

УДК 378.147:004.9:005.8

DOI <https://doi.org/10.12958/3083-6514-2025-4-94-103>

Лучкевич Михайло Михайлович,

кандидат фізико-математичних наук, доцент,
доцент кафедри інформаційних систем та мереж
Національного університету «Львівська політехніка»,
м. Львів, Україна.

luchkevychmm@gmail.com

<https://orcid.org/0000-0002-2196-252X>

ЦІЛЬОВИЙ БЛОК МОДЕЛІ ПЕДАГОГІЧНОЇ РАМКИ DEVOPS ЯК СИСТЕМОУТВОРЮВАЛЬНИЙ КОМПОНЕНТ ПРОФЕСІЙНОЇ ПІДГОТОВКИ ІТ-ФАХІВЦІВ

Цифрова трансформація освіти зумовлює потребу у формуванні нової парадигми підготовки ІТ-фахівців, орієнтованої на практику автоматизації, інтеграції та безперервного вдосконалення. Одним із ефективних шляхів досягнення цього є використання DevOps-підходу, який поєднує процеси розроблення, тестування, розгортання та моніторингу програмних продуктів у єдиному циклі.

Педагогічна адаптація DevOps дає змогу створювати освітні системи, у яких цілі, зміст і процес навчання узгоджуються з реальними виробничими процесами ІТ-індустрії. У центрі такої системи – цільовий блок моделі DevOps-освіти, який визначає стратегічні, тактичні та операційні орієнтири формування професійної компетентності майбутніх спеціалістів. Саме цей компонент забезпечує цілісність педагогічної системи та її спрямування на розвиток компетентностей, що відповідають викликам цифрової економіки.

Сучасний науковий дискурс свідчить про активне формування міждисциплінарного поля досліджень, у межах якого DevOps-методологія розглядається не лише як технологічний підхід до управління життєвим циклом програмного забезпечення, а й як культурно-організаційна та педагогічна парадигма (Allen, 1994).

Основи DevOps як методології безперервної інтеграції та постачання ґрунтовно висвітлено у книзі Н. Van Merode (2023) та праці М. L. Gupta et al. (2024), які розглядають DevOps як концепцію організації виробничого циклу на принципах автоматизації, узгодженості процесів і постійного вдосконалення.

G. Kim et al. (2021) у праці *The DevOps Handbook* наголошують на соціотехнічній природі DevOps, де технологічні інструменти є лише складником ширшої культури співпраці, відкритості та відповідальності.

DevOps трансформує не лише способи створення програмних продуктів, а й мислення команд, формуючи єдину систему цінностей, орієнтовану на ефективність, довіру та спільне навчання (López-Fernández et al., 2021).

Подібних висновків дотримуються El Aouni et al. (2025), які визначають DevOps як еволюцію інтеграційних практик, що поєднують філософію Agile, інструменти автоматизації та культуру постійного поліпшення.

М. S. Khan et al. (2022), провівши систематичний огляд літератури, вказують, що головним чинником успішності DevOps є не так технологічні рішення, як ефективна комунікація та узгодженість ролей між командами розроблення й експлуатації.

Таким чином, у технічному дискурсі DevOps трактується як інтеграційна модель управління знаннями, процесами та людьми.

Попередні дослідження здебільшого акцентують увагу на технологічних інструментах DevOps, залишаючи поза увагою педагогічну логіку цілепокладання. Бракує системних робіт, які б описували структуру цільового блоку DevOps-освіти, взаємозв'язки між стратегічними, тактичними й операційними цілями, а також показники їх реалізації.

Отже, актуальним є теоретичне осмислення цього блоку як методологічного підґрунтя формування професійної компетентності IT-фахівців.

Мета статті – розкрити зміст і структуру цільового блоку моделі DevOps-освіти та визначити його роль у формуванні професійної компетентності майбутніх фахівців з інформаційних технологій.

Дослідження ґрунтується на використанні таких методів: системного підходу – для розгляду DevOps-освіти як цілісної педагогічної системи; структурно-функціонального аналізу – для виокремлення компонентів цільового блоку та їх взаємозв'язків; моделювання – для графічного відображення структури блоку; контент-аналізу – для узагальнення педагогічних і технічних концепцій DevOps у наукових джерелах.

У сучасних умовах цифрової трансформації освіти DevOps розглядається не лише як технологічна практика автоматизації життєвого циклу програмного забезпечення, а й як педагогічна концепція, що забезпечує безперервне вдосконалення освітнього процесу.

