

Ocena jakości wybranych narzędzi do automatyzacji testów aplikacji

Łukasz Szczepkowicz*, Beata Pańczyk

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

Streszczenie. Celem niniejszego artykułu jest porównanie wybranych narzędzi do realizacji testów oprogramowania. Do analizy wybrano 7 narzędzi typu open source: Selenium IDE, Selenium WebDriver, JMeter, SoapUI, SikuliX, AutoIT, Katalon Studio. Badania przeprowadzono za pomocą dedykowanych aplikacji testowych (desktopowej, internetowej oraz usługi sieciowej).

Słowa kluczowe: testowanie automatyczne; narzędzia automatyzacji testów

*Autor do korespondencji.

Adres e-mail: lukasz.szczepkowicz@gmail.com

Quality evaluation of selected tools to automate application testing

Łukasz Szczepkowicz*, Beata Pańczyk

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

Abstract. The subject of this article is a comparison of selected tools for the implementation of software tests. Seven open source tools were selected for the analysis: Selenium IDE, Selenium WebDriver, JMeter, SoapUI, SikuliX, AutoIT, Katalon Studio. The research was carried out using dedicated test applications (desktop, web and web service) with the same functionality.

Keywords: automatic testing; test automation tools

*Corresponding author.

E-mail address: lukasz.szczepkowicz@gmail.com

1. Wstęp

Aplikacje internetowe, webowe czy mobilne są integralną częścią życia człowieka ale większość użytkowników spotkało się przynajmniej raz z niepoprawnym ich działaniem. Aby zapewnić wysoką jakość oprogramowania jego producenci przed wydaniem produktu przeprowadzają testy aplikacji, dzięki czemu minimalizują ryzyko awarii systemu. Coraz większe znaczenie ma automatyzacja testów, która staje się niezastąpionym elementem testowania oprogramowania, minimalizując jednocześnie czas i koszty wytworzenia gotowego produktu.

Niniejszy artykuł został poświęcony porównaniu programów do automatyzacji testów różnych aplikacji, dostępnych w postaci rozwiązań typu open source. Analizy i oceny dokonano dla 7 najpopularniejszych:

- Selenium IDE,
- Selenium WebDriver,
- JMeter,
- SoapUI,
- SikuliX,
- AutoIT,
- Katalon Studio.

2. Kryteria porównania

W celu porównania narzędzi zostały zaimplementowane dedykowane aplikacje testowe w postaci aplikacji

desktopowej, internetowej oraz usługi sieciowej (Web service). Każda aplikacja posiada taką samą złożoność (liczbę ekranów, liczbę pól na ekranie itp.). Ocena narzędzi została wystawiona na podstawie następujących kryteriów:

- łatwość instalacji,
- łatwość obsługi,
- wsparcie techniczne,
- szybkość działania,
- wady oraz zalety,
- umiejętności potrzebne do pracy z danym narzędziem,
- dodatkowe czynniki wpływające na ocenę narzędzia.

3. Ocena narzędzi

3.1. Selenium IDE

Narzędzie Selenium IDE to zintegrowane oprogramowanie będące dedykowaną wtyczką dla przeglądarki Mozilla Firefox, służące do przeprowadzania testów funkcjonalnych aplikacji webowych. Narzędzie to jest łatwe w użyciu z uwagi na dołączoną funkcjonalność nagrywania testu, dlatego też może być odpowiednie dla początkujących testerów [1].

Selenium IDE jest wtyczką do przeglądarki dlatego jego instalacja nie jest problematyczna. Zintegrowane narzędzie do nagrywania testu znacznie upraszcza jego obsługę.

Narzędzie nie posiada wsparcia technicznego producenta z uwagi na to, że jest to rozwiązanie typu open source.

Jednakże jest to popularne narzędzie, w Internecie nie jest trudno znaleźć informacje na jego temat bądź uzyskać wsparcie innych użytkowników.

Jest to bardzo proste narzędzie i celowo szybkość jego działania została obniżona. Jednakże przeglądarka (na której uruchamia się test), jest włączona przed jego uruchomieniem, dlatego nie traci się czasu uruchamianie samej przeglądarki.

