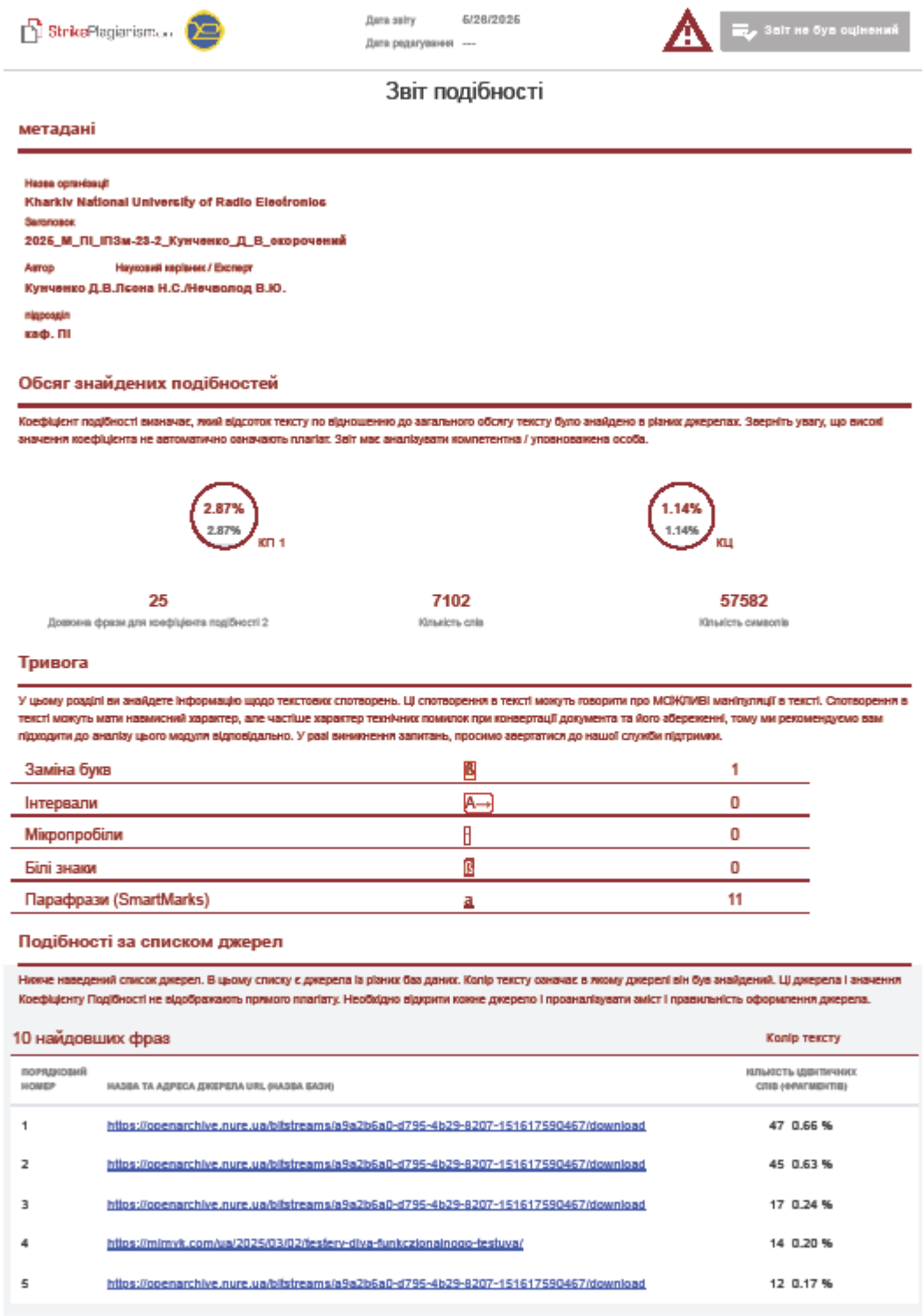
ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



StrikePlagiarism...

