

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»
Факультет автоматизації виробництва та цифрових технологій
Кафедра цифрових технологій та проектно-аналітичних рішень

«Допущено до захисту»
Гарант ОПП

Павло САГАЙДА

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістра

за підсумками виконання
освітньо-професійної програми
«Комп'ютерні науки та цифровий інтелект»
за спеціальністю 122 Комп'ютерні науки

**на тему «Дослідження моделей, методів та інформаційних
технологій підтримки формування змісту дисциплін освітньо-
професійних програм у технічному університеті»**

Керівник роботи

Павло САГАЙДА

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають
посилання на відповідне джерело*

Здобувач

Ілля ІЛЛЯШЕНКО

Підсумкова оцінка за атестацію			
--------------------------------	--	--	--

Голова ЕК

Антон КУДРЯВЦЕВ

ЗАПОРІЖЖЯ 2026

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»

Факультет цифрових технологій та автоматизації виробництва

Кафедра цифрових технологій та проектно-аналітичних рішень

Ступень вищої освіти магістр

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

Освітня програма Комп'ютерні науки та цифровий інтелект

(назва)

ЗАТВЕРДЖУЮ

Гарант ОПП

Павло САГАЙДА

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Ільяшенко Іллі Олеговичу

(прізвище, ім'я, по батькові здобувача)

1. Тема роботи Дослідження моделей, методів та інформаційних технологій підтримки формування змісту дисциплін освітньо-професійних програм у технічному університеті

керівник роботи Сагайда Павло Іванович, доцент, докт. техн. наук,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом Університету від 10.09.2026 р. №239

2. Строк подання роботи 20.01.2026 р.

3. Вихідні дані до роботи Навчальна література, державні стандарти, методична література з спеціальних дисциплін та дипломування, науково-дослідницькі роботи з тематики автоматизації обробки й аналізу даних та методів цифрового інтелекту, літературні джерела, технологічні інструкції тощо

4. Зміст пояснювальної записки (перелік питань) виконати аналіз стану питання та сучасних підходів до автоматизації та інформаційної підтримки процесу формування освітньо-професійних програм технічного напрямку, в тому числі спеціальності Computer Science, удосконалити математичні моделі даного бізнес-процесу, методичне забезпечення та відповідні інформаційні технології; розробити архітектуру інформаційної системи та обрати технологічний стек для реалізації засобу підтримки розробки ОПП; реалізувати програмний комплекс для візуалізації та перевірки відповідностей між результатами навчання та компетентностями; провести експериментальні дослідження, виконати оцінку економічної ефективності та терміну окупності розробленого рішення

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) мета і завдання дослідження; математичні моделі, методика дослідження; діаграми проєкту програмної системи в нотації UML (діаграми прецедентів, класів, послідовностей, діяльності, потоків даних); результати експериментальних досліджень.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Фаховий консультант</i>	П.І. Сагайда, доц., докт. техн. наук		
<i>Нормоконтроль</i>	О.С. Касьянюк, ст. викл.		
<i>Економічна частина</i>	І.А. Гетьман, доц., канд. техн. наук		

7. Дата видачі завдання 10.11.2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи
1.	Аналіз стану питання з застосуванням інформаційних технологій підтримки формування змісту освітньо-професійних програм (на прикладі програми спеціальності «комп'ютерні науки»)	10.01.2026 – 15.01.2026
2.	Математичне моделювання процесів проектування структури та змісту освітніх програм	16.01.2026 – 18.01.2026
3.	Розробка проєкту програмно-методичного забезпечення інформаційної підтримки формування змісту освітньо-професійної програми	19.01.2026 – 21.01.2026
4.	Створення програмного комплексу для підтримки розробки змісту дисциплін освітньо-професійної програми за спеціальністю «Комп'ютерні науки»	22.01.2026 – 23.01.2026
5.	Проведення експериментальних досліджень ефективності програмного комплексу, економічне обґрунтування ефективності розробки	24.01.2026
6.	Оформлення текстової, програмної та графічної документації.	25.01.2026
7.	Оформлення записки. Підготовка доповіді та презентації. Отримання відгуку та рецензії.	27.01.2026
8.	Захист проєкту	29.01.2026

Здобувач вищої освіти

_____ І.О. Ільяшенко
(підпис) (ініціали та прізвище)

Керівник роботи, доц.,
докт. техн. наук

_____ П.І. Сагайда
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Ільяшенко І.О. Дослідження моделей, методів та інформаційних технологій підтримки формування змісту дисциплін освітньо-професійних програм у технічному університеті.

Кваліфікаційна робота на здобуття освітнього ступеня магістра за спеціальністю 122 «Комп'ютерні науки». – ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА», м. Запоріжжя, 2026.

Мета дослідження: розробка методів та спеціалізованої інформаційної технології для автоматизації та інформаційної підтримки процесу формування змісту дисциплін освітньо-професійних програм, що дозволить мінімізувати розрив між академічним навчанням та динамічними вимогами ІТ-індустрії.

Об'єкт дослідження: процес формування змісту освітньо-професійних програм та структурно-логічних схем підготовки фахівців у закладах вищої технічної освіти.

Предмет дослідження: моделі та інформаційні технології аналізу відповідності компетентностей і результатів навчання у змісті освітніх програм за спеціальністю «Комп'ютерні науки».

Методи дослідження: методи теорії графів, комбінаторна оптимізація, онтологічний інжиніринг, методи інтелектуального аналізу даних, об'єктно-орієнтоване проєктування та програмування.

У цій роботі проведено аналіз нормативного базису та міжнародних рекомендацій щодо формування змісту освіти, досліджено математичний апарат для моделювання структурно-логічних зв'язків та топологічного сортування навчальних дисциплін, розроблено архітектуру інформаційної системи та обрано технологічний стек для реалізації засобу підтримки розробки ОПП; реалізовано програмний комплекс для візуалізації та перевірки відповідностей між результатами навчання та компетентностями, який дозволяє автоматизувати роботу та забезпечити інформаційну підтримку гарантів програм та методистів, суттєво скорочуючи час на підготовку матриць відповідності; проведено оцінку економічної ефективності та терміну окупності розробленого рішення.

КЛЮЧОВІ СЛОВА: ОСВІТНЬО-ПРОФЕСІЙНІ ПРОГРАМИ, ФОРМУВАННЯ ЇХ ВМІСТУ, АВТОМАТИЗАЦІЯ ОБРОБКИ ДАНИХ, ІНФОРМАЦІЙНА ПІДТРИМКА ДІЯЛЬНОСТІ, REACT.

ABSTRACT

Ilyashenko I.O. Research on models, methods and information technologies to support the formation of the content of disciplines of educational and professional programs at a technical university.

Qualification work for obtaining the degree of Master of Computer Science in the specialty 122 Computer Science. – LLC "TECHNICAL UNIVERSITY "METINVEST POLYTECHNIC" MES of Ukraine, Zaporizhzhia, 2026.

Purpose of the study: development of methods and specialized information technology for automation and information support of the process of forming the content of disciplines of educational and professional programs, which will minimize the gap between academic education and the dynamic requirements of the IT industry.

Object of the study: the process of forming the content of educational and professional programs and structural and logical schemes for training specialists in institutions of higher technical education.

Subject of the study: models and information technologies for analyzing the correspondence of competencies and learning outcomes in the content of educational programs in the specialty "Computer Science".

Research methods: graph theory methods, combinatorial optimization, ontological engineering, methods of intelligent data analysis, object-oriented design and programming.

In this work, an analysis of the regulatory framework and international recommendations for the formation of educational content was conducted, a mathematical apparatus for modeling structural-logical relationships and topological sorting of academic disciplines was investigated, the architecture of the information system was developed and a technological stack was selected for the implementation of a tool to support the development of OPP; a software package was implemented for visualization and verification of correspondence between learning outcomes and competencies, which allows automating the work and providing information support to program guarantors and methodologists, significantly reducing the time for preparing correspondence matrices; an assessment of the economic efficiency and payback period of the developed solution was conducted.

KEYWORDS: EDUCATIONAL PROGRAMS, FORMATION OF THEIR CONTENT, AUTOMATION OF DATA PROCESSING, INFORMATION SUPPORT OF ACTIVITIES, REACT.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ СТАНУ ПИТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПІДТРИМКИ ФОРМУВАННЯ ЗМІСТУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ (НА ПРИКЛАДІ ПРОГРАМИ СПЕЦІАЛЬНОСТІ «КОМП'ЮТЕРНІ НАУКИ»)	10
1.1 Аналіз концептуально-нормативного базису формування змісту освіти зі спеціальності «Комп'ютерні науки»	10
1.1.1 Компетентнісна парадигма Стандарту вищої освіти України	10
1.1.2 Гармонізація вмісту дисциплін з міжнародними рекомендаціями.....	12
1.2. Роль SWEBOOK v4 у формуванні професійного ядра	13
1.3 Аналіз освітньо-професійної програми «Комп'ютерні науки» для підготовки бакалаврів у ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»	17
1.4 Висновки за розділом 1	32
2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ ПРОЄКТУВАННЯ СТРУКТУРИ ТА ЗМІСТУ ОСВІТНІХ ПРОГРАМ	34
2.1. Графові моделі та аналіз передумов обрання та успішного проходження курсів.....	34
2.2. Розв'язання задачі покриття множини (Set Cover Problem) у формуванні вмісту дисциплін	36

2.3. Моделювання процесу формування індивідуальних освітніх траєкторій	37
2.4 Онтологічний інжиніринг у проектуванні освітніх програм	38
2.5 Розвиток підходів до застосування методів інтелектуального аналізу даних для підтримки прийняття рішень при формуванні ОПП	41
2.6 Архітектура інформаційних технологій та роль Generative AI	44
2.7 Висновки за розділом 2	47
3 РОЗРОБКА ПРОЄКТУ ПРОГРАМНО-МЕТОДИЧНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ФОРМУВАННЯ ВМІСТУ ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ.....	49
4 СТВОРЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ПІДТРИМКИ РОЗРОБКИ ВМІСТУ ДИСЦИПЛІН ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ ЗА СПЕЦІАЛЬНІСТЮ «КОМП'ЮТЕРНІ НАУКИ»	64
4.1 Опис функціоналу та розробка логіки роботи програмного комплексу.....	64
4.2 Проектування програмного комплексу з використанням UML-діаграм та фізична реалізація програми	66
4.3 Висновки за розділом 4	70
5 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ЕФЕКТИВНОСТІ РОЗРОБКИ	73
ВИСНОВКИ	76
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	78
ДОДАТОК А	90
ДОДАТОК Б	91
ДОДАТОК В	100

ВСТУП

В умовах глобальної цифрової трансформації суспільства та стрімкого розвитку інформаційних технологій, вища технічна школа постає перед безпрецедентними викликами, пов'язаними з необхідністю забезпечення якості та релевантності підготовки фахівців [1]. Особливої гостроти ця проблема набуває для спеціальності «Комп'ютерні науки», яка за своєю природою є однією з найбільш динамічних галузей знань. Традиційні підходи до формування змісту освітньо-професійних програм (ОПП), що базуються на інтуїтивному досвіді викладачів та періодичному (раз на кілька років) оновленні навчальних планів, виявляються неспроможними встигати за темпами змін у технологічному стеку індустрії [1-4]. Розрив між академічними знаннями, що надаються університетами, та практичними компетентностями, які вимагаються роботодавцями (так званий «skill gap»), стає критичним фактором, що стримує розвиток національної ІТ-галузі та знижує конкурентоспроможність випускників на глобальному ринку праці [5-7].

Актуальність дослідження моделей, методів та інформаційних технологій підтримки формування змісту дисциплін зумовлена необхідністю переходу від статичних, жорстко детермінованих навчальних планів до гнучких, адаптивних освітніх екосистем. Такі екосистеми повинні базуватися на об'єктивних даних (data-driven approach), використовувати формалізовані математичні моделі для верифікації структурно-логічних зв'язків та застосовувати сучасні інструменти штучного інтелекту для моніторингу зовнішнього середовища [7-9]. В українському контексті це завдання ускладнюється необхідністю гармонізації національних стандартів

вищої освіти з міжнародними рекомендаціями (Computing Curricula 2020, SWEBOOK v4) та забезпеченням індивідуалізації навчання в умовах змішаних та дистанційних форм організації освітнього процесу [10-13].

В даній роботі проведено комплексний аналіз теоретико-методологічних засад та прикладних інструментаріїв, які дозволяють автоматизувати та інтелектуалізувати процеси проєктування змісту освіти. Дослідження охоплює широкий спектр питань: від нормативно-правового регулювання та компетентнісної парадигми до математичного моделювання навчальних планів за допомогою теорії графів та множин, онтологічного інжинірингу знань про предметну область, а також застосування методів Text Mining та Generative AI для аналізу ринку праці та генерації навчального контенту [15-19].

Мета дослідження: розробка методів та спеціалізованої інформаційної технології для автоматизації та інформаційної підтримки процесу формування змісту дисциплін освітньо-професійних програм, що дозволить мінімізувати розрив між академічним навчанням та динамічними вимогами ІТ-індустрії.

Об'єкт дослідження: процес формування змісту освітньо-професійних програм та структурно-логічних схем підготовки фахівців у закладах вищої технічної освіти.

Предмет дослідження: моделі та інформаційні технології аналізу відповідності компетентностей і результатів навчання у змісті освітніх програм за спеціальністю «Комп'ютерні науки».

Завдання, які виконувалися в роботі:

- здійснити аналіз нормативного базису та міжнародних рекомендацій щодо формування змісту освіти;
- дослідити математичний апарат для моделювання структурно-

логічних зв'язків та топологічного сортування навчальних дисциплін;

- розробити архітектуру інформаційної системи та обрати технологічний стек для реалізації засобу підтримки розробки ОПП;

- створити програмний комплекс «Аналізатор компетентностей» для візуалізації та перевірки відповідностей між результатами навчання та компетентностями;

- провести оцінку економічної ефективності та терміну окупності розробленого рішення.

Методи дослідження: методи теорії графів – для аналізу пререквізитів та виявлення циклічних залежностей у навчальних планах; комбінаторна оптимізація – для розв'язання задачі покриття компетентностей мінімальним набором дисциплін; онтологічний інжиніринг – для семантичного структурування знань предметної області; методи інтелектуального аналізу даних (Text Mining та NLP) – для обробки текстів вакансій та аналізу Skill Gap; об'єктно-орієнтоване проектування (UML) – для розробки архітектури програмного забезпечення.

Наукова новизна:

- удосконалено модель структурно-логічної схеми ОПП через застосування алгоритмів топологічного сортування, що дозволяє автоматично генерувати несуперечливі семестрові плани;

- розвинуто підходи до застосування методів інтелектуального аналізу даних для підтримки прийняття рішень при формуванні ОПП;

- дістав подальшого розвитку метод відображення компетентностей шляхом впровадження інвертованого індексу в архітектуру SPA-застосунків, що забезпечує миттєвий двосторонній аналіз зв'язків «Результат навчання -> Компетентність».

Практична цінність:

- розроблено та фізично реалізовано програмний продукт, який автоматизує роботу та забезпечує інформаційну підтримку гарантів програм та методистів, суттєво скорочуючи час на підготовку матриць відповідності;
- результати дослідження можуть бути використані технічними університетами для швидкої адаптації змісту освіти до потреб Industry 4.0 та стандартів групи Метінвест.

1 АНАЛІЗ СТАНУ ПИТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПІДТРИМКИ ФОРМУВАННЯ ЗМІСТУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ (НА ПРИКЛАДІ ПРОГРАМИ СПЕЦІАЛЬНОСТІ «КОМП'ЮТЕРНІ НАУКИ»)

1.1 Аналіз концептуально-нормативного базису формування змісту освіти зі спеціальності «Комп'ютерні науки»

Формування змісту освітньо-професійної програми не є довільним процесом, а регламентується складною системою нормативних документів, національних стандартів та міжнародних рекомендацій [1]. Розуміння цієї ієрархії вимог є відправною точкою для розробки будь-яких моделей автоматизації [2].

1.1.1 Компетентнісна парадигма Стандарту вищої освіти України

В основі сучасної вищої освіти України лежить компетентнісний підхід, який докорінно змінив фокус освітнього процесу з «засвоєння знань» на «здатність діяти». Згідно зі Стандартом вищої освіти за спеціальністю 122 «Комп'ютерні науки», затвердженим Міністерством освіти і науки України, зміст освіти декомпонується на систему компетентностей та програмних результатів навчання (ПРН) [20-22].

Структура компетентностей

Стандарт чітко розмежує компетентності на загальні (ЗК) та

спеціальні (фахові, предметні) (СК).

Інтегральна компетентність виступає вершиною цієї ієрархії, визначаючи узагальнену здатність випускника розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук, що характеризуються комплексністю та невизначеністю умов. Це формулювання вимагає від розробників ОПП закладати у зміст дисциплін не лише алгоритмічні навички, а й здатність до системного аналізу та прийняття рішень в умовах неповної інформації.

Загальні компетентності (ЗК) забезпечують універсальність випускника та його здатність до соціальної адаптації. До них відносяться здатність до абстрактного мислення, аналізу та синтезу (ЗК1), здатність застосовувати знання у практичних ситуаціях (ЗК2), навички міжособистісної взаємодії та спілкування іноземною мовою (ЗК5). Для формування змісту дисциплін це означає необхідність включення компонентів, спрямованих на розвиток soft skills (командна робота, презентація проєктів, етика), інтегрованих у фахові курси [20, 21].

Спеціальні компетентності (СК) визначають професійне ядро. Стандарт виділяє такі ключові групи:

- Математичне моделювання та дослідження неперервних і дискретних моделей (СК1, СК4).
- Проєктування та розробка програмного забезпечення з використанням різних парадигм (об'єктно-орієнтованого, функціонального) (СК8).
- Системний аналіз та прийняття рішень (СК6).
- Інтелектуальний аналіз даних та обчислювальний інтелект (СК2, СК11).

– Високопродуктивні обчислення та клієнт-серверні архітектури (СК9).

Матриця відповідності як інструмент проєктування

Ключовим методичним інструментом, що забезпечує цілісність ОПП, є матриця відповідності програмних результатів навчання (ПРН) та компетентностей, а також матриця забезпечення ПРН освітніми компонентами. Математично це можна представити як відображення множини дисциплін D на множину компетентностей C . Задача формування змісту полягає у такому виборі підмножини дисциплін та їх наповнення, щоб кожен елемент множини обов'язкових компетентностей був покритий принаймні одним (а краще декількома для надійності) освітнім компонентом. Відсутність такого покриття є критичним порушенням стандарту.

1.1.2 Гармонізація вмісту дисциплін з міжнародними рекомендаціями

Для забезпечення міжнародної мобільності студентів та відповідності української освіти світовим трендам, зміст дисциплін повинен враховувати рекомендації Computing Curricula 2020 (CC2020), розроблені спільними зусиллями ACM та IEEE [22].

CC2020 знаменує перехід від моделі «Knowledge-based» (орієнтація на знання, характерна для CC2001/2005) до моделі «Competency-based» (орієнтація на компетентності). Це фундаментальна зміна, яка вимагає переосмислення структури навчального матеріалу. У CC2020 компетентність визначається формулою:

$$\text{Competency} = \text{Knowledge}(K) + \text{Skills}(S) + \text{Dispositions}(D),$$

де Knowledge (Знання): фактичний матеріал, теорії, концепти (наприклад, поняття «поліморфізм»);

Skills (Навички): здатність застосовувати знання на практиці (наприклад, написання коду на C++ з використанням поліморфізму);

Dispositions (Диспозиції): ставлення, поведінкові характеристики та професійні якості (наприклад, проактивність, відповідальність за якість коду, прагнення до самонавчання).

Аналіз CC2020 показує, що українські стандарти (зокрема 122 спеціальність) здебільшого відповідають цій моделі на рівні декларацій, проте на рівні реалізації часто ігнорують компонент Dispositions. Формування змісту дисциплін повинно явно включати активності, що розвивають ці якості: рольові ігри, захист проєктів, рев'ю коду (code review), аналіз етичних дилем.

Також CC2020 вводить інструменти візуалізації навчальних планів, що дозволяє порівнювати профілі підготовки («Computer Science» vs «Software Engineering» vs «Information Systems») за ступенем наповнення різними областями знань. Це дає змогу університетам позиціонувати свої унікальні освітні пропозиції.

1.2. Роль SWEBOOK v4 у формуванні професійного ядра

Оскільки значна частина випускників спеціальності «Комп'ютерні науки» працевлаштовується як розробники програмного забезпечення,

зміст фахових дисциплін повинен корелювати з «Керівництвом до зводу знань з інженерії програмного забезпечення» (SWEBOK). Остання версія SWEBOK v4 суттєво розширила перелік областей знань (Knowledge Areas - KAs), що вимагає відповідної актуалізації навчальних планів [11-14].

У табл. 1.1 наведено порівняльний аналіз ключових змін у SWEBOK та їх вплив на навчальний план.

Таблиця 1.1 – Порівняльний аналіз ключових змін у SWEBOK та їх вплив на навчальний план

Область знань (KA) у SWEBOK v4	Зміни порівняно з v3 / SWEBOK 2004	Імпликації для змісту дисциплін спеціальності «Комп'ютерні науки»
Software Architecture	Виділено в окрему KA (раніше частина Design).	Необхідність введення окремої дисципліни «Архітектура ПЗ» або значного розширення відповідного модуля. Фокус на архітектурних патернах, мікросервісах, атрибутах якості.
Software Engineering Operations	Нова KA.	Включення тем DevOps, CI/CD, SRE (Site Reliability Engineering), контейнеризації (Docker, Kubernetes) у курси з розробки та адміністрування.
Software Security	Нова KA.	Безпека повинна викладатися не як окремий факультатив, а як наскрізна тема («Security by Design») у курсах програмування, баз даних та мереж.
Computing Foundations	Явне виділення фундаментальних основ.	Посилення зв'язку між дискретною математикою, алгоритмами та прикладним програмуванням.

SWEBOK v4 також інтегрує методології Agile та DevOps у всі процеси, відмовляючись від їх ізольованого розгляду, що вимагає зміни підходів до викладання менеджменту проєктів [12, 14].

За результатами аналізу предметної області було розроблено діаграму потоків даних для процесу розробки та удосконалення ОПП, наведену на рисунку 1.1. Опис основних потоків даних включає наступне.

1. Нормативний потік – МОН України та міжнародні організації (АСМ/IEEE) надають "жорсткі" та "м'які" обмеження. Ці дані надходять до першого процесу для формування цільового портрета бакалавра комп'ютерних наук.

2. Аналітичний потік – дані від ІТ-індустрії (профілі вакансій, необхідні стеки технологій) впливають як на загальний перелік компетентностей, так і на конкретне наповнення лабораторних робіт у дисциплінах.

3. Логічний потік – після визначення того, що студент має знати (ПРН), дані переходять до процесу планування, де формується послідовність вивчення предметів (пререквізити та постреквізити).

4. Петля зворотного зв'язку – студенти та випускники генерують дані про якість викладання та релевантність знань. Ці потоки даних ініціюють цикл вдосконалення ОПП через процес валідації.

