuzalovako



Karina
Kiuzalova

Всем спасибо!