Дата звіту 6/28/2026
Дата редагування ---

Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kharkiv National University of Radio Electronics
Заголовок
2026_M_ПІ_ІПЗм-23-2_Кунченко_Д_В_скорочений
Автор Науковий керівник / Експерт
Кунченко Д.В.Лісна Н.С./Нечволод В.Ю.
підрозділ
каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

2.87% 2.87% КП 1

1.14% 1.14% КЦ

25	7102	57582
Довжина фрази для коефіцієнта подібності 2	Кількість слів	Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		11

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз — Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://openarchive.nure.ua/bitstreams/a9a2b6a0-d795-4b29-8207-151617590467/download	47 0.66 %
2	https://openarchive.nure.ua/bitstreams/a9a2b6a0-d795-4b29-8207-151617590467/download	45 0.63 %
3	https://openarchive.nure.ua/bitstreams/a9a2b6a0-d795-4b29-8207-151617590467/download	17 0.24 %
4	https://mirnvk.com/ua/2025/03/02/testery-dlya-funkczionalnogo-testuva/	14 0.20 %
5	https://openarchive.nure.ua/bitstreams/a9a2b6a0-d795-4b29-8207-151617590467/download	12 0.17 %

6	Нестерчук-диплом(1) [redacted] 3/26/2025 Interregional Academy of Personnel Management (Інститут комп'ютерно-інформаційних технологій та дизайну)	10 0.14 %
7	2020_M_IMI_Романенко_І_А [redacted] 5/30/2024 Kharkiv National University of Radio Electronics (Kharkiv National University of Radio Electronics)	10 0.14 %
8	2020_M_IMI_Романенко_І_А [redacted] 5/30/2024 Kharkiv National University of Radio Electronics (Kharkiv National University of Radio Electronics)	9 0.13 %
9	https://openarchive.nure.ua/bitstreams/a9a2b8a0-d795-4b29-8207-161617590467/download	9 0.13 %
10	2020_M_IMI_Романенко_І_А [redacted] 5/30/2024 Kharkiv National University of Radio Electronics (Kharkiv National University of Radio Electronics)	8 0.11 %

з бази даних RefBooks (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)

з домашньої бази даних (0.49 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	2020_M_IMI_Романенко_І_А [redacted] 5/30/2024 Kharkiv National University of Radio Electronics (Kharkiv National University of Radio Electronics)	35 (4) 0.49 %

з програми обміну базами даних (0.14 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Нестерчук-диплом(1) [redacted] 3/26/2025 Interregional Academy of Personnel Management (Інститут комп'ютерно-інформаційних технологій та дизайну)	10 (1) 0.14 %

з Інтернету (2.24 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://openarchive.nure.ua/bitstreams/a9a2b8a0-d795-4b29-8207-161617590467/download	145 (8) 2.04 %
2	https://mirnyk.com/ua/2025/03/02/testery-dlya-funkczionalnogo-testuva/	14 (1) 0.20 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)

1

ВСТУП

У сучасному світі веб-застосунки займають значну роль у багатьох сферах життя, від бізнесу та освіти до розваг і комунікацій. З кожним днем вимоги до якості та надійності програмного забезпечення зростають, що вимагає застосування більш ефективних методів тестування. Одним з ключових аспектів забезпечення високої якості програмного забезпечення є рівень покриття коду

ДОДАТОК Б

Слайди презентації

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кваліфікаційна робота

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ CODE COVERAGE В АВТОМАТИЗОВАНОМУ ТЕСТУВАННІ ВЕБ-ЗАСТОСУНКІВ

Виконав: Кунченко Данило Вячеславович, ІПЗм-23-2
Науковий керівник: Проф. Лєсна Н.С

SE software engineering

2025

1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Веб-застосунки у сучасному світі