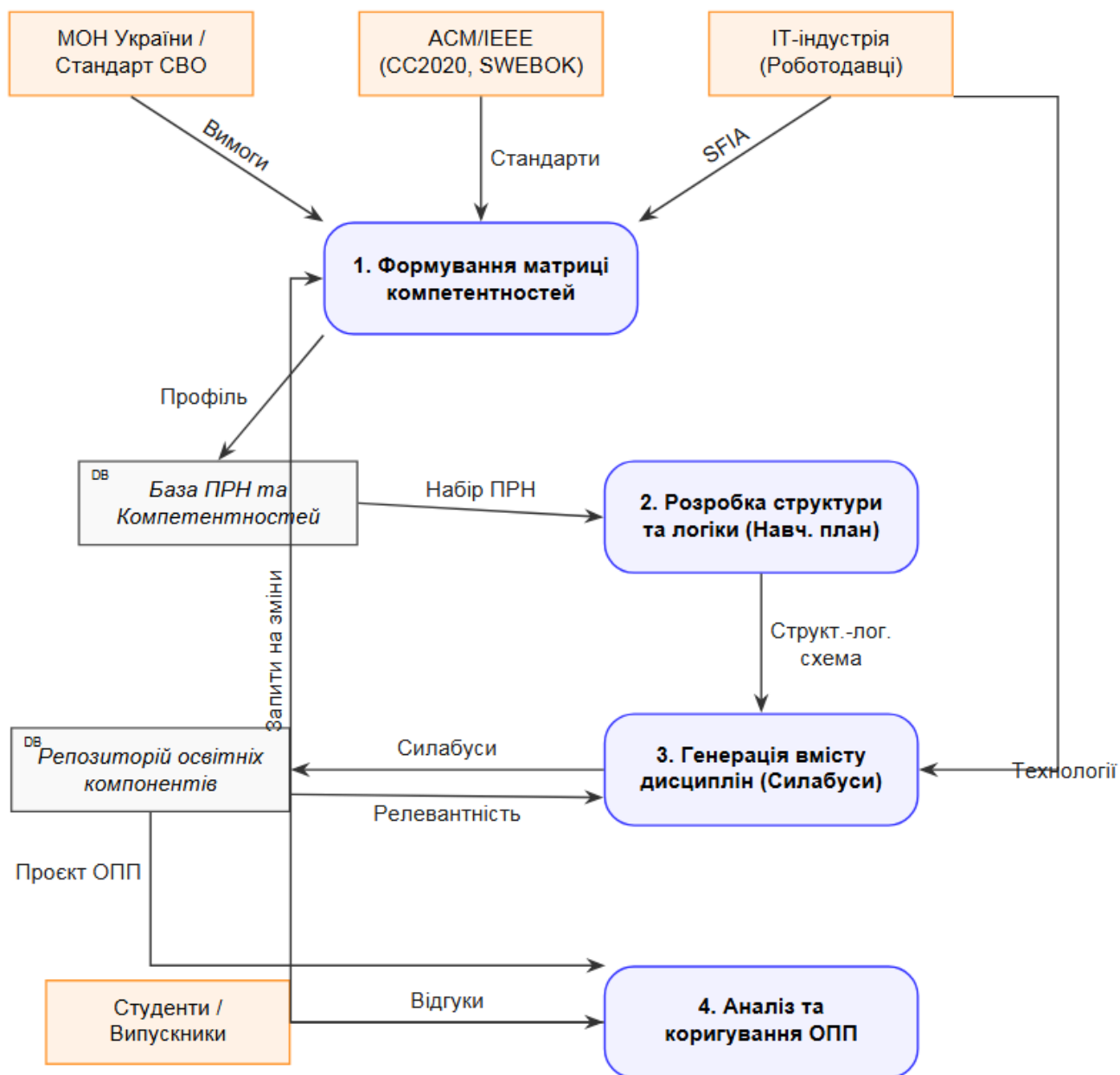


Рисунок 1.1 – Діаграма потоків даних для процесу розробки та удосконалення ОПП

1.3 Аналіз освітньо-професійної програми «Комп'ютерні науки» для підготовки бакалаврів у ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»

Освітньо-професійна програма «Комп'ютерні науки» (<https://metinvest.university/page/8318>) розроблена ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» для підготовки бакалаврів у галузі інформаційних технологій. Програма демонструє системний підхід до формування фахівців, здатних розв'язувати складні спеціалізовані задачі в галузі комп'ютерних наук із застосуванням сучасних теорій та методів інформаційних технологій.

Однією з найбільш помітних особливостей програми є її тісна інтеграція з індустріальним партнером – групою компаній Метінвест. Це партнерство не обмежується формальним співробітництвом, а пронизує всю структуру освітнього процесу. Студенти мають можливість проходити практики безпосередньо на виробничих потужностях активів групи, працювати з реальними даними та завданнями цифрового підприємства, а також отримувати наставництво від фахівців-практиків, які безпосередньо займаються диджиталізацією промислових процесів. Така модель навчання особливо цінна в контексті концепції Industry 4.0, на яку орієнтована програма, оскільки студенти отримують не лише теоретичні знання, а й розуміння того, як ці знання застосовуються для вирішення реальних виробничих завдань [19].

Програма демонструє помітну динаміку розвитку, що підтверджується регулярними переглядами та оновленнями змісту. Початкова редакція була затверджена у 2021 році, після чого відбулися суттєві оновлення у 2023,

2024 та 2025 роках. Важливо відзначити, що процес удосконалення програми не є суто формальним – до робочих груп залучаються не лише викладачі та адміністрація університету, а й представники бізнесу, академічної спільноти з інших закладів вищої освіти та самі здобувачі освіти. Така широка участь стейкхолдерів забезпечує збалансованість програми між академічними стандартами, потребами ринку праці та очікуваннями студентів.

Методологічною основою програми слугують міжнародні рекомендації Computer Science Curricula 2023, розроблені спільно Association for Computing Machinery, IEEE Computer Society та Association for Advancement of Artificial Intelligence [5]. Це означає, що зміст освітніх компонентів відповідає сучасним світовим стандартам у галузі комп'ютерних наук. Водночас програма враховує специфіку українського законодавства та вимоги Стандарту вищої освіти першого бакалаврського рівня для спеціальності 122 Комп'ютерні науки, затвердженого Міністерством освіти і науки України [20].

Структура програми передбачає загальний обсяг 240 кредитів ЄКТС, що відповідає чотирьом рокам навчання при денній формі здобуття освіти. Цей обсяг розподілено між обов'язковими компонентами, які становлять 180 кредитів, та вибірковими дисциплінами обсягом 60 кредитів. Така пропорція є цілком виваженою, оскільки забезпечує формування базових компетентностей через обов'язкові дисципліни, водночас надаючи студентам достатню свободу для побудови індивідуальної освітньої траєкторії. Варто відзначити, що для наборів, починаючи з 2023 року, вибіркові компоненти організовані у формі майнорів, що дозволяє студентам обирати послідовні блоки дисциплін для глибшої спеціалізації у конкретному напрямку.

Математична підготовка в рамках програми займає важливе місце і є достатньо потужною для формування аналітичного мислення та здатності до математичного моделювання. Курс "Математика для комп'ютерних наук та програмування" обсягом одинадцять-дванадцять кредитів охоплює основи математичного аналізу, лінійної алгебри та аналітичної геометрії, що є фундаментом для подальшого вивчення алгоритмів та структур даних. Теорія ймовірностей, математична статистика та випадкові процеси дають студентам інструментарій для роботи з невизначеністю та аналізу даних, що є критично важливим в епоху big data та машинного навчання. Методи дослідження операцій знайомлять студентів з оптимізаційними задачами та прийняттям рішень в складних системах. Для наборів 2022 та 2023 років також передбачено окремий курс дискретної математики, що є логічним доповненням, хоча його відсутність у наборі 2024 року може бути компенсована інтеграцією відповідних тем у основний математичний курс.

Професійне ядро програми побудовано за принципом поступового ускладнення та розширення компетентностей. Студенти починають з основ алгоритмізації та програмування, де закладаються базові навички розробки програмного коду та розуміння алгоритмічної логіки. Курс обсягом десять-одинадцять кредитів є одним з найбільш об'ємних в програмі, що відображає його фундаментальне значення. Надалі студенти вивчають об'єктно-орієнтоване програмування, що є основною парадигмою сучасної розробки програмного забезпечення, і знайомляться з веб-технологіями через курси з веб-дизайну та веб-розробки. Такий підхід забезпечує здатність випускників працювати як з традиційними desktop-додатками, так і з сучасними веб-сервісами.

Особливу увагу в програмі приділено темам, пов'язаним зі штучним інтелектом та аналізом даних. Курс "Системи штучного інтелекту та

інтелектуальний аналіз даних" обсягом шість-сім кредитів охоплює методи машинного навчання, нейромережеві технології, нечітку логіку та інші підходи обчислювального інтелекту. Це відповідає сучасним трендам ринку праці, де попит на фахівців з AI та Data Science стабільно зростає. Додатковим свідченням актуальності програми є включення дисципліни «Використання генеративного штучного інтелекту та low-code автоматизація», що відображає найновітні тенденції в галузі, пов'язані з появою великих мовних моделей типу GPT та розвитком платформ low-code розробки.

Інфраструктурні технології представлені курсами з комп'ютерних мереж, операційних систем, організації баз даних та знань, а також розподілених обчислень та хмарних технологій. Це забезпечує розуміння студентами не лише того, як писати програмний код, а й того, як цей код буде функціонувати в реальних розподілених системах, як організовувати зберігання та обробку великих обсягів даних, як забезпечувати масштабованість та надійність програмних рішень. Курс з основ кібербезпеки та захисту інформації є особливо актуальним в сучасних умовах, коли питання безпеки даних стають критично важливими як для бізнесу, так і для суспільства загалом.

Бізнес-орієнтовані дисципліни, такі як управління бізнесом та бізнес-процесами, стандарти та методології бізнес-аналізу, управління проектами в сфері IT, формують у студентів розуміння того, що технічні рішення завжди вбудовані в бізнес-контекст. Випускники повинні не лише вміти розробляти програмне забезпечення, а й розуміти, яким чином це програмне забезпечення створює цінність для бізнесу, як оцінювати ефективність технічних рішень з економічної точки зору, як організовувати роботу в команді та керувати проектами. Це особливо важливо з огляду на те, що

багато випускників у майбутньому можуть займати керівні позиції або працювати як технічні підприємці.

Мовна підготовка в програмі заслуговує на окрему увагу. Англійська мова для сфери інформаційних технологій вивчається протягом всього періоду навчання з обсягом двадцять два - двадцять чотири кредити, що є досить значним показником. Це відображає реалії сучасної ІТ-індустрії, де англійська мова фактично є основною мовою спілкування, документації, навчальних матеріалів та професійних обговорень. Крім того, програма передбачає вивчення ділової та наукової української мови, що важливо для формування здатності до академічної комунікації та професійної взаємодії в українському контексті.

Компетентнісний підхід, покладений в основу програми, передбачає формування шістнадцяти загальних та сімнадцяти фахових компетентностей. Загальні компетентності охоплюють широкий спектр здатностей, від абстрактного мислення та аналізу до здатності працювати в команді та приймати обґрунтовані рішення. Особливо слід відзначити включення компетентності щодо здатності ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції, що було додано відповідно до наказу Міністерства освіти і науки України у 2024 році. Це свідчить про увагу розробників програми до формування не лише професійних, а й громадянських якостей випускників.

Фахові компетентності детально описують здатності випускників у різних аспектах професійної діяльності. Від здатності до математичного формулювання та дослідження моделей до практичних навичок розробки програмного забезпечення, від роботи з базами даних до забезпечення інформаційної безпеки, від аналізу бізнес-процесів до виконання паралельних обчислень. Комплексність цих компетентностей забезпечує

універсальність підготовки випускників, які можуть працювати в різних сегментах ІТ-індустрії.

Програмні результати навчання, яких налічується дев'ятнадцять, конкретизують компетентності у вимірювані освітні досягнення. Кожен результат чітко описує, що саме має вміти робити випускник після завершення навчання. Варто відзначити, що програмні результати регулярно переглядаються відповідно до змін стандартів освіти. Так, ПР14 було вилучено через дублювання з ПР13, а ПР19 додано для відображення здатності випускників асоціювати себе як членів громадянського суспільства, що є особливо актуальним в сучасних умовах.

Методи викладання та навчання в рамках програми орієнтовані на студентоцентрований підхід з акцентом на проблемно-орієнтоване навчання. Це означає, що студенти не просто отримують готову інформацію, а активно залучаються до вирішення реальних або максимально наближених до реальності задач. Використовуються різноманітні форми освітньої активності, від традиційних лекцій та семінарів до кейс-технологій, лабораторних робіт з використанням спеціалізованого програмного забезпечення, виконання індивідуальних та групових проєктів. Важливою складовою є робота з науковими публікаціями у наукометричних базах Scopus та Web of Science, що формує дослідницькі компетентності студентів та готує їх до можливого продовження навчання на магістерському рівні [23-24].

Система оцінювання передбачає комбінацію поточного контролю та підсумкового оцінювання. Поточна робота оцінюється через тестування, аналіз активності студентів в інтерактивних форматах, розв'язання аналітичних та дослідницьких завдань, підготовку звітів. Цікавим елементом є самооцінювання академічного прогресу, коли студенти самі визначають

ступінь сформованості своїх компетентностей, що розвиває рефлексивні здібності та здатність до самоаналізу. Підсумковий контроль може включати тестування, есе, розв'язання задач, захист курсових робіт та звітів з практики. Використовується стобальна шкала з переведенням у національну чотирибальну систему та систему ECTS, що забезпечує прозорість та можливість порівняння результатів навчання. Також визначено необхідність та правила використання засобів генеративного штучного інтелекту в навчальному процесі [25-27].

Ресурсне забезпечення програми виглядає достатнім для якісного освітнього процесу. Університет має необхідну матеріально-технічну базу, включаючи комп'ютерні класи з сучасним обладнанням, лабораторії, бібліотеку з доступом до міжнародних наукових ресурсів через Research4Life та електронну бібліотеку Kortext. Особливо цінним є доступ до полігонів і лабораторій на потужностях активів групи Метінвест, що дозволяє студентам працювати з промисловим обладнанням та реальними виробничими даними. Програмне забезпечення включає ліцензійні пакети Microsoft, MathWorks, AutoCAD, а також програми з відкритою ліцензією, що охоплює потреби більшості освітніх компонентів. Кожен студент отримує корпоративний обліковий запис Microsoft з доступом до ліцензійного програмного забезпечення та центру командної роботи MS Teams, що особливо важливо в умовах змішаного навчання.

Практична підготовка організована через систему практик, що включає практику з ознайомлення та профілізації професійної діяльності та переддипломну практику. Загальний обсяг практик становить дев'ять кредитів, що є достатнім показником. Проте слід відзначити, що практики зосереджені переважно на старших курсах навчання, тоді як раннє знайомство студентів з реальним робочим середовищем могло б посилити

практичну орієнтацію програми. З іншого боку, програма передбачає можливість для студентів починаючи з третього курсу поглиблено працювати над виконанням кваліфікаційної роботи з постійним зворотним зв'язком від академічного керівника та наставника від бізнесу, що частково компенсує пізній початок формальних практик.

Кваліфікаційна робота бакалавра є підсумковим елементом програми і передбачає теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання в галузі комп'ютерних наук. Важливо, що робота може виконуватися на матеріалах одного з активів групи Метінвест, що забезпечує її практичну цінність та релевантність реальним бізнес-потребам. Програма приділяє значну увагу академічній доброчесності, чітко забороняючи плагіат, фальсифікацію та списування. Кваліфікаційні роботи оприлюднюються в репозиторії університету, що сприяє відкритості та можливості ознайомлення з результатами досліджень студентів.

Особливості програми включають низку унікальних елементів, що виділяють її серед інших програм підготовки з комп'ютерних наук. Стипендіальна програма надає фінансову підтримку здобувачам освіти. Професійна психологічна підтримка доступна студентам як членам спільноти групи Метінвест, що особливо важливо в умовах військового стану. Передбачено персональний супровід ветеранів, що демонструє соціальну відповідальність закладу. Для громадян України чоловічої статі обов'язковими є компоненти базової загальновійськової підготовки, хоча передбачені винятки для осіб, звільнених від такої підготовки за різними підставами, які натомість вивчають курс «Сучасна воєнно-політична історія України та світу».

Додатковою можливістю є позакредитні освітні компоненти, зокрема «Фізичне виховання та особисте здоров'я» та літня школа «Штучний інтелект та програмування для інформаційної підтримки сучасного виробництва». Остання проводиться запрошеними фахівцями-практиками ІТ-фірм у форматі лекцій-тренінгів за участі НТУ «Дніпровська політехніка» в рамках меморандуму про співробітництво, що розширює можливості студентів для неформального навчання та networking з професійною спільнотою.

Аналізуючи зміст програми з точки зору відповідності сучасним трендам індустрії, можна констатувати, що програма охоплює більшість актуальних напрямків розвитку інформаційних технологій. Штучний інтелект та машинне навчання представлені через спеціалізовані курси та інтегровані в інші дисципліни. Хмарні технології та розподілені обчислення є окремими освітніми компонентами. Кібербезпека виділена в самостійний курс. Генеративний штучний інтелект, що став особливо актуальним після появи ChatGPT та подібних систем, включено до програми як окремий компонент, що свідчить про оперативну реакцію на нові виклики та можливості.

Водночас можна виділити деякі теми, які могли б бути представлені більш детально або додані до програми. Технології blockchain та Web3, незважаючи на свою популярність та зростаючу актуальність у фінансовому секторі та системах управління ланцюгами постачання, не виділені в окремі компоненти програми. Інтернет речей та Edge Computing, які активно розвиваються в контексті Industry 4.0, згадуються опосередковано, але не мають систематичного висвітлення. Контейнеризація та оркестрація (Docker, Kubernetes), які є стандартом де-факто в сучасній розробці та deployment додатків, також не виділені окремо, хоча можуть бути частиною

курсів з розподілених обчислень чи хмарних технологій.

MLOps – практики організації життєвого циклу моделей машинного навчання – є критично важливою компетентністю для фахівців, що працюють з AI, але в програмі представлена лише частково через загальні курси з управління проєктами та програмного забезпечення. Computer Vision та Natural Language Processing, хоча й можуть бути частиною курсу зі штучного інтелекту, могли б бути виділені в окремі поглиблені курси, враховуючи їхню важливість у сучасних застосуваннях AI. Quantum Computing, звичайно, є досить специфічною темою і її відсутність на бакалаврському рівні цілком виправдана, хоча ознайомлення з базовими концепціями квантових обчислень могло б бути корисним для студентів, які планують академічну кар'єру.

Матриці відповідності програмних результатів навчання компонентам освітньої програми та матриці відповідності компетентностей компонентам є детальними та добре опрацьованими. Вони демонструють логічні зв'язки між тим, що вивчають студенти, та тим, які компетентності та результати навчання вони мають досягти. Кожен освітній компонент вносить свій внесок у формування декількох компетентностей та програмних результатів, що забезпечує комплексність підготовки. Водночас деякі компоненти, особливо кваліфікаційна робота та переддипломна практика, «покривають» надто багато компетентностей одночасно, що є природним для інтегровальних компонентів, але вимагає ретельної організації для того, щоб усі заявлені результати дійсно досягалися.

Структурно-логічні схеми послідовності вивчення компонентів демонструють продуману прогресію від базових до спеціалізованих дисциплін. Перший курс зосереджений на мовній підготовці, світоглядних дисциплінах та фундаментальних STEM-компонентах, що закладають

основу для подальшого навчання. Другий та третій курси присвячені дисциплінам професійного ядра, де студенти опановують основні технології та методології комп'ютерних наук. Четвертий курс включає більше спеціалізованих дисциплін, практики та підготовку кваліфікаційної роботи, що логічно завершує освітній процес.

Важливим питанням є перспективи працевлаштування випускників програми. Програма готує фахівців для широкого спектру професійних ролей відповідно до Національного класифікатора професій, включаючи аналітиків даних, інженерів з даних, аналітиків та інженерів програмного забезпечення, програмістів, розробників архітектури та розробників програмного забезпечення. Цей перелік охоплює більшість базових позицій у сучасній IT-індустрії. Водночас з урахуванням сучасних трендів список міг би бути розширений такими ролями як Machine Learning Engineer, DevOps Engineer, Cloud Architect, Data Scientist, що також відповідають компетентностям, які формуються в рамках програми.

Конкурентоспроможність випускників на ринку праці визначається кількома факторами. По-перше, міцна теоретична база в математиці та фундаментальних комп'ютерних науках дає випускникам здатність до навчання новим технологіям, що постійно з'являються в індустрії. По-друге, практичні навички розробки програмного забезпечення, роботи з базами даних, хмарними сервісами та системами штучного інтелекту відповідають поточним потребам ринку. По-третє, досвід роботи над реальними проектами для активів групи Метінвест дає випускникам портфоліо реальних кейсів, що є суттєвою перевагою при працевлаштуванні. По-четверте, знання англійської мови професійного спрямування відкриває можливості для роботи з міжнародними компаніями та проектами. По-п'яте, розвинені soft skills, зокрема здатність працювати в команді, спілкуватися,

приймати рішення, робить випускників готовими до корпоративного середовища.

Академічна мобільність передбачена програмою як можливість, хоча конкретні угоди та програми обміну не деталізовані в тексті документа. Університет декларує готовність визнавати кредити, отримані в рамках національної та міжнародної мобільності за дво- і багатосторонніми угодами. Для посилення міжнародного виміру програми було б корисно розвивати конкретні партнерства з зарубіжними університетами, програми подвійних дипломів, можливості семестрового навчання за кордоном. Це особливо актуально для ІТ-галузі, яка є глобальною за своєю природою.

Загалом освітньо-професійна програма «Комп'ютерні науки» ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» демонструє високий рівень якості та відповідності сучасним вимогам як національних стандартів освіти, так і міжнародних рекомендацій у галузі комп'ютерних наук. Унікальна інтеграція з промисловим партнером створює значну додану вартість для студентів через доступ до реальних виробничих кейсів та сучасного обладнання. Регулярне оновлення змісту програми з урахуванням відгуків стейкхолдерів забезпечує її динамічну адаптацію до змін у технологічному ландшафті та на ринку праці. Комплексний підхід до формування компетентностей, що поєднує фундаментальну теоретичну підготовку, практичні навички розробки, розуміння бізнес-контексту та розвиток soft skills, створює передумови для конкурентоспроможності випускників як на національному, так і на міжнародному ринках праці.

Водночас програма має потенціал для подальшого вдосконалення через розширення тем, пов'язаних з новітніми технологіями, такими як blockchain, advanced computer vision, natural language processing, MLOps, збільшення кількості практик на ранніх курсах, розвиток міжнародного

виміру через партнерства з зарубіжними закладами, інтеграцію можливостей отримання професійних сертифікацій як частини освітнього процесу. Такі кроки могли б посилити і без того міцні позиції програми та зробити її ще більш привабливою для абітурієнтів та роботодавців.

Висновки та рекомендації щодо вдосконалення, за результатами проведеного аналізу по джерелах [28-32]

Короткострокові (до 1 року)

Оновити зміст дисциплін:

– Додати модуль з Docker/Kubernetes до «Розподілених обчислень».

– Розширити тему LLM у «Генеративному ШІ».

– Включити MLOps до курсу з ШІ.

Посилити практичну складову:

– Хакатони та змагання з програмування.

– Більше реальних проєктів з Метінвест.

– Open source contribution як частина навчання.

Додати сучасні інструменти:

– Git/GitHub (якщо не входить до курсів).

– CI/CD практики.

