

# АНАЛІЗ ПРОБЛЕМ ВПРОВАДЖЕННЯ CI/CD PIPELINE ТА ШЛЯХІВ ЇХ ВИРІШЕННЯ

Вінницький національний технічний університет

## Анотація

У роботі досліджено роль CI/CD pipeline у побудові сучасної архітектури програмного забезпечення. Розглянуто основні принципи організації процесів безперервної інтеграції та доставки, а також їх вплив на якість, стабільність і швидкість розробки програмних систем. Проведено аналіз типових проблем, що виникають при впровадженні CI/CD pipeline, зокрема недостатності стратегій тестування, відсутності моніторингу та механізмів зворотного зв'язку, а також неналежного резервного копіювання даних і конфігурацій. На основі проведеного аналізу запропоновано заходи щодо їх усунення, що сприяють підвищенню ефективності функціонування програмних систем та надійності їх архітектури.

**Ключові слова:** CI/CD pipeline, безперервна інтеграція, безперервна доставка, архітектура програмного забезпечення, автоматизоване тестування, моніторинг, резервне копіювання, надійність системи, DevOps.

## Abstract

This article examines the role of the CI/CD pipeline in building modern software architecture. It discusses the key principles of organizing continuous integration and delivery processes, as well as their impact on the quality, stability, and speed of software system development. An analysis is conducted of typical problems arising during the implementation of a CI/CD pipeline, including inadequate testing strategies, a lack of monitoring and feedback mechanisms, as well as improper backup of data and configurations. Based on this analysis, measures are proposed to address these issues, which contribute to improving the operational efficiency of software systems and the reliability of their architecture.

**Keywords:** CI/CD pipeline, continuous integration, continuous delivery, software architecture, automated testing, monitoring, backup, system reliability, DevOps.

## Вступ

У сучасних умовах розвитку інформаційних технологій зростають вимоги до швидкості розробки, якості та надійності програмного забезпечення. Традиційні підходи до розгортання і тестування часто не забезпечують достатньої гнучкості та оперативності змін. У зв'язку з цим актуальним стає використання CI/CD (безперервної інтеграції та доставки), що дозволяє автоматизувати ключові етапи життєвого циклу ПЗ. CI/CD впливає не лише на процес розробки, а й на архітектуру систем, підходи до тестування та супроводу [1]. Водночас неправильна організація CI/CD може спричинити проблеми, що знижують стабільність і ефективність роботи системи.

Метою роботи є дослідження ролі CI/CD pipeline у побудові сучасної архітектури програмного забезпечення та аналіз типових проблем його впровадження і шляхів їх вирішення.

## Результати дослідження

CI/CD – це процес розробки програмного забезпечення, що поєднує безперервну інтеграцію та доставку і дозволяє автоматизовано впроваджувати зміни в код. У DevOps він використовується для швидкого збирання та тестування застосунків, що знижує ризики та підвищує якість, а також дозволяє вносити зміни без порушення роботи продукційного середовища [2].

Суть підходу полягає в тому, що код розробляється невеликими частинами (pipeline), які проходять автоматичне тестування перед об'єднанням у головну гілку для подальшого розгортання [3]. Це зменшує кількість людських помилок завдяки автоматизації повторюваних процесів, зокрема тестування в staging (тестовому середовищі) або QA середовищах перед релізом у production (робоче середовище).

CI/CD передбачає автоматизацію процесів на всіх етапах життєвого циклу: від інтеграції змін у код з автоматичним тестуванням до доставки та розгортання. CI забезпечує регулярне об'єднання змін у код і їх перевірку, тоді як CD відповідає за автоматизоване розгортання перевірених змін у робоче середовище, що зменшує кількість помилок і ризики під час випуску нових версій [1].

У процесі аналізу сучасних підходів до впровадження CI/CD pipeline [1; 4] було виявлено низку типових проблем, що впливають на стабільність, надійність та ефективність програмних систем. Узагальнення цих проблем, їх наслідків та можливих шляхів усунення наведено в таблиці 1.

