УДК 378:004.056.5
DOI: 10.31499/2307-4906.4.2025.345455

ІНТЕГРАЦІЯ ПРАКТИК DEVSECOPS У ПРАКТИЧНІ ЗАВДАННЯ ТЕМИ «ЗАХИСТ ПРОГРАМНИХ ПРОДУКТІВ» ПРИ ПІДГОТОВЦІ ФАХІВЦІВ З КОМП'ЮТЕРНИХ НАУК

Анна Пузікова, кандидат фізико-математичних наук, доцент, доцент кафедри інформаційних та цифрових технологій, Центральноукраїнський державний університет імені Володимира Винниченка.
ORCID: 0000-0002-6843-5583
E-mail: a.v.puzikova@cuspu.edu.ua
Ірина Лупан, кандидат педагогічних наук, доцент, доцент кафедри інформаційних та цифрових технологій, Центральноукраїнський державний університет імені Володимира Винниченка.
ORCID: 0000-0002-4791-0445
E-mail: ilupan@cuspu.edu.ua
Олена Присяжнюк, кандидат технічних наук, доцент, доцент кафедри інформаційних та цифрових технологій, Центральноукраїнський державний університет імені Володимира Винниченка.
ORCID: 0000-0002-7135-3124
E-mail: o.v.prysiazhniuk@cuspu.edu.ua

В умовах сучасного ринку ІТ-праці безпеку розглядають як ключовий компонент компетентностей випускника з комп'ютерних наук. У статті запропоновано систему практичних завдань з використанням технології DevSecOps для опанування студентами актуальних для розробника навичок інтеграції засобів безпеки у програмний продукт під час його життєвого циклу. Виконання таких завдань сприяє формуванню у студентів розуміння того, що заходи безпеки є невід'ємною складовою процесу розробки та розгортання програмних продуктів у рамках стратегії DevSecOps.
Ключові слова: *світогляд; мистецька діяльність; свідомі дії; художньо-образне відображення дійсності; саморозвиток особистості; художньо-рольова ситуація; формування здатності художньо-образного відображення дійсності; професійна підготовка; формування фахівця.*

INTEGRATION OF DEVSECOPS PRACTICES INTO APPLIED TASKS OF THE "SOFTWARE SECURITY" TOPIC IN THE TRAINING OF COMPUTER SCIENCE SPECIALISTS

Anna Puzikova, Candidate of Physical and Mathematical Sciences, Associate Professor at the Department of Information and Digital Technologies, Volodymyr Vynnychenko Central Ukrainian State University.
ORCID: 0000-0002-6843-5583

E-mail: a.v.puzikova@cuspu.edu.ua
Iryna Lupan, Candidate of Pedagogical Sciences, Associate Professor at the Department of Information and Digital Technologies, Volodymyr Vynnychenko Central Ukrainian State University.
ORCID: 0000-0002-4791-0445
E-mail: ilupan@cuspu.edu.ua
Olena Prysiazhniuk, Candidate of Technical Sciences, Associate Professor at the Department of Information and Digital Technologies, Volodymyr Vynnychenko Central Ukrainian State University.
ORCID: 0000-0002-7135-3124
E-mail: o.v.prysiazhniuk@cuspu.edu.ua

In today's IT job market, security is viewed as a key component of a computer science graduate's competencies, since security must be integrated into all software products. The aim of the article is to present practical tasks, the inclusion of which in the student training system will help them acquire the skills relevant for a software developer in integrating security into software products.
Traditionally, the security of software products was viewed as a separate phase of the development life cycle and was implemented by a separate security team. Modern software development requirements call for maintaining security measures at every stage of the software product life cycle. This is achieved, in particular, through the introduction of the DevSecOps methodology, which integrates security practices into the DevOps process. Students who have been introduced to DevOps technology during classes on collaborative software development are asked to create a new project on GitLab and reproduce all stages of the software development process. While completing each task, students must implement the corresponding security measures: ensure secure access control, configure project privacy settings, maintain secure work with local repository copies, define automated CI/CD stages, manage version control, and more.
According to the authors, completing these tasks helps students develop an understanding that security measures are an integral part of the software development and deployment process within the DevSecOps strategy. It was concluded that using tasks involving software product development based on the DevSecOps methodology aligns students' learning activities as closely as possible with real-world working conditions in an IT company. Further research can be directed toward the development of tasks for interdisciplinary comprehensive learning projects that maximize consideration of the needs of modern students and the demands of the labor market.
Keywords: *data protection; information security; software quality control; software life cycle; collaborative development; software pipeline; DevSecOps; GitLab; professional training of computer science specialists; professional competencies.*