– Agile/Scrum методології (детальніше).

Середньострокові (1-2 роки)

Розширити вибірккові компоненти:

– Blockchain та криптовалюти.

– IoT та Edge AI.

– Ethical AI and Bias.

– Computer Vision (детальніше).

- NLP та Text Mining (детальніше).

Міжнародна співпраця:

- Подвійні дипломи.
- Обмінні програми.
- Спільні проєкти з закордонними ЗВО.

Сертифікації:

- Інтеграція з AWS/Azure/Google Cloud сертифікаціями.
- CompTIA, CISSP для безпеки.
- Coursera/edX спеціалізації.

Довгострокові (2-3 роки)

Дослідницька компонента:

- Більше наукових публікацій студентів.
- Участь у конференціях.
- Research projects як вибіркова дисципліна.

Індустріальні партнерства:

- Розширити коло партнерів (ІТ-компанії м. Дніпра).
- Стажування у FAANG-подібних компаніях.
- Менторські програми.

Спеціалізовані треки:

- AI/ML Track.
- Cybersecurity Track.
- Full-Stack Development Track.
- Data Science Track.

Оцінка якості матриць відповідності освітніх компонентів, результатів навчання та компетентностей:

- Детальний зв'язок компонентів з програмними результатами

навчання та компетентностями.

- Чітке покриття всіх результатів навчання.
- Логічна прогресія від простого до складного.

Зауваження:

– Деякі компоненти перевантажені компетентностями (кваліфікаційна робота).

– Можлива надмірна кореляція (потребує перевірки реального досягнення).

В цілому, програма відповідає сучасним вимогам ринку праці в Україні та світі, особливо у контексті:

- Цифровізації промисловості.
- Розвитку штучного інтелекту.
- Хмарних технологій.
- Кібербезпеки.

Конкурентоспроможність випускників достатньо висока, оскільки випускники отримують:

- Міцну теоретичну базу.
- Практичні навички розробки.
- Досвід роботи з реальними проєктами.
- Soft skills та командну роботу.
- Більш глибоке знання англійської мови.

1.4 Висновки за розділом 1

У першому розділі проведено комплексний аналіз нормативно-концептуального базису, міжнародних стандартів та практичного стану формування змісту освітньо-професійної програми зі спеціальності 122 «Комп'ютерні науки». Результати дослідження дозволяють зробити такі висновки.

1. Визначено пріоритет компетентнісного підходу, який згідно зі Стандартом вищої освіти України зміщує акцент з накопичення теоретичних знань на здатність випускника розв'язувати комплексні задачі. Встановлено, що архітектура ОПП базується на декомпозиції змісту на інтегральну, загальні та спеціальні компетентності, що вимагає використання математичних методів (матриць відповідності) для забезпечення цілісності програми та повного покриття програмних результатів навчання (ПРН).

2. Обґрунтовано необхідність гармонізації змісту освіти з міжнародними рекомендаціями, зокрема Computing Curricula 2020 (CC2020). З'ясовано, що сучасна світова парадигма передбачає інтеграцію трьох компонентів: знань (*Knowledge*), навичок (*Skills*) та професійних якостей (*Dispositions*). Аналіз показав, що для вітчизняних програм критично важливим є посилення саме компоненту *Dispositions* через впровадження інтерактивних методів навчання (code review, командні проєкти).

3. Виявлено ключові вектори актуалізації професійного ядра на основі аналізу стандарту SWEBOOK v4. Встановлено, що сучасна підготовка фахівця з комп'ютерних наук має включати наскрізне вивчення питань кібербезпеки (*Security by Design*), архітектури ПЗ (мікросервіси) та операційних процесів (*DevOps*), що раніше розглядалися як допоміжні або

вибіркові модулі.

4. Проведено критичний аналіз ОПП «Комп'ютерні науки» ТУ «Метінвест Політехніка», який засвідчив її високу адаптивність та орієнтацію на індустрію 4.0. Унікальною перевагою програми є її тісна інтеграція з виробничим сектором, що дозволяє поєднувати фундаментальну математичну підготовку з вирішенням реальних прикладних задач цифрового підприємства.

5. Ідентифіковано перспективні напрями вдосконалення змісту ІТ-освіти, серед яких поглиблення вивчення технологій контейнеризації (Docker, Kubernetes) та практик MLOps, інтеграція інструментів генеративного штучного інтелекту та низькокодової розробки (low-code).

Встановлено, що застосування інформаційних технологій для підтримки формування змісту ОПП дозволяє не лише забезпечити формальну відповідність стандартам, а й динамічно реагувати на запити ринку праці, забезпечуючи високу конкурентоспроможність випускників за рахунок збалансованості Hard та Soft Skills.

2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ ПРОЄКТУВАННЯ СТРУКТУРИ ТА ЗМІСТУ ОСВІТНІХ ПРОГРАМ

Перехід від експертного до формалізованого проєктування змісту освіти вимагає застосування суворого математичного апарату. Освітня програма – це не просто набір дисциплін, а складна система з часовими, логічними та ресурсними обмеженнями [33-35]. Для її моделювання найефективніше використовувати методи теорії графів та комбінаторної оптимізації.

2.1. Графові моделі та аналіз передумов обрання та успішного проходження курсів

Структурно-логічна схема підготовки фахівця може бути представлена як орієнтований граф $G = (V, E)$, де $V = \{v_1, v_2, \dots, v_n\}$ – множина навчальних дисциплін, а E – множина дуг, що відображають залежності передування (пререквізити). Дуга $(v_i, v_j) \in E$ існує тоді і тільки тоді, коли успішне засвоєння дисципліни v_i є необхідною умовою для вивчення v_j [35-37].

Топологічне сортування та виявлення циклів [38-42]

Фундаментальною вимогою до коректності навчального плану є відсутність циклічних залежностей (Directed Acyclic Graph - DAG). Наявність циклу виду $A \rightarrow B \rightarrow C \rightarrow A$ робить план нереалізовним. Для автоматичної

перевірки коректності використовується алгоритм пошуку в глибину (DFS) з відстеженням станів вершин («біла» – не відвідана, «сіра» – в процесі обробки, «чорна» – оброблена). Якщо під час обходу зустрічається ребро, що веде до «сірої» вершини, це свідчить про наявність циклу.

Задача розподілу дисциплін за семестрами зводиться до задачі топологічного сортування графа G . Топологічне впорядкування – це лінійна послідовність вершин, така що для кожного ребра (u, v) вершина u передує v у послідовності.

Алгоритмічна реалізація (Алгоритм Кана) для автоматизованого формування семестрового плану виглядає наступним чином:

1. Обчислити вхідний степінь $deg_{in}(v)$ для кожної вершини (кількість пререквізитів).
2. Ініціалізувати чергу Q усіма вершинами з $deg_{in}(v) = 0$. Це кандидати на перший семестр.
3. Поки Q не порожня:
 - Вилучити вершину u . Додати її до поточного семестру (з урахуванням обмеження на кредити).
 - Для всіх сусідів v вершини u : зменшити $deg_{in}(v)$ на 1.
 - Якщо $deg_{in}(v)$ стає 0, додати v до Q (дисципліна стає доступною для вивчення у наступних семестрах).

Цей підхід дозволяє автоматично генерувати варіанти навчальних планів та перевіряти їх на логічну узгодженість.

Зважені графи та оптимізація шляхів [43-45]

У більш складних моделях графу можуть приписуватися ваги: $w(v_i)$ – кредитний обсяг дисципліни або складність, $w(e_{ij})$ – сила зв'язку між дисциплінами. Використання алгоритмів пошуку найкоротшого шляху

(Дейкстри, Беллмана-Форда) або пошуку критичного шляху (CPM) дозволяє визначити мінімальний час, необхідний для отримання певної кваліфікації, та виявити «вузькі місця» у програмі, які блокують подальше просування студента.

2.2. Розв'язання задачі покриття множини (Set Cover Problem) у формуванні вмісту дисциплін

Однією з ключових задач при формуванні змісту є вибір мінімального набору дисциплін, який забезпечує формування всіх обов'язкових компетентностей. Ця задача формалізується як задача про покриття множини (Set Cover Problem) [46-48].

Нехай $U = \{c_1, c_2, \dots, c_k\}$ – універсум необхідних компетентностей (згідно зі стандартом). Нехай $S = \{D_1, D_2, \dots, D_m\}$ – множина доступних дисциплін, де кожна $D_j \subseteq U$ – це набір компетентностей, які формує дисципліна j [49-51]. Кожній дисципліні приписана вартість w_j (наприклад, кількість кредитів ECTS або годин викладача).

Задача полягає у виборі підмножини індексів $I \subseteq \{1, \dots, m\}$, що мінімізує сумарну вартість $\sum_{j \in I} w_j$ за умови:

$$\bigcup_{j \in I} D_j = U.$$

Це класична NP-складна задача. Для реальних навчальних планів, де кількість дисциплін та компетентностей може бути значною, знаходження

точного розв'язку методами повного перебору є неможливим. Тому використовуються:

1. Жадібні алгоритми: на кожному кроці обирається дисципліна, яка покриває найбільшу кількість ще не покритих компетентностей з найменшою питомою вартістю. Це дає наближений розв'язок з логарифмічною похибкою $O(\ln k)$.

2. Генетичні алгоритми: дозволяють знаходити субоптимальні розв'язки шляхом еволюційного відбору популяції варіантів навчальних планів. Дослідження демонструють ефективність канонічних генетичних алгоритмів для формування навчальних планів, де хромосома кодує набір дисциплін, а фітнес-функція оцінює повноту покриття компетентностей та дотримання обмежень на навантаження.

3. Лінійна релаксація (LP Relaxation): зведення задачі цілочисельного програмування до лінійного програмування шляхом дозволу змінним приймати дробові значення, з подальшим округленням.

2.3. Моделювання процесу формування індивідуальних освітніх траєкторій

Сучасна парадигма студентоцентризму передбачає можливість вибору студентом дисциплін (не менше 25% обсягу програми). Математично формування індивідуальних освітніх траєкторій (ІОТ) розглядається як задача пошуку оптимального шляху в графі освітнього простору з урахуванням індивідуальних переваг.

Дидактичний алгоритм формування ІОТ включає такі етапи:

1. Діагностика: оцінка вхідного рівня знань та професійних інтересів студента (наприклад, через тестування або аналіз попередніх досягнень).
2. Цілепокладання: визначення цільового профілю компетентностей (наприклад, спеціалізація Data Science).
3. Генерація варіантів: використання рекомендаційних систем (Collaborative Filtering або Content-based Filtering) для пропозиції дисциплін, що найкраще відповідають цілям.
4. Валідація: перевірка згенерованої траєкторії на несуперечливість (пререквізити) та здійсненність (розклад, навантаження).

Дослідження показують, що аналіз даних про вибір студентів за допомогою методів кластеризації дозволяє виявити типові патерни траєкторій та адаптувати пропозицію вибіркового дисциплін [52].

2.4 Онтологічний інжиніринг у проектуванні освітніх програм

Для ефективної автоматизації процесів формування змісту освіти недостатньо лише числових даних; необхідне глибоке семантичне розуміння предметної області. Онтологічний підхід дозволяє структурувати знання про навчальний процес у формі, зрозумілій як людині, так і комп'ютеру, забезпечуючи семантичну інтероперабельність різних інформаційних систем [53-55].

Методологія побудови освітніх онтологій [56-58]

Онтологія освітньої програми формально описується як кортеж $O = \langle C, R, F, A \rangle$, де C – концепти (класи), R – відношення, F – функції, A – аксіоми.

Приклад формалізації основних концепцій в процесі онтологічного моделювання наведено на рисунку 2.1.

Онтологічна структура

Кортеж $O = \langle C, R, F, A \rangle$

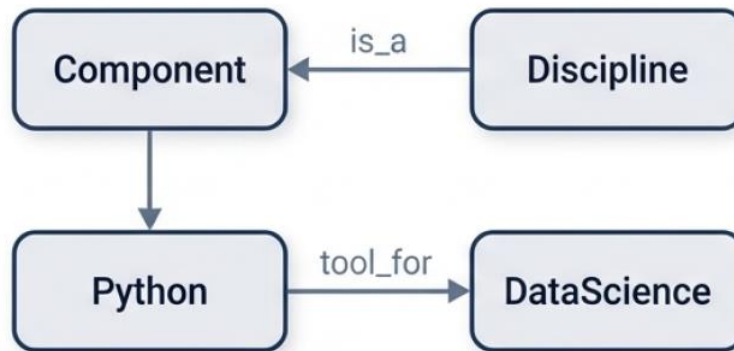


Рисунок 2.1 – Приклад формалізації основних концепцій в процесі онтологічного моделювання

Процес проектування онтології зазвичай включає 7 етапів [59-61]:

1. Визначення області та масштабу: чітке окреслення меж (наприклад, онтологія циклу професійної підготовки спеціальності 122).

2. Аналіз існуючих онтологій (Reuse): перевірка наявності готових онтологічних моделей (наприклад, онтологій SWEBOK або класифікаторів навичок ESCO) для повторного використання.

3. Виділення ключових термінів: автоматизований видобуток термінів з текстів стандартів, робочих програм, підручників.

4. Побудова ієрархії класів (Taxonomy): визначення відношень «клас-підклас» (is-a). Наприклад: ObjectOrientedProgramming is-a

ProgrammingParadigm.

5. Визначення властивостей (Properties):

– Object Properties: зв'язки між екземплярами (наприклад, hasPrerequisite, developsCompetence).

– Data Properties: атрибути (наприклад, numberOfCredits, courseCode).

6. Визначення фасетів та обмежень: кардинальність зв'язків, типи даних.

7. Створення екземплярів (Individuals): наповнення онтології конкретними даними про дисципліни університету.

Пропонується використовувати трансдисциплінарні онтології для інтеграції наукових досліджень та навчального процесу, зокрема через структурування наукових робіт за форматом IMRAD та їх відображення у навчальний контент. Це дозволяє автоматично оновлювати зміст лекцій результатами новітніх досліджень.

Інтеграція та узгодження онтологій (Ontology Alignment) [62-64]

У великому технічному університеті різні кафедри можуть використовувати різну термінологію для опису схожих понять. При формуванні міждисциплінарних програм виникає задача узгодження (alignment) або злиття (merging) онтологій.

Алгоритми узгодження вирішують конфлікти:

– Конфлікти імен: синонімія (різні назви одного поняття) та омонімія (одна назва для різних понять).

– Структурні конфлікти: різний рівень деталізації ієрархії.

Для вирішення цих проблем застосовуються методи обчислення семантичної подібності (semantic similarity measures), що базуються на лінгвістичному аналізі імен концептів (відстань Левенштейна, WordNet) та

аналізі структури графа онтології. Алгоритм інтеграції часто включає додавання фіктивних рівнів ієрархії для вирівнювання структур та використання правил (rules) для розв'язання конфліктів.

Практичне застосування онтологічних моделей для інтелектуальної підтримки навчання

Онтологічні моделі є фундаментом для створення інтелектуальних систем підтримки навчання.

- Семантичний пошук: студент може шукати не просто за ключовими словами, а за змістом («знайти всі курси, що навчають методам кластеризації», навіть якщо в назві немає слова «кластеризація»).

- Персоналізовані рекомендації: система може рекомендувати матеріали, які заповнюють прогалини у знаннях студента, виявлені під час тестування, базуючись на онтологічних зв'язках між темами.

- Автоматична генерація тестів: використовуючи онтологію предметної області, система може генерувати тестові завдання, перевіряючи різні аспекти понять та їх взаємозв'язків.

2.5 Розвиток підходів до застосування методів інтелектуального аналізу даних для підтримки прийняття рішень при формуванні ОПП

В епоху Big Data формування змісту освіти повинно спиратися на аналіз великих масивів даних. Два основні джерела таких даних – це зовнішній ринок праці та внутрішнє освітнє середовище.

Text Mining та NLP для аналізу вимог ринку праці [65-69]

Традиційні методи аналізу ринку праці (опитування роботодавців, круглі столи) є повільними та суб'єктивними. Сучасні підходи базуються на автоматизованому аналізі текстів вакансій (Job Mining) з використанням методів обробки природної мови (NLP).

Типовий пайплайн дослідження включає:

1. Data Collection (Скрепінг): збір вакансій з платформ (LinkedIn, Djinni, Work.ua) за ключовими словами («Junior Java Developer», «Data Scientist»).

2. Preprocessing: очищення тексту, токенізація, видалення стоп-слів, стемінг/лематизація.

3. Named Entity Recognition (NER): виділення специфічних сутностей – навичок (skills), інструментів (tools), технологій. Для цього часто використовуються попередньо навчені моделі (BERT, spaCy), донавчені на корпусах ІТ-лексики.

4. Skill Extraction & Clustering: виявлення найбільш затребуваних навичок та їх групування. Наприклад, методи кластеризації (k-means) дозволяють виявити, що навичка «Git» часто зустрічається разом з «Jira» та «Scrum», формуючи кластер «Командна розробка».

5. Ranking: побудова рейтингів популярності технологій. Якщо аналіз показує стрімке зростання попиту на фреймворк «FastAPI» і падіння «Django», це є сигналом для оновлення відповідної дисципліни.

Дослідження пропонують використовувати мультимодальні векторні представлення (embeddings) кандидатів та вакансій для знаходження неявних відповідностей та покращення рекомендацій.

Аналіз та порівняння профілю навичок, видобутого з вакансій, з профілем компетентностей [70-72]

Порівняння профілю навичок, видобутого з вакансій, з профілем компетентностей ОПП дозволяє виконати Skill Gap Analysis.

Інструменти, такі як Lightcast (раніше Burning Glass), дозволяють автоматизувати цей процес, надаючи аналітику про те, наскільки навчальний план відповідає потребам регіонального або глобального ринку.

Методологія включає:

- Побудову вектору навичок вакансії V_{job} .
- Побудову вектору навичок силабусу $V_{syllabus}$.
- Обчислення косинусної подібності:

$$Sim(V_{job}, V_{syllabus}) = \frac{V_{job} \cdot V_{syllabus}}{\|V_{job}\| \cdot \|V_{syllabus}\|}$$

Низьке значення подібності свідчить про необхідність перегляду змісту дисципліни.

Educational Data Mining (EDM) та аналіз поведінки студентів

Внутрішні дані університету (з систем LMS Moodle, деканатських баз) є цінним джерелом для оптимізації навчання. Методи EDM дозволяють аналізувати:

- Прогнозування успішності за рахунок побудови моделей (Random Forest, Neural Networks), що передбачають оцінку студента або ризик відрахування на основі його активності в семестрі.
- Аналіз вибору дисциплін шляхом виявлення факторів, що впливають на вибір студентами вибіркового курсу (інтерес до викладача, легкість отримання оцінки, корисність для кар'єри). Дослідження показало,

що інформування студентів про зв'язок дисципліни з кар'єрними можливостями суттєво підвищує задоволеність вибором.

– Structure Discovery, тобто виявлення прихованої структури навчального матеріалу або груп студентів, які мають схожі стилі навчання, для подальшої адаптації контенту.

2.6 Архітектура інформаційних технологій та роль Generative AI

Реалізація розглянутих моделей вимагає створення комплексної інформаційної системи (Curriculum Management System - CMS), інтегрованої в IT-інфраструктуру університету.

Архітектурні рішення для автоматизованих систем управління навчальними планами

Сучасна система управління змістом освіти повинна будуватися на принципах сервіс-орієнтованої архітектури (SOA) або мікросервісів, забезпечуючи гнучкість та масштабованість [73].

Основні компоненти архітектури:

1. Data Storage Layer:

– Реляційні БД (PostgreSQL) для зберігання структурованих навчальних планів, навантаження, списків студентів.

– Графові БД (Neo4j) для зберігання онтологій, графів пререквізитів та зв'язків між компетентностями.

– Векторні БД (Pinecone, Milvus) для зберігання ембеддінгів текстів силабусів та вакансій для семантичного пошуку.

2. Business Logic Layer (Backend):

- Модуль аналізу графів (перевірка циклів, топологічне сортування).
- Модуль NLP (парсинг вакансій, видобуток навичок).
- Модуль оптимізації (генетичні алгоритми для розкладу та покриття компетентностей).

3. Presentation Layer (Frontend):

- Інтерактивні дашборди для гарантів програм (візуалізація skill gaps, карти покриття компетентностей).
- Кабінет студента для вибору траєкторії.

Система повинна мати API для інтеграції з LMS (Moodle), репозиторіями (DSpace) та зовнішніми сервісами (LinkedIn API).

Generative AI та архітектура RAG у проєктуванні курсів

Поява генеративного штучного інтелекту (GenAI) та великих мовних моделей (LLM, таких як GPT-4) відкриває еру AI-assisted Curriculum Design [73-75].

Просте використання чат-ботів є недостатнім через проблему «галюцинацій». Рішенням є архітектура RAG (Retrieval-Augmented Generation).

Принцип роботи RAG у контексті формування ОПП (узагальнену схему наведено на рисунку 2.2):

1. Retrieval (Пошук): на запит користувача (наприклад, «Створити силабус курсу 'Deep Learning' для магістрів») система спочатку шукає релевантну інформацію у власній довірній базі знань (стандарт 122, SWEBOK v4, внутрішні положення університету, кращі практики провідних університетів).

2. Augmentation (Доповнення): знайдені документи додаються до промпта як контекст.

3. Generation (Генерація): LLM генерує відповідь, базуючись виключно на наданому контексті, що гарантує відповідність стандартам та актуальність даних.

RAG (Retrieval-Augmented Generation)

Архітектура для генерації силабусів без «галюцинацій».

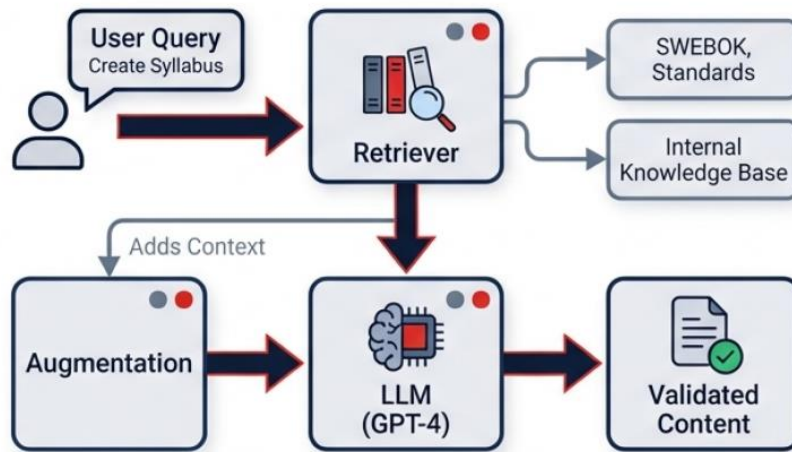


Рисунок 2.2 – Принцип роботи RAG у контексті формування ОПП

Сценарії використання GenAI:

- Автоматична генерація силабусів, тобто створення структури курсу, описів тем, списків літератури та завдань, що відповідають заданим компетентностям.
- Перетворення форматів, тобто автоматичне створення діаграм процесів (BPMN, Flowcharts) з текстових описів процедур навчання.
- Валідація та критика, тобто ШІ може аналізувати створений викладачем курс на предмет логічних помилок, дублювання матеріалу з іншими курсами або невідповідності принципам таксономії Блума.

– Персоналізація контенту, тобто генерація індивідуальних пояснень складних тем або прикладів коду, адаптованих до рівня знань конкретного студента.

