

Пундик В.І.

Тернопільський національний технічний університет імені Івана Пулюя

РОЗРОБКА ПРОГРАМНИХ СИСТЕМ В КОНТЕКСТІ ХМАРНИХ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ МЕТОДОЛОГІЇ DEVOPS ТА ГНУЧКОЇ МОДЕЛІ РОЗРОБКИ

У даній роботі досліджується інтеграція методологій Agile та DevOps, зосереджуючись на їх подібності та перевагах у поєднанні. Agile сприяє ітераційній та поетапній розробці, надаючи невеликі часті випуски для швидкої адаптації до мінливих вимог. DevOps покращує співпрацю між командами розробки та операцій, наголошуючи на постійній інтеграції та безперервній доставці (CI/CD) і автоматизації.

Дослідження включає експериментальний проект для демонстрації функціональності гібридної моделі Agile DevOps. Проект складається з розробки хмарного веб-додатку для електронної комерції в три етапи: визначення вимог клієнта та створення резерву спринтів, розробка та тестування компонентів проекту та доставка кожного компонента клієнту.

Структура проекту включає вісімнадцять спринтів, кожен з яких зосереджений на різних модулях веб-додатку, таких як інтерфейсна розробка, автентифікація, керування продуктами, кошик для покупок, оформлення замовлення, профіль користувача. У статті докладно описано використання систем контролю версій (GitHub, GitLab), інструментів CI/CD (Jenkins, CircleCI), інструментів інфраструктури як коду (IaC) (Terraform, AWS CloudFormation) та інструментів моніторингу (Prometheus, Grafana) для ефективного розробки, тестування та розгортання.

Перша ітерація використовувала методологію Agile, призначаючи членів команди для окремих секторів розробки, тестування та розгортання. Спостереження показали, що хоча сектор розробки виконував завдання достроково, тестування та розгортання відставали, що свідчить про брак координації.

У другій ітерації було реалізовано гібридну методологію Agile DevOps із спільною роботою крос-функціональних команд. Такий підхід сприяв кращій координації та послідовному прогресу в усіх секторах, що призвело до покращення загальної продуктивності та своєчасного виконання завдань.

У статті робиться висновок, що гібридна методологія Agile DevOps покращує розробку проектів, сприяючи безперервній співпраці та інтеграції, зрештою виявляється більш ефективною, ніж традиційний підхід Agile.

Ключові слова: гнучка розробка, методологія DevOps, безперервна інтеграція/безперервна доставка, крос-функціональні команди, хмарна електронна комерція, інфраструктура як код.

Постановка проблеми. У сфері розробки програмного забезпечення, методології, що керують виконанням проекту, відіграють ключову роль у формуванні результатів. Команди в різних галузях постійно стикаються з проблемою ефективної доставки високоякісних програмних продуктів, які відповідають вимогам користувачів і ринків, що постійно змінюються. У відповідь на це з'явилася різноманітна низка методологій, кожна з яких пропонує унікальні рамки та принципи для керування процесами розробки.

Ці методології пропонують структуровані підходи до управління проектами, співпраці та доставки. Дане дослідження методологій розробки програмного забезпечення заглиблюється в стратегії та принципи, які забезпечують успішні

результати проекту. Від початку до реалізації досліджуються нюанси управління проектами розробки в сучасному динамічному ландшафті, з оглядом на підходи, які дозволяють командам прогресувати в сфері розробки програмного забезпечення, що постійно розвивається.

Аналіз останніх досліджень і публікацій.

У науково-дослідницькому просторі сьогодення з'являються роботи, присвячені винаходу та аналізу методологій по створенню методів та моделей розробки програмних систем із використанням хмарних технологій.

