

## Porównanie wydajności testów automatycznych napisanych w technologii SeleniumWebDriver i HP UFT

Krzysztof Drążek\*, Maria Skublewska-Paszkowska

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

**Streszczenie.** Testy automatyczne stosowane są do weryfikowania funkcjonalności tworzonego oprogramowania. W artykule porównana została szybkość wykonywania automatycznych testów napisanych w popularnych technologiach: HP UFT i WebDriver Selenium. Dla każdej z nich powstały po 3 testy, wykonujące te same czynności. Na podstawie otrzymanych wyników zostały wyciągnięte wnioski dotyczące tego, która technologia jest lepszym wyborem, gdy liczy się wydajność wykonywanych testów.

**Słowa kluczowe:** testy automatyczne; testowanie; Selenium; UFT

\*Autor do korespondencji.

Adres e-mail: krzydrazek@gmail.com

## Performance comparison of automatic tests written in Selenium WebDriver and HP UFT

Krzysztof Drążek\*, Maria Skublewska-Paszkowska

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

**Abstract.** Automation tests are used for verification of functionality of developing applications. The performance of automation tests written in HP UFT and Selenium WebDriver was compared. There were created 3 tests for both technologies that were executing the same tasks. According to the obtained results, it was concluded which technology is a better choice if the time efficiency of tests execution is important.

**Keywords:** automation tests; testing; Selenium; UFT

\*Corresponding author.

E-mail address: krzydrazek@gmail.com

### 1. Wstęp

Testy automatyczne są jedną z gałęzi działu testów w procesie tworzenia oprogramowania. Ich głównym założeniem jest zmniejszenie nakładu pracy, którą trzeba wykonać by przetestować tworzoną aplikację [6]. Automatyzowane powinny być te testy, które są powtarzalne, a testy wykonywane są regularnie. Daje to pewność, że testy automatyczne spełnią swoją rolę.

Wiedząc, że automaty są potrzebne w projekcie, należy podjąć decyzję o tym, z jakiego narzędzia należy skorzystać. Może to zaważyć na tym, czy projekt automatyzacji odniesie sukces, czy też okaże się porażką [3]. Pewien wpływ na to ma wydajność testów, które są pisane. Jeśli się okaże, że są one niewiele szybsze od testów manualnych, to wysiłek i koszt poniesione do ich napisania okaże się zmarnowany. Dlatego w artykule został podjęty temat wydajności testów pisanych w dwóch popularnych technologiach dostępnych na rynku: Selenium i UFT. Przedstawione wyniki mają na celu odpowiedzi na pytanie, która z przedstawionych technologii jest wydajniejsza, czyli w rozpatrywanym przypadku szybsza.

### 2. Przegląd literatury

Porównanie Selenium i UFT było już poruszane w literaturze naukowej. Między innymi Prof. Dashrath Mane i Prachi Kunte zajmowali się tym zagadnieniem [2]. Porównywali obie technologie na tle wspieranych systemów i przeglądarek, wydajności sprzętowej i szybkości, z jaką testy działają.

Również Yogesh Kumar porównuje między innymi Selenium i UFT [1]. Na podstawie różnych parametrów ocenia, która technologia jest najlepsza. Parametry brane pod uwagę to: możliwość nagrywania, możliwość generowania skryptów, testowanie oparte na danych, wygląd raportów, ponowne użycie skryptów, szybkość wykonywania testów, łatwość w nauczaniu się danej technologii, koszt.

Dodatkowo Harsh Bajaj pisał na temat wyboru narzędzi do automatyzacji do projektu [3]. Szukając najlepszej technologii brał pod uwagę następujące czynniki: łatwość zaadoptowania technologii do projektu, łatwość tworzenia skryptów i możliwości raportowania, techniczne możliwości technologii (wspierane systemy, przeglądarki itp.).

### 3. Materiały i metody

W celu sprawdzenia wydajności wybranych narzędzi należało stworzyć z użyciem tych technologii kilka testów. W artykule zostały utworzone 3 przypadki testowe dla testowanej aplikacji, którą była platforma e-learningowa Moodle, z której korzysta Politechnika Lubelska. Na podstawie tych przypadków testowych powstały testy automatyczne, które posłużyły do przetestowania wydajności Selenium i UFT.

#### 3.1. Technologie

##### 3.1.1. Selenium

Selenium to prawdopodobnie najpopularniejszy framework (szkielet aplikacji) do pisania testów automatycznych aplikacji internetowych. Jest całkowicie darmowy, dostępny na licencji open-source. Wspiera wiele różnych języków programowania (Java, C#, Python itp.) [1,7]. Wspiera też wiele narzędzi ciągłej integracji (ang. continuous integration) takich jak Bamboo czy Jenkins [5].