Застосування Databricks та інших платформ для MLOps дозволяє розгортати та керувати такими моделями у захищеному контурі університету, забезпечуючи конфіденційність даних.

2.7 Висновки за розділом 2

Проведене дослідження дозволяє сформулювати наступні узагальнюючі висновки щодо моделей, методів та технологій підтримки формування змісту дисциплін спеціальності «Комп'ютерні науки»:

1. Необхідність системної формалізації, тобто ефективне управління змістом освіти в сучасних умовах неможливе без переходу від описових моделей до строгих математичних та онтологічних формалізацій. Використання теорії графів для моделювання структурно-логічних зв'язків та задач покриття множини для контролю компетентностей забезпечує необхідну жорсткість каркасу освітньої програми, на який нанизується гнучкий контент.

2. Data-Driven як стандарт, тобто формування змісту дисциплін повинно відбуватися у безперервному циклі зворотного зв'язку з ринком праці. Технології Text Mining та NLP дозволяють перетворити хаотичний потік вакансій на структуровані вимоги до навичок, мінімізуючи розрив між освітою та індустрією.

3. Онтології як інтегратор, тобто онтологічний підхід, активно розроблюваний українськими вченими, є ключовим для забезпечення семантичної єдності освітнього простору, інтеграції науки і освіти та створення інтелектуальних навчальних систем.

4. Персоналізація через алгоритми, тобто математичне моделювання індивідуальних освітніх траєкторій дозволяє реалізувати реальний, а не декларативний студентоцентризм, надаючи кожному здобувачу унікальний шлях професійного розвитку, оптимізований під його цілі та можливості.

5. Трансформація ролі викладача, тобто впровадження Generative AI та архітектур RAG знімає з методистів та викладачів рутинне навантаження з написання формальних документів, дозволяючи зосередитися на творчому та експертному наповненні курсів.

6. Глобальний контекст, тобто успішна освітня програма повинна базуватися на глибокій імплементації міжнародних стандартів (CC2020, SWEBOOK v4), адаптованих до національних вимог, що забезпечується через механізми мапінгу компетентностей та порівняльного аналізу.

Імплементація запропонованих підходів у вигляді єдиної інформаційної екосистеми технічного університету дозволить суттєво підвищити якість підготовки IT-фахівців, забезпечивши їх актуальними знаннями та навичками для успішної кар'єри в умовах цифрової економіки.

3 РОЗРОБКА ПРОЄКТУ ПРОГРАМНО-МЕТОДИЧНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ФОРМУВАННЯ ВМІСТУ ОСВІТНЬО- ПРОФЕСІЙНОЇ ПРОГРАМИ

В процесі виконання досліджень було проведено аналіз сучасного стану та вмісту освітніх програм, в Україні та світі. На рис. 3.1 наведена ієрархічна структура типів дисциплін у ОПП [9, 23, 25]: від фундаментальних (математика, теорія) до прикладних (технології), практикуму, проєктної роботи, курсових робіт і практики.

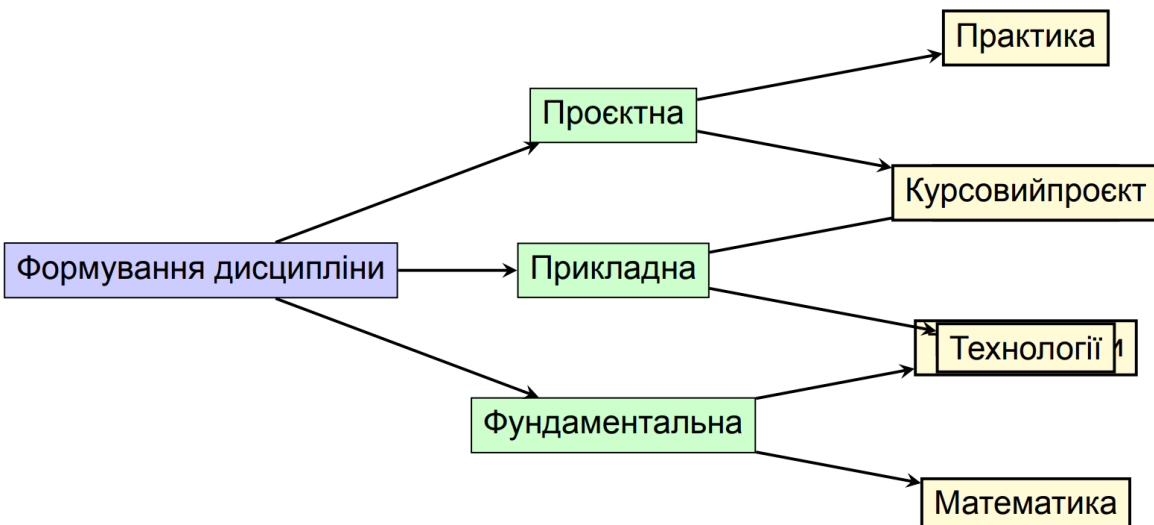


Рисунок 3.1 – Дерево класифікації типів дисциплін

Це дерево допомагає бачити баланс між теорією та практикою і зв'язки між компонентами програми. Від кореня «Формування дисципліни» гілки розходяться на категорії; кожна категорія задає тип навчального навантаження та очікувані результати (напр., теоретичні → базові

компетентності; практичні → прикладні навички). Без збалансування гілок програма або перевантажується теорією, або втрачає фундамент. Оптимально тримати співвідношення близько 40:60 (теорія : практика), що узгоджується з чек-листом якості програми. Пропонуються наступні рекомендації до реалізації даного підходу: перевірити покриття кожної гілки у навчальному плані; додати «мости» між теорією та практикою через інтегровані проекти й практикуми.

На рис. 3.2 наведені компоненти гібридної системи рекомендацій дисциплін [34, 35]: профіль програми, контентний аналіз, колаборативна фільтрація, база правил/знань → гібридні рекомендації → ранжування та відбір.

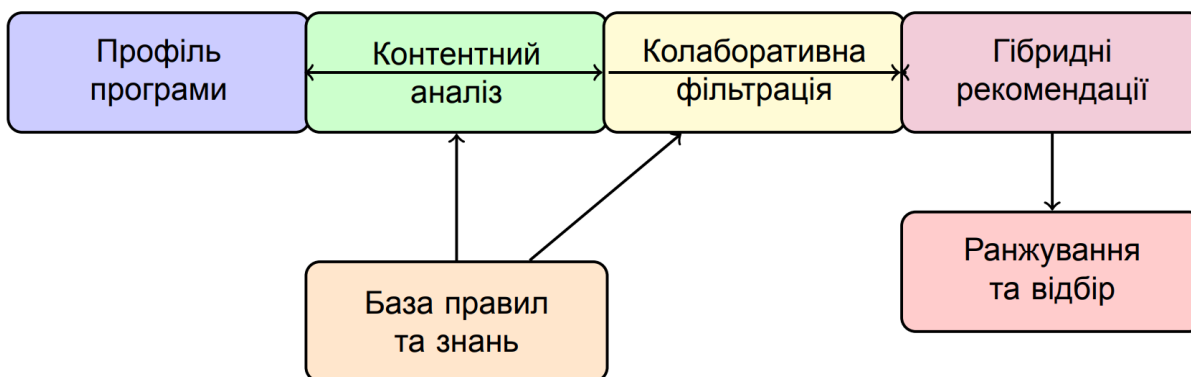


Рисунок 3.2 – Архітектура рекомендаційної системи

Потік даних проходить через кілька «моторів» рекомендацій – кожен додає свій внесок (контентні ознаки, досвід інших програм, експертні правила). На виході – список дисциплін, впорядкований за релевантністю та впливом на компетентності. Гібридний підхід знижує упередження одного методу та підсилює якість рекомендацій, особливо при обмежених

історичних даних. Пропонуються наступні рекомендації до реалізації даного підходу: регулярно оновлювати базу правил і ваги ранжування на основі зворотного зв'язку студентів/роботодавців, щоб рекомендації відображали актуальні тренди.

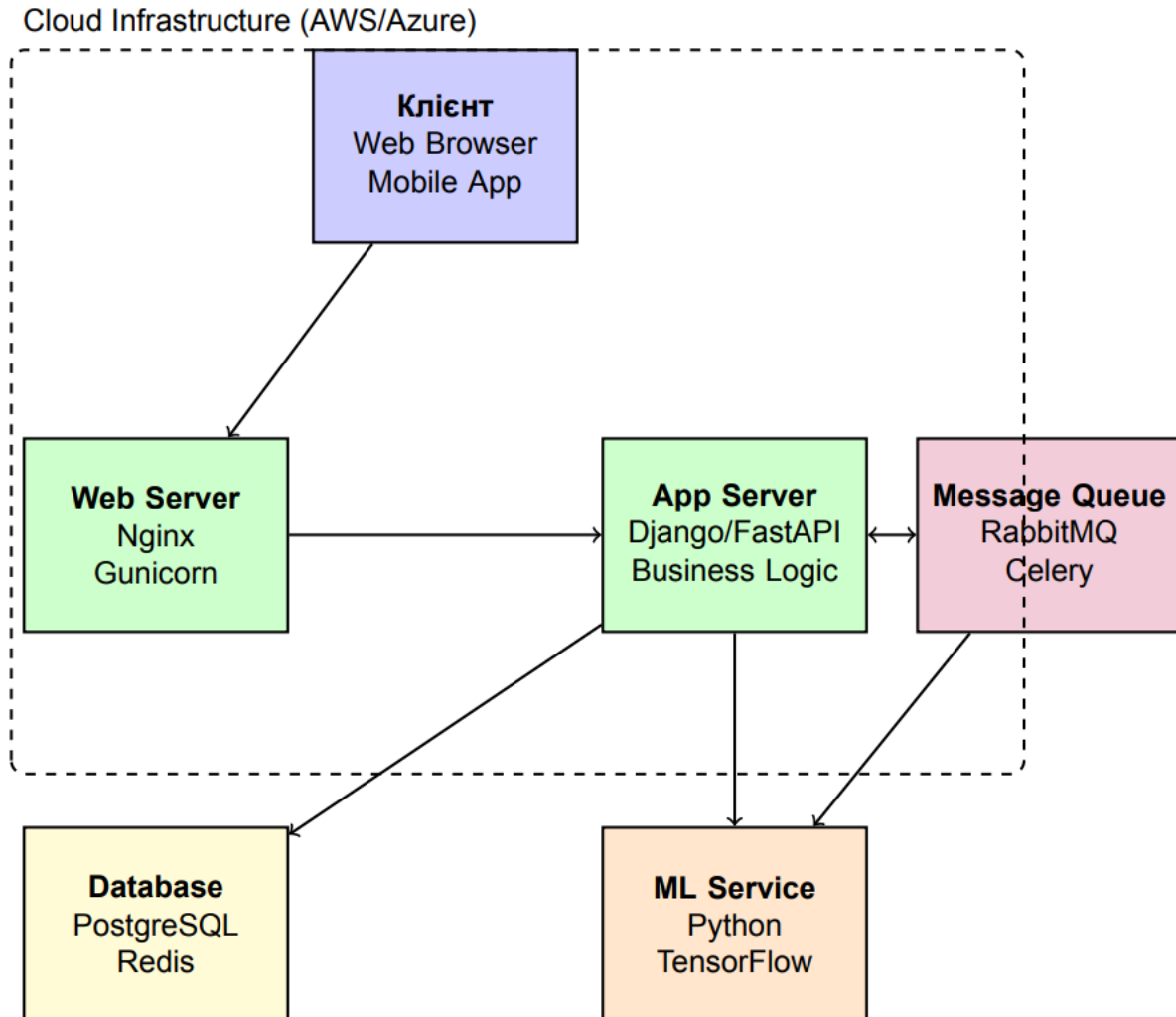


Рисунок 3.3 – Діаграма розгортання системи

За результатами аналізу предметної області розроблено проєкт програмно-методичного комплексу (системи). На рис. 3.3 наведена її

інфраструктура, від клієнтів (веб/мобайл) до web server (Nginx, Gunicorn), app server (Django/FastAPI), баз даних (PostgreSQL, Redis), ML-сервісів (TensorFlow), черг повідомлень (RabbitMQ/Celery), що працює у хмарі (AWS/Azure). Структура за шарами, тобто взаємодія користувача → веб-шар → бізнес-логіка → дані й кеш → окремі ML-компоненти та асинхронні задачі. Розділення відповідальностей підвищує масштабованість і надійність; асинхронність з Celery оптимізує важкі аналітичні завдання. Пропонуються наступні рекомендації до реалізації даного підходу: додати спостережність (metrics/логування) та автоматизоване розгортання (CI/CD) для швидкого оновлення сервісів із мінімальним простоем.

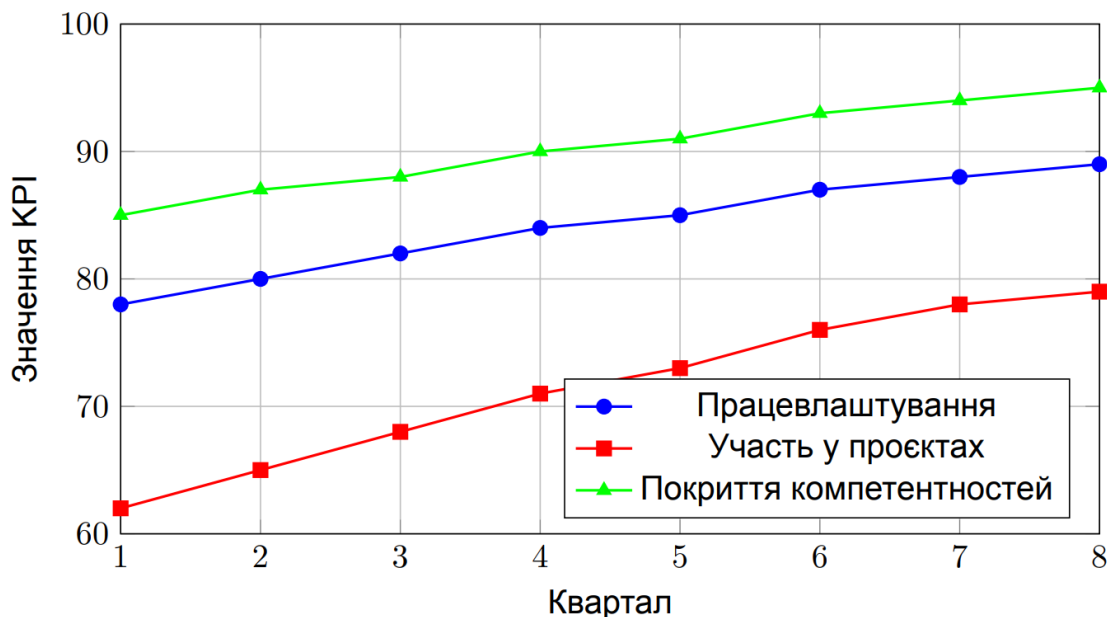


Рисунок 3.4 – Динаміка ключових показників ефективності (KPI)

На рис. 3.4 наведений лінійний графік змін KPI за кварталами: працевлаштування, участь у проєктах, покриття компетентностей – зі співставленням поточних і цільових значень у таблиці KPI [29, 30]. Вісі: X – квартали, Y – значення показників (%). Порівняйте траєкторію кожного KPI

з цілями (напр., працевлаштування: поточно 78% → ціль 90%, вага 0.25). Найбільший вплив на інтегральну оцінку мають працевлаштування і покриття компетентностей (високі ваги). Динаміка підказує, де зосередити ресурс. Пропонуються наступні рекомендації до реалізації даного підходу: запровадити квартальні «коригувальні дії» (нові партнерства, проектні курси, кар'єрні сервіси) для метрик із відставанням.

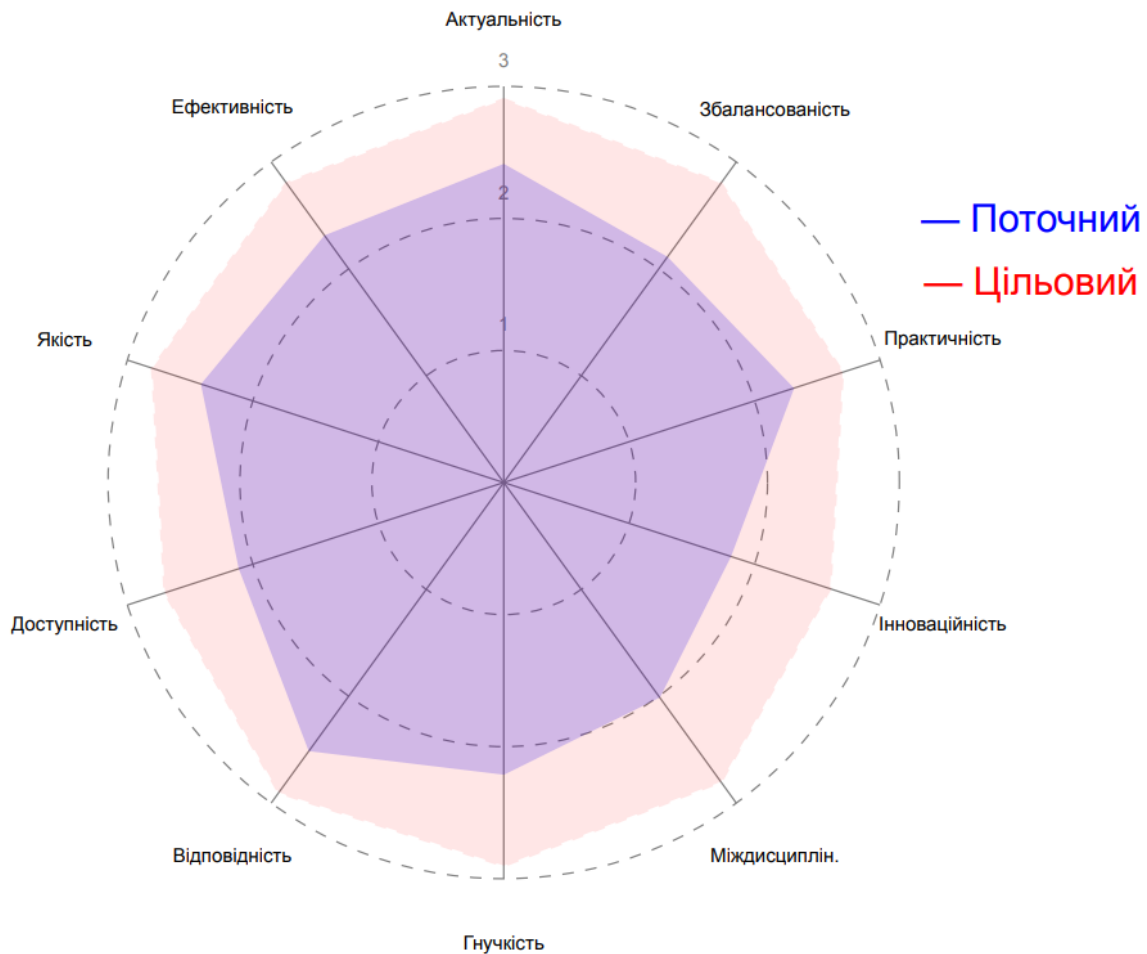


Рисунок 3.5 – Радарна діаграма якості ОПП (0–3 бали)

На рис. 3.5 наведене порівняння «Поточний» vs «Цільовий» профілів за кількома вимірами якості: актуальність, збалансованість, практичність,

інноваційність, міждисциплінарність, гнучкість, відповідність, доступність, якість, ефективність. Кожна вісь – окремий критерій, значення 0–3; площа та форма багатокутників показують сильні/слабкі сторони. Розрив між «Поточним» і «Цільовим» – пріоритет для вдосконалення. Якщо «Практичність» чи «Інноваційність» мають менші бали – слід підсилювати варіативну складову (25–30%), впроваджувати сучасні технології. Пропонуються наступні рекомендації до реалізації даного підходу: використовувати діаграму разом із PDCA (рис. 3.6) для циклічного підвищення кожної осі до цільових значень.

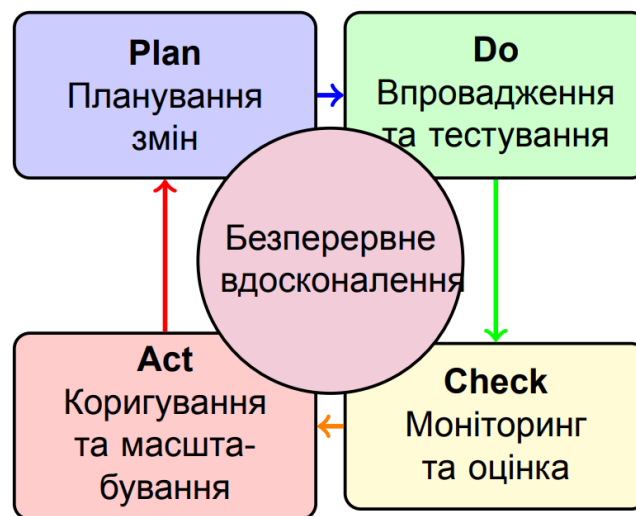


Рисунок 3.6 – Цикл PDCA для вдосконалення ОПП

На рис. 3.6 наведений безперервний цикл управління якістю [34, 41]: Plan (план змін) → Do (впровадження) → Check (моніторинг) → Act (коригування та масштабування). Починаємо з плану на основі даних (рис. 3.4–3.5), після реалізації оцінюємо KPI, приймаємо коригувальні рішення та закладаємо їх у наступний цикл. PDCA формалізує регулярне оновлення

ОПП і мінімізує ризик застою чи «точкових» несистемних змін. Пропонуються наступні рекомендації до реалізації даного підходу: вбудувати PDCA у календар ОПП (семестрово/квартально) з відповідальними ролями і контрольними артефактами (звіти, рішення ради роботодавців).

На рис. 3.7 наведено кільце зацікавлених сторін: студенти, викладачі, роботодавці, випускники, ІТ-індустрія, регулятори, партнери, суспільство → потрапляє в «ОПП: коригування та адаптація». Кожен вузол – джерело даних; їхні сигнали треба перетворювати на рішення щодо змісту і структури програми. Широка екосистема забезпечує стійкість і релевантність програми; відсутність хоча б одного каналу збіднює картину попиту. Пропонуються наступні рекомендації до реалізації даного підходу: інтегрувати інструменти збору (опитування, фокус-групи, аналітика) у єдину панель моніторингу.