У роботі [1] було проведено дослідження щодо поєднання DevOps з методологією бережливої розробки програмного забезпечення (Lean Software Development). Це дослідження предста-

вило Lean Software Development як методологію, яка зосереджена на зменшенні відходів і постійному вдосконаленні, а DevOps було представлено як набір практик, спрямованих на покращення загального життєвого циклу розробки шляхом інтеграції розробки та операцій разом у багатofункціональних командах. Метод дослідження передбачає впровадження практик DevOps для усунення виявлених прогалин у ошадливій розробці програмного забезпечення. Цей метод зміг використати практики DevOps для подолання проблем ошадливої розробки програмного забезпечення в різних сферах, включаючи затримки, дефекти, додаткові функції, перемикавання завдань тощо. Обмеження цієї роботи полягає в тому, що застосування принципів DevOps для подолання існуючих проблем мало бути перевірено лише за допомогою Lean Software Development, а дослідження подібних випадків застосування принципів DevOps для Scrum, XP та інших методів було залишено як перспективи для майбутніх досліджень.

У роботі [2] приділяється увага об'єднанню DevOps зі штучним інтелектом. У статті підкреслюється альтернативна вартість, пов'язана з переходом від ліцензування програмного забезпечення до програмного забезпечення як послуги (Software as Service – SaaS), і підкреслюються переваги, отримані завдяки ранньому та регулярному випуску програмного забезпечення організаціями, які прийняли цю практику. DevOps представляється як революційний підхід, спрямований на усунення прогалин у двох центральних процесах, а саме розробці та експлуатації. Нові технології, які включають великі дані та хмарні обчислення, вимагають швидкого розгортання програмного забезпечення, яке досягається завдяки використанню концепцій безперервної інтеграції (CI) і безперервної доставки (CD), вказуючи на економічну ефективність і роль автоматизації у виробничому процесі.

Крім того, варто зазначити праці наступних науковців: Доріс Л., Поттер К., [3], Аль Масуд С., Маснун М., Султан А., Ахмед Ф., Бегум Н. [4], Гантус Г., Гілл А. [5], Мохаммад С. М. [6], Алмейда Ф., Сімоєш Х., Феррейра Лопес С. Ф. [7], Россберг Дж. [8], Гупта К., Фернандес-Креует Х. М., Гупта Ст [9], Нгуен Т., Хюїнь Т., Тран Л. [10], Арімура Ю., Іто М. [11], Чженлун П., Чжунхуей О., Юлан Х. [12], Хан Х., Алі Ф., Назір С. [13], Огала Дж., Мугеле С. [14], Паніграхі До., Молл Р., Паті Б. [15], Ся Ф. [16], Одун-Айо І., Адесола Ф., Семюел Ст [17] та інших.

Проте, беручи до уваги вище зазначену наукову документацію, питання, пов'язане з методологією по створенню методів та моделей розробки програмних систем із використанням хмарних технологій, все ще залишається недостатньо дослідженим та потребує подальшого опрацювання.

Постановка завдання. Метою роботи є створення методів та моделей розробки програмних систем із використанням хмарних технологій.

Виклад основного матеріалу дослідження. Гнучка модель розробки (Agile) та методологія DevOps (Development & Operations – Розробка і операції) являють собою концепції з певними відмінностями, які насамперед полягають у тому, що Agile представляються як спільний та ітеративний підхід до розробки програмного забезпечення, який зосереджується на постачанні невеликих поетапних випусків, що забезпечує гнучкість і швидку адаптацію до мінливих вимог, а DevOps, з іншого боку, відповідає набір практик, які спрямовані на покращення співпраці та зв'язку між командами розробки та операцій, сприяючи культурі постійної інтеграції та безперервної доставки (CI/CD – continuous integration, continuous delivery) і автоматизації.

Тим не менш, обидві концепції мають ряд ключових спільних характеристик, які дозволяють комбіновану інтеграцію у процес розробки. Насамперед, серед спільностей, найважливішою виявляється ітеративна та поетапна розробка, яка полягає у тому, що як Agile, так і DevOps виступають за поділ роботи на менші, керовані кроки. Цей ітеративний підхід дозволяє командам збирати відгуки на ранній стадії та часто, забезпечуючи швидку адаптацію до мінливих вимог і динаміки ринку. Окрім того, обидві концепції співпраці та спільній відповідальності між командами, включаючи розробників, тестувальників, інженерів з експлуатації та інших зацікавлених сторін. Також, не менш важливі фундаментальні риси Agile та DevOps проявляються у важливості постійного вдосконалення та циклах зворотного зв'язку протягом життєвого циклу розробки програмного забезпечення.

