DOI 10.34185/1562-9945-4-147-2023-07
УДК 004.415.53

М.О. Бубенко, В.В. Герасимов, Н.В. Карпенко, О.С. Морозов

# АНАЛІЗ МЕТОДІВ ТЕСТУВАННЯ ВЕБ-ДОДАТКІВ

*Анотація. У статті розглядається тестування веб додатків через призму власного досвіду роботи в IT компанії. Показано, що під час тестування додатків різних видів слід звертати увагу на їх особливості та на критичні зони, які повинні бути перевірені. Авторами зроблено порівняння тестування web додатків та desktop програм за різними параметрами.*
*Ключові слова: веб додаток, тестування, критичні зони, модель тестування за сценарієм.*

**Постановка проблеми.** Важливим практичним завданням, що стоїть перед розробниками, є швидке створення та супровід якісного багаторівневого програмного забезпечення (ПЗ). При цьому передбачається, що розроблений продукт відповідатиме характеристикам якості, таким як стійкість, тестованість, корисність, доступність, масштабованість, відкритість, гнучкість. Це накладає на процес розробки додаткові обмеження та правила, а саме: документування розробки на різних рівнях; покриття тестами компонентів, модулів, підсистем; управління проектами, процесами тощо [1]. І, якщо казати про тестування додатків різних видів, то слід звертати увагу на їх особливості, наприклад, веб-додатки мають критичні зони, які повинні бути перевірені. Таким чином, метою даної статті є аналіз різноманітних методів тестування веб-додатків.

**Аналіз останніх досліджень і публікацій.** Мануальне та автоматизоване тестування ПЗ мають спільну теорію тестування. Досить корисною в плані ознайомлення з термінами, які використовуються під час тестування, є стаття [2], в якій також визначені цілі тестування. Але ця стаття не відповідає на такі питання: «Що потрібно тестувати?», «Яким чином потрібно виконувати дії з тестування?», «Що знадобиться для того, щоб виконати тестування ПЗ?» та інші.

Не дивлячись на те, що автоматизоване тестування широко застосовується в IT-компаніях під час розробки різноманітного ПЗ, існує досить мало статей, які б показували практичні аспекти такого тестування. Однією з найбільш цікавих статей в цьому напрямку є [3], де автори порівняли




ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

декілька поколінь популярного інструменту автоматизації тестування веб-додатків Selenium, проаналізували його функціональні особливості, можливості та зручність роботи. Автори статті [4] описали деякі популярні засоби для тестування API, а також надали рекомендації щодо областей їх застосування. Стаття [5] повністю присвячена огляду інструментів автоматизованого тестування ПЗ, таких як Selenium, IBM Rational, SilkPerformer, TestComplete, HP QuickTest Professional, JUnit та їх порівнянню за ефективністю використання у різних додатках.

Практичні поради щодо тестування веб-додатків можна знайти в [6], де автор надає чек-лист з пунктами, які потрібно перевірити під час функціонального, інтеграційного, крос-платформного тестування, а також тестування безпеки, зручності використання, локалізації та глобалізації. Автор [7] пропонує універсальну схему для тестування веб-додатків, яка доповнює чек-лист попереднього автора та надає поради з тестування Performance та Configuration. Автор статті [8] звертає увагу на те, що інформативними є Ad-hock та негативне тестування, а також написання еквівалентних тестів і проведення інтуїтивного тестування (Exploratory testing). Окрім цього, важливим кроком є виконання регресійного тестування, що не було акцентовано в попередніх статтях.

Всі вищезазначені джерела, окрім [2], стосуються практики тестування веб-додатків, однак слід згадати й ті статті, з яких почався розвиток трьох теорій тестування: Goodenough-Gerhart [9], Weyuker-Ostrand [10] та Gourlay [11]. Ці роботи дозволяють зрозуміти основні проблеми тестування та шляхи їх вирішення.