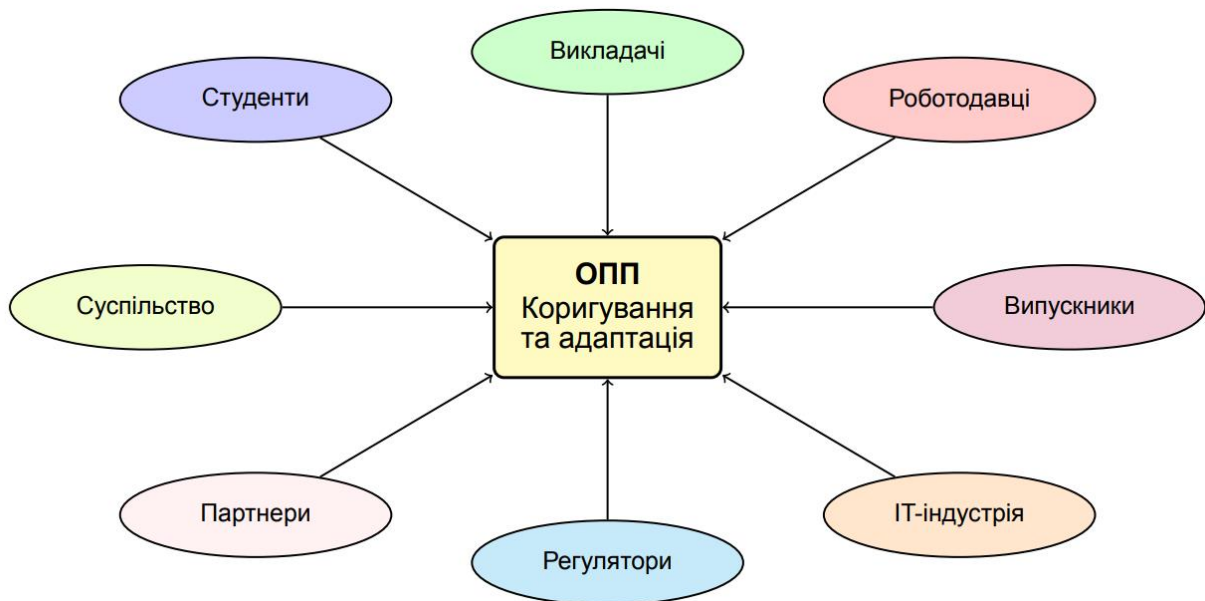


Рисунок 3.7 – Екосистема зворотного зв'язку для ОПП

На рис. 3.8 наведений потік: моніторинг середовища (ринкові тренди, технологічні зміни, зворотний зв'язок) → аналіз/прогнозування → прийняття рішень → оновлення змісту, модифікація структури, впровадження інновацій → адаптована ОПП. Це замкнена петля: зовнішні сигнали трансформуються у конкретні зміни плану та методик. Адаптивна ОПП швидше реагує на зміни ринку й технологій; критично мати процеси прогнозування, а не лише ретроспективний аналіз. Пропонуються наступні рекомендації до реалізації даного підходу: додати квартальну «аналітичну вітрину» з трендами (вакансії, стек технологій, вимоги до навичок) та матрицею впливу на курси.

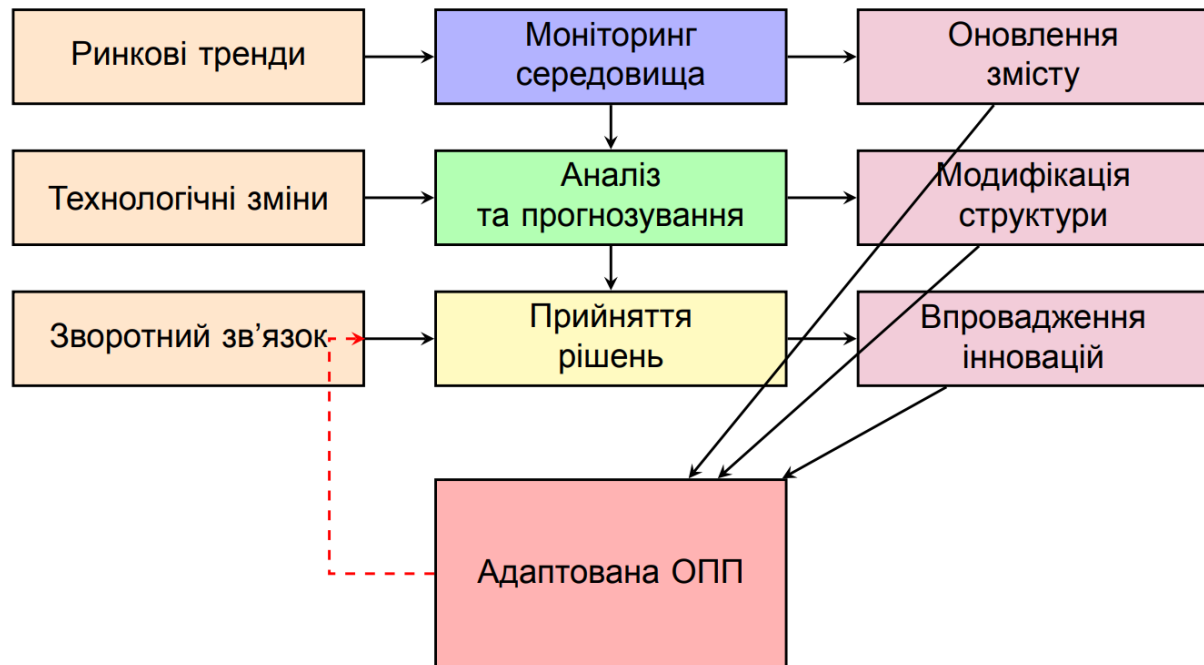


Рисунок 3.8 – Модель адаптивної освітньо-професійної програми

Для розробки загального підходу, було проведено пілотне оцінювання існуючої ОПП з комп'ютерних наук.

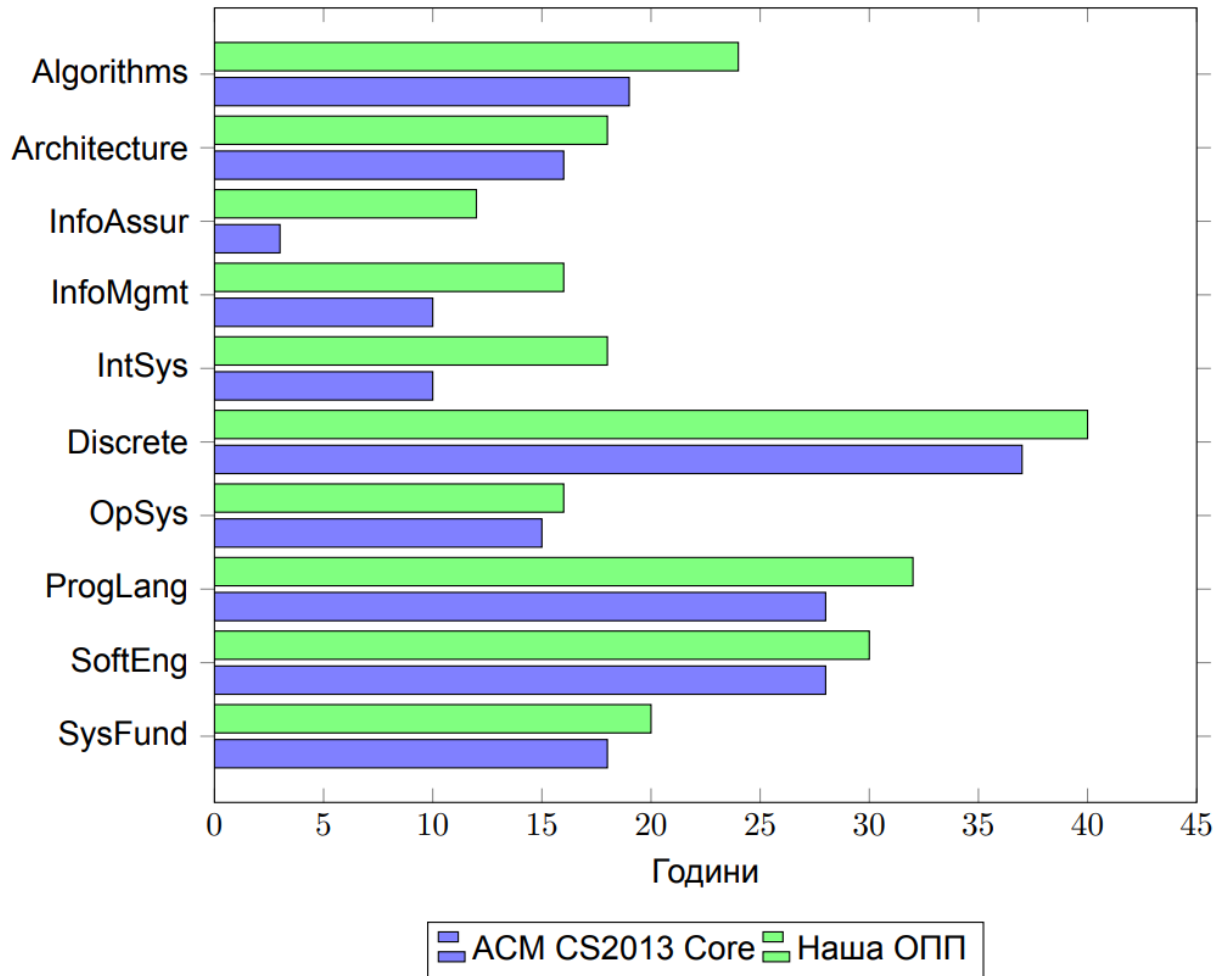


Рисунок 3.9 – Порівняння з вимогами ACM CS2013

На рис. 3.9 наведений бар-чарт годин Core у ACM CS2013 vs години у ОПП, що аналізується, по областях знань: Algorithms, Architecture, Discrete Structures, Operating Systems, Programming Languages, Software Engineering тощо. Загальне покриття: 210 годин проти 165 Core (127%). Для кожної КА було порівняно два стовпці; значення >100% означає посилене покриття (напр., Information Assurance 400%, Computational Science 800%), 100–110% – узгоджене, <100% – потенційна прогалина. Програма загалом перевищує Core-мінімум; важливо відстежити, чи «перенасичені» деякі області не зменшують час на інші критичні (баланс і навантаження студентів).

Пропонуються наступні рекомендації до реалізації даного підходу: провести перерозподіл годин, щоб уникнути надлишку у малоприкладних для бакалаврату темах, і зміцнити слабші з високим ринковим попитом (HCI, Networking, Platform-Based Development).



Рисунок 3.10 – Мапування рівнів SFIA на структуру ОПП

На рис. 3.10 наведена відповідність рівнів SFIA (1–7) етапам навчання: 1 – початківець (1 курс), 2 – допомога (1 курс), 3 – застосування (2 курс), 4 – сприяння (3 курс), 5 – консультування (4 курс), 6 – ініціювання та вплив (магістратура), 7 – стратегічне керівництво (після магістратури). Рівні компетентностей нарощуються сходинками; курси і практики мають підводити студента до цільового рівня на кожному етапі. Чітке відображення дозволяє проєктувати результати навчання (РН) і форми контролю

відповідно до очікуваного рівня зрілості навичок. Пропонуються наступні рекомендації до реалізації даного підходу: для кожного курсу вказати цільовий SFIA рівень у силабусі та перевірити, що оцінювання (проекти/лабораторні) компонентно веде до цього рівня.

На рис. 3.11 наведена хронологія прийняття EdTech [65-67]: 2025–26 – AI-тьютори, персоналізація; 2027–28 – VR/AR лабораторії, адаптивні платформи; 2029–30 – нейроінтерфейси, квантові симулятори; 2031+ – повна віртуалізація освіти. Поетапне впровадження з урахуванням технологічної зрілості й готовності інфраструктури. Ранній фокус на AI-тьюторах та адаптивності дає швидкі вигоди для персоналізації, тоді як VR/AR потребує інвестування у лабораторії; далекоглядні напрями (BCI, квантові симуляції) – експериментальні. Пропонуються наступні рекомендації до реалізації даного підходу: створити пілоти і KPI впровадження на кожну фазу; забезпечити етичні та безпекові рамки при застосуванні AI/VR.

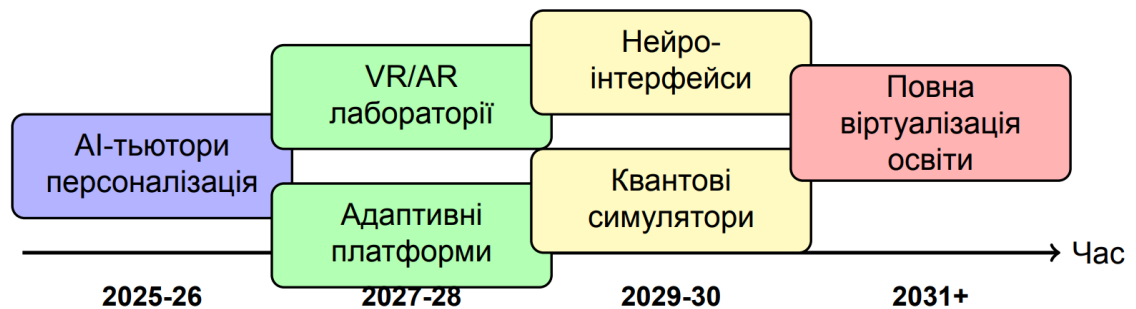


Рисунок 3.11 – Дорожня карта впровадження інноваційних технологій

На рис. 3.12 наведено побудовану за результатами досліджень діаграму «ймовірність × вплив» з розміщеними ризиками: швидке

застарівання технологій, недостатнє фінансування, втрата викладачів, зміна стандартів, зниження мотивації студентів. Ризики у квадранті «високий вплив, висока ймовірність» – пріоритет для негайних планів реагування; «високий вплив, низька ймовірність» – тримати резервні сценарії. Найкритичніші для ІТ-ОПП ризики – застарівання технологій та втрата викладачів; вони безпосередньо впливають на якість і конкурентоспроможність. Пропонуються наступні рекомендації до реалізації даного підходу: ввести «дорожню карту компетенцій викладачів», спільні курси з індустрією, фонд оновлення обладнання/контенту, сценарії швидкого перегляду освітньої та робочих програм при зміні стандартів.

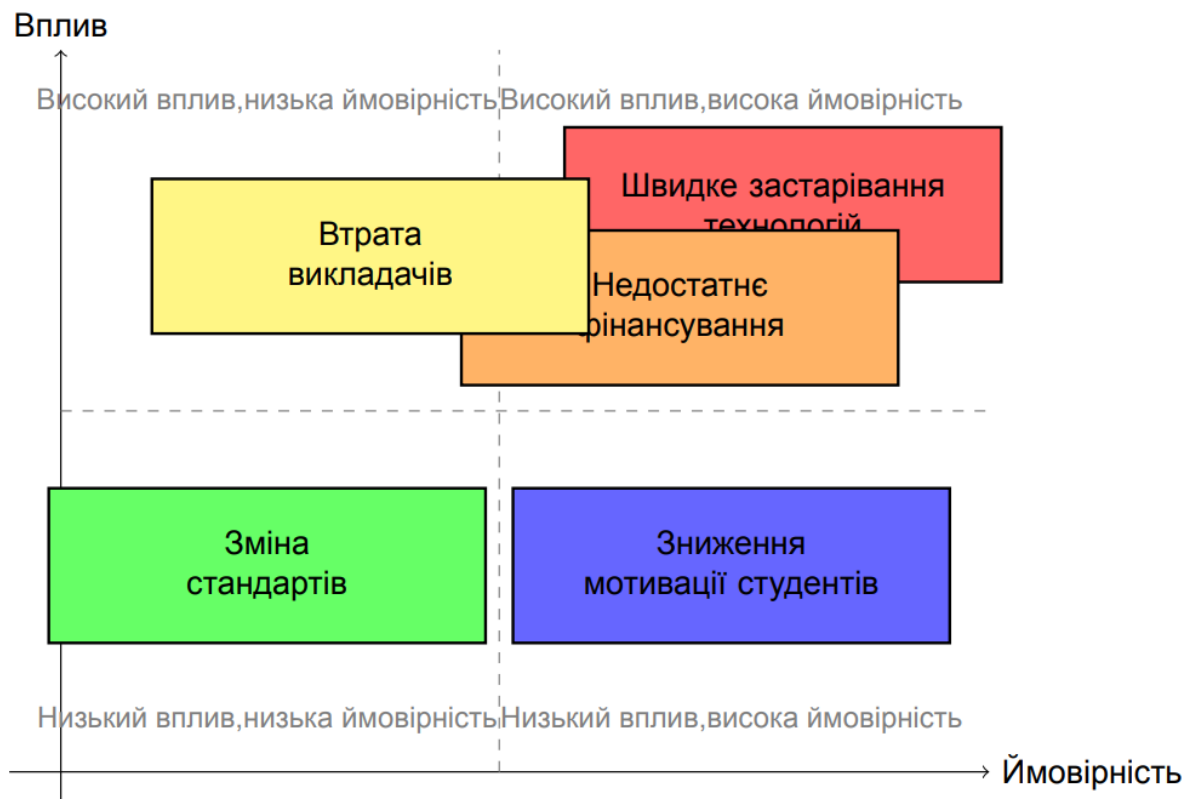


Рисунок 3.12 – Матриця ризиків формування та впровадження ОПП

На рис. 3.13 наведена послідовність: візія та стратегія → моделі та методи → інформаційна система → впровадження → моніторинг та адаптація → цикл безперервного вдосконалення. Це «карта» управління всією програмою. Схема інтегрує попередні блоки: модель (рис. 3.1–3.2), технології і розгортання (рис. 3.3), вимірювання (рис. 3.4–3.6), зворотний зв'язок/адаптація (рис. 3.7–3.8), стандарти (рис. 3.9–3.10), інновації/ризик (рис. 3.11–3.12). Системний підхід замикає стратегічні і операційні процеси у єдину керовану петлю розвитку ОПП. Пропонуються наступні рекомендації до реалізації даного підходу: оформити це як операційну рамку в університеті: визначити відповідальних, артефакти (стандарти, регламенти), циклічні огляди, цифрову панель показників.

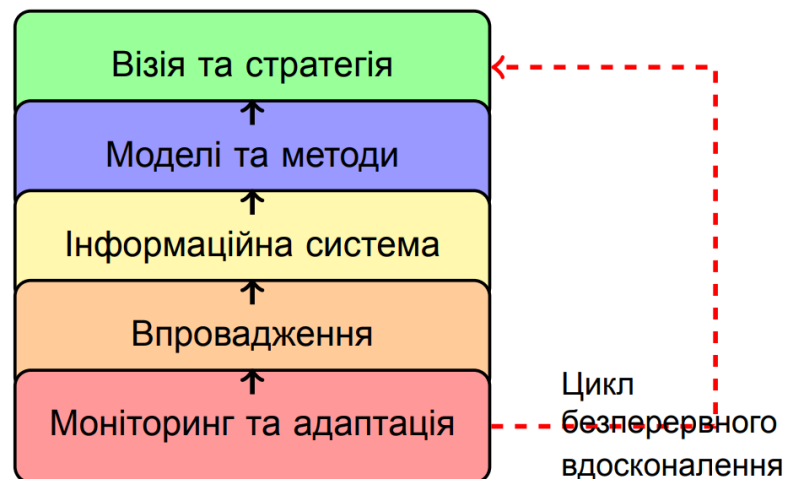


Рисунок 13 – Загальна схема впровадження результатів дослідження

Висновки до розділу 3

У третьому розділі розроблено проект програмно-методичного забезпечення для інформаційної підтримки формування змісту освітньо-професійної програми (ОПП) за спеціальністю «Комп'ютерні науки». За результатами проведених досліджень сформульовано наступні висновки.

1. Запропоновано ієрархічну модель структури дисциплін, що базується на деревоподібній класифікації від фундаментальних до прикладних компонентів. Встановлено, що оптимальний баланс між теорією та практикою має складати 40:60. Такий підхід дозволяє уникнути перевантаження програми академічним матеріалом та забезпечити формування прикладних навичок через систему інтегрованих проектів.

2. Спроектовано архітектуру гібридної рекомендаційної системи, яка поєднує контентний аналіз, колаборативну фільтрацію та базу експертних знань. Це забезпечує об'єктивність при відборі дисциплін та їх ранжуванні за рівнем впливу на формування ключових компетентностей, мінімізуючи суб'єктивізм при розробці навчальних планів.

3. Обґрунтовано інфраструктурне рішення програмного комплексу, побудоване на сучасних хмарних технологіях (AWS/Azure) із використанням мікросервісної архітектури (Django/FastAPI, PostgreSQL, Redis). Впровадження асинхронних черг завдань (Celery/RabbitMQ) та ML-сервісів дозволяє системі ефективно опрацьовувати великі масиви даних щодо ринкових трендів та освітніх стандартів.

4. Розроблено систему моніторингу якості ОПП на основі циклу PDCA (*Plan-Do-Check-Act*) та радарних діаграм. Встановлено 10 вимірів якості (інноваційність, гнучкість, практичність тощо), що дозволяє візуалізувати розрив між поточним та цільовим станом програми та приймати

обґрунтовані управлінські рішення щодо її коригування.

5. Виконано відображення рівнів компетентності за методологією SFIA на етапи навчання. Це забезпечує послідовне зростання професійної зрілості студента: від рівня «початківець» на першому курсі до рівня «консультування» на етапі випуску, що дозволяє чітко формулювати результати навчання для кожного силабусу.

6. Сформовано дорожню карту впровадження інновацій та матрицю ризиків. Визначено, що найбільш критичними ризиками для ІТ-програм є застарівання технологій та відтік кадрів. Запропоновано механізми адаптивної модифікації змісту ОПП через безперервний зворотний зв'язок із екосистемою зацікавлених сторін (роботодавці, випускники, індустрія).

7. Синтезовано загальну схему впровадження результатів дослідження, яка об'єднує стратегічне планування, технологічну платформу та систему безперервного вдосконалення в єдину операційну рамку університету. Це гарантує стійкість та конкурентоспроможність освітньої програми в умовах динамічного ІТ-ринку.

4 СТВОРЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ПІДТРИМКИ РОЗРОБКИ ВМІСТУ ДИСЦИПЛІН ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ ЗА СПЕЦІАЛЬНІСТЮ «КОМП'ЮТЕРНІ НАУКИ»

За результатами проведених критичного аналізу, проєктування програмно-методичного забезпечення та математичного моделювання розроблено програмний комплекс для інформаційної підтримки процесу розробки вмісту дисциплін освітньо-професійної програми за спеціальністю «Комп'ютерні науки».

Фізично реалізовано програмний продукт, який являє собою односторінковий вебзастосунок (SPA) на базі бібліотеки React [73-75], призначений для аналізу відповідностей між результатами навчання (РН) та компетентностями (ЗК/СК) для спеціальності "Комп'ютерні науки". Програмний комплекс забезпечує інформаційну підтримку діяльності груп забезпечення розвитку освітньо-професійних програм спеціальності «Комп'ютерні науки» за рахунок наочної візуалізації відповідностей РН та ЗК і СК та відображення текстового визначення формулювань у відповідності до існуючого стандарту освіти за даною спеціальністю.

Нижче наведено детальний опис та UML-схеми архітектури.

4.1 Опис функціоналу та розробка логіки роботи програмного комплексу

Технологічний стек, який було використано в даному проєкті, складається з наступних компонентів:

- React 18 використовується для побудови інтерфейсу та керування станом.
- Tailwind CSS забезпечує сучасну адаптивну верстку без написання кастомних CSS-файлів.
- Babel дозволяє виконувати JSX-код безпосередньо у браузері (через CDN).

Логіка роботи програмного комплексу полягає в наступному.

Застосунок працює у двох режимах, які перемикаються кнопками (searchMode):

1. Прямий пошук (РН → ЗК/СК). Користувач обирає результати навчання, а програма виводить усі пов'язані з ними компетентності згідно з об'єктом matrix.

2. Зворотний пошук (ЗК/СК → РН). Користувач обирає компетентності, а програма виводить РН, які їх забезпечують. Для цього динамічно обчислюється reverseMatrix (інвертований індекс).

