

КЛАСТЕРИЗАЦІЯ ДЕФЕКТІВ Е2Е-ТЕСТІВ ВЕБ-ЗАСТОСУНКІВ

Жорнокльов А. А.

e-mail: artem.zhornoklov@nure.ua

Науковий керівник – к. т. н. Обрізан В. І.

Харківський національний університет радіоелектроніки, каф. АПОТ
м. Харків, Україна

Engineers spend significant time diagnosing failures of end-to-end web tests written with tools such as Selenium. Because tests typically depend on shared PageObject components, a single defect may cause many tests to fail. This paper proposes clustering test failures using Selenium exception types and PageObject information extracted from stack traces to identify common root causes and reduce the effort required for analyzing failing tests.

Автоматизоване тестування веб-застосунків стало стандартною практикою забезпечення якості програмного забезпечення. Одним із поширених підходів є використання наскрізних (end-to-end, e2e) тестів, які імітують поведінку користувача через браузер. У середовищі Python для цього часто застосовують інструменти Selenium WebDriver, Playwright у поєднанні з тестовими фреймворками, такими як pytest. Однак зі зростанням кількості автотестів виникає проблема масштабованості їхнього аналізу: інженери з автоматизації тестування витрачають значну частину часу на розбір причин падіння тестів [1, 2].

Типова e2e-інфраструктура тестування веб-застосунків організована у вигляді багаторівневої архітектури, де тест взаємодіє із застосунком через допоміжні абстракції. Найчастіше використовується шаблон Page Object, що інкапсулює роботу з елементами сторінки. У спрощеному вигляді залежності виглядають так:

тест → PageObject → Selenium WebDriver → веб-застосунок

Падіння тесту може бути викликане різними причинами:

1. Дефект у самому веб-застосунку.
2. Помилка в логіці тесту.
3. Помилка у PageObject (наприклад, некоректний локатор елемента).
4. Нестабільність середовища виконання (мережа, пам'ять, диск).

Практика показує, що значна частина збоїв пов'язана не з помилками бізнес-логіки застосунку, а з проблемами у тестовій інфраструктурі. Особливо важливою є роль PageObject-класів, оскільки вони використовуються одночасно багатьма тестами. Таким чином, одна помилка в PageObject може спричинити каскадні падіння десятків або навіть сотень тестів.

Більшість сучасних систем аналізу результатів тестування оперують бінарною інформацією: тест пройшов або не пройшов. Такий підхід не дозволяє ефективно визначити першопричину масових падінь тестів. У результаті інженери змушені аналізувати кожен збій окремо, що призводить до значних витрат часу.

Гіпотеза даного дослідження полягає в тому, що значна частина падінь e2e-тестів може бути згрупована за спільною першопричиною, пов'язаною з конкретним PageObject або певним класом винятків Selenium. Якщо ця гіпотеза підтвердиться, то виправлення однієї помилки у PageObject потенційно дозволить одночасно усунути велику кількість падінь тестів.

Пропонується виконувати кластеризацію падінь тестів на основі таких ознак:

1. Тип винятку Selenium, зокрема: NoSuchElementException (елемент не знайдено); TimeoutException (перевищено час очікування); ElementNotInteractableException (елемент недоступний для взаємодії); StaleElementReferenceException (посилання на елемент стало неактуальним).

2. Елемент або метод PageObject, у якому виник виняток.

3. Контекст виконання, включаючи назву тесту, клас PageObject, метод та повідомлення винятку.

Для реалізації підходу у тестовий раннер pytest було інтегровано механізм збору стек-трейсів та метаданих про падіння тестів. У результаті формується структурований запис, який містить такі поля: назва тесту, PageObject-клас, метод PageObject, тип винятку, текст повідомлення винятку. На основі цих даних виконується кластеризація тестів, що завершилися помилкою. Якщо декілька тестів мають однаковий PageObject-метод і тип винятку, вони можуть бути віднесені до одного кластеру.

Список використаних джерел:

1. Parry O., Kapfhammer G. M., Hilton M., McMinn P. A Survey of Flaky Tests. ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 31, Issue 1. Article No.: 17, Pages 1—74. <https://doi.org/10.1145/3476105>

2. Bylina B., Antończak A. Analysis of end-to-end test automation tools based on the examples of Selenium WebDriver and Playwright. Position Papers of the 19th Conference on Computer Science and Intelligence Systems. ACSIS, Vol. 40, pages 1–8 (2024). doi.org/10.15439/2024F3747

3. Leotta M.; García B.; Ricca F.; Whitehead J. (2023). Challenges of End-to-End Testing with Selenium WebDriver and How to Face Them: A Survey. 2023 IEEE 16th International Conference on Software Testing, Verification and Validation, 16–20 April 2023, Dublin, Ireland: Proceedings, Piscataway, NJ: IEEE. Pp.: 339-350. doi.org/10.1109/ICST57152.2023.00039