Структура розробленої моделі DevOps-освіти включає чотири взаємопов'язані блоки: цільовий (визначає педагогічні орієнтири й принципи); змістовий (відображає навчальні модулі й компетентності); процесуально-технологічний (охоплює методи, інструменти та DevOps-практики); результативний (вимірює сформованість компетентностей).

У межах педагогічної рамки DevOps одним із ключових структурних компонентів виступає цільовий блок, який забезпечує логічну послідовність і системність навчально-професійної підготовки фахівців IT-галузі. Його зміст визначає педагогічну стратегію формування DevOps-компетентності, зокрема встановлення зв'язків між стратегічними, тактичними та операційними цілями освіти, їх реалізацію через систему принципів та оцінку результатів за допомогою показників KPI.

Рис. 1 відображає авторську структуру цього блоку, що функціонує як центральна ланка моделі педагогічної рамки DevOps.

На рис. 1 представлено структурно-логічну схему цільового блоку, який інтегрує мету, принципи та індикатори ефективності педагогічної діяльності, спрямованої на формування DevOps-компетентності студентів.

Стратегічні орієнтири DevOps-освіти формують ціннісно-смыслову основу педагогічної системи та визначають вектор професійного становлення майбутніх IT-фахівців. До них належать: DevOps-орієнтаційна готовність, соціальна відповідальність та спільотно-професійна культура.

Ці цілі мають світоглядний характер, оскільки задають напрям розвитку освіти не лише у площині знань, а й професійної ідентичності. Вони спрямовані на формування стійкої системи цінностей, що поєднує технічну компетентність із гуманістичними принципами взаємодії, відповідальності та відкритості – характеристиками культури DevOps.

DevOps-орієнтаційна готовність передбачає розуміння логіки взаємозалежності всіх етапів життєвого циклу програмного забезпечення та володіння інструментами CI/CD, контейнеризації, оркестрації (Singh et al., 2019). Педагогічно вона охоплює когнітивний, емоційно-ціннісний і діяльнісний рівні: усвідомлення принципів continuous integration, прийняття командної відповідальності та здатність діяти самостійно в умовах технологічної невизначеності.