При використанні гібридної моделі Agile DevOps організуються крос-функціональні команди, до складу яких входять члени з різними наборами навичок. Крос-функціональні команди сприяють співпраці, спільному володінню та колективній відповідальності та можуть працювати в суміжних приміщеннях або віртуально, використовуючи інструменти та технології співпраці для ефективного спілкування та координації.

Реалізація гібридної моделі розробки Agile DevOps підтримується набором інструментів і практик, призначених для оптимізації співпраці, автоматизації процесів і підвищення загальної ефективності проекту. У своїй основі ця методологія наголошує на безперервній інтеграції команд розробки та операцій.

Системи контролю версій, такі як платформи Git, GitHub і GitLab, служать основою спільної розробки. Дозволяючи командам відстежувати зміни та ефективно керувати сховищами коду, ці платформи сприяють безперервній співпраці та процесам перевірки коду, що важливо для підтримки якості коду та узгодженості в робочих процесах Agile DevOps.

Інструменти безперервної інтеграції та доставки (CI/CD) автоматизують процес створення, тестування та розгортання змін програмного забезпечення, дозволяючи командам швидко та надійно надавати високоякісний код. Популярні інструменти CI/CD (Continuous Integration/Delivery), як-от Jenkins, CircleCI, Travis CI та GitLab CI/CD, оптимізують створення конвеєрів розгортання, забезпечуючи ретельне тестування змін коду та безпроблемне розгортання у виробничих середовищах.

Інструменти «Інфраструктура як код» (Infrastructure as Code – IaC), такі як Terraform, AWS CloudFormation і Ansible, дозволяють командам програмно надавати і керувати ресурсами інфраструктури за допомогою декларативних файлів конфігурації. Розглядаючи інфраструктуру як код, команди можуть досягти узгодженості, повторюваності та масштабованості під час створення

хмарних середовищ, спрощуючи керування та автоматизуючи зміни інфраструктури відповідно до принципів Agile DevOps.

Інструменти моніторингу та спостереження, зокрема Prometheus, Grafana та стек ELK (Elasticsearch, Logstash, Kibana), відіграють вирішальну роль у підтримці продуктивності та надійності програм у виробничих середовищах. Ці інструменти дають змогу командам контролювати ключові показники продуктивності, відстежувати працездатність системи та аналізувати дані журналу в реальному часі, полегшуючи проактивне виявлення проблем, усунення несправностей та оптимізацію продуктивності для забезпечення позитивної взаємодії з користувачем.

По суті, інтеграція цих інструментів і методів у методологію Agile DevOps сприяє спільному, автоматизованому та керованому даними підходу до розробки програмного забезпечення та операцій. Ефективно використовуючи ці технології, команди можуть прискорити цикли доставки, покращити якість продукції та швидко адаптуватися до мінливих вимог, сприяючи безперервним інноваціям і доставці цінностей у середовищі розробки хмарного програмного забезпечення.

Наступним чином були сформульовані вимоги щодо структури проекту, схематичне зображення яких представлено на рисунку 1.

Вимоги, що описують структуру проекту перетворюються в результати в беклог спринту, що складається з вісімнадцяти спринтів для кожного модуля веб-додатку з трьома-чотирма сторінками, що позначають функції модуля, як показано в таблиці 1.