**Основна частина.** Для того, щоб виявити ті фактори, які відрізняють тестування web-додатків від тестування desktop-додатків, складено таблицю з їх порівнянням за різними критеріями (табл.1). Це порівняння показує, що веб-додатки мають кілька особливостей, які потрібно враховувати під час їх тестування. Так, навколишнє середовище висуває вимогу щодо перевірки на правильність відображення інтерфейсів веб-додатку в різних браузерах. Під правильністю слід розуміти відповідність макетам інтерфейсів, які були попередньо розроблені дизайнерами та погоджені замовником. Таким чином, при кросбраузерному тестуванні перевіряється правильність відображення елементів графіки, шрифтів, розмір текстових символів; доступність елементів керування, чи є вони інтерактивними тощо.

Таблиця 1

Порівняння web-додатків та desktop-програм

| | **Web-додаток** | **Desktop-додаток** |
|---|---|---|
| Навколишнє середовище | Додаток розгорнутий на сервері, до якого інші комп'ютери можуть отримати доступ лише за допомогою браузера. | Додаток обмежений комп'ютером, на якому він розгорнутий. |
| Платформа | Не залежать від платформи, можуть працювати на кількох платформах без окремої розробки. | Для різних платформ потрібно розробляти окремі настільні програми. |
| Розгортання / Оновлення | Розгортання або будь-яке оновлення / виправлення потрібно виконувати лише на одному наборі серверних машин. | Розгортання або будь-яке оновлення коду потрібно виконувати на всіх клієнтських машинах. |
| Архітектура | Клієнтська частина зазвичай базується на веб-технологіях та працює в браузері, використовується протокол HTTP для передачі даних. Серверна частина може бути розподілена на багато серверів. | Програма встановлюється на комп'ютер користувача і використовує механізми операційної системи для комунікації з іншими процесами або мережевими сервісами. |
| Підключення | Веб-програми залежать від підключення до Інтернету для їх функціонування. | Настільні програми не завжди вимагають підключення до Інтернету. |
| Доступність | Веб-програми не мають жодних обмежень щодо розташування, оскільки вони розгорнуті на серверах, до яких можна отримати доступ із будь-якого місця. | Настільні програми обмежені фізичними розташуваннями, тому до них можна отримати доступ лише з машин, на яких вони за своєю суттю розгорнуті. |

Веб-додаток за допомогою різних елементів керування надає користувачу можливість виконувати дії, які визначені в функціональних вимогах. Таким чином, слід перевірити, чи всі функції реалізовані в програмному продукті, а



ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

також коректність їх роботи. Під час перевірки функціоналу веб-додатку, слід звернути особливу увагу на форми для заповнення, через які здійснюється взаємодія клієнта з сервером, а саме на таке:

— Обов'язкові поля для заповнення: чи зазначено, що вони є обов'язковими? Чи обов’язкові поля насправді є обов'язковими? Що буде, якщо поле залишити пустим?

— Введення в поле спецсимволів #, %, @, №, &, ^, $, *, <, >, { }, [ ] залежить від вимог – є поля, де вони потрібні, є поля, де їх потрібно заборонити. Наприклад, в полі з ім'ям, прізвищем та по-батькові спецсимволів не повинно бути, в полі email повинен бути один обов’язковий символ "@". Поля пароля можуть включати введення, як мінімум, одного будь-якого спеціального символу тощо.

— Наявність валідації: що відбувається, коли користувач вводить невалідні значення? Чи отримує користувач повідомлення про помилку? Що відбувається з даними після введення?

Окрім форм для заповнення є й інші елементи керування, такі як радіокнопки, випадаючі списки, прапорці, які також слід перевірити. Оскільки таких елементів може бути досить багато і, відповідно, випадків для перевірки, то рекомендується складати чек-лист UI-компонентів, які потрібно перевірити.

При тестуванні веб-додатків не можна забувати про базу даних (БД). Отже, потрібно перевіряти:

- чи збереглися в БД дані, введені з боку клієнта;
- чи коректно відображаються на стороні клієнта дані, записані в БД;
- чи можна видалити / змінити дані;
- кількість відкритих з'єднань з БД;
- швидкість обробки запитів.

Одночасно з веб-програмою може працювати велика кількість користувачів. Тому потрібно:

— проводити навантажувальне тестування для з'ясування того, скільки користувачів можуть без проблем працювати одночасно;

— перевіряти аутентифікацію та авторизацію — на сервері перевіряються ідентифікаційні дані користувача та його доступ до ресурсів веб-додатка. Після успішної аутентифікації сервер зберігає дані про користувача та його доступ, що використовуються для подальшої авторизації;

— тестувати конкурентний (паралельний) доступ до ресурсів — швидкість роботи додатків не повинна змінюватися зі збільшенням кількості користувачів;

— рівні доступу користувачів.