У сучасних умовах стрімкого зростання кіберзагроз – від витоків даних до загроз з боку хакерів, ринок ІТ-праці висуває посилені вимоги до кваліфікації фахівців з комп'ютерних наук у розрізі кібербезпеки. Відповідно до аналітики спільноти програмістів щодо стану вакансій в ІТ-ринку [1], попит на фахівців із кібербезпеки стабільно зростає, а також зростають вимоги до обізнаності в питаннях інформаційної безпеки в усіх фахівців ІТ-галузі. Отже, розширюється коло актуальних питань із захисту/безпеки програмних продуктів (ПП) та даних, які потрібно врахувати у процесі підготовки фахівців.

Експерти міжнародної Асоціації обчислювальної техніки розглядають безпеку як ключовий компонент компетентностей випускника з комп'ютерних наук і наголошують на тому, щоб безпека була вбудована в усі їхні робочі продукти [2, с. 255]. Тому особливої уваги потребує розробка практичних завдань, виконання яких

допоможе студентам, які ще не мають професійного досвіду створення ПП, сформувати такі компетентності та усвідомити необхідність засвоєння теоретичних питань, пов'язаних із проєктуванням, розробкою та обслуговуванням систем захисту ПП.

Про зростання кількості вразливостей ПП та серйозну загрозу цих вразливостей для урядових та приватних організацій, і, відповідно, про необхідність організаціям впроваджувати питання безпеки в процес розробки своїх додатків з метою запобігання порушенням та поглиблення захисту ПП зазначається у роботі В. Сусукайла [3].

Аналіз останніх досліджень і публікацій, показує, що проблема підготовки майбутніх фахівців та підвищення кваліфікації вже працюючих спеціалістів залишається актуальною. Так у дослідженні М. Мілойкович (Milojković) та ін. [4] наведено результати опитування керівників 38 IT-компаній з чотирьох європейських країн (Сербії, Болгарії, Польщі, України) стосовно конкретних навичок, необхідних для профілю експерта з великих даних. В результаті дослідження зокрема з'ясувалося, що для IT-фахівців, які працюють з великими даними, важливими є компетентності в галузі інформаційної безпеки, такі як: застосовувати механізми та засоби контролю безпеки даних на кожному етапі їх обробки, управляти та впроваджувати засоби кібербезпеки, розробляти стратегії інформаційної безпеки, здійснювати захист IT-активів тощо.

Проблеми впровадження безпеки у продукт-орієнтовану розробку з використанням інструментів розподіленої командної цифрової взаємодії досліджується у статті О. Макарова та ін. [5]. Автори обговорюють питання контролю і управління безпекою та ризиками у DevOps-процесах під час розроблення наукомістких технологічних продуктів та встановлюють відповідність між цілями управління стратегією безпеки у методологіях виробничих процесів за спільної командної взаємодії «Вищий навчальний заклад – Бізнес» із цілями та засобами управління DevOps.

Стаття Аль Хашімі (Al Hashimi) та ін. [6] присвячена представленню нового способу підвищення кваліфікації команд розробників для досягнення найвищого рівня безпеки програмного забезпечення (ПЗ), тобто йдеться про розробку моделі прийняття рішень, використання якої дозволить керівникам проектів та зацікавленим сторонам визначити відповідні сфери, необхідні для покращення, та розробити правильні стратегії та дії для формування *безпечної та зрілої культури розробки*. У роботі В. Сусукайла [3], яка присвячена дослідженню можливостей використання підходу DevSecOps для аналізу сучасних загроз інформаційної безпеки в контексті важливих складових цього підходу, також зазначається необхідність постійного навчання команди з питань інформаційної безпеки.

У дослідженні С.-Ф. Вень (Wen) [7] наголошується, що проблеми з безпекою ПЗ значною мірою виникають через брак знань щодо захисту ПЗ серед розробників з різним досвідом та рівнем підготовки. Відмічається, що без фундаментальних знань з безпеки розробники все ще можуть налаштувати своє ПЗ, однак вони не завжди можуть виявляти проблеми під час кодування несправностей або розпізнавати вразливості після виявлення.

