

ПЕРЕХРЕСНІ РЕВ'Ю КОДУ ТА ІНТЕГРАЦІЯ КОМАНДНИХ ПРОЄКТІВ ПРИ ВИВЧЕННІ ДИСЦИПЛІН DEVOPS

CROSS CODE REVIEWS AND TEAM PROJECT INTEGRATION IN DEVOPS COURSE STUDIES

У статті розглядається значення перехресних ревію коду (code review) та інтеграції командних проєктів у процесі вивчення дисциплін DevOps, що є важливою складовою підготовки майбутніх фахівців у галузі розробки та експлуатації програмного забезпечення. Перехресне ревію коду визначається як процес взаємної перевірки програмного коду учасниками команди з метою виявлення помилок, оптимізації рішень та забезпечення відповідності заданим стандартам. Ця практика розглядається не лише як технічний інструмент, але й як засіб розвитку аналітичного мислення, критичного підходу до виконання завдань та здатності до конструктивного зворотного зв'язку.

Інтеграція командних проєктів у навчальний процес зосереджується на використанні сучасних інструментів і платформ, таких як системи управління версіями (Git), інструменти для спільної роботи (GitHub, GitLab) та автоматизовані середовища (Jenkins, Docker, Kubernetes). Це забезпечує студентам можливість працювати в умовах, наближених до реальних, моделюючи процеси, що зазвичай застосовуються у промисловій розробці. У статті обговорюються методології, які підтримують ці процеси, зокрема гнучкі підходи до управління проєктами (Agile, Scrum), принципи безперервної інтеграції та доставки (CI/CD) та інструменти автоматизації тестування.

Особливу увагу приділено ролі таких практик у формуванні практичних навичок студентів. Перехресне ревію сприяє вдосконаленню технічної грамотності та підвищенню якості створеного коду, тоді як інтеграція командних проєктів розвиває навички командної роботи, включаючи управління конфліктами, планування ресурсів та ефективну комунікацію. Окремо розглянуто їхній вплив на підвищення здатності студентів адаптуватися до реальних умов професійної діяльності, що вимагає вміння швидко реагувати на зміни, співпрацювати з іншими розробниками й використовувати сучасні технології.

Ключові слова: DevOps, перехресне ревію коду, командні проєкти, автоматизація, навчання, CI/CD.

The article examines the importance of cross code reviews and the integration of team projects in the process of studying DevOps disciplines, which are essential components in training future specialists in software development and operations. Cross code review is defined as a process of mutual examination of program code by team members to identify errors, optimize solutions, and ensure compliance with established standards. This practice is considered not only as a technical tool but also as a means of developing analytical thinking, a critical approach to task execution, and the ability to provide constructive feedback.

The integration of team projects into the learning process focuses on the use of modern tools and platforms, such as version control systems (Git), collaboration tools (GitHub, GitLab), and automated environments (Jenkins, Docker, Kubernetes). This enables students to work in conditions close to real-world scenarios, simulating processes typically used in industrial development. The article discusses methodologies supporting these processes, including agile project management approaches (Agile, Scrum), the principles of continuous integration and delivery (CI/CD), and test automation tools.

Particular attention is given to the role of these practices in developing students' practical skills. Cross code reviews enhance technical proficiency and improve code quality, while the integration of team projects fosters teamwork skills, including conflict management, resource planning, and effective communication. Additionally, their impact on enhancing students' adaptability to real-world professional conditions is analyzed, emphasizing the ability to respond quickly to changes, collaborate with other developers, and utilize modern technologies.

Key words: DevOps, cross code review, team projects, automation, education, CI/CD.

УДК 004.4:378.1
DOI <https://doi.org/10.32782/2663-6085/2024/78.1.25>

Лучкевич М.М.,

канд. фіз.-мат. наук, доцент,
доцент кафедри інформаційних систем та мереж
Національного університету
«Львівська політехніка»

Постановка проблеми у загальному вигляді.

