

УДК 004.514

DOI: 10.30977/BUL.2219-5548.2022.96.0.30

ВИБІР ІНСТРУМЕНТУ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Гризун Л. Е., Скорін Ю. І., Деточенко І. М.

Харківський національний економічний університет ім. С. Кузнеця

Анотація. У роботі проаналізовано можливості основних інструментів безперервної інтеграції для автоматизованого тестування та визначено критерії їхнього оцінювання. На основі визначених критеріїв розроблено модель оцінювання зазначених інструментів за допомогою СППР та використано її для вибору відповідного інструменту. Рекомендований СППР інструмент безперервної інтеграції CircleCI було запроваджено на підприємстві для тестування реального програмного продукту. Отримано та проаналізовано результати використання моделі оцінювання.

Ключові слова: інструменти безперервної інтеграції, автоматизоване тестування ПЗ, модель оцінювання, система підтримки прийняття рішень.

Вступ

Автоматизоване тестування сьогодні є невід'ємним складником процесу розроблення програмного забезпечення. Під автоматизованим тестуванням розуміють залучення до тестування програмних продуктів спеціальних інструментів та технологій з метою підвищення якості ПЗ.

У практиці сучасних компаній існує проблема впровадження автоматизованого тестування, що пов'язано із недостатністю засобів для оцінювання та вибору відповідних інструментів для реалізації цього процесу.

Автоматизоване тестування безпосередньо пов'язано із безперервною інтеграцією (continuous integration / CI), яка полягає у виконанні частих автоматизованих об'єднань складників проекту для швидшого виявлення та вирішення інтеграційних проблем. Кожний коміт у репозиторії з кодом має перевірятися автоматично, а результати повідомлятися розробникам. CI/CD дозволяє розв'язати проблеми, які інтеграція нового коду може спричинити для команд розробників.

Відповідно до джерел [3; 4], безперервна інтеграція / безперервна доставка (CI/CD) є методом доставки програм клієнтам шляхом впровадження автоматизації на етапах розроблення застосунків. Постійне розгортання за допомогою автоматизації вирішує проблему перевантаження операційних команд ручними процесами, які уповільнюють доставку застосунків. Вибір коректного інструменту безперервної інтеграції для автоматизованого тестування є запорукою ефективної безперервної інтеграції.

Аналіз публікацій

Аналіз спеціалізованих джерел засвідчує, що проблеми автоматизованого тестування програмного забезпечення вивчаються у безпосередньому зв'язку із завданнями безперервної інтеграції, яка передбачає, що всі, хто бере участь у проекті розробки ПЗ, регулярно публікують свої зміни в кодовій базі та перевіряють, чи працює код належним чином після кожної зміни. Безперервна інтеграція вважається дослідниками ключовим елементом DevOps-підходу до розроблення та випуску програмного забезпечення, який сприяє спільній роботі, автоматизації та швидкому отриманню зворотного зв'язку [3; 4; 6].

Основними елементами безперервної інтеграції є [6; 7]: джерело або система контролю версій з єдиною кодовою базою, зокрема файли вихідного коду, бібліотеки, конфігураційні файли та скрипти; автоматизовані білд-скрипти; автоматизовані тести; інфраструктура для складання та тестів.

Щоб усі учасники проекту працювали з однаковим кодом, вони мають використовувати один репозиторій та регулярно публікувати свої зміни. Наступним кроком після відправлення змін є проведення автоматизованих тестів для перевірки поведінки коду. Автоматизація цього процесу є невід'ємною частиною безперервної інтеграції. Якщо тестування здійснюється вручну, це забирає багато часу і може призвести до помилок. Унаслідок щоденна інтеграція змін втрачає сенс. Конкретні інструменти складання та фреймворки тестування залежать від робочої мови програмування.

Після налаштування скриптів та тестів необхідно контролювати та за необхідності

оновлювати процес. Це передбачає додавання автоматизованих тестів із появою будь-яких нових функцій, усунення збоїв та відстеження продуктивності.