##### 3.1.2. UFT

UFT to narzędzie do automatyzacji stworzone przez firmę HP (Hewlett-Packard), a obecnie rozwijane przez firmę MicroFocus [2]. Umożliwia tworzenie testów automatycznych GUI, API, aplikacji mobilnych, aplikacji desktopowych i innych. UFT jest dostępne dla systemu Windows i w swojej głównej funkcjonalności wykorzystuje język VBScript [4]. Największą zaletą tej technologii jest wbudowane repozytorium obiektów, gdzie przechowywane są definicje obiektów wykorzystywanych w testach [8]. Wystarczy kliknąć w obiekt na stronie, by ten pojawił się w repozytorium. Dzięki temu nie ma potrzeby pisać definicji obiektów w kodzie, tak jak to ma miejsce w Selenium.

#### 3.2. Przypadki testowe

Przypadki testowe powstały, jako lista kroków, które należy wykonać by test zakończył się sukcesem. Każdy krok testu miał opisywać to, co należy wykonać oraz oczekiwany rezultat, który świadczył o tym, że akcja została wykonana prawidłowo.

Pierwszy test polegał przede wszystkim na uzupełnianiu pól formularza. Poza wpisywaniem określonego tekstu do pól tekstowych, test musiał też rozwijać zakładki, w których te pola się znajdowały. Test został przedstawiony w Tabeli 1.

Uzupełnianie formularza to jedno z podstawowych zadań testów automatycznych, w czym sprawdzają się bardzo dobrze. Test automatyczny robi to o wiele szybciej od człowieka. Takie testy nadają się doskonale do automatyzacji.

Drugi test polegał na dodaniu do sprawozdania z laboratorium studenta pliku i dodania komentarza do tego pliku. Następnie Użytkownik wypisywał się z kursu i ponownie się do niego zapisywał. Test został zaprezentowany w Tabeli 2.

Tabela 1. Pierwszy przypadek testowy stworzony na potrzeby badań

Lp.	Opis	Rezultat
1.	Zaloguj się do aplikacji, jako student	Użytkownik zalogowany
2.	Wejść do profilu użytkownika	Otwiera się ekran profilu użytkownika
3.	Uruchom modyfikację profilu użytkownika	Pojawił się formularz do modyfikacji profilu
4.	W zakładce ogólne zmień: imię, nazwisko email i miasto	Dane zostały zmienione
5.	W zakładce zdjęcie użytkownika dodaj zdjęcie	Zdjęcie zostało dodane
6.	W zakładce "Dodatkowe nazwy" uzupełnij pola: imię- fonetyczne, nazwisko- fonetyczne, drugie imię, alternatywna nazwa	Dane zostały zmienione
7.	W zakładce "Zainteresowana" uzupełnij pole: lista zainteresowań	Pole uzupełnione
8.	W zakładce "Opcjonalne" uzupełnij wszystkie pola	Pola uzupełnione
9.	W zakładce "Inne pola" zmień zawartość pól: Grupa dziekańska i rok rozpoczęcia studiów	Pola zmienione
10.	Zapisz zmiany	Zmiany zapisane. Powrót do ekranu profilu
11.	Wyloguj się	Użytkownik wylogowany

Test zawiera, poza standardowymi czynnościami jak klikanie i uzupełnianie pól tekstowych, załadowanie pliku z dysku komputera do sprawozdania.

Trzeci test miał na celu sprawdzenie, czy prawidłowo działa funkcja dodania do aplikacji nowego kursu dla studentów i dodanie do niego pliku, który będzie można pobrać na dysk. Test został przedstawiony w Tabeli 3.

Ponownie test polegał na załadowaniu pliku na serwer, jednak dodatkowo, ten plik jest następnie pobierany i test sprawdza, czy faktycznie plik się pobrał.

#### 3.3. Testy automatyczne

Każdy z przypadków testowych został zaimplementowany w formie testu automatycznego dla obu badanych technologii. Zamierzeniem było, by testy były napisane w sposób zbliżony do siebie. Jeśli w jakimś kroku zostały kliknięte po sobie 3 przyciski, a następnie sprawdzone pojawienie się jakiegoś obiektu na stronie to w obu wersjach testu, dla Selenium i UFT powinny zostać wykonane te same czynności. W tym zakresie cel został osiągnięty, co widać na przykładach 1 i 2. Ten sam fragment testu pokazany w obu technologiach.

