BUC testing

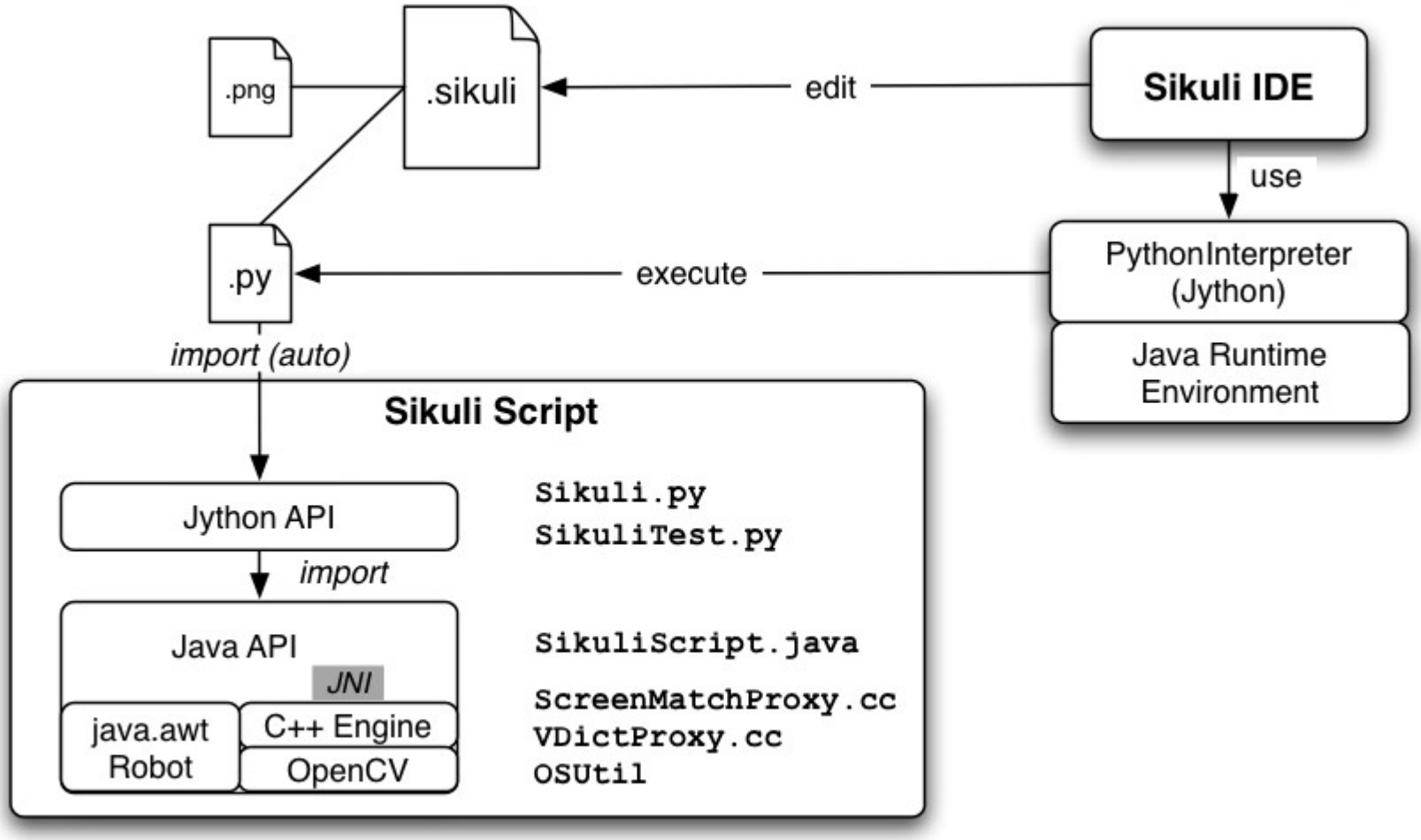
organizacja automatycznych testów funkcjonalnych, na przykładzie Sikuli

Narzędzia typu record-playback

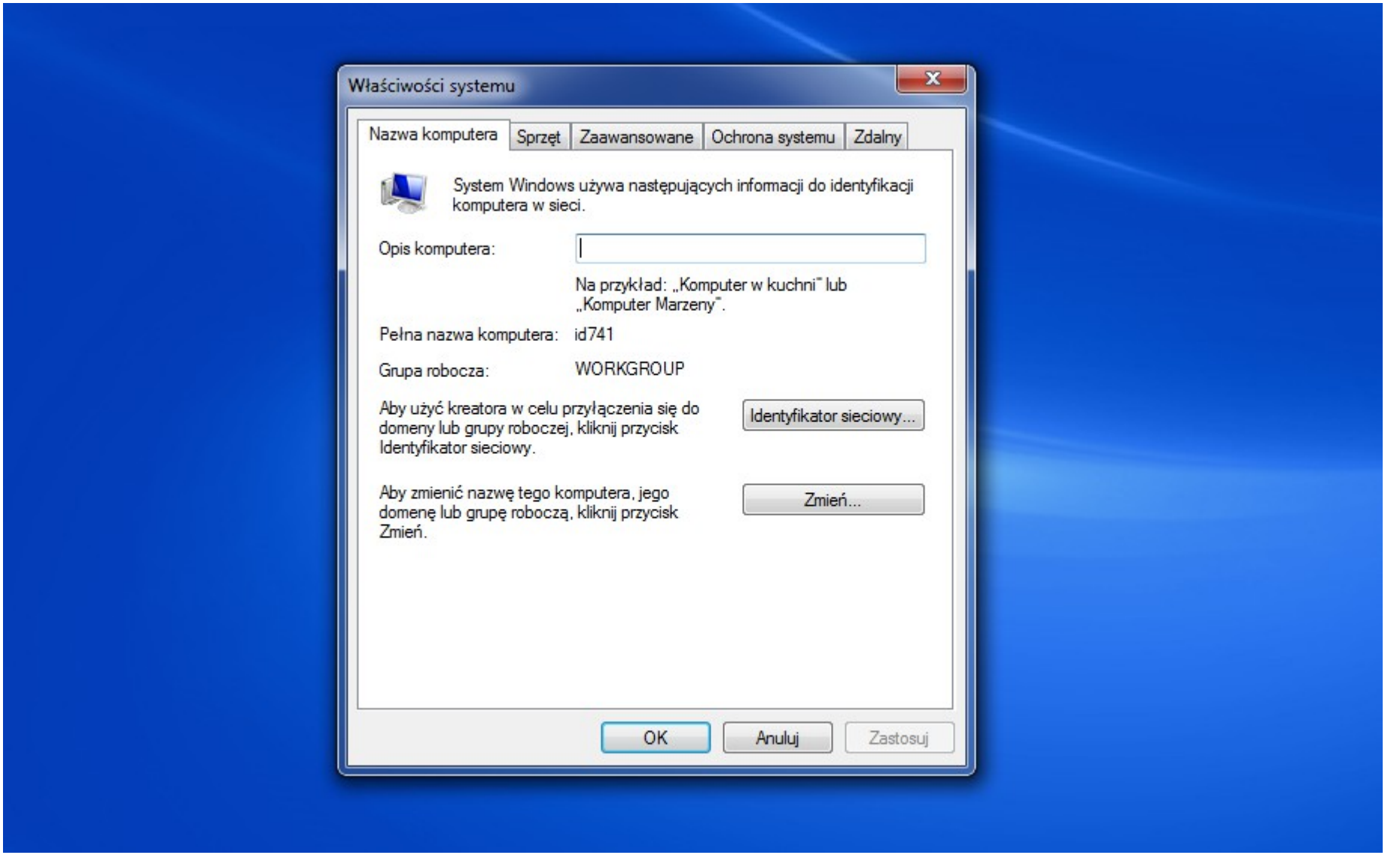
Sikuli

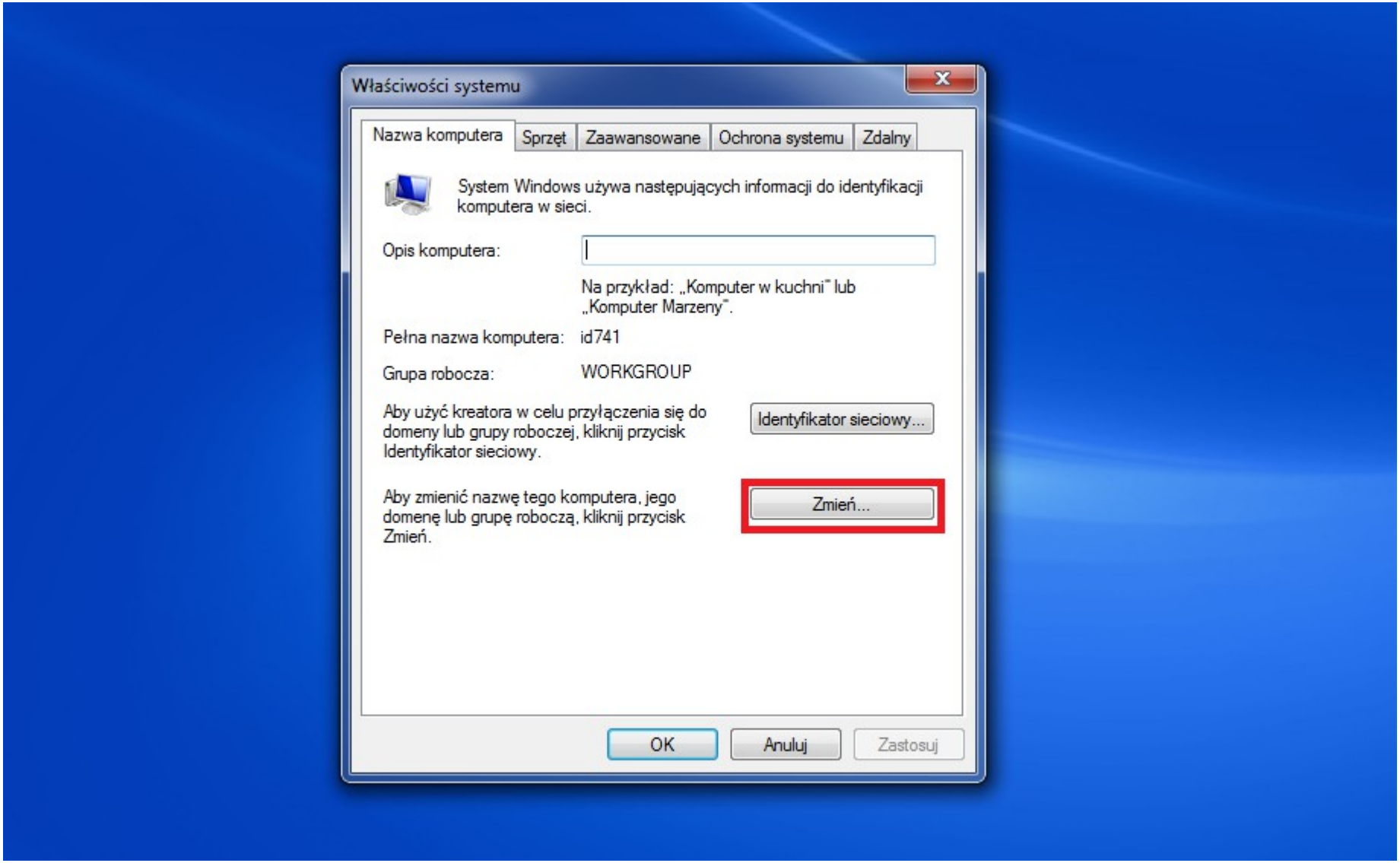
Sikuli

http://www.sikuli.org/


.png
.sikuli
edit
Sikuli IDE
use
.py
execute
PythonInterpreter
(Jython)
Java Runtime
Environment
import (auto)
Sikuli Script
Jython API
import
Java API
JNI
java.awt
Robot
C++ Engine
OpenCV
Sikuli.py
SikuliTest.py
SikuliScript.java
ScreenMatchProxy.cc
VDictProxy.cc
OSUtil