Narzędzie nie wymaga umiejętności programowania lub specjalnej wiedzy informatycznej. Obsługuje tylko jedną przeglądarkę, na której zostało zainstalowane, dlatego za jego pomocą nie jest możliwe zrealizowanie profesjonalnych testów automatycznych. Wady oraz zalety narzędzia przedstawia tabela 1.

Tabela 1. Wady i zalety Selenium IDE

| Wady | Zalety |
|---|---|
| Brak możliwości wyboru daty z rozwijalnego kalendarza, Brak możliwości dodania pliku, Skrypt powstały poprzez narzędzie do rejestracji akcji jest często konwertowany niewłaściwie. | Łatwe w obsłudze, Wbudowane narzędzie do rejestrowania akcji, Możliwość eksportu skryptu na różne języki programowania (java, php). |

3.2. Selenium Web Driver

Selenium WebDriver to powszechnie stosowany framework do testów automatycznych na przeglądarkach internetowych. Skrypty testowe pisane są za pomocą środowiska programistycznego, dlatego też możliwe jest połączenie bibliotek Selenium z wieloma dodatkowymi narzędziami, przez co jego możliwości są ogromne. Selenium dostępne jest dla kilku popularnych języków programowania takich jak: Java, C#, Ruby, Python [1, 2].

Selenium WebDriver nie jest łatwy w instalacji. Wymagana jest znajomość narzędzia maven lub obsługa środowiska programistycznego. Nie jest też łatwe w obsłudze - niezbędne jest posiadanie umiejętności programowania na poziomie minimum podstawowym.

Nie ma wsparcia technicznego producenta z uwagi na to, że jest to rozwiązanie open source. Jednakże jest to popularne narzędzie, dlatego też w Internecie nie jest trudno znaleźć informacje na jego temat bądź uzyskać wsparcie innych użytkowników.

Selenium WebDriver to biblioteki dedykowane dla wybranego języka programowania. Najdłużej podczas testu trwa uruchamianie przeglądarki. Sam test jest szybki i często wymaga dodania czasów oczekiwania, aby strona internetowa zdołała się prawidłowo wczytać.

Narzędzie wymaga umiejętności programowania. Przy bardziej złożonych testach wymagane jest również znajomość obsługi baz danych. Narzędzie to nie jest łatwe w obsłudze ale z uwagi na swoją specyfikę możliwe jest jego integracja z takimi narzędziami jak maven, git, Jenkins, TestNG + TestNG Reporting, AutoIT czy Apache POI. Dzięki temu możliwe jest stworzenie profesjonalnych testów

automatycznych. Wady oraz zalety narzędzia przedstawia tabela 2.

Tabela 2. Wady i zalety Selenium WebDriver

| Wady | Zalety |
|--|--|
| Trudne w utrzymaniu testy z dużą liczbą kroków, Wymagana umiejętność programowania i obsługi środowiska programistycznego, Przymusowe dodawanie czasu oczekiwania z powodu zbyt szybkiego działania narzędzia. | Testy dla wielu przeglądarek, Możliwość integracji z innymi narzędziami testowymi, Niezależny od języka programowania. |

3.3. Apache JMeter

Jest to program w języku java, dedykowany do tworzenia, wykonywania oraz monitorowania przebiegu testów obciążeniowych, przeciążeniowych oraz wydajnościowych (częściowo również testów funkcjonalnych). Z powodzeniem wykorzystywany jest do symulowania i badania wzmoczonego ruchu na serwerach, grupie serwerów oraz w sieci [3, 4].

JMeter jest łatwy w instalacji. Wystarczy rozpakować paczkę zip i uruchomić plik z programem. Niestety nie jest łatwy w obsłudze - niezbędna jest znajomość protokołów sieciowych.

JMeter nie posiada wsparcia technicznego producenta (jest darmowe). W Internecie natomiast można znaleźć informacje oraz wsparcie innych użytkowników.

JMeter przeznaczony jest do badania wydajności, obciążenia oraz przeciążenia. Nie jest możliwe porównanie jego szybkości działania z innymi testowanymi aplikacjami z uwagi na charakter narzędzia, dlatego też szybkość działania nie była oceniana. Wady oraz zalety narzędzia przedstawia tabela 3.