В роботі О. Вахули [8] пропонується вивчати тему «Безпека як код», використовуючи міждисциплінарний підхід. Стверджується, що включення цієї теми «в різні дисципліни, такі як програмування, системне адміністрування, управління

проєктами дозволяє студентам бачити цілісну картину та розуміти важливість безпеки в усіх аспектах».

Таким чином підготовка фахівців з комп'ютерних наук з питань безпеки є досить актуальною.

Мета статті – представлення практичних завдань з використанням DevSecOps-практик, включення яких у систему підготовки студентів допоможе їм опанувати актуальні для розробника ПП навички інтеграції безпеки у програмні продукти.

Ідея впровадження DevSecOps-практик в процес розробки ПП під час практичної підготовки бакалаврів на спеціальності «Комп'ютерні науки» виникла з досвіду викладання та сформувалась в процесі обговорення відповідної освітньо-професійної програми (ОПП) зі стейкхолдерами та студентами. Під час проходження «Навчальної практики з організації командної роботи при розробці програмного забезпечення» студенти спеціальності «Комп'ютерні науки» ознайомлюються зі стратегією співпраці між технічними командами, залученими до розробки ПП, яка отримала назву – DevOps (Development & Operation). Ця стратегія забезпечує ефективну взаємодію та подолання ізоляції між розробниками та фахівцями інформаційно-технологічного обслуговування [9]. DevOps реалізується через низку практик, що поєднують в собі методології та інструменти з метою покращення й пришвидшення процесів розробки, збірки, розгортання і моніторингу ПП, а також підвищення його якості [10–12]. Стратегія DevOps технічно реалізується через *конвеєр програмного забезпечення* (*pipeline*) – процес, що використовує автоматизацію та інструменти для полегшення руху через кожну фазу життєвого циклу розробки ПП. Для виконання завдань практики студентам пропонується використовувати GitLab – вебменеджер сховища Git, який надає конвеєр програмного забезпечення та інші функції, що допомагають керувати життєвим циклом розробки ПП. Таким чином, студенти набувають навичок командної роботи під час його спільної розробки, використовуючи можливість одночасно працювати над окремими фрагментами коду, не перериваючи інші етапи процесу розробки та тестування.

При вивченні дисципліни «Захист інформації» на спеціальності «Комп'ютерні науки» доцільно розглянути питання захисту розроблюваного ПП від помилок, вразливостей, неконтрольованих змін та потенційних дій зловмисників, спрямованих на його компрометацію, спираючись на отриманий досвід роботи з DevOps.

Традиційно безпека ПП розглядалась як окрема фаза життєвого циклу його розробки та впроваджувалась окремою командою безпеки [13]. Таким чином, вразливості та ризики безпеки могли залишатися без уваги або усуватися надто пізно під час розробки, що в подальшому могло призвести до необхідності впровадження дорогих та трудомістких рішень із забезпечення безпеки в сучасних умовах зростання частоти та складності кіберзагроз. Одним із сучасних підходів, який підтримують фахівці з компанії Google, і з яким, на думку авторів, слід ознайомлювати здобувачів вищої освіти, є запровадження методології DevSecOps, яка інтегрує практики безпеки в процес DevOps і забезпечує реалізацію та підтримку заходів захисту на всіх етапах життєвого циклу розробки ПЗ [14, 15]. DevSecOps підтримує принцип «безпека передусім», передбачаючи її інтеграцію на ранніх етапах розробки, тобто зміщення фокусу безпеки «вліво» – від фінальної стадії до початку процесу створення ПЗ.

Автоматизацію більшості процесів тестування та перевірки коду на наявність

вразливостей в стратегії DevSecOps забезпечує конвеєр програмного забезпечення CI/CD (Continuous Integration / Continuous Delivery and Continuous Deployment). Цей процес гарантує послідовність перевірки безпеки та функціональності ПП на таких фазах життєвого циклу його розробки, як написання коду, збірка, тестування, розгортання та оновлення. У конвеєрах програмного забезпечення автоматизація відіграє вирішальну роль. По-перше, програмне забезпечення, що розробляється, менш вразливе до людських помилок, коли автоматизовано такі завдання, як перевірка безпеки та функціональності. По-друге, тестування здійснюється протягом усього життєвого циклу розробки програмного забезпечення, а не тільки на фінальному етапі. GitLab підтримує роботу з CI/CD, тому його рекомендується використовувати студентам для практичного вивчення заходів безпеки при розробці додатків.