Сучасна індустрія програмного забезпечення є однією з найдинамічніших сфер економіки, що характеризується стрімким розвитком технологій, постійними змінами у вимогах до продуктів та зростанням очікувань користувачів щодо їхньої якості, надійності та швидкості оновлення. В умовах глобалізації, цифровізації та автоматизації бізнес-процесів необхідність адаптації до цих викликів змушує компанії вдосконалювати свої підходи до розробки програмного забезпечення, що вимагає інтеграції різних етапів його створення і впровадження.

Одним із найбільш ефективних підходів, який відповідає цим викликам, є DevOps – методологія,

що поєднує розробку (Development) та експлуатацію (Operations) у єдиний безперервний процес [1]. Ця інтеграція дозволяє оптимізувати життєвий цикл програмного забезпечення шляхом використання таких принципів, як безперервна інтеграція (Continuous Integration, CI) та безперервне постачання (Continuous Delivery, CD). Впровадження DevOps забезпечує високу якість кінцевого продукту, мінімізацію часу між розробкою і випуском оновлень, а також підвищення загальної ефективності розробницьких команд.

Ключовим аспектом підготовки фахівців у галузі DevOps є не лише оволодіння сучасними інструментами, такими як Docker, Kubernetes, Ansible чи Jenkins, але й розвиток навичок командної роботи,

що є надзвичайно важливим у контексті багатоконандних середовищ. Успіх DevOps багато в чому залежить від узгодженості дій між командами розробників, тестувальників та інженерів із підтримки, що вимагає чіткої комунікації, спільного розуміння цілей і здатності ефективно вирішувати конфлікти.

Серед найважливіших практик, які сприяють формуванню цих навичок, виділяються перехресні ревію коду (code review) та інтеграція командних проєктів. Перехресні ревію сприяють підвищенню якості коду за рахунок виявлення та усунення помилок, дотримання стандартів програмування та впровадження кращих практик. Окрім технічної користі, цей процес стимулює розвиток критичного мислення, навичок конструктивної критики та вміння аргументовано захищати запропоновані рішення.

Інтеграція командних проєктів у навчальний процес дозволяє студентам працювати в умовах, наближених до реальних. Це забезпечує формування комплексного розуміння процесів DevOps, від початкового етапу планування до розгортання готового продукту. Завдяки використанню сучасних платформ для командної роботи (GitHub, GitLab, Bitbucket) та автоматизації (CI/CD пайплайнів), студенти отримують практичний досвід співпраці в багатоконандному середовищі, навчаються вирішувати конфлікти злиття (merge conflicts) та адаптуватися до швидкозмінних умов проєктної діяльності.

Таким чином, сучасна підготовка спеціалістів у сфері програмного забезпечення повинна бути комплексною, охоплюючи як технічні, так і соціальні аспекти, адже саме така синергія забезпечує їхню готовність до професійної діяльності в умовах реальної індустрії.

Аналіз останніх досліджень і публікацій. Перехресне ревію коду (code review) є однією з ключових практик, що активно застосовується у сучасній індустрії розробки програмного забезпечення [2]. У навчальному контексті ця практика передбачає взаємну перевірку коду студентами, які виконують роль як авторів, так і рецензентів. Основною метою такого підходу є не лише виявлення помилок чи недоліків у написаному коді, але й оптимізація алгоритмів, підвищення якості програмного продукту та забезпечення дотримання загальноприйнятих стандартів кодування [3].

Перехресне ревію сприяє розвитку важливих технічних навичок, таких як аналіз програмних рішень, рефакторинг та впровадження оптимальних практик програмування [4]. Однак цінність цієї практики виходить за межі суто технічних аспектів. Вона стимулює розвиток здатності до конструктивної критики, що є надзвичайно важливим у командній роботі. Учасники не лише вчать аргументовано висловлювати зауваження, але й отримують досвід прийняття зворотного зв'язку,

що сприяє формуванню професійної етики та навичок міжособистісної комунікації [5].

