



Дата першого надходження статті: 11 березня січня 2026 р.  
Дата прийняття до друку статті після рецензування: 16 квітня 2026 р.  
Дата оприлюднення стаття: 28 травня 2026 р.

DOI: [https://doi.org/10.31891/2308-4081/2026-16\(1\)-17](https://doi.org/10.31891/2308-4081/2026-16(1)-17)

Аспірант **В'ЯЧЕСЛАВ БОЙКО**  
Хмельницький національний університет, Україна  
E-mail: [bviacheslav.12@gmail.com](mailto:bviacheslav.12@gmail.com)  
ORCID ID: <https://orcid.org/0009-0004-9443-0286>

### ПРОГНОСТИЧНІ ІДЕЇ БРИТАНСЬКОГО ДОСВІДУ ПРОФЕСІЙНОЇ ПІДГОТОВКИ БАКАЛАВРІВ З ПРОГРАМНОЇ ІНЖЕНЕРІЇ ДЛЯ МОДЕРНІЗАЦІЇ ІТ-ОСВІТИ В УКРАЇНІ

#### АНОТАЦІЯ

*У статті обґрунтовано можливості використання прогностичних ідей британського досвіду професійної підготовки бакалаврів з програмної інженерії для модернізації ІТ-освіти в Україні. Дослідження виконано у межах порівняльної педагогіки з опорою на компетентнісний підхід, результативно-орієнтоване проектування освітніх програм та концепцію практико-інтегрованого навчання. Емпіричну основу дослідження становив аналіз офіційних матеріалів 10 українських і 10 британських університетів, а також рамкових і професійних документів.*

*Встановлено, що британські програми характеризуються вищою модульною гнучкістю, сильнішою інтеграцією індустріального середовища в освітній процес, системним використанням автентичного оцінювання, а також чітким поєднанням фундаментальної комп'ютерної підготовки з командними та проектними практиками. Водночас українські програми демонструють міцну нормативну й фундаментальну основу, проте вони частіше зберігають жорстку структурну логіку, менш розгорнуту практико-орієнтовану траєкторію та слабку інституціалізовану взаємодію з роботодавцями на рівні дизайну й оцінювання програм. На цій підставі сформульовано прогностичні ідеї для України: перехід до багаторівневого проектування змісту підготовки, запровадження наскрізної проектної траєкторії, розбудова екосистеми автентичного оцінювання, інституціалізація партнерства з ІТ-індустрією, гнучка профілізація освітніх програм та посилення зовнішньої професійної валідації результатів навчання. Доведено, що британський досвід доцільно розглядати не як готову модель для перенесення, а як джерело окремих організаційно-педагогічних рішень, які можуть бути адаптовані до українського стандарту, ресурсів конкретного ЗВО та реальних запитів ІТ-ринку.*

*Визначено перспективи подальших досліджень, що полягають у розробленні інструментарію для системного узгодження результатів навчання з професійними ролями, профілями компетентностей та типовими робочими сценаріями сучасної програмної інженерії.*

**Ключові слова:** професійна підготовка, програмна інженерія, бакалавр, ІТ-освіта, Велика Британія, Україна, порівняльна педагогіка, освітня програма, практико-орієнтоване навчання, автентичне оцінювання, прогностичні ідеї.



## PROGNOSTIC IDEAS OF THE BRITISH EXPERIENCE OF BACHELOR'S TRAINING IN SOFTWARE ENGINEERING FOR THE MODERNIZATION OF IT EDUCATION IN UKRAINE

### ABSTRACT

*The article substantiates the possibilities of using prognostic ideas from the British experience of professional training for bachelor's degree in software engineering to modernize IT education in Ukraine. The study was conducted within the framework of comparative pedagogy, relying on the competence-based approach, outcomes-based design of educational programs, and the concept of work-integrated learning. The empirical basis of the study comprised an analysis of the official materials of 10 Ukrainian and 10 British universities, alongside framework and professional documents defining training benchmarks in the fields of computing and software engineering.*

*It was established that within the analyzed sample, British programs are characterized by higher modular flexibility, stronger integration of the industry environment into the educational process, systematic use of authentic assessment, and a clearer combination of fundamental computer training with team and project practices. At the same time, Ukrainian programs demonstrate a solid regulatory and fundamental basis. However, they more frequently retain a more rigid structural logic, a less developed practice-oriented trajectory, and weaker institutionalized interaction with employers at the level of program design and assessment.*

*On this basis, the following prognostic ideas for Ukraine are formulated: transitioning to a multi-level design of training content, implementing a cross-cutting project trajectory, developing an authentic assessment ecosystem, institutionalizing partnerships with the IT industry, flexible profiling of educational programs, and strengthening the external professional validation of learning outcomes. The article argues that the British experience should be considered not as a ready-made model for transferring, but as a source of individual organizational and pedagogical solutions that can be adapted to the Ukrainian standard, the resources of a specific higher education institution, and the real demands of the IT market.*

*The prospects for further research involve developing tools for systematically aligning learning outcomes with professional roles, competency profiles, and typical work scenarios in modern software engineering.*

**Keywords:** *professional training, software engineering, bachelor, IT education, United Kingdom, Ukraine, comparative pedagogy, educational program, practice-oriented learning, authentic assessment, prognostic ideas.*

### ВСТУП

Програмна інженерія належить до тих спеціальностей, у яких відставання освітнього змісту від практик галузі особливо швидко виявляється на рівні компетентностей випускника. Дослідження останніх років фіксують не лише потребу в міцній технічній підготовці, а й у розвинених комунікативних, командних, проєктних і рефлексивних вміннях, без яких випускник не здатний повноцінно включитися в сучасні процеси розроблення, тестування, супроводу та еволюції програмного забезпечення (Akdur, 2022; Assyne et al., 2022; Daun et al., 2023).

