

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(назва факультету)

Кафедра систем штучного інтелекту та аналізу даних

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістра

(освітній ступінь)

на тему: Технологія підтримки прийняття рішень при формуванні команди для виконання ІТ-проектів

Виконала: студентка 6 курсу, групи САмд-61

спеціальності 124 «Системний аналіз»

(шифр і назва спеціальності)

	Кіт М.Є. (підпис) (прізвище та ініціали)
Керівник	Яцишин В.В. (підпис) (прізвище та ініціали)
Нормоконтроль	Гладько С.В. (підпис) (прізвище та ініціали)
Завідувач кафедри	Яцишин В.В. (підпис) (прізвище та ініціали)
Рецензент	Никитюк В.В. (підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра систем штучного інтелекту та аналізу даних

ЗАТВЕРДЖУЮ

Завідувач кафедри Яцишин В.В.

« 07 » листопада 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 124 «Системний аналіз»
(шифр і назва спеціальності)

студенту Kim Марії Євгенівні
(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) Технологія підтримки прийняття рішень при формуванні команди для виконання ІТ-проектів

Керівник проекту (роботи) Яцишин Василь Володимирович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» листопада 2025 року №4/7-951

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Фактори впливу на найм ІТ-спеціалістів, методи колаборативної фільтрації, методи препроцесингу даних, методи аналізу в умовах невизначеності

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз сучасних підходів та факторів впливу при формуванні команд для виконання ІТ-проектів. 2. Розроблення технології підтримки прийняття рішень щодо формування команд для виконання ІТ-проектів. 3. Реалізація та апробація програмного засобу підтримки прийняття рішень при формуванні команд для виконання ІТ-проектів 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Актуальність і мета дослідження. 2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Фактори впливу на підбір команди. 4. Методи колаборативної фільтрації 4. Модель підбору команди. 5. Метод підбору команди 6. Вимоги до системи підтримки прийняття рішень. 7. Архітектура системи прийняття рішень. 8. Результати експерименту. 9. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>Окіпний І.Б., зав. каф. МТ</i>		
	<i>Стручок В.С., ст. викладач каф. ОХ</i>		

7. Дата видачі завдання 10.11.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Огляд літературних джерел. Постановка завдання на кваліфікаційну роботу.</i>	<i>10.11.2025р. – 18.11.2025р.</i>	<i>виконано</i>
2.	<i>Розроблення технології підтримки прийняття рішень щодо формування команд для виконання ІТ-проєктів</i>	<i>29.11.2025р. – 02.12.2025р.</i>	<i>виконано</i>
3.	<i>Реалізація та апробація програмного засобу підтримки прийняття рішень при формуванні команд для виконання ІТ-проєктів</i>	<i>03.12.2025р. – 10.12.2025р.</i>	<i>виконано</i>
4.	<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>04.12.2025р. – 10.12.2025р.</i>	<i>виконано</i>
5.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>11.12.2025р. – 19.12.2025р.</i>	<i>виконано</i>
6.	<i>Попередній захист кваліфікаційної роботи магістра</i>	<i>16.12.2025р.</i>	<i>виконано</i>
7.	<i>Захист кваліфікаційної роботи магістра</i>	<i>23.12.2025</i>	

Студент

(підпис)

Kim M.C.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Яцишин В.В.

(прізвище та ініціали)

АНОТАЦІЯ

Кіт М.Є. Технологія підтримки прийняття рішень при формуванні команди для виконання ІТ-проектів: робота на здобуття кваліфікаційного ступеня магістра: спец. 124 — системний аналіз / наук. кер. В.В. Яцишин. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: технологія, прийняття рішення, ІТ, проєкт, команда.

У кваліфікаційній роботі проведено аналітичний огляд сучасних підходів до підтримки прийняття рішень у задачах формування команд для виконання ІТ-проектів, проаналізовано існуючі методи підбору персоналу, рекомендаційні системи та моделі колаборативної фільтрації, а також досліджено фактори, що впливають на ефективність командної роботи в проєктно-орієнтованих ІТ-середовищах.

Запропоновано та обґрунтовано модель представлення об'єктів предметної області, яка поєднує характеристики розробників, вимоги ІТ-проектів і організаційні обмеження, що дало змогу формалізувати задачу підбору команди у вигляді задачі рекомендаційного аналізу. Розроблено метод формування рекомендацій на основі метрик подібності та методів колаборативної фільтрації з використанням регуляризаторів, який враховує технологічний стек, професійний досвід, освітні характеристики та просторові фактори, що підвищує точність і стійкість результатів прийняття рішень.

Реалізовано програмний засіб автоматизованого підбору команди розробників з багаторівневою архітектурою, який включає рівень представлення, рівень бізнес-логіки та рівень даних. Запропоновано структуру бази даних і алгоритми обробки інформації для формування ранжованих списків кандидатів.

ABSTRACT

Kit M.Ye. Decision Support Technology for IT Project Team Formation. Master's Graduation Thesis: speciality 124 – System Analysis/supervisor V.V. Yatsyshyn. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: technology, decision making, project, team.

In the qualification thesis, an analytical review of modern approaches to decision support in team formation for IT project execution is conducted. Existing personnel selection methods, recommender systems, and collaborative filtering models are analyzed, and the factors influencing the effectiveness of teamwork in project-oriented IT environments are investigated.

A domain object representation model is proposed and substantiated, integrating developer characteristics, IT project requirements, and organizational constraints, which made it possible to formalize the team selection problem as a recommender analysis task. A recommendation generation method based on similarity metrics and collaborative filtering with regularization is developed; it takes into account the technology stack, professional experience, educational background, and spatial factors, thereby improving the accuracy and robustness of decision-making outcomes.

A software tool for automated developer team selection with a multi-layer architecture is implemented, including presentation, business logic, and data layers. A database structure and information processing algorithms for generating ranked lists of candidates are proposed.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ТА ФАКТОРІВ ВПЛИВУ ПРИ ФОРМУВАННІ КОМАНД ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ	14
1.1. Системний підхід до формування команд.....	14
1.2. Моделі компетентностей і методи оцінювання учасників ІТ-проєкту	18
1.3. Аналіз вимог і пропозицій ІТ- ринку в контексті підготовки фахівців навчальними закладами	21
1.4. Висновки до розділу	29
РОЗДІЛ 2 РОЗРОБЛЕННЯ ТЕХНОЛОГІЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ФОРМУВАННЯ КОМАНД ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ	30
2.1. Моделювання процесу формування команд на основі методів колаборативної фільтрації.....	30
2.2. Метрики подібності в моделях колаборативної фільтрації	33
2.2.1. Метрика подібності Жаккара.....	35
2.2.2. Метрика на основі обчислення косинусу кута між векторами	36
2.2.3. Метрика на основі коефіцієнта кореляції Пірсона	37
2.3. Модель підтримки прийняття рішень щодо формування команди для виконання ІТ-проєктів	38
2.3.1. Формалізована модель опису об'єктів у задачі формування команди для виконання ІТ-проєктів	41
2.3.2. Метод багатокритеріального підбору команди розробників із застосуванням метрик подібності.....	44
2.4. Метод підвищення стійкості колаборативної фільтрації на основі регуляризованих метрик подібності.....	47
2.5. Висновки до розділу	52

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАСОБУ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ ФОРМУВАННІ КОМАНД ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ.....	53
3.1. Визначення функціональних вимог до засобу автоматизованого підбору команди розробників комп'ютерних систем.....	53
3.2. Моделювання та реалізація бази даних системи підтримки прийняття рішень щодо підбору команди розробників.....	56
3.3. Побудова моделі архітектури програмного засобу підтримки прийняття рішень	70
3.4. Результати апробації запропонованого методу і засобу	73
3.5. Висновки до розділу	77
РОЗДІЛ 4 Охорона праці та Безпека в надзвичайних ситуаціях	78
4.1. Охорона праці.....	78
4.2. Розроблення та впровадження режимів радіаційного захисту працюючих і службовців у виробничій діяльності об'єкту в умовах радіоактивного забруднення місцевості.....	80
ВИСНОВКИ	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	87
Додаток А Тези конференцій	91

**ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ,
СИМВОЛІВ І СКОРОЧЕНЬ**

БД	База Даних
ЖЦ	Життєвий Цикл
ПС	Програмні Системи
CASE	Computer Aided Software Engineering
SQL	Structured Query Language
UML	Unified Modeling Language
XML	Extended Markup Language

ВСТУП

Актуальність теми. Сучасний розвиток ІТ-індустрії супроводжується швидким зростанням складності проєктів, багатоваріантністю технологічних рішень і підвищеними вимогами до ефективності командної взаємодії. У таких умовах формування команди для виконання ІТ-проєкту перестає бути виключно кадровим завданням і перетворюється на багатокритеріальну задачу підтримки прийняття рішень. Від правильного найму фахівців залежать строки розроблення програмного забезпечення, якість результату, адаптивність до змін і загальна успішність проєкту.

Незважаючи на широку представленість інструментів управління людськими ресурсами, значна частина рішень приймається інтуїтивно або на основі досвіду менеджера, що підвищує ризики формування неефективних команд. Системний аналіз, як наука і практика, пропонує методи формалізації процесів, побудови моделей і застосування математичних методів для прийняття раціональних рішень у складних, багатofакторних умовах. Застосування цих підходів у задачі формування команд ІТ-проєктів дозволяє перейти від суб'єктивних оцінок до структурованого, відтворюваного та обґрунтованого вибору.

Вагомим фактором є також зростання доступності даних про компетентності розробників, їхній професійний досвід, попередні результати, технологічні вподобання та стилі роботи. Це створює передумови для використання інтелектуальних методів аналізу даних і побудови моделей, здатних прогнозувати потенціал окремих фахівців та синергію команди. Водночас на практиці бракує комплексних технологій, що поєднують багатокритеріальний аналіз, методи оптимізації та рекомендаційні моделі у єдину систему підтримки прийняття рішень.

У науковій літературі розглядаються різні аспекти автоматизації рекрутингу та оцінювання розробників, зокрема моделювання компетентностей, аналіз впливу освіти та досвіду на професійну успішність, застосування

колаборативної фільтрації для виявлення подібності між вимогами проєктів і профілями фахівців. Однак ці підходи переважно орієнтовані на вибір окремого розробника і рідко охоплюють задачу формування збалансованої команди, що враховує компетентнісний склад, ролі, внутрішню сумісність, ризики та специфіку проєкту.

Таким чином, існує науково-практична проблема щодо створення технології підтримки прийняття рішень, яка б дозволила менеджеру системно формувати команду для ІТ-проєкту, спираючись на аналіз компетентностей, моделей сумісності, багатокритеріальну оцінку та використання інтелектуальних методів обробки даних.

Мета роботи полягає у розробці технології підтримки прийняття рішень для формування команди ІТ-проєкту, яка поєднує методи системного аналізу, багатокритеріального вибору та інтелектуальних алгоритмів оцінювання кандидатів.