Інтеграція командних проєктів у навчальний процес є важливою складовою підготовки студентів до реальної роботи в умовах багатоконандного середовища [6]. Основою таких проєктів є використання систем управління версіями, таких як Git, що забезпечують ефективну координацію роботи над спільним кодом. Платформи для спільної розробки, зокрема GitHub, GitLab чи Bitbucket, створюють середовище для організації командної роботи, дозволяючи контролювати історію змін, відстежувати прогрес виконання завдань та вирішувати конфлікти злиття [7].

Важливим компонентом інтеграції командних проєктів є застосування інструментів автоматизації, таких як Jenkins, Docker та Kubernetes [8]. Ці технології дозволяють студентам ознайомитися з основами автоматизації процесів побудови, тестування та розгортання програмного забезпечення. Наприклад, Jenkins сприяє впровадженню принципів безперервної інтеграції (CI), що дає змогу виявляти помилки на ранніх етапах розробки, тоді як Docker і Kubernetes забезпечують створення контейнеризованих середовищ та оркестрацію додатків, імітуючи реальні виробничі сценарії.

Таким чином, комбінація перехресного ревію коду та інтеграції командних проєктів у навчальний процес створює комплексне середовище для формування як технічних, так і «м'яких» навичок (soft skills) [9]. Цей підхід дозволяє студентам отримати досвід командної роботи, вирішення конфліктів, ефективного планування й управління проєктами, що є критично важливим для їхньої майбутньої професійної діяльності у сфері DevOps та розробки програмного забезпечення.

Виділення невирішених раніше частин загальної проблеми. Невирішені частини загальної проблеми зосереджуються на відсутності комплексного підходу до інтеграції DevOps-практик у навчальний процес, який би одночасно розвивав технічні та соціальні навички студентів. Існує недостатній рівень адаптації навчальних курсів до умов реальної роботи в DevOps, особливо щодо впровадження автоматизації та організації роботи в багатоконандному середовищі. Також є потреба у створенні відповідної навчальної інфраструктури, яка б включала сучасні інструменти, середовища та хмарні сервіси, а також у розробці методологій для оцінювання індивідуального і командного внеску студентів.

Мета статті. Обґрунтування важливості інтеграції DevOps-практик, зокрема перехресного ревію коду та командних проєктів, у навчальний процес для підготовки фахівців у сфері розробки та експлуатації програмного забезпечення, а також аналіз їхнього впливу на формування технічних

і соціальних навичок, необхідних для ефективної роботи в умовах реальної індустрії.

Виклад основного матеріалу. Дослідження ефективності впровадження практик DevOps у навчальному середовищі було проведено на базі Національного університету «Львівська політехніка». Експеримент охопив 64 студентів, які були розподілені на команди чисельністю 4–5 осіб для імітації реальних умов командної роботи. Навчальний курс тривав один семестр і включав завдання різного рівня складності, що відображали типові задачі DevOps: автоматизацію процесів безперервної інтеграції та доставки (CI/CD), розробку веб-додатків із використанням сучасних фреймворків, підтримку програмного забезпечення та проведення перехресних рев'ю коду.

Завдання було спроектовано таким чином, щоб студенти могли інтегрувати теоретичні знання з практичними навичками, працюючи з реальними інструментами, такими як Git, Docker, Jenkins і Kubernetes. Особливу увагу було приділено впровадженню практики перехресного рев'ю коду, що виконувалося після кожного значного етапу проекту. Цей процес забезпечував студентам можливість не лише отримувати зворотний зв'язок щодо своїх рішень, але й критично оцінювати чужий код, аналізуючи його відповідність стандартам, продуктивність та потенційні вразливості.

Результати дослідження свідчать про високу ефективність впроваджених практик у процесі навчання. Зокрема, перехресні рев'ю коду дозволили зменшити кількість дефектів у кінцевих проектах на 25–30% у порівнянні з проектами, які виконувалися без застосування цієї практики. Це підкреслює значення регулярного взаємного контролю та обміну знаннями між студентами, що сприяє підвищенню загальної якості програмного продукту.