У новітніх підходах до підготовки програмного інженера принципово важливою є відмінність між індивідуальним програмуванням і, власне, програмною



інженерією як колективною, довготривалою та інструментально насиченою діяльністю. У документі CS2023 наголошено, що визначальними є не лише кодування, а передусім час, командна взаємодія, тестування, колаборація, комунікація, проєктування, супровід, еволюція й інструменти розроблення. Водночас значна частина таких умінь найкраще формується в практичних і проєктних форматах, особливо за співпраці з реальними партнерами й замовниками (ACM/IEEE-CS/AAAI Joint Task Force, 2023). Це узгоджується і з узагальненнями систематичних оглядів щодо викладання програмної інженерії, де проєктно-орієнтоване навчання та активні форми колективної роботи послідовно розглядаються як продуктивні механізми поєднання фундаментальної підготовки з вимогами професійної практики (Pažur Aničić & Starić, 2022; Nguyen, 2025).

Для України ця проблематика має особливу актуальність. Підготовка бакалаврів з програмної інженерії спирається на державний стандарт спеціальності 121 «Інженерія програмного забезпечення», який залишається нормативним ядром проєктування освітніх програм, хоча в актуалізованому переліку спеціальностей 2024 року цій спеціальності відповідає новий код F2 «Інженерія програмного забезпечення» (Міністерство освіти і науки України, 2018; Кабінет Міністрів України, 2024). Отже, українські програми функціонують в умовах, коли нормативна визначеність є водночас їхньою перевагою і обмеженням: вона забезпечує порівнюваність та фундаментальність, але не завжди гарантує достатню гнучкість, швидкість оновлення та глибину зв'язку з практиками ІТ-індустрії.

У цьому контексті британський досвід є цінним не як «готова модель» для механічного копіювання, а як зразок ефективного поєднання академічної автономії, рамкового регулювання, предметних орієнтирів, професійної акредитації, модульності, автентичного оцінювання та партнерства з роботодавцями. Так, Агентство забезпечення якості вищої освіти Великої Британії (QAA) прямо вказує, що освітні програми галузі Computing (комп'ютерні науки та суміжні напрями підготовки) мають інтегрувати розвиток працевлаштовуваності, можливості для автентичного оцінювання, прикладне навчання в реальному або змодельованому професійному контексті, а також взаємодію з роботодавцями під час розроблення й оновлення курсів (QAA, 2022; QAA, 2024). Британське комп'ютерне товариство (BCS), своєю чергою, фіксує професійний вимір цієї логіки через чіткі вимоги до випускного кваліфікаційного проєкту, оцінюваної виробничої практики і загальної відповідності програм професійним результатам навчання (BCS, 2022).

У цьому дослідженні порівняння використано не для формального зіставлення двох країн, а для виявлення тих елементів британської практики, які можуть мати прикладне значення для українських освітніх програм. Сучасна компаративістика дедалі більше орієнтується не на описове порівняння, а на виявлення структурних тенденцій, механізмів трансферу та умов адаптації іноземного досвіду до конкретного національного середовища (Neubauer, 2024). Саме в такому аналітичному ракурсі доцільно розглядати професійну підготовку бакалаврів із програмної інженерії у Великій Британії та її потенціал для реформування української ІТ-освіти.

#### **МЕТА ДОСЛІДЖЕННЯ**

Мета дослідження полягає в тому, щоб на основі порівняльно-педагогічного аналізу професійної підготовки бакалаврів з програмної інженерії в університетах України та Великої Британії обґрунтувати прогностичні ідеї й науково-методичні рекомендації щодо модернізації вітчизняної ІТ-освіти. Для досягнення мети було



визначено такі завдання: проаналізувати нормативні та методичні засади підготовки бакалаврів з програмної інженерії в Україні та Великій Британії; узагальнити особливості освітніх програм 10 українських і 10 британських університетів; визначити спільні та відмінні характеристики підготовки; сформулювати прогностичні ідеї британського досвіду, релевантні для модернізації ІТ-освіти в Україні.

### **ТЕОРЕТИЧНА ОСНОВА ТА МЕТОДИ ДОСЛІДЖЕННЯ**

Теоретичну основу дослідження становлять положення порівняльної педагогіки, компетентнісного підходу, результативно-орієнтованого проектування освітніх програм, а також концептуальні ідеї практико-інтегрованого навчання. Така рамка дає змогу розглядати освітню програму не як статичний перелік дисциплін, а як педагогічно спроектовану траєкторію формування компетентностей, у якій зміст, методи, оцінювання, практика та професійна соціалізація є взаємоузгодженими (Neubauer, 2024; Vuorikainen et al., 2025).

Методологічно важливим є уточнити й термінологічний аспект. У статті паралельно використано позначення «I21/F2». Код I21 відповідає чинному бакалаврському стандарту вищої освіти України зі спеціальності «Інженерія програмного забезпечення», тоді як F2 є новим кодом цієї спеціальності в оновленому переліку галузей знань і спеціальностей 2024 року. Таке подвійне позначення дає змогу коректно відобразити реальний стан українського нормативного поля та уникнути плутанини між стандартом і новою класифікацією спеціальностей (Міністерство освіти і науки України, 2018; Кабінет Міністрів України, 2024).