Під час розгляду питання про захист ПП в процесі його розробки доцільно звернути увагу студентів на захисні механізми GitLab, які використовуються на кожному з етапів.

На початковому етапі студенти застосовують знання з контролю доступу та автентифікації, набуваючи вмінь та навичок розподілу ролей для користувачів (TechLead, Developer, CI/CD Developer, QA Engineer), що визначають рівень прав на внесення змін до проєкту, а також використання двофакторної автентифікації для підвищення безпеки.

Наступним кроком є створення паралельних ліній розробки коду – гілок, які використовуються для ізоляції змін у коді, що дозволяє тестувати та перевіряти їх безпечно, запобігаючи випадковому внесенню помилок або вразливостей у стабільну версію продукту.

Під час завантаження коду студенти бачать, як безпека інтегрується на ранніх етапах: завантаження коду запускає автоматичну збірку, яка перетворює код у виконуваний образ. Образ проходить автоматичне сканування на уразливості та відповідність політикам безпеки. Автоматичне сканування вразливостей забезпечує безпеку образу. Якщо сканування виявляє проблеми при створенні образу, автоматизація запускає виправлення та повторно розгортає образ. Потім знову розгорнутий образ проходить повторне сканування на наявність уразливостей. Образ, який проходить цей раунд перевірок безпеки, зберігається в *сховищі артефактів* і є придатним для подальшого використання. Таким чином демонструється важливість автоматизації та контролю цілісності продукту. Також слід звернути увагу студентів на те, що попередньо налаштовані права керування ідентифікацією та доступом (IAM) до образів у репозиторії гарантують, що доступ до образу збірки матимуть лише авторизовані користувачі.

У той час, коли студенти завершили роботу над власними гілками і мають необхідність злити внесені зміни з різних паралельних ліній розробки в одну з головних гілок dev або main (*master*), здійснюється процес *контролю змін* (*merge request*); це надає можливість іншим членам команди переглянути код, залишити коментарі та перевірити відповідність стандартам, що запобігає внесенню помилок або вразливостей. у GitLab merge часто супроводжується запуском CI/CD-конвеєра, який виконує збірку, тести і сканування на уразливості. Це дозволяє виявити проблеми до того, як зміни потраплять у стабільну гілку. Перегляд роботи конвеєру і послідовності виконаних стадій, створених у файлі CI/CD продемонстровано на рисунку 1.

На цьому етапі студенти продовжують знайомитися із заходами безпеки, інтегрованими у DevSecOps-процес.

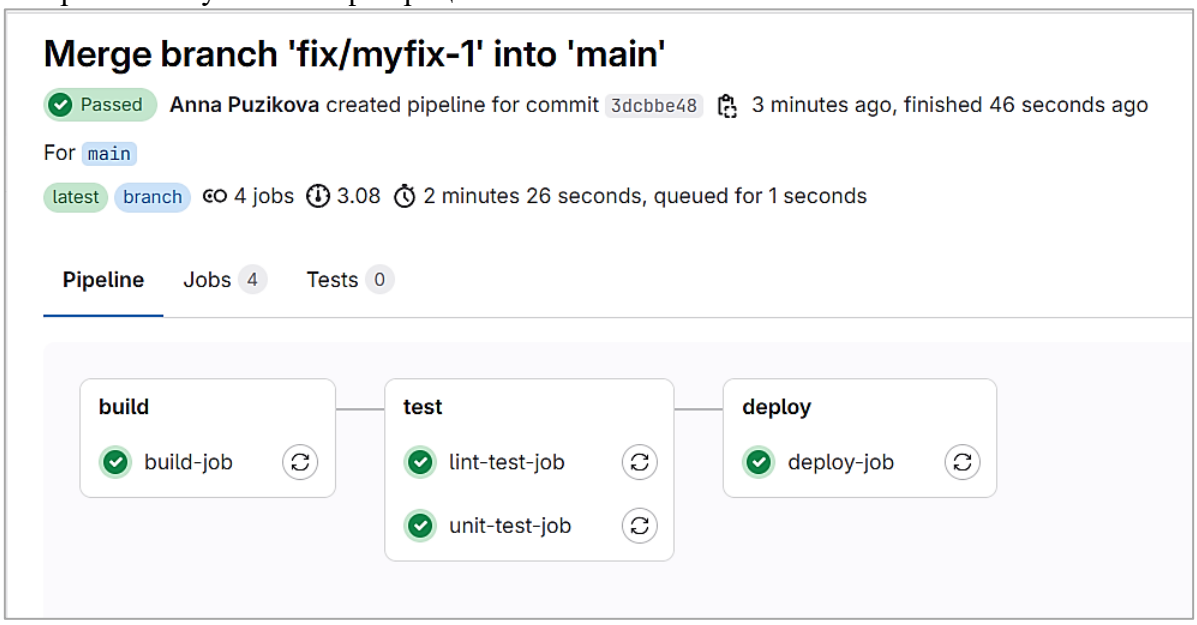

Рис. 1. Результати роботи CI/CD-конвеєра на кожній із стадій збірки, тестування і розгортання