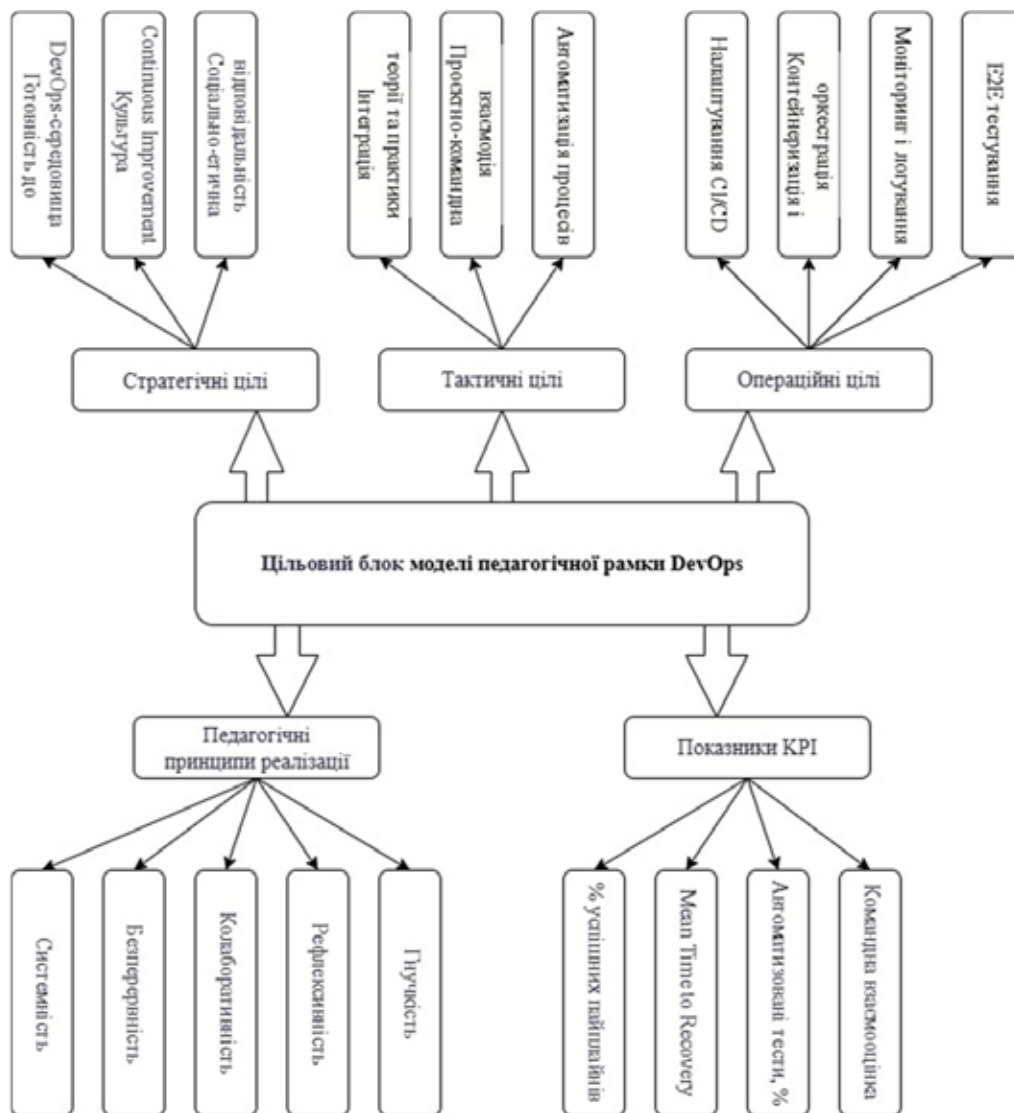


Рис. 1. Цільовий блок моделі педагогічної рамки DevOps

Формування цієї готовності відбувається через навчальні ситуації, у яких студент виконує повний цикл створення цифрового продукту – від планування до моніторингу, що сприяє інтеграції технічних і педагогічних аспектів навчання.

Соціальна відповідальність передбачає вміння приймати етичні рішення, діяти у спільних проєктах та усвідомлювати вплив технологій на суспільство. У DevOps-освіті ця мета пов’язана з розвитком професійної етики, інформаційної безпеки, чесності та відкритості до зворотного зв’язку. Вона формується через обговорення етичних кейсів, командну взаємодію та впровадження підходів Human-Centered DevOps, які поєднують технологічну ефективність із людським фактором.

Спільотно-професійна культура означає здатність діяти в мережевому середовищі знань і колективно вирішувати завдання. У DevOps-контексті це перехід від індивідуального до колаборативного навчання: студенти спільно працюють над кодом, проводять code-review, практикують pair programming.

Таке середовище розвиває соціальний інтелект, вміння працювати в гетерогенних командах і підтримувати професійне зростання інших, формуючи спільноту практиків: менторів, викладачів і студентів.

Узагальнюючи, стратегічні цілі DevOps-освіти утворюють методологічний фундамент моделі, орієнтуючи педагогічний процес на розвиток професійної ідентичності, системного бачення взаємодії «людина – технологія – команда» та узгодження педагогічних і професійних цінностей у єдину концептуальну рамку DevOps-освіти – як культури безперервного вдосконалення.

Тактичний рівень цілей у педагогічній рамці DevOps виконує роль операціоналізації стратегічних орієнтирів, конкретизуючи загальні педагогічні ідеї через практичні завдання навчального процесу. Якщо стратегічні цілі визначають «що» і «навіщо» формувати у майбутнього IT-фахівця, то тактичні уточнюють «як» цього досягти.

У моделі DevOps-освіти цей рівень охоплює чотири напрями: інтеграцію теорії та практики, проектну взаємодію, автоматизацію процесів та налаштування CI/CD-пайплайнів.

Інтеграція теорії та практики спрямована на подолання розриву між академічними знаннями й інженерною практикою. Навчання розглядається як процес моделювання реальних виробничих сценаріїв, у якому знання набувають цінності через застосування. Це досягається завдяки використанню *problem-based* і *project-based learning*, створенню навчальних лабораторій GitLab CI/CD, Jenkins, Docker, Kubernetes та симуляцій реальних інцидентів.

Педагогічно така інтеграція ґрунтується на конструктивістській парадигмі (Angraini et al., 2024), за якою студент не лише виконує завдання, а й бере участь у повному DevOps-циклі – від планування архітектури до релізу та моніторингу.