Аналітична рамка дослідження має багаторівневу структуру. На першому рівні використано національні та секторальні британські рамки як основу для інтерпретації загальної логіки побудови комп'ютерних програм: Subject Benchmark Statement for Computing та The Frameworks for Higher Education Qualifications of UK Degree-Awarding Bodies (QAA, 2022; QAA, 2024). На другому рівні застосовано стандарт CS2023 як інструмент виділення спільного computing-ядра. На третьому рівні залучено SWEBOOK v4.0 для операціоналізації власне програмно-інженерних знань і процесів, а також стандарти BCS та рамку SFIA для аналізу професійної, практичної та працевлаштувальної складових (ACM/IEEE-CS/AAAI Joint Task Force, 2023; IEEE Computer Society, 2024; BCS, 2022; SFIA Foundation, 2024a; SFIA Foundation, 2024b). Саме така модель дає змогу уникнути дисциплінарної редукції програмної інженерії до «просто комп'ютерних наук» і водночас зберегти британську предметну специфіку, де вона інтегрована у ширший загальногалузевий контекст.

Емпіричну базу становили офіційно доступні публічні документи та сторінки програм, зібрані впродовж січня – травня 2026 року. До аналізу включалися лише бакалаврські програми, які або прямо належали до програмної інженерії, або мали виразний програмно-інженерний профіль у межах комп'ютерних наук або загальної computing-галузі. Критеріями включення програм до вибірки були: наявність бакалаврської програми з програмної інженерії або виразного програмно-інженерного профілю в межах computing-галузі; доступність офіційного опису програми, навчального плану або переліку освітніх компонентів; наявність інформації про практичну підготовку, оцінювання або взаємодію з роботодавцями. З вибірки виключалися магістерські й докторські програми, короткі сертифікатні курси, а також програми, у яких програмна інженерія не була представлена як окремий профіль або змістовий компонент.



До британської вибірки увійшли такі університети: Шеффілдський університет (University of Sheffield), Единбурзький університет (University of Edinburgh), Саутхемптонський університет (University of Southampton), університет Глазго (University of Glasgow), Королівський університет Белфасту (Queen's University Belfast), Ланкастерський університет (Lancaster University), Королівський коледж Геллоуей (Royal Holloway University of London), університет Стратклайд (University of Strathclyde), університет Аберіствіта (Aberystwyth University) та Кардіфський університет (Cardiff University). Важливо, що ця вибірка спеціально поєднала як програми, де програмна інженерія представлена окремим ступенем, так і програми, де вона реалізується всередині ширшої computing-рамки. Така конфігурація точніше відображає британську практику і ще раз підтверджує доцільність використання не лише SWEBOOK, а й QAA Computing Benchmark та CS2023 як інструментів інтерпретації змісту.

До української вибірки увійшли такі заклади вищої освіти: Хмельницький національний університет, КПІ ім. Ігоря Сікорського, Національний університет «Львівська політехніка», Національний університет «Одеська політехніка», Луцький національний технічний університет, Національний університет «Запорізька політехніка», Вінницький національний технічний університет, Харківський національний університет радіоелектроніки, Львівський національний університет імені Івана Франка та Тернопільський національний технічний університет імені Івана Пулюя.

Методи дослідження: порівняльно-педагогічний аналіз, контент-аналіз офіційних документів і навчальних програм, метод педагогічної інтерпретації.

#### **ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ**

Порівняння показало, що відмінність між українською та британською моделями в межах дослідженої вибірки полягає не стільки в наявності чи відсутності окремих освітніх компонентів, скільки в самій логіці їхнього педагогічного складання. Українські програми переважно будуються від державного стандарту до набору освітніх компонентів, тоді як британські – від рамкових і професійних орієнтирів до модульної архітектури та способів демонстрації результатів навчання. Внаслідок цього британські програми, як правило, демонструють більшу свободу профілювання, сильнішу підпорядкованість оцінювання практичним результатам та глибше занурення в індустріальний контекст, тоді як українські – більшу нормативну впорядкованість і фундаментальність, але менш виражену гнучкість освітньої траєкторії.

Передусім привертає увагу різниця у нормативній архітектурі. Український бакалаврський стандарт 121 задає достатньо чітке предметне поле: об'єктом визначено програмне забезпечення, процеси, інструментальні засоби та ресурси його розробки, супроводу й забезпечення якості; ціль навчання пов'язується з підготовкою фахівця, здатного розв'язувати завдання розробки, супроводу й якості програмного забезпечення (Міністерство освіти і науки України, 2018). Британський підхід є менш предметно обмеженим: QAA не встановлює жорсткого нормативного ядра для computing-галузі, а натомість визначає рівневі очікування та кінцеві результати навчання. Водночас національні кваліфікаційні рамки наголошують на автономії інституцій із правом присудження ступенів щодо формування власних освітніх програм та підтримки академічних стандартів (QAA, 2022; QAA, 2024). Саме тому в дослідженій британській вибірці співіснують як окремі спеціалізовані програми з програмної інженерії, так і програми типу «Computer Science (Software Engineering)» або «Applied Software Engineering», що демонструє ширшу свободу інституційного профілювання.