Для досягнення вказаної мети, в роботі поставлено та розв'язано наступні **задачі**:

- проаналізувати сучасні підходи до формування команд та методи підтримки прийняття рішень у контексті управління ІТ-проєктами;
- сформувати систему критеріїв оцінювання кандидатів, включно з компетентностями, досвідом, ролями та сумісністю;
- побудувати математичну модель задачі формування команди як багатокритеріальної задачі оптимізації;
- обґрунтувати та адаптувати методи аналізу даних і інтелектуальних алгоритмів для підвищення точності оцінювання кандидатів;
- розробити технологію підтримки прийняття рішень та алгоритм найму команди;
- реалізувати програмний прототип системи та провести експериментальну оцінку її ефективності.

Об'єкт дослідження: процес формування команди для виконання ІТ-проектів.

Предмет дослідження: моделі, методи та алгоритми підтримки прийняття рішень при доборі фахівців і формуванні збалансованої команди розробників.

Методи дослідження: Для вирішення поставлених у роботі задач використано такі методи дослідження

- аналіз і узагальнення – при дослідженні сучасних підходів до формування команд для виконання ІТ-проектів, аналізі вимог ІТ-ринку, освітніх факторів та існуючих рекомендаційних моделей;
- методи системного аналізу та формалізації – при побудові формалізованої моделі предметної області, визначенні сутностей, атрибутів і критеріїв оцінювання відповідності розробників вимогам ІТ-проектів, а також при розробці методу підтримки прийняття рішень;
- методи інтелектуального аналізу даних і машинного навчання – при застосуванні колаборативної фільтрації, метрик подібності та регуляризації для формування рекомендацій щодо підбору складу команди;
- методи проєктування програмних систем та програмування – при розробці архітектури програмного засобу, проєктуванні реляційної бази даних і реалізації функціональних модулів системи;
- експериментальні методи та методи вимірювання – при апробації запропонованого методу, аналізі результатів рекомендацій і оцінюванні ефективності розробленого програмного засобу.

Наукова новизна отриманих результатів. Наукова новизна роботи полягає у вирішенні науково-практичної задачі підтримки прийняття рішень при формуванні команд для виконання ІТ-проектів, у межах якої отримано такі нові результати:

- уперше запропоновано та формалізовано модель представлення об'єктів предметної області задачі формування команди ІТ-проекту, яка інтегрує характеристики розробників, вимоги проєктів і контекстні фактори (освітні, географічні та досвідні), що дало змогу системно описати процес підбору

команди та забезпечити можливість його автоматизованої реалізації в системі підтримки прийняття рішень.

– уперше розроблено метод підбору команди для виконання ІТ-проектів на основі метрик подібності та колаборативної фільтрації, який, на відміну від існуючих підходів, враховує не лише технологічні компетентності розробників, а й рейтинг закладів освіти, досвід роботи та просторову близькість, що дало змогу підвищити точність формування ранжованих списків кандидатів і обґрунтованість управлінських рішень.

– удосконалено метод колаборативної фільтрації шляхом використання регуляризаторів і нормалізації оцінок, що дало змогу зменшити вплив розрідженості даних і суб'єктивності оцінювання, підвищити стійкість моделі та забезпечити прийнятну точність рекомендацій при автоматизованому формуванні команд ІТ-проектів.

Практичне значення одержаних результатів. Розроблена технологія може бути використана компаніями для формування команд розробників, оптимізації кадрових рішень, автоматизації процесу аналізу компетентностей та ризиків, підвищення продуктивності проектів та зменшення витрат, пов'язаних із неефективним підбором персоналу.

Публікації. Результати кваліфікаційної роботи апробовані на XIII науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (17-18 грудня 2025 року) як тези конференцій.

1. Яцишин В.В., Кіт Є.В. Системний підхід у формуванні команди для виконання ІТ-проектів. Матеріали XIII науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (17-18 грудня 2025 року). Тернопіль: ТНТУ. 2025. С. 217.

2. Яцишин В.В., Кіт Є.В. Фактори впливу на ефективність виконання ІТ-проектів. Матеріали XIII науково-технічної конференції Тернопільського

національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (17-18 грудня 2025 року). Тернопіль: ТНТУ. 2025. С. 218.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 4 розділів, висновків, списку використаної літератури та додатківавв.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ТА ФАКТОРІВ ВПЛИВУ ПРИ ФОРМУВАННІ КОМАНД ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ

1.1. Системний підхід до формування команд

Формування команди для виконання ІТ-проєкту є складною багатофакторною задачею, у якій поєднуються технічні, організаційні та людські аспекти. На якість роботи команди впливають компетентності фахівців, їхній досвід, ролі у проєктній структурі, здатність до взаємодії, психологічні особливості, а також фактори зовнішнього середовища: специфіка проєкту, обмеження бюджету, часові рамки та вимоги замовника. Така багатовимірність робить процес формування команди типовою проблемою системного аналізу, де необхідно не лише оцінити окремі характеристики кандидатів, але й визначити інтегральну ефективність їхньої взаємодії [2-5].

У традиційних підходах вибір складу команди здебільшого базується на досвіді менеджера, експертних оцінках та попередніх результатах роботи фахівців. Проте суб'єктивність таких рішень призводить до значної варіативності результатів та підвищує ризики формування неефективних груп. Системний підхід дозволяє розглядати команду як цілісну структуру, у якій взаємодія між елементами (учасниками команди) має не менше значення, ніж властивості кожного елемента окремо. Відповідно, критерії оцінювання повинні охоплювати не тільки професійний рівень розробників, а й параметри командної динаміки, конфліктність, комунікаційні особливості, стиль роботи та потенційний вплив учасників один на одного. У табл. 1.1 представлено порівняння традиційного і системного підходів при формуванні команди для виконання ІТ-проєктів [4].

Різні дослідження в галузі управління ІТ-проєктами показують, що успіх команди визначається поєднанням професійних компетентностей (hard skills) та

«м'яких» навичок (soft skills), що формуються у процесі навчання, професійної діяльності та набуття практичного досвіду.

Таблиця 1.1 – Порівняння підходів до формування команди

Критерій	Традиційний підхід	Системний підхід
Основа рішення	Досвід менеджера	Дані + моделі DSS
Оцінювання кандидатів	Окремо	У системі взаємодій
Помилки	Висока ймовірність	Виявляються автоматично
Формалізація	Низька	Висока
Відтворюваність	Відсутня	Забезпечена
Прогнозування результату	Майже неможливе	Присутнє (ML, аналіз сумісності)

Формування команди повинно враховувати як технічні компетентності, зокрема, володіння мовами програмування, технологіями розроблення, інструментами DevOps, так і фактори психологічної та комунікативної сумісності [1]. У табл. 1.2 наведено фактори впливу на ефективність команди при виконанні IT-проєктів.

Таблиця 1.2 – Фактори впливу на ефективність команди

Група факторів	Приклади	Вплив на команду
Технічні компетентності (Hard Skills)	Мови програмування, фреймворки, DevOps, архітектура	Визначають здатність виконувати технічні задачі
М'які навички (Soft Skills)	Комунікація, емоційний інтелект, лідерство	Формують командну динаміку та атмосферу
Проєктні ролі	Developer, Architect, QA, PM, BA, DevOps	Забезпечують розподіл відповідальності

Продовження таблиці 1.2

Група факторів	Приклади	Вплив на команду
Психологічна сумісність	Темперамент, стиль роботи, конфліктність	Впливає на стабільність та продуктивність
Організаційні обмеження	Бюджет, терміни, віддаленість	Формують рамки прийняття рішень

У сучасних ІТ-компаніях активно розвиваються моделі компетентностей, що структурують вимоги до фахівців різних ролей. Зокрема, поширеними є T-shaped та π -shaped моделі, що відображають необхідність поєднання глибоких спеціалізованих знань із широким розумінням суміжних сфер [6]. До того ж стандарти, такі як SFIA (Skills Framework for the Information Age), надають формальні описи професійних ролей і компетентностей, що дозволяє більш об'єктивно оцінювати кандидатів під час формування команди. У табл. 1.3 проведено порівняння моделей компетентностей учасників команди при виконанні ІТ-проектів.

Таблиця 1.3 – Порівняння моделей компетентностей учасників команди

Модель	Характеристика	Застосування при формуванні команд
T-shaped	Одна глибока компетентність + широкі знання	формування універсальних членів команди
π -shaped	Дві сильні спеціалізації	підвищення гнучкості ролей
SFIA	формальний опис навичок і рівнів	стандартизоване оцінювання фахівців
Dreyfus	розвиток компетентностей у часі	прогнозування потенціалу кандидата

Системний підхід також передбачає декомпозицію задачі формування команди на підзадачі, серед яких: визначення ролей у проекті, формування вимог

до кожної ролі, зіставлення профілів кандидатів із вимогами, оцінювання сумісності між кандидатами, моделювання ризиків та прогнозування потенційної продуктивності команди. Кожна з цих підзадач може бути формалізована та описана математично, що відкриває можливість використання методів підтримки прийняття рішень, аналізу даних і оптимізації.

Застосування системного підходу також дозволяє враховувати динамічний характер ІТ-проектів.

Команди не є статичними структурами: вони змінюються в залежності від етапу життєвого циклу проекту, появи нових вимог або ризиків. Тому моделі формування команд повинні бути адаптивними та враховувати можливість перебудови ролей та функцій у процесі виконання завдань[7-9].

Особливе значення у сучасних умовах відіграють дані про результати роботи фахівців, їхню історію участі у проектах, прохідність завдань, складність реалізованих модулів, а також взаємодію з іншими членами команди. Аналіз таких даних дозволяє виявляти закономірності, прогнозувати сумісність та ефективність співпраці. Саме це стало передумовою для активного впровадження рекомендаційних систем та алгоритмів машинного навчання у сферу управління персоналом.

Отже, системний підхід до формування команди поєднує:

- оцінку компетентностей і досвіду;
- аналіз міжособистісної та професійної сумісності;
- моделювання структур і ролей команди;
- використання об'єктивних даних;
- застосування методів підтримки прийняття рішень та методів аналізу даних.

Це створює основу для побудови технології підтримки прийняття рішень, яка дозволяє обґрунтовано, ефективно та відтворювано формувати команди для ІТ-проектів.

1.2. Моделі компетентностей і методи оцінювання учасників ІТ-проєкту

У сучасних ІТ-проєктах ефективність команди визначається не лише наявністю технічних навичок у розробників, але й збалансованим розподілом ролей, рівнем комунікаційної взаємодії та здатністю фахівців працювати в умовах невизначеності. Це зумовлює необхідність формалізації компетентностей та створення моделей, які дозволяють об'єктивно оцінювати кандидатів. Компетентнісний підхід у формуванні команди забезпечує чітке структурування навичок учасників, що дає змогу будувати алгоритми підтримки прийняття рішень для порівняння кандидатів, агрегування показників та прогнозування ефективності командної роботи. У табл. 1.4 показано ключові компетентності учасників команди та використання моделей при підтримці прийняття рішень.