Tabela 3. Wady i zalety JMeter

| Wady narzędzia | Zalety narzędzia |
|--|---|
| Wymagane realistyczne odwzorowanie ustawień klient/server, Ograniczenia maszyny testowej mogą pojawić się w opóźnieniach w czasach odpowiedzi, Narzędzie zużywa wiele zasobów. | Możliwość testowania wielu typów systemów, Duża skalowalność dla zwiększenia zakresu obciążenia. |

3.4. Soap UI

SoapUI jest idealnym narzędziem wspomagającym testy funkcjonalne usług sieciowych. Możliwości tego narzędzia wykraczają poza sam protokół SOAP, ponieważ pozwalają na testowanie aplikacji wykorzystujących takie technologie jak REST, JMS, AMF oraz JDBC. Dodatkowo SoapUI korzysta z języka Groovy przez co możliwa jest jego rozbudowa [5, 6].

SoapUI jest łatwy w instalacji i w obsłudze. Realizacja prostego testu jak w przypadku testowanej aplikacji Web-service nie jest skomplikowana. Bardziej złożone testy

wymagają jednak większej znajomości narzędzia oraz sposobów komunikacji REST oraz SOAP.

SoapUI nie posiada wsparcia technicznego. W Internecie natomiast można znaleźć informacje na jego temat oraz wsparcie innych użytkowników.

Narzędzie przeznaczone jest do testów Web-service. Z uwagi na charakter testowanych systemów, działanie narzędzia jest błyskawiczne. Wymaga znajomości sposobów komunikacji REST oraz SOAP. W wersji PRO (wersja płatna) posiada dodatkowe funkcjonalności umożliwiające testowanie wydajności. Niestety w wersji darmowej opcja ta nie jest dostępna. Wady i zalety narzędzia przedstawia tabela 4.

Tabela 4. Wady i zalety SoapUI

| Wady narzędzia | Zalety narzędzia |
|---|--|
| Własny wbudowany klient więc testy nie odzwierciedlają standardowego użycia Web-service, Testy SoapUI nie zastępują testów funkcjonalnych aplikacji. | Możliwość testowania aplikacji typu Web-service, Używanie narzędzia nie wymaga znajomości programowania, Automatyczne generowanie zapytań w oparciu o plik WSDL, Wbudowane asercje. |

3.5. SikuliX

Działanie SikuliX polega na sprawdzeniu czy zdefiniowany wycinek ekranu jest dostępny na ekranie komputera. W przypadku jego znalezienia możliwe jest podjęcie na nim akcji takiej jak kliknięcie elementu czy przekazanie tekstu. SikuliX jest dostępny w postaci programu desktopowego, ale również jako biblioteki, dzięki czemu możliwe jest programowanie testów za jego pomocą [7].

Instalując narzędzie SikuliX należy wiedzieć jaki rodzaj instalacji jest potrzebny (dostępne są różne instalacje produktu). Jest łatwy w obsłudze jeśli wybrano aplikację typu desktop. Wersja bibliotek wykonywalnych wymaga jednak umiejętności programowania, tak jak w przypadku Selenium WebDriver. SikuliX nie posiada wsparcia technicznego producenta. W Internecie także mało jest informacji na jego temat. SikuliX nie jest szybkim narzędziem, ale samo działanie testów jest efektywne.

Narzędzie wymaga umiejętności programowania tylko jeśli użyto bibliotek wykonywalnych SikuliX. Aplikacja typu desktop nie wymaga umiejętności programowania lub specjalnej wiedzy informatycznej aby go używać. Z uwagi na GUI - narzędzie sprawdza się dla aplikacji desktopowych i internetowych. Wady oraz zalety narzędzia przedstawia tabela 5.