Концепція безперервної інтеграції стала вирішенням проблем, з якими зіткнулися команди, коли намагалися звести до купи роботу безлічі розробників, щоб створити цілісний продукт після тривалого періоду написання коду. Процес інтеграції незалежно розроблених модулів міг призвести до істотних затримок у роботі над проектом, оскільки припущення та залежності виявлялися через тривалий час після написання коду та часто вимагали його суттєвого перероблення [10; 11].

У цьому контексті в літературі й на практиці розглядається також безперервне тестування як процес виконання автоматизованих тестів як частини конвеєра постачання програмного забезпечення з метою якомога швидшого отримання зворотного зв'язку щодо бізнес-ризиків, пов'язаних із випуском програмного забезпечення [1; 2]. Серед ключових характеристик безперервного тестування дослідники виокремлюють такі: (1) тестування є вбудованим у процес розроблення, а не буде відкладено до фінальних етапів; (2) забезпечує ефективний зворотний зв'язок, відповідний для кожного етапу; (3) передбачає постійний перегляд і оптимізацію набору тестів для усунення надмірності та максимального покриття бізнес-ризиків; (4) end-to-end-тести, що оцінюють роботу кінцевого користувача з усіма пов'язаними технологіями (фронт-енд і бек-енд).

Здійснення безперервного автоматизованого тестування стає можливим завдяки спеціальним інструментам безперервної інтеграції. Аналіз теорії та практики засвідчує, що вибір адекватного інструменту безперервної інтеграції для підтримки тестування, який буде відповідати усім вимогам бізнесу та клієнта, залишається проблемою. Відповідно до джерел, складним завданням також є пошук та поєднання інструментів, сумісних між собою, з метою створення функціонального конвеєра. Для того, щоб повністю впровадити конвеєр безперервного розгортання, потрібно багато інструментів (автоматизована система складання, тестування, розгортання тощо). Проблема, що виникає внаслідок комбінації інструментів, полягає в їхній конфігурації. Більшість проектів потребують своїх власних конфігурацій для задоволення бізнес-потреб. Самі інструменти мають власні способи конфігурації, які ще не стандар-

тизовані. Це призводить до гальмування процесу налаштування кожного сегмента конвеєра. Для деяких проектів ця кількість зусиль для створення конвеєру є однією з основних причин, через які вони не використовують постійне розгортання [7; 10; 11].

Викладене вище актуалізує вирішення завдання вибору відповідного інструменту CI для автоматизованого тестування ПЗ шляхом побудови моделі оцінювання таких інструментів.

Мета та постановка завдання

Метою цієї роботи є створення моделі оцінювання інструментів CI для процесу тестування програмних продуктів за допомогою системи підтримки прийняття рішень (СППР).

Для досягнення поставленої мети необхідно проаналізувати можливості основних інструментів CI та визначити критерії їхнього оцінювання; обрати СППР та розробити модель оцінювання інструментів CI за допомогою обраної системи; запровадити рекомендований системою інструмент CI для тестування програмного продукту, отримати та проаналізувати результати використання моделі оцінювання.

Виклад основного матеріалу

Під час роботи було проведено порівняльний аналіз переваг та недоліків найбільш уживаних сьогодні засобів CI, таких як Jenkins, GitlabCI, CircleCI, TeamCity, Bamboo [3; 6; 8; 9; 10].

Виходячи із проведеного аналізу характеристик зазначених інструментів, було визначено основні критерії, що полегшують вибір CI-засобу для автоматизованого тестування. До них належать такі характеристики:

- вартість інструменту;
- інтеграція з інструментами сторонніх розробників (Maven, Allure Report);
- синхронізація з системами управління репозиторіями (Github/Gitlab);
- можливість налагодження для перевірки коду, який був оновлений;
- середовище розміщення інструменту (чи є інструмент готовим рішенням із вебінтерфейсом або потребує інсталяції).

Для побудови моделі оцінювання інструментів CI за визначеними критеріями було використано СППР «Вибір», що є аналітичною системою, основою на застосуванні методу аналізу ієрархій [5]. Метод аналізу ієрархій (MAI), що належить до класу критеріальних методів, є математичним інструме-

нтом системного підходу до складних проблем прийняття рішень.