Rozpoznawanie elementów GUI


Właściwości systemu
Nazwa komputera
Sprzęt
Zaawansowane
Ochrona systemu
Zdalny
System Windows używa następujących informacji do identyfikacji komputera w sieci.
Opis komputera:
Na przykład: „Komputer w kuchni" lub „Komputer Marzeny".
Pełna nazwa komputera: id741
Grupa robocza: WORKGROUP
Aby użyć kreatora w celu przyłączenia się do domeny lub grupy roboczej, kliknij przycisk Identyfikator sieciowy.
Identyfikator sieciowy...
Aby zmienić nazwę tego komputera, jego domenę lub grupę roboczą, kliknij przycisk Zmień.
Zmień...
OK
Anuluj
Zastosuj

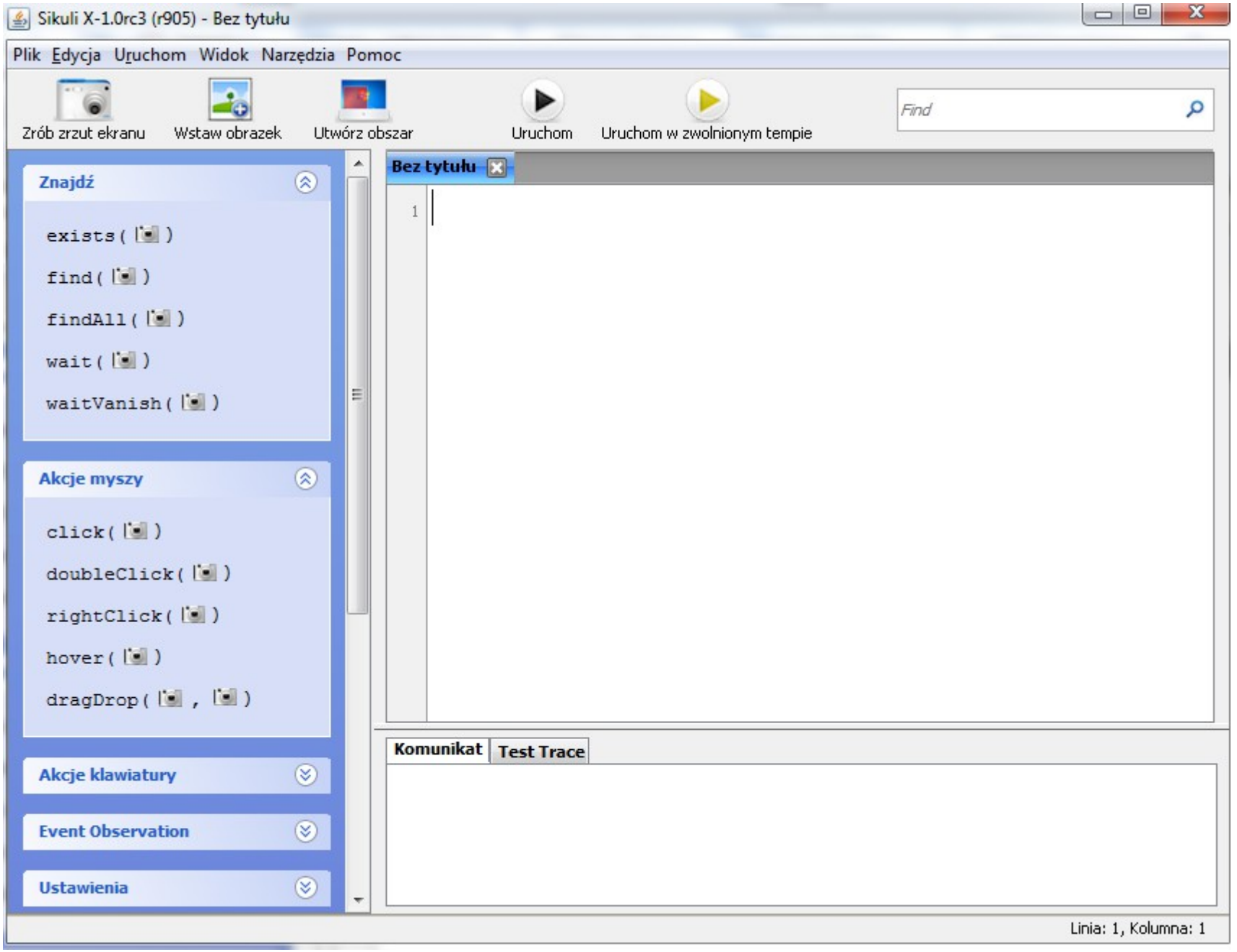

Właściwości systemu
Nazwa komputera
Sprzęt
Zaawansowane
Ochrona systemu
Zdalny
System Windows używa następujących informacji do identyfikacji komputera w sieci.
Opis komputera:
Na przykład: „Komputer w kuchni" lub „Komputer Marzeny".
Pełna nazwa komputera: id741
Grupa robocza: WORKGROUP
Aby użyć kreatora w celu przyłączenia się do domeny lub grupy roboczej, kliknij przycisk Identyfikator sieciowy.
Identyfikator sieciowy...
Aby zmienić nazwę tego komputera, jego domenę lub grupę roboczą, kliknij przycisk Zmień.
Zmień...
OK
Anuluj
Zastosuj

Sikuli X-1.0(r905):

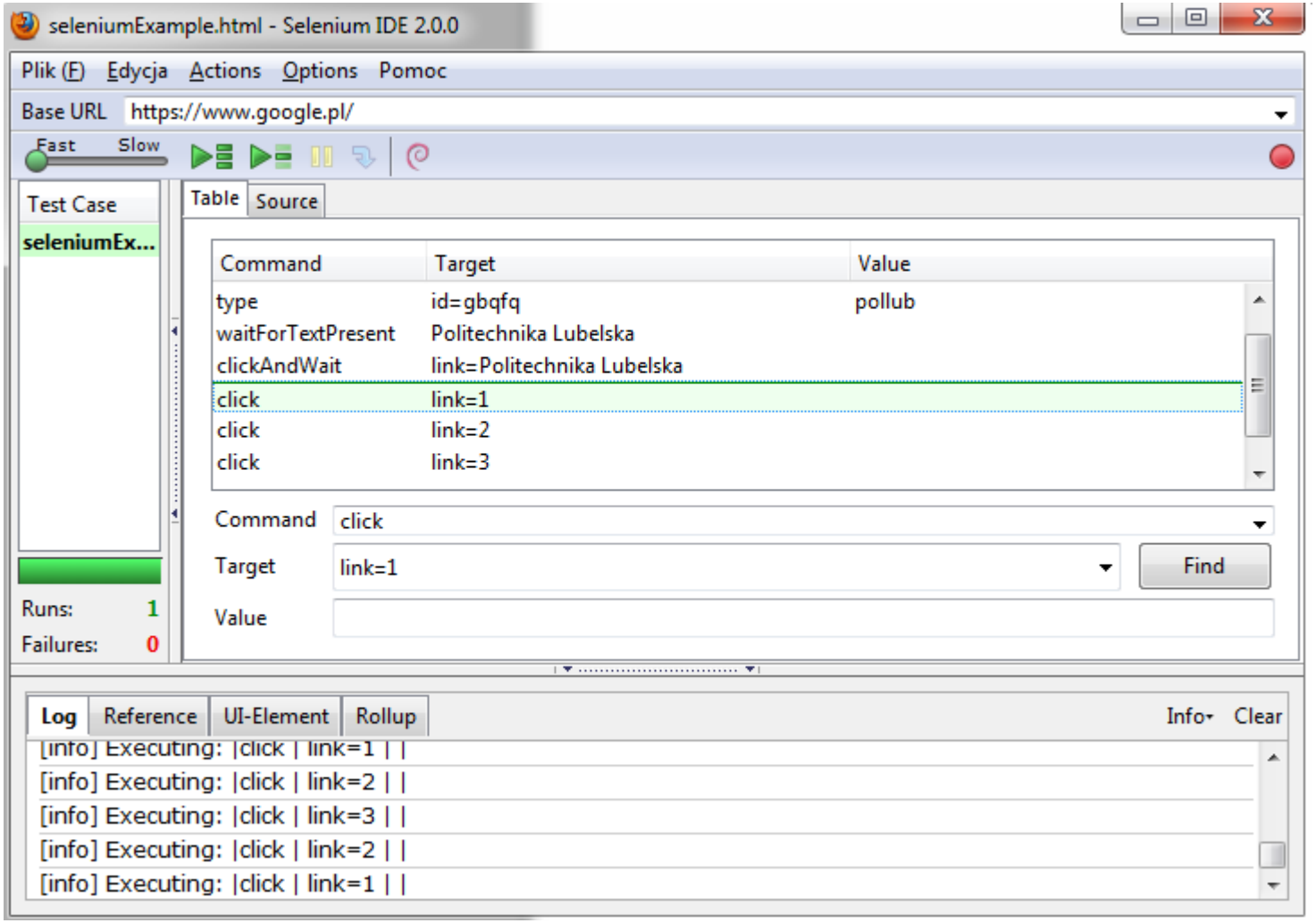
Sikuli X-1.0(r905):
- Sikuli Script (sikuli-script.jar)

Sikuli X-1.0(r905):
- Sikuli Script (sikuli-script.jar)
- Moduł Jython (API do pisania skryptów)

Sikuli X-1.0(r905):
- Sikuli Script (sikuli-script.jar)
- Moduł Jython (API do pisania skryptów)
- Sikuli IDE

