

УДК 004.42:004.7

Стремецький П. - ст. гр. СНм-61

Тернопільський національний технічний університет імені Івана Пулюя

ІНСТРУМЕНТИ ТА ПРАКТИКИ ДЛЯ АНАЛІЗУ ЯКОСТІ ВИХІДНОГО КОДУ JAVA- ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Науковий керівник: д.е.н., професор Матійчук Л.П.

Stremetsky P.

Ternopil Ivan Puluj National Technical University

TOOLS AND PRACTICES FOR ANALYZING THE QUALITY OF JAVA SOFTWARE SOURCE CODE

Supervisor: Doctor of Economic Sciences, Professor Matiichuk L.P.

Проблема якості коду справді є фундаментальною для створення надійного програмного продукту. Ручні методи контролю, зокрема процедура перегляду коду, дозволяють виявити логічні помилки, недоліки архітектури та порушення стандартів програмування. Такий підхід сприяє підвищенню командної взаємодії та обміну знаннями між розробниками, проте він потребує значних часових витрат і не усуває вплив людського фактора, що може призвести до суб'єктивності оцінки.

У свою чергу, автоматизовані засоби аналізу коду забезпечують систематичний і швидкий контроль якості. Інструменти на кшталт SonarQube, PMD чи FindBugs здатні виявляти потенційні дефекти, проблеми безпеки та порушення стилю програмування ще на етапі розробки. Вони інтегруються у процес CI/CD, що дозволяє проводити перевірку коду безперервно та зменшувати ризик потрапляння критичних помилок у фінальний продукт. Автоматизація також сприяє стандартизації процесів і зменшенню технічного боргу. Найбільш ефективним є комбінований підхід, коли ручний перегляд коду доповнюється автоматизованим аналізом. Така практика забезпечує високу якість вихідного коду, підвищує його безпечність та підтримуваність, а також створює умови для довготривалої стабільності програмного продукту.

Аналіз якості вихідного коду Java- додатків охоплює кілька взаємопов'язаних напрямів, які забезпечують комплексний контроль над надійністю та підтримкою програмного продукту. Першим етапом виступає статичний аналіз, що дозволяє виявити синтаксичні помилки, порушення стилю та потенційні дефекти ще до запуску програми.

Другим аспектом є застосування метрик якості програмного забезпечення, які дають кількісну оцінку стану вихідного коду. До ключових показників належать цикломатична складність, рівень покриття тестами, кількість дублювань та коефіцієнт підтримуваності. Третім напрямом є динамічний аналіз та тестування, що включає виконання юніт-тестів, інтеграційних тестів та профілювання продуктивності. Інструменти, як-от JUnit, Mockito чи JaCoCo, дозволяють перевірити функціональність окремих модулів, оцінити рівень покриття тестами та виявити проблеми продуктивності. Поєднання статичного та динамічного аналізу створює комплексний підхід до контролю якості, забезпечуючи стабільну роботу Java- додатків у реальних умовах експлуатації та підвищуючи довіру до кінцевого продукту.