Завдання для виконання та заходи безпеки, які реалізуються під час виконання цих завдань, наведені в таблиці 1.

Таблиця 1

Перелік завдань та впроваджених заходів безпеки

	Завдання	*Заходи безпеки*
1.	Зареєструватись на GitLab.	Забезпечення безпечного доступу через створення облікового запису та встановлення двофакторної автентифікації.
2.	Створити новий проєкт в GitLab.	Автоматичне застосування базових прав доступу та налаштування приватності проєкту для обмеження доступу сторонніх користувачів.
3.	Створити користувачів TechLead, Developer, CI/CD Developer, QA Engineer.	Визначення прав доступу для захисту репозиторію та управління змінами.
4.	Створити гілки проєкту (Dev, Main, Fix).	Ізоляція змін для безпечної розробки та тестування без впливу на стабільну версію.
5.	Встановити Git на локальну машину.	Гарантування цілісності та безпечної роботи з локальними копіями репозиторію.
6.	Клонувати репозиторій проєкту на локальний комп'ютер.	Використання захищеного з'єднання (HTTPS або SSH) для безпечного отримання коду.
7.	Створити файл .gitlab-ci.yml на локальній машині і завантажити його у проєкт в GitLab.	Визначення автоматизованих етапів CI/CD, включаючи тестування і сканування на уразливості для забезпечення безпечної збірки.
8.	Додати в проєкт файл з власним кодом і завантажити його у проєкт в GitLab.	Контроль версій та перевірка авторизації користувача перед внесенням змін; забезпечення ізоляції змін від стабільної гілки.
9.	Переглянути результати (логи) виконання робіт на стадіях конвеєру,	Перевірка успішності збірки, проходження тестів і результатів автоматичного сканування на безпеку.

	Завдання	*Заходи безпеки*
	описаних у файлі .gitlab-ci.yml.	
10.	Внести локально зміни у файл проєкту і завантажити їх у проєкт в GitLab.	Забезпечення безпечного контролю змін, створення комітів та можливість перевірки на конфлікти перед злиттям (merge).
11.	Переглянути результати (логи) виконання робіт на стадіях конвеєру, описаних у файлі .gitlab-ci.yml.	Контроль якості та безпеки коду, підтвердження, що внесені зміни не порушують політики безпеки та не містять вразливостей.

На думку авторів, виконання цих завдань сприяє формуванню у студентів розуміння того, що заходи безпеки є невід'ємною складовою процесу розробки та розгортання ПП у рамках стратегії DevSecOps. Студенти також отримують знання про те, що використання конвеєра програмного забезпечення сприяє захисту ПП від помилок, вразливостей та неконтрольованих змін, забезпечуючи автоматизоване тестування, контроль якості коду, безпечне об'єднання змін і розгортання лише перевірених версій у стабільному середовищі.

Використання завдань зі створення ПП на основі методології DevSecOps дає можливість максимально наблизити навчальну діяльність студентів до реальних умов роботи в IT-компанії. Колективна робота над проєктом дозволяє розглядати етапи розробки ПП – проєктування інформаційної системи, програмування коду, збірку, тестування, реліз, розгортання, підтримку проєкту, – з погляду організації командної роботи, розпаралелювання завдань, реалізації заходів безпеки ПП тощо. Це також сприяє більш глибокому розумінню складних концепцій розробки ПП, таких, як автоматизація процесів, контейнеризація та моніторинг і забезпечення надійності систем, та формуванню ключових компетентностей фахівця з комп'ютерних наук.