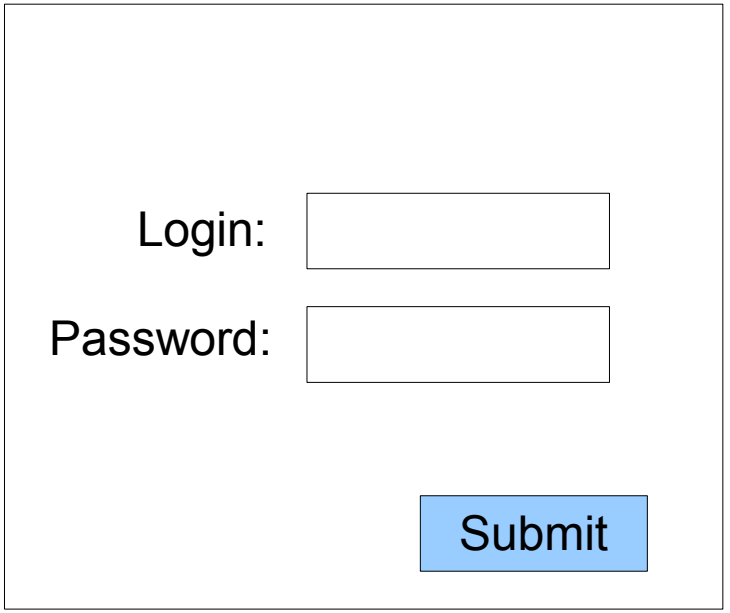

Sikuli X-1.0rc3 (r905) - Bez tytułu
Plik Edycja Uruchom Widok Narzędzia Pomoc
Zrób zrzut ekranu
Wstaw obrazek
Utwórz obszar
Uruchom
Uruchom w zwolnionym tempie
Find
Znajdź
exists( )
find( )
findAll( )
wait( )
waitVanish( )
Akcje myszy
click( )
doubleClick( )
rightClick( )
hover( )
dragDrop(,)
Akcje klawiatury
Event Observation
Ustawienia
Bez tytułu
1
Komunikat
Test Trace
Linia: 1, Kolumna: 1

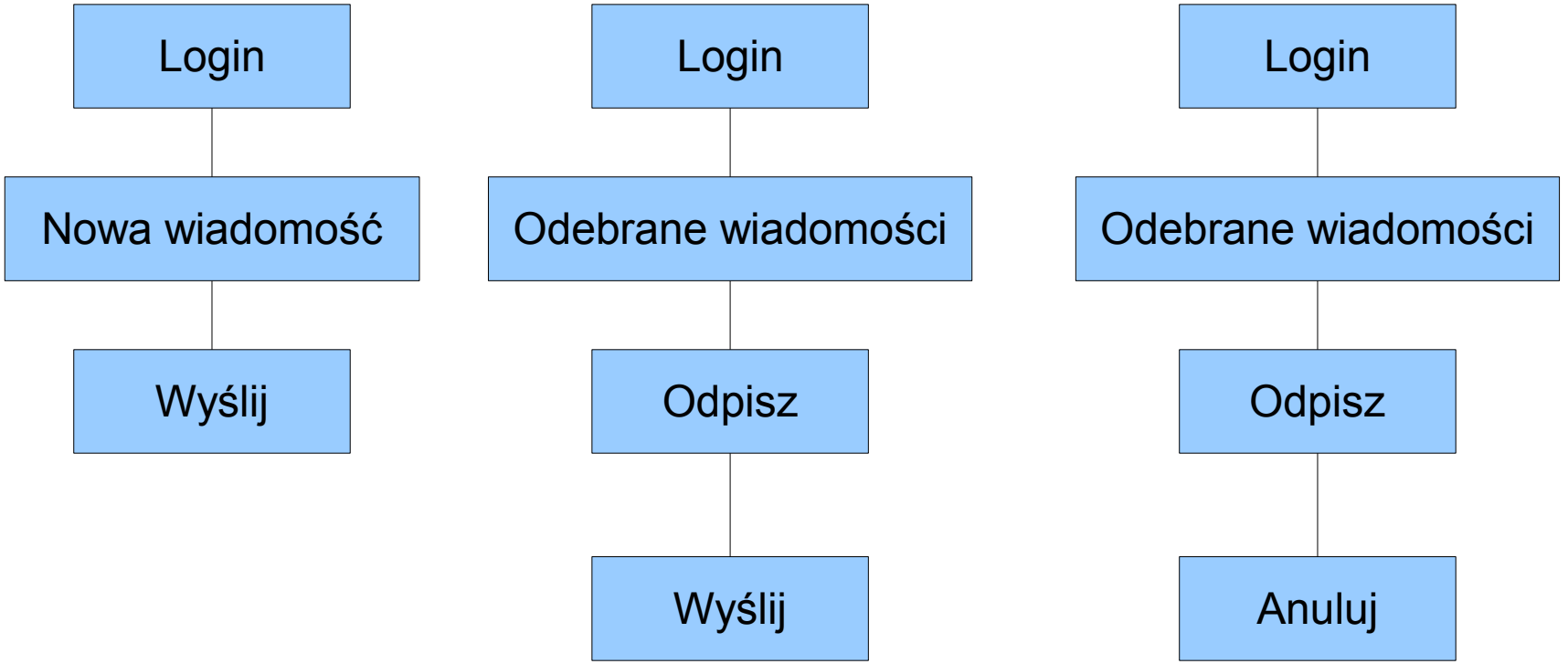
Selenium IDE



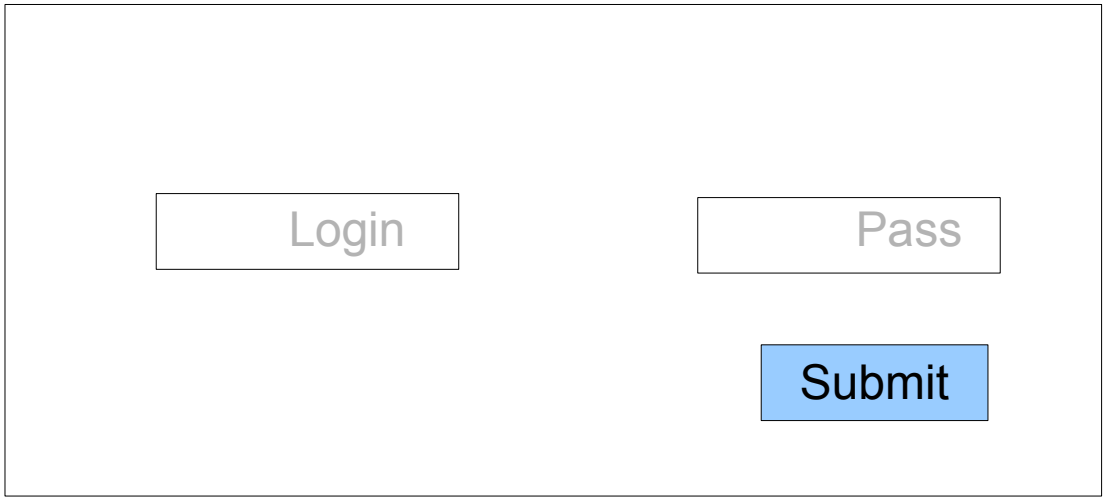
Jaki jest problem z narzędziami typu „Record and Playback” ?

Przykład, testowanie poczty WWW.




Login
Nowa wiadomość
Wyślij
Login
Odebrane wiadomości
Odpisz
Wyślij
Login
Odebrane wiadomości
Odpisz
Anuluj

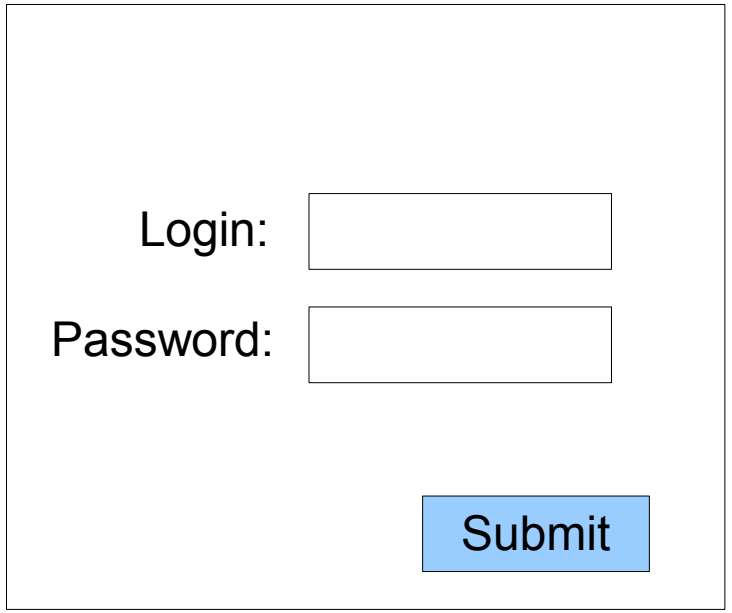
Co jeśli zmieni się wygląd strony logowania?

