



**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ПОЛТАВСЬКА ПОЛІТЕХНІКА  
ІМЕНІ ЮРІЯ КОНДРАТЮКА**

**ЗБІРНИК МАТЕРІАЛІВ**

**77-ї НАУКОВОЇ КОНФЕРЕНЦІЇ ПРОФЕСОРІВ,  
ВИКЛАДАЧІВ, НАУКОВИХ ПРАЦІВНИКІВ,  
АСПІРАНТІВ ТА СТУДЕНТІВ УНІВЕРСИТЕТУ**

**16 травня – 22 травня 2025 р.**

Застосування передових технологій блокчейну у поєднанні з ретельно розробленим UX/UI дизайном дозволило створити діючий прототип. Він має потенціал стати підґрунтям для повноцінного кінцевого продукту.

*Література:*

1. *Uniswap v2 Core Whitepaper*. URL: <https://uniswap.org/whitepaper.pdf> (дата звернення – 07.05.2025 р.)
2. *Ethereum Whitepaper* / Vitalik Buterin. URL: <https://ethereum.org/en/whitepaper> (дата звернення – 07.05.2025 р.)

**УДК 004.415**

*В.В. Загнойко, асистент  
Національний університет  
«Полтавська політехніка імені Юрія Кондратюка»*

## **РИЗИКИ І ПОМИЛКИ ПРИ ВПРОВАДЖЕННІ АВТОМАТИЧНОГО ТЕСТУВАННЯ В РЕАЛЬНОМУ ПРОЄКТІ**

Автоматичне тестування є критично важливою частиною життєвого циклу програмного забезпечення, особливо в умовах сучасної розробки, що орієнтована на Agile-методології та безперервну інтеграцію і доставку (CI/CD). Його головна мета — забезпечити швидку зворотну реакцію на зміни в коді, зменшити кількість помилок у продуктиві та оптимізувати ресурси тестувальників. Проте, незважаючи на очевидні переваги, автоматизація тестування часто стикається з рядом складнощів під час впровадження в реальних проєктах.

Однією з найпоширеніших помилок є початок автоматизації без чітко сформованої стратегії. Часто команда просто починає «писати тести», не маючи відповіді на базові питання:

- Що саме має бути автоматизовано?
- У якій послідовності?
- Який рівень покриття є цільовим?
- Як тести будуть інтегровані у загальний процес розробки?

В результаті створюється велика кількість неузгоджених, дубльованих або надмірно складних тестів, які важко підтримувати. Іншою проблемою є неправильний вибір інструментів. Інструменти для автоматизації мають відповідати технологічному стеку, типу продукту, а також рівню компетенції команди. Наприклад, вибір Selenium WebDriver для

тестування мобільного застосунку є типовою помилкою, як і використання складного фреймворку з крутою кривою навчання для невеликої команди без достатнього досвіду. У деяких випадках рішення приймаються на основі популярності, а не реальних потреб, що призводить до неефективної реалізації або повної відмови від автоматизації.

Автоматичні тести — це також код, який потребує підтримки. У реальних проєктах, де продукт постійно змінюється, тести, які не оновлюються, швидко застарівають. З'являються фальшиві позитивні або негативні результати, що знижує довіру до автоматизації. Команди починають ігнорувати звіти, що в підсумку зводить користь тестів нанівець.

До помилок також належить недооцінка початкових витрат часу та ресурсів. Автоматизація потребує інвестицій у планування, розробку інфраструктури, навчання персоналу. Часто керівництво очікує швидких результатів, не враховуючи, що повернення інвестицій настає лише після певного часу.

Комунікаційні проблеми між розробниками, тестувальниками та DevOps-фахівцями також ускладнюють впровадження. Без спільного бачення і співпраці автоматизоване тестування не стане повноцінною частиною процесу CI/CD.

Іноді автоматичне тестування впроваджується формально — для галочки. Наприклад, у документації зазначено, що тести є, але фактично вони не запускаються або не відображають реальний стан системи. Також розповсюдженою проблемою є накопичення технічного боргу — коли застарілі тести не видаляються або не оновлюються, а їх підтримка стає надто дорогою.

У підсумку, успішне впровадження автоматичного тестування потребує:

- розробки чіткої стратегії автоматизації;
- правильного вибору інструментів;
- навчання та мотивації команди;
- тісної інтеграції з CI/CD;
- регулярного перегляду і підтримки тестової бази;
- постійної комунікації між всіма учасниками проєкту.

Лише комплексний підхід дозволяє автоматизації стати не просто додатковим інструментом, а повноцінним елементом, що гарантує якість і стабільність програмного забезпечення.

#### *Література:*

*I. Garousi, V., Felderer, M., & Mäntylä, M. V.. The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature. Information and Software Technology, 2019, 183–193c.*