Проектна взаємодія формує здатність працювати у команді на засадах спільної відповідальності – одній із ключових характеристик DevOps-культури. Навчання організовується у форматі спринтів, хакатонів і групових лабораторій, де студенти виконують різні ролі (розробників, тестувальників, DevOps-інженерів). Викладач виступає Scrum Master або Product Owner (Pilster, 2025), координуючи спільну діяльність.

Проектна робота сприяє розвитку навичок комунікації, лідерства, управління ризиками та формує компетентності командної співпраці.

Автоматизація процесів у DevOps-освіті розглядається не лише як технічний інструмент, а й як педагогічний механізм підвищення ефективності навчання, що зменшує рутину та акцентує увагу на творчих і аналітичних аспектах. Вона реалізується через створення автоматизованих конвеєрів CI/CD, використання скриптів та конфігураційних файлів, а також систем моніторингу ефективності освітніх проєктів.

Автоматизація забезпечує принцип безперервного вдосконалення: кожна збірка чи тест створює зворотний зв'язок, на основі якого студент удосконалює власні рішення.

Тактичні цілі DevOps-освіти є містком між ціннісними орієнтирами та освітніми діями. Вони забезпечують перехід від філософії DevOps до практичної реалізації її принципів у навчальному середовищі, сприяють розвитку критичного мислення, гнучкості, здатності діяти автономно та спільно.

Таким чином, тактичний рівень виступає функціональною ланкою між концептуальним і операційним, визначаючи, як педагогічна система DevOps перетворює ідеї співпраці, автоматизації та безперервного вдосконалення на вимірювані навчальні результати.

Операційні цілі відображають практико-діяльнісний рівень реалізації педагогічної рамки DevOps, на якому стратегічні орієнтири та тактичні завдання втілюються через конкретні дії здобувачів освіти.

Якщо стратегічні цілі визначають «навіщо», а тактичні – «як», то операційні уточнюють «що робити» і «який результат отримати». Відповідно до моделі, цей рівень охоплює чотири складника: моніторинг і логування, оркестрацію контейнерів і сервісів, налаштування CI/CD-процесів та E2E-тестування.

Вони реалізують принципи continuous feedback, automation і integration, які у педагогічному вимірі відповідають самоконтролю, рефлексії та безперервному вдосконаленню.

Моніторинг і логування забезпечують прозорість і контроль якості навчальної діяльності. У DevOps вони спрямовані на виявлення помилок і реагування на збої, а в освіті – на відстеження динаміки формування компетентностей.

Моніторинг реалізується через аналітичні дашборди (Grafana, Kibana), навчальні журнали та метрики (time to deploy, success build rate, error density), які виконують роль освітніх KPI. Це формує рефлексивне мислення і здатність керувати власним навчанням.

Оркестрація контейнерів і сервісів розвиває вміння координувати взаємодію між компонентами проєкту. У DevOps це автоматизоване керування середовищем контейнерів (Docker, Kubernetes), а в педагогіці – організація складних навчальних процесів.

Реалізація передбачає використання контейнеризованих середовищ для лабораторних завдань, модульну структуру курсу та навчання координації командної роботи. Оркестрація формує системне мислення й вміння балансувати між автономією та колективною відповідальністю.

Налаштування CI/CD є ядром DevOps-педагогіки, що забезпечує автоматизований зв'язок між усіма етапами освітнього циклу. Технічно CI/CD охоплює інтеграцію змін, тестування, збірку, деплой і моніторинг, а педагогічно – відображає модель безперервного навчання.

Реалізація цієї цілі передбачає налаштування GitLab CI/CD pipelines, створення YAML-скриптів, аналіз логів та використання інтеграційних метрик (build time, deployment success rate, test coverage).

Студенти набувають автономності, відповідальності та рефлексії, що відповідає принципу Continuous Learning and Improvement.

E2E-тестування (end-to-end) завершує DevOps-цикл і забезпечує комплексну оцінку результатів навчання.

У технічному сенсі це перевірка працездатності системи «від початку до кінця», а у педагогічному – оцінювання рівня сформованості компетентностей, логіки мислення та якості командної взаємодії.

Тестування через інструменти (Cypress, Playwright, Postman) перетворюється на засіб не контролю, а навчання, коли студент аналізує власні дії й ефективність спільної роботи.

Таким чином, операційний рівень DevOps-освіти реалізує педагогічну логіку безперервного циклу «планування – виконання – оцінка – вдосконалення».

