

реалізувати власний творчий потенціал, розвинути самостійність та сформувати позитивне ставлення до праці. Залучення учнів до художньо-трудової діяльності у процесі навчання створює сприятливі умови для гармонійного розвитку особистості. Поєднання творчості, практичної діяльності та пізнавальної активності сприяє формуванню різнобічно розвинутої особистості, здатної до творчого мислення, самовираження та активної участі у суспільному житті.

### **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Зеленкова Н. Формування пізнавальної самостійності старшокласників у навчальному процесі. Рідна школа. 2005. № 8. С. 31–33.
2. Кузьмінський А. І., Омеляненко В.Л. Педагогіка: підручник. Київ: Знання-Прес, 2008. 447 с.
3. Оршанський Л.В., Матвісів Я.Я., Ясеницький В.Є. Педагогічні умови формування проектно-технологічної культури школярів на уроках технологій. *Молодь і ринок*. № 5 (213). 2023. С. 24 – 29.

**ПАЛАГУТА Катерина**

*кандидат економічних наук,*

*доцент кафедри інженерії програмного забезпечення та кібербезпеки*

*Державного торговельно-економічного університету*

### **ІНТЕГРАЦІЯ DEVOPS, CLOUD ТА GIT-ТЕХНОЛОГІЙ У ПІДГОТОВКУ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ**

Сучасна інженерія програмного забезпечення зазнає стрімких трансформацій, зумовлених цифровізацією, автоматизацією та глобальними змінами у методології створення програмних продуктів. Ринок праці дедалі більше орієнтується на фахівців, здатних працювати у високодинамічних середовищах, де цикли розроблення і доставки ПЗ скорочуються до годин, а не тижнів або місяців. У таких умовах DevOps-підходи, хмарні технології та Git як стандарт керування версіями стали невід’ємними складовими професійної компетентності інженера-програміста. Інтеграція відповідних інструментів і методологій у систему вищої освіти є не лише актуальною, але й критично необхідною. Вона забезпечує підготовку конкурентоспроможних фахівців, здатних працювати зі складними інфраструктурними рішеннями, впроваджувати CI/CD-процеси, управляти хмарними сервісами та співпрацювати у командних середовищах.

Метою дослідження є обґрунтування теоретичних засад та розроблення практичних рекомендацій щодо інтеграції DevOps-підходів, хмарних сервісів і Git-технологій у систему професійної підготовки інженерів-програмістів з метою формування у здобувачів вищої освіти сучасних ІТ-компетентностей, необхідних для роботи в умовах цифрової трансформації та високодинамічних середовищ розроблення програмного забезпечення.

DevOps об’єднує розробників (Dev) та спеціалістів з експлуатації (Ops) в єдиний безперервний процес створення та доставки програмного забезпечення. Це не просто набір інструментів, а зміна парадигми взаємодії в команді: культура

співпраці, прозорості, автоматизації та швидкої реакції на зміни. У підготовці інженерів-програмістів засвоєння DevOps сприяє формуванню розуміння життєвого циклу ПЗ від планування до експлуатації; розвитку навичок автоматизації рутинних операцій; здатності працювати в командах із розподіленими ролями; оволодінню сучасними CI/CD-підходами.

У межах освітнього процесу доцільним є використання таких DevOps-інструментів для навчання:

- CI/CD-платформ: GitHub Actions, GitLab CI/CD, Jenkins;
- інструментів контейнеризації та оркестрації: Docker, Kubernetes;
- систем моніторингу: Prometheus, Grafana;
- інфраструктури як коду (IaC): Terraform, Ansible.

Використання цих інструментів дозволяє студентам моделювати реальні сценарії промислової розробки та експлуатації ПЗ.