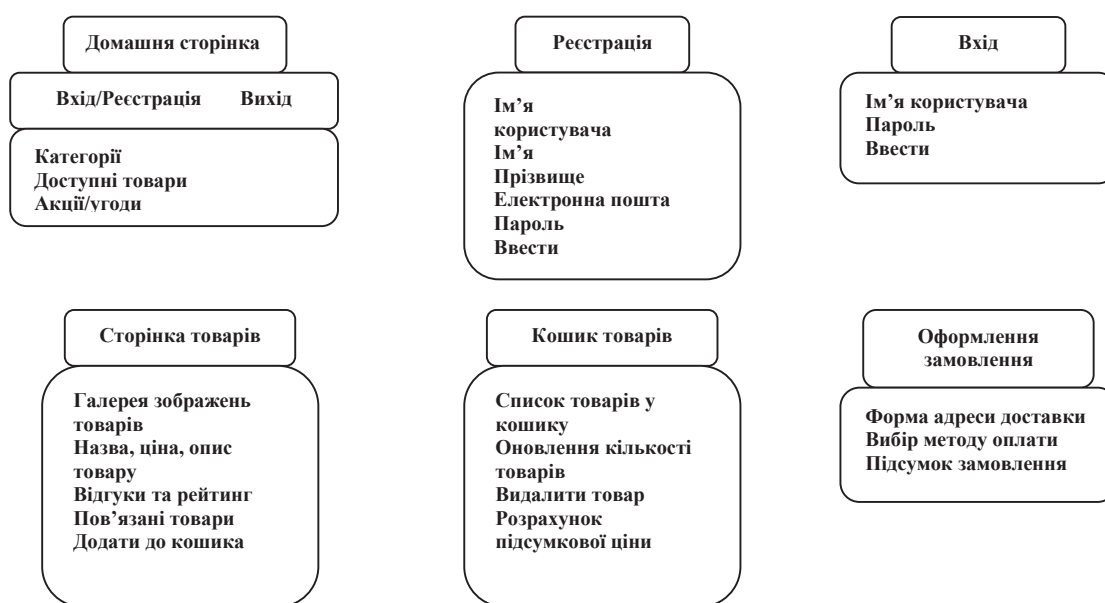


Рис. 1. Діаграма вимог до експериментального проекту

Беклог вимогу експериментального проекту

Спринт	Опис спринта	Сторі-поінт	Опис сторі-поінта
1	Фронт-енд	1	Структура фронт-енд системи
		2	Дизайн фронт-енд системи
		3	Навігація через фронт-енд систему
		4	Сторінка зі списком товарів
2	Автентифікація	5	Регістрація
		6	Вхід
		7	Функція «забув пароль»
		8	Вихід
3	Менеджмент товарів	9	Додати товар
		10	Редагувати товар
		11	Видалити товар
		12	Сторінка з інформацією про товар
4	Купувальний кошик	13	Додати до кошика
		14	Видалити з кошика
		15	Показати кошик
5	Оформлення замовлення	16	Заповнення форми адреси доставки
		17	Вибір методу оплати
		18	Підсумок замовлення