Моніторинг забезпечує прозорість, оркестрація – узгодженість дій, CI/CD – автоматизацію та інтеграцію, а E2E-тестування – рефлексивну оцінку результатів.

Ці цілі формують практичну компетентність DevOps-інженера, де кожна навчальна дія має зворотний зв'язок, аналітичну оцінку та потенціал до вдосконалення, що відповідає філософії DevOps as Learning Culture.

Принципи реалізації DevOps-освіти визначають педагогічну логіку функціонування моделі, забезпечують її узгодженість і стійкість. Вони формують методологічну основу, що поєднує освітній зміст із технологічними процесами DevOps.

У структурі моделі ці принципи виконують роль інваріантів, що пронизують усі рівні – від стратегічного до операційного.

До ключових належать п'ять взаємопов'язаних принципів: системність, безперервність, колаборативність, рефлексивність і гнучкість – педагогічні проєкції базових DevOps-ідей integration, automation, collaboration, feedback, improvement.

Системність визначає цілісність і взаємозв'язок усіх компонентів DevOps-освітнього процесу. У DevOps окремі етапи (планування, кодування, тестування, реліз) утворюють замкнений цикл; в освіті це означає безперервну взаємодію між метою, змістом, діяльністю та резуль-

татом. Системність реалізується через інтегровані навчальні модулі, спільну цифрову інфраструктуру (репозиторії, аналітичні панелі, системи оцінювання) та узгодження освітніх етапів із фазами DevOps-циклу (Plan → Code → Build → Test → Deploy → Monitor → Improve).

Цей принцип формує інтегративне мислення й забезпечує логічну єдність між усіма рівнями моделі.

Безперервність відображає ідею Continuous Everything – постійного вдосконалення. Вона означає ітераційний розвиток знань і компетентностей через регулярні оновлення завдань, самооцінювання та рефлексію. Реалізується через короткі навчальні спринти, принцип incremental learning та безперервний обмін результатами (continuous feedback loops). Безперервність формує культуру навчання впродовж життя, переходячи від епізодичного засвоєння до систематичного саморозвитку.

Колаборативність виражає соціально-командний характер DevOps-освіти. У технологічному контексті вона ґрунтується на спільній роботі з інструментами (Git, CI/CD, Jira, Slack), а в педагогічному – на організації навчання як спільної діяльності. Її форми: командні проекти, хакатони, взаємооцінювання, колективні retrospective meetings. Колаборативність розвиває комунікативність, відповідальність і вміння приймати колективні рішення, формуючи DevOps-культуру довіри й відкритості.

Рефлексивність є педагогічним утіленням принципу feedback & improvement. Вона означає здатність студента аналізувати власну діяльність, оцінювати результати та вдосконалювати дії. Реалізується через регулярні retrospective sessions, цифрові дашборди самооцінювання (GitLab, SonarQube) й ведення особистих журналів (DevLog). Рефлексивність формує метапізнання – усвідомлення власного способу мислення, що є основою аналітичного підходу до навчання.

Гнучкість відображає здатність DevOps-освіти адаптуватися до змін технологій і потреб студентів. Подібно до agile-підходів до розроблення, вона забезпечує динамічну перебудову змісту, методів і форм навчання. Гнучкість реалізується через модульну структуру курсів, microlearning, micro-credentials і короткі навчальні ітерації з відкритим плануванням. Цей принцип сприяє індивідуалізації навчання, розвитку адаптивності та підтриманню мотивації студентів.

Сукупність цих принципів формує інноваційну педагогічну систему DevOps-освіти, у якій технічна логіка інженерних процесів поєднується з педагогічною логікою розвитку компетентностей. Вони перетворюють DevOps із технологічної методології на філософію освітнього вдосконалення, у якій навчання, діяльність і розвиток утворюють єдиний процес Continuous Learning and Improvement.

Показники ефективності (Key Performance Indicators – KPI) у педагогічній рамці DevOps виступають інструментом кількісного та якісного вимірювання результативності навчального процесу. Вони забезпечують зворотний зв'язок між стратегічними цілями, тактичними завданнями та операційними діями, формуючи основу для аналітики, оцінювання й удосконалення освітнього середовища.

На відміну від традиційного оцінювання система KPI у DevOps-освіті має процесний характер: вона фіксує не лише кінцевий результат, а й ефективність освітніх дій, їх послідовність і вплив на розвиток компетентностей. Це дає змогу оцінювати не лише що зроблено, а й як цього досягнуто, тобто поєднати педагогічні, технічні й поведінкові параметри навчальної діяльності.