Tabela 5. Wady i zalety SikuliX.

| Wady narzędzia | Zalety narzędzia |
|---|---|
| Testy wykonywane są powoli, Podobne wycinki ekranu mogą być błędnie interpretowane, Brak wsparcia technicznego producenta oraz informacji w Internecie. | Możliwość testowania aplikacji z GUI, Bardzo łatwa obsługa narzędzia typu desktop. |

3.6. AutoIT

AutoIT to język zaprojektowany do automatyzowania aplikacji na systemach Microsoft Windows już w 1999 roku. Jest rozwijany do dziś przez AutoIt Team. Aktualna wersja języka (v3.3.14.5) ma składnię podobną np. do Visual Basic czy JavaScript [8, 9].

AutoIT nie jest trudne w instalacji. Składa się z kilku komponentów wchodzących w skład jednego narzędzia. Aby prawidłowo ich używać należy zapoznać się z wszystkimi oferowanymi komponentami. Dodatkowo skrypty testowe są pisane w specjalnym języku podobnym do Basic co jest niewielkim utrudnieniem. Nie posiada wsparcia technicznego producenta. W Internecie natomiast mało jest informacji związanych z tym narzędziem.

AutoIT działa szybko. Skrypt może działać bez potrzeby uruchamiania któregoś z komponentów AutoIT. Wymaga znajomości języka skryptowego w przypadku realizacji bardziej skomplikowanych testów. Jest dostępne jako zestaw aplikacji typu desktop oraz bibliotek wykonywalnych. Wady oraz zalety narzędzia przedstawia tabela 6.

Tabela 6. Wady i zalety AutoIT

| Wady narzędzia | Zalety narzędzia |
|---|--|
| Narzędzie korzysta z własnego języka skryptowego, Dostępny tylko dla systemu Windows, Brak wsparcia technicznego producenta oraz informacji w Internecie. | Możliwość testowania aplikacji desktopowych, Dodatkowe narzędzie do rejestrowania testu z którego następnie budowany jest skrypt testowy, |

3.7. Katalon Studio

Katalon Studio to darmowe rozwiązanie do testów automatycznych. Jest oparte na frameworkach Selenium oraz Appium. Pierwsze publiczne wydanie narzędzia odbyło się we wrześniu 2016 roku, więc jest to stosunkowo nowe rozwiązanie [10].

Jest łatwe w instalacji. Do jego używania potrzebne jest jednak założenie konta na stronie producenta <https://www.katalon.com/>. Katalon Studio nie jest trudny w obsłudze. Posiada intuicyjny interfejs a same składowe testów są podzielone w osobnych modułach. Narzędzie mimo to jest bardzo rozbudowane.

Katalon Studio został utworzony przez firmę Katalon LLC we wrześniu 2016 roku. Na oficjalnej stronie programu znajdują się poradniki i filmy jak korzystać z narzędzia. Dodatkowo, bardziej zaawansowane wsparcie techniczne może być płatne. Poza oficjalną stroną w Internecie nie ma o nim zbyt wielu informacji. Katalon Studio działa ze średnią prędkością w odniesieniu do innych weryfikowanych narzędzi. Narzędzie nie wymaga umiejętności programowania. W przypadku jednak testów aplikacji mobilnych dobrze jest znać technologie mobilne jak np. Android natomiast w przypadku usług sieciowych - sposobów komunikacji z serwerem. Narzędzie pozwala grupować przypadki testowe w zestawy oraz przechowuje oddzielnie obiekty interpretujące

elementy strony, dlatego też może służyć do realizacji profesjonalnych testów automatycznych. Wady oraz zalety narzędzia przedstawia tabela 7.

Tabela 7. Wady i zalety Katalon studio.

| Wady narzędzia | Zalety narzędzia |
|--|---|
| Mało popularne narzędzie, Płatne dodatkowe wsparcie techniczne, | Możliwość testowania na wielu przeglądarkach, Możliwość testowania aplikacji mobilnych, webowych oraz Web-service. |

3.8. Zestawienie wyników

W tabeli 8 przedstawiono wyniki przeprowadzonej analizy. Poszczególne kryteria uzyskały oceny w skali 1-5 (gdzie 5 jest oceną najwyższą). W przypadku konieczności posiadania dodatkowych umiejętności – odejmowany jest 1 lub 2 punkty. Dodatkowe czynniki natomiast oceniano uwzględniając ich pozytywny bądź negatywny wpływ na oceniane narzędzie. Ilość punktów ujemnych lub dodatnich ustalono na podstawie istotności badanego czynnika.