Не менш суттєвою є різниця у структурі програм. Українські програми в дослідженій вибірці загалом зберігають класичну 240-кредитну логіку бакалаврського рівня. Водночас британські програми набагато активніше працюють із варіативністю тривалості й маршруту: типова бакалаврська програма з відзнакою триває три роки. Проте до неї можуть додаватися такі складники, як рік виробничої практики в індустрії, навчання за кордоном у межах академічної мобільності або інтегрована магістерська траєкторія. Університети на кшталт Королівського університету Белфасту, Королівського коледжу Гелловей, університету Аберіствіта, Кардіфського університету та Ланкастерського університету виразно виносять у публічний опис програми стажування, виробничу практику, «рік в індустрії» або аналогічні компоненти, тоді як в Україні вибірковість зазвичай не змінює базову логіку освітньої траєкторії, а лише варіює набір окремих дисциплін (QAA, 2024; Queen's University Belfast, 2026; Royal Holloway, University of London, 2026; Aberystwyth University, 2026; Cardiff University, 2026; Lancaster University, 2026).

Показовою є також відмінність у способах інтеграції практики. QAA прямо зазначає, що освітні програми галузі computing часто передбачають роботу в компаніях у статусі стажистів або практикантів, а реалізація курсів має забезпечувати прикладне навчання в автентичному чи змодельованому професійному контексті. Це охоплює виробничу практику в індустрії, дуальну освіту, масштабні командні проекти та безпосередню співпрацю з промисловими партнерами (QAA, 2022). BCS додатково уточнює: якщо виробнича практика передбачена програмою, вона має бути її формально оцінюваним складником, який здебільшого реалізується як один цілісний блок (BCS, 2022). У межах проаналізованої вибірки ця логіка набуває цілком конкретних форм: Кардіфський університет декларує реальні проекти з клієнтами та дві оплачувані літні виробничі практики (placements); Королівський університет Белфасту акцентує проектно-орієнтоване навчання (project-based learning) та траєкторію зі стажуванням (placement pathway); Королівський коледж Гелловей пропонує «рік в індустрії» (year in industry); університет Аберіствіта пов'язує програмну інженерію з інтегрованим роком виробничої практики (integrated year in industry); Ланкастерський університет та університет Глазго наголошують на здобутті реального професійного досвіду (Cardiff University, 2026; Queen's University Belfast, 2026; Royal Holloway, University of London, 2026; Aberystwyth University, 2026; Lancaster University, 2026; University of Glasgow, 2026). В українських програмах практика також є обов'язковою або фактично передбачуваною складовою, однак вона частіше зберігає форму окремого етапу, а не наскрізного механізму організації всієї програми, хоча окремі позитивні приклади партнерської взаємодії вже присутні, зокрема, в ЛНУ імені Івана Франка, де програма пов'язується з можливістю практики в компаніях Львівського ІТ Кластера, у ВНТУ, де прямо названо установи-партнери, і в ХНУ, де акцент зроблено на командній розробці та управлінні проектами (Львівський національний університет імені Івана Франка, 2026; Вінницький національний технічний університет, 2026; Хмельницький національний університет, 2025).

Особливо виразною є відмінність у підходах до оцінювання. У QAA наголошено, що планування курсів має передбачати залучення роботодавців до визначення можливостей для автентичного оцінювання, а BCS вимагає випускний кваліфікаційний проект (major individual project), який демонструє здатність студента застосовувати практичні й аналітичні навички програми в цілому, критично



оцінювати власні рішення, а в разі групової роботи – чітко фіксувати індивідуальний внесок (QAA, 2022; BCS, 2022). У межах дослідженої вибірки британські програми послідовно виносять у публічний опис командні проекти (team projects), завдання на замовлення клієнтів (client-based tasks), практичні комп'ютерні проекти (practical computing project), промисловий інструментарій (industrial-strength tooling), реальні проекти (real-world projects), проектно-орієнтоване навчання (project-based learning), оцінювання виробничої практики (placement assessment) тощо, тобто оцінювання, зорієнтоване не лише на перевірку знань, а на демонстрацію професійної дії в контексті, максимально наближеному до роботи програмного інженера (Cardiff University, 2026; Queen's University Belfast, 2026; Royal Holloway, University of London, 2026; University of Strathclyde, 2026). В Україні ж у відкрито оприлюднених описах програм частіше фігурують традиційні параметри – освітні компоненти, компетентності, результати навчання, кваліфікаційна робота, інколи курсова підготовка, однак набагато рідше деталізується саме архітектура оцінювання, роль рецензування програмного коду, взаємного оцінювання, портфоліо чи презентацій для замовників. Це не означає відсутності таких практик, але свідчить про їх слабшу інституційну видимість на рівні проектування освітніх програм.

Крім того, істотно відрізняється ступінь наскрізної професійної соціалізації. У британських документах здатність до працевлаштування (employability) розглядається як вбудований складник. QAA прямо підкреслює необхідність інтеграції навичок працевлаштовуваності у курси, а стандарт CS2023 фактично зміщує фокус освіти у бік командної роботи, колаборації, комунікації, тестування, проектування, супроводу, еволюції та інструментарію (QAA, 2022; ACM/IEEE-CS/AAAI Joint Task Force, 2023). Рамка SFIA додатково посилює цей вектор, оскільки описує зв'язок між освітніми результатами та реальними професійними вміннями й рівнями відповідальності. Це надає університетським програмам можливість чіткіше перекладати результати навчання мовою професійних ролей і навичок (SFIA Foundation, 2024a; 2024b). Українські програми також містять компетентності, пов'язані з командною роботою, управлінням проектами й забезпеченням якості, але в дослідженій вибірці їхня операціоналізація частіше локалізується в межах окремих курсів або здійснюється через практику наприкінці навчання, а не розгортається як наскрізна освітня логіка від першого до четвертого курсу. Узагальнені результати аналізу наведено в таблиці 1.