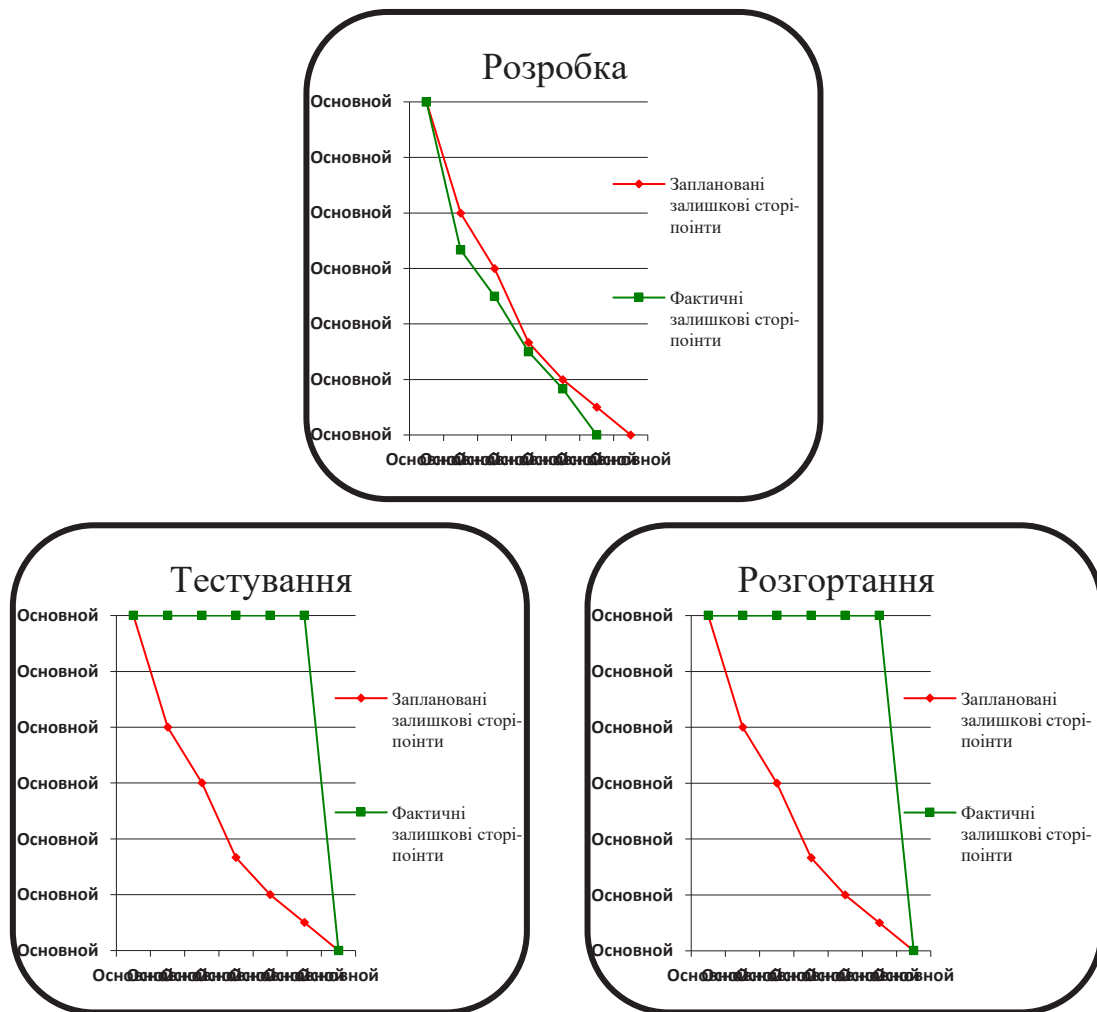


Рис. 2. Серія діаграм згорання задач з використанням методології Agile

Під час першої ітерації використовувалася методологія Agile, з метою призначення членам команди працювати окремо над секторами розробки, тестування та розгортання протягом семи днів. Спостережувані дані каталогізувалися щодо продуктивності та прогресу у формі діаграм згорання задач, що відображають фактичні та залишкові сторі-поінти у протигагу кількості днів, що пройшли. Діаграми цієї ітерації, які зображені на рисунку 2, відстежують ефективність секторів розробки, тестування та доставки за допомогою методології Agile.

На першій діаграмі цієї ітерації, видно, що кількість залишкових сторі-поінтів, постійно залишалася нижчою за очікувану їх кількість, що залишилися протягом усього проекту, і розробка була завершена до передбачуваного терміну на сьомий день, тобто що сектор розвитку працював краще, та закінчував раніше, ніж очікувалося. З іншої сторони, аналізуючи другу та третю діа-

граму серії, виявляється, що фактичні сторі-поінти, взагалі не зменшилися до останнього дня. Це означає, що сектори тестування та розгортання не змогли досягти жодного прогресу до завершення сектору розробки на шостий день, що свідчить про те, що спостережувана продуктивність не відповідає оціночним стандартам.