Взаємодія між командами стала важливим елементом у процесі вирішення конфліктів злиття коду, що часто виникали через одночасне редагування різних компонентів. Студенти не лише отримали практичний досвід роботи з системою контролю версій Git, але й навчилися ефективно вирішувати конфлікти, аналізуючи зміни та дотримуючись принципів спільної відповідальності за результат. Це сприяло покращенню їхніх навичок роботи з розподіленими репозиторіями та формуванню вміння ефективно комунікувати у команді.

Застосування інструментів автоматизації, таких як Jenkins для CI/CD, Docker для створення контейнеризованих середовищ та Kubernetes для оркестрації, прискорило загальний процес розробки на 15–20%. Цей показник було досягнуто завдяки зменшенню часу на ручні операції, підвищенню точності тестування та мінімізації людського фактору у процесах побудови і розгортання програмного забезпечення.

Отримані дані свідчать про те, що використання практик DevOps у навчальних курсах не лише сприяє формуванню ключових технічних навичок, але й розвиває важливі міжособистісні компетенції, необхідні для успішної роботи у сучасній індустрії програмного забезпечення. Перехресні рев'ю коду, інтеграція командних проектів та застосування автоматизації дозволяють створити середовище, яке максимально наближене до реальних умов розробки, готуючи студентів до викликів професійної діяльності.

Застосування практик DevOps у навчальному процесі є важливим кроком у підготовці фахівців, здатних ефективно працювати у високодинамічних умовах сучасної IT-індустрії. Такий підхід дозволяє створити умови, які імітують реальні виробничі процеси, що включають автоматизацію, командну роботу, управління конфліктами та інтеграцію нових функціональних можливостей. Завдяки цьому студенти отримують не лише технічні знання, але й цінний досвід співпраці, який є критично важливим у багатокомандних середовищах.

Однією з найбільш значущих переваг впровадження практики перехресного рев'ю коду є її комплексний вплив на розвиток студентів. Окрім технічного вдосконалення, пов'язаного з аналізом коду, оптимізацією алгоритмів та виявленням помилок, студенти розвивають так звані «soft skills» – навички, які забезпечують ефективну міжособистісну взаємодію. Зокрема, перехресне рев'ю сприяє покращенню комунікаційних здібностей: студенти вчаться надавати конструктивний зворотний зв'язок, аргументувати свої зауваження, висловлювати власну думку та сприймати критику. Також цей процес допомагає розвивати навички управління часом, оскільки дотримання дедлайнів та синхронізація роботи команди є невід'ємною частиною ефективного рев'ю. Окрім того, робота в умовах обмежених ресурсів і високих вимог до якості виховує вміння працювати у стресових ситуаціях, що є невід'ємною частиною професійної діяльності в сфері DevOps.

Інтеграція командних проектів у навчальний процес значно посилює практичну підготовку студентів, однак ставить перед викладачами низку викликів. Для ефективної реалізації такого підходу необхідно створити відповідне середовище, яке включає налаштування сучасних інструментів і платформ, таких як Git, Docker, Kubernetes, Jenkins, а також забезпечення доступу до хмарних сервісів для тестування та розгортання додатків. Розробка завдань вимагає врахування рівня підготовки студентів, адекватності складності проектів і відповідності їх реальним сценаріям з галузі DevOps.

Моніторинг виконання завдань є ще одним важливим аспектом. Викладачі повинні не лише

слідкувати за прогресом команд, але й забезпечувати прозорість оцінювання, мотивувати студентів і надавати їм своєчасний зворотний зв'язок. Це потребує значних ресурсів, зокрема автоматизації процесів оцінки, використання трекерів задач (наприклад, Jira, Trello) та створення чітких критеріїв оцінювання. Попри всі ці складнощі, інвестовані зусилля виправдовуються високим рівнем підготовки випускників, які виходять на ринок праці з глибокими практичними знаннями, адаптованими до сучасних умов роботи.