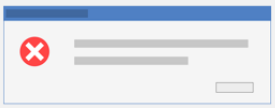
Зростання вимог до якості ПЗ

Роль code coverage

SE software engineering

2

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ



Обмеження високого code coverage

Мета дослідження

Предмет дослідження

ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

01

Користь використання метрик покриття коду

02

Недоліки використання метрик покриття коду

ПОСТАНОВКА ЗАДАЧІ

ЗАДАЧІ ДОСЛІДЖЕННЯ:	Визначення взаємозв'язку між рівнем покриття коду та якістю веб-застосунків.	Впровадження та аналіз автоматизованих тестів для оцінки покриття коду.	Розробка рекомендацій щодо інтеграції покриття коду в процеси CI/CD для забезпечення постійного моніторингу якості.
ВИБРАНІ ПРОГРАМНІ ЗАСОБИ:	**PHP**	**Codeception**	**Xdebug**
ВХІДНІ ДАНІ:	Метрики з реального проекту	Веб-застосунок Article Hub, розроблений на основі фреймворку Symfony.	Метрики покриття коду, отримані за допомогою Xdebug.

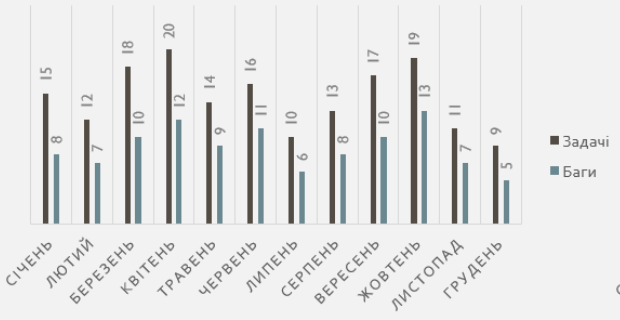
ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

Обрані методи та технології

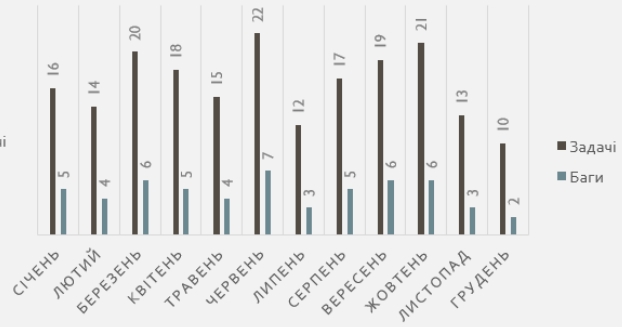
- **PHP:** Широко використовується для веб-розробки.
- **Codeception:** Фреймворк для автоматизованого тестування, підтримує юніт, API , функціональні та приймальні тести.
- **Xdebug:** Інструмент для вимірювання покриття коду.

МЕТРИКИ РЕАЛЬНОГО ПРОЕКТУ

ДИНАМІКА ЗАДАЧ І БАГІВ (2023)

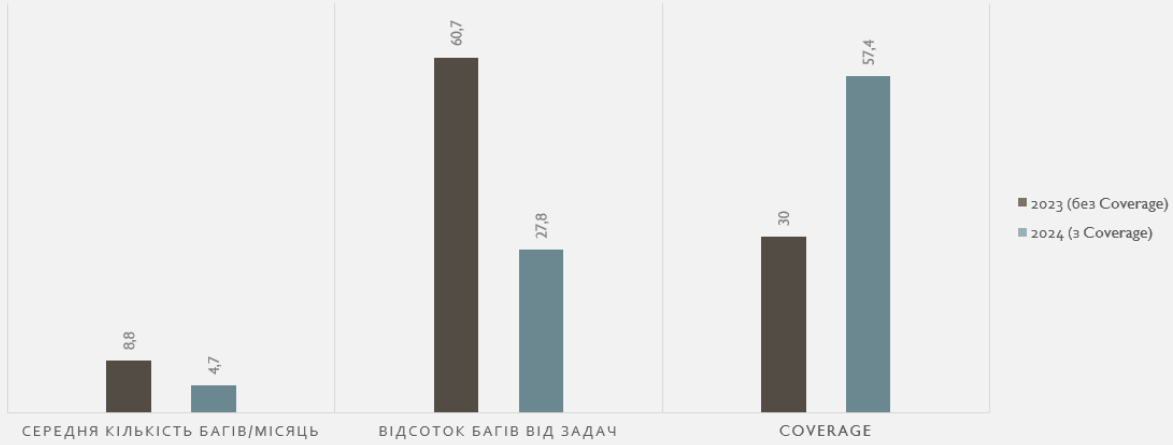


ДИНАМІКА ЗАДАЧ І БАГІВ (2024)