Таблиця 1.4 – Порівняння ключових моделей компетентностей

Модель	Сутність	Переваги	Недоліки	Використання у DSS
T-shaped	Широкі загальні знання + глибока спеціалізація	Гнучкість у ролях, універсальність	Не відображає поведінкових аспектів	Порівняння кандидатів за шириною та глибиною навичок
π-shaped	Дві основні глибокі експертизи + широка база знань	Підтримка складних міждисциплінарних проєктів	Важко знайти таких фахівців	Аналіз потенціалу для ролей, які вимагають мультифункціональності

Продовження таблиці 1.4

Модель	Сутність	Переваги	Недоліки	Використання у DSS
SFIA	Стандартизований опис навичок з рівнями 1–7	Універсальність, формальність	Потребує адаптації під конкретний проєкт	Створення профілів компетентностей для DSS
Dreyfus Model	Поступовий розвиток компетентності	Прогнозування потенціалу	Суб'єктивність визначення рівнів	Моделювання траєкторії розвитку фахівця

Системи підтримки прийняття рішень нині оперують не окремими навичками кандидата, а комплексним профілем компетентностей, що включає:

- Професійні навички – технології, мови, інструменти розроблення, архітектура;
- Soft Skills – комунікація, адаптивність, лідерство, командність;
- Domain Knowledge – предметна область проєкту;
- Experience Metrics – роки досвіду, складність виконаних завдань;
- Team-Fit Indicators – схильність до співпраці, стиль роботи, поведінкові характеристики.

На основі компетентнісного профілю системи прийняття рішень здійснює порівняння, класифікацію та прогнозування. Профіль компетентностей кандидата представлений у табл. 1.5.

Існує низка методів, що дозволяють формалізувати дані про компетентності та перетворити їх у числові показники, придатні для математичних моделей.

Таблиця 1.5 – Структурований профіль компетентностей кандидата

Категорія	Показники	Методи вимірювання
Професійні навички	Стек технологій, знання фреймворків, рівень програмування	Тести, сертифікації, аналіз Git-активності
Soft Skills	Комунікація, командність, стресостійкість	Поведінкові оцінки, анкетування, ML-аналіз текстів
Досвід	Ролі в минулих проєктах, тривалість, складність завдань	Резюме, база корпоративних даних
Поведінкові індикатори	Конфліктність, стиль роботи	Психологічні методики, аналіз історії взаємодій
Сумісність	Типологія особистості, робочі стилі	Спряженість профілів, рекомендаційні моделі

Найпоширенішими підходами, які дають змогу формалізувати компетентності кандидатів є:

1. Метод експертного оцінювання – використовується для формування ваг критеріїв або початкової оцінки навичок. Перевага: швидкість. Недолік: суб'єктивність, яка компенсується математичною нормалізацією та узгодженням оцінок.

2. Тестування та автоматизована перевірка навичок – застосовується для Hard Skills за допомогою Code Challenges, Online-judge систем, Technical interviews automation.

3. Кластеризація та ML-підходи – застосовуються для виявлення закономірностей. Наприклад, K-means забезпечує групування кандидатів за компетентнісними профілями, GMM дозволяє оцінювати ймовірності належності до ролі, Decision Trees / Random Forest застосовується для прогнозування успішності кандидати при виконанні проєкту.

4. Методи багатокритеріального аналізу, зокрема, АНР, TOPSIS використовуються як універсальні методи для задач вибору.

У табл. 1.6 наведено суть і доцільність використання методів багатокритеріального аналізу.

Таблиця 1.6 – Методи багатокритеріального аналізу

Метод	Суть	Використання
АНР	Ієрархічна декомпозиція + порівняння попарно	Обчислення ваг критеріїв
TOPSIS	Відстань до ідеального рішення	Рейтингування кандидатів
ELECTRE	Принцип переваги	Відсів неефективних альтернатив

Таким чином, моделі компетентностей забезпечують основу для формалізації профілю кандидата, а застосування таблиць та кластерних моделей дозволяє об'єктивно оцінювати фахівців. Методи MCDA (АНР, TOPSIS) інтегруються в системи прийняття рішень для обчислення зважених рейтингів, а використання машинного навчання підвищує точність оцінювання та дозволяє прогнозувати потенціал учасника.

1.3. Аналіз вимог і пропозицій ІТ- ринку в контексті підготовки фахівців навчальними закладами

Одним із додаткових чинників, що враховується під час формування команди для виконання ІТ-проектів, є здобутий освітній рівень фахівців, зокрема репутація та сприйняття закладу вищої освіти, який вони закінчили. Для роботодавців та менеджерів проектів цей показник не є визначальним, однак він може слугувати індикатором якості базової підготовки та здатності випускника адаптуватися до професійних вимог [1].

Для оцінювання ефективності освітніх програм у сфері інформаційних технологій доцільно використовувати результати незалежних опитувань фахівців, які вже працюють у галузі. Зокрема, аналітичні матеріали платформи DOU ґрунтуються на зворотному зв'язку від випускників, які оцінюють свої навчальні заклади з позиції реального професійного досвіду [1]. Важливою особливістю такого підходу є використання критерію рекомендації кафедри або факультету як потенційного місця навчання, що зменшує вплив суб'єктивних емоційних оцінок і підвищує репрезентативність вибірки [1].

На відміну від формальних рейтингів, оцінки випускників враховують не лише перелік вивчених дисциплін, але й неявні чинники, зокрема здатність випускника працевлаштуватися, ефективно застосовувати отримані знання та конкурувати на IT-ринку. Таким чином формується узагальнена характеристика відповідності навчального процесу реальним вимогам галузі.

У межах кваліфікаційної роботи, для об'єктивності аналізу таких показників, використано результати опитування від DOU, яке проводилось у 2019 р. Участь в опитуванні взяли понад дві тисячі спеціалістів, які працюють у сфері інформаційних технологій. Структура зайнятості респондентів за професійними ролями наведена на рис. 1.1.

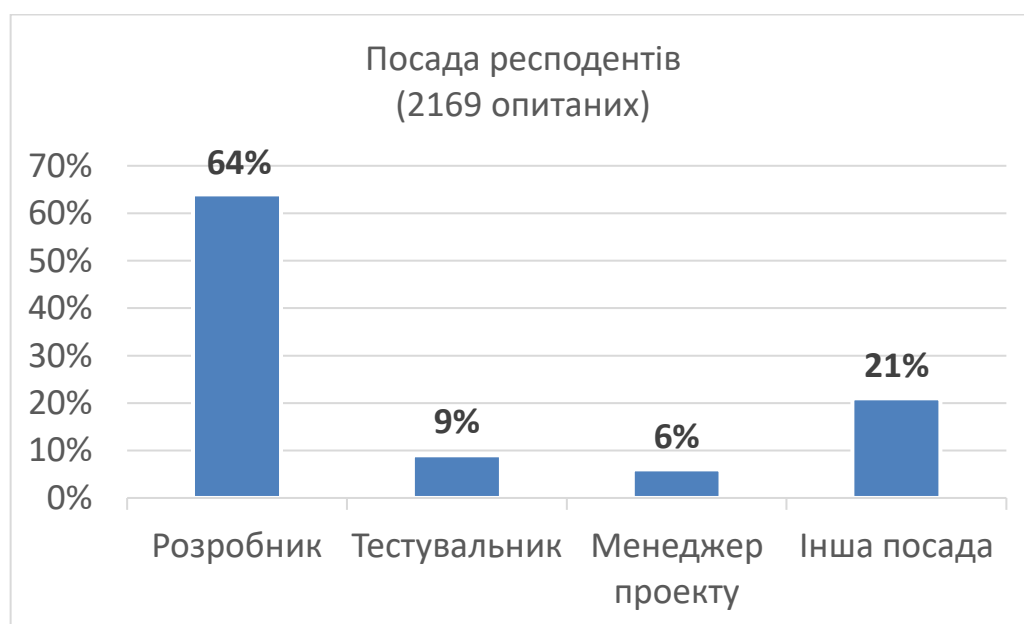


Рисунок 1.1 – Структура професійних ролей опитаних фахівців в IT-галузі

Аналіз отриманих даних свідчить, що домінуючою категорією є розробники програмного та апаратного забезпечення. Частка спеціалістів, задіяних у напрямках системного адміністрування, інтеграції рішень і DevOps, становить приблизно п'яту частину вибірки. Представники ролей тестування та управління проєктами формують найменшу групу серед опитаних, що вказує на орієнтацію ринку праці передусім на інженерні компетентності [10].

Для аналізу результативності освітніх програм респондентам було запропоновано вказати рівень освіти, який вони здобули у відповідних закладах. Розподіл відповідей за освітніми ступенями представлено на рис. 1.2.

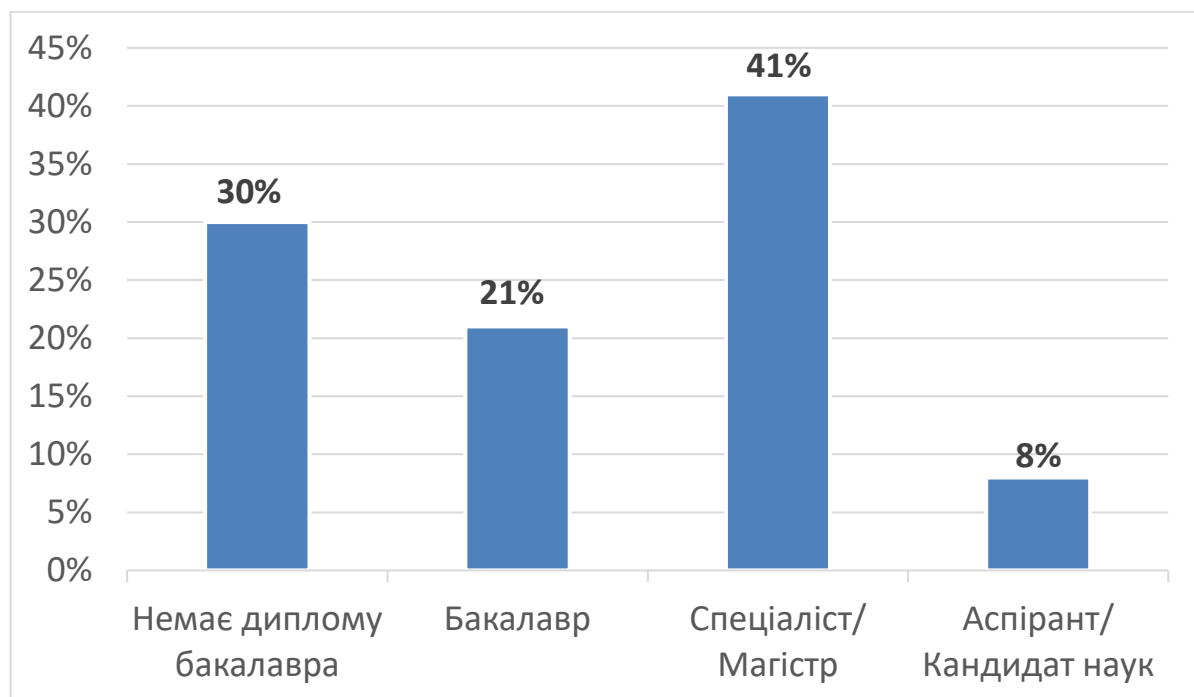


Рисунок 1.2 – Розподіл ІТ-фахівців за здобутим освітнім рівнем

Як видно з наведених даних, переважну частину вибірки становлять випускники, які отримали диплом спеціаліста або магістра. Значна частка респондентів має освіту молодшого спеціаліста, здобуту в коледжах, тоді як бакалаврський рівень представлений меншою кількістю опитаних. Частка осіб з науковими ступенями є незначною, що відображає практичну орієнтацію ІТ-ринку [1].

