

УДК 004.5

В.А. Готович, к.т.н., А.В. Мачужак

Тернопільський національний технічний університет імені Івана Пулюя

ЗАСТОСУВАННЯ МЕТОДОЛОГІЇ CI/CD ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ТЕСТУВАННЯ ТА РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

V.A. Hotovych, Ph.D., A.V. Machuzhak

APPLICATION OF CI/CD METHODOLOGY FOR AUTOMATION OF SOFTWARE TESTING AND DEPLOYMENT PROCESSES

Для успішності технологічного процесу розробки програмного забезпечення (ПЗ) важливим є вирішення завдань підтримки якості як програмного коду зокрема, так і програмного продукту в цілому, а також постачання чергових версій програмного продукту до споживача. На практиці вирішення цих задач відбувається автоматизованим чином шляхом застосування методології CI/CD (англ. Continuous Integration, Continuous Delivery) [1]. Дана методологія складається з двох етапів [2]:

1. Безперервна інтеграція (CI). Це практика інтегрування програмного коду в основну гілку репозиторію при автоматичному тестуванні якості коду. Зокрема, дана практика вирішує проблему наявності занадто великої кількості гілок в системі контролю версій в процесі розробки, які можуть конфліктувати між собою.

2. Безперервне постачання (CD). Це практика безперервного постачання протестованого на попередньому етапі програмного коду (нових версій ПЗ) до кінцевого користувача. Це етап автоматизації, на якому програмне забезпечення розгортається в одному або декількох середовищах виконання. Команди розробників зазвичай використовують декілька окремих середовищ для тестування програмного забезпечення.

В даній роботі наведено результати практичного застосування методології CI/CD для автоматизації процесів тестування та розгортання програмного забезпечення при роботі над конкретним програмним продуктом. Зокрема, реалізовано 2 конвеєри CI/CD [3], що схематично показано на рис. 1.

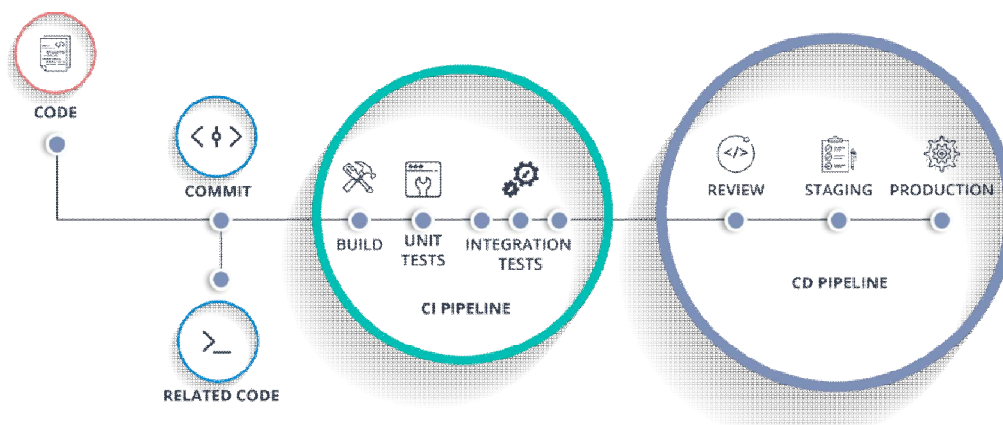


Рисунок 1. Конвеєри CI/CD

На першому етапі (постійної інтеграції), використовувався сервіс GitHub Actions, який надає можливості виконувати різні дії над програмним кодом в репозиторії проекту. Цей сервіс специфічний для GitHub. Аналогічні до GitHub сервіси для репозиторіїв надають відповідні можливості. Зокрема, для роботи з GitHub Actions створено конфігураційний yml-файл, в якому задано налаштування щодо того, при яких умовах слід виконувати дії, в якому середовищі, послідовність кроків тощо. Наприклад,

при відкритті пул реквеста (англ. pull request) чи нового коміта (англ. commit) запуститься програмне середовище з операційною системою Ubuntu та буде виконано необхідні кроки по інсталяції залежностей, збиранні коду, запуску тестів тощо.

На першому етапі застосування методології CI/CD вирішено наступні задачі:

- автоматизація процесів збирання коду;
- тестування ПЗ;
- швидке впровадження змін в спільний репозиторій.

На етапі безперервного постачання використано інструмент Jenkins. В конфігураційному файлі за допомогою спеціальної нотації описано різні етапи, такі як збірка, тестування та розгортання. Також задаються змінні середовища, параметри, секретні ключі, сертифікати та інші налаштування.

Результатом другого етапу є конвеєр безперервної доставки з такими етапами (рис. 2):

- отримання останніх змін в програмному коді з репозиторію;
- інфраструктурні зміни (збільшити чи зменшити потужності хмарної інфраструктури);
- переміщення програмного коду в цільове обчислювальне середовище та управління змінними середовища;
- виконання тестів та згорання середовища, якщо тести не виконалися успішно;
- надання даних з журналів (логів) та сповіщення про стан постачання ПЗ;
- розгортання нових змін в цільовому середовищі;
- сповіщення про завершення етапу.

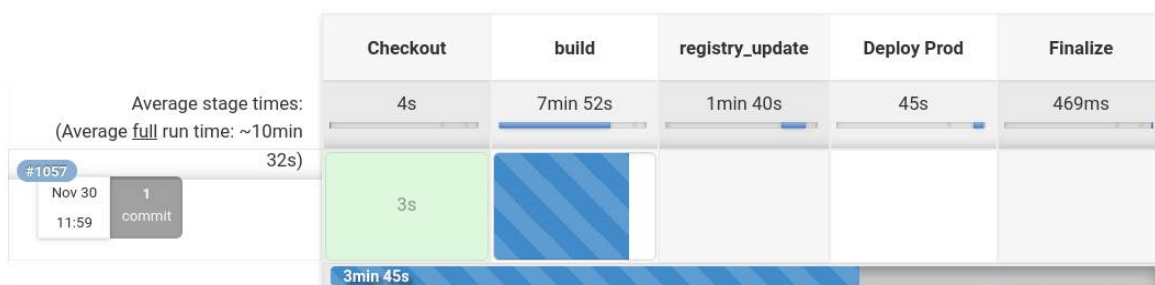


Рисунок 2. Процес постачання нових версій ПЗ

Серед конкретних переваг, яких було досягнуто в результаті впровадження методології CI/CD, можна назвати:

- Підвищення продуктивності роботи команди розробників;
- Автоматичні збірки на зміни в коді, що дозволяє підготувати та протестувати нову версію програмного забезпечення;
- Автоматичне розгортання нових версій ПЗ.

Перспективами подальшого дослідження є тематика оркестрації контейнерів – керування контейнерами у великих масштабах за допомогою технології Kubernetes.

Література

1. CI/CD. URL: <https://en.wikipedia.org/wiki/CI/CD> (дата звернення: 28.11.2022).
2. What is CI/CD? Continuous integration and continuous delivery explained. URL: <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html> (дата звернення: 28.11.2022).
3. ROSSEL, Sander. Continuous Integration, Delivery, and Deployment: Reliable and faster software releases with automating builds, tests, and deployment. Packt Publishing Ltd, 2017.