Tabela 2. Drugi przypadek testowy stworzony na potrzeby badań

Lp	Opis	Rezultat
1.	Zaloguj się do aplikacji, jako student	Użytkownik zalogowany
2.	Przejdź do zakładki "Kursy"	Pojawia się lista działów
3.	Wyszukaj kurs "Zespołowy Projekt Programistyczny TWO IMNS_MP" i wejdź do niego	Pojawia się zawartość kursu
4.	Pobierz pierwszy plik na dysk	Plik zapisany na dysku
5.	Wejdź do sprawozdania i dodaj plik	Plik dodany do sprawozdania
6.	Dodaj komentarz do sprawozdania	Komentarz widoczny na karcie sprawozdania
7.	Wypisz użytkownika z kursu	Użytkownik wypisany z kursu. Zniknął z listy "Moje kursy"
8.	Wyszukaj ponownie kurs "Zespołowy Projekt Programistyczny TWO IMNS_MP" i wejdź do niego	Pojawia się ekran do wpisania klucza dostępu do kursu
9.	Podaj klucz dostępu by zapisać się na kurs	Pojawia się ekran kursu
10.	Wyloguj się z aplikacji	Użytkownik wylogowany

Tabela 3. Trzeci przypadek testowy stworzony na potrzeby badań

Lp	Opis	Rezultat
1.	Zaloguj się do aplikacji	Użytkownik się zalogował
2.	Przejdź do zakładki 'Kursy'	Wyświetlona została lista działów
3.	Wejdź do działu 'Kursy testowe'	Wyświetlone zostały lista kursów w dziale
4.	Wciśnij przycisk 'Dodaj nowy kurs'.	Wyświetlony został formularz dla nowego kursu
5.	Uzupełnij pola obowiązkowe	Obowiązkowe pola zostały wypełnione
6.	Uzupełnij pole: 'Podsumowanie kursu'	Pole uzupełnione
7.	Wciśnij przycisk 'Zapisz i wyświetl'	Ekran kursu wyświetlony
8.	Włącz tryb edycji	Kurs przeszedł w tryb edycji
9.	Dodaj aktywność 'Plik' do 'Temat 1'	Wyświetlony ekran dodawania 'lekcji'
10.	Uzupełnij pola 'Nazwa' i 'Opis'	Pola uzupełnione
11.	Dodaj plik	Plik dodany
12.	Wciśnij przycisk 'Zapisz i wyświetl'	Plik jest widoczny na ekranie kursu
13.	Pobierz plik na dysk	Plik pobrany na dysk
14.	Wyloguj się z aplikacji	Użytkownik się wylogował

Przykład 1. Fragment kodu testu modyfikującego profil użytkownika, uzupełniający pola w sekcji „Inne nazwy” (Selenium)

```
.clickOptionalSection()
.setUrl("www.test.te")
.setContact(Contact.icq, "test")
.setContact(Contact.skype, "test")
.setContact(Contact.aim, "test")
.setContact(Contact.yahoo, "test")
.setContact(Contact.msn, "test")
.setIdNumber("111")
.setPhone1("134234223")
```

```
.setPhone2("123432344")
.setAddress("testowa 1/3")
```

Przykład 2. Fragment kodu testu modyfikującego profil użytkownika, uzupełniający pola w sekcji „Inne nazwy” (UFT)

```
'<----- Step 7 ----->
Browser("Moodle").Page("Common").WebButton("Button role button").SetTOProperty "name", Opcjonalne"
Browser("Moodle").Page("Common").WebButton("Button role button").Click

blnCheck = FnSetWebEdit("Strona WWW", "Opcjonalne", test.pl")
blnCheck = blnCheck AND FnSetWebEdit("Icq", "Opcjonalne", "test")
blnCheck = blnCheck AND FnSetWebEdit("Skype", "Opcjonalne", "test")
blnCheck = blnCheck AND FnSetWebEdit("AIM", "Opcjonalne", "test")
blnCheck = blnCheck AND FnSetWebEdit("Yahoo", "Opcjonalne", "test")
blnCheck = blnCheck AND FnSetWebEdit("MSN", "Opcjonalne", "test")
blnCheck = blnCheck AND FnSetWebEdit("Telefon", "Opcjonalne", "883388332")
blnCheck = blnCheck AND FnSetWebEdit("Tel. komórkowy", "Opcjonalne", "111111222")
blnCheck = blnCheck AND FnSetWebEdit("Adres", "Opcjonalne", "Moodlowa 2/2")
```

### 3.4. Porównanie testów

Napisane testy zostały poddane pomiarowi. Każdy z nich został uruchomiony po 10 razy, a otrzymane wyniki zostały porównane ze sobą. Pomiarzy miały na celu zebranie wyników i wyciągnięcie z nich wartości średniej oraz odchylenia standardowego.