До опитування були залучені випускники різних закладів вищої освіти України. Узагальнений розподіл респондентів за навчальними закладами подано на рис. 1.3.

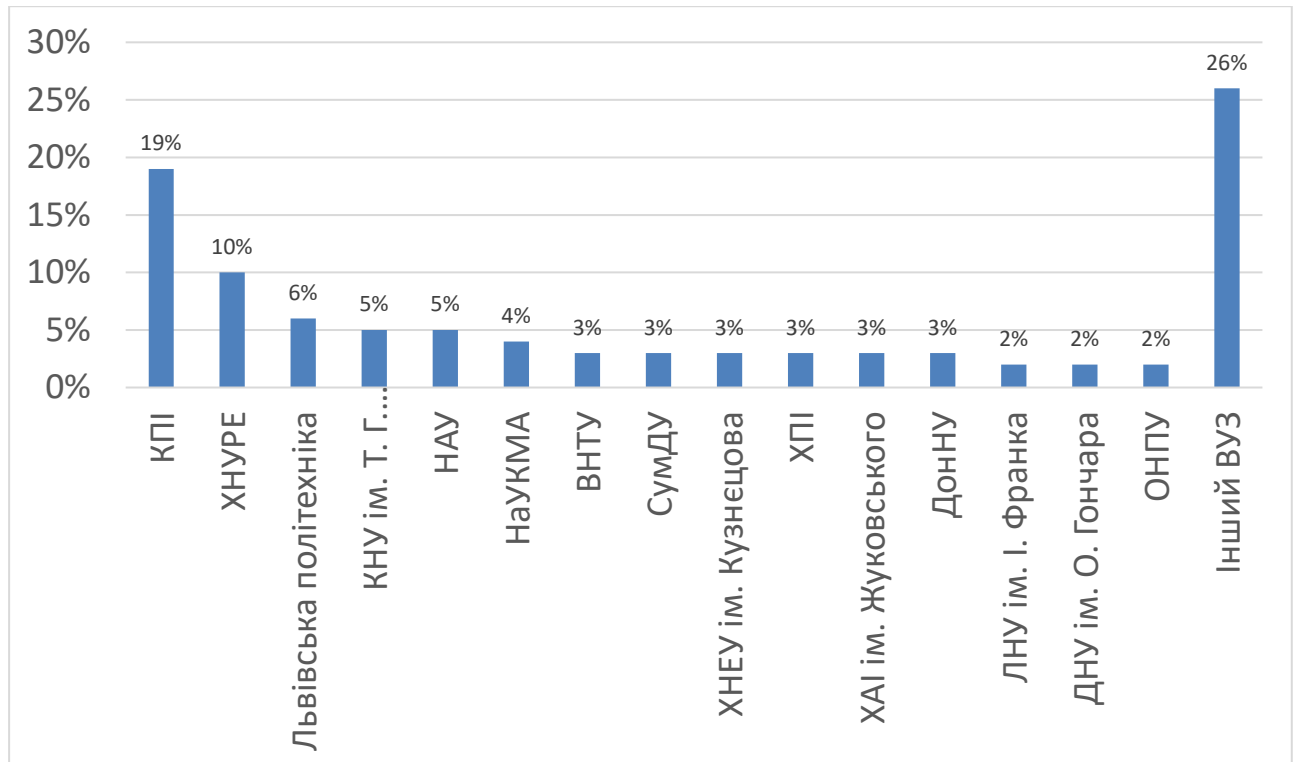


Рисунок 1.3 – Представленість випускників різних закладів вищої освіти серед ІТ-фахівців

Для кількісного оцінювання якості навчального процесу було використано шкалу від 0 до 10 за низкою атрибутів, що характеризують освітнє середовище. Перелік таких атрибутів та їх середні значення наведено в табл. 1.7.

Таблиця 1.7 – Атрибути зворотного зв'язку випускників

Показник	Середня оцінка
Репутація факультету	6,4
Доступність навчальних матеріалів	6,3
Якість викладання	6,1
Орієнтація адміністрації на студентів	6,0

Продовження таблиці 1.7

Показник	Середня оцінка
Рівень підготовки навчальної групи	6,0
Технічні можливості навчання	5,9
Організація навчального процесу	5,7
Матеріально-технічний стан	5,6
Зацікавленість у навчанні	5,6
Корисність професійних контактів	5,6
Складність навчання	5,3
Практична цінність знань	5,3
Практичний досвід викладачів	5,3
Контроль академічної доброчесності	5,0
Економічна доцільність навчання	4,5
Актуальність знань	4,4
Активність спільноти випускників	4,2

Узагальнення результатів показує, що лише незначна кількість закладів освіти отримує високі оцінки, тоді як середній рівень підготовки оцінюється як помірний. Математичне сподівання готовності рекомендувати навчальний заклад коливається на рівні близько 6 балів, що свідчить про обмежену відповідність освітніх програм вимогам ІТ-ринку. Візуалізація інтегральних оцінок наведена на рис. 1.4.

Встановлено, що на рішення щодо рекомендації навчального закладу суттєво впливають практична орієнтованість знань, їх актуальність та можливість застосування у професійній діяльності. Окрему роль відіграє рівень практичного досвіду викладачів, який, за оцінками респондентів, не повною мірою відповідає потребам ІТ-галузі. Мотиваційні чинники вступу до закладів вищої освіти також мають істотний вплив на подальше сприйняття якості навчання. Узагальнені результати щодо цілей здобуття освіти подано на рис. 1.5.



Рисунок 1.4 – Інтегральна оцінка якості підготовки фахівців за результатами опитування

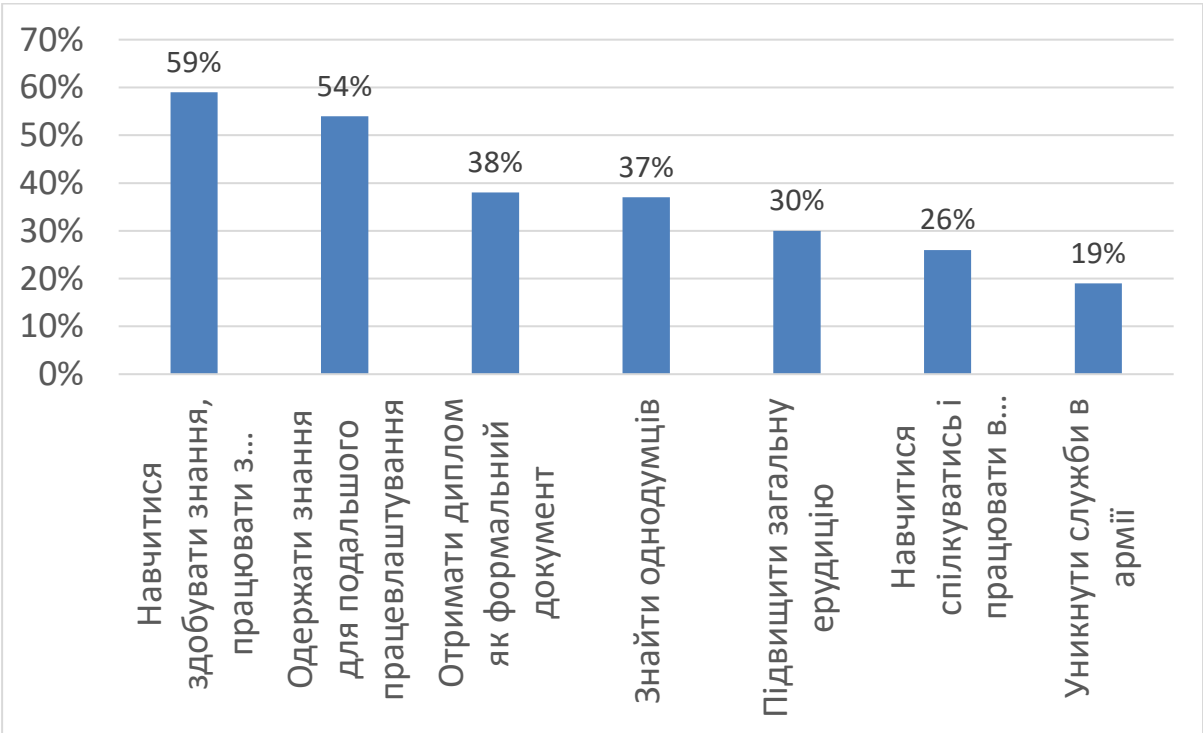


Рисунок 1.5 – Мотиваційні цілі вступу до закладів вищої освіти серед ІТ-фахівців

Аналіз показує, що на початкових етапах навчання значна частина студентів орієнтується на формування навичок самостійного здобуття знань. Водночас очікування отримання прикладних професійних компетентностей є спільним для студентів незалежно від форми навчання [11-13].

Дослідження також засвідчують, що перелік технологій і дисциплін, які вивчаються у вищих навчальних закладах, здебільшого охоплює базові концепції програмування та проєктування інформаційних систем. Розподіл технологій, що викладалися у ВНЗ, наведено на рис. 1.6.