Хмарні рішення стали стандартом у розробці та розгортанні застосунків. Для майбутніх інженерів вони забезпечують можливість працювати зі змінними обчислювальними ресурсами, сервісами автоматизованого розгортання (PaaS), інструментами CI/CD у хмарі, контейнерними середовищами, серверлес-платформами. Освітні програми мають охоплювати такі сервіси, як Microsoft Azure, Amazon AWS, Google Cloud Platform, оскільки більшість ІТ-компаній використовує саме ці екосистеми. Використання хмарних платформ у навчанні забезпечує доступ до реальної інфраструктури без додаткових витрат на локальне обладнання; розвиває компетентності роботи з безпекою, масштабуванням та адмініструванням; формує здатність інтегрувати застосунки в глобальні хмарні екосистеми; дозволяє створювати повні канали CI/CD у хмарному середовищі. Окрему увагу варто приділяти вивченню Azure DevOps, AWS CodePipeline, Google Cloud Build.

Git-технології є фундаментом командної роботи та управління версіями. Git - це інструмент контролю версій у сучасній розробці. Знання Git — обов'язкова базова компетентність, без якої неможлива участь у командних проєктах. У межах освітнього курсу студенти мають оволодіти базовими операціями (clone, commit, push, pull, merge); розгалуженням та моделями гілок (Git Flow, GitHub Flow); роботою з pull requests; розв'язанням конфліктів; автоматизацією процесів через Git-hooks. Git є «точкою інтеграції» для CI/CD-конвеєрів; систем контейнеризації; хмарних сервісів розгортання; інфраструктури як коду. У підготовці програмістів варто приділяти увагу не лише командній роботі, а й інтеграції Git із DevOps-процесами.

Освітні програми мають передбачати формування таких компетентностей:

- технічні: володіння DevOps-інструментами, CI/CD, контейнеризацією, хмарними сервісами;
- організаційні: робота в команді, планування релізів, документування;
- аналітичні: оцінка продуктивності, оптимізація інфраструктури, моніторинг;

- практичні: створення повного циклу розроблення від коду до розгортання.

Ефективною є організація роботи через:

- проєктні завдання (створення сервісу та CI/CD-ланцюга);
- мікропроєкти у командах із використанням GitFlow;
- лабораторні роботи в Azure/AWS/GCP;
- хакатони та спринти за методологією Scrum;
- симуляції DevOps-процесів.

Студенти повинні мати можливість працювати з реальними репозиторіями, розгортати контейнери, створювати pipeline'и та тестувати застосунки в хмарному середовищі.

Навчання має бути побудоване модульно:

1. Git + командна розробка
2. Основи хмарних сервісів
3. Контейнеризація та оркестрація
4. CI/CD та автоматизація
5. Моніторинг та експлуатація

Обов'язковим елементом є міжпредметна інтеграція дисциплін: програмування, тестування, адміністрування, кібербезпека.

Інтеграція DevOps, хмарних та Git-технологій у підготовку інженерів-програмістів сприяє підвищенню якості професійних компетентностей; формуванню практичних навичок роботи із сучасними інструментами індустрії; підготовці фахівців, здатних працювати в умовах цифрової трансформації; адаптації освітнього процесу до вимог ринку праці. Впровадження таких підходів у вищій школі дозволяє створити сучасну, динамічну, практично орієнтовану модель професійної підготовки, що відповідає міжнародним стандартам та тенденціям розвитку ІТ-сектору.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wagner G., Thurner L. Rethinking How We Teach Git: Pedagogical Recommendations and Practical Strategies for the Information Systems Curriculum [Електронний ресурс] // *Journal of Information Systems Education*. – 2025. – Режим доступу: <http://jise.org/Volume36/n1/JISE2025v36n1pp1-12.pdf>
2. DevOps Training Institute. Top 20 DevOps Learning Resources for 2025 [Електронний ресурс]. – 2025. – Режим доступу: <https://www.devopstraininginstitute.com/blog/Top-20-DevOps-Learning-Resources>
3. DevOpsSchool. Top DevOps Resources to Accelerate Your Learning Journey [Електронний ресурс]. – 2025. – Режим доступу: <https://www.devopsschool.com/blog/top-devops-resources-to-accelerate-your-learning-journey/>
4. theDataOps. Best DevOps Resources: Learn, Implement, and Advance Your Career [Електронний ресурс]. – 2025. – Режим доступу: <https://www.thedataops.org/best-devops-resources/>