Педагогічна природа KPI полягає у вимірюванні трьох аспектів освітнього процесу: цільової узгодженості – відповідності навчальних результатів стратегічним орієнтирам DevOps-освіти; процесної ефективності – рівня організації командної взаємодії, автоматизації, оркестрації та тестування; рефлексивної результативності – здатності студентів аналізувати власну діяльність і застосовувати механізми самовдосконалення.

До основних груп показників належать мотиваційні (рівень DevOps-орієнтованої готовності), технологічні (успішність CI/CD-процесів, середній час на усунення помилок), колаборативні (якість взаємооцінювання), аналітичні (глибина рефлексії, частота оновлення DevLog) та якісні (відсоток успішних тестів, рівень інтеграції знань).

Такі показники дають змогу комплексно оцінити як технічну компетентність, так і педагогічну зрілість здобувачів освіти.

Система DevOps-аналітики в освітньому процесі поєднує кількісні метрики з якісними характеристиками командної роботи.

Типові показники, запозичені з DevOps-практик: Build Success Rate, Deployment Frequency, Mean Time to Recovery (MTTR), Change Failure Rate, адаптуються як освітні індикатори, що відображають швидкість навчального реагування, глибину розуміння матеріалу та ефективність командної співпраці.

Показники рефлексивного типу, як-от Reflexivity Index чи Peer Review Score, фіксують не лише продуктивність, а й здатність студентів до саморегуляції, критичного мислення й комунікації.

У педагогічному сенсі KPI виконують три ключові функції:

1. Діагностичну – виявлення динаміки розвитку компетентностей і навчальних бар'єрів.
2. Мотиваційну – підтримку змагання, усвідомлення власного прогресу, формування внутрішньої відповідальності.
3. Управлінську – забезпечення зворотного зв'язку для вдосконалення освітніх стратегій.

Результати вимірювань візуалізуються через дашборди навчальної аналітики, які відображають рівень активності студентів, ефективність команд, швидкість виконання завдань та співвідношення помилок і успішних рішень. Це дає можливість викладачеві своєчасно коригувати навчальний процес, а студенту – спостерігати власну динаміку розвитку в режимі реального часу.

Педагогічна значущість системи KPI полягає у тому, що вона перетворює DevOps-підхід на механізм доказового навчання (evidence-based education), у якому аналітика використовується для підтримки процесів самонавчання та вдосконалення.

Застосування показників та індикаторів результативності створює умови для формування культури відповідальності, рефлексії та взаємопідтримки, роблячи освітній процес прозорим і керованим.

Таким чином, система KPI у DevOps-освіті забезпечує баланс між технічною точністю й педагогічною гуманістичністю, даючи змогу навчальному процесу діяти як самоналагоджуваній системі безперервного вдосконалення – Continuous Learning and Improvement.

У результаті проведеного дослідження узагальнено теоретико-методологічні засади побудови цільового блоку педагогічної рамки DevOps, який виступає системоутворювальним елементом моделі професійної підготовки майбутніх IT-фахівців.

DevOps у педагогічному вимірі розглядається як інтеграційна освітня парадигма, що поєднує технологічні, організаційні та ціннісні аспекти навчання. Її сутність полягає у перенесенні принципів безперервної інтеграції, автоматизації та співпраці у сферу освіти.

Цільовий блок DevOps-освіти визначає логіку формування освітніх цілей і завдань. Він узгоджує стратегічний (ціннісний), тактичний (методологічний) та операційний (практичний) рівні, забезпечуючи єдність між метою навчання та його результатами.

Реалізація цільового блоку спирається на систему педагогічних принципів: системності, безперервності, колаборативності, рефлексивності та гнучкості, які забезпечують узгодженість усіх компонентів освітнього процесу та підтримують цикл постійного вдосконалення.

Запропонована система показників ефективності та індикаторів результативності дає змогу вимірювати рівень сформованості DevOps-компетентностей за технічними, аналітичними та

рефлексивними критеріями. Це створює підґрунтя для доказового (data-driven) управління якістю освіти.

Модель цільового блоку DevOps-освіти сприяє підвищенню практичної спрямованості навчання, розвитку командної взаємодії та формуванню готовності студентів до безперервного професійного вдосконалення.