Подальші дослідження можуть бути спрямовані на розробку завдань для міжпредметних комплексних навчальних проєктів, які максимально враховують потреби сучасних студентів і вимоги ринку праці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Яких вакансій в IT стає більше. Огляд ринку праці, березень 2025. *Спільнота програмістів* | URL: https://dou.ua/lenta/articles/it-job-market-march-2025/ (дата звернення: 10.11.2025).
2. Computing Science Curricula 2023. The Joint Task Force on Computer Science Curricula. Association for Computing Machinery (ACM). IEEE-Computer Society (IEEE-CS). Association for the Advancement of Artificial Intelligence (AAAI). 2024. New York. 434 p.
3. Сусукайло В. А. Використання підходу DevSecOps для аналізу сучасних загроз інформаційної безпеки. *Кібербезпека: освіта, наука, техніка.* 2021. № 2(14). С. 26–35. DOI: 10.28925/2663-4023.2021.14.2635 (дата звернення 11.11.2025).
4. Milojković M., Stojanović D., Stojanović N, Kłos-Witkowska A., Witos K., Bernać M., Martsenyuk V., Dimitrov G., Rancic D., Bychkov O. *Specifyng graduate competencies in data science: research of employers*. 2022. P. 165–194. DOI:10.53052/9788367652018.07 (дата звернення 11.11.2025).
5. Макаров О., Борщ О., Брем В. Дослідження стратегічних підходів до забезпечення безпеки цифрових платформ університетського рівня під час використання сучасних методологій бізнес-взаємодії. *Вісник КрНУ імені Михайла Остроградського*. 2024. Вип. 6 (149). С. 112–118. DOI: 10.32782/1995-0519.2024.6.13 (дата звернення 12.11.2025).
6. Khan R. A., Hashimi H. A .Al., Alwageed H. S., Keshta Is., Almagrabi Al. O., Ayuni S. Securing Software Development through People Maturity: A Fuzzy-AHP Decision Making Framework. *Journal of Software: Evolution and Process*. 2025. Vol. 37, № 9. DOI:10.1002/smr.70045 (дата звернення 10.11.2025).
7. Wen S.-F. Context-Based Support to Enhance Developers' Learning of Software Security. *Education.*

Sciences. 2023, Vol. 13, № 7. DOI: 10.3390/educsci13070631 (дата звернення 11.11.2025).
8. Вахула О. Інтеграція принципів «безпека як код» у методики викладання дисципліни «Хмарні технології та їх захист»: підходи, виклики та перспективи. *Актуальні проблеми професійної педагогіки та освіти: досвід, новації, перспективи:* збірник матеріалів Міжнар. науково-практ. конференції (м. Львів, 25 квітня 2024 р.). С. 129–131. URL: https://files.znu.edu.ua/files/Bibliobooks/Inshi81/0060785.pdf#page=129 (дата звернення 12.11.2025).
9. Amaro R., Pereira R., da Silva M. M. DevOps Metrics and KPIs: A Multivocal Literature Review. *ACM Computing Surveys*. 2024. Vol. 56, № 9. P. 1–41. DOI:10.1145/3652508 (дата звернення 11.11.2025).
10. Da Gi˜ao H., Flores A., Pereira R., Cunha J. Chronicles of CI/CD: A Deep Dive into its Usage Over Time *arXiv preprint.* arXiv: https://arxiv.org/pdf/2402.17588. 2024 (дата звернення 12.11.2025).
11. 2021 State of DevOps Report. *Puppet Labs*. URL: https://media.webteam.puppet.com/uploads/2021/07/Puppet-State-of-DevOps-Report-2021.pdf (дата звернення 12.11.2025).
12. Пузікова А. Захист програмних продуктів на етапах розробки. *Цифрова гуманістика: Інформаційні технології та інформаційне моделювання на сучасному етапі розвитку суспільства*: матеріали II Всеукр. наук.-практ. конф., 22-23 травня 2025 р. Кропивницький: ЦДУ, 2025. С. 175–178.
13. Khan R. A., Khan S. U., Khan H. U., Ilyas M. Systematic literature review on security risks and its practices in secure software development. *IEEE Access.* 2022. Vol. 10. P. 5456–5481. DOI: 10.1109/ACCESS.2022.3140181 (дата звернення 12.11.2025).
14. Desai R., Nisha T. N. Best Practices for Ensuring Security in DevOps: A Case Study Approach. *Journal of Physics: Conference Series. Advances in Computer Science Engineering.* 2021. Vol. 1964. DOI: 10.1088/1742-6596/1964/4/042045 (дата звернення 11.11.2025).
15. Empowered development: GitLab on Google Cloud for streamlined delivery and enhanced security. URL: https://cloud.google.com/blog/topics/partners/understand-the-google-cloud-gitlab-integration (дата звернення 12.11.2025).