Tabela 8. Ocena wybranych narzędzi

| Ocena | Selenium IDE | Selenium WebDriver | JMeter | SoapUI | SikuliX | AutoIT | Katalon Studio |
|------------------------|--------------|--------------------|-----------|-----------|-----------|-----------|----------------|
| Łatwość instalacji | 5 | 3 | 5 | 5 | 4 | 4 | 4 |
| Łatwość obsługi | 5 | 2 | 3 | 4 | 4 | 3 | 4 |
| Wsparcie techniczne | 4 | 4 | 3 | 3 | 2 | 2 | 4 |
| Szybkość działania | 4 | 5 | brak | 5 | 3 | 5 | 4 |
| Dodatkowe umiejętności | 0 | -2 | -1 | -1 | 0 | -1 | -1 |
| Dodatkowe czynniki | -2 | 5 | 3 | 0 | 2 | 2 | 3 |
| Ocena końcowa | 16 | 17 | 14 | 16 | 15 | 15 | 18 |

4. Wnioski

Na podstawie przeprowadzonych badań na bazie trzech różnych aplikacji testowych stwierdzono, że nie ma uniwersalnego narzędzia testowego, umożliwiającego kompleksowe testy automatyczne. Wybór narzędzi musi być przemyślany i dostosowany do konkretnego rodzaju aplikacji.

Automatyzacja testów jest niezbędna, szczególnie w przypadku złożonych systemów. Istnieje wiele narzędzi do wyboru, również tych darmowych, analizowanych w niniejszym artykule. Niestety nie jest możliwe przetestowanie, za pomocą jednego narzędzia, różnych rodzajów systemów lub wykonanie różnych rodzajów testów.

Te bardziej zaawansowane narzędzia automatyzacji testów wymagają od użytkownika dodatkowych umiejętności np. programowania, obsługi komunikacji z serwerem, znajomości algorytmów itp. Są również narzędzia, z których można korzystać bez specjalistycznej wiedzy np. SikuliX czy Selenium IDE. Takie narzędzia są najbardziej odpowiednie dla osób, które dopiero zaczynają przygodę z automatyzacją testów. Stosowanie bardziej złożonych narzędzi takich jak Katalon Studio czy Selenium WebDriver wymaga większego zaangażowania, ale testy napisane za ich pomocą są lepsze, bardziej profesjonalne i efektywne.

Każde z zaprezentowanych w artykule narzędzi, posiada opisane wcześniej wady i zalety. Ocena końcowa została wystawiona na podstawie indywidualnych przemyśleń i doświadczeń, więc każdy użytkownik może wykorzystać ten materiał jako wskazówkę, które z nich będzie najlepszym wyborem do realizacji testów różnego rodzaju aplikacji.

Literatura

- [1] Sams P.: Selenium. Automatyczne testowanie aplikacji. Wydawnictwo Helion, 2017.
- [2] Gundecka U.: Selenium i testowanie aplikacji. Receptury. Wydawnictwo Helion, 2017.
- [3] Halili E. H.: Apache JMeter. Wydawnictwo Packt Publishing, 2008.
- [4] <https://www.youtube.com/watch?v=mgTE8it2KvA> [15.06.2018]
- [5] https://www.youtube.com/watch?v=izA_Xgbj7II [15.06.2018]
- [6] <http://cezarywalenciuk.pl/blog/programing/post/soapui--program-do-testowania-uslug-soap-i-rest> [06.2018]
- [7] https://www.youtube.com/watch?annotation_id=annotation_1950573397&feature=iv&index=2&list=PL1A2CSdiySGJJNe3WzCezcI7by5SPfJTS&src_vid=VdCOg1bCmGo&v=IUYoezac4Q [06.2018]
- [8] <https://www.youtube.com/watch?v=3RmxYAxhzl0> [06.2018]
- [9] <https://pl.wikipedia.org/wiki/AutoIt> [06.2018]
- [10] https://en.wikipedia.org/wiki/Katalon_Studio [06.2018]