Подальші розвідки доцільно спрямувати на емпіричну перевірку ефективності моделі у різних освітніх середовищах; створення цифрових інструментів педагогічної аналітики КРІ; розроблення DevOps-орієнтованих мікрокваліфікацій для підготовки ІТ-фахівців.

Список використаної літератури

1. Allen J. A. The constructivist paradigm: Values and ethics. *Journal of Teaching in Social Work*. 1994. Vol. 8. Issue 1–2. P. 31–54. DOI: https://doi.org/10.1300/J067v08n01_03
2. Van Merode H. Continuous Integration (CI) and Continuous Delivery (CD): a practical guide to designing and developing pipelines. Berkeley, CA, USA: Apress, 2023. DOI: <https://doi.org/10.1007/978-1-4842-9228-0>
3. Gupta M. L., Puppala R., Vadapalli V. V., Gundu H., Karthikeyan C. V. S. S. Continuous integration, delivery and deployment: A systematic review of approaches, tools, challenges and practices. *International Conference on Recent Trends in AI Enabled Technologies*. Springer, Cham, 2024. P. 76–89. DOI: https://doi.org/10.1007/978-3-031-59114-3_7
4. Kim G., Humble J., Debois P., Willis J. The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations. Portland: IT Revolution Press, LLC, 2021.
5. López-Fernández D., Díaz J., García J., Pérez J., González-Prieto Á. DevOps team structures: Characterization and implications. *IEEE Transactions on Software Engineering*. 2021. Vol. 48, No. 10. P. 3716–3736. DOI: <https://doi.org/10.1109/TSE.2021.3102982>
6. El Aouni F., Moumane K., Idri A., Najib M., Jan S. U. A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits. *Information and Software Technology*. 2025. Vol. 177. Art. no. 107569. DOI: <https://doi.org/10.1016/j.infsof.2024.107569>
7. Khan M. S., Khan A. W., Khan F., Khan M. A., Whangbo T. K. Critical challenges to adopt DevOps culture in software organizations: A systematic review. *IEEE Access*. 2022. Vol. 10. P. 14339–14349. DOI: <https://doi.org/10.1109/ACCESS.2022.3145970>
8. Singh C., Gaba N. S., Kaur M., Kaur B. Comparison of different CI/CD tools integrated with cloud platform. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Noida, India, 2019. P. 7–12. DOI: <https://doi.org/10.1109/CONFLUENCE.2019.8776985>
9. Angraini L. M., Kania N., Gürbüz F. Students' proficiency in computational thinking through constructivist learning theory. *International Journal of Mathematics and Mathematics Education*. 2024. Vol. 2. No. 1. P. 45–59. DOI: <https://doi.org/10.56855/ijmme.v2i1.963>
10. Pilster J. Professional Scrum Master und Product Owner für Dummies. John Wiley & Sons, 2025.

References

1. Allen, J. A. (1994). The constructivist paradigm: *Values and ethics*. *Journal of Teaching in Social Work*, 8(1-2), 31–54. DOI: https://doi.org/10.1300/J067v08n01_03
2. Van Merode, H. (2023). Continuous Integration (CI) and Continuous Delivery (CD): a practical guide to designing and developing pipelines. Berkeley, CA, USA: Apress. DOI: <https://doi.org/10.1007/978-1-4842-9228-0>
3. Gupta, M. L., Puppala, R., Vadapalli, V. V., Gundu, H., & Karthikeyan, C. V. S. S. (2024). Continuous integration, delivery and deployment: A systematic review of approaches, tools, challenges and practices. In *International Conference on Recent Trends in AI Enabled Technologies*. (pp. 76–89). Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-59114-3_7