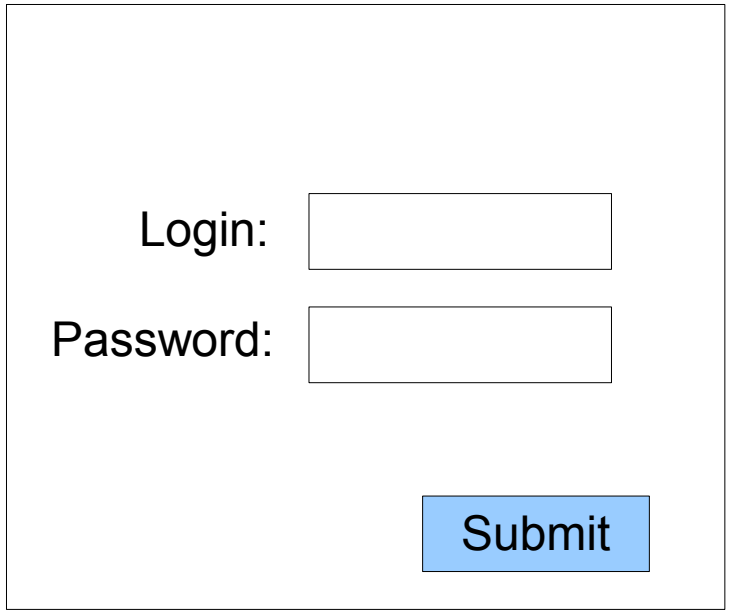


Login 2.0

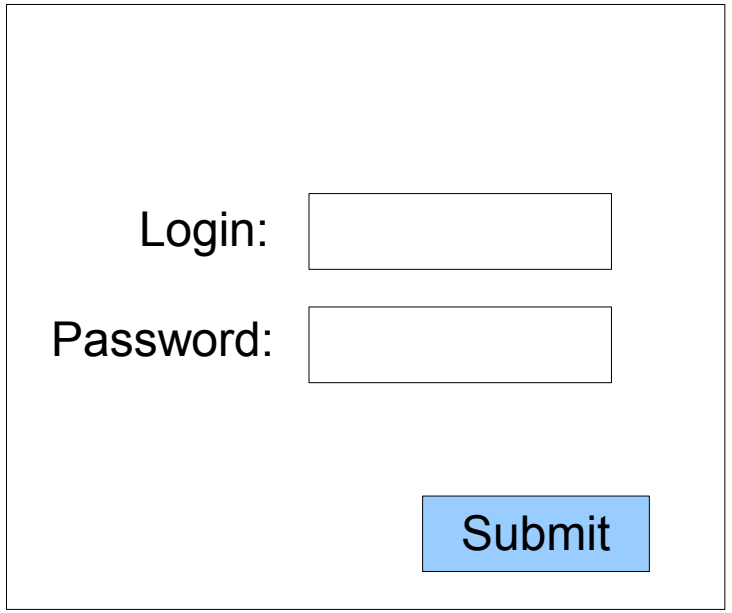
<UI Map Pattern>

- poprawa czytelnosci testów
- usunięcie redundancji elementów GUI

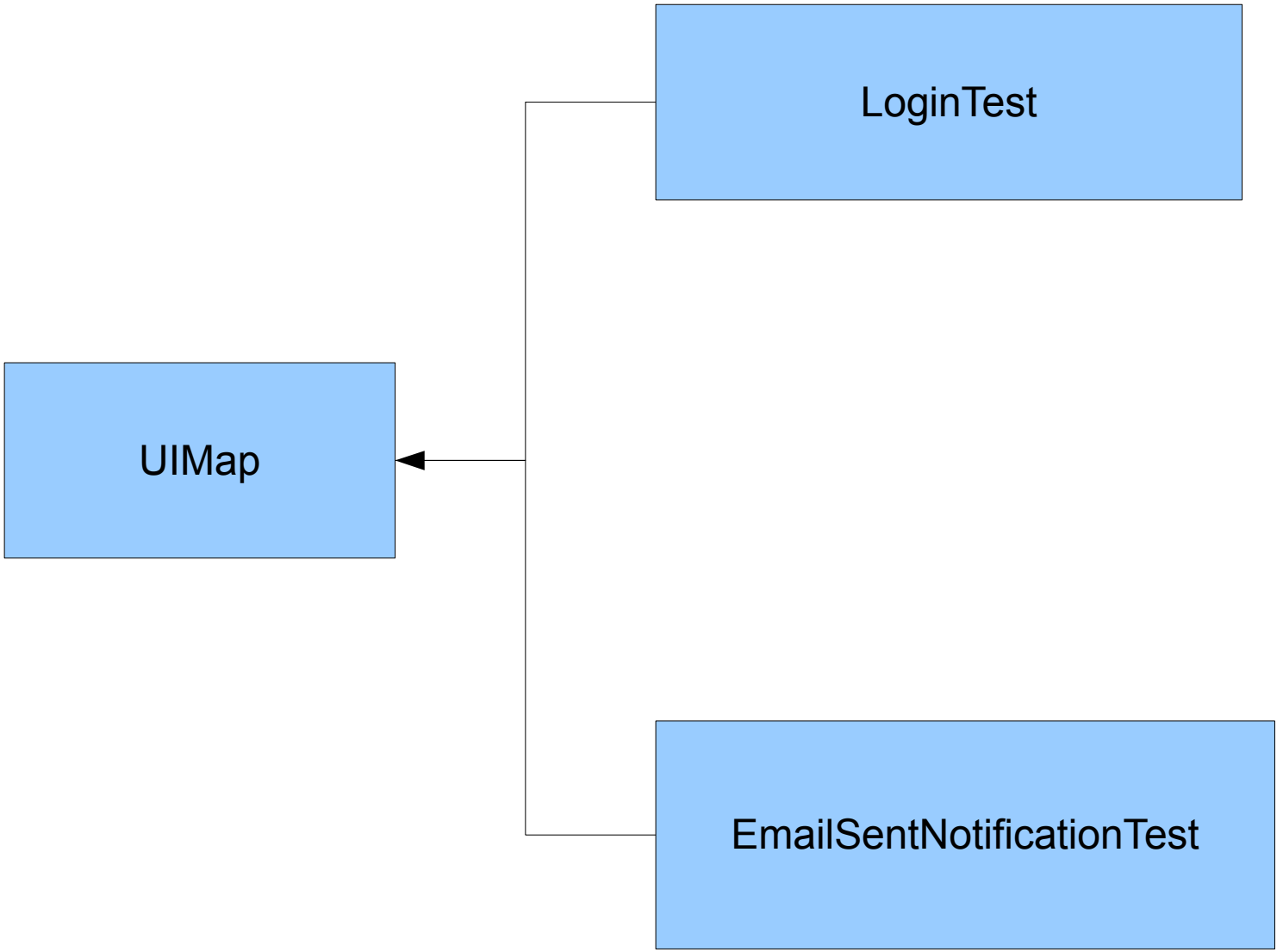

Login:
Password:
Submit



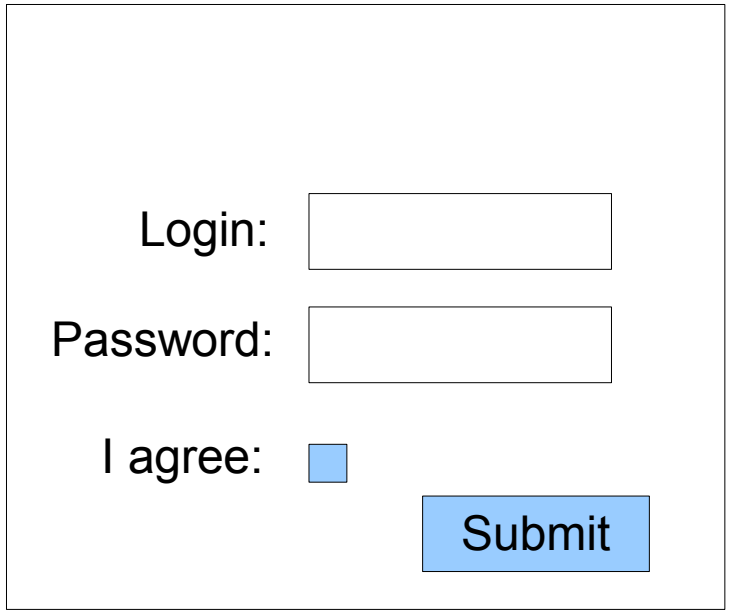

```
public class Test {
    public void Test(){
        seleniumDriver.getElement(By.id("4555")).click();
        sikuliDriver.click("OK.png");
        sikuliDriver.click("OK.png");
    }
}
```




```
public class Test {

    private final String loginFieldId = "4555";
    private final String sumbitButtonImage = "OK.png";

    public void Test(){
        seleniumDriver.getElement(By.id(loginFieldId)).click();
        sikuliDriver.click(sumbitButtonImage);
        sikuliDriver.click(sumbitButtonImage);
    }
}
```


LoginTest
UIMap
EmailSentNotificationTest

Co jeśli zmieni się zachowanie ?

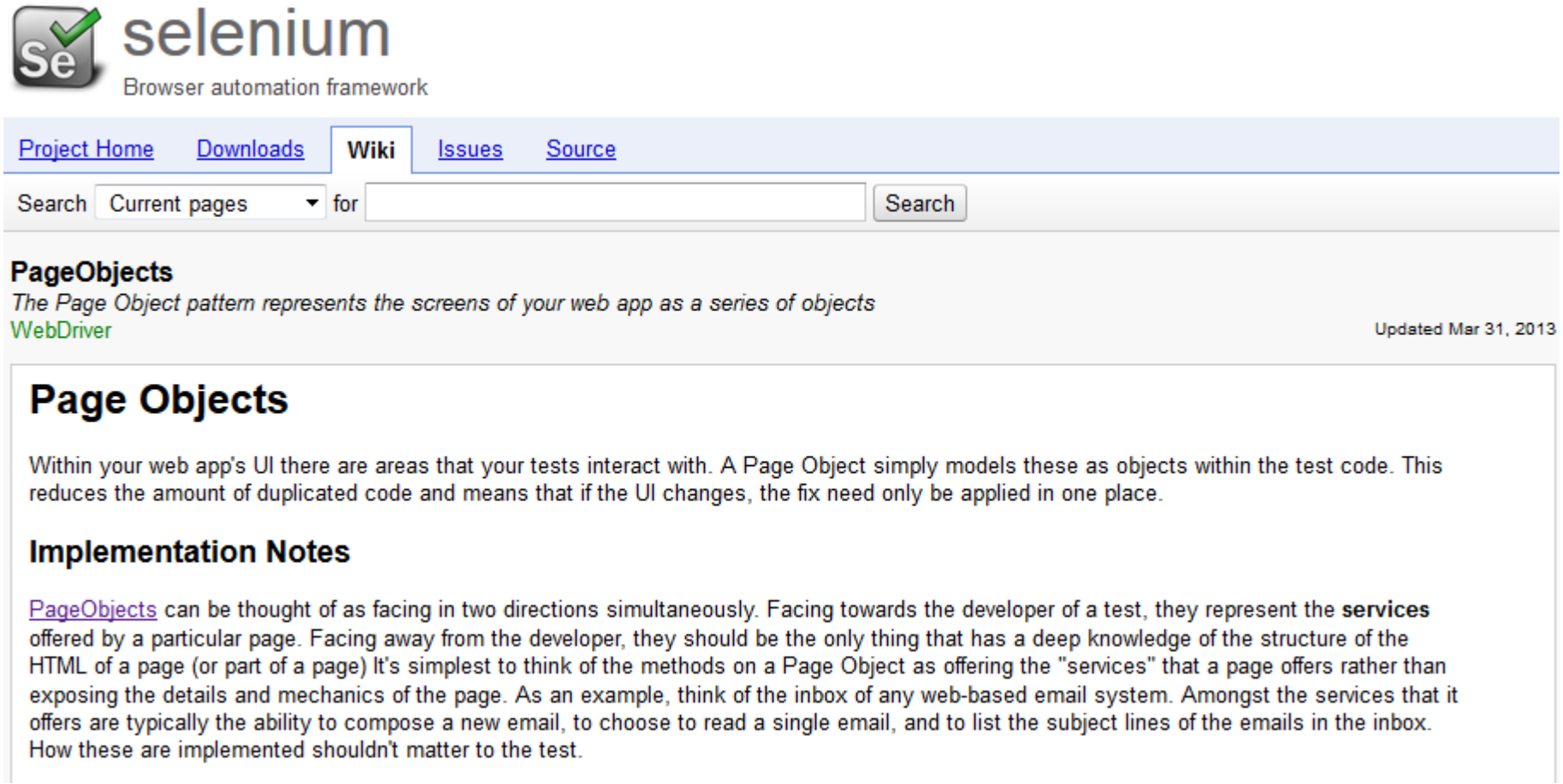

Login:
Password:
I agree:
Submit

<Page Objects Pattern>

<Page Objects Pattern>

http://code.google.com/p/selenium/wiki/PageObjects

<<http://code.google.com/p/selenium/wiki/PageObjects>>



selenium

Browser automation framework

Project Home | Downloads | Wiki | Issues | Source

Search Current pages for Search

PageObjects

The Page Object pattern represents the screens of your web app as a series of objects