Отже, результати порівняння засвідчують, що найсуттєвіші відмінності між моделями виявляються не у фундаментальному змісті підготовки, а в рівні гнучкості програм, інституціалізації практики, автентичності оцінювання та ступені залучення роботодавців до освітнього процесу.

Отримані результати показують, що для українських програм важливим є не додавання окремих британських елементів, а перегляд самої логіки підготовки: від набору дисциплін до послідовної професійної траєкторії студента. Сучасні дослідження про компетентності програмної інженерії, розрив між університетом та індустрією, а також про співпрацю вищої освіти з роботодавцями послідовно вказують на необхідність більш тісної інтеграції змісту, оцінювання, практики, командної роботи та зовнішнього професійного середовища (Akdur, 2022; Assyne et al., 2022; Vuoriainen et al., 2025; Zarour et al., 2025). Саме в цьому сенсі британський досвід у межах дослідженої вибірки є прогностично цінним для України. На основі здійсненого порівняльно-педагогічного аналізу нижче подано комплекс прогностичних ідей, спрямованих на модернізацію вітчизняної системи підготовки ІТ-фахівців.



Таблиця 1

**Порівняльна характеристика моделей професійної підготовки бакалаврів з програмної інженерії в Україні та Великій Британії**

Категорія порівняння	Україна	Велика Британія
Нормативно-програмна архітектура	Домінує проектування від державного стандарту 121/F2; подібність ключових рамкових параметрів зберігається навіть за різних інституційних профілів	Висока автономія програм у межах QAA/FHEQ/FQHEIS і професійних орієнтирів BCS; назви програм та структурування дисциплін є більш варіативними
Структура і гнучкість	Типовий обсяг 240 ЄКТС і чотирирічна або майже чотирирічна траєкторія; вибірковість є, але модульна реконфігурація обмежена	Три типові траєкторії: ступінь бакалавра з відзнакою, рік виробничої практики в індустрії та інтегрована магістратура
Практика та індустрія	Практика здебільшого існує як окремий блок; роботодавці залучаються нерівномірно, частіше через локальні партнерства, хоча робота у цьому напрямі проводиться все активніше	Стажування, виробничі практики, реальні проекти (live projects), робота на замовлення клієнтів та експертна участь роботодавців інтегруються в дизайн і реалізацію програм
Оцінювання	У публічно доступних описах домінують екзамени, курсові, кваліфікаційні роботи; автентичне оцінювання декларується фрагментарно	Курсові роботи, командні проекти, випускні кваліфікаційні роботи, оцінювані стажування, портфоліо, презентації для замовників, рефлексивні компоненти та інші комплексні інструменти автентичного оцінювання представлені виразніше
Цифрове освітнє середовище	Використання систем управління навчанням, середовищ програмування, репозиторіїв і засобів дистанційного навчання залежить від конкретного ЗВО та освітніх компонентів; професійний інструментарій не завжди експліковано як наскрізну вимогу програми	Системи управління навчанням, репозиторії коду, інструменти командної розробки, тестування, хмарні сервіси, трекінг задач та інші професійні цифрові інструменти частіше інтегруються в проектну і практичну діяльність
Компетентнісний профіль	Сильна фундаментальна підготовка; командні, професійні і працевлаштовувальні компетентності ізольовані в окремих курсах і практиках, а не інтегровані наскрізно	Командна робота, комунікація, здатність до працевлаштування, професіоналізм, супровід, тестування і володіння інструментарієм вбудовуються наскрізно в програму
Підсумкова інтеграція	Завершальна кваліфікаційна робота є обов'язковою, проте вона не завжди виконує функцію інтегративного підсумкового проекту, що акумулює результати навчання за всією програмою	Випускна кваліфікаційна робота (dissertation), індивідуальний дипломний проект або орієнтована на індустрію комплексна підсумкова робота часто є кульмінацією всієї освітньої траєкторії

Джерело: розроблено автором дослідження



**Перехід до багаторівневого проектування змісту підготовки.** Для українських університетів важливо зберегти державний стандарт як національну рамку якості, але доповнити його системою зовнішніх орієнтирів для точнішого налаштування профілю випусника. Практично це означає співвіднесення результатів навчання не лише зі стандартом «121/F2», а й із галузевими орієнтирами за зразком QAA, базовим ядром стандарту CS2023, областями знань SWEBOOK, а також професійно-рольовими картами SFIA. Така багаторівнева модель дає змогу уникнути ситуації, коли програма формально відповідає національним вимогам, але залишається непрозорою для роботодавців і професійної спільноти. Особливо продуктивним цей підхід є для тих українських університетів, які вже впроваджують внутрішню спеціалізацію, як-от окремі освітні програми КПІ ім. Ігоря Сікорського чи профільовані траєкторії інших провідних технічних ЗВО (КПІ ім. Ігоря Сікорського, 2026a; 2026b).