Таким чином, інтеграція практик DevOps у навчальний процес сприяє формуванню комплексного набору технічних і міжособистісних навичок у студентів, дозволяючи їм адаптуватися до професійних викликів. Незважаючи на високі вимоги до організації та проведення таких курсів, вони забезпечують значну перевагу для студентів, які готуються до роботи у висококонкурентному середовищі сучасної IT-індустрії.

Висновки. Запровадження практик перехресних ревію коду та інтеграції командних проєктів у навчальні курси з DevOps продемонструвало свою ефективність як засобу підготовки студентів до роботи у сучасній індустрії розробки програмного забезпечення. Ці підходи забезпечують всебічний розвиток студентів, дозволяючи їм не лише вдосконалити свої технічні навички, але й сформуванню критично важливі соціальні компетенції, такі як комунікація, колективне вирішення проблем і управління часом.

Практика перехресного ревію коду сприяє підвищенню якості програмних продуктів завдяки своєчасному виявленню помилок і впровадженню оптимальних рішень. Студенти отримують можливість навчитися об'єктивно оцінювати код, працювати з відгуками, а також розвивати здатність до конструктивного спілкування, що є необхідним у командних середовищах. Інтеграція командних проєктів, у свою чергу, дозволяє студентам отримати реальний досвід роботи з такими інструментами, як Git, Docker, Jenkins, та платформами для спільної розробки. Це створює умови, наближені до реальних виробничих процесів, де важливими є як технічна досконалість, так і здатність ефективно співпрацювати в команді.

Водночас реалізація таких підходів вимагає значних зусиль з боку викладачів. Зокрема, необхідно створити відповідну інфраструктуру, забезпечити навчання студентів використанню складних інструментів та розробити методології для оцінювання їхньої роботи. Проте отримані результати підтверджують, що ці зусилля виправдані, оскільки випускники курсів з інтегрованими

DevOps-практиками демонструють вищий рівень готовності до професійної діяльності, кращу адаптацію до реальних умов роботи та здатність ефективно вирішувати поставлені задачі.

Перспективи подальших досліджень.

Подальші дослідження в цьому напрямку мають бути зосереджені на оптимізації існуючих методів та розробці практичних рекомендацій для масштабного впровадження таких підходів у навчальний процес. Це включає аналіз довгострокових результатів навчання, адаптацію курсів до різних рівнів підготовки студентів і створення універсальних методик для використання у різних освітніх контекстах. Зокрема, важливим є впровадження інструментів автоматизованого моніторингу прогресу, систем оцінювання командної роботи та методів, що сприяють індивідуалізації навчання у контексті командних проєктів.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Amaro, Ricardo, Rúben Pereira, and Miguel Mira da Silva. Mapping DevOps capabilities to the software life cycle: A systematic literature review. *Information and Software Technology* (2024): 107583.
2. Turzo, Asif Kamal, and Amiangshu Bosu. What makes a code review useful to opendev developers? an empirical investigation. *Empirical Software Engineering* 29.1 (2024): 6.
3. Tufano, Rosalia, et al. Code review automation: strengths and weaknesses of the state of the art. *IEEE Transactions on Software Engineering* (2024).
4. Ferreira, Isabella, Ahlaam Rafiq, and Jinghui Cheng. Incivility detection in open source code review and issue discussions. *Journal of Systems and Software* 209 (2024): 111935.
5. Tufano, Rosalia, et al. Towards automating code review activities. *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021.
6. Skilton, Paul F., David Forsyth, and Otis J. White. Interdependence and integration learning in student project teams: Do team project assignments achieve what we want them to?. *Journal of Marketing Education* 30.1 (2008): 57–65.
7. Ghodke, Gayatri Makrand, and Trupti Chavan. An Overview of Git. *International Journal of Scientific Research in Modern Science and Technology* 3.6 (2024): 17–23.
8. Riti, Pierluigi. Pro DevOps with Google Cloud Platform: With Docker, Jenkins, and Kubernetes. Apress, 2018.
9. Pang, Candy, Abram Hindle, and Denilson Barbosa. Understanding devops education with grounded theory. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training*. 2020.