З цієї причини для наступної ітерації використовувалася гібридна методологія DevOps Agile, де члени крос-функціональних команд працювали разом протягом семи днів. Це призвело до безперервної співпраці та кооперації між членами команди, що дозволило скоординувати сектори розробки, тестування та розгортання без виконання високих вимог до навичок повного переходу на DevOps. Серія діаграм з використання даної методології зображена на рисунку 3.

На першій діаграмі ітерації з використанням методології DevOps Agile видно, що сектор розробки показав кращі результати, ніж очікувалося,

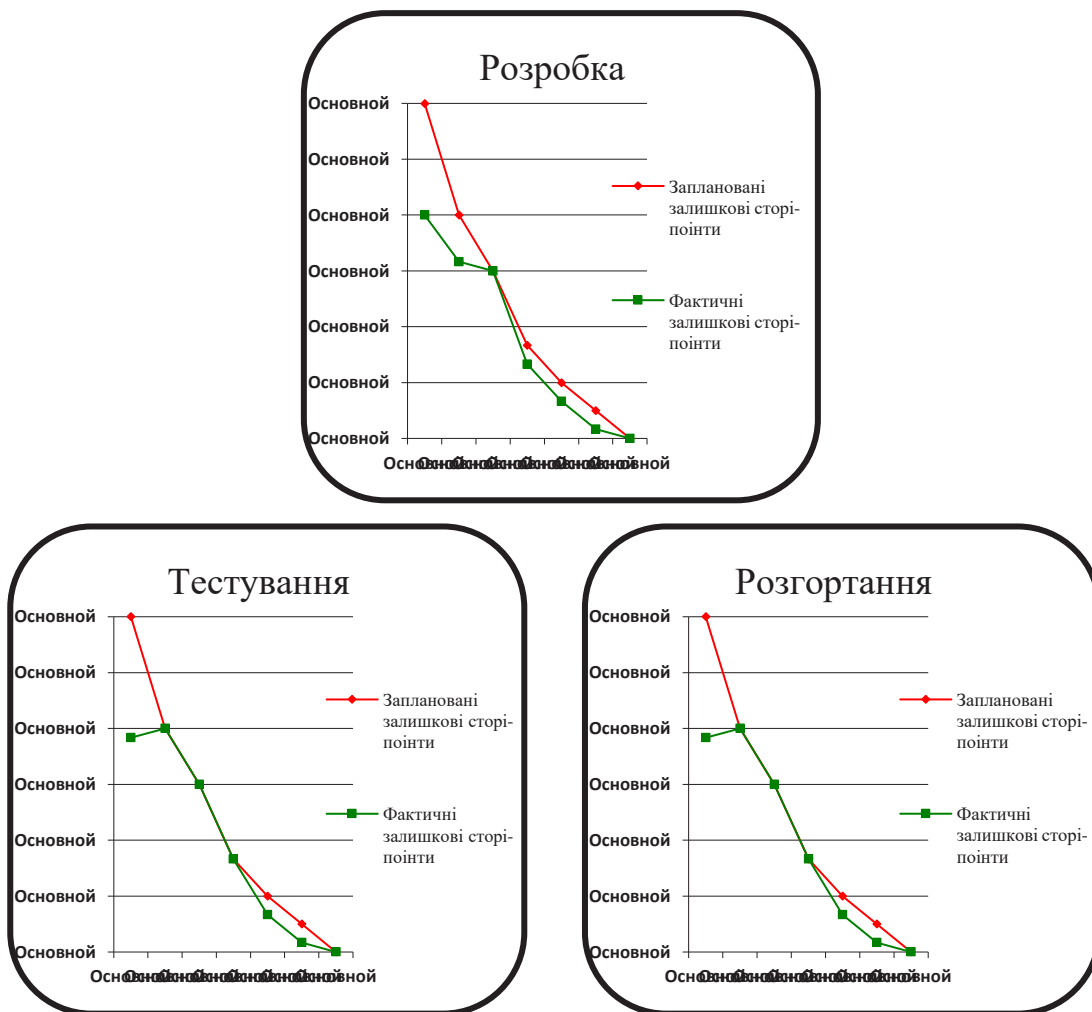


Рис. 3. Серія діаграм згорання задач з використанням гібридної методології DevOps Agile