## 4. Wyniki

Dla wszystkich testów, średni czas trwania testu jest niższy niż dla testów pisanych z użyciem Selenium. Różnica, w przypadku testu nr 3 sięga aż pół minuty. To dużo, biorąc pod uwagę, że oba testy robią to samo. Wyniki zostały przedstawione w Tabeli 4.

Tabela 4. Pomiary czasu trwania egzekucji poszczególnych testów w obu technologiach

	Test 1	Test 2	Test 3
	<b>Czas [minuty]</b>		
<b>HP UFT</b>	01:45	02:48	02:10
	01:30	02:19	02:17
	01:37	02:23	02:26
	01:33	02:12	02:06
	01:25	02:40	02:11
	01:29	02:17	02:19
	01:25	02:14	02:25
	01:30	02:13	02:09
	01:25	02:33	02:11
	01:29	02:32	02:16
<b>Średnia</b>	01:30	02:25	02:15
<b>Odchylenie standardowe</b>	0,004353	0,008676	0,004721
<b>Selenium</b>	01:05	02:12	01:32
	01:06	02:02	01:53
	01:06	02:48	01:52
	01:13	02:35	01:43
	01:13	02:15	01:40
	00:59	02:22	01:46
	01:08	02:30	01:41
	01:03	02:09	01:41
	01:10	02:19	01:57
01:10	02:29	01:50	
<b>Średnia</b>	01:07	02:22	01:45
<b>Odchylenie standardowe</b>	0,003072	0,009502	0,00523

## 5. Wnioski

Z przeprowadzonych badań wynika, że jeśli ważnym argumentem przy wyborze technologii automatyzacji jest, by czas potrzebny na uruchomienie wszystkich testów był jak najkrótszy, to lepszym wyborem będzie zdecydowanie Selenium WebDriver. Jest ono szybsze pod każdym względem od UFT. Jednak wybór odpowiedniej technologii to o wiele bardziej złożona sprawa niż wybranie tej najszybszej. To czy dana technologia będzie najbardziej odpowiednia mają wpływ również takie aspekty jak:

- obsługiwane technologie;
- cena;
- łatwość znalezienia specjalistów od danej technologii, bądź wyszkolenia nowego;
- łatwość tworzenia skryptów.

Im większą posiada się wiedzę na temat tego, czego się oczekuje, tym łatwiej jest podjąć ostateczną decyzję. W przypadku, gdy nie zachodzi potrzeba automatyzowania zwykłej aplikacji komputerowej, to Selenium można ocenić, jako lepszy wybór niż UFT.

## Literatura

- [1] AUTOMATION TESTING Tutorial: Process, Planning & Tools, <http://www.guru99.com/automationtesting.html> [11.03.2018]
- [2] H. P. Bajaj, Choosing the right automation tool and framework is critical to project success, <https://pdfs.semanticscholar.org/526a/5b00714e1cad6809234020cd8f81fd905a.pdf> [11.03.2018]
- [3] P. Kunte, Prof. D. Mane, Automation Testing of Web based application with Selenium and HP UFT (QTP), International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 06 June -2017 <https://www.irjet.net/archives/i6/IRJETV4I6500.pdf> [11.03.2018]
- [4] Y. Kumar, Journal of Emerging Technologies and Innovative Research (IRJET),, 09-2015, Volume 2, Issue 9 Comparative Study of Automated Testing Tools: Selenium, SoapUI, HP Unified Functional Testing and Test Complete, <http://www.jetir.org/papers/JETIR1509007.pdf> [11.03.2018]
- [5] P. Wałachowski, Porównanie narzędzi do automatyzacji cross-przeglądarkowych testów funkcjonalnych aplikacji webowej, <https://moodle.cs.pollub.pl/mod/resource/view.php?id=6709> [11.03.2018]
- [6] Test Automation Tool Comparison – HP UFT/QTP vs. Selenium, <http://www.aspiresys.com/WhitePapers/QTPvsSelenium.pdf> [11.03.2018]
- [7] Introduction to HP Unified Functional Testing (UFT) - Latest version of Quick Test Professional (QTP), <http://www.softwaretestingclass.com/introduction-to-hp-unified-functional-testing-uft/> [11.03.2018]
- [8] HP UFT/QTP vs. Selenium – Automated Test Tool Comparison, <http://www.optimusinfo.com/blog/7392> [11.03.2018]