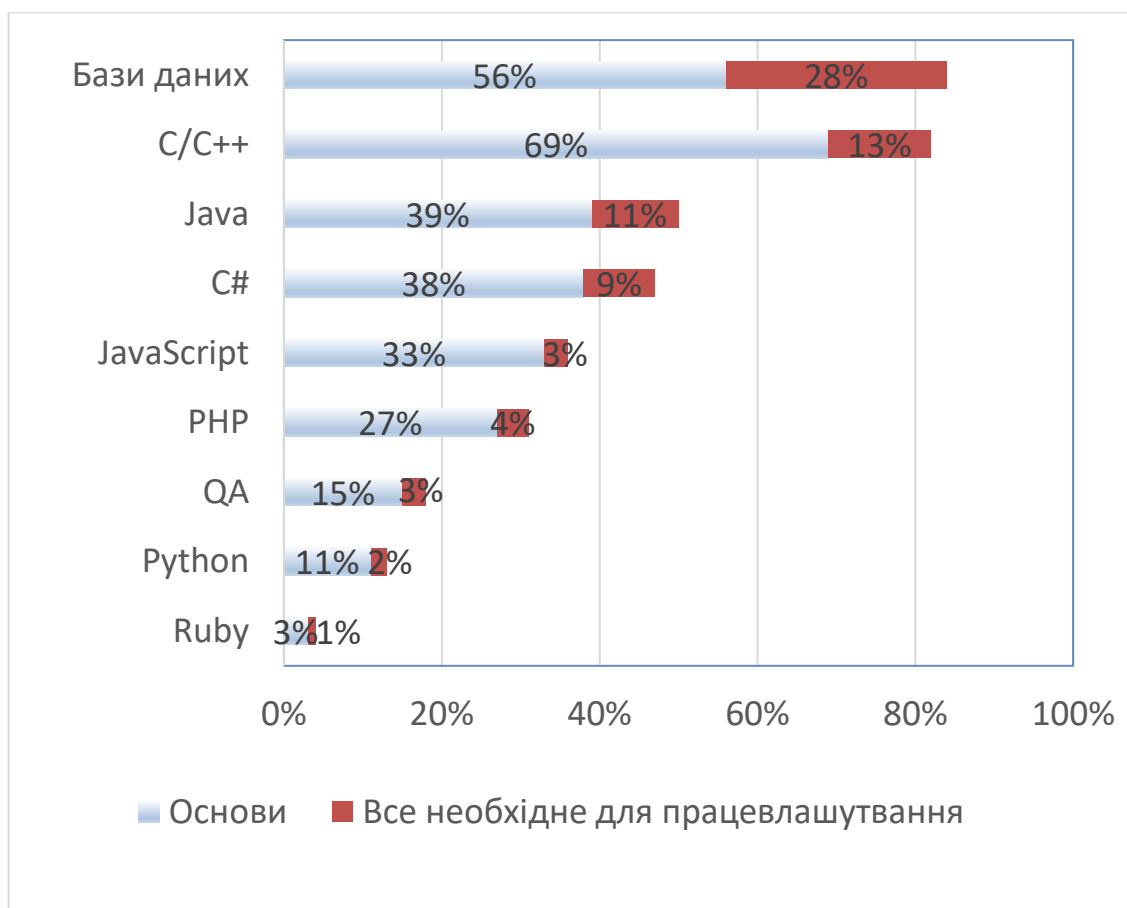


Рисунок 1.6 – Структура технологічної підготовки у закладах вищої освіти

Отримані результати свідчать, що навчальні програми формують фундаментальні знання, проте не забезпечують повною мірою готовність випускників до практичної роботи в ІТ-проєктах. Це зумовлює необхідність додаткового навчання через зовнішні освітні платформи та курси.

Водночас простежується позитивна тенденція до поглиблення співпраці між закладами вищої освіти та ІТ-компаніями. Дані щодо залучення практиків до навчального процесу та підтримки працевлаштування випускників подано на рис. 1.7 та рис. 1.8 відповідно.

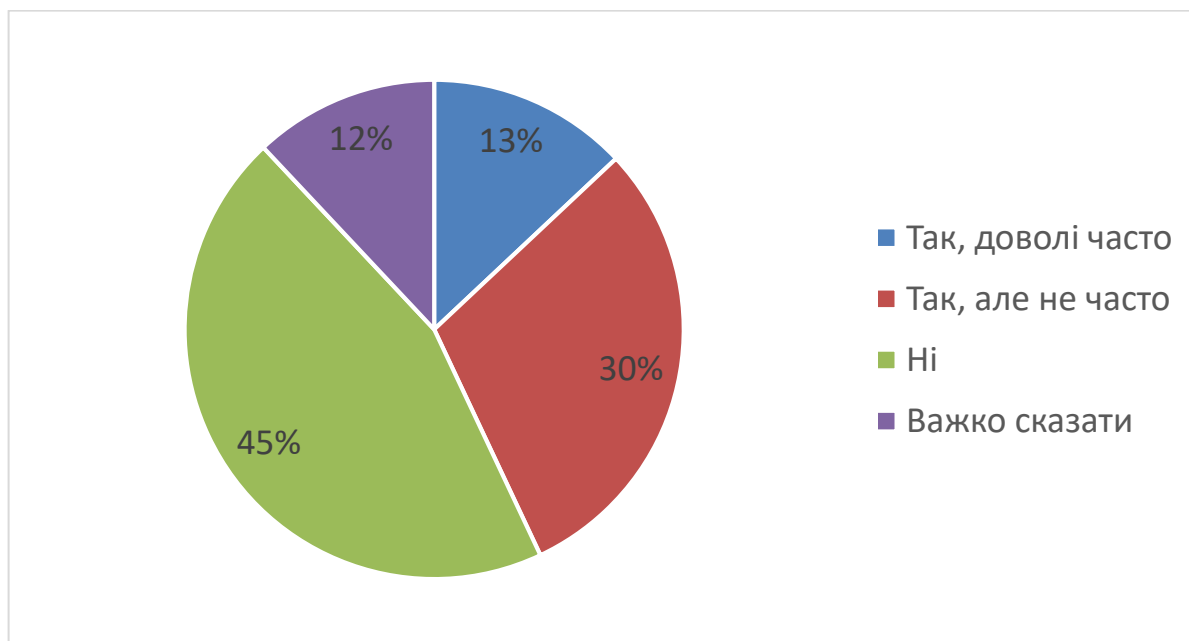


Рисунок 1.7 – Участь ІТ-практиків у навчальному процесі закладів вищої освіти

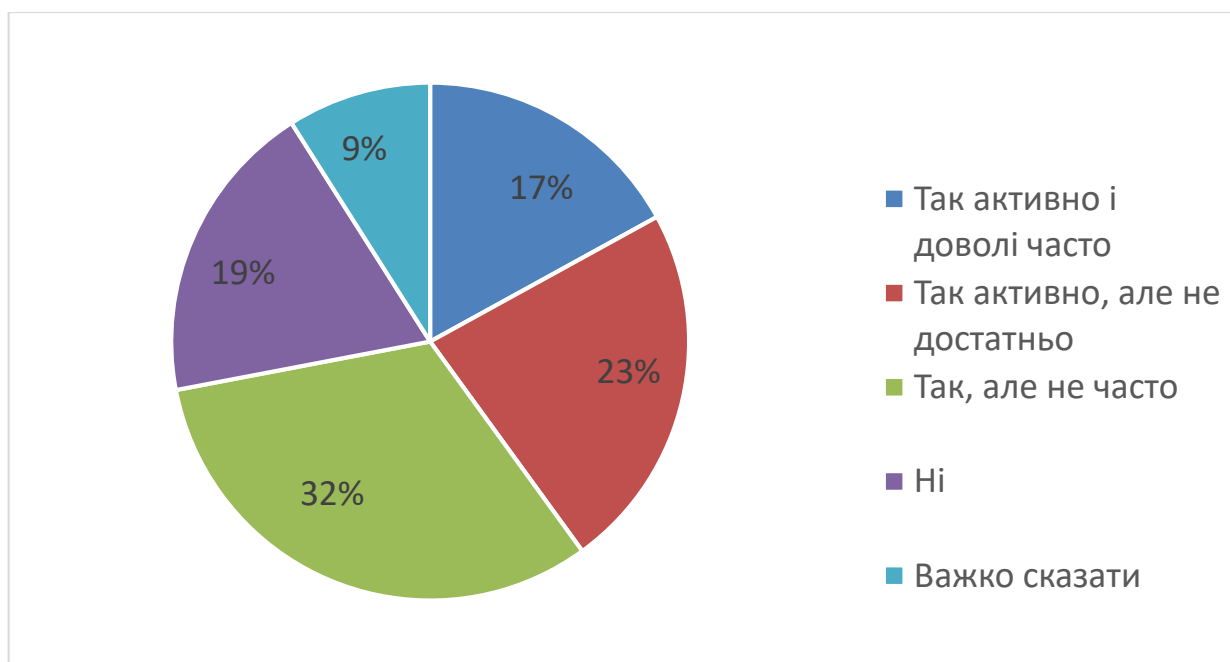


Рисунок 1.8 – Рівень взаємодії закладів освіти з ІТ-компаніями у працевлаштуванні випускників

Таким чином, проведений аналіз підтверджує доцільність врахування характеристик освітнього середовища та репутації закладів вищої освіти як допоміжного чинника при формуванні команд для виконання ІТ-проектів. Отримані результати створюють підґрунтя для подальшої інтеграції освітніх та ринкових даних у системи підтримки прийняття рішень.

1.4. Висновки до розділу

У першому розділі магістерської роботи виконано комплексний аналіз підходів, моделей і факторів, що впливають на процес формування команд для виконання ІТ-проектів, з позицій системного аналізу та підтримки прийняття рішень. Основні результати полягають в наступному:

1. Обґрунтовано доцільність застосування системного підходу до задачі формування команд для виконання ІТ-проектів, що дало змогу розглядати команду як цілісну динамічну систему, у якій ефективність визначається не лише рівнем компетентностей окремих учасників, а й характером їх професійної та міжособистісної взаємодії, а також формалізувати процес прийняття рішень у вигляді багатокритеріальної задачі.

2. Проаналізовано сучасні моделі компетентностей і методи оцінювання учасників ІТ-проектів, що дало змогу структурувати профіль кандидата за професійними, поведінковими, рольовими та досвідними характеристиками і створити інформаційну основу для подальшого використання методів підтримки прийняття рішень, багатокритеріального аналізу та машинного навчання.

3. Досліджено вимоги та тенденції ІТ-ринку в контексті підготовки фахівців закладами вищої освіти, що дало змогу виявити розбіжність між фундаментальною підготовкою випускників і практичними потребами ІТ-проектів, а також обґрунтувати доцільність урахування характеристик освітнього середовища та репутації закладів освіти як додаткових факторів при формуванні команд.

РОЗДІЛ 2

РОЗРОБЛЕННЯ ТЕХНОЛОГІЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ФОРМУВАННЯ КОМАНД ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ

2.1. Моделювання процесу формування команд на основі методів колаборативної фільтрації

Рекомендаційні системи належать до класу систем інтелектуальної фільтрації інформації, які призначені для прогнозування значень корисності об'єктів та формування впорядкованих списків рекомендацій відповідно до заданих уподобань або вимог користувачів. У контексті даної роботи як користувачі розглядаються компанії, що реалізують ІТ-проєкти, а як об'єкти рекомендацій – фахівці з розроблення інформаційних систем, зокрема їх професійні знання, навички та досвід.

Таким чином, задача рекомендаційної системи полягає у визначенні ступеня відповідності між вимогами ІТ-проєкту та характеристиками потенційних учасників команди. У загальному випадку процес формування рекомендацій може бути поданий у вигляді класичної схеми, що відображає взаємодію множин користувачів, об'єктів та оцінок між ними (рис. 2.1).

Залежно від використовуваних джерел інформації та способів її обробки, рекомендаційні системи умовно поділяють на три основні класи [15]:

- системи рекомендацій, побудовані на основі колаборативної фільтрації;
- системи, що використовують контентно-орієнтований аналіз;
- гібридні рекомендаційні системи, які поєднують властивості двох попередніх підходів.

Найбільш поширеними у практичних застосуваннях є контентно-орієнтований підхід та методи колаборативної фільтрації. У контентних рекомендаційних системах здійснюється побудова профілів користувачів і профілів об'єктів. Профіль користувача, як правило, включає інформацію про

професійні компетентності розробника, його досвід, вік, географічне розташування та інші релевантні характеристики. Профіль об'єкта описує вимоги до ІТ-проєкту, зокрема необхідний технологічний стек, тривалість реалізації, складність завдань і додаткові атрибути, що характеризують предметну область.