REFERENCES

1. What IT vacancies are becoming more numerous. (2025). Labor market overview, March 2025. *Spilnota prohramistiv*. URL: https://dou.ua/lenta/articles/it-job-market-march-2025/ [in Ukrainian].
2. Computing Science Curricula 2023 (2024). *The Joint Task Force on Computer Science Curricula*. Association for Computing Machinery (ACM). IEEE-Computer Society (IEEE-CS). Association for the Advancement of Artificial Intelligence (AAAI). New York.
3. Susukailo, V. (2021) Use Of Devsecops Approach For Information Security Threats Analysis. *Kiberbezpeka: osvita, nauka, tekhnika, 2(14), 26–35*. DOI: 10.28925/2663-4023.2021.14.2635 [in Ukrainian].
4. Milojković, M., Stojanović, D., Stojanović, N, Kłos-Witkowska,A., Witos, K., Bernać, M., Martsenyuk, V., Dimitrov, G., Rancic, D., & Bychkov, O. (2022). *Specifyng graduate competencies in data science: research of employers, 165–194*. DOI:10.53052/9788367652018.07.
5. Makarov, O., Borshch, O., & Brem, V. (2024) Research Of Strategic Approaches To Ensuring The Security Of University-Level Digital Platforms Using Modern Business Interaction Methodologies. *Visnyk KrNU imeni Mykhaila Ostrohradskoho, 6 (149), 112–118*. DOI: 10.32782/1995-0519.2024.6.13 [in Ukrainian].
6. Khan, R. A., Hashimi, H. A .Al., Alwageed, H. S., Keshta, Is., Almagrabi, Al. O.,. & Ayuni, S. (2025). Securing Software Development through People Maturity: A Fuzzy-AHP Decision Making Framework. *Journal of Software: Evolution and Process, 37(9)*. DOI:10.1002/smr.70045.
7. Wen, S.-F. (2023). Context-Based Support to Enhance Developers' Learning of Software Security. *Education. Sciences, 13 (7)*. DOI: 10.3390/educsci13070631.
8. Vakhula, O. (2024) Integration of the principles of "security as code" in the teaching methodology of the discipline "Cloud Technologies and Their Protection": approaches, challenges and prospects. *Aktualni problemy profesiinoi pedahohiky ta osvity: dosvid, novatsii, perspektyvy:* zbirnyk materialiv mizhnarodnoyi naukovo-praktychnoyi konferentsiyi (pp. 129–131). Lviv. URL: https://files.znu.edu.ua/files/Bibliobooks/Inshi81/0060785.pdf#page=129 [in Ukrainian]
9. Amaro, R., Pereira, R., & da Silva, M. M. (2024). DevOps Metrics and KPIs: A Multivocal Literature Review. *ACM Computing Surveys, 56(9), 1–41*. DOI:10.1145/3652508.

10. Da Gi˜ao, H., Flores, A., Pereira, R., & Cunha, J. (2024). Chronicles of CI/CD: A Deep Dive into its Usage Over Time Preprint. URL: https://arxiv.org/pdf/2402.17588.
11. 2021 State of DevOps Report. *Puppet Labs*. https://media.webteam.puppet.com/uploads/2021/07/Puppet-State-of-DevOps-Report-2021.pdf.
12. Puzikova, A. (2025). Protection of software products during the development stages. *Tsyfrova humanistyka: Informatsiini tekhnolohii ta informatsiine modeliuvannia na suchasnomu etapi rozvytku suspilstva:* materialy Vseukrainskoi naukovo-praktychnoi konferentsii (pp. 175–178). Kropyvnytskyi [in Ukrainian].
13. Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2022). Systematic literature review on security risks and its practices in secure software development. *IEEE Access, 10, 5456–5481*. DOI: 10.1109/ACCESS.2022.3140181.
14. Desai, R., & Nisha, T. N. (2021). Best Practices for Ensuring Security in DevOps: A Case Study Approach. *Journal of Physics: Conference Series. Advances in Computer Science Engineering,* 1964. DOI: 10.1088/1742-6596/1964/4/042045.
15. Empowered development: GitLab on Google Cloud for streamlined delivery and enhanced security. URL: https://cloud.google.com/blog/topics/partners/understand-the-google-cloud-gitlab-integration.