WebDriver

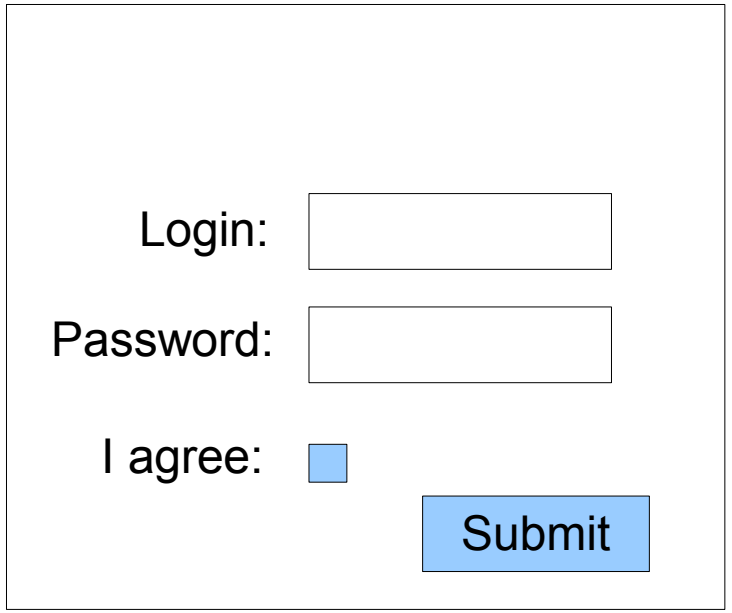
Updated Mar 31, 2013

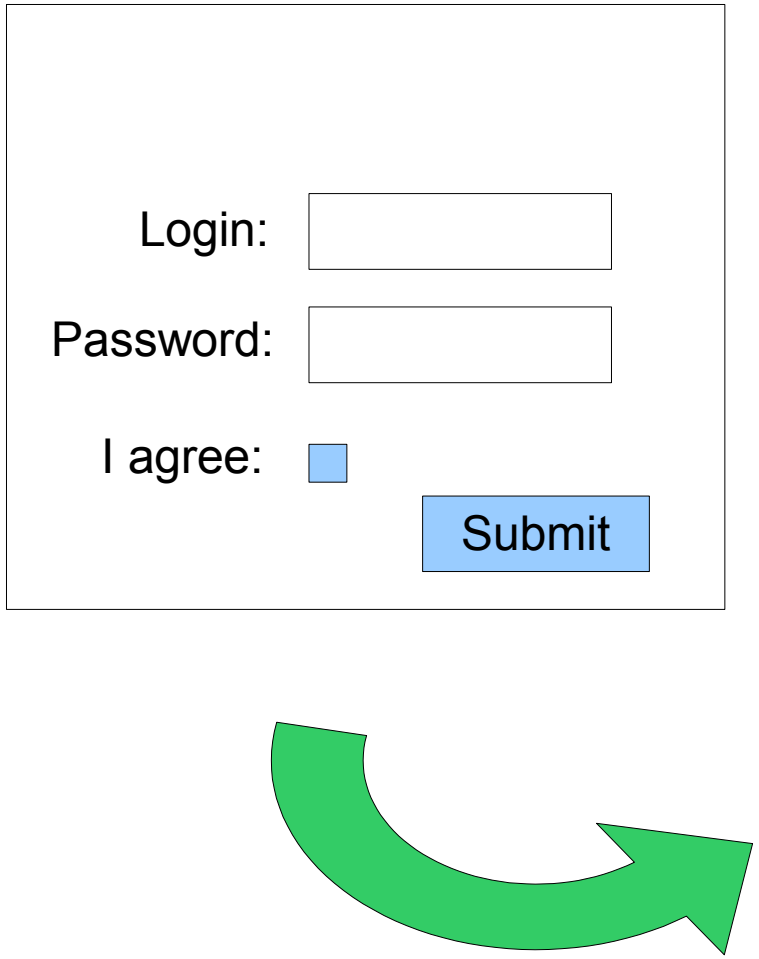
Page Objects

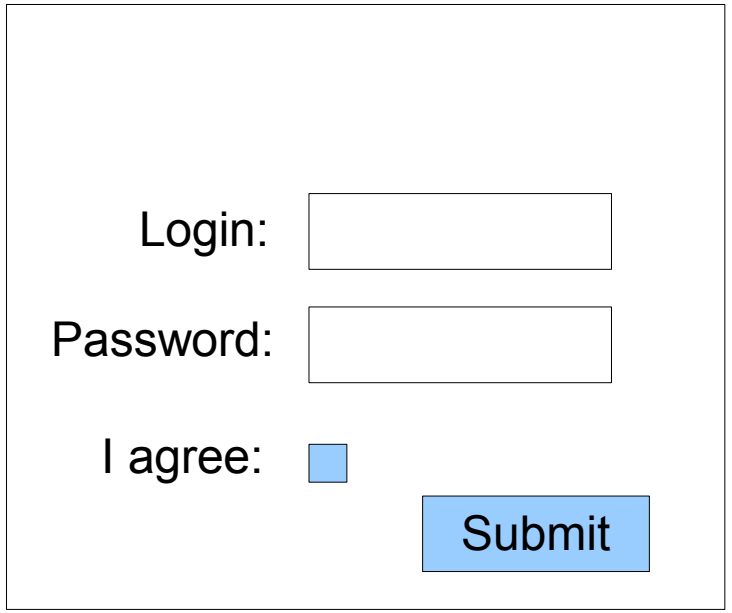
Within your web app's UI there are areas that your tests interact with. A Page Object simply models these as objects within the test code. This reduces the amount of duplicated code and means that if the UI changes, the fix need only be applied in one place.

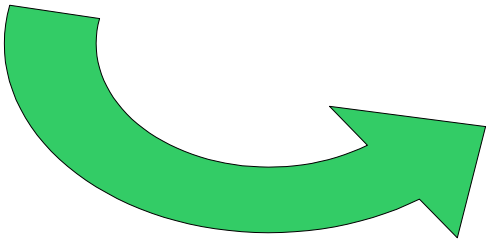
Implementation Notes

PageObjects can be thought of as facing in two directions simultaneously. Facing towards the developer of a test, they represent the **services** offered by a particular page. Facing away from the developer, they should be the only thing that has a deep knowledge of the structure of the HTML of a page (or part of a page) It's simplest to think of the methods on a Page Object as offering the "services" that a page offers rather than exposing the details and mechanics of the page. As an example, think of the inbox of any web-based email system. Amongst the services that it offers are typically the ability to compose a new email, to choose to read a single email, and to list the subject lines of the emails in the inbox. How these are implemented shouldn't matter to the test.

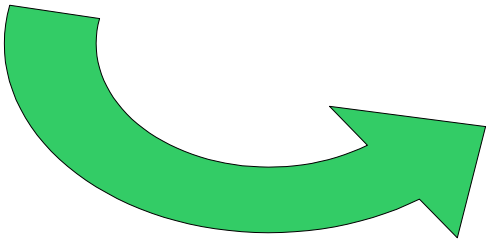

Login:
Password:
I agree:
Submit


Login:
Password:
I agree:
Submit



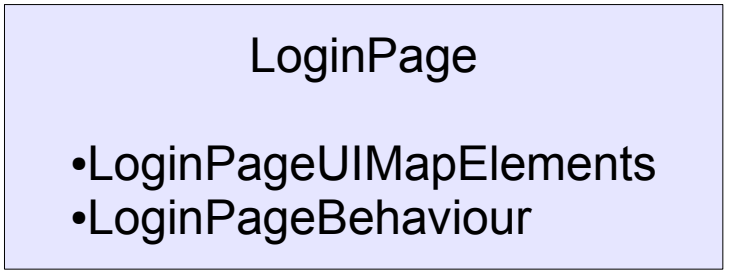


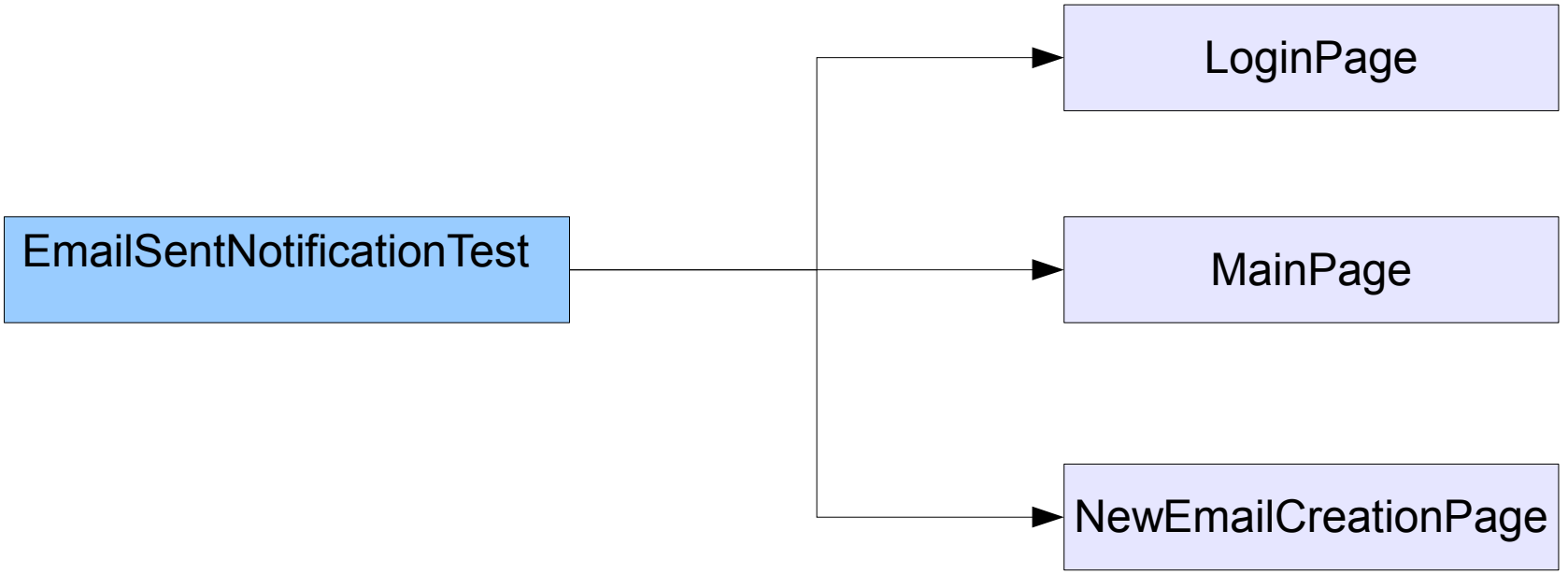
```
public class LoginPage {
    public void performLogin(String userName, String Password){
        //....//
    }

    public void performLoginWithCorrectCredentials(){
        //...//
    }

    public void performLoginWithIncorrectCredentials(){
        //...//
    }
}
```



```java
public class LoginPage {
    public void performLogin(String userName, String Password){
        //....//
    }

    public void performLoginWithCorrectCredentials(){
        //...//
    }

    public void performLoginWithIncorrectCredentials(){
        //...//
    }
}
```

```
public class EmailSentNotificationTest {

    public void shallShowNotificationWhenEmailSent(){

        LoginPage.performLoginWithCorrectCredentials();
        MainPage.createNewEmail();
        NewEmailCreationPage.fillCorrectReceiverEmailAddress();
        NewEmailCreationPage.fillSubjectField ("Sample e-mail subject.");
        NewEmailCreationPage.fillEmailBody ("Sample e-mail body text.");

        //Actual assertion
        Assert.true(driver.getNotificationText().equals("e-mail hes been sent.") );

    }

}
```