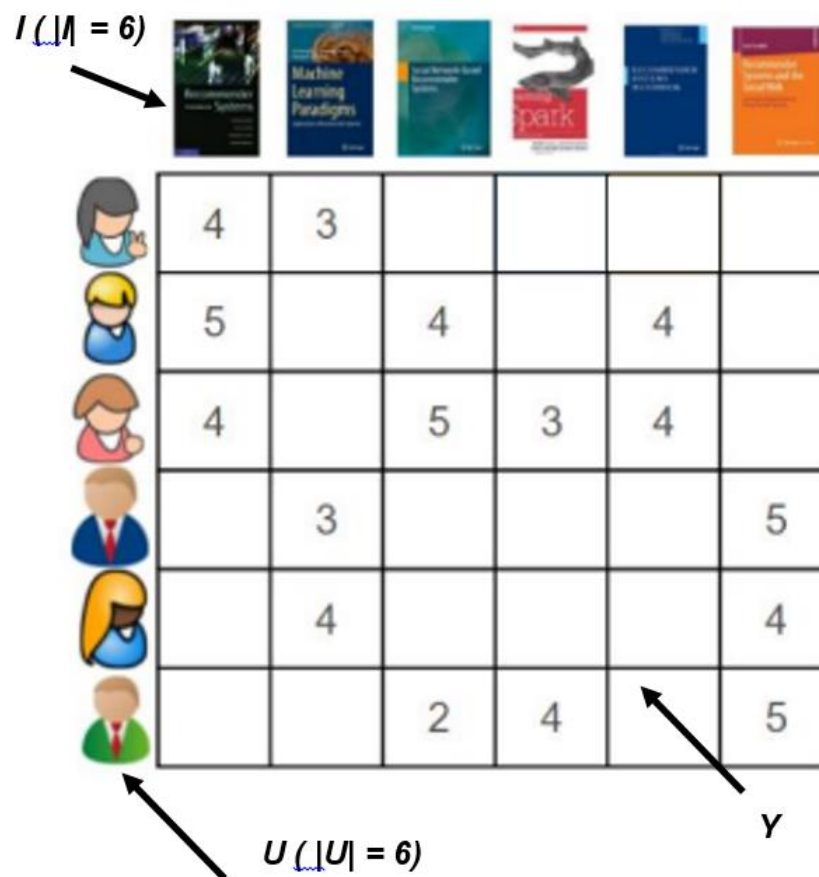


Рисунок 2.1 – Візуалізація класичної задачі формування рекомендацій

На відміну від контентного підходу, методи колаборативної фільтрації базуються переважно на аналізі історичних даних про взаємодію користувачів з об'єктами. У задачі формування команд такими даними можуть бути відомості про участь розробників у попередніх проєктах, оцінки їх роботи, рейтинги закладів освіти, у яких вони навчалися, а також експертні або користувацькі відгуки. Особливістю колаборативної фільтрації є відсутність жорсткої

прив'язки до предметної категорії об'єктів, однак ключовою складністю є виявлення прихованих властивостей, які не задані в явному вигляді [16].

Однією з найбільш відомих проблем колаборативної фільтрації є проблема «холодного старту», яка виникає у випадках, коли інформація про нового розробника або новий проєкт є відсутньою або недостатньою. За таких умов ускладнюється пошук подібних об'єктів та побудова коректних рекомендацій, що потребує застосування додаткових механізмів або гібридних моделей.

Класифікацію основних методів та їх різновидів, що застосовуються при побудові рекомендаційних систем, наведено на рис. 2.2.

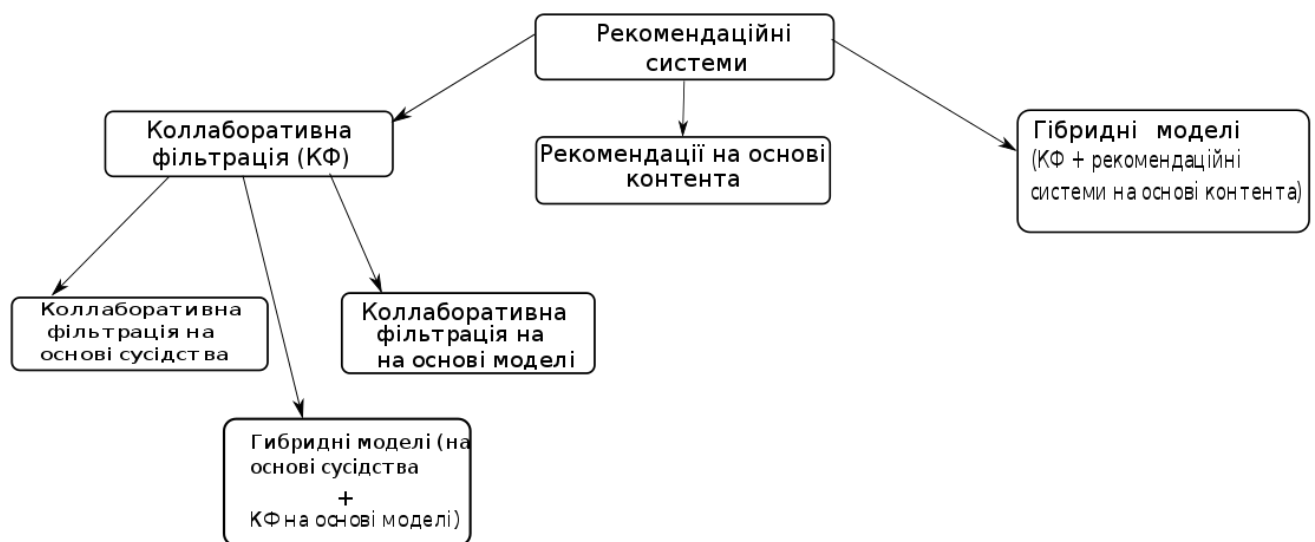


Рисунок 2.2 – Класифікація методів побудови рекомендаційних систем

Серед методів колаборативної фільтрації виділяють алгоритми, засновані на пошуку k найближчих сусідів, підходи на основі статистичних і факторизаційних моделей, а також гібридні рішення, які комбінують декілька методів для підвищення точності рекомендацій. Такі підходи добре масштабуються та дозволяють враховувати складні взаємозв'язки між учасниками команди та вимогами ІТ-проєктів [17-19].

Функціонування рекомендаційної системи передбачає накопичення даних про розробників із використанням як явних, так і неявних способів збору інформації. До методів явного збору даних належать:

- оцінювання розробниками один одного за заданою шкалою;
- експертне ранжування кандидатів менеджерами з підбору персоналу;
- вибір компаніями кращої альтернативи за сукупністю критеріїв;
- формування упорядкованих списків розробників відповідно до визначених вимог.

Неявний збір інформації ґрунтується на аналізі поведінки користувачів та може включати:

- спостереження за активністю розробника під час пошуку роботи та подання резюме;
- аналіз навігації та часу перебування на веб-ресурсах;
- опрацювання цифрових слідів, що характеризують професійну діяльність розробника.

Загальний принцип роботи рекомендаційних сервісів полягає у виявленні подібності між об'єктами або користувачами та формуванні пропозицій на основі цієї подібності [20]. У задачі формування команди розробників результатом роботи системи є ранжований список кандидатів, упорядкований за рівнем відповідності вимогам ІТ-проєкту з використанням відповідних метрик подібності та критеріїв оцінювання.

2.2. Метрики подібності в моделях колаборативної фільтрації

У результаті аналізу та формалізації задач, у яких застосовуються методи колаборативної фільтрації, встановлено, що їх ключовим елементом є обчислення кількісної міри схожості між об'єктами. Саме від вибору відповідної метрики подібності залежить точність прогнозування та якість сформованих рекомендацій [21].

Розглянемо та обґрунтуємо доцільність використання різних метрик подібності для задачі підбору потенційних розробників з метою реалізації проєктів комп'ютерних систем. Для ілюстрації підходу сформуємо матрицю типу «розробник – компетентності», елементи якої містять числові оцінки рівня

володіння відповідними технологіями. У такій матриці рядки відповідають окремим розробникам (А, В, С, D), а стовпці – переліку навичок і вмінь, необхідних для виконання ІТ-проєкту (рис. 2.3).

							
A 	4			5	1		
B 	5	5	4				
C 				2	4	5	
D 		3					3

Рисунок 2.3 – Приклад матриці відповідності «розробник – компетентності»

На основі побудованої крос-табуляції для кожного розробника формується вектор технологічних компетентностей r_i . Для демонстрації принципу визначення подібності розглянемо пари розробників А–В та А–С. Послідовність аналізу включає такі кроки:

- вибір об'єктів x і y , яким відповідають вектори r_x та r_y ;
- визначення числової міри подібності $sim(x, y)$;
- інтуїтивну перевірку отриманих результатів на основі структури матриці, з якої випливає, що схожість між А і В є вищою, ніж між А і С.

Далі наведено аналіз основних метрик подібності, що застосовуються у методах колаборативної фільтрації.

Побудувавши матрицю крос-табуляції (рис. 2.3), для кожного розробника формуємо вектори технологій розробки комп'ютерних систем – r_i . Розглянемо конкретний приклад для визначення міри подібності між розробниками А і В, а також А і С. Для цього:

- визначаємо розробників x і y з відповідними векторами r_x та r_y ;

- необхідно визначити метрики подібності між розробниками $sim(x, y)$;
- з рис. 2.8 інтуїтивно видно, що $sim(A, B) > sim(A, C)$;

Проведемо аналіз та обґрунтуємо метрики подібності для визначення подібності розробників.

2.2.1. Метрика подібності Жаккара

Коефіцієнт Жаккара є однією з найпростіших метрик оцінювання схожості та визначається як відношення кількості спільних елементів двох множин до кількості елементів їх об'єднання [22]:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2.1)$$

У практичних задачах інформатики коефіцієнт Жаккара широко використовується для пошуку схожих текстових об'єктів, зокрема у задачах виявлення дублікатів та плагіату. Для прискорення обчислень у великих вибірках застосовується алгоритм MinHash. Структуру даних при використанні цієї метрики подано на рис. 2.4.

З аналізу прикладу випливає, що значення подібності між розробниками A і B дорівнює $sim(A, B) = 1/5$, тоді як для пари A і C — $sim(A, C) = 2/4$, тобто $sim(A, B) < sim(A, C)$.

							
A 	4			5	1		
B 	5	5	4				
C 				2	4	5	
D 		3					3

Рисунок 2.4 – Схема обчислення подібності за коефіцієнтом Жаккара

Основним недоліком цієї метрики є ігнорування числових значень оцінок у матриці, оскільки враховується лише факт наявності або відсутності навички. Водночас коефіцієнт Жаккара доцільно застосовувати в умовах «холодного старту», коли детальна інформація про рівень компетентностей розробників відсутня.

2.2.2. Метрика на основі обчислення косинусів кута між векторами

Оскільки оцінки знань і вмінь розробників можуть бути представлені у вигляді векторів у багатовимірному просторі, для вимірювання їх схожості доцільно використати метрику косинуса кута між векторами [23] (рис. 2.5).

