

ЗАСТОСУВАННЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ В ПРОЦЕСІ РОЗРОБКИ КЛІЄНТ-СЕРВЕРНОЇ СИСТЕМИ

Вінницький національний технічний університет

***Анотація:** У даній роботі приділено увагу застосуванню автоматизованого тестування в процесі розробки клієнт-серверної системи. Проведено аналіз інструментів для автоматизованого тестування.*

Ключові слова: автоматизоване тестування, тестування клієнт-серверної системи, інструмент Selenium Webdriver.

***Abstract:** This paper focuses on the use of automated testing in the process of developing a client-server system. An analysis of tools for automated testing.*

Keywords: automated testing, client-server system testing, Selenium Webdriver tool.

Вступ

Веб-додатки вже давно увійшли в наше життя і стали популярними серед користувачів. Галузь розробки не стоїть на місці. З кожним роком створюються нові і удосконалюються вже існуючі веб-додатки. За сучасного підходу до гнучкої розробки програмного забезпечення agile звичайний підхід до тестування на базі задокументованих вимог не завжди доцільний. Саме тому автоматизація процесу тестування програмного продукту за автоматично створеними тестами є одним з сучасних підходів і з кожним роком набуває актуальності.

Тестування - ітераційний процес. Після виявлення та виправлення кожної помилки обов'язково треба повторити тести, для того щоб бути впевненим, що програма працює як заплановано. Більше того, для ідентифікації причини виявленої проблеми може знадобитися проведення додаткового тестування. При цьому потрібно завжди пам'ятати про фундаментальний висновок, зроблений професором Едджером Дейкстри у 1972 році: "Тестування програм може

служити доказом наявності помилок, але ніколи не доведе їхню відсутність!" [1].

Аналіз інструментів для автоматизації тестування на мові програмування Python

Selenium - це популярний безкоштовний пакет інструментів для автоматизації тестування веб додатків, які емулюють дії користувача. Він простий в використанні і має зручний інтерфейс. Якщо порівнювати Selenium з інструментами автоматизації інших виробників, такими як QTP, UFT, SilkTest, виявиться, що Selenium потребує менше апаратного забезпечення. Для розробників і тестувальників, які мають досвід та навички програмування та створення сценаріїв, Selenium пропонує гнучкість, яку не можна побачити у багатьох інших інструментах автоматизації тестів та фреймворках. Користувачі можуть писати тестові сценарії багатьма різними мовами (такими як Java, Groovy, Python, C #, PHP, Ruby та Perl), які працюють у різних системних середовищах (Windows, Mac, Linux) та браузерах (Chrome, Firefox, IE).

До недоліків Selenium відносять необхідність володіння навичками програмування. Мобільні і десктопні додатки не можуть бути протестовані за допомогою цього інструменту. Крім того, Selenium має обмежений функціонал в порівнянні з платними аналогами, такими як, наприклад TestComplete. Немає власної системи звітування. Але можна вирішити це питання використовуючи зовнішні інструменти, такі як TestNG або Junit.

Katalon Studio - це потужне рішення для автоматизації тестування API, веб-додатків, тестування мобільних пристроїв. Він також має багатий набір функцій для цих типів тестування та підтримує декілька платформ, включаючи Windows, macOS та Linux.

Використовуючи Selenium та Appium, Katalon Studio забезпечує унікальне інтегроване середовище для тестувальників, які мають труднощі в інтеграції та розгортанні різних фреймворків та бібліотек. Перевагою Katalon Studio є те, що це безкоштовний інструмент, який забезпечує функцію запису, відтворення та створення сценаріїв у ручному режимі для створення тестових випадків.

Недолік Katalon Studio в тому, що на відміну від таких інструментів, як Selenium та TestComplete, створення сценарію обмежується лише Java та Groovy. Крім того даний інструмент не може автоматизувати десктопні додатки на відміну від деяких ліцензованих інструментів, таких як UFT та TestComplete. Katalon має невелику спільноту, закритий вихідний код, має затримку в оновленні документації, чого не можна сказати про Selenium. Є

проблеми з продуктивністю, наприклад інструмент може зависнути. Мобільне тестування займає більше часу через необхідність збору та запису коду [12].

Unified Functional Testing (UFT) - популярний комерційний інструмент для тестування десктопних, веб та мобільних додатків. Простий у встановленні, використовує можливості для тестування API. Підтримує тестування GUI та API за допомогою віртуалізованих засобів.

