

УДК 004.415.538

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ CODE COVERAGE В АВТОМАТИЗОВАНОМУ ТЕСТУВАННІ ВЕБ-ЗАСТОСУНКІВ

Кунченко Д.В., Чуприна А.С.

e-mail: danylo.kunchenko@nure.ua, anastasiya.chupryna@nure.ua

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

This study examines the efficiency of code coverage metrics in automated web application testing. The research analyzes the impact of test coverage on software quality and explores its methodological aspects. Practical implementation using PHP, Codeception, and Xdebug demonstrated that high code coverage positively influences defect reduction but is not a sole quality indicator. The integration with CI/CD pipelines improves test reliability but requires proper configuration and monitoring. The findings suggest that code coverage should be used alongside other quality metrics to enhance software maintainability and regression detection.

Дослідження присвячене оцінці ефективності застосування метрик покриття коду у процесі автоматизованого тестування веб-застосунків. У роботі розглянуто методологічні особливості покриття коду, його вплив на якість програмного забезпечення та роль у процесах контролю якості. Дослідження проведене на основі PHP-фреймворку Codeception із використанням Xdebug для збору та аналізу метрик покриття.

Автоматизоване тестування є одним із ключових інструментів забезпечення якості програмного забезпечення, особливо у сфері веб-розробки. Однією з важливих метрик ефективності тестування є рівень покриття коду, який демонструє, наскільки велика частина вихідного коду перевірена тестами. Однак високий відсоток покриття не завжди є гарантією відсутності помилок і не може слугувати єдиним критерієм якості програмного продукту.

Метрики покриття коду набувають особливого значення в контексті сучасних практик розробки програмного забезпечення. В умовах швидкого та регулярного випуску оновлень програмного забезпечення, автоматизоване тестування стає критично важливим для забезпечення стабільності та надійності продукту. При цьому метрики покриття коду допомагають контролювати повноту та якість тестового покриття, що є важливим фактором для прийняття рішень про готовність продукту до випуску.

У ході дослідження було проаналізовано предметну галузь автоматизованого тестування веб-застосунків та специфіку використання метрик покриття коду. Аналіз існуючих підходів та їх обмежень показав, що питання забезпечення якості програмного забезпечення залишається актуальним, а метрики покриття коду є важливим, але не єдиним

інструментом її оцінки. В контексті аналізу предметної області варто виділити основні типи покриття коду, які використовуються при тестуванні програмного забезпечення:

1. покриття рядків коду (Lines Coverage) – вимірює відсоток виконаних рядків коду програми в процесі тестування;
2. покриття функцій (Function Coverage) – вимірює відсоток викликаних функцій або методів;
3. покриття класів (Class Coverage) – вимірює відсоток використаних класів у процесі тестування.

Кожен із цих типів покриття має свої переваги та недоліки, і вибір конкретного типу залежить від специфіки проекту, вимог до якості та доступних ресурсів. У контексті веб-застосунків особливої уваги заслуговують покриття функцій та покриття класів, оскільки вони найкраще відображають бізнес-логіку застосунку та потенційні точки відмови.

Для вимірювання покриття коду в РНР-застосунках існує кілька інструментів, серед яких найбільш популярними є Xdebug – розширення для РНР, яке окрім налагодження коду дозволяє збирати дані про покриття коду. Було обрано комбінацію Codeception та Xdebug, оскільки вона забезпечує найбільш гнучкі можливості для тестування та вимірювання покриття коду в контексті веб-застосунків. В рамках практичної частини роботи було успішно реалізовано інтеграцію інструментів вимірювання покриття коду у процес розробки веб-застосунку. Цей застосунок представляє собою платформу для публікації та управління статтями, що містить типові компоненти сучасних веб-застосунків: авторизацію, управління користувачами, CRUD-операції з контентом, пошук та фільтрацію.

Особливу увагу в рамках дослідження було приділено інтеграції процесів вимірювання покриття коду з системами безперервної інтеграції (CI/CD). Для цього було налаштовано GitLab CI/CD пайплайн, який автоматично виконує тести та генерує звіти про покриття коду. Така інтеграція дозволила автоматизувати процес контролю якості та забезпечити постійний моніторинг рівня покриття коду, що є важливою складовою сучасних практик розробки програмного забезпечення.

У ході експериментів було проведено серію тестів із різним рівнем покриття коду та проаналізовано вплив цього рівня на якість застосунку. Було виявлено залежність між рівнем покриття коду та кількістю виявлених помилок – зі збільшенням рівня покриття коду зростала кількість виявлених помилок, однак ця залежність не була лінійною. Після досягнення приблизно 80% покриття, подальше збільшення давало все менший приріст у виявленні нових помилок. Крім того, було встановлено вплив покриття коду на час регресійного тестування – високий рівень

покриття коду дозволив значно пришвидшити процес регресійного тестування при внесенні змін до коду.

На основі отриманих результатів було розроблено наступні рекомендації щодо ефективного використання метрик покриття коду в процесі розробки веб-застосунків. Рекомендується встановлювати реалістичні цілі щодо рівня покриття коду – замість прагнення до 100% покриття, рекомендується встановлювати цільові показники на рівні 70-80% для загального покриття та 85-90% для критичних компонентів.

Для досягнення максимальної ефективності рекомендується використовувати комбінацію різних типів покриття (рядків коду, класів, функцій). Доцільно інтегрувати вимірювання покриття коду в CI/CD пайплайни – це дозволить автоматизувати процес контролю якості та забезпечити постійний моніторинг рівня покриття. Покриття коду слід використовувати як один з інструментів у комплексному підході до забезпечення якості – поєднувати з іншими практиками, такими як статичний аналіз коду, навантажувальне тестування, тестування безпеки.

Відповідаючи на головне дослідницьке питання "Чи впливає високий рівень покриття коду на якість продукту?", можна стверджувати, що такий вплив є позитивним, але не вичерпним. Покриття коду слід розглядати як важливий індикатор якості, який, проте, повинен використовуватися в комплексі з іншими метриками та практиками забезпечення якості програмного забезпечення.

Список використаних джерел:

1. Everything You Need to Know About Test Coverage vs Code Coverage. URL: <https://www.perfecto.io/blog/test-coverage-vs-code-coverage> (дата звернення: 12.02.2025).
2. Smelyakov K., Prokopenko O., Chupryna A. "Object-Based Image Comparison Algorithm Development for Data Storage Management Systems" URL: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85134747140&origin=recordpage> (дата звернення: 03.03.2025).
3. Smelyakov K., Smelyakov S., Chupryna A. "Adaptive edge detection models and algorithms" URL: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85077431702&origin=recordpage> (дата звернення: 03.01.2025).
4. Medium: Demystifying the Pursuit of 100% Test Code Coverage. URL: <https://andremoniy.medium.com/demystifying-the-pursuit-of-100-test-code-coverage-77f38b9d2e41> (дата звернення: 15.12.2024).
5. Brainhub: What Is Code Coverage and How to Optimize It? URL: <https://brainhub.eu/library/measuring-code-coverage> (дата звернення: 05.02.2025).