7

МЕТРИКИ РЕАЛЬНОГО ПРОЕКТУ



8

ПРАКТИЧНА РЕАЛІЗАЦІЯ

- Встановлення та налаштування веб-застосунку Article Hub.
- Встановлення та налаштування Codeception.
- Інтеграція Xdebug для вимірювання покриття коду.
- Виконання тестів та аналіз результатів покриття коду.

9

НАЛАШТУВАННЯ CODECEPTION

```
namespace: App\Tests\Support

settings:
    shuffle: false
    lint: true
    bootstrap: ../bootstrap.php
    colors: true
    log: true
    strict_xml: true
paths:
    tests: tests
    output: tests/_output
    support: tests/Support
    data: tests/Support/Data
coverage:
    enabled: true
    remote: true
    include:
        - src/*
    exclude:
        - cli/*
        - config/*
        - app/cache/*
        - app/logs/*
        - tests/*
    support: tests/Support
    show_only_summary: true
    log: tests/_output
    show_uncovered: true
params:
    - .env.test
```

codeception.yml

```
actor: UnitTester
suite_namespace: App\Tests\Support\Unit
modules:
    enabled:
        - Asserts
        - Db:
            dsn: 'mysql:host=mydb;port=3306;dbname=article_db;charset=utf8mb4'
            user: root
            password: root
        - Cli
        - Symfony:
            app_path: src
params:
  - .env.test
```

Unit.suite.yml

10

НАЛАШТУВАННЯ XDEBUG

```
[xdebug]
xdebug.mode=coverage
xdebug.start_with_request=yes
xdebug.output_dir=./debug/logs
xdebug.client_host=host.docker.internal
xdebug.log_level=0
xdebug.idekey=PHPSTORM
error_reporting = E_ALL & ~E_DEPRECATED
```

