

УДК 004.415.3:681.6

**Мионов Ю.А., студент гр. 123м-21-1**

**Науковий керівник: Ткаченко С.М., к.т.н., доцент кафедри ІТКІ**

*(Національний технічний університет "Дніпровська політехніка", Дніпро, Україна)*

## **ІНТЕГРАЦІЯ РОЗДІЛЕНИХ СИСТЕМ CONTINUOUS INTEGRATION ТА CONTINUOUS DELIVERY НА ОСНОВІ СЕРВІСУ ЧЕРГИ ПОВІДОМЛЕНЬ**

Запровадження систем Continuous Integration (CI) та Continuous Delivery (CD) мало чи не найбільший вплив на підвищення ефективності розробки програмного забезпечення за останні десятиліття[1]. Автоматизуючи збірку та базове тестування програмного коду, процеси Continuous Integration дозволили розробникам швидко отримувати інформацію про те, як їхні зміни впливають на кінцевий продукт; а за рахунок процесів Continuous Delivery розгортання нового програмного коду стало займати лічені хвилини.

Однією з найуживаніших функцій систем CI/CD була можливість створення залежних задач, що виконувалися послідовно. Задачі Continuous Delivery вставали у певну чергу і очікували успішного завершення задач Continuous Integration, за рахунок чого дотримувалась послідовність та інтеграція цих процесів. Однак на сьогодні розвиток провайдерів систем контролю версій призвів до переміщення процесів Continuous Integration ближче до місце зберігання коду – GitHub Actions, GitLab CI, Atlassian Bamboo фактично інтегровані із репозиторіями, у яких зберігається початковий код. Це призвело до відділення процесів Continuous Integration від Continuous Delivery, і втрати інтеграції між цими системами, що ускладнило процеси розробки, тестування і розгортання програмного забезпечення.

Частково цю проблему намагалися вирішити інженери JetBrains, які для власної платформи TeamCity передбачили[2] технічну можливість вивільнення обчислювальних ресурсів для очікування завершення роботи зовнішніх задач (завершення роботи задач інших систем, платформ або програм). Тим не менш, така технічна можливість є лише підготовчим етапом для вирішення проблеми інтеграції розділених систем CI/CD, але не кінцевим рішенням.

Для остаточного ж вирішення проблеми інтеграції процесів пропонується відтворити втрачену систему черг у вигляді окремого програмного сервісу черги повідомлень, що керує чергами задач Continuous Integration та Continuous Delivery і працює за принципом «публікація-підписка» (publish-subscribe)[3]. У якості «публікацій» виступатимуть задачі Continuous Integration, результати виконання яких передаються «підписникам» – системам Continuous Delivery, які мають функціональну можливість безпосередньо зчитувати публікації, або ж проміжним сервісам, адаптованим для зчитування «публікацій» і створення запитів до систем Continuous Delivery на основі даних «публікації» (тобто результатів виконання задач Continuous Integration). Взаємодія між сервісом черг повідомлень та системами Continuous Integration і Continuous Delivery має відбуватися через HTTP-запити програмних інтерфейсів API цих систем. Для спрощення архітектури такого сервісу, як це наведено на рисунку 1, слід використовувати платформи Pub/Sub (RabbitMQ, Apache Kafka, Google Pub/Sub, Amazon SQS тощо).

Запровадження сервісу черги повідомлень дозволяє налагодити послідовність виконання процесів Continuous Integration та Continuous Delivery, надає технічну можливість створення залежних задач між розділеними системами CI/CD і забезпечує інтеграцію цих систем. Крім того, сервіс має усі переваги по контролю над чергами виконання задач, включаючи додавання і виключення задач із черги, зміну

пріоритетності виконання задач чи виконання додаткового адміністративного нагляду за виконуваними задачами.

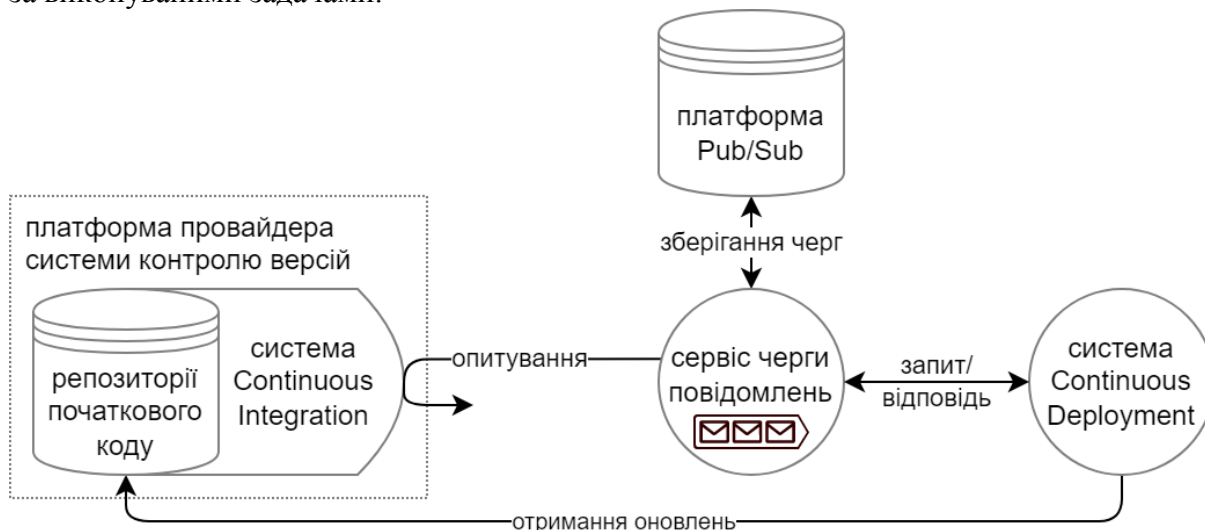


Рисунок 2 – Архітектура системи інтеграції процесів Continuous Integration та Continuous Delivery на основі сервісу черги повідомлень

Подальший розвиток запропонованого рішення полягає у запровадженні функціональної можливості по передачі мета-інформації між розділеними системами, зберіганні історії виконуваних задач, розширенні списку підтримуваних систем CI/CD і створенні універсального програмного інтерфейсу для мультиплатформної підтримки розділених процесів Continuous Integration та Continuous Delivery.

### ПЕРЕЛІК ПОСИЛАНЬ

1. Sten Pittet. Continuous integration vs delivery vs deployment [Електронний ресурс] – Atlassian, 2022. – Режим доступу до ресурсу: <https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment> (дата звернення: 03.11.22).
2. Agentless Build Step [Електронний ресурс] – TeamCity On-Premises Documentation, 2022. – Режим доступу до ресурсу: <https://www.jetbrains.com/help/teamcity/agentless-build-step.html> (дата звернення: 04.11.22).
3. Publisher-Subscriber pattern [Електронний ресурс] – Microsoft Learn Portal, 2022. – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/azure/architecture/patterns/publisher-subscriber> (дата звернення: 03.11.22).