Виходячи з вищесказаного, можна виділити такі основні **критичні зони** веб-додатків, які потребують додаткового опису та інструкцій:

- єдність дизайну;
- навігація та «дружність» до користувача;
- функціональність;
- безпека;
- сумісність з браузерами та операційними системами;
- продуктивність.

Єдність дизайну веб-ресурсу є запорукою того, що ресурс справить враження цілісної структури, яка є гармонійною та логічною, тому перевірка єдності потребує чіткої документації, унікальної для кожного окремого ресурсу, проте не вимагає значних затрат ресурсів та часу.

Тестування продукту на навігацію та «дружність» допомагає спрямувати зусилля на покращення першого враження користувача про продукт та перетворити випадкову аудиторію на прихильників. Тому таке тестування є доволі важливим, при цьому не потребує багато ресурсів, але бажана наявність чітких рекомендацій чи втручання спеціаліста з предметної області.

У переважній кількості розроблюваних додатків функціональність є ключовою зоною уваги як при тестуванні, так і під час користування кінцевим користувачем. Отже, функціональне тестування вимагає ресурсів, досвіду від працівників, чітких вимог та рекомендацій з точки зору спеціаліста у відповідній сфері. Позитивним моментом є те, що функціональні тестові сценарії добре підлягають автоматизації, а це скорочує час перевірки та ресурси.

Перевірка на безпеку є особливо критичним аспектом саме у тестуванні веб-додатків, оскільки інтерфейс кожного користувача безпосередньо взаємодіє з єдиним сервером, тому за певних умов та зусиль секретна інформація може опинитися в руках людини, що не повинна мати доступ до неї. Це може призвести до негативних наслідків та збитків як для представника послуг, так і для інших користувачів. Часто таке тестування може ефективно виконуватися командою з 1-3 людей з досвідом та кваліфікацією у відповідній сфері і не залежить від специфіки певного веб-ресурсу.



ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Сумісність з браузерами та операційними системами є принципово важливою для продуктів, спрямованих на велику аудиторію (наприклад Інтернет магазини, соціальні мережі). У час розвинених технологій та широкої різноманітності пристроїв, що мають доступ до мережі Інтернет, кожен наявний пристрій додає перевірок. Зазвичай фізично неможливо, а за можливості вкрай не рентабельно проводити тестування на всіх існуючих пристроях, а тому, враховуючи аудиторію користувача та популярність пристроїв, для перевірок обираються найпоширеніші з них. Тестування сумісності є ресурсозатратним незалежно від специфіки продукту.

Перевірка на продуктивність є ключовою для ресурсів, що розробляються для великої кількості користувачів, які користуються *веб*-сервісом одночасно. Оскільки загалом такі тести виконуються за допомогою емуляторів та спеціальних програм імітаторів навантаження, то така перевірка є автоматизованою, але вимагає високого рівня практики та знань від виконавця.

Здатність веб-додатків обробляти і поширювати інформацію через Інтернет робить їх уразливішими, порівняно з desktop-додатками. Веб-додатки повинні забезпечувати публічний доступ до даних, а це значно підвищує вимоги до стійкості цього ПЗ. По-друге, що важливіше з точки зору тестування на проникнення, вони обробляють дані, отримані із запитів *HTTP*-протоколу — протоколу, який може використовувати безліч способів кодування і інкапсуляції інформації.

Таким чином, можна виділити тести, які зазвичай використовують під час перевірки web та desktop додатків (табл. 2).