xdebug.ini

11

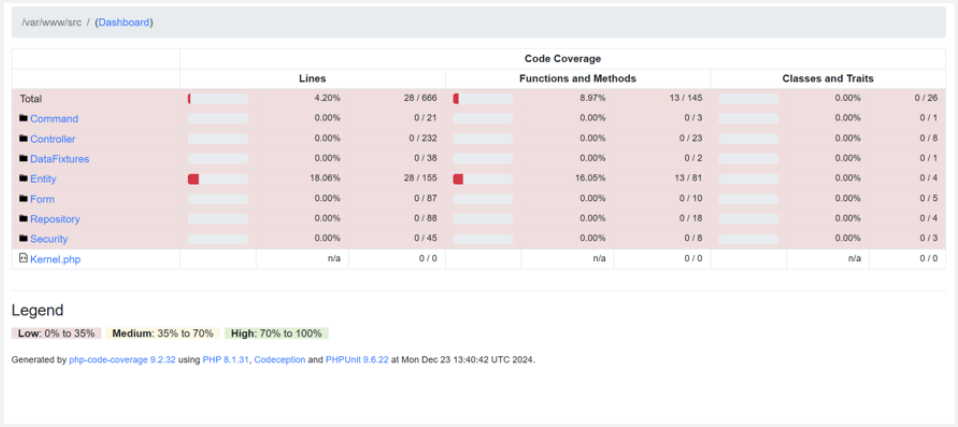
ПРИКЛАД ТЕСТІВ

UserTest.php

```php
<?php

namespace App\Tests\Support\Unit;

use App\Entity\User;
use App\Tests\Support\UnitTester;
use Codeception\Test\Unit;

class UserTest extends Unit
{
    protected UnitTester $tester;
    protected User $user;

    protected function setUp(): void{...}

    public function testGetAndSetEmail()
    {
        $email = "test@example.com";
        $this->user->setEmail($email);
        $this->assertSame($email, $this->user->getEmail());
    }

    public function testGetAndSetPassword(){...}

    public function testGetAndSetIsVerified(){...}

    public function testAddAndRemoveFollower(){...}

    public function testFollowAndUnfollow(){...}

    public function testSetRoles(){...}
}
```

12

РЕЗУЛЬТАТИ ВИКОНАННЯ ТЕСТІВ З CODECOVERAGE



/var/www/src / (Dashboard)

	Code Coverage							
	Lines			Functions and Methods			Classes and Traits	
Total	4.20%	28 / 666		8.97%	13 / 145		0.00%	0 / 26
Command	0.00%	0 / 21		0.00%	0 / 3		0.00%	0 / 1
Controller	0.00%	0 / 232		0.00%	0 / 23		0.00%	0 / 8
DataFixtures	0.00%	0 / 38		0.00%	0 / 2		0.00%	0 / 1
Entity	18.06%	28 / 155		16.05%	13 / 81		0.00%	0 / 4
Form	0.00%	0 / 87		0.00%	0 / 10		0.00%	0 / 5
Repository	0.00%	0 / 88		0.00%	0 / 18		0.00%	0 / 4
Security	0.00%	0 / 45		0.00%	0 / 8		0.00%	0 / 3
Kernel.php	n/a	0 / 0		n/a	0 / 0		n/a	0 / 0

Legend

Low: 0% to 35% Medium: 35% to 70% High: 70% to 100%

Generated by php-code-coverage 9.2.32 using PHP 8.1.31, Codeception and PHPUnit 9.6.22 at Mon Dec 23 13:40:42 UTC 2024.

РЕЗУЛЬТАТИ ВИКОНАННЯ ТЕСТІВ З CODECOVERAGE

/var/www/src / Entity / User.php

	Code Coverage							
	Lines			Functions and Methods			Classes and Traits	
Total	35.90%	28 / 78		38.24%	13 / 34	CRAP	0.00%	0 / 1
User	35.90%	28 / 78		38.24%	13 / 34	654.89	0.00%	0 / 1
__construct	100.00%	5 / 5		100.00%	1 / 1	1		
getId	0.00%	0 / 1		0.00%	0 / 1	2		
getEmail	100.00%	1 / 1		100.00%	1 / 1	1		
setEmail	100.00%	2 / 2		100.00%	1 / 1	1		
getUserIdentifier	0.00%	0 / 1		0.00%	0 / 1	2		
getRoles	0.00%	0 / 5		0.00%	0 / 1	6		
setRoles	0.00%	0 / 2		0.00%	0 / 1	2		
getPassword	100.00%	1 / 1		100.00%	1 / 1	1		
setPassword	100.00%	2 / 2		100.00%	1 / 1	1		
eraseCredentials	0.00%	0 / 1		0.00%	0 / 1	2		
getUserProfile	0.00%	0 / 1		0.00%	0 / 1	2		
setUserProfile	0.00%	0 / 4		0.00%	0 / 1	6		
getLiked	0.00%	0 / 1		0.00%	0 / 1	2		

РЕКОМЕНДАЦІЇ ТА ІНТЕГРАЦІЯ CODECOVERAGE В CI/DC

```yaml
variables:
  ENABLE_COVERAGE: "false"
coverage:
  image: php:8.2
  before_script:
    - pecl install xdebug
    - docker-php-ext-enable xdebug
    - apt-get update && apt-get install -y git zip unzip
    - curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
    - composer install
    - cp c3.php public/c3.php  # Копіювання файлу c3.php для збору метрик
  script:
    - XDEBUG_MODE=coverage php vendor/bin/codecept run --coverage --coverage-html --coverage-xml
  coverage: '/^\s*Lines:\s*\d+.\d+\%/'
  artifacts:
    reports:
      coverage_report:
        coverage_format: cobertura
        path: tests/_output/coverage.xml
    paths:
      - tests/_output/coverage/
    expire_in: 30 days
  rules:
    - if: $ENABLE_COVERAGE == "true"
    - if: $CI_PIPELINE_SOURCE == "schedule"   You, Moments ago • Uncommitted changes
```