На підготовчому етапі побудови моделі оцінювання в СППР було встановлено вагомість визначених критеріїв шляхом опитування експертів команди DevOps. Результати опитування (рис. 1) показали, що найбільш вагомими критеріями є вартість інструменту та синхронізація із системами управління репозиторіями. Однаково важливими є інтеграція з інструментами сторонніх розробників та можливість налагодження, а найменш вагомим критерієм є середовище розміщення продукту.



Рис. 1. Результати опитування експертів щодо вагомості критеріїв оцінювання CI інструментів

Крім цього, експертам було запропоновано оцінити відповідність кожного інструменту критеріям за 5-бальною шкалою. Результати опитування експертів було використано для побудови моделі оцінювання інструментів.

Унаслідок було отримано ієрархію для оцінювання інструментів, яка пов'язує мету оцінювання, критерії та альтернативи, що є в нашому випадку інструментами CI, з яких необхідно здійснити вибір. Отримана ієрархія для оцінювання інструментів зображена на рис. 2.

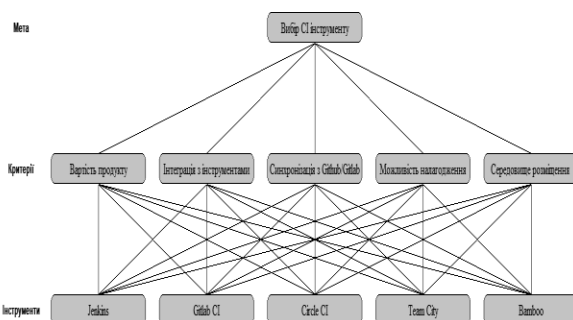


Рис. 2. Побудована ієрархія для оцінювання інструментів CI

Далі було побудовано матрицю попарних порівнянь критеріїв (рис. 3). Вона була запо-

внена результатами опитування експертів щодо вагомості критеріїв.

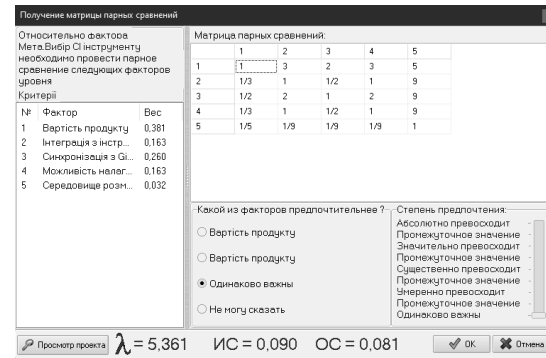


Рис. 3. Побудована матриця попарних порівнянь критеріїв

На цьому етапі також було розраховано індекс узгодженості (IU) суджень експерта, який визначається за такою формулою:

$$IU = \frac{\lambda_{\max} - n}{(n-1)}$$

де λ_{\max} – максимальне власне значення (ВЗ); n – розмірність матриці.

Розрахунок максимального власного значення λ_{\max} здійснюється за матрицею парних порівнянь таким чином: підсумовується кожен стовпець суджень, сума першого множиться на величину першої компоненти нормалізованого вектора пріоритетів, сума другого стовпця – на величину другої компоненти тощо, потім отримані числа підсумовуються.

Відношення узгодженості розраховується як $VU = (IU / C3) * 100\%$. Якщо VU більше ніж 10 %, необхідний перегляд суджень.

На наступному етапі були складені матриці парних порівнянь альтернатив за всіма критеріями, і в такий спосіб проведено порівняння інструментів CI за всіма критеріями (рис. 4).

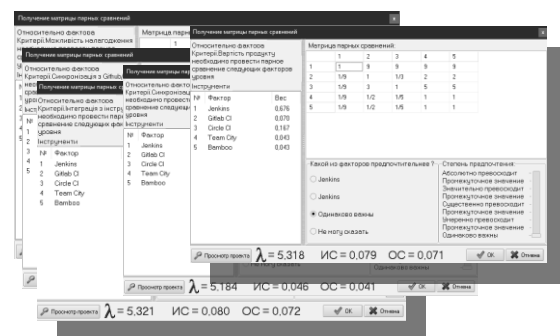


Рис. 4. Побудовані матриці попарних порівнянь інструментів CI

Після використання моделі оцінювання було отримано результат у вигляді діаграми (рис. 5), де рекомендованим інструментом інтеграції визначено інструмент CircleCI.