Перевага данного інструменту в тому, що UFT має дуже багато функцій, але в той же час, його досить просто освоїти, і для цього не потрібно витратити багато часу для навчання персоналу. Існує активна спільнота підтримки. UFT може масштабуватись та працювати на декількох машинах з одного контролера, що дозволяє виконувати сотні тестів протягом ночі.

До недоліків UFT відносять його вартість, він обійдеться компанії дорожче ніж інші інструменти. І незважаючи на те, що час написання сценаріїв у UFT менший, час виконання в середньому більше, ніж порівняно з іншими інструментами, такими як наприклад Selenium. В UFT неможливо запустити кілька потоків / екземплярів, на відміну від Selenium Grid, де користувач може запускати декілька екземплярів програми в різних браузерах одночасно. Окремі витрати на додатки - Java, Oracle, SAP, .Net, Seibel, PeopleSoft тощо.

TestComplete - це автоматизований інструмент тестування Smartbear. Він підтримує різні платформи, такі як тестування веб, десктопних та мобільних додатків. Також має декілька функцій автоматизації тестів, таких як тестування на основі даних, тестування API, тестування за допомогою ключових слів і тестування між браузерами. Як і UFT, TestComplete має механізм розпізнавання об'єктів, який може точно виявляти динамічні елементи інтерфейсу користувача.

До переваг користування інструментом відносять можливість написання сценарію на 7 різних мовах програмування (JavaScript, Python, VBScript, Jscript, DelphiScript, C # и C +). Інструмент також дає можливість користуватися функцією запису та відтворення. Тестери можуть вставити контрольні точки в тестові етапи для перевірки результатів.

До недоліків відносять високу вартість ліцензії, а також те що інструмент має проблеми зі стабільністю. TestComplete не дозволяє тестувати додатки для Mac, тому з'являється необхідність в придбанні певного програмного забезпечення для віртуалізації. Відсутність можливості спрощеного запису скриптів та доступність використання тільки на ОС Windows [13]. Можливі випадкові збої,

іноді записаний тест може спричинити збій програми, якщо він занадто великий для системи, на якій працює. Порівняно з іншими інструментами, такими як UFT, TestComplete не вистачає таких областей, як завантаження бібліотеки сценаріїв під час виконання, у TestComplete потрібно мати більше досвіду, використовуючи її відповідно до потреб користувача.

Висновки

Після детального аналізу існуючих інструментів для розробки автоматизованих тестів для клієнт-серверної системи проведення платежів був обраний Selenium WebDriver. Даний фреймворк простий у використанні, підтримує багато мов програмування та взаємодіє майже з усіма відомими браузерами.

Після огляду і розбору існуючих мов програмування для розробки програмного забезпечення обрано мову Python. Це популярна об'єктно-орієнтована мова програмування високого рівня, яку часто вибирають завдяки своїй простоті, широкому вибору створених бібліотек, наявності великої кількості модулів і зрозумілому синтаксису.

Для перевірки роботи розробленого програмного забезпечення був створений набір тест-кейсів. Цей набір був виконаний спочатку мануально, а потім за допомогою розроблених автоматизованих тестів.

Проаналізувавши отримані результати можна зробити висновок, що розроблене програмне забезпечення працює коректно, тому воно було використано для автоматизованого тестування клієнт-серверної системи проведення платежів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1.Тестування програмного продукту [Електронний ресурс]. Режим доступу: <http://lib.mdpu.org.ua/e-book/vstup/L11.htm>. – Назва з екрану.

12.Дастин Э. Автоматизированное тестирование программного
13.Обеспечения. Внедрение. Управление и эксплуатация пер з англ. /
Э. Дастин, Д. Рэшка, Д. Пол – М: Лорі, 2013. – 589 с. – ISBN
5-85582-186-2

Основенко Наталя Володимиріна - студентка групи 2СІ-16Б, факультет комп'ютерних систем і автоматики, Вінницький національний технічний університет, м.Вінниця, email:nataliosn3452@gmail.com

Науковий керівник: **Володимир Юрійович Коцюбинський** — к.т.н., доцент, Вінницький національний технічний університет, м. Вінниця

Osnoenko Natalia Volodymyrina - student of group 2CI-16Б, Faculty of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, email: nataliosn3452@gmail.com

Supervisor: **Volodymyr Yuriyovych Kotsyubynsky** - Ph.D., Associate Professor, Vinnytsia National Technical University, Vinnytsia