LoginPage
•LoginPageUIMapElements
•LoginPageBehaviour


EmailSentNotificationTest
LoginPage
MainPage
NewEmailCreationPage


JIRA


Test
Link

User stories: opis i kryteria akceptacji

User stories: opis i kryteria akceptacji

Dokumentacja testów, raporty

DSL – Domain Specific Language

```
Narrative:
In order to communicate effectively to the business some functionality
As a development team
I want to use Behaviour-Driven Development

Scenario:  A scenario is a collection of executable steps of different type

Given step represents a precondition to an event
When step represents the occurrence of the event
Then step represents the outcome of the event

Scenario:  Another scenario exploring different combination of events

Given a [precondition]
When a negative event occurs
Then a the outcome should [be-captured]

Examples:
|precondition|be-captured|
|abc|be captured    |
|xyz|not be captured|
```

User Story description

```
Narrative:
In order to communicate effectively to the business some functionality
As a development team
I want to use Behaviour-Driven Development
```

```
Scenario:  A scenario is a collection of executable steps of different type

Given step represents a precondition to an event
When step represents the occurrence of the event
Then step represents the outcome of the event

Scenario:  Another scenario exploring different combination of events

Given a [precondition]
When a negative event occurs
Then a the outcome should [be-captured]

Examples:
|precondition|be-captured|
|abc|be captured    |
|xyz|not be captured|
```

```
Narrative:
In order to communicate effectively to the business some functionality
As a development team
I want to use Behaviour-Driven Development
```

Acceptance Criteria

```
Scenario:  A scenario is a collection of executable steps of different type

Given step represents a precondition to an event
When step represents the occurrence of the event
Then step represents the outcome of the event

Scenario:  Another scenario exploring different combination of events

Given a [precondition]
When a negative event occurs
Then a the outcome should [be-captured]

Examples:
|precondition|be-captured|
|abc|be captured    |
|xyz|not be captured|
```


jbehave


Cucumber

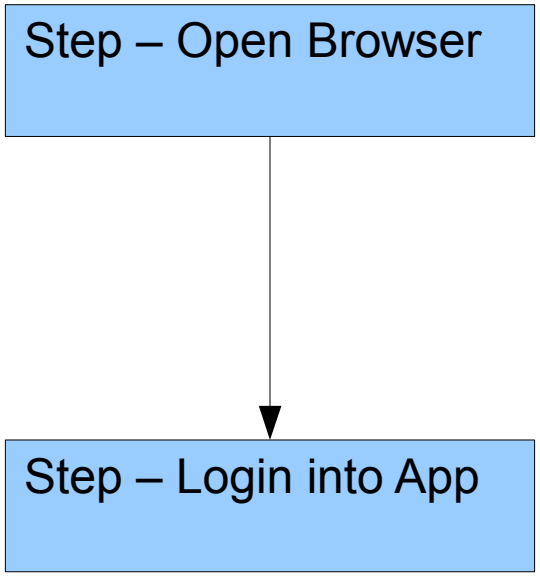
Zobaczmy to na przykładzie...

```java
public class TsmServiceOrderWorklistStep extends SikuliStep {


    public void createNewOrder()
    {
        driver.waitClick("TsmServiceOrderWorklistAddButton.png");

        //service ordering context
        driver.waitFor("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.clickOffset("TsmServiceOrderingPopupFindServiceLabel.png", new Location(0,-24));

        //Temperature check order
        driver.paste("A0001");
        driver.waitVanish("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.keyHit(Key.ENTER);
```

```
public class TsmServiceOrderWorklistStep extends SikuliStep {

    @LeadsToStepId("TSMSpecifyServiceTemperatureCheckProcess - ProcessLastStep")
    public void createNewOrder()
    {
        driver.waitClick("TsmServiceOrderWorklistAddButton.png");

        //service ordering context
        driver.waitFor("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.clickOffset("TsmServiceOrderingPopupFindServiceLabel.png", new Location(0,-24));

        //Temperature check order
        driver.paste("A0001");
        driver.waitVanish("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.keyHit(Key.ENTER);
```

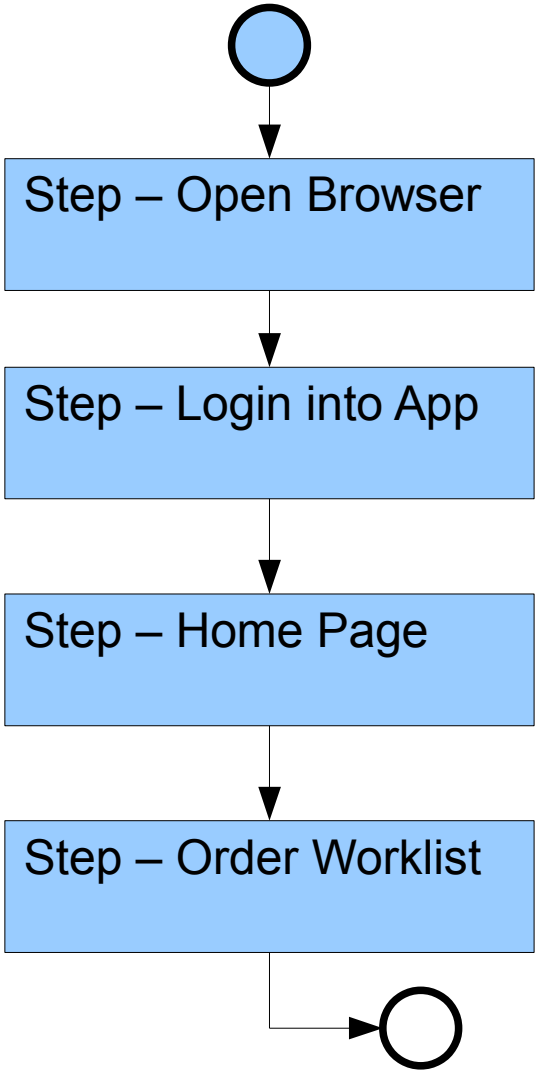
Step

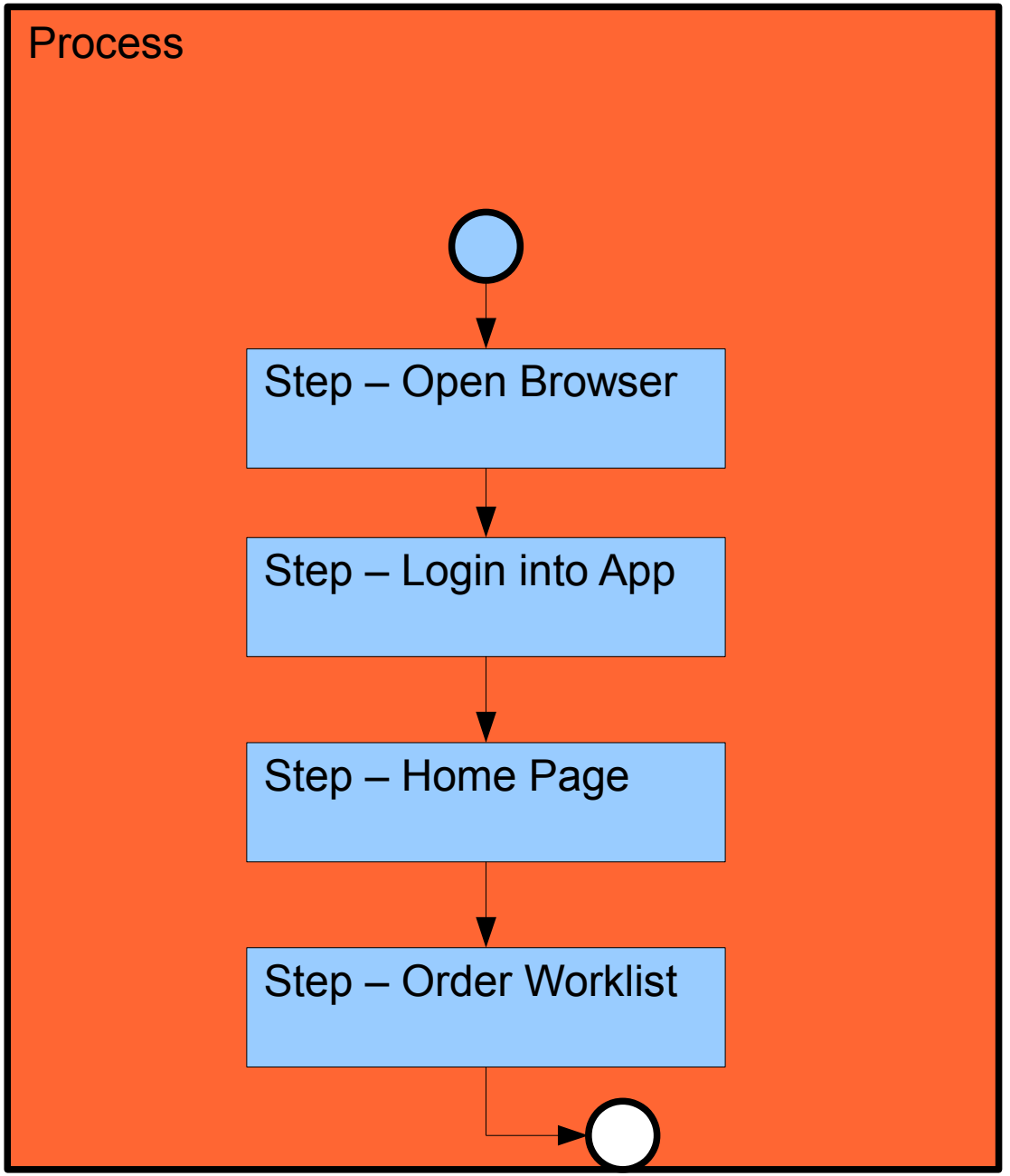
```
public class TsmServiceOrderWorklistStep extends SikuliStep {


    public void createNewOrder()
    {
        driver.waitClick("TsmServiceOrderWorklistAddButton.png");

        //service ordering context
        driver.waitFor("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.clickOffset("TsmServiceOrderingPopupFindServiceLabel.png", new Location(0,-24));

        //Temperature check order
        driver.paste("A0001");
        driver.waitVanish("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.keyHit(Key.ENTER);
```

