

МЕТРИКИ ТЕСТУВАННЯ ЯК ІНСТРУМЕНТ ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вінницький національний технічний університет

Анотація

У роботі розглядаються основні метрики тестування програмного забезпечення як кількісні інструменти оцінювання якості продукту та ефективності процесу тестування. Аналізуються ключові показники: покриття тестами, щільність дефектів, коефіцієнт успішного виконання тестів та швидкість обробки помилок. Обґрунтовується важливість систематичного використання метрик для виявлення слабких місць у тестуванні, прогнозування кількості не знайдених дефектів та прийняття обґрунтованих рішень щодо якості програмного продукту.

Ключові слова: метрики тестування, покриття тестами, щільність дефектів, якість програмного забезпечення, QA, тестовий процес.

Abstract

This paper examines the main software testing metrics as quantitative tools for assessing product quality and the efficiency of the testing process. Key indicators are analyzed: test coverage, defect density, test pass rate, and bug processing speed. The importance of systematically using metrics to identify weaknesses in testing, predict the number of undetected defects, and make informed decisions about software product quality is substantiated.

Keywords: testing metrics, test coverage, defect density, software quality, QA, test process.

Вступ

Забезпечення якості програмного забезпечення є одним із ключових завдань сучасної розробки. Проте якість – поняття багатовимірне, і для її об'єктивного оцінювання необхідні конкретні кількісні показники. Метрики тестування – це набір вимірюваних характеристик, які дозволяють оцінити покриття коду продукту тестами, спрогнозувати кількість не знайдених дефектів та оцінити характеристики тестованої системи [1]. Без їх систематичного застосування команда не має чіткого уявлення про стан якості продукту, що може призвести до випуску програмного забезпечення з критичними дефектами або до надмірних витрат на «зайве» тестування [1]. Таким чином, правильний вибір та регулярний аналіз метрик є необхідною умовою ефективного тестового процесу.

Результати дослідження

Одним із найважливіших показників є покриття тестами. Ця метрика являє собою щільність охоплення тестами виконуваного коду програми або вимог до неї: чим більше тестів написано, тим вищим є рівень покриття [2]. Розрізняють кілька підходів до оцінки покриття: покриття вимог (Requirements Coverage), покриття рядків коду (Statement Coverage), покриття гілок (Branch Coverage) та покриття шляхів (Path Coverage) [3]. Покриття гілок є надійнішим за покриття рядків, оскільки вказує, скільки гілок потоку управління було протестовано, проте навіть стовідсоткове покриття не гарантує відсутності помилок [3]. Важливо розуміти, що повне 100% тестове покриття практично недосяжне для реального програмного забезпечення [4].

Не менш важливою метрикою є щільність дефектів – кількість знайдених помилок відносно розміру програми. Програма, що тестується, може оцінюватися на основі підрахунку та класифікації знайдених дефектів, а для кожного класу визначається відношення між кількістю відповідних дефектів і розміром програми в термінах обраних метрик [4]. Цей показник дозволяє порівнювати якість різних модулів між собою та виявляти найбільш проблемні ділянки коду, на які варто спрямувати додаткові зусилля тестувальників. Окремо відстежується коефіцієнт успішного виконання тестів, кількість знайдених дефектів та швидкість тестування – їх аналіз дає змогу оптимізувати процес тестування та покращити якість продукту [5].

Серед практично важливих показників виділяють також швидкість обробки помилок – здатність команди тестування швидко реагувати на виявлені баги, передавати інформацію розробникам і контролювати процес виправлення [6]. Висока швидкість опрацювання помилок прискорює цикл розроблення та дозволяє уникати накопичення технічного боргу. Поряд із цим відстежуються покриття конкретних функціональних і нефункціональних вимог, частота виявлення дефектів та рівень задоволеності користувача кінцевим продуктом [6]. Вибір конкретного набору метрик залежить від цілей тестування та потреб проєкту, і їх важливо регулярно переглядати та адаптувати відповідно до змін у стратегії розробки.

Висновок

У результаті дослідження встановлено, що метрики тестування є ефективним інструментом об'єктивного оцінювання якості програмного забезпечення та ефективності роботи команди QA. Систематичне відстеження таких показників, як покриття тестами, щільність дефектів і швидкість їх обробки, дає змогу виявити слабкі зони у тестовому покритті, пріоритизувати зусилля тестувальників і приймати обґрунтовані рішення щодо готовності продукту до релізу. Важливо усвідомлювати, що жодна окремо взята метрика не є вичерпною – лише комплексний підхід до їх аналізу забезпечує повноцінне уявлення про стан якості продукту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Метрики та критерії тестування. Ternopil National Technical University. [Електронний ресурс] – Режим доступу: https://elartu.tntu.edu.ua/bitstream/123456789/12682/2/Conf_2015v1_Pysarenko_A-Metrics_and_criteria_118-119.pdf (дата звернення: 03.06.2026).
2. Тестове покриття. QATestLab Training Center. [Електронний ресурс] – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/requirements-coverage/> (дата звернення: 03.06.2026).
3. Метрики покриття й глибини тестування. Луцький національний технічний університет. [Електронний ресурс] – Режим доступу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/Тестування/page12.html (дата звернення: 03.06.2026).
4. Метрики покриття й глибини тестування (тема 9). ЛНТУ. [Електронний ресурс] – Режим доступу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/Тестування/page12.html (дата звернення: 03.06.2026).
5. Метрики вимірювання ефективності процесів тестування програмних продуктів. Донецький національний університет. [Електронний ресурс] – Режим доступу: <https://jpasmd.donnu.edu.ua/article/view/16765> (дата звернення: 03.06.2026).
6. Метрики, які дійсно важливі для тестування. FoxMinded. [Електронний ресурс] – Режим доступу: <https://foxminded.ua/yak-otsinyty-robotu-testuvalnyka/> (дата звернення: 03.06.2026).

Петлюк Альона Григорівна – студентка групи ЗАКІТР-24б, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: alena.petluk1@gmail.com

Мацішина Олександра Романівна – студентка групи ЗАКІТР-24б, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: o.matsyshyna2024@pu.org.ua

Богач Ілона Віталіївна – к.т.н., доцент, професор кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com

Petlyuk Alyona Hryhorivna – student of group 3ACITR-24b, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: alena.petluk1@gmail.com

Matsyshyna Oleksandra Romanivna – student of group 3ACITR-24b, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: o.matsyshyna2024@pu.org.ua

Bogach Ilona Vitaliivna – PhD, Associate Professor of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com