4. Kim, G., Humble, J., Debois, P., & Willis, J. (2021). *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations*. Portland: IT Revolution Press, LLC.
5. López-Fernández, D., Díaz, J., García, J., Pérez, J., & González-Prieto, Á. (2021). DevOps team structures: Characterization and implications. *IEEE Transactions on Software Engineering*, 48(10), 3716–3736. DOI: <https://doi.org/10.1109/TSE.2021.3102982>
6. El Aouni, F., Moumane, K., Idri, A., Najib, M., & Jan, S. U. (2025). A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits. *Information and Software Technology*, (177), 107569. DOI: <https://doi.org/10.1016/j.infsof.2024.107569>
7. Khan, M. S., Khan, A. W., Khan, F., Khan, M. A., & Whangbo, T. K. (2022). Critical challenges to adopt DevOps culture in software organizations: A systematic review. *IEEE Access*, (10), 14339–14349. DOI: <https://doi.org/10.1109/ACCESS.2022.3145970>
8. Singh, C., Gaba, N. S., Kaur, M., & Kaur, B. (2019). Comparison of different CI/CD tools integrated with cloud platform. In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. (pp. 7–12). Noida, India. DOI: <https://doi.org/10.1109/CONFLUENCE.2019.8776985>
9. Angraini, L. M., Kania, N., & Gürbüz, F. (2024). Students' proficiency in computational thinking through constructivist learning theory. *International Journal of Mathematics and Mathematics Education*, 2(1), 45–59. DOI: <https://doi.org/10.56855/ijmme.v2i1.963>
10. Pilster, J. (2025). *Professional Scrum Master und Product Owner für Dummies*. John Wiley & Sons.

Лучкевич М. М. Цільовий блок моделі педагогічної рамки DevOps як системоутворювальний компонент професійної підготовки ІТ-фахівців

У статті розкрито теоретико-методологічні засади побудови цільового блоку педагогічної рамки DevOps як системоутворювального елемента професійної підготовки майбутніх ІТ-фахівців. Обґрунтовано доцільність інтеграції DevOps-методології в освітній процес як педагогічної концепції, що поєднує технологічні, організаційні та ціннісні аспекти навчання. Метою дослідження є визначення змісту, структури та функцій цільового блоку, який забезпечує узгодження стратегічного, тактичного й операційного рівнів підготовки. Методологічну основу становлять системний підхід, структурно-функціональний аналіз і моделювання. З'ясовано, що стратегічні цілі DevOps-освіти формують ціннісно-сміслові орієнтири (орієнтаційна готовність, соціальна відповідальність, спільотно-професійна культура), тактичні – конкретизують шляхи досягнення результатів через інтеграцію теорії та практики, проєктну взаємодію, автоматизацію й налаштування CI/CD-процесів, а операційні – реалізують принципи безперервного зворотного зв'язку, оркестрації, моніторингу та тестування. Визначено систему педагогічних принципів: системності, безперервності, колаборативності, рефлексивності та гнучкості, які забезпечують цілісність моделі DevOps-освіти. Розроблена система показників ефективності дає змогу здійснювати доказове оцінювання рівня сформованості DevOps-компетентностей за технічними, аналітичними й рефлексивними критеріями. Зроблено висновок, що цільовий блок є концептуальним ядром педагогічної моделі DevOps, що поєднує технічну точність із гуманістичною спрямованістю освіти, сприяючи формуванню культури безперервного вдосконалення.

Ключові слова: DevOps-освіта, педагогічна модель, компетентності ІТ-фахівців, безперервне вдосконалення.

Luchkevych M. M. The target block of the DevOps pedagogical framework model as a system-forming component of professional training for IT specialists

The article reveals the theoretical and methodological foundations for constructing a target block of the DevOps pedagogical framework as a system-forming element of professional training for

future IT specialists. It substantiates the expediency of integrating the DevOps methodology into the educational process as a pedagogical concept that combines the technological, organizational, and value-based aspects of learning. The study aims to determine the content, structure, and functions of the target block to ensure coordination between the strategic, tactical, and operational levels of training. The methodological basis consists of a systematic approach, structural-functional analysis, and modeling. The study found that the strategic goals of DevOps education provide value-semantic guidelines (e.g., orientation readiness, social responsibility, and community-professional culture), the tactical goals specify ways to achieve results through theory-practice integration, project interaction, automation, and CI/CD process configuration, and the operational goals implement continuous feedback, orchestration, monitoring, and testing principles. Five pedagogical principles have been defined to ensure the integrity of the DevOps education model: systematicity, continuity, collaboration, reflexivity, and flexibility. The developed system of performance indicators enables an evidence-based evaluation of DevOps competency levels based on technical, analytical, and reflective criteria. It can be concluded that the target block is the conceptual core of the DevOps pedagogical model. It combines technical precision with a humanistic approach to education and contributes to the formation of a culture of continuous improvement.

Key words: DevOps education, pedagogical model, IT specialist competencies, continuous improvement.

Creative Commons Attribution 4.0
International (CC BY 4.0)



Дата першого надходження рукопису до видання: 28.10.2025

Дата прийнятого до друку рукопису після рецензування: 21.11.2025

Дата публікації: 26.12.2025 р.