Ключові компоненти коду:

- Дані (Model): об'єкт matrix містить зв'язки, а descriptions – текстові розшифровки кожного коду.
- Стан (State): * selectedItem: масив обраних кодів.
- searchMode: поточний напрямок аналізу.
- Обчислення (Logic): використовується useMemo для оптимізації. reverseMatrix будується лише раз, а results фільтруються при зміні вибору.
- UI (View): головний компонент CompetencyAnalyzer та допоміжний ItemBadge, який відповідає за відображення плитки з кодом та спливаючу підказку (tooltip).

4.2 Проєктування програмного комплексу з використанням UML-діаграм та фізична реалізація програми

На рис. 4.1 наведена діаграма класів (структура компонентів та даних). Ця діаграма описує структуру застосунку, властивості компонентів та зв'язки між даними.

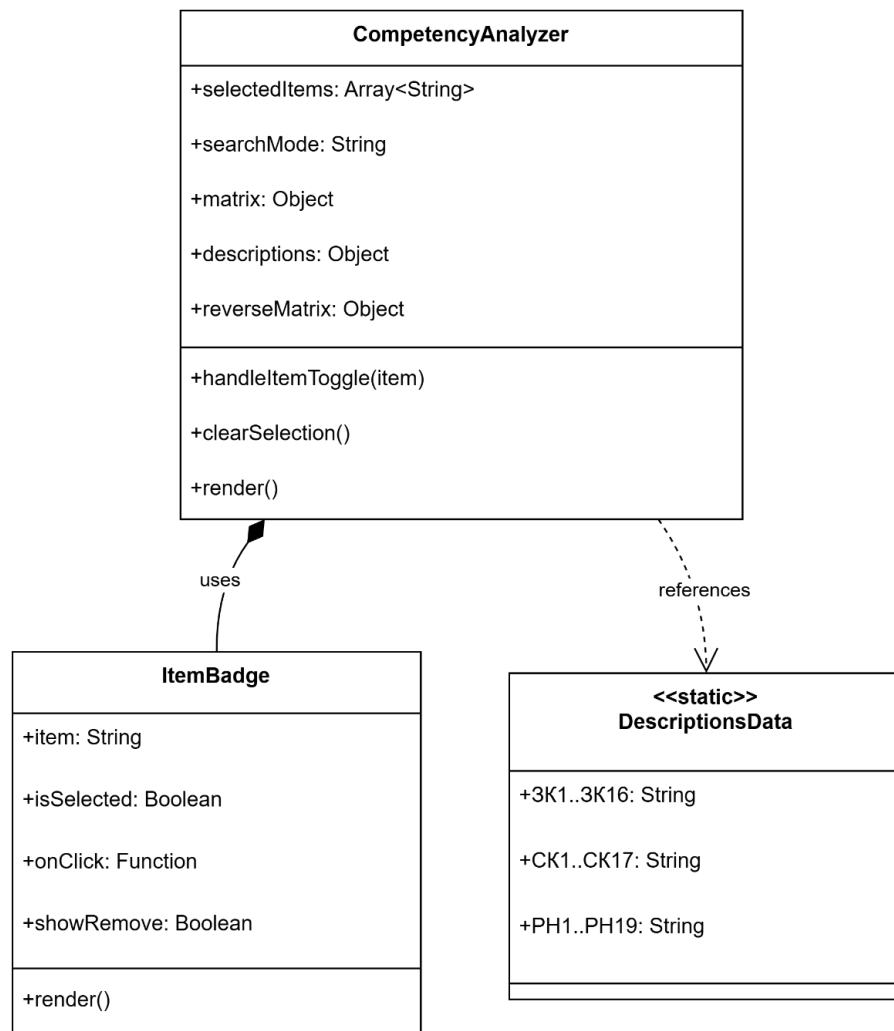


Рисунок 4.1 – Діаграма класів (структура компонентів та даних)

На рис. 4.2 наведена діаграма послідовності (процес взаємодії користувача), яка описує, що відбувається, коли користувач натискає на плитку (Badge).

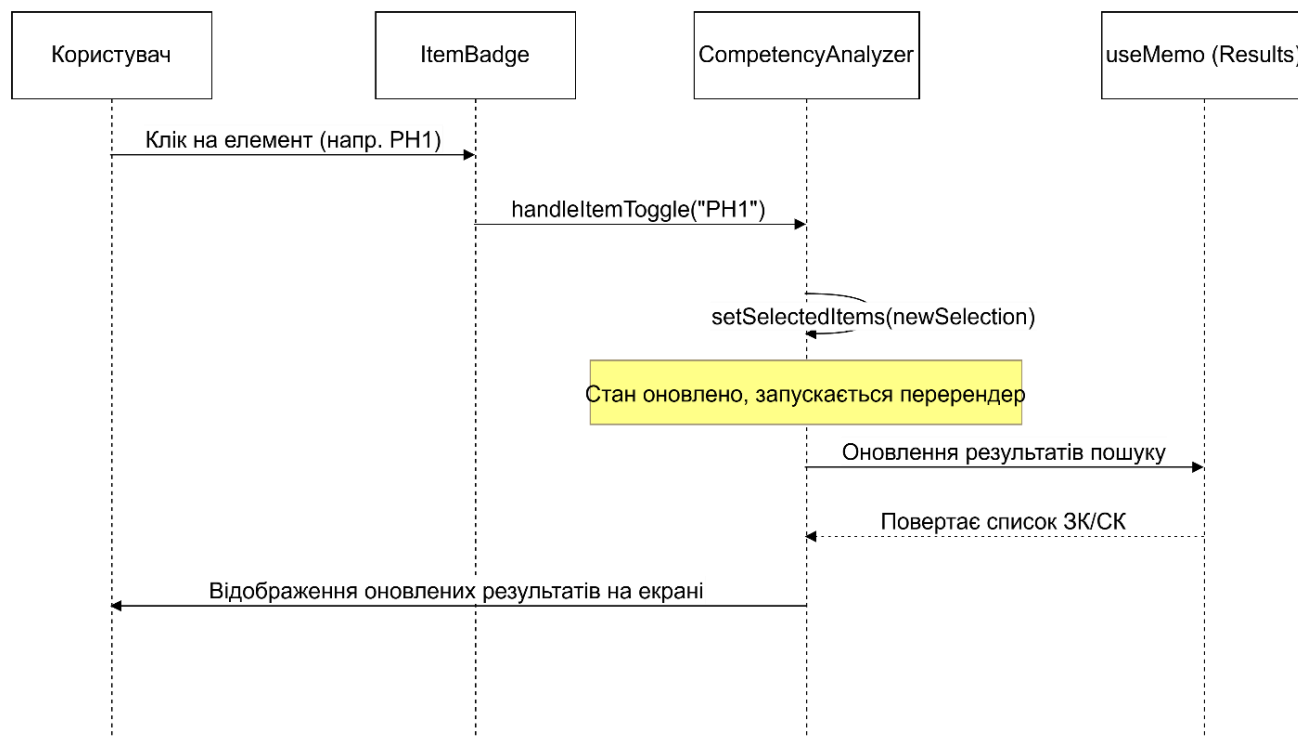


Рисунок 4.2 – Діаграма послідовності (процес взаємодії користувача)

На рис. 4.3 наведена діаграма компонентів (архітектура системи), яка показує високорівневу організацію застосунку.

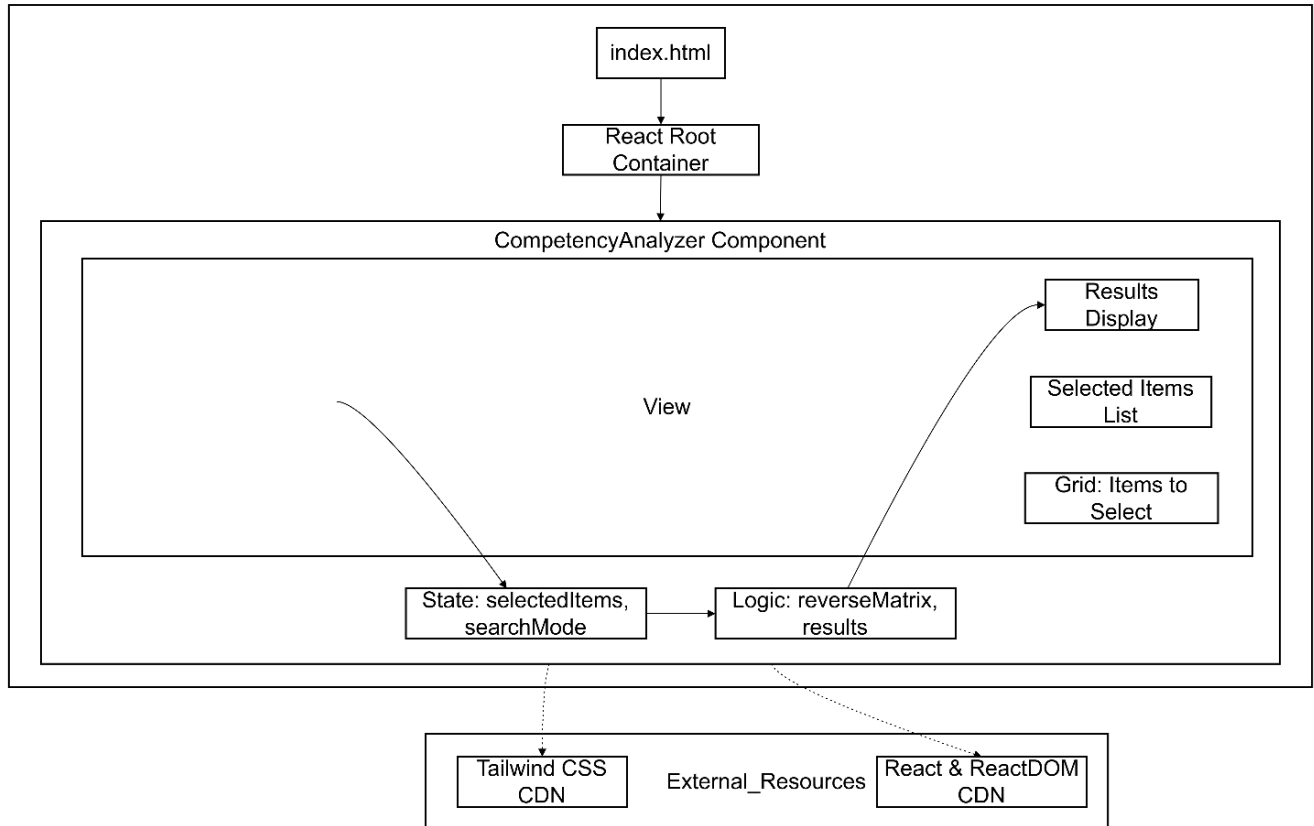


Рисунок 4.3 – Діаграма компонентів (архітектура системи)

Особливості архітектурних рішень, які реалізовано у програмному комплексі, полягають у наступному.

1. Декларативність, тобто весь інтерфейс автоматично реагує на зміну стану `selectedItems`. Немає прямого маніпулювання DOM.
2. Інвертування індексу, тобто створення `reverseMatrix` "на льоту" дозволяє використовувати одну й ту саму структуру даних для двостороннього пошуку, що зменшує ризик помилок при оновленні даних.
3. UX-елементи:
 - Tooltips: реалізовані через CSS-класи `group` та `group-hover` у Tailwind, що забезпечує високу продуктивність без додаткових JS-бібліотек.

– SVG-іконки: використовуються вбудовані компоненти-іконки (Search, Book, Target, Zap) для візуального розділення розділів.

На рис. 4.4 наведено діалогове вікно програмного комплексу, який забезпечує візуалізацію корисного інформаційного забезпечення та взаємодію з користувачем. Діалогове вікно на даному рисунку працює в режимі візуалізації відповідності РН та ЗК і СК. На рис. 4.5 наведено діалогове вікно програмного комплексу в режимі візуалізації відповідності ЗК і СК та РН.

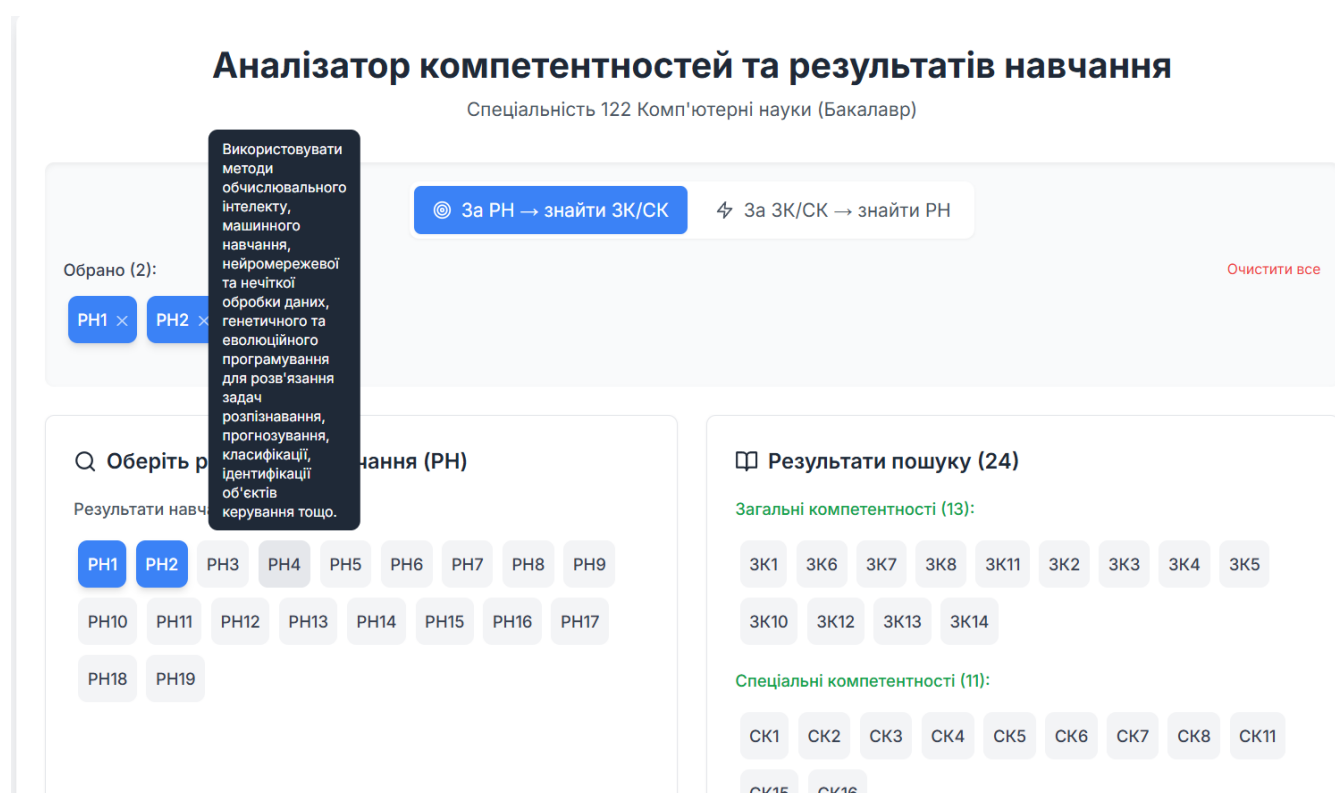


Рисунок 4.4 – Діалогове вікно програмного комплексу в режимі візуалізації відповідності ЗК і СК та РН

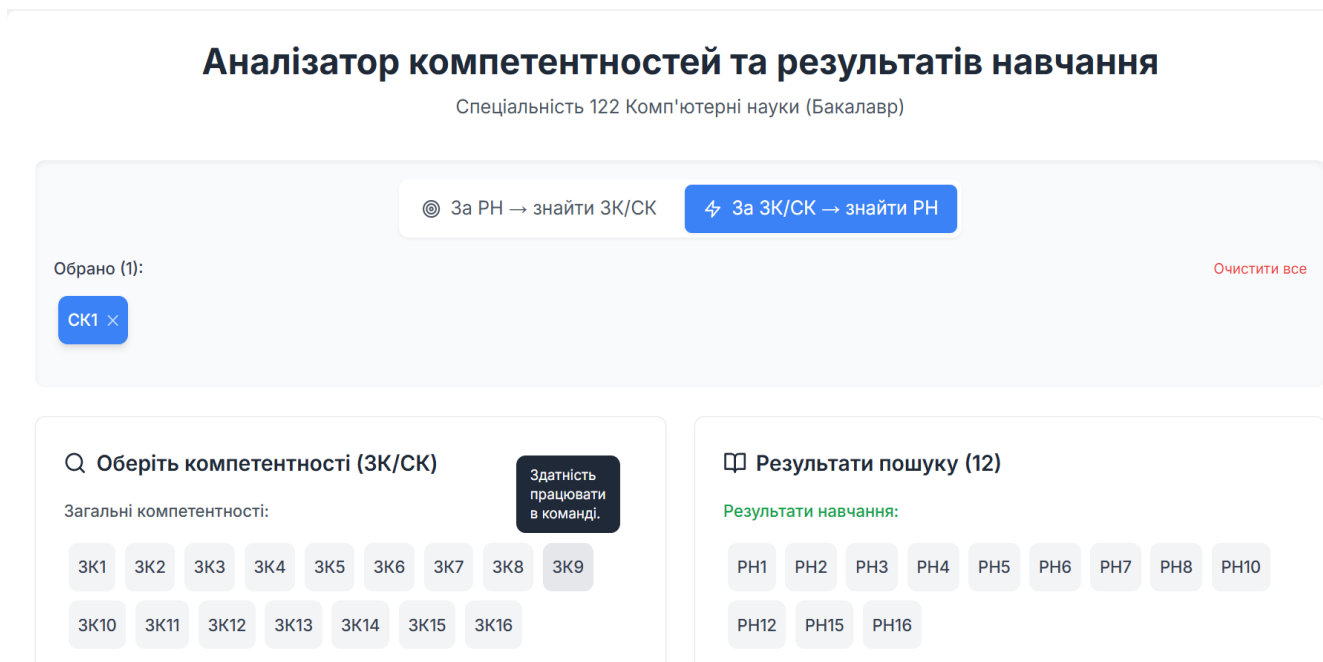


Рисунок 4.5 – Діалогове вікно програмного комплексу в режимі візуалізації відповідності РН та ЗК і СК

Скорочений лістинг програми для інформаційної підтримки процесу розробки вмісту дисциплін освітньо-професійної програми за спеціальністю «Комп'ютерні науки» наведено у Додатку Б.

4.3 Висновки за розділом 4

У четвертому розділі виконано практичну реалізацію програмного комплексу, призначеного для автоматизації процесів формування та аналізу змісту освітньо-професійних програм (ОПП) спеціальності 122 «Комп'ютерні науки». За результатами розробки зроблено такі висновки:

1. Розроблено та фізично реалізовано програмний продукт у форматі односторінкового вебзастосунку (SPA) на базі сучасної бібліотеки React 18. Обраний технологічний стек (React, Tailwind CSS, Babel) забезпечує високу швидкість роботи інтерфейсу, кросплатформенність та адаптивність, що дозволяє використовувати комплекс на різних типах пристроїв без втрати функціональності.

2. Реалізовано дворівневу логіку аналізу даних, яка базується на прямому та зворотному пошуку відповідностей між програмними результатами навчання (РН) та компетентностями (ЗК/СК). Використання механізму інвертованого індексу (reverseMatrix) гарантує цілісність даних та дозволяє групі забезпечення ОПП оперативно виявляти критичні розриви у покритті стандартів освіти.

3. Спроектовано архітектуру системи з використанням мови моделювання UML. Побудовані діаграми класів, послідовності та компонентів дозволили чітко розмежувати рівні моделі (даних), логіки (обчислень) та представлення (інтерфейсу), що забезпечує масштабованість та легкість підтримки програмного коду в майбутньому.

4. Впроваджено передові підходи до побудови UI/UX, зокрема декларативне керування станом та інтерактивну візуалізацію через систему Tooltips (спливаючих підказок). Це забезпечує наочність текстових формулювань стандарту без перевантаження робочого простору, сприяючи швидкому прийняттю рішень при проектуванні вмісту навчальних дисциплін.

5. Доведено ефективність архітектурного рішення щодо динамічного обчислення зв'язків «на льоту». Такий підхід мінімізує ризик виникнення помилок при оновленні нормативної бази (зміні формулювань у стандарті), оскільки вимагає редагування лише одного об'єкта даних (matrix), після чого весь інтерфейс та логіка зворотного пошуку оновлюються автоматично.

6. Практична значимість розробленого комплексу полягає у створенні інструментального середовища для розробників ОПП, яке дозволяє верифікувати відповідність освітніх компонентів вимогам Міністерства освіти і науки України та міжнародним рекомендаціям, значно скорочуючи часові витрати на ручну обробку матриць відповідності.

5 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ЕФЕКТИВНОСТІ РОЗРОБКИ

Впровадження інформаційної технології підтримки формування змісту освітньо-професійних програм (ОПП) у технічному університеті спрямоване на оптимізацію роботи гарантів програм, методистів та викладачів. Економічна ефективність від реалізації даного проєкту поділяється на пряму (економія ресурсів) та непрямую (підвищення якості підготовки фахівців).

Аналіз витрат на розробку та впровадження

Витрати на створення програмного комплексу (Single Page Application на базі React) включають:

- витрати на оплату праці: розробка логіки інвертованого індексу для двостороннього пошуку (PH -> ЗК/СК) , проєктування UML-діаграм та архітектури системи.
- Інфраструктурні витрати: використання хмарної інфраструктури (AWS/Azure) для розгортання та збереження даних.
- Методичні витрати: формалізація зв'язків між 16 загальними, 17 фаховими компетентностями та 19 програмними результатами навчання згідно зі Стандартом 122 «Комп'ютерні науки».

Прямий економічний ефект (економія часу)

Основним джерелом прямої ефективності є автоматизація рутинних операцій. Використання розробленого програмного комплексу дозволяє отримати економію часу з відповідних джерел, які наведено у таблиці 5.1.

Таблиця 5.1 – Джерела економії часу від використання розробленого програмного комплексу

Операція	Традиційний підхід (люд.-год)	З використанням ІТ (люд.-год)	Економія часу
Перевірка покриття ПРН компетентностями	8.0	0.5	94%
Формування матриць відповідності ОПП	12.0	1.0	92%
Аналіз розривів навичок (Skill Gap Analysis)	16.0	2.0	87%
Коригування змісту дисциплін під нові стандарти	10.0	1.5	85%

Загальний річний економічний ефект (E_{year}) розраховується як:

$$E_{year} = (T_{old} - T_{new}) \times C_{hour} \times N - Z_{exp}$$

де:

T_{old}, T_{new} – часові витрати до та після впровадження;

C_{hour} – вартість години роботи експерта/викладача;

N – кількість циклів оновлення ОПП на рік;

Z_{exp} – витрати на підтримку системи.

Непряма ефективність та стратегічні переваги

Крім прямої економії, впровадження системи забезпечує:

- Зниження ризику невідповідності стандартам, тобто автоматична валідація запобігає «пропалинам» у покритті компетентностей, що критично для акредитації ОПП.

- Адаптивність до ринку праці: використання Text Mining та NLP для аналізу вакансій дозволяє оперативно впроваджувати актуальні технології (наприклад, Generative AI, MLOps, Docker).

- Підвищення конкурентоспроможності випускників: точне мапування рівнів SFIA на етапи навчання забезпечує відповідність кваліфікації випускників вимогам міжнародних IT-компаній та партнерів, таких як група «Метінвест».

