

ОПТИМІЗАЦІЯ ВНУТРІШНІХ ПРОЦЕСІВ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПЗ ШЛЯХОМ ВПРОВАДЖЕННЯ ПРАКТИКИ ДЕВ-ТЕСТУВАННЯ

Олександра Нікішина

OPTIMIZATION OF INTERNAL SOFTWARE QUALITY ASSURANCE PROCESSES BY
IMPLEMENTING DEV-TESTING PRACTICES

Oleksandra Nikishyna

*Дніпровський національний університет імені Олеся Гончара
пр. Науки, 72, м. Дніпро, , 49045, Україна*

Abstract. *The paper highlights the effectiveness of implementing dev-testing as a tool for improving software quality at the development stage. The approach helps identify configuration and integration issues before QA, reducing bug rates and enhancing team workflow. Results demonstrate the method's potential in reducing development cycle time and improving defect metrics.*

Вступ

У сучасному процесі розробки програмного забезпечення якість кінцевого продукту значною мірою залежить від своєчасного виявлення дефектів та ефективної командної взаємодії. У традиційній моделі, де контроль якості починається лише з етапу тестування, часто виникають ситуації, коли задачі повертаються до розробника через помилки, пов'язані не з логікою коду, а з конфігураційними збоями або порушенням послідовності деплою. Це призводить до затримок, перевитрат ресурсів, зниження продуктивності команди та зменшення довіри до стабільності продукту з боку користувачів і замовників.

Окрім цього, через повторне повернення задач на доопрацювання втрачається важливий час, відведений на функціональне тестування, що може призводити до зниження покриття тестами або перенесення задач на наступні ітерації. Це особливо критично у гнучких методологіях, де ритм спринтів вимагає злагодженої та швидкої взаємодії між усіма учасниками процесу.

Проблеми до впровадження дев-тестування

Велика кількість дефектів, виявлених одразу після деплою задач на тестове середовище, стали ознакою того, що існуючий підхід до внутрішньої перевірки є недостатнім. При цьому більшість помилок не стосувалася логіки програмного коду, а виникала через неправильну конфігурацію середовища, некоректні змінні, неправильний порядок міграцій або відсутність залежностей. Результатом було те, що задачі нерідко навіть не доходили до тестувальників і потребували повторної участі розробника ще до старту ручного тестування.

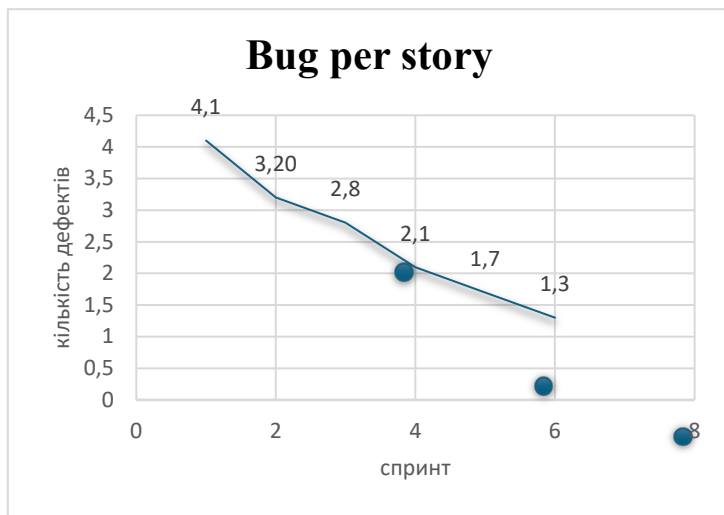
Ці помилки фіксувалися вже на етапі відкриття задачі на тестовому середовищі або при спробі виконати базову дію. Як наслідок, зростали показники метрик типу *bug per story*, що негативно впливало як на планування, так і на оцінку продуктивності команди. У QA-фахівців накопичувалися додаткові комунікаційні навантаження, оскільки необхідно було часто повертати задачі, описувати конфігураційні проблеми та чекати на повторне розгортання.

Переваги впровадження дев-тестування

Для покращення ситуації була впроваджена практика дев-тестування — внутрішньої перевірки працездатності функціональності розробником одразу після деплою. Цей підхід передбачає проходження happy path сценарію, тобто ключового користувацького шляху, який демонструє базову функціональність задачі. Така перевірка дозволяє переконатися, що зміни

були коректно розгорнуті, працюють у передбаченому контексті та не викликають очевидних конфігураційних помилок.

Це дозволило усунути чималу кількість базових проблем ще до того, як задача потрапляла до QA на тестування. Розробники почали більш відповідально ставитися до завершення задачі та брати на себе частину контролю якості, що покращило командну взаємодію. Тестувальники, у свою чергу, отримували задачі в більш стабільному стані, могли одразу приступати до глибшого функціонального або регресійного тестування. Це дозволило не тільки знизити навантаження, а й підвищити покриття сценаріїв, особливо в умовах обмеженого часу.



Метрики проекту також продемонстрували позитивну динаміку: *bug per story* знизився, зменшилась кількість повернень задач та підвищився відсоток задач, що проходять тестування з першого разу. Час, витрачений на комунікацію щодо базових дефектів, скоротився, що дало змогу сфокусувати зусилля на перевірці бізнес-логіки, неочевидних сценаріїв або нефункціональних вимог. Таким чином, дев-тестування виявилось ефективним способом зменшити технічний борг ще до етапу тестування.

Висновок

Практика дев-тестування показала себе як важливий інструмент в оптимізації процесів забезпечення якості на ранніх етапах розробки. Простота впровадження, мінімальні часові витрати та висока ефективність роблять її доцільною для широкого використання в командах, які працюють за гнучкими методологіями.

Цей підхід сприяє не лише підвищенню якості програмного забезпечення, але й покращенню внутрішньої командної взаємодії, зниженню кількості технічних обговорень щодо базових проблем та формуванню культури відповідальності за стабільність і працездатність кожної зміни, яка вноситься до коду.

Таким чином, дев-тестування є вагомим внеском у побудову надійного, зрозумілого та контрольованого процесу розробки, що відповідає сучасним вимогам до якості та швидкості реалізації програмного продукту.