**Запровадження наскрізної проєктної траєкторії.** Цей підхід передбачає наскрізний розвиток практичних навичок замість концентрації найважливішого практичного досвіду лише на фінальному етапі навчання. Логіка спірального навчання в програмній інженерії полягає в послідовному ускладненні проєктних завдань: від індивідуальних невеликих задач на першому курсі до командних проєктів, клієнтських кейсів, інтегрованих системних завдань і комплексного випускного проєкту на останньому курсі. Саме така траєкторія узгоджується і з рекомендаціями стандарту CS2023 щодо цінності тривалих проєктів спільної розробки, і з висновками сучасних досліджень про ефективність проєктно-орієнтованого навчання у підготовці до складної професійної діяльності (ACM/IEEE-CS/AAAI Joint Task Force, 2023; Nguyen, 2025). Для України це означає необхідність переглянути місце курсових проєктів, міждисциплінарних модулів та командної роботи так, щоб вони утворювали не розрізнений набір окремих активностей, а внутрішньо узгоджену лінію професійного розвитку студента.

**Побудова цілісної екосистеми автентичного оцінювання.** Йдеться не про відмову від підсумкових контрольних заходів як таких, а про зміну співвідношення між формами контролю на користь тих, що дають змогу фіксувати реальну професійну продуктивність студента. До них належать: рецензування коду, експертна оцінка архітектури та дизайну ПЗ, підбиття підсумків командного спринту, дні презентацій функціональності розроблених програмних продуктів, технічні звіти, аналітика систем відстеження завдань, рефлексивні звіти із самооцінювання, портфоліо, відгуки замовників тощо. У британських орієнтирах така логіка прямо підтримується: QAA рекомендує створювати можливості для автентичного оцінювання, BCS вимагає виконання випускного кваліфікаційного проєкту із критичною самооцінкою та фіксацією індивідуального внеску, а університети вносять у публічні описи саме практико-діяльні формати роботи (QAA, 2022; BCS, 2022; Cardiff University, 2026; Queen's University Belfast, 2026; Royal Holloway, University of London, 2026). Для української системи вищої освіти це означає необхідність розроблення карт оцінювання освітніх програм, у яких кожен рік навчання матиме чітко визначений набір доказів сформованості професійних компетентностей студента.

**Інституціалізація партнерства з IT-індустрією.** Йдеться не лише про запрошені лекції чи окремі бази практики, а про партнерство як наскрізний програмний механізм. Це передбачає участь роботодавців у періодичному перегляді результатів навчання, формуванні банку проєктних завдань, спільному керівництві



випускними проектами, зовнішній експертизі контрольних завдань, а також у розширенні можливостей для довготривалих стажувань та виробничих практик. Систематичний огляд Vuoriainen et al. (2025) показує, що успішна співпраця між вищою освітою та індустрією базується на шести чинниках: ясність, комунікація, спільність цілей, зобов'язання, безперервність та взаємна довіра. Саме їх доцільно покласти в основу українських моделей співпраці з IT-кластерами, компаніями та продуктивними командами. В українському контексті потенціал для такого переходу вже простежується в локальних партнерських практиках ЛНУ, ВНТУ, ХНУ, ЛНТУ, та інших ЗВО, але він потребує переведення з рівня окремих ініціатив на системний програмний рівень (Lviv IT Cluster, 2026; Львівський національний університет імені Івана Франка, 2026; Вінницький національний технічний університет, 2026; Хмельницький національний університет, 2025; Луцький національний технічний університет, 2025).

**Гнучка профілізація програм.** Британські університети доводять, що програмна інженерія може успішно реалізовуватися і як окремих ступінь (освітня програма), і як спеціалізація в межах ширшої комп'ютерної програми. Для України це відкриває можливість проектувати варіативні треки всередині однієї освітньої програми, наприклад: тестування та якість програмного забезпечення, хмарні технології та DevOps-інженерія, вбудоване та кіберфізичне програмне забезпечення, системи обробки великих обсягів даних, безпечна розробка програмного забезпечення тощо. Важливо, однак, щоб така профілізація не руйнувала фундаментальної основи, а спиралася на неї. Саме тому гнучкість має поєднуватися з прозорим обов'язковим складником (ядром програми), який охоплює алгоритмічні, програмні, системні, інженерно-процесні та професійні складові (QAA, 2022; ACM/IEEE-CS/AAAI Joint Task Force, 2023; IEEE Computer Society, 2024).

**Посилення зовнішньої професійної валідації програм.** Цей напрям передбачає ефективне поєднання академічної автономії з професійним визнанням за прикладом британської моделі взаємодії університетів із BCS. Для України пряме інституційне копіювання цієї структури не є обов'язковим, однак сам принцип зовнішнього професійного підтвердження якості є надзвичайно цінним. У цьому контексті запропоновані прогностичні ідеї узгоджуються з логікою внутрішнього та зовнішнього забезпечення якості освітніх програм, зокрема з необхідністю обґрунтування цілей освітньої програми, залучення стейкхолдерів, прозорості результатів навчання та відповідності змісту підготовки потребам ринку праці, що відображено у вітчизняній практиці акредитації освітніх програм (Національне агентство із забезпечення якості вищої освіти, 2024). Варто визнати, що цей вектор розвитку вже простежується у вітчизняному освітньому просторі. Зокрема, через функціонування рад роботодавців та експертну валідацію навчальних планів, а також через перші спроби інтеграції міжнародних стандартів на кшталт SFIA у профільних IT-програмах (наприклад, в межах спільних проєктів ЛНУ з IT-кластерами) (Lviv IT Cluster, 2026). Проте наразі цей процес має переважно фрагментарний або ініціативний характер на рівні окремих кафедр. Головне завдання полягає в тому, щоб перетворити ці поодинокі успішні кейси на загальносистемну, інституціоналізовану практику.