подібно до попередньої ітерації, демонструючи швидший прогрес протягом усього проекту та досягнення завершення раніше розрахункового терміну. На другій і третій діаграмах цієї ітерації, виявляється, що фактичні залишкові сторі-поінти, які відповідають роботі секторів тестування та розгортання, здебільшого залишилися нижче оцінених сторі-поінтів, і досягли завершення раніше запланованого. Це означає, що сектори тестування та розгортання показали кращі результати, що призвело до послідовного та ефективного прогресу цілей проекту та завершення секторів тестування та розгортання до розрахункового терміну.

Таким чином, реалізацію експериментального проекту було завершено з використанням як методології Agile, так і методології DevOps Enabled Agile з діаграмами згоряння задач, створеними для обох ітерацій, що дозволило провести глибоке порівняння їх продуктивності, щоб визначити, що методологія DevOps Enabled Agile є кращою з точки зору прогресування у розробці проекту.

Висновки. Загалом, імплементація гібридної моделі Agile DevOps дозволяє організаціям прискорити час виходу на ринок шляхом оптимізації

процесів розробки, тестування та розгортання та скорочення часу. Конвеєри CI/CD автоматизують процес створення, тестування та розгортання змін програмного забезпечення, дозволяючи організаціям випускати нові функції та оновлення швидко та надійно. Методологія гнучкої розробки наголошує на ітераційних циклах розробки та частому зворотному зв'язку від зацікавлених сторін, що дозволяє командам повторювати функції продукту та включати відгуки користувачів на ранній стадії, що сприяє швидкому виходу на ринок. Інфраструктура як код дозволяє організаціям програмно надавати ресурси інфраструктури та керувати ними, скорочуючи час і зусилля, необхідні для розгортання та масштабування програм у динамічних хмарних середовищах.

У підсумку, імплементація у розробку експериментального проекту гібридної методології Agile DevOps із спільною роботою крос-функціональних команд, продемонструвала, що даний метод сприяє кращій координації та послідовному прогресу в усіх секторах, що призводить до покращення загальної продуктивності та своєчасного виконання завдань.

Список літератури:

1. Farid A. B., Helmy Y. M., Bahloul M. M. Enhancing Lean Software Development by using Devops Practices. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 2017. № 8.7. URL: <https://thesai.org/Publications/ViewPaper?Volume=8&Issue=7&Code=IJACSA&SerialNo=36> (accessed 26.06.2024)
2. Vemuri, Naveen & Thaneeru, Naresh & Tatikonda, Venkata. (2024). AI-Optimized DevOps for Streamlined Cloud CI/CD. *International Journal of Innovative Science and Research Technology*. 9. 7. 10.5281/zenodo.10673085.
3. Doris L., Potter K. Agile DevOps Practices: Implement agile and DevOps methodologies to streamline development, testing, and deployment processes. *Software*. 2024.
4. Al Masud S., Masnun M., Sultana A., Ahmed F., Begum N. DevOps Enabled Agile: Combining Agile and DevOps Methodologies for Software Development. *International Journal of Advanced Computer Science and Applications*. 2022. № 13. DOI:10.14569/IJACSA.2022.0131131.
5. Ghantous G., Gill A. An Agile-DevOps Reference Architecture for Teaching Enterprise Agile. *International Journal of Learning, Teaching and Educational Research*. 2019. № 18. P. 128-144. DOI:10.26803/ijlter.18.7.9.
6. Mohammad S. M. DevOps automation and Agile methodology. *SSRN Electronic Journal*. 2017. № 5. P. 946-949. DOI:10.1729/Journal.24060.
7. Almeida F., Simoes J., Ferreira Lopes S. F. Exploring the Benefits of Combining DevOps and Agile. *Future Internet*. 2022. № 14. 63 p. DOI:10.3390/fi14020063.
8. Rossberg J. Agile Project Management with Azure DevOps: Concepts, Templates, and Metrics. 2019. DOI:10.1007/978-1-4842-4483-8.
9. Gupta C., Fernández-Crehuet J. M., Gupta V. Citation: Measuring Impact of Cloud Computing and Knowledge Management in Software Development and Innovation. *Systems*. 2022. № 10. 151p. DOI:10.3390/systems10050151.
10. Nguyen T., Huynh T., Tran L. Agile Software Development with Cloud Computing. 2023. DOI:10.13140/RG.2.2.15440.99842.
11. Arimura Y., Ito M. Cloud Computing for Software Development Environment -In-house Deployment at Numazu Software Development Cloud Center. *Fujitsu scientific & technical journal*. 2011. № 47. P. 325-334.
12. Zhenlong P., Zhonghui O., Youlan H. The Application and Development of Software Testing in Cloud Computing Environment. 2012. P. 450-454. DOI:10.1109/CSSS.2012.119.
13. Khan H., Ali F., Nazir S. Systematic analysis of software development in cloud computing perceptions. *Journal of Software: Evolution and Process*. 2022. № 36. DOI:10.1002/smr.2485.