Таблиця 2

Тести, які використовують для перевірки web та desktop-додатків

| **Web-додаток** | **Desktop-додаток** |
|---|---|
| User Interface Testing, Load, Stress, Volume Browser/OS Compatibility, Security Data, Volume Testing, Cross Browser Testing, Static Page Testing, Broken Link Testing | GUI, Functionality, Load, Backend |

Найпопулярніший інструмент для автоматизації тестування — Selenium Web Driver. Найбільш популярною сферою застосування Selenium є автоматизація тестування веб-додатків. Однак, за допомогою Selenium можна автоматизувати будь-які інші рутинні дії, що виконуються через браузер. Розробка Selenium підтримується виробниками майже всіх браузерів. Вони адап-

тують браузери для більш тісної інтеграції із Selenium, а іноді навіть реалізують вбудовану підтримку Selenium у браузері.

**Висновки.** Зроблено порівняння web-додатків та desktop-програм за такими критеріями, як навколишнє середовище, платформа, архітектура, розгортання тощо. Відповідно до цього було наведено рекомендації щодо особливостей тестування web-додатків та виділено їх критичні зони.

Описано специфіку тестування критичних зон та наведено оцінку ресурсозатратності їхнього тестування.

Рекомендовано тести, які доцільно використовувати для тестування web та desktop-додатків.

**ЛІТЕРАТУРА**

1. Литвинов А.А., Карпенко Н.В. Тестирование информационных систем: модульное, интеграционное, системное: учебное пособие. / А. А. Литвинов, Н. В. Карпенко – Д.: Лира, 2016. – 284 с.

2. Mishchevskii Gennadii. Тестирование. Фундаментальная теория. URL: https://dou.ua/forums/topic/13389/ (дата звернення 12.02.2023).

3. Герасимов В.В., Беличенко А.Я. Развитие инструмента автоматизированного тестирования – SELENIUM // Системні технології. Регіональний міжвузівський збірник наукових праць. — Вип. 1 (114). — Дніпро, 2018. С. 38–44.

4. Герасимов В. В., Никитин Н. Э., Щербак А. Е., Карпенко Н. В. Тестирование API и актуальные средства его реализации // Системні технології. Регіональний міжвузівський збірник наукових праць. - Вип. 1 (120). — Дніпро, 2019. С. 71–80.

5. Герасимов В. В., Кронфельд М. Ф., Озерова Д. М. Исследование технологий автоматизированного тестирования // Системні технології. Регіональний міжвузівський збірник наукових праць. — Вип. 1(96). 2015. – С. 130-136.

6. Чек-лист тестирования WEB приложений.
URL: https://habr.com/ru/post/542422/ (дата звернення 12.02.2023).

7. Nani Davitadze. Ничего не забыть: универсальная схема для тестирования веб-приложений. URL: https://dou.ua/lenta/articles/scheme-for-qa/ (дата звернення 12.02.2023).

8. Тестирование веб-проектов: основные этапы и советы.
URL: https://qalight.ua/ru/baza-znaniy/testirovanie-veb-proektov-osnovnye-etapy-i-sovety/ (дата звернення 12.02.2023).

9. Goodenough J. B. Toward a theory of test data selection. / J. B. Goodenough, S. L. Gerhart // SIGPLAN Notices. – 1975. – 10 (6). – P. 495-510.

10. E. J. Weyuker , T. J. Ostrand. Theories of Program Testing and the Application of Revealing Subdomains. // IEEE Trans. Software Engrg. SE-6 (3) (1980). – P. 236-246.



ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

11. Gourlay J. S. A mathematical framework for the investigation of testing. / J. S. Gourlay // IEEE Trans. Software Engrg. SE. – 1983. – 9 (21). – P. 686-709.