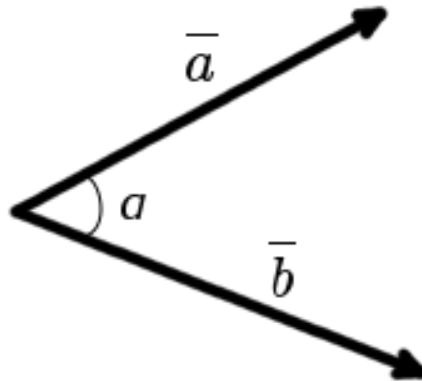


Рисунок 2.5 – Геометрична інтерпретація кута між векторами компетентностей

Косинус кута між двома векторами r_x та r_y обчислюється за формулою:

$$\text{sim}(x, y) = \frac{r_x \cdot r_y}{\|r_x\| \cdot \|r_y\|} \quad (2.2)$$

Для коректного застосування цієї метрики матрицю «розробник – технологія» доповнюють нульовими значеннями у невизначених комірках, як показано на рис. 2.6.

							
A 	4	0	0	5	1	0	0
B 	5	5	4	0	0	0	0
C 	0	0	0	2	4	5	0
D 	0	3	0	0	0	0	3

Рисунок 2.6 – Крос-табуляція після доповнення нульовими значеннями

У результаті обчислень одержано, що $sim(A, B) = 0,38$, тоді як $sim(A, C) = 0,32$, що відповідає інтуїтивним очікуванням щодо більшої подібності між розробниками A і B . Водночас недоліком цієї метрики є можливі втрати інформації через значну кількість нульових значень у розріджених матрицях [21].

2.2.3. Метрика на основі коефіцієнта кореляції Пірсона

З метою зменшення впливу індивідуальних масштабів оцінювання та усунення недоліків класичної косинусної міри доцільно використовувати коефіцієнт кореляції Пірсона, який також називають центрованим косинусом:

$$sim(x, y) = \frac{\sum(r_{x,i} - \bar{r}_x) \cdot (r_{y,i} - \bar{r}_y)}{\sqrt{\sum(r_{x,i} - \bar{r}_x)^2} \cdot \sqrt{\sum(r_{y,i} - \bar{r}_y)^2}} \quad (2.3)$$

На першому етапі обчислюється середнє значення оцінок для кожного розробника, після чого вихідна матриця трансформується шляхом віднімання відповідних середніх значень. Отриману матрицю подано на рис. 2.7.

							
A 	2/3			5/3	-7/3		
B 	-2/3	1/3	-2/3				
C 				-5/3	1/3	4/3	
D 		0					0

Рисунок 2.7 – Центрована матриця «розробник – технології»

Подальше обчислення косинуса кута між центрованими векторами дає такі результати:

- $\text{sim}(A, B) = 0,09$;
- $\text{sim}(A, C) = -0,56$.

Це свідчить про значно вищу подібність між розробниками А і В. Таким чином, у даному прикладі метрика Пірсона забезпечує найбільш адекватні результати та краще узгоджується з очікуваннями користувачів системи.

Отже, проведений аналіз і обґрунтування метрик подібності об'єктів створює передумови для подальшої формалізації задачі прогнозування значень оцінок та побудови рекомендацій у системах підтримки прийняття рішень на основі методів колаборативної фільтрації.

2.3. Модель підтримки прийняття рішень щодо формування команди для виконання ІТ-проектів

Для побудови технології підтримки прийняття рішень щодо формування команди для виконання ІТ-проектів необхідно здійснити комплексний аналіз предметної області та визначити множину чинників, які безпосередньо впливають на вибір кожного потенційного учасника команди. У межах такої

задачі процес підбору розробників розглядається не як окреме кадрове рішення, а як елемент загального механізму управління ІТ-проєктом, що потребує формалізації та подальшої автоматизації [24].

Прийняття рішення щодо складу команди повинно базуватися як на характеристиках окремих фахівців, так і на вимогах, які формуються з боку компанії-замовника або організації, що реалізує ІТ-проєкт. Саме поєднання цих двох груп факторів створює основу для побудови системи підтримки прийняття рішень, здатної забезпечити обґрунтований вибір кандидатів із множини альтернатив.

У загальному вигляді процес ухвалення бізнес-рішення щодо ініціювання та реалізації ІТ-проєкту, включно з формуванням команди розробників, може бути представлений у вигляді структурної моделі, що відображає взаємозв'язок між цілями проєкту, вимогами замовника та ресурсними обмеженнями.

Для формалізованого опису потенційного учасника команди доцільно виділити такі ключові групи характеристик:

- професійні компетентності – сукупність мов програмування, інструментів та технологій, якими володіє розробник;
- технологічна спеціалізація – досвід роботи з підходами та методами проєктування програмних і апаратних компонентів комп'ютерних систем;
- освітній профіль – базова спеціальність та заклад вищої освіти, що характеризують фундаментальний рівень підготовки;
- практичний досвід – тривалість і характер участі фахівця в попередніх ІТ-проєктах;
- територіальний чинник – місце проживання, яке може впливати на формат співпраці, можливість релокації або дистанційної роботи.

На рис. 2.8 показано узагальнену модель для прийняття рішення щодо формування команди розробника для виконання ІТ-проєкту.

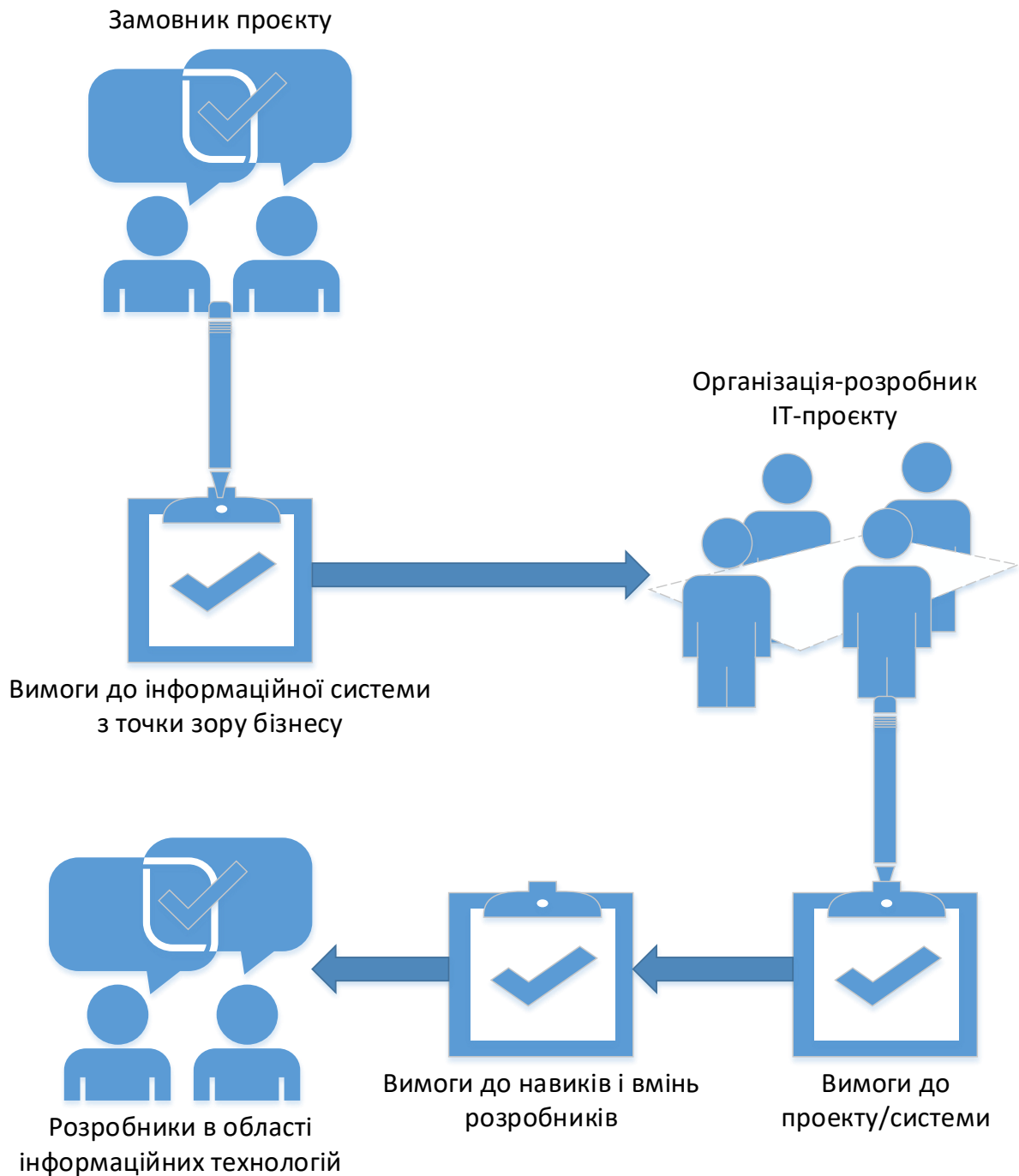


Рисунок 2.8 – Узагальнена модель процесу прийняття рішення щодо реалізації ІТ-проекту

З метою використання зазначених факторів у системі підтримки прийняття рішень кожній характеристиці доцільно поставити у відповідність ваговий коефіцієнт, що відображає її відносну важливість у контексті конкретного ІТ-проекту. Це дозволяє переходити від якісного опису кандидатів до кількісного

оцінювання та формувати ранжований список альтернатив при комплектуванні команди.

Окрім характеристик розробників, у процесі прийняття рішень необхідно формалізувати параметри, що описують компанію-замовника та сам ІТ-проект. До таких параметрів доцільно віднести:

- інтегральний показник компанії, який враховує її досвід діяльності на ІТ-ринку та ділову репутацію;
- географічні характеристики, що визначають місце розташування офісу або центру розробки;
- проєктні вимоги, які включають технічні та технологічні характеристики комп'ютерної системи, а також планові терміни виконання робіт.

Таким чином, після ідентифікації основних сутностей та атрибутів, релевантних для задачі формування команди ІТ-проекту, виникає необхідність побудови формалізованої моделі процесу підбору розробників. Така модель є основою для подальшої реалізації технології підтримки прийняття рішень і дозволяє автоматизувати процедуру вибору оптимального складу команди з урахуванням множини критеріїв і обмежень.

2.3.1. Формалізована модель опису об'єктів у задачі формування команди для виконання ІТ-проектів

Для формального опису сутностей і атрибутів предметної області у задачі підтримки прийняття рішень щодо формування команд доцільно застосовувати апарат теорії множин і реляційної алгебри. Такий підхід дозволяє уніфікувати представлення різномірних характеристик, забезпечити їх подальшу обробку та використання в алгоритмах аналізу і ранжування альтернатив.

Розробника ІТ-проекту доцільно розглядати як складну багатокомпонентну сутність, формалізований опис якої може бути поданий у вигляді кортежу:

$$D = \langle S_p, S_h, E, X \rangle, \quad (2.4)$$

де S_p – сукупність знань і навичок у сфері програмної розробки, зокрема мов програмування, фреймворків, систем керування базами даних та супутніх технологій;

S_h – компетентності, пов'язані з проєктуванням і реалізацією апаратних компонентів комп'ютерних систем;;

E – інформація про освітній профіль фахівця, включно із закладом освіти та спеціальністю;

X – досвід практичного використання технологій при реалізації ІТ-проєктів.