14. Ogala J., Mughele S. Agile Software Development Methodologies in Cloud Computing. № 1. 2022. P. 67-80. DOI:10.22624/AIMS/REBK2022-P7.
15. Panigrahi C., Mall R., Pati B. Software Development Methodology for Cloud Computing and Its Impact. 2021. DOI:10.4018/978-1-7998-3016-0.ch008.
16. Xia F. GIS Software Product Development Challenges in the Era of Cloud Computing. 2022. DOI:10.1007/978-981-19-3816-0_15.
17. Odun-Ayo I., Adesola F., Samuel V. Cloud Computing and Open Source Software - Issues and Developments. 2018. URL: https://www.iaeng.org/publication/IMECS2018/IMECS2018_pp140-145.pdf (accessed 26.06.2024)

Pundyk V.I. SOFTWARE SYSTEMS DEVELOPMENT IN THE CONTEXT OF CLOUD TECHNOLOGIES USING THE DEVOPS METHODOLOGY AND AGILE DEVELOPMENT

This paper explores the integration of Agile and DevOps methodologies, focusing on their similarities and benefits when combined. Agile promotes iterative and incremental development, providing small, frequent releases to quickly adapt to changing requirements. DevOps improves collaboration between development and operations teams by emphasizing continuous integration and continuous delivery (CI/CD) and automation.

The research includes a pilot project to demonstrate the functionality of the hybrid Agile DevOps model. The project consists of developing a cloud-based web-based e-commerce application in three stages: defining client requirements and creating a sprint pool, developing and testing project components, and delivering each component to the client.

The project structure includes eighteen sprints, each of which focuses on different modules of the web application, such as front-end development, authentication, product management, shopping cart, checkout, user profile. The article describes in detail the use of version control systems (GitHub, GitLab), CI/CD tools (Jenkins, CircleCI), infrastructure-as-code (IaC) tools (Terraform, AWS CloudFormation), and monitoring tools (Prometheus, Grafana) for efficient development, testing, and deployment.

The first iteration utilized the Agile methodology, assigning team members to specific sectors of development, testing, and deployment. Observations showed that while the development sector was completing tasks ahead of schedule, testing and deployment lagged behind, indicating a lack of coordination.

In the second iteration, a hybrid Agile DevOps methodology was implemented with cross-functional teams working together. This approach facilitated better coordination and consistent progress across all sectors, resulting in improved overall productivity and timely delivery of tasks.

The article concludes that the hybrid Agile DevOps methodology improves project development by facilitating continuous collaboration and integration, ultimately proving to be more effective than the traditional Agile approach.

Key words: agile development, DevOps methodology, continuous integration/continuous delivery, cross-functional teams, cloud e-commerce, infrastructure as code.