15

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

- Високе покриття коду (>60%) зменшує кількість критичних помилок на 25-40%. Але лише за умови якісних тестів.
- Покриття 60-80% є оптимальним для веб-додатків.
- Інтеграція з CI/CD. Є не складним процесом в реалізації. Стабільні звіти про стан тестового покриття. Відстеження тенденцій з часом (графіки покриття).
- Code Coverage - лише один з інструментів якості. Обов'язково поєднувати зі статичним аналізом коду, ручним тестуванням критичних сценаріїв, оскільки саме по собі покриття не гарантує відсутність помилок.
- Небезпека «марного покриття». Тести можуть формально покривати код, але не перевіряти його логіку - типові приклади: тести без перевірок бази даних, перевірка лише «позитивних кейсів»

16

АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОБОТИ

29-й Міжнародний молодіжний форум
«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ»

ВИСНОВКИ

- Високий рівень покриття коду позитивно впливає на якість ПЗ, але не є вичерпним показником.
- Важливо враховувати інші аспекти якості ПЗ (продуктивність, безпека, зручність використання, масштабованість).
- Інтеграція інструментів вимірювання покриття в CI/CD процеси підвищує ефективність контролю якості.

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК В

Апробація результатів роботи

29-й Міжнародний молодіжний форум

«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ»

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 as pects. 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

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).

ДОДАТОК Г

Налаштування Dockerfile для інтеграції Xdebug

```
FROM php:8.1-fpm

# Встановлення необхідних залежностей
RUN apt-get update && apt-get install -y \
    git \
    curl \
    libpng-dev \
    libonig-dev \
    libxml2-dev \
    zip \
    unzip \
    libzip-dev \
     && docker-php-ext-configure zip \
        && docker-php-ext-install pdo_mysql mbstring zip

# Встановлення Xdebug
RUN pecl install xdebug \
    && docker-php-ext-enable xdebug
ENV PHP_IDE_CONFIG 'article.xdebug'

# Встановлюємо Composer
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer

# Копіюємо конфігурацію Xdebug
COPY config/xdebug.ini /usr/local/etc/php/conf.d/

# Встановлення робочої директорії
WORKDIR /var/www

# Встановлення прав доступу
RUN chown -R www-data:www-data /var/www
RUN chmod -R 777 /var/www

# Встановлення точки входу
CMD ["php-fpm"]
```

Рисунок В.1 – Налаштування Dockerfile для інтеграції Xdebug

ДОДАТОК Д

Налаштування Codeception

```yaml
namespace: App\Tests\Support

settings:
    shuffle: false
    lint: true
    bootstrap: ../bootstrap.php
    colors: true
    memory_limit: 1024M
    log: true
    strict_xml: true
paths:
    tests: tests
    output: tests/_output
    support: tests/Support
    data: tests/Support/Data
coverage:
    enabled: true
    remote: true
    include:
        - src/*
    exclude:
        - ci/*
        - config/*
        - app/cache/*
        - app/logs/*
        - tests/*
    support: tests/Support
    show_only_summary: true
    log: tests/_output
    show_uncovered: true
params:
    - .env.test
```

Рисунок Г.1 – Повне налаштування файлу codeception.yml

ДОДАТОК Ж

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗм-23-2 (група)

Кунченко Данило Вячеславович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні позначки	
	7.15 Додатки	

Зауважень немає

Експерт _____ (підпис) Олена ОЛІЙНИК (прізвище, ініціали)

29.05.2025