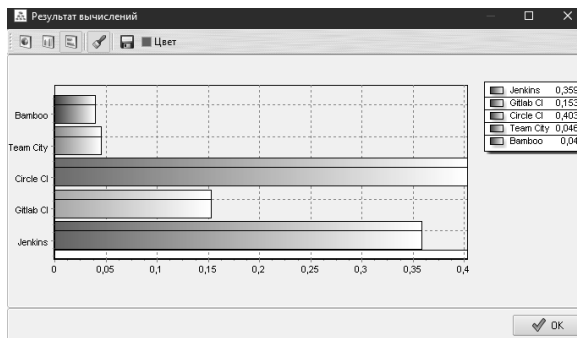


Рис. 5. Результат використання моделі оцінювання інструментів CI

Рекомендований СППР інструмент безперервної інтеграції CircleCI було запроваджено на підприємстві для тестування реального програмного продукту (вебзастосунку для роботи з постачальниками деяких товарів).

У процесі підготовки до використання рекомендованого інструменту було розроблено 20 тест-кейсів та відповідний конфіг-файл, після чого до CircleCI було приєднано репозиторій із тестами. Завдяки налаштуванням у конфіг-файлі тести запускалися автоматично кожного дня. Унаслідок на головній сторінці інструменту відображалися всі запуски, що виконувалися після кожного коміту до репозиторію. Перегляд запусків дає змогу зрозуміти, чи є проблеми з продуктом після певної зміни в коді. Разом із статусом виконання вказується назва гілки в Git та код коміту, що дозволяє дізнатися, які саме зміни вплинули на роботу вебзастосунку.

Підсумовуючи результати використання інструменту CircleCI, рекомендованого СППР для автоматизованого тестування реального програмного продукту, можна зробити висновки щодо доцільності його застосування у практиці безперервної інтеграції. Зокрема встановлено, що використання CircleCI полегшує роботу тестувальників за рахунок налаштованої автоматизованості, а звіт Allure Report, що є артефактом кожного запуску тестів, значно підвищує швидкість формування звітів під час регресійного тестування.

Установлено також, що інструмент Circle CI відповідає загалом усім критеріям оцінювання. Вартість інструменту виправдана,

оскільки для забезпечення автоматизованого тестування невеликого проекту достатньо безкоштовної версії. Проблем з інтеграцією з інструментами сторонніх розробників не виявлено, проте встановлено, що артефакти доступні для завантаження лише протягом 30 днів. Під час використання інструменту репозиторій з кодом розміщується на GitHub, проблем із синхронізацією не виникало, вона реалізована зручно та швидко. Робота з інструментом CircleCI, що є самостійним вебзастосунком з інтуїтивно зрозумілим інтерфейсом і не потребує спеціальних завантажень та налаштувань, не викликала ускладнень і додаткового навчання персоналу.

Висновки

Отже, у роботі проаналізовано можливість основних інструментів CI для автоматизованого тестування та визначено критерії їхнього оцінювання. На основі визначених критеріїв розроблено модель оцінювання інструментів CI за допомогою СППР та використано її для вибору відповідного інструменту. Рекомендований СППР інструмент безперервної інтеграції CircleCI було запроваджено на підприємстві для тестування реального програмного продукту. Отримано й проаналізовано результати використання моделі оцінювання.

Література

1. Автоматизоване тестування: підготовка стратегії. URL: <https://dou.ua/lenta/articles/preparation-for-test-automation/>.
2. Автоматизація тестування: як уникнути поширених помилок. URL: <https://www.globallogic.com/ua/insights/blogs/qa-automation-2/>.
3. Безперервна інтеграція: CircleCI vs Travis CI vs Jenkins. URL: <https://habr.com/ru/company/southbridge/blog/>.
4. Принципи CI/CD. URL: <https://medium.com/southbridge/>.
5. СППР Вибір. URL: <https://allsoft.ru/software/vendors/tsiritas/-spgr-vybor/>.
6. Що таке безперервна інтеграція Дженкінса. URL: <https://uk.myservername.com/jenkins-ci>.
7. A Basic Introduction to DevOps Tools. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.735.292>.
8. Atlassian Bamboo. URL: <https://www.atlassian.com/ru/software/bamboo>.
9. Comparison of Most Popular Continuous Integration Tools: Jenkins, TeamCity, Bamboo, Travis CI and more. URL: <https://www.altexsoft.com/blog/engineering/comparison-of-most-popular-continuous-integration-tools>.