З практичного погляду це означає, що розділ модернізації української IT-освіти має розгортатися щонайменше на трьох рівнях. На програмному рівні необхідні перепроєктування змісту навчання, трансформація форм контролю та методів оцінювання навчальних досягнень та оновлення проєктної траєкторії. На кафедральному й інституційному рівнях потрібні механізми сталого партнерства з



роботодавцями, цифрова інфраструктура командної розробки, а також підтримка викладачів у переорієнтації курсів на практико-діяльні формати. На галузевому рівні – створення умов, за яких результати навчання на бакалавраті можуть бути прозоро співвіднесені з професійними траєкторіями, компетентнісними профілями та секторальними очікуваннями. Отже, практична цінність британського досвіду полягає не в його прямому перенесенні, а в можливості використати окремі рішення для посилення гнучкості, практичної спрямованості та зв'язку українських програм із професійним середовищем.

#### **ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК**

Проведений порівняльно-педагогічний аналіз дає змогу стверджувати, що в межах дослідженої вибірки професійна підготовка бакалаврів із програмної інженерії в університетах Великої Британії вирізняється модульною гнучкістю, глибокою інтеграцією індустріальних практик, системнішим застосуванням автентичного оцінювання та тіснішим зв'язком між дизайном освітніх програм і професійною соціалізацією майбутніх фахівців. Українські програми натомість демонструють фундаментальність, нормативну впорядкованість і потужне базове ядро, однак потребують посилення практико-інтегрованої логіки навчання, прозорішого зв'язку з міжнародними рамками професійних навичок та гнучкішої побудови освітньої траєкторії.

На цій основі обґрунтовано шість ключових прогностичних ідей для модернізації ІТ-освіти в Україні: багаторівневе проектування змісту підготовки; наскрізна проєктна траєкторія; екосистема автентичного оцінювання; інституціалізоване партнерство з ІТ-індустрією; гнучка профілізація програм; зовнішня професійна валідація результатів навчання. Їхня адаптація є педагогічно доцільною за умови, що вона відбуватиметься з урахуванням державного стандарту спеціальності, інституційних ресурсів конкретного закладу вищої освіти та реальних потреб регіонального й національного ІТ-ринку.

Перспективу подальших досліджень вбачаємо в апробації запропонованих рекомендацій під час модернізації освітніх програм зі спеціальності «F2 Інженерія програмного забезпечення», а також у розробленні інструментарію для системного узгодження результатів навчання з професійними ролями, профілями компетентностей та типовими робочими сценаріями сучасної програмної інженерії.

#### **ЛІТЕРАТУРА**

1. Вінницький національний технічний університет. (2026). *Освітня програма «Інженерія програмного забезпечення» (бакалавр)*. <https://vntu.edu.ua/uk/information-for-enrollee/osvitnya-programa-inzheneriya-programnogo-zabezpechennya-bakalavr-1331.html>
2. КПІ ім. Ігоря Сікорського. (2026a). *F2 Інженерія програмного забезпечення*. <https://osvita.kpi.ua/F2>
3. КПІ ім. Ігоря Сікорського. (2026b). *Інженерія програмного забезпечення інформаційних систем*. [https://osvita.kpi.ua/121\\_OPPB\\_IPZIS](https://osvita.kpi.ua/121_OPPB_IPZIS)
4. Луцький національний технічний університет. (2026). *Освітня програма «Інженерія програмного забезпечення» (бакалавр)*. <https://lntu.edu.ua/uk/op/2025/f2-inzheneriya-programnoho-zabezpechennya>
5. Львівський національний університет імені Івана Франка. (2026). *Інженерія програмного забезпечення*. <https://admission.lnu.edu.ua/specialization/software-engineering>