Таблиця 1 – Типові проблеми впровадження CI/CD та шляхи їх усунення

| Проблема   | Наслідки   | Шляхи усунення   |
|--|--|--|
| Недостатньо розвинена стратегія тестування               | Виникнення проблем після релізу, погіршення користувацького досвіду    | Розширення тестування (модульне, інтеграційне, end-to-end, навантажувальне, без пекове)                        |
| Відсутність моніторингу та механізмів зворотного зв'язку | Збільшення часу виявлення та усунення проблем, зниження якості сервісу | Впровадження систем моніторингу та сповіщення, збір метрик продуктивності та використання                      |
| Відсутність резервного копіювання даних і конфігурацій   | Втрата даних, простої системи, складне відновлення                     | Налаштування резервного копіювання та процедур відновлення, зберігання конфігурацій у системах контролю версій |

Узагальнення, наведене в таблиці 1, демонструє, що проблеми впровадження CI/CD pipeline мають системний характер і повторюються в різних проєктах незалежно від масштабу чи предметної області. Найчастіше вони пов'язані з обмеженістю ресурсів тестових середовищ, складністю інтеграції інструментів безпеки та недостатньою зрілістю процесів автоматизації. Окремо варто відзначити, що значна частина труднощів виникає не стільки на рівні інструментів, скільки на рівні організації процесів розробки та взаємодії між етапами життєвого циклу ПЗ [2]. Через це навіть технічно правильно налаштований pipeline може втрачати ефективність у реальних умовах використання.

У результаті можна зробити висновок, що впровадження CI/CD потребує комплексного підходу, який поєднує технічну автоматизацію, належну організацію середовища та достатній рівень підготовки команди.

### Висновки

CI/CD pipeline є базовим підходом до автоматизації процесів інтеграції, тестування та розгортання програмного забезпечення, що забезпечує підвищення швидкості випуску змін і стабільності систем. Основними проблемами його впровадження є недостатній рівень автоматизованого тестування, відсутність моніторингу та зворотного зв'язку, а також недоліки в організації резервного копіювання даних і конфігурацій. Зазначені проблеми мають системний характер і виникають як на технічному, так і на організаційному рівнях.

Підвищення ефективності CI/CD досягається за рахунок розширення тестових підходів, впровадження механізмів моніторингу та забезпечення надійного резервного копіювання і керування конфігураціями.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Singh V. Developing a CI/CD pipeline with GitLab, Bachelor's Thesis: Turku University of Applied Sciences, 2022. 56 p.
2. What is CI/CD? (Continuous Integration and Continuous Delivery) [Електронний ресурс]. 2025. Режим доступу: <https://visuresolutions.com/alm-guide/ci-cd/> (дата звернення: 27.04.2026).
3. Ahir K. Design and Implementation of Automated CI/CD Pipelines in DevOps, Asian Journal of Computer Science Engineering, 2026. 8 p.
4. Parmar Tarun. Implementing CI/CD in Data Engineering: Streamlining Data Pipelines for Reliable and Scalable Solutions. International Journal of Innovative Research in Multidisciplinary Science (IJIRMP), 2025. 8 p.

**Скидан Тетяна Миколаївна** – студентка групи КІТС-25м, Факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, м. Вінниця, e-mail: tanaskidan1@gmail.com

Науковий керівник: **Зоря Ірина Сергіївна** – ас. каф. Менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, м. Вінниця, e-mail: ira.zoria@vntu.edu.ua

**Skydan Tetyana M.** – student of group CITS-25m, Faculty of Management and Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: tanaskidan1@gmail.com

Supervisor: **Zoria Iryna S.** – assistant of the Department of Management and Security of Information Systems, Vinnytsia National Technical University, Vinnytsia, e-mail: ira.zoria@vntu.edu.ua