10. Continuous integration and delivery – Circle CI. URL: <https://circleci.com/>.
11. DevOps methodology and process. URL: <https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd>.

References

1. Automated testing: strategy preparation. URL: <https://dou.ua/lenta/articles/preparation-for-test-automation/> [in Ukrainian].
2. Automated testing: how to avoid common mistakes. URL: <https://www.globallogic.com/ua/insights/blogs/qa-automation-2/> [in Ukrainian].
3. Continuous integration: CircleCI vs Travis CI vs Jenkins. URL: <https://habr.com/ru/company/southbridge/blog/332836/> [in Ukrainian].
4. Principles of CI/CD. URL: <https://medium.com/southbridge/> [in Ukrainian].
5. SSTD Vybor. URL: <https://allsoft.ru/software/vendors/tsiritas/-sppr-vybor/> [in Russian].
6. What is continuous integration of Jenkins. URL: <https://uk.myservername.com/-jenkins-ci-tutorial-what-is-jenkins/> [in Ukrainian].
7. A Basic Introduction to DevOps Toolsю. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.735.292>.
8. Atlassian Bamboo. URL: <https://www.atlassian.com/ru/software/bamboo>.
9. Comparison of Most Popular Continuous Integration Tools: Jenkins, TeamCity, Bamboo, Travis CI and more. URL: <https://www.altexsoft.com/blog/engineering/comparison-of-most-popular-continuous-integration-tools>.
10. Continuous integration and delivery – Circle CI. URL: <https://circleci.com/>
11. DevOps methodology and process. URL: <https://medium.com/@raycad.seedotech/devops-methodology-and-process-dde388eb65bd>.

Гризун Людмила Едуардівна, д.п.н., проф. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, Lgr2007@ukr.net, тел. +38 0954139386,

Скорін Юрій Іванович, к.т.н., доц. каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, skorin.yuriy@gmail.com, тел. +38 066-748-47-51,

Деточенко Ірина Миколаївна, магістрант каф. інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця,

detochenko@hneu.net, tel. +38 066-423-85-02.

Choosing a continuous integration tool for software automated testing

Abstract. Problem. Automated testing is directly related to continuous integration (CI), which is the performance of frequent automated grouping of project components to identify and resolve integration issues. Choosing the adequate CI tool for automated testing is the key to effective CI process. Analysis of theory and practice shows that the choice of such an instrument to support testing remains a challenge. According to the sources, finding and combining compatible tools to create a functional pipeline is also a difficult task. The problem that arises due to the combination of tools is their configuration. The said above actualizes the search of the solution to the problem of choosing the appropriate CI tool for software testing by building a model for evaluating these tools. **Goal.** The aim of the work is to create a model for evaluating CI tools for the software testing process using the decision support system (DSS). **Methodology.** The analytical methods of research, the method of the hierarchy analysis, and the method of modeling are used. **Results.** In the course of the work, the possibilities of the main CI tools for automated testing were analyzed and the criteria for their evaluation were determined. **Originality.** Based on the defined criteria, a model for evaluating CI tools using DSS was developed and used to select the appropriate instrument. **Practical value.** The tool for continuous integration recommended by DSS, CircleCI, has been implemented at the enterprise to test a real software product. The results of using the evaluation model are obtained and analyzed.

Key words: tools for continuous integration, automated software testing, evaluation model, decision support system.

Gryzun Liudmyla, Doct. of Science, Professor, Information Systems Department, Simon Kuznets Kharkiv National University of Economics, Lgr2007@ukr.net, tel. +38 0954139386,

Skorin Yuriy, Ph.D., Assoc. Prof. Information Systems Department, Simon Kuznets Kharkiv National University of Economics, skorin.yuriy@gmail.com, tel. +38 066-748-47-51,

Detochenko Iryna, master-student, Information Systems Department, Simon Kuznets Kharkiv National University of Economics, detochenko@hneu.net, tel. +38 066-423-85-02.