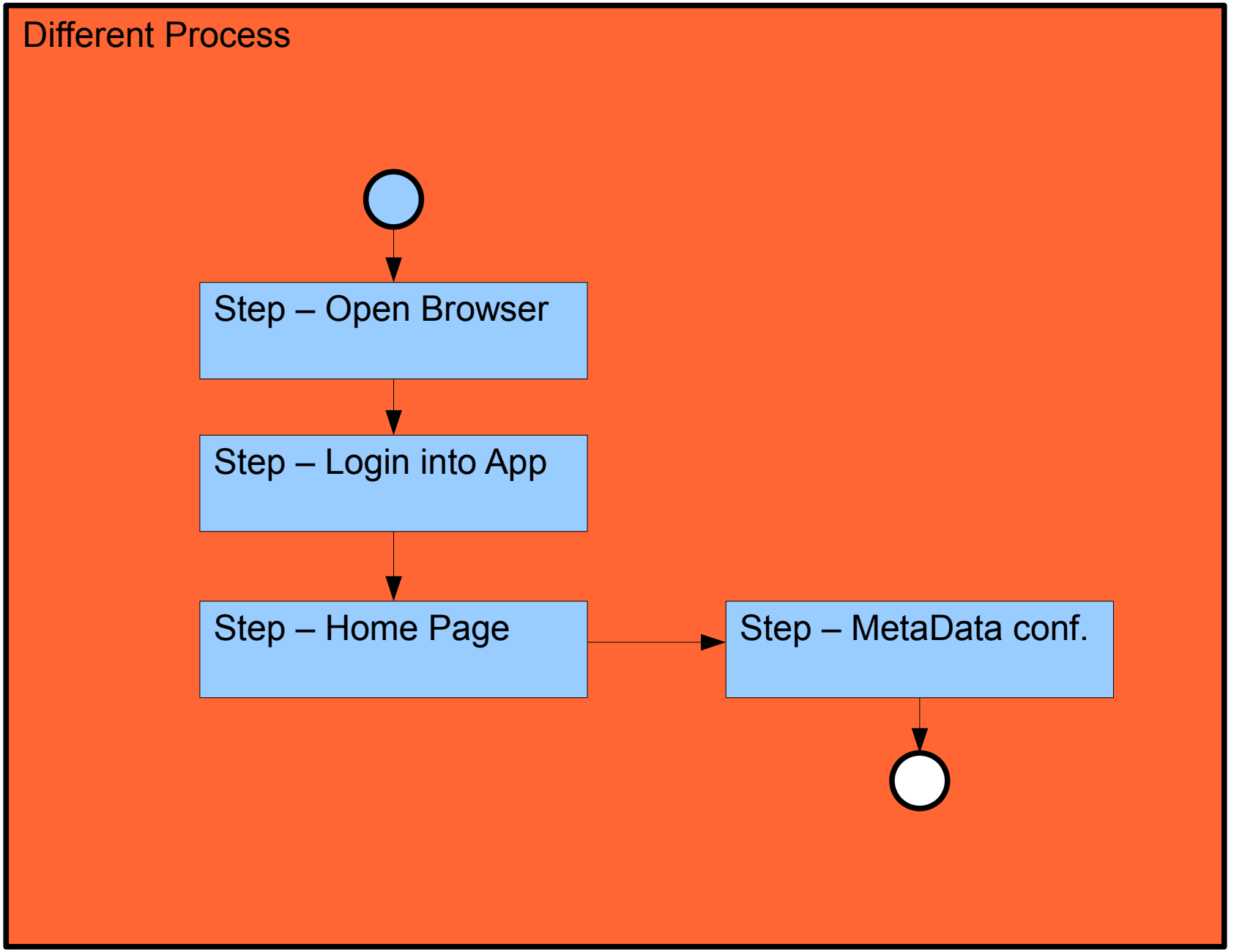

Step – Open Browser
Step – Login into App

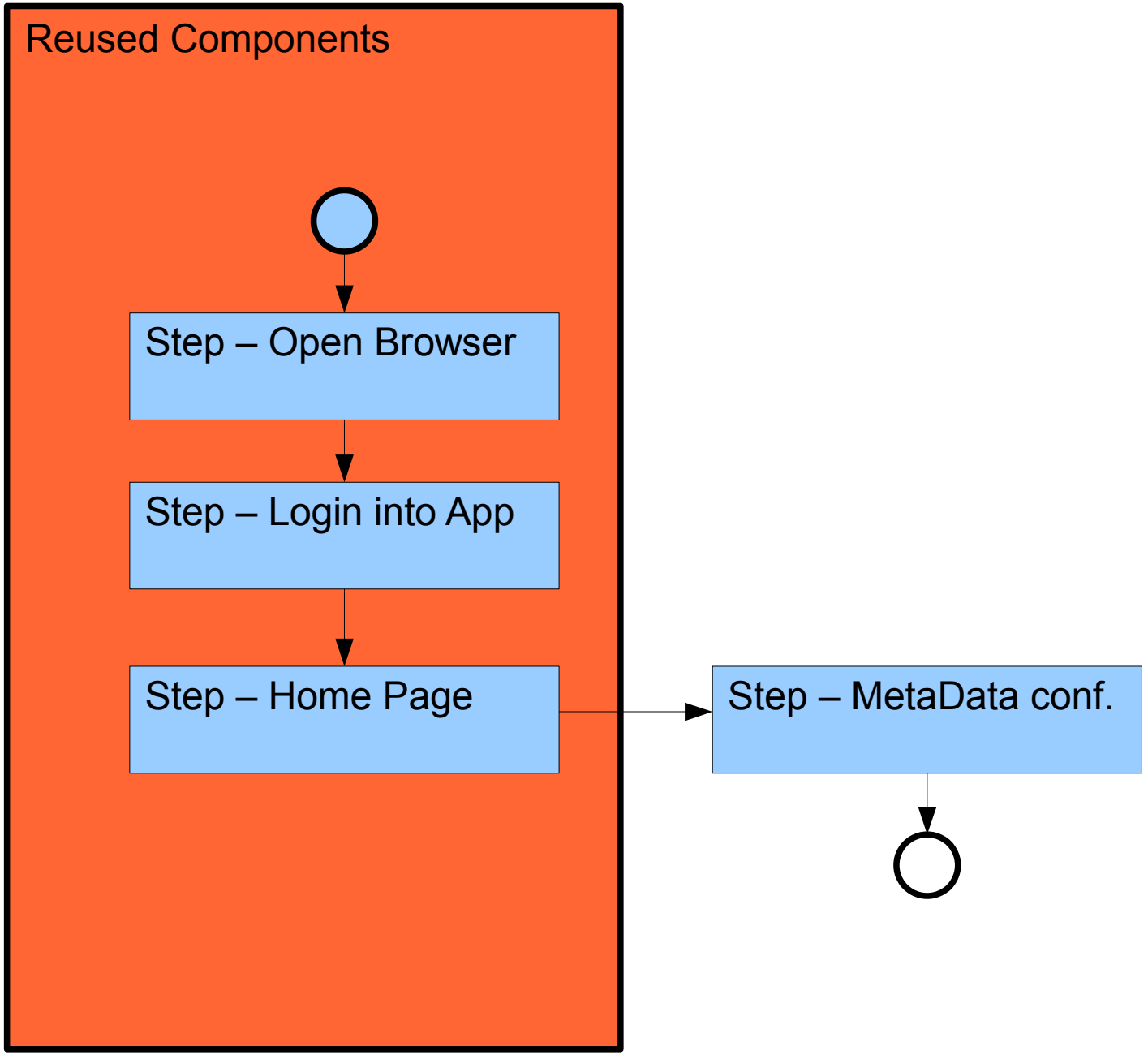

Step – Open Browser
@LeadsToStepId("TSMSpecifyServiceTemperatureCheckProcess - ProcessLastStep")
Step – Login into App

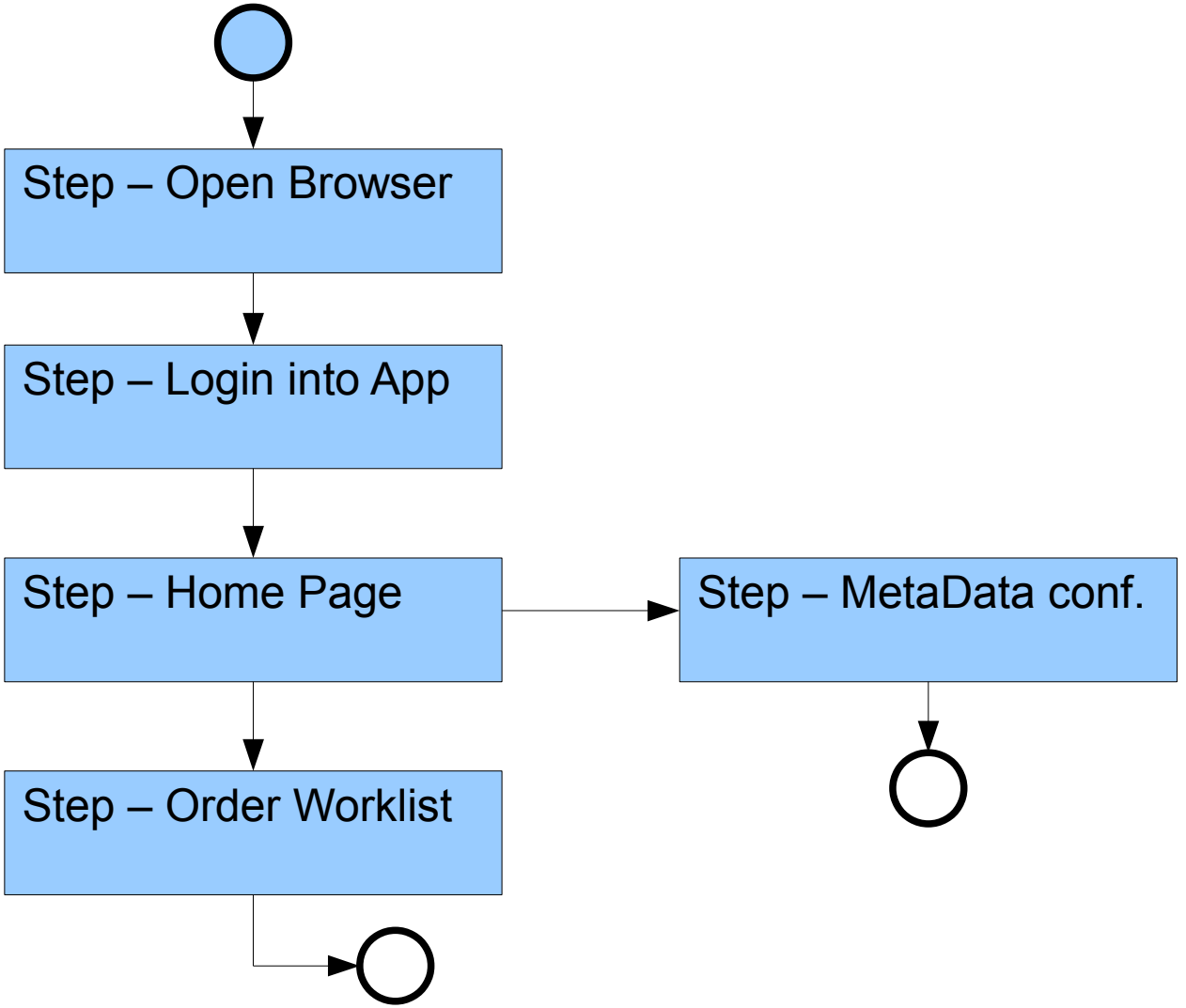
```java
public class TsmServiceOrderWorklistStep extends SikuliStep {


    public void createNewOrder()
    {
        driver.waitClick("TsmServiceOrderWorklistAddButton.png");

        //service ordering context
        driver.waitFor("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.clickOffset("TsmServiceOrderingPopupFindServiceLabel.png", new Location(0,-24));

        //Temperature check order
        driver.paste("A0001");
        driver.waitVanish("TsmServiceOrderingPopupFindServiceLabel.png");
        driver.keyHit(Key.ENTER);
```