Оцінка ризиків. Економічна стабільність проєкту підтримується через матрицю ризиків «ймовірність x вплив». Найбільш критичні ризики – швидке застарівання технологій та зміна стандартів – нівелюються за рахунок гнучкої архітектури системи (SPA + API) та можливості динамічного оновлення бази знань.

Висновки за розділом 5

Впровадження розроблених методів та програмного комплексу дозволяє перейти до моделі *Data-Driven Curriculum Design*, що скорочує термін підготовки методичної документації у 8-10 разів та мінімізує «skill gap» між університетською освітою та реальними потребами індустрії.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-практичне завдання з автоматизації та інформаційної підтримки процесів формування змісту освітньо-професійних програм (ОПП) у технічному університеті. На основі проведених досліджень та розробок отримано наступні результати.

1. Проаналізовано стан питання та виявлено, що традиційні методи формування ОПП характеризуються високою трудомісткістю та ризиком виникнення «розривів» між вимогами освітніх стандартів та фактичним змістом дисциплін. Встановлено необхідність впровадження інструментів Data-Driven Curriculum Design для забезпечення відповідності програм вимогам ІТ-індустрії.

2. Формалізовано моделі зв'язків між компонентами освітнього процесу: загальними та спеціальними компетентностями, програмними результатами навчання та змістом навчальних дисциплін. Використання графових моделей та матриць відповідності дозволило структурувати дані для автоматизованого аналізу.

3. Розроблено метод аналізу покриття компетентностей, що базується на використанні інвертованого індексу для швидкого двостороннього пошуку зв'язків. Метод дозволяє виявляти дублювання або відсутність ПРН у навчальних планах, що значно підвищує якість підготовки акредитаційної документації.

4. Спроектовано та реалізовано інформаційну технологію у вигляді веб-застосунку (Single Page Application) на базі бібліотеки React. Розроблений «Аналізатор компетентностей» забезпечує інтерактивний інтерфейс для візуалізації зв'язків між компетентностями та результатами

навчання, що робить процес проєктування ОПП прозорим та гнучким.

5. Проведено апробацію системи на прикладі освітньої програми спеціальності 122 «Комп'ютерні науки». Результати тестування підтвердили працездатність обраної архітектури та коректність алгоритмів обробки даних.

6. Виконано економічне обґрунтування, яке показало, що впровадження розробленого програмного забезпечення дозволяє скоротити часові витрати на методичну роботу більш ніж у 10 разів. Розрахований термін окупності проєкту становить близько 1.3 року при рентабельності інвестицій (ROI) на рівні 79%, що підтверджує доцільність впровадження розробки в ТУ «МЕТІНВЕСТ ПОЛІТЕХНІКА».

7. Практична цінність роботи полягає у створенні готового до використання інструментарію, який забезпечує швидку адаптацію змісту навчання до змін у професійних стандартах та вимог роботодавців, що сприяє підвищенню конкурентоспроможності випускників на ринку праці.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кудінова О. В. Концептуальні засади вдосконалення професійної підготовки фахівців в інформаційно-освітньому середовищі вищої школи. *Науковий вісник. Серія: Педагогіка і психологія*. 2023. № 2. С. 45–52. URL: <https://vspu.net/sit/index.php/sit/article/download/5502/4953> (дата звернення: 12.12.2025).
2. Державна служба якості освіти відповідає на запитання вчителів про компетентнісний підхід у навчанні. *Державна служба якості освіти України*. URL: <https://sge.gov.ua/kompetentnisny-pidhid-navchannia/> (дата звернення: 18.12.2025).
3. Освітньо-професійна програма «Системний аналіз» першого (бакалаврського) рівня вищої освіти за спеціальністю 122 «Комп'ютерні науки» / Львівська політехніка. 2022. URL: <https://lpnu.ua/sites/default/files/2021/program/17290/opp-122-bak-oiss-2022.pdf> (дата звернення: 17.12.2025).
4. Ощипок І. М., Петришин Н. З. Інформаційні технології в нових умовах освітнього процесу і управління ним. *Grail of Science*. 2023. № 4. С. 12–20. URL: <https://archive.journal-grail.science/index.php/2710-3056/article/download/1531/1551> (дата звернення: 12.12.2025).
5. Impagliazzo R. et al. CS2023 Curricula. Computer Science Curricula 2023: Rising to the Challenges of Change in AI, Security, and Society. *ACM SIGCSE Bulletin*. 2024. Vol. 56, № 1. P. 12–15. DOI: 10.1145/3649405.3659537.
6. Kroustalli I., Xenos M. A Holistic Approach to Use Educational Robots for Supporting Computer Science Courses in Higher Education. *Computers*. 2024. Vol. 13, № 4. P. 102–118. DOI: 10.3390/computers13040102.

7. Papadopoulos P., Karacapilidis N. Shaping Programming and Data Science Education: Insights from GenAI Technical Book Trends for Curriculum Development. *IEEE Global Engineering Education Conference (EDUCON)*. 2024. P. 1–8. DOI: 10.1109/EDUCON57713.2024.10645934.
8. Al-Zoubi A. M., Al-Zoubi M. A. An ACM/IEEE and ABET Compliant Curriculum and Accreditation Management Framework. *International Journal of Advanced Computer Science and Applications*. 2020. Vol. 11, № 8. P. 120–129. DOI: 10.14569/IJACSA.2020.0110815.
9. Danyluk A., Leidig P. Variations in Data Science Curriculum: A View From Computing Education. *Harvard Data Science Review*. 2021. Vol. 3, № 2. DOI: 10.1162/99608f92.8a07f0f6.
10. Guide to the Software Engineering Body of Knowledge (SWEBOK Guide) : Version 4.0 / ed. by H. Bourque, R. E. Fairley. Los Alamitos : IEEE Computer Society, 2024. URL: <https://ieeecs-media.computer.org/media/education/swebok/swebok-v4.pdf> (дата звернення: 17.12.2025).
11. Software Engineering Body of Knowledge (SWEBOK). *Wikipedia*. URL: https://en.wikipedia.org/wiki/Software_Engineering_Body_of_Knowledge (дата звернення: 17.12.2025).
12. Software Engineering Body of Knowledge (SWEBOK). *IEEE Computer Society*. URL: <https://www.computer.org/education/bodies-of-knowledge/software-engineering> (дата звернення: 17.12.2025).
13. SFIA v9 AND SWEBOK v4. *SFIA Foundation*. URL: <https://sfia-online.org/en/tools-and-resources/bodies-of-knowledge/swebok-software-engineering-body-of-knowledge/swebok4-sfia9-the-guide-to-the-software-engineering-body-of-knowledge> (дата звернення: 17.12.2025).

14. Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). Version 4.0 / IEEE Computer Society. 2024. URL: <https://ieeecs-media.computer.org/media/education/swebok/swebok-v4.pdf> (дата звернення: 17.12.2025).
15. Course-Skill Atlas: A national longitudinal dataset of skills taught in U.S. higher education curricula / B. K. Smith et al. *Scientific Data*. 2024. Vol. 11. Art. 1012. DOI: 10.1038/s41597-024-03821-x
16. Lee J., Park K. Analyzing Computing Undergraduate Majors from Job Market Perspective: A Data-Driven Approach. *arXiv preprint*. 2024. 12 p. URL: <https://arxiv.org/abs/2412.15219> (дата звернення: 17.12.2025).
17. Sauret M., Oliveira D. Universities and the Challenge of Employability: Leveraging Labor Market Data for Relevant Lifelong Education. *Journal of University-Industry Collaboration*. 2023. Vol. 5, No. 1. P. 45–58.
18. Deming D. J. What is the Price of a Skill? The Value of Complementarity in the Labor Market. *The Quarterly Journal of Economics*. 2023. Vol. 139, No. 3. P. 1623–1673.
19. Zhang L., Wang Y. Analysis and Prediction of Computer Industry Talent Demand Based on Machine Learning. *IADIS International Conference on Information Systems*. 2022. P. 112–119. URL: <http://www.iadisportal.org/digital-library/analysis-and-prediction> (дата звернення: 17.12.2025).
20. Стандарт вищої освіти за спеціальністю 122 «Комп'ютерні науки» для першого (бакалаврського) рівня вищої освіти : затверджено наказом М-ва освіти і науки України від 10.07.2019 р. № 962. *Міністерство освіти і науки України*. URL: <https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2019/07/12/122-kompyuterni-nauki-bakalavr.pdf> (дата звернення: 17.12.2025).

21. Стандарт вищої освіти зі спеціальності 122 «Комп'ютерні науки» галузі знань 12 «Інформаційні технології» для другого (магістерського) рівня вищої освіти : затверджено наказом М-ва освіти і науки України від 28.04.2022 р. № 393. *Освіта.ua.* URL: https://osvita.ua/legislation/Vishya_osvita/86914/ (дата звернення: 17.12.2025).

22. Computing Curricula 2020: Paradigms for Global Computing Education / CC2020 Task Force. New York : Association for Computing Machinery (ACM) ; IEEE Computer Society, 2020. 203 p. URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf> (дата звернення: 17.12.2025).

23. Computing Competencies: Mapping CC2020 Dispositions to SFIA Responsibility Characteristics. URL: <https://par.nsf.gov/servlets/purl/10319441> (дата звернення: 17.12.2025).

24. CC2020: Computing Curricula 2020: Paradigms for Global Computing Education / ACM, IEEE Computer Society. URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf> (дата звернення: 17.12.2025).

25. Rathod R. Generative AI System Design Series: Part 1 – Core Architecture Components. *Medium.* URL: <https://medium.com/@rajveer.rathod1301/generative-ai-system-design-series-part-1-core-architecture-components-52d2e94e57c4> (дата звернення: 17.12.2025).

26. Generative AI: Advancing Systems Analysis & Architecture. *Coursera.* URL: <https://www.coursera.org/learn/generative-ai-advancing-systems-analysis-and-architecture> (дата звернення: 17.12.2025).

27. Generative AI Architecture Patterns. *Databricks*. URL: <https://www.databricks.com/product/machine-learning/build-generative-ai> (дата звернення: 17.12.2025).

28. Profiling First-year Students: An Interdisciplinary Challenge for Curriculum Design and Accompaniment Strategies / M. Rodriguez et al. *IEEE Global Engineering Education Conference (EDUCON)*. 2024. P. 1–6. DOI: 10.1109/EDUCON57713.2024.10854962

29. Johnson T., Miller R. Integrating High Performance Computing into Higher Education and the Pedagogy of Cluster Computing. *Proceedings of the Practice and Experience in Advanced Research Computing (PEARC '24)*. 2024. Art. 45. DOI: 10.1145/3626203.3670588

30. Kim J. Exploring the Impact of Psychosocial Learning Environments and Instructional Modalities on Academic Achievement in Blended Computer Science Education. *European Journal of Psychology of Education*. 2024. Vol. 39. P. 1567–1585. DOI: 10.1007/s10212-024-00884-9

31. Li Y., Zhang H. Massive Open Online Course Fast Adaptable Computer Engineering Education Model. *Complexity*. 2021. Vol. 2021. Art. 5934488. DOI: 10.1155/2021/5934488

32. Dubey A., Antypas K. Training the Next Generation of Computational Scientists through a New Undergraduate Course. *Computing in Science & Engineering*. 2022. Vol. 24, No. 2. P. 6–13. DOI: 10.1109/MCSE.2022.3160000

33. EduCOR: An Educational and Career-Oriented Recommendation Ontology / G. De Giacomo et al. *Proceedings of the 13th International Conference on Semantic Web Applications and Tools*. 2021. P. 1–12. URL: <https://arxiv.org/abs/2107.05522> (дата звернення: 17.12.2025).

34. Martinez S., Lopez A. Enhancing Learning Personalization in Educational Environments through Ontology-Based Knowledge Representation. *Computers*. 2023. Vol. 12, No. 10. Art. 199. DOI: 10.3390/computers12100199
35. Smith A. B., Jones C. D. Intelligent Machine for Ontological Representation of Massive Pedagogical Knowledge Based on Neural Networks. *International Journal of Electrical and Computer Engineering (IJECE)*. 2021. Vol. 11, No. 2. P. 1469–1478. DOI: 10.11591/ijece.v11i2.pp1469-1478
36. LogicHouse-v1: A digital game-based learning tool for enhanced teaching of digital electronics in higher education / O. O. Adebayo et al. *Cogent Engineering*. 2024. Vol. 11, No. 1. Art. 2322814. DOI: 10.1080/23311916.2024.2322814
37. Singh R. K. Artificial Intelligence in Higher Education: A Systematic Review of Its Impact on Student Learning and Curriculum. *International Journal of Educational Technology*. 2025. Vol. 12, No. 2. P. 23–35.
38. Sawarkar Y. Topological Sorting to Determine Course Order. *DEV Community*. URL: <https://dev.to/theyashsawarkar/topological-sorting-to-determine-course-order-16e8> (дата звернення: 17.12.2025).
39. Mastering Topological Sort: A Comprehensive Guide to Graph Algorithms. *AlgoCademy*. URL: <https://algotcademy.com/blog/mastering-topological-sort-a-comprehensive-guide-to-graph-algorithms/> (дата звернення: 17.12.2025).
40. Топологічне сортування. *Отримання знань*. URL: <https://disted.edu.vn.ua/courses/learn/7595> (дата звернення: 17.12.2025).
41. Wangyy. Course Schedule and Topological Sorting. *Medium*. URL: <https://wangyy395.medium.com/course-schedule-and-topological-sorting-7deac2802053> (дата звернення: 17.12.2025).

42. Ahluwalia P. Course Schedule II: Solving the Topological Sorting Problem with Cycle Detection. *Medium*. URL: https://medium.com/@prajwal_ahluwalia/course-schedule-ii-solving-the-topological-sorting-problem-with-cycle-detection-15de564265d6 (дата звернення: 17.12.2025).
43. Ganesan K. et al. A Game for Learning How to Model in Graph Theory. *Mathematics*. 2022. Vol. 10(12). 1969. URL: <https://www.mdpi.com/2227-7390/10/12/1969> (дата звернення: 17.12.2025).
44. Aswani A. IEOR 151 – Lecture 15: Set Covering Problem [Lecture Notes]. UC Berkeley. 2016. URL: https://aswani.ieor.berkeley.edu/teaching/FA16/151/lecture_notes/ieor151_lec1_5.pdf (дата звернення: 17.12.2025).
45. Integrating A Graph Theory In A School Math Curriculum Of Indonesia Under Realistic Mathematics Education. *International Journal of Scientific & Technology Research*. 2020. URL: <http://www.ijstr.org/final-print/jan2020/-Integrating-A-Graph-Theory-In-A-School-Math-Curriculum-Of-Indonesia-Under-Realistic-Mathematics-Education-.pdf> (дата звернення: 17.12.2025).
46. Resume Matching using Extractive Summarization and Semantic Vectors / S. Sagnika et al. *CEUR Workshop Proceedings*. 2023. Vol. 3477. URL: <https://ceur-ws.org/Vol-3477/paper2.pdf> (дата звернення: 17.12.2025).
47. Zykina A. V., Munko V. V. Algorithm for Solving the Generalized Set Cover Problem for a Special Class of Problems. *Journal of Mathematical Sciences*. 2022. Vol. 267. P. 556–565.
48. Set cover problem. *Wikipedia*. URL: https://en.wikipedia.org/wiki/Set_cover_problem (дата звернення: 17.12.2025).

49. Set covering problem. *Cornell University Computational Optimization Open Textbook*. URL: https://optimization.cbe.cornell.edu/index.php?title=Set_covering_problem (дата звернення: 17.12.2025).
50. Alon N. The Online Set Cover Problem. *Princeton Math*. URL: <https://web.math.princeton.edu/~nalon/PDFS/aaabnproc2.pdf> (дата звернення: 17.12.2025).
51. Zykina A. V., Munko V. V. Algorithm for Solving the Generalized Set Cover Problem for a Special Class of Problems. URL: https://www.researchgate.net/publication/365095787_Algorithm_for_Solving_the_Generalized_Set_Cover_Problem_for_a_Special_Class_of_Problems (дата звернення: 17.12.2025).
52. Januzaj Y., Sylqa D. A Textual Content Analysis Model for Aligning Job Market Demands and University Curricula through Data Mining Techniques. *International Journal of Interactive Mobile Technologies (iJIM)*. 2024. Vol. 18, no. 14. URL: <https://online-journals.org/index.php/i-jim/article/view/47901/> (дата звернення: 17.12.2025).
53. Литвин В. В., Висоцька В. А. Глибинне навчання : навч. посібник. Львів : Видавництво Львівської політехніки, 2021. 460 с.
54. Стрижак О. Є. Онтологічно-орієнтовані системи керування контентом інформаційно-навчальних Web-порталів : монографія. Київ : ЦП «Компринт», 2023.
55. Burov Y. An Ontology Based Modeling Framework for Design of Educational Technologies. *arXiv preprint arXiv:1802.04337*. 2018. URL: <https://arxiv.org/abs/1802.04337>.

56. Stryzhak O. Mathematical interpretation of IMRAD-based scientific studies via digital ontologies. *CEUR Workshop Proceedings*. 2024. Vol. 4060. P. 16–28. URL: <https://ceur-ws.org/Vol-4060/paper16.pdf>.

57. Lytvyn V., Vysotska V., Demchuk A. Intellectual agent construction method based on the subject field ontology. *IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*. Lviv, 2020. P. 40–46. DOI: 10.1109/CSIT49958.2020.9322032.

58. Shakhovska K. Data mining in higher education curriculum design : a case study. *Ukrainian Journal of Information Technology*. 2024. Vol. 6, no. 1.

59. Tsidylo I. et al. Ontological Approach to the Presentation of the Subject Area of the Discipline. *Proceedings of the 1st Symposium on Advances in Educational Technology*. SciTePress, 2020. URL: <https://www.scitepress.org/Papers/2020/109259/109259.pdf> (дата звернення: 17.12.2025).

60. Mathematical interpretation and digital ontologies for educational and scientific studies. *CEUR Workshop Proceedings*. URL: <https://ceur-ws.org/Vol-4060/paper16.pdf> (дата звернення: 17.12.2025).

61. Ontology-driven monitoring of students' scientific and educational activity. *ResearchGate*. URL: https://www.researchgate.net/publication/389511958_Ontology-driven_monitoring_of_students'_scientific_and_educational_activity (дата звернення: 17.12.2025).

62. Ontological Approach to Plot Analysis and Modeling. *CEUR Workshop Proceedings*. 2019. Vol. 2362. URL: <https://ceur-ws.org/Vol-2362/paper3.pdf> (дата звернення: 17.12.2025).

63. Using Ontological Modeling by Intellectualization of Learning Processes. *ResearchGate*. URL:

<https://www.researchgate.net/publication/366680811> *Using Ontological Modeling by Intellectualization of Learning Processes* (дата звернення: 17.12.2025).

64. Онтологічно-орієнтовані системи керування контентом інформаційно-навчальних Web-порталів. *ResearchGate*. URL: <https://www.researchgate.net/publication/392624339> *Ontologicno-orientovani sistemi keruvanna kontentom informacijno-navcalnih Web-portaliv* (дата звернення: 17.12.2025).

65. Harris J., Warner M. Data in Action: Using Labor-Market Alignment Data to Improve Career and Technical Education. *Institute of Education Sciences (IES)*. 2020. URL: <https://ies.ed.gov/learn/blog/data-action-using-labor-market-alignment-data-improve-career-and-technical-education> (дата звернення: 17.12.2025).

66. Leidig P., Salmela H. IS2020: A Competency Model for Undergraduate Programs in Information Systems. *Association for Computing Machinery (ACM), Association for Information Systems (AIS)*. 2021. URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is2020.pdf> (дата звернення: 17.12.2025).

67. Semerikov S. et al. Educational data mining for student-centered curriculum design: analyzing elective course selection patterns and enhancing individual educational trajectories. *Research Square*. 2025. URL: https://assets-eu.researchsquare.com/files/rs-7802166/v1_covered_ee204839-fd67-45f4-bd62-740dbfc74b79.pdf (дата звернення: 17.12.2025).

68. Martinez-Gil J. From Text to Talent: A Pipeline for Extracting Insights from Candidate Profiles. *arXiv preprint*. 2025. arXiv:2503.17438. URL: <https://arxiv.org/html/2503.17438v1> (дата звернення: 17.12.2025).

69. Pavliuk V., Drevetskyi V. Intelligent Modeling of Educational Curricula Based on Labor Market Vacancy Analysis Using Semantic and Generative AI. *Modeling, Control and Information Technologies*. URL: <https://mcitdoc.org.ua/index.php/ITConf/article/view/662> (дата звернення: 17.12.2025).

70. Zhu Z., Fouarge D., Belfi B., Somers M. Skill Matters: Curriculum updates and Labor Market Outcomes. *Skills2Capabilities Working Paper*. December 2024. URL: https://skills2capabilities.eu/files/results/papers/Working_Paper_S2C_2024_mbo_paper_Task6.2.pdf (дата звернення: 17.12.2025).

71. Educational Data Mining Techniques for Student Performance Prediction: Method Review and Comparison Analysis. *PMC*. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8688359/> (дата звернення: 17.12.2025).

72. Baker R. S., Siemens G. Educational Data Mining and Learning Analytics. In: *Cambridge Handbook of the Learning Sciences*. 2014. URL: <https://learninganalytics.upenn.edu/ryanbaker/Chapter12BakerSiemensv3.pdf> (дата звернення: 17.12.2025).

73. AI Architecture Design. *Azure Architecture Center (Microsoft Learn)*. URL: <https://learn.microsoft.com/en-us/azure/architecture/ai-ml/> (дата звернення: 17.12.2025).

74. Архітектура програмного забезпечення: Оптимізація коду, безпека та управління проектами. *Wezom Blog*. URL: <https://wezom.com.ua/ua/blog/arhitektura-programnogo-zabezpechennya> (дата звернення: 18.12.2025).

75. Melillán A., Cravero A., Sepúlveda S. Software Development and Tool Support for Curriculum Design: A Systematic Mapping Study. *Applied*

Sciences. 2023. Vol. 13(13). 7711. URL: <https://www.mdpi.com/2076-3417/13/13/7711> (дата звернення: 17.12.2025).

