

Evaluation – GUI Testing

Projekt: DigitalSchoolNotes

Projekt Team: Adler, Brinnich, Hohenwarter, Karic, Stedronsky

Version 1.0

07.10.2015

Status: [RELEASE]

	Datum	Name	Unterschrift
Erstellt	06.10.2015	Niklas Hohenwarter	
Geprüft	07.10.2015	Selina Brinnich	
Freigegeben			
Git-Pfad: <i>/doc/evaluation</i> Dokument: <i>evaluation_gui_testing.doc</i>			

1 Changelog

Version	Datum	Status	Bearbeiter	Kommentar
0.1	2015-10-07	Erstellt	Selina Brinnich	Dokument erstellt
1.0	2015-10-07	Geprüft	Niklas Hohenwarter	QA
1.1	2015-10-12	Bearbeitet	Thomas Stedronsky	Rechtschreibfehler, Code Formatierung und Prototypen Fehlermeldungen

2 Vergleich

	Sahi OS	Watir	Robot Framework	Selenium
Dokumentation	7/10 Gut dokumentiert, leicht unübersichtlich	4/10 Schwer zu finden, unübersichtlich	4/10 Viel Dokumentation, keine Schritt für Schritt Beispiele	7/10 Gut dokumentiert, leicht unübersichtlich
Lizenz	5/10 Nicht alle Features frei verfügbar	10/10 BSD License	10/10 Apache 2 License	10/10 Apache 2 License
Installation / Konfiguration	10/10 Client Installation via JAR	10/10 Via Ruby Packet Manager	10/10 Via Python Packet Manager	8/10 Server als JAR Client als Firefox Plugin
Community	7/10 180 Fragen auf Stackoverflow	9/10 2.600 Fragen auf Stackoverflow	8/10 927 Fragen auf Stackoverflow	10/10 32.590 Fragen auf Stackoverflow
Benutzerfreundlichkeit	10/10 Testerstellung über GUI Tool und per Script	5/10 Nur Programmierung via Ruby	8/10 Testerstellung via eigener Scriptsprache	10/10 Testerstellung über GUI Tool und per Script
Multi Browser Integration	8/10 Firefox, Chrome, IE	8/10 Firefox, Chrome, IE	10/10 Firefox, Chrome, IE, Opera, Safari	10/10 Firefox, Chrome, IE, Opera, Safari
Jenkins Integration	10/10 Vorhanden und gut dokumentiert	8/10 3 rd Party Product	10/10 Via Plugin, gut dokumentiert	8/10 Testergebnisse nur per Konsole
Prototyp	0/10 Funktioniert nicht	0/10 Funktioniert nicht	0/10 Funktioniert nicht	10/10 Funktionierte sofort nach 5 Minuten
Summe	57/80	54/80	60/80	73/80

3 *Sahi*

3.1 Installation

Installation von Sahi[1]:

- Download des Installers
- Execute Installer
- In den bin Ordner gehen und dort sahi.sh starten (Port 9999)

3.2 Konfiguration

- 'Go to Tools > Options > General > Connection Settings >
- Set to "Manual Proxy Configuration"
- Set "HTTP Proxy" to "localhost"
- Set "Port" to "9999". (This setting can be modified through `<sahi_root>/config/sahi.properties`)
- Do the same for "SSL Proxy" too, if you wish to record and playback https sites
- Keep "Use the same proxy for all protocol" unchecked as Sahi does not understand protocols other than HTTP
- NOTE: "No Proxy for" should NOT have localhost in it.' [2]

3.3 Prototyp

Konsole startet nicht, keine Lösung gefunden

4 *Watir*

4.1 Installation

Installation von Watir[3]:

```
sudo apt-get install ruby ruby-dev
sudo apt-get install rubygems
gem update --system --no-rdoc --no-ri
gem install watir --no-rdoc --no-ri
```

4.2 Prototyp

Rubygem Watir-Web-Framework kann nicht installiert werden, wird allerdings benötigt um Tests in einem anderen Browser als dem Internet Explorer auszuführen.

5 Robot Framework

5.1 Installation

Installation von Robot Framework[4]:

```
sudo apt-get install python-pip
sudo pip install robotframework
sudo pip install docutils
```

5.2 Prototyp

Test kann nicht ausgeführt werden, obwohl die Installationsanleitung befolgt wurde können bestimmte Librarys nicht gefunden werden.

6 Selenium

6.1 Installation

Installieren der Library in Eclipse um Tests in Java zu schreiben[5]:

- Installation von Eclipse
- Herunterladen der Selenium Libs
- Hinzufügen der Libs zum Projekt

6.2 Prototyp

Tests werden an folgender Seite durchgeführt:

Untitled Form

This is your form description. Click here to edit.

Text

Checkboxes

- ☐ First option
- ☐ Second option
- ☐ Third option

Submit

Der Test läuft nach folgendem Code ab[6]:

```
package eval;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.firefox.FirefoxDriver;

public class EvalSelenium {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new FirefoxDriver();

        driver.get("file:///home/niklas/Desktop/form/form.html");

        WebElement textfield = driver.findElement(By.id("element_1"));
        textfield.sendKeys("Test");

        WebElement check = driver.findElement(By.id("element_2_2"));
        check.click();

        Thread.sleep(5000);

        driver.quit();
    }
}
```

7 Endergebnis

Da Selenium als einziges Produkt innerhalb kürzester Zeit funktioniert hat, entscheidet sich das Team für **Selenium**. Desweiteren hat es eine sehr große Community und lässt sich gut komplett automatisieren und in CI Tools integriert werden.

8 Quellen

[1], Narayan Raman, "Installation",
<http://sahi.sourceforge.net/install.html#install>, zuletzt besucht: 05.10.2015

[2], Narayan Raman, "Getting started",
<http://sahi.sourceforge.net/using.html>, zuletzt besucht: 05.10.2015

[3], Watir, "Installation",
<http://watir.com/installation/>, zuletzt besucht: 05.10.2015

[4], Robotframework, "User Guide",
<http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>, zuletzt besucht: 06.10.2015

[5], Guru99, "Guide to install Selenium WebDriver",
<http://www.guru99.com/installing-selenium-webdriver.html>, zuletzt besucht: 06.10.2015

[6], Google Code, "The 5 Minute Getting Started Guide",
<https://code.google.com/p/selenium/wiki/GettingStarted>, zuletzt besucht: 06.10.2015