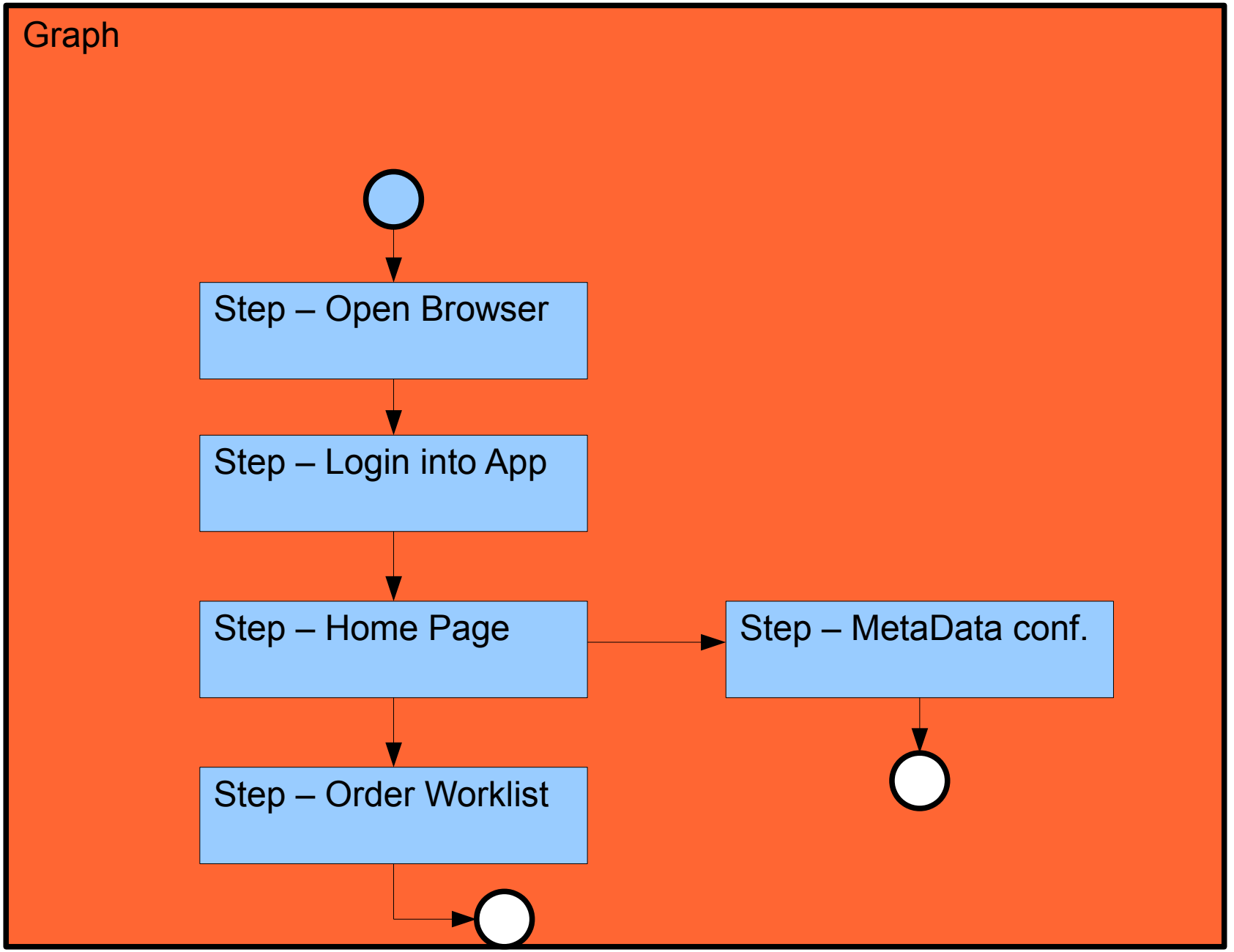

Step – Open Browser
Step – Login into App
Step – Home Page
Step – Order Worklist


Process
Step – Open Browser
Step – Login into App
Step – Home Page
Step – Order Worklist


Different Process
Step – Open Browser
Step – Login into App
Step – Home Page
Step – MetaData conf.


Reused Components
Step – Open Browser
Step – Login into App
Step – Home Page
Step – MetaData conf.

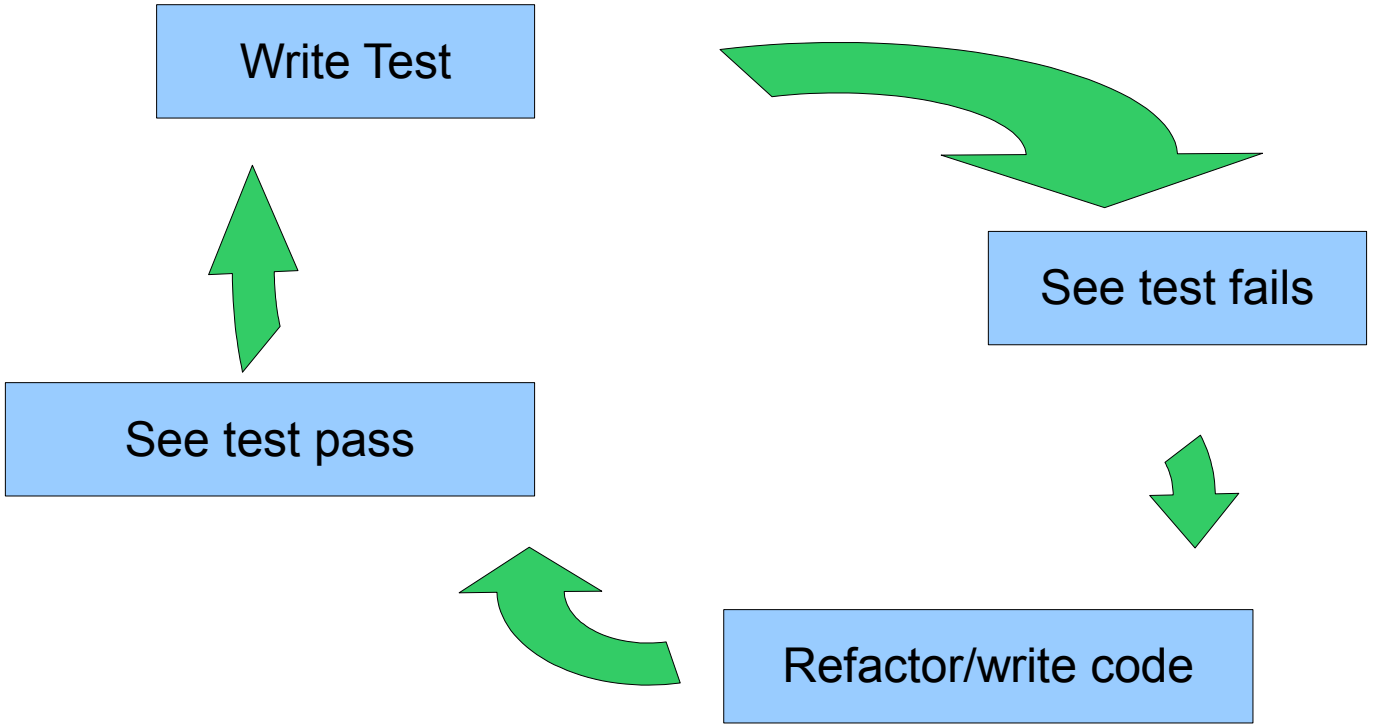

Step – Open Browser
Step – Login into App
Step – Home Page
Step – MetaData conf.
Step – Order Worklist


Graph
Step – Open Browser
Step – Login into App
Step – Home Page
Step – MetaData conf.
Step – Order Worklist

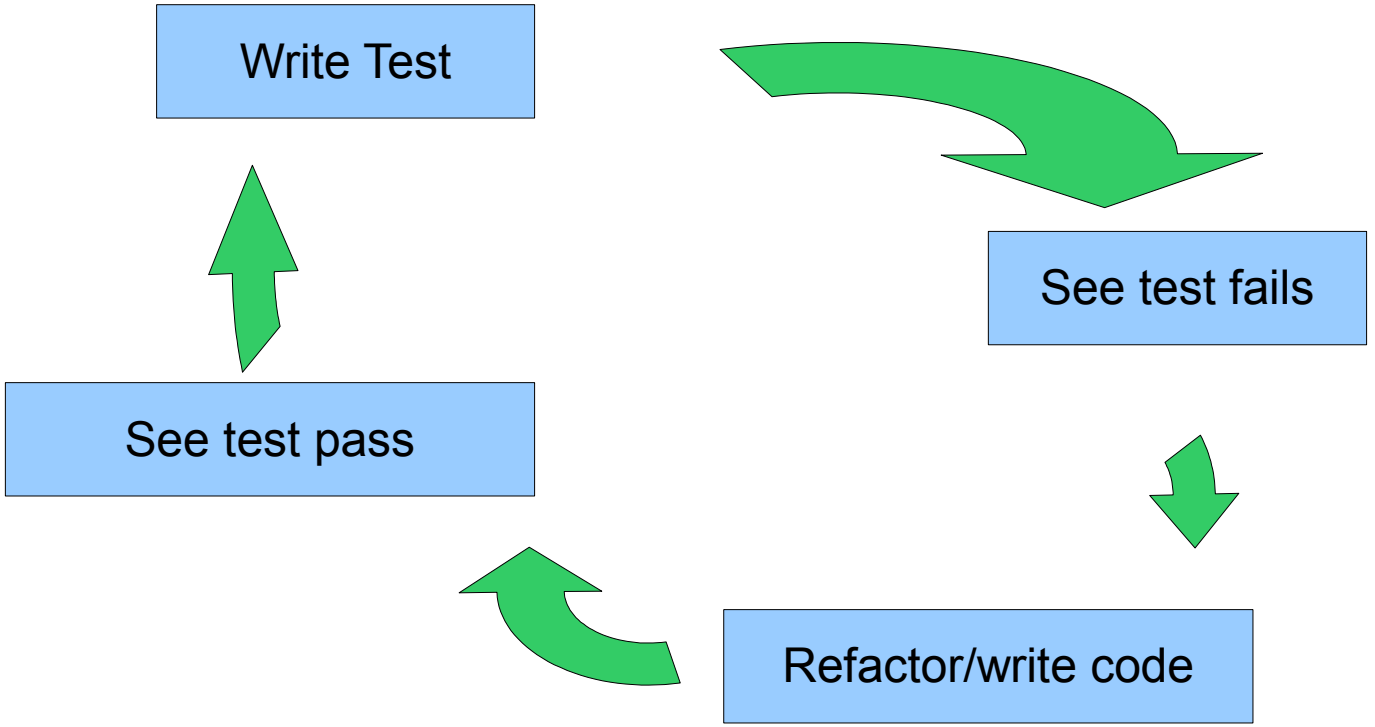
BDD...

BDD is TDD done really well...

TDD



BDD



Sikuli

Wykorzystanie:
- testy funkcjonalne na poziomie integracyjnym i systemowym
- testy niefunkcjonalne (pamięć)

Zalety:
- szybkość i łatwość pisania testów
- możliwość przetestowania większości rodzajów GUI

Wady:
- szybkość działania
- mała odporność na gruntowne zmiany GUI (np. zmiana czcionki)
- brak wsparcia dla i18n

Dziękuję za uwagę...