**REFERENCES**

1. Lytvynov A.A., Karpenko N.V. Testyrovanye informatsionnykh system: modulnoe, intehratsionnoe, systemnoe: uchebnoe posobie. / A. A. Lytvynov, N. V. Karpenko – D.: Lyra, 2016. – 284 p.
2. Mishchevskii Gennadii. Testirovanie. Fundamentalnaia teoryia. URL: https://dou.ua/forums/topic/13389/
3. Herasymov V. V., Belychenko A. Ya. Razvitie instrumenta avtomatyzyro-vannoho testyrovanyia – SELENIUM // Systemni tekhnolohii. Rehionalnyi mizh-vuzivskyi zbirnyk naukovykh prats. — N1 (114). — Dnipro, 2018. P. 38–44.
4. Herasymov V. V., Nykytyn N. E., Shcherbak A. E., Karpenko N. V. Testyrovanye API i aktualnye sredstva eho realyzatsyy // Systemni tekhnolohii. Rehionalnyi mizhvuzivskyi zbirnyk naukovykh prats. — N1 (120). — Dnipro, 2019. P. 71–80.
5. Herasymov V. V., Kronfeld M. F., Ozerova D. M. Issledovanye tekhnolohyi avtomatyzyrovannoho testyrovanyia // Systemni tekhnolohii. Rehionalnyi mizhvuzivskyi zbirnyk naukovykh prats. — N1(96). 2015. P. 130-136.
6. Chek-lyst testyrovanyia WEB prylozhenyi. URL: https://habr.com/ru/post/542422/
7. Nani Davitadze. Nycheho ne zabyt: unyversalnaia skhema dlia testyrovanyia web-prylozhenyi. URL: https://dou.ua/lenta/articles/scheme-for-qa/
8. Testyrovanye web-proektov: osnovnye etapy i sovety.
URL: https://qalight.ua/ru/baza-znaniy/testirovanie-veb-proektov-osnovnye-etapy-i-sovety/
9. Goodenough J. B. Toward a theory of test data selection. / J. B. Goodenough, S. L. Gerhart // SIGPLAN Notices. – 1975. – 10 (6). – P. 495-510.
10. E. J. Weyuker, T. J. Ostrand. Theories of Program Testing and the Applica-tion of Revealing Subdomains. // IEEE Trans. Software Engrg. SE-6 (3) (1980). – P. 236-246.
11. Gourlay J. S. A mathematical framework for the investigation of testing. / J. S. Gourlay // IEEE Trans. Software Engrg. SE. – 1983. – 9 (21). – P. 686-709.

Received 16.05.2023.
Accepted 18.05.2023.

***Analysis of web application testing methods***

*An important practical task for developers is the rapid creation and maintenance of high-quality multi-level software. It is assumed that the developed product will meet the quality characteristics. And, if we talk about testing applications of different types, then you should pay attention to their features. For example, web applications have critical areas that must be checked. Thus, the purpose of this article is to analyse various methods and technics for testing web applications.*

*The article provides a detailed analysis of the latest publications related to testing web applications. It turned out that most of the articles are aimed at describing terms or general information about testing. Several articles describe automated testing with Selenium, IBM Rational, SilkPerformer, TestComplete, HP QuickTest Professional, JUnit and compare them in terms of efficiency in various applications. However, most of the articles are devoted to various aspects of manual testing.*

*In order to identify the factors that distinguish web application testing from desktop application testing, a table has been compiled comparing them according to the following criteria: environment, platform, deployment and updating, architecture, connectivity, availability. This comparison shows that web applications have several features that need to be considered when testing them.*

*In our opinion, the main critical areas of web applications that require additional description and instructions are unity of design, navigation and "friendliness" to the user, functionality, security, compatibility with browsers and operating systems, and productivity. The article describes the specifics of testing critical zones and gives an estimate of the resource consumption of their testing. Tests are also recommended, which are useful for testing web and desktop applications.*

**Бубенко Максим Олександрович** - бакалавр 4 курсу кафедри ЕОМ Дніпровського національного університету імені Олеся Гончара.
**Карпенко Надія Валеріївна** - к.ф.-м.н., доцент кафедри ЕОМ Дніпровського національного університету імені Олеся Гончара.
**Герасимов Володимир Володимирович** - к.т.н, доцент кафедри ЕОМ Дніпровського національного університету імені Олеся Гончара.
**Морозов Олександр Сергійович** - асистент кафедри ЕОМ Дніпровського національного університету ім. Олеся Гончара.

**Bubenko Maksym** - 4th year bachelor's student, Computer Engineering Department, Oles Honchar Dnipro National University.
**Karpenko Nadiia -** Ph.D in Physics and Mathematical Sciences, Associate Professor, Computer Engineering Department, Oles Honchar Dnipro National University.
**Gerasimov Volodymyr -** Ph.D. in Technical Sciences, Associate Professor, Computer Engineering Department, Oles Honchar Dnipro National University.
**Morozov Alexander Sergeyevich** – Assistant, Computer Engineering Department, Oles Honchar Dnipro National University.



ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)