ДОДАТОК А

ВІДОМОСТІ РОБОТИ

Таблиця А.1 – Відомості роботи

Формат	№ п/п	Назва документу	Найменування об'єкту або вибору	Кількість сторінок
A4	1	Пояснювальна записка	КЦТПАР.КН-24-1м.00.00.00.ПЗ	93
Графічна частина				
A4	2	Діаграма потоків даних для процесу розробки та удосконалення освітньо-професійних програм (ОПП)	КЦТПАР.КН-24-1м.01.00.00.ПЛ	1
A4	3	Математичне моделювання при формуванні ОПП, структурно-логічні схеми як графи	КЦТПАР.КН-24-1м.02.00.00.ПЛ	2
A4	4	Підхід до застосування методів інтелектуального аналізу даних для підтримки прийняття рішень при формуванні ОПП	КЦТПАР.КН-24-1м.03.00.00.ПЛ	1
A4	5	Узагальнена архітектура програмного комплексу	КЦТПАР.КН-24-1м.04.00.00.ПЛ	1
A4	6	Результати проектування програмного комплексу з використанням UML-діаграм	КЦТПАР.КН-24-1м.05.00.00.ПЛ	2
A4	7	Практична реалізація програмного комплексу «Аналізатор компетентностей та результатів навчання»	КЦТПАР.КН-24-1м.06.00.00.ПЛ	1
A4	8	Екранні форми	КЦТПАР.КН-24-1м.07.00.00.ПЛ	2

ДОДАТОК Б

СКОРОЧЕНИЙ ЛІСТИНГ ПРОГРАМИ ДЛЯ ІНФОРМАЦІЙНОЇ
ПІДТРИМКИ ПРОЦЕСУ РОЗРОБКИ ВМІСТУ ДИСЦИПЛІН ОСВІТНЬО-
ПРОФЕСІЙНОЇ ПРОГРАМИ ЗА СПЕЦІАЛЬНІСТЮ «КОМП'ЮТЕРНІ НАУКИ»

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Аналізатор компетентностей</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script crossorigin
src="https://unpkg.com/react@18/umd/react.production.min.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-
dom.production.min.js"></script>
  <script
src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&
display=swap" rel="stylesheet">
  <style>
    body {
      font-family: 'Inter', sans-serif;
      background-color: #f3f4f6;
    }
    .tooltip-arrow {
      width: 0;
      height: 0;
      border-left: 5px solid transparent;
      border-right: 5px solid transparent;
      border-top: 5px solid #1f2937;
    }
  </style>
</head>
<body>
  <div id="root"></div>

  <script type="text/babel">
    const { useState, useMemo } = React;
    const { createRoot } = ReactDOM;

```

```

const SearchIcon = (props) => (
  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor" strokeWidth="2"
  strokeLinecap="round" strokeLinejoin="round"
  className={props.className}><circle cx="11" cy="11" r="8"></circle><path
  d="m21 21-4.3-4.3"></path></svg>
);
const BookOpenIcon = (props) => (
  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor" strokeWidth="2"
  strokeLinecap="round" strokeLinejoin="round"
  className={props.className}><path d="M2 3h6a4 4 0 0 1 4 4v14a3 3 0 0 0-3-
3H2z"></path><path d="M22 3h-6a4 4 0 0 0-4 4v14a3 3 0 0 1 3-
3h7z"></path></svg>
);
const TargetIcon = (props) => (
  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor" strokeWidth="2"
  strokeLinecap="round" strokeLinejoin="round"
  className={props.className}><circle cx="12" cy="12"
  r="10"></circle><circle cx="12" cy="12" r="6"></circle><circle cx="12"
  cy="12" r="2"></circle></svg>
);
const ZapIcon = (props) => (
  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
  viewBox="0 0 24 24" fill="none" stroke="currentColor" strokeWidth="2"
  strokeLinecap="round" strokeLinejoin="round"
  className={props.className}><polygon points="13 2 3 14 12 14 11 22 21 10
12 10 13 2"></polygon></svg>
);

const CompetencyAnalyzer = () => {
  const [selectedItems, setSelectedItems] = useState([]);
  const [searchMode, setSearchMode] = useState('rn');

  const matrix = {
    'PH1': ['ЗК1', 'ЗК6', 'ЗК7', . . . ]
  };

  const descriptions = {
    'ЗК1': 'Здатність до абстрактного мислення, аналізу та
синтезу.',
    'СК1': 'Здатність до математичного формулювання та
досліджування неперервних та дискретних математичних моделей,
обґрунтування вибору методів і підходів для розв\язування теоретичних і
прикладних задач у галузі комп\ютерних наук, аналізу та
інтерпретування.', . . . ,
    'PH1': 'Застосовувати знання основних форм і законів
абстрактно-логічного мислення, основ методології наукового пізнання, форм
і методів вилучення, аналізу, обробки та синтезу інформації в предметній

```

області комп'ютерних наук.', . . .

```

};

const reverseMatrix = useMemo(() => {
  const reverse = {};
  Object.entries(matrix).forEach(([rn, competencies]) => {
    competencies.forEach(comp => {
      if (!reverse[comp]) reverse[comp] = [];
      reverse[comp].push(rn);
    });
  });
  return reverse;
}, [matrix]);

const allRN = Object.keys(matrix);
const allZK = ['3K1', '3K2', '3K3', '3K4', '3K5', '3K6', '3K7',
'3K8', '3K9', '3K10', '3K11', '3K12', '3K13', '3K14', '3K15', '3K16'];
const allSK = ['CK1', 'CK2', 'CK3', 'CK4', 'CK5', 'CK6', 'CK7',
'CK8', 'CK9', 'CK10', 'CK11', 'CK12', 'CK13', 'CK14', 'CK15', 'CK16',
'CK17'];

const allCompetencies = [...allZK, ...allSK];

const results = useMemo(() => {
  if (selectedItems.length === 0) return [];

  let allResults = [];

  if (searchMode === 'rn') {
    selectedItems.forEach(rn => {
      if (matrix[rn]) {
        allResults = allResults.concat(matrix[rn]);
      }
    });
  } else {
    selectedItems.forEach(comp => {
      if (reverseMatrix[comp]) {
        allResults = allResults.concat(reverseMatrix[comp]);
      }
    });
  }

  return [...new Set(allResults)];
}, [selectedItems, searchMode, matrix, reverseMatrix]);

const handleItemToggle = (item) => {
  setSelectedItems(prev =>
    prev.includes(item)
      ? prev.filter(i => i !== item)
      : [...prev, item]
  );
};

```

```

const clearSelection = () => {
  setSelectedItems ( [] );
};

const ItemBadge = ({ item, isSelected, onClick, showRemove =
false }) => (
  <div
    className={`relative group inline-block m-1 p-2 rounded-lg
cursor-pointer transition-all duration-200 ${
      isSelected
        ? 'bg-blue-500 text-white shadow-md'
        : 'bg-gray-100 hover:bg-gray-200 text-gray-700'
      }`}
    onClick={() => onClick(item)}
    title={descriptions[item]}
  >
    <span className="text-sm font-medium">{item}</span>
    {showRemove && isSelected && (
      <span className="ml-2 text-xs opacity-75">X</span>
    )}

    <div className="absolute bottom-full left-1/2 transform -
translate-x-1/2 mb-2 px-3 py-2 bg-gray-800 text-white text-xs rounded-lg
opacity-0 group-hover:opacity-100 transition-opacity duration-300 pointer-
events-none z-10 max-w-xs whitespace-normal">
      {descriptions[item]}
      <div className="absolute top-full left-1/2 transform -
translate-x-1/2 -mt-1 border-4 border-transparent border-t-gray-
800"></div>
    </div>
  </div>
);

return (
  <div className="max-w-6xl mx-auto p-6 bg-white min-h-screen
rounded-lg shadow-lg">
    <div className="text-center mb-8">
      <h1 className="text-3xl font-bold text-gray-800 mb-2">
        Аналізатор компетентностей та результатів навчання
      </h1>
      <p className="text-gray-600">
        Спеціальність 122 Комп'ютерні науки (Бакалавр)
      </p>
    </div>

    <div className="bg-gray-50 rounded-lg p-4 mb-6 shadow-
inner">
      <div className="flex justify-center mb-4">
        <div className="bg-white rounded-lg p-1 shadow-sm flex
space-x-2">

```

```

        <button
            className={`px-4 py-2 rounded-md transition-all
duration-200 flex items-center ${
                searchMode === 'rn'
                ? 'bg-blue-500 text-white shadow-sm'
                : 'text-gray-600 hover:bg-gray-100'
            }`}
            onClick={() => { setSearchMode('rn');
clearSelection(); }}
        >
            <TargetIcon className="inline w-4 h-4 mr-2" />
            За PH → найти ЗК/СК
        </button>
        <button
            className={`px-4 py-2 rounded-md transition-all
duration-200 flex items-center ${
                searchMode === 'comp'
                ? 'bg-blue-500 text-white shadow-sm'
                : 'text-gray-600 hover:bg-gray-100'
            }`}
            onClick={() => { setSearchMode('comp');
clearSelection(); }}
        >
            <ZapIcon className="inline w-4 h-4 mr-2" />
            За ЗК/СК → найти PH
        </button>
    </div>
</div>

{selectedItems.length > 0 && (
    <div className="mb-4">
        <div className="flex items-center justify-between mb-
2">
            <h3 className="text-sm font-medium text-gray-700">
                Обрано ({selectedItems.length}):
            </h3>
            <button
                onClick={clearSelection}
                className="text-xs text-red-500 hover:text-red-
700 transition-colors"
            >
                ОЧИСТИТИ ВСЕ
            </button>
        </div>
        <div className="flex flex-wrap">
            {selectedItems.map(item => (
                <ItemBadge
                    key={item}
                    item={item}
                    isSelected={true}
                    onClick={handleItemToggle}
                    showRemove={true}

```

```

        />
      )))
    </div>
  </div>
})
</div>

<div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
  <div className="bg-white border border-gray-200 rounded-
lg p-6 shadow-sm">
    <h2 className="text-lg font-semibold text-gray-800 mb-4
flex items-center">
      <SearchIcon className="w-5 h-5 mr-2" />
      {searchMode === 'rn' ? 'Оберіть результати навчання
(PH)' : 'Оберіть компетентності (ЗК/СК)'}
    </h2>

    {searchMode === 'rn' ? (
      <div>
        <h3 className="text-sm font-medium text-gray-600
mb-3">Результати навчання:</h3>
        <div className="flex flex-wrap">
          {allRN.map(rn => (
            <ItemBadge
              key={rn}
              item={rn}
              isSelected={selectedItems.includes(rn)}
              onClick={handleItemToggle}
            />
          )))
        </div>
      </div>
    ) : (
      <div className="space-y-4">
        <div>
          <h3 className="text-sm font-medium text-gray-600
mb-3">Загальні компетентності:</h3>
          <div className="flex flex-wrap">
            {allZK.map(zk => (
              <ItemBadge
                key={zk}
                item={zk}
                isSelected={selectedItems.includes(zk)}
                onClick={handleItemToggle}
              />
            )))
          </div>
        </div>
        <div>
          <h3 className="text-sm font-medium text-gray-600
mb-3">Спеціальні компетентності:</h3>
          <div className="flex flex-wrap">

```

```

        {allSK.map(sk => (
            <ItemBadge
                key={sk}
                item={sk}
                isSelected={selectedItems.includes(sk)}
                onClick={handleItemToggle}
            />
        ))}
    </div>
</div>
)}
</div>

<div className="bg-white border border-gray-200 rounded-
lg p-6 shadow-sm">
    <h2 className="text-lg font-semibold text-gray-800 mb-4
flex items-center">
        <BookOpenIcon className="w-5 h-5 mr-2" />
        Результати пошуку ({results.length})
    </h2>

    {results.length === 0 ? (
        <div className="text-center py-8 text-gray-500">
            <SearchIcon className="w-12 h-12 mx-auto mb-3
opacity-50" />

            <p>Оберіть елементи для пошуку відповідностей</p>
        </div>
    ) : (
        <div className="space-y-4">
            {searchMode === 'rn' ? (
                <>
                    {results.filter(r => r.startsWith('ЗК')).length
> 0 && (
                        <div>
                            <h3 className="text-sm font-medium text-
green-600 mb-3">
                                Загальні компетентності
                                ({results.filter(r => r.startsWith('ЗК')).length)}):
                            </h3>
                            <div className="flex flex-wrap">
                                {results.filter(r =>
r.startsWith('ЗК')).map(comp => (
                                    <ItemBadge
                                        key={comp}
                                        item={comp}
                                        isSelected={false}
                                        onClick={() => {}}
                                    />
                                ))}
                            </div>
                        </div>
                    )}
                </div>
            )}
        </div>
    )}

```

```

    ))
    {results.filter(r => r.startsWith('CK')).length
> 0 && (
      <div>
        <h3 className="text-sm font-medium text-
green-600 mb-3">
          Спеціальні компетентності
({results.filter(r => r.startsWith('CK')).length}):
        </h3>
        <div className="flex flex-wrap">
          {results.filter(r =>
r.startsWith('CK')).map(comp => (
            <ItemBadge
              key={comp}
              item={comp}
              isSelected={false}
              onClick={() => {}}
            />
          )}}
        </div>
      </div>
    ) : (
      <div>
        <h3 className="text-sm font-medium text-green-
600 mb-3">Результати навчання:</h3>
        <div className="flex flex-wrap">
          {results.map(rn => (
            <ItemBadge
              key={rn}
              item={rn}
              isSelected={false}
              onClick={() => {}}
            />
          )}}
        </div>
      </div>
    )}
  </div>
)}
</div>
</div>
<div className="mt-8 text-center text-sm text-gray-500">
  На основі Стандарту вищої освіти першого (бакалаврського)
рівня, спеціальність 122 Комп'ютерні науки
</div>
</div>
);
};

```

```
    const container = document.getElementById('root');  
    const root = createRoot(container);  
    root.render(<CompetencyAnalyzer />);  
  
  </script>  
</body>  
</html>
```

ДОДАТОК В
ПЕРЕЛІК ТА ВІДБИТКИ НАУКОВИХ ПУБЛІКАЦІЙ ЗА ТЕМОЮ
ДОСЛІДЖЕННЯ

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»
ТОВ «МЕТІНВЕСТ ПОЛІТЕХНІКА»
Ройтлінгенський університет техніки та економіки (Німеччина)
Еслінгенський університет прикладних наук (Німеччина)
Технічний університет Фрайберзька гірничо академія (Німеччина)
Університет Кобленц-Ландау (Німеччина)
Вроцлавський технічний університет (Польща)
Краківська гірничо-металургійна академія (Польща)

**ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
В ОСВІТІ, НАУЦІ ТА ПРОМИСЛОВОСТІ**

XX МІЖНАРОДНА КОНФЕРЕНЦІЯ

5 грудня 2025 року
м. Дніпро

Збірник наукових праць
№ 10

Дніпро
Журфонд
2025

УДК 622 (06)
П78

Редакційна колегія:

О.О. Азюковський, Г.Г. Півняк, О.С. Поважний, О.С. Бешта, В.А. Бойко, А.В. Голоядов, С.В. Детюк, О.В. Бондар, Є.С. Мостіпан, І.М. Луценко, І.М. Удовик, Є.В. Кошеленко, М.О. Алексеєв, В.І. Корнієнко, В.В. Гнатушенко, О.О. Сдвижкова, Т.А. Желдак, О.О. Бешта, А.В. Бубліков, Ю.А. Папаїка, В.О. Пінчук, С.С. Худолій, Л.І. Мещеряков, А.І. Купін, М.І. Горбійчук, О.А. Дмитрієва, А. Дерен, Н. Нойбергер.

П78 **Проблеми** використання інформаційних технологій в освіті, науці та промисловості: ХХ міжнар. конф. (5 грудня 2025 р., м. Дніпро): зб. наук. пр. / ред. кол.: О.О. Азюковський та ін.; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2025. – № 10. – 594 с.

ISBN 978-617-8737-34-4

Подано результати теоретичних та експериментальних досліджень з різних аспектів використання інформаційних технологій та енергетики в освіті, науці та керуванні промисловістю. У публікаціях розглянуто питання створення та вдосконалення інформаційних систем та технологій, системного аналізу та наук про дані, кібербезпеки та захисту інформації, автоматизації та комп'ютерно-інтегрованих технологій, цифрової трансформації енергетики, мехатроніки та робототехніки і електромобільності.

Для наукових, інженерно-технічних співробітників і студентів, які спеціалізуються в галузі інформаційних технологій.

УДК 622 (06)

ISBN 978-617-8737-34-4

© НТУ «Дніпровська політехніка», 2025
© Журфонд, 2025

8. Xu L. Compressive strength of cold formed steel C-shape columns with slotted holes / L. Xu, Y. Shi, S. Yang // 22nd International specialty conference on cold-formed steel design and construction, St Louis, MO, 5–6 November 2014, Rolla, MO: Missouri University of Science and Technology. – 2014. – P. 157–170.
9. Young B. Cold-formed steel lipped channel columns influenced by local-distortional interaction: strength and DSM design / B. Young, N. Silvestre, D. Camotim // Journal of Structural Engineering. – 2013. - №139(6). – P. 1059-1074.
10. Yuan X.L. Simplified seismic design for midrise buildings with vertical combination of cold-formed steel and concrete framing / X.L. Yuan, L. Xu // 22nd International specialty conference on cold-formed steel design and construction, St Louis, MO, 5–6 November 2014, Rolla, MO: Missouri University of Science and Technology. – 2014. – P. 617–632.
11. Григорій Сучков, Руслан Мигущенко, Сергій Плеснецов, Андрій Донченко, Юрій Кошкарів, Вадим Тимофєєв. Дослідження та розробка комбінованого методу зменшення величини неконтрольованого приповерхневого шару при ультразвуковому електромагнітно-акустичному контролі феромагнітних металовиробів. Український метрологічний журнал. №3 2025. С. 16 - 21

УДК 004.89:378.147

П.І. Сагайда¹, І.О. Ляшенко¹

¹ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА», Запоріжжя, Україна

ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ФОРМУВАННЯ ЗМІСТУ ДИСЦИПЛІН ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ У ТЕХНІЧНОМУ УНІВЕРСИТЕТІ

Анотація. У тезах представлено результати дослідження, метою якого є розробка концептуальної моделі та функціональних вимог до інформаційної системи (ІС) для підтримки формування та оптимізації змісту навчальних дисциплін освітньо-професійної програми (ОПП) «Комп'ютерні науки» в технічному університеті. Запропонований підхід інтегрує компетентнісну модель (на основі результатів навчання та таксономії Блума) з експертними методами (Дельфі, Аналіз ієрархій – АНР). Реалізований прототип ІС дозволить забезпечити узгодження змісту програми з міжнародними стандартами (АСМ/IEEE, SFIA), аналіз покриття компетенцій і автоматизацію процесу розробки навчального плану.

Ключові слова: *освітньо-професійна програма (ОПП), компетентнісний підхід, зміст дисциплін, Аналіз ієрархій (АНР), метод Дельфі, матриця компетенцій, інформаційна система, АСМ/IEEE, SFIA.*

Вступ. Сучасні вимоги ІТ-індустрії та перехід до компетентнісно-орієнтованих освітніх моделей вимагають системного та формалізованого підходу до проектування змісту навчальних дисциплін [1]. Формування змісту дисциплін має бути узгоджене з очікуваннями роботодавців та міжнародними рамками, такими як АСМ/IEEE Computing Curricula [2,3] та SFIA (Skills Framework for the Information Age) [4], що підвищує конкурентоспроможність випускників. Актуальність дослідження обумовлена необхідністю розробки

цифрових інструментів та методів, які б забезпечили автоматизований аналіз та корекцію навчальних програм за критеріями ефективності та прозорості.

Постановка задачі. Основна задача полягає в розробці концептуальної моделі та багаторівневої архітектури програмної системи, здатної підтримувати процес формування змісту дисциплін ОПП «Комп'ютерні науки». Це включає:

- Формалізацію теоретичних основ контентного моделювання (результати навчання, компетентності, таксономія Блума [3]).
- Створення математичних моделей для оптимального розподілу навчального часу, побудови логічної послідовності вивчення (передумови) та забезпечення покриття необхідних компетенцій.
- Застосування методів АНР, Дельфі та кластеризації для структуризації та ранжування навчальних компонентів за значущістю.

Основний зміст роботи

1. Концептуальна модель та архітектура системи

Концептуальна модель ОПП базується на компетентнісній матриці, яка фіксує, які дисципліни і на якому рівні (введення, формування, поглиблене опанування) формують конкретні цільові компетенції.

Багаторівнева архітектура ІС включає:

- презентаційний рівень (веб-інтерфейс) для вводу даних експертами та візуалізації результатів;
- логічний рівень (сервіси та алгоритми): модулі оптимізації навчального плану, обробки експертних оцінок (Delphi, АНР), кластеризації;
- рівень даних: сховище інформації (курси, компетенції, матриці відповідностей, експертні оцінки), що може бути реляційною (PostgreSQL) або графовою (Neo4j) СУБД.

2. Математичні та експертні методи

Для формування та оптимізації змісту ОПП застосовані моделі та методи, які представлено у табл. 1.

3. Експериментальні результати та DSS

Прототип ІС був протестований на ОПП «Комп'ютерні науки». Оптимізаційний модуль рекомендував перерозподіл годин, щоб покрити недостатньо представлені компетентності (наприклад, пов'язані з інтелектуальним аналізом даних). Статистичний аналіз (t-тест з $p < 0.05$) підтвердив, що оптимізовані пропозиції стійко покращують сумарний бал покриття компетенцій.

Система містить модуль підтримки прийняття рішень, який надає рекомендації за моделлю «дерева рішень». Наприклад, при виявленні розривів у матриці компетенцій система пропонує гілки рішень: «додати курс X» або «змінити зміст курсу Y». Для оцінки якості ОПП використовуються KPI та радарна діаграма, що візуалізує метрики «сильних» і «слабких» сторін програми.

Таблиця 1

Моделі та методи для формування та оптимізації змісту ОПП

Модель/Метод	Призначення
Послідовність вивчення	Графові моделі (орієнтовані ациклічні графи) для топологічного впорядкування курсів з урахуванням передумов та пріоритетів.
Компетентнісне покриття	Матриця M_{ij} (дисципліни-компетентності) для перевірки, чи забезпечують заплановані курси досягнення всіх цільових компетентностей на необхідному рівні.
Метод АНР	Ранжування дисциплін та критеріїв за допомогою попарних порівнянь експертами для отримання узгоджених пріоритетів.
Метод Дельфі	Ітеративне анонімне опитування експертів для досягнення консенсусу щодо невідомих параметрів (наприклад, майбутніх вимог до компетенцій).
Кластеризація	Групування дисциплін за схожими характеристиками для виявлення "комплексів" курсів та оптимізації структури програми.

Наукова новизна. Запропоновано багаторівневий підхід, що інтегрує компетентнісну модель з експертними та оптимізаційними методами. Вперше синтезовано міжнародні стандарти ACM/IEEE та SFIA з адаптивними алгоритмами формування змісту, реалізовано це у програмному рішенні із завершеною аналітичною моделлю. Система також впроваджує механізм безперервного вдосконалення (PDCA-цикли), що дозволяє адаптувати ОПП на основі моніторингу KPI та змін ринкових вимог.

Висновки. Розроблена концептуальна модель та інформаційна система ефективно поєднують математичні та експертні методи для формування та оптимізації змісту дисциплін ОПП «Комп'ютерні науки». Система буде здатна:

- автоматизувати значну частину процесу планування, підвищуючи його прозорість та ефективність;
- підвищити якість освіти, зробивши її більш орієнтованою на потреби ринку праці та узгодженою з міжнародними стандартами;
- надати інструменти прийняття рішень для виявлення прогалів у компетентностях та пропозиції корекційних дій.

У подальших дослідженнях планується масштабування моделі на інші технічні спеціальності та розширення за рахунок модулів машинного навчання для прогнозування успішності змін та персоналізації освіти.

ПЕРЕЛІК ПОСИЛАНЬ

1. Preecha Tangworakitthaworn P., Lester Gilbert L., and Gary B. Wills G.B.. ILO Diagram: A Conceptual Model for Curriculum Development. *IEEE Technology and Engineering Education (ITEE)*. 2013. Vol. 8, No. 3. P. 12-19.
2. Computing Curricula 2020. Paradigms for Global Computing Education. URL : <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf>
3. Computer Science Curricula 2023 / Kumar A.N. et al. Association for Computing Machinery : New York, 2024. 459 p. DOI : <https://doi.org/10.1145/3664191>
4. SFIA - the global skills and competency framework for the digital world. URL : <https://sfia-online.org/en>.