6. Міністерство освіти і науки України. (2024). *Про внесення змін до деяких стандартів вищої освіти (Наказ № 842 від 13.06.2024)*. <https://mon.gov.ua/static-objects/mon/sites/1/vishcha-osvita/zatverdzeni%20standarty/2024/Nakaz-842.vid.13.06.2024.pdf>
7. Міністерство освіти і науки України. (2018). *Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 – Інформаційні технології, спеціальність 121 – Інженерія програмного забезпечення*. <https://mon.gov.ua/static-objects/mon/sites/1/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzhener.programn./zabezp.bakalavr-1.pdf>
8. Національне агентство із забезпечення якості вищої освіти. (2024). *Критерії оцінювання якості освітньої програми*. <https://naqa.gov.ua>
9. Національний університет «Запорізька політехніка». (2023). *Інженерія програмного забезпечення*. <https://catalogop.zp.edu.ua/EProg.php?Id=93&Mode=1>
10. Національний університет «Львівська політехніка». (2025). *Інженерія програмного забезпечення*. <https://directory-new.lpnu.ua/majors/ikni/6.F2.00.00/8/2025/ua/full>
11. Національний університет «Одеська політехніка». (2026). *Інженерія програмного забезпечення*. <https://op.edu.ua/education/programs/bac-121-0>
12. Тернопільський національний технічний університет імені Івана Пулюя. (2025). *Освітньо-професійна програма «Інженерія програмного забезпечення»*. <https://m.tntu.edu.ua/storage/pages/00000120/opF2b.pdf>
13. Харківський національний університет радіоелектроніки. (2024). *Загальна інформація про спеціальність 121 – Інженерія програмного забезпечення*. <https://software.nure.ua/specialty>
14. Хмельницький національний університет. (2025). *Освітня програма «Інженерія програмного забезпечення»*. <https://khmnu.edu.ua/wp-content/op/b/f2-ipz-2025.pdf>
15. Aberystwyth University. (2026). *Software Engineering (with integrated year in industry)*. <https://courses.aber.ac.uk/undergraduate/software-engineering>
16. Akdur, D. (2022). Analysis of Software Engineering Skills Gap in the Industry. *ACM Transactions on Computing Education*, 23(1), Article 16. <https://doi.org/10.1145/3567837>
17. Assyne, N., Ghanbari, H., & Pulkkinen, M. (2022). The state of research on software engineering competencies: A systematic mapping study. *Journal of Systems and Software*, 185, Article 111183. <https://doi.org/10.1016/j.jss.2021.111183>
18. BCS, The Chartered Institute for IT. (2022). *Academic Accreditation Guidelines*. <https://www.bcs.org/media/1209/accreditation-guidelines.pdf>
19. Cardiff University. (2026). *Applied Software Engineering (BSc)*. <https://www.cardiff.ac.uk/study/undergraduate/courses/2026/applied-software-engineering-bsc>
20. CM/IEEE-CS/AAAI Joint Task Force on Computer Science Curricula. (2023). *Computer Science Curricula 2023 (Version Gamma)*. <https://csed.acm.org/wp-content/uploads/2023/09/Version-Gamma.pdf>
21. Daun, M., Grubb, A. M., & Tenbergen, B. (2023). A systematic literature review of requirements engineering education. *Requirements Engineering*, 28, 145–175. <https://doi.org/10.1007/s00766-022-00381-9>
22. Lancaster University. (2026). *Software Engineering BSc Hons (G602)*. <https://www.lancaster.ac.uk/study/undergraduate/courses/software-engineering-bsc-hons-g602/2027>



23. Lviv IT Cluster. (2026). *IT Expert. Інноваційні. Експертні. Орієнтовані на ринок: IT-програми, які готують лідерів галузі*. <https://itcluster.lviv.ua/projects/it-expert>
24. Neubauer, A. (2024). Trends in Comparative Education in the 21st Century: *Research Production and Areas of Interest*. *European Public & Social Innovation Review*, 9, 1–20. <https://doi.org/10.31637/epsir-2024-288>
25. Nguyen, D. (2025). Project-Based Learning (PBL) as an Experiential Pedagogical Methodology in Interdisciplinary Education: A Review of the Literature. *International Journal of Education in Mathematics, Science and Technology*, 13(4), 1016–1039. <https://doi.org/10.46328/ijemst.4869>
26. Pažur Aničić, K., & Stapić, Z. (2022). Teaching Methods in Software Engineering: A Systematic Review. *IEEE Software*, 39(6), 73–79. <https://doi.org/10.1109/MS.2022.3152629>
27. Quality Assurance Agency for Higher Education. (2022). *Subject Benchmark Statement: Computing*. <https://www.qaa.ac.uk/docs/qaa/sbs/sbs-computing-22.pdf>
28. Quality Assurance Agency for Higher Education. (2024). *The Frameworks for Higher Education Qualifications of UK Degree-Awarding Bodies*. <https://www.qaa.ac.uk/docs/qaa/quality-code/the-frameworks-for-higher-education-qualifications-of-uk-degree-awarding-bodies-2024.pdf>
29. Queen's University Belfast. (2026). *Software Engineering With Placement*. <https://www.qub.ac.uk/courses/undergraduate/software-engineering-placement-beng-g604>
30. Royal Holloway, University of London. (2026). *Computer Science (Software Engineering) (BSc)*. <https://www.royalholloway.ac.uk/studying-here/undergraduate/computer-science/computer-science-software-engineering>
31. SFIA Foundation. (2024a). *SFIA 9*. <https://sfia-online.org/en/sfia-9>
32. SFIA Foundation. (2024b). *Training and education*. <https://sfia-online.org/>
33. The University of Edinburgh. (2026). *Software Engineering BEng (Hons)*. <https://study.ed.ac.uk/programmes/undergraduate/59-software-engineering>
34. University of Glasgow. (2026). *Software Engineering BSc/MSci*. <https://www.gla.ac.uk/undergraduate/degrees/softwareengineering>
35. University of Sheffield. (2026). *Computer Science (Software Engineering) BEng*. <https://sheffield.ac.uk/undergraduate/courses/2027/computer-science-software-engineering-beng>
36. University of Southampton. (2026). *Software Engineering BEng*. <https://www.southampton.ac.uk/courses/software-engineering-degree-beng>
37. University of Strathclyde. (2026). *BSc Hons Software Engineering Degree*. <https://www.strath.ac.uk/courses/undergraduate/softwareengineering>
38. Vuoriainen, A., Rikala, P., Heilala, V., Lehesvuori, S., Oz, S., Kettunen, L., & Hämäläinen, R. (2025). The six C's of successful higher education-industry collaboration in engineering education: A systematic literature review. *European Journal of Engineering Education*, 50, 6–50. <https://doi.org/10.1080/03043797.2024.2432440>
39. Zarour, M., Alenezi, M., & Akour, M. (2025). A Framework to Evaluate Software Engineering Program Using SWEBOK Version 4. *Journal of Communications Software and Systems*, 21(1), 66–78. <https://doi.org/10.24138/jcomss-2024-0088>