Професійні компетентності розробника у сфері програмного забезпечення можуть бути представлені у вигляді множини:

$$S_p = \{L, F\}, \quad (2.5)$$

де $L = \{l_1, l_2, \dots, l_n\}$ – множина мов програмування, якими володіє розробник, n – їх кількість;

$F = \{f_1, f_2, \dots, f_m\}$ – множина фреймворків і програмних технологій, освоєних фахівцем, m – кількість таких технологій.

З урахуванням того, що сучасні інформаційні системи є програмно-апаратними комплексами, навички у сфері апаратного проєктування доцільно подати у вигляді множини:

$$S_h = \{H, C\}, \quad (2.6)$$

де $H = \{h_1, h_2, \dots, h_k\}$ – множина технологій апаратного проєктування, k – їх кількість;.

$C = \{c_1, c_2, \dots, c_q\}$ – набір середовищ та інструментів проєктування комп'ютерних систем, q – кількість середовищ.

Для урахування історичних і контекстних факторів, що впливають на процес формування команди, доцільно формалізувати освітній та соціально-географічний профіль розробника у вигляді множини:

$$E = \{U, G, R\}, \quad (2.7)$$

де $U = \{u_1, u_2, \dots, u_s\}$ – множина навчальних закладів або освітніх програм, які закінчив або пройшов фахівець, s – їх кількість;

$G = \{g_1, g_2, \dots, g_t\}$ – множина географічних регіонів, пов'язаних із місцем перебування розробника, t – кількість регіонів;

$R = \{r_1, r_2, \dots, r_p\}$ – кількісні оцінки репутації або рейтингу закладів освіти чи курсів, p – кількість оцінок.

У формулі (2.4) визначено також компонент E_{xp} , який представляє динамічну змінну, що описує термін використання технологій розробником комп'ютерних систем.

Таким чином, для опису навиків конкретного розробника комп'ютерних систем формується множина елементів, які можна представити у вигляді

$$D = \{S_p, S_h, E, X\}, \quad (2.8)$$

Аналогічним чином формалізується й суб'єкт прийняття рішень – компанія, що здійснює формування команди для реалізації ІТ-проєкту. Її опис може бути поданий у вигляді множини:

$$Comp = \{N_{comp}, Q_{comp}, L_{comp}\} \quad (2.9)$$

де N_{comp} – назва компанії;

Q_{comp} – інтегральний показник репутації або рейтингу компанії на ІТ-ринку;

L_{comp} – географічне розташування компанії або центру розробки.

Ключовим елементом у процесі формування команди є сам ІТ-проект, вимоги якого визначають критерії добору фахівців. Формалізований опис проекту доцільно представити у вигляді:

$$Proj = \{N_{proj}, T_{proj}, S_p, S_h\}, \quad (2.10)$$

де N_{proj} – назва проекту;

T_{proj} – запланована тривалість виконання;

S_p та S_h – множини вимог до програмних і апаратних технологій відповідно.

У результаті виконано формалізацію базових сутностей процесу підтримки прийняття рішень щодо формування команди для виконання ІТ-проектів. Наступним етапом є розроблення методу кількісного оцінювання подібності між компетентнісними профілями розробників та технологічними вимогами проекту, що дозволить перейти до побудови рекомендаційної моделі.

2.3.2. Метод багатокритеріального підбору команди розробників із застосуванням метрик подібності

У підрозділі 2.2 було обґрунтовано доцільність використання метрик подібності в межах методів колаборативної фільтрації для розв’язання задач підтримки прийняття рішень. У контексті формування команди для виконання ІТ-проекту ключовим є кількісне оцінювання відповідності між компетентнісним профілем розробника та технологічними вимогами конкретного проекту.

З огляду на результати попереднього аналізу, для оцінювання подібності між навичками розробників і необхідними вміннями пропонується використати

метрику косинуса кута, яка дозволяє порівнювати вектори у багатовимірному просторі та враховує числові значення компонент.

Принцип визначення подібності між знаннями розробника та вимогами проєкту реалізується окремо для кожної групи технологій. Зокрема, для мов програмування подібність може бути визначена за формулою:

$$sim_L = \frac{L_d \cdot L_p}{\|L_d\| \cdot \|L_p\|}, \quad (2.11)$$

де L_d – вектор мов програмування, якими володіє розробник;

L_p – вектор мов програмування, необхідних для реалізації ІТ-проєкту.

Аналогічно визначається міра подібності для програмних фреймворків:

$$sim_F = \frac{F_d \cdot F_p}{\|F_d\| \cdot \|F_p\|}, \quad (2.12)$$

де F_d – вектор фреймворків, освоєних розробником;

F_p – вектор фреймворків, що вимагаються проєктом.

Для оцінювання відповідності у сфері технологій проєктування інформаційних систем використовується вираз:

$$sim_T = \frac{T_d \cdot T_p}{\|T_d\| \cdot \|T_p\|}, \quad (2.13)$$

де T_d – вектор технологій, з якими працює розробник;

T_p – вектор технологій, необхідних для реалізації проєкту

Подібність за середовищами проєктування визначається формулою:

$$sim_E = \frac{E_d \cdot E_p}{\|E_d\| \cdot \|E_p\|}, \quad (2.14)$$

де E_d – вектор середовищ проєктування, відомих розробнику;

E_p – вектор середовищ, передбачених проєктними вимогами.

У результаті виконання обчислень за формулами (2.11)–(2.14) формується узагальнений вектор подібності, який описує відповідність розробника вимогам проєкту за всіма групами компетентностей:

$$S = \langle sim_L, sim_F, sim_T, sim_E \rangle \quad (2.15)$$

Окрім технологічних характеристик, у процесі прийняття рішення доцільно враховувати додаткові фактори, що впливають на ефективність участі фахівця у проєкті. Зокрема, для урахування освітнього бекграунду та репутації навчального закладу або пройдених курсів пропонується використовувати нормовану оцінку рейтингу:

$$R_n = \frac{R}{R_{max}}, \quad (2.16)$$

де R – значення рейтингу навчального закладу або курсу;

R_{max} – максимальне можливе значення рейтингу;

Важливим додатковим критерієм при формуванні команди є просторовий фактор, що характеризує віддаленість між розробником і компанією-замовником. Його оцінювання може бути здійснене за допомогою функції відстані між географічними координатами:

$$G = f(c_d, c_{comp}), \quad (2.17)$$

де c_d – координати місця перебування розробника;

c_{comp} – координати офісу компанії.

Узагальнену оцінку відповідності розробника вимогам ІТ-проєкту пропонується визначати як агреговану функцію всіх розглянутих факторів:

$$Q = F(G, X, R_n, \|S\|), \quad (2.18)$$

де G – оцінка географічної відповідності;

X – досвід розробника;

R_n – нормована оцінка освітнього рейтингу;

$\|S\|$ – довжина вектора подібності технологічних компетентностей.

У розгорнутому вигляді формулу (2.18) можна подати як зважену суму критеріїв:

$$Q = \omega_1 \cdot G + \omega_2 \cdot X + \omega_3 \cdot R_n + \omega_4 \cdot \|S\| \quad (2.19)$$

У результаті дослідження запропоновано формалізовану модель та метод підтримки прийняття рішень щодо формування команди розробників комп'ютерних систем, який може бути застосований як для внутрішнього комплектування команд, так і для залучення зовнішніх фахівців на основі багатокритеріального аналізу та метрик подібності.

2.4. Метод підвищення стійкості колаборативної фільтрації на основі регуляризованих метрик подібності

Застосування методів колаборативної фільтрації у задачі формування команд для виконання ІТ-проектів дозволяє з високою точністю враховувати відповідність між компетентностями розробників та технологічними вимогами проектів. Водночас практична реалізація таких методів супроводжується значними витратами обчислювальних і ресурсів пам'яті, що зумовлює необхідність оптимізації моделей та процедур обробки даних [25].

У межах колаборативної фільтрації виникає потреба у зберіганні та обробці матриці крос-табуляції, яка відображає взаємозв'язки між розробниками

та технологіями (мовами програмування, фреймворками, середовищами та інструментами проєктування). Значеннями такої матриці можуть бути числові оцінки або ідентифікатори відповідних сутностей. Ускладненням є поява нових властивостей розробників або технологій, для яких відсутні попередні оцінки, що негативно впливає на точність прогнозування.

З метою зменшення обчислювальної складності немає необхідності використовувати повну, надлишкову матрицю взаємодій «розробник – технологія». Доцільним є попереднє групування розробників і технологій за подібними ознаками із застосуванням кластерного аналізу, що дозволяє знизити розмірність задачі та підвищити ефективність обчислень.

Кластерні моделі мають низку переваг, зокрема масштабованість і можливість порівняння об'єктів не з усією базою даних, а лише в межах відповідного кластера. При цьому процес кластеризації може виконуватися в офлайн-режимі, що зменшує навантаження на систему підтримки прийняття рішень у режимі реального часу [26].

Для розв'язання задач кластерного аналізу запропоновано значну кількість підходів, однак у даній роботі акцент зроблено на методах визначення подібності елементів із нормалізацією даних. У такому підході предметну область утворюють розробники, які володіють певними навичками та знаннями, і технології проєктування інформаційних систем. Оцінка взаємодії між ними може бути подана у вигляді трійки «розробник – технологія – оцінка».

Інтерпретація оцінок залежить від сценарію використання рекомендаційної системи. Зокрема, компанія може формувати рекомендації щодо залучення розробників до конкретного ІТ-проєкту, або ж розробник може здійснювати пошук компанії, орієнтуючись на власні компетентності, досвід та професійні уподобання.

У процесі формування оцінок можуть застосовуватися різні шкали вимірювання. Найпоширенішою є дискретна шкала, наприклад у діапазоні від 0 до 5. При цьому значна частина елементів матриці може залишатися

невизначеною, оскільки розробник або не надає оцінку власним навичкам, або володіє технологією на рівні, який не вважає за доцільне декларувати [26-28].

Застосування колаборативної фільтрації у такому контексті передбачає розв'язання двох ключових задач:

- прогнозування відсутніх значень оцінок, що дозволяє оцінити рівень володіння технологіями, які явно не були зазначені розробником;
- формування рекомендацій, тобто побудову ранжованого списку кандидатів для виконання завдань з проєктування та реалізації комп'ютерних систем.

Формально матриця колаборативної фільтрації містить множину розробників U та множину технологій I , для яких визначаються такі характеристики:

I_u – множина технологій, з якими працює розробник u ;

U_i – множина розробників, що володіють технологією i ;

$r_{u,i}$ – оцінка рівня володіння технологією i розробником u ;

r_u – вектор оцінок розробника;

r_i – вектор оцінок технології;

$\bar{r}_u, \bar{r}_i$ – середні значення оцінок для розробника і технології відповідно;

$\hat{r}_{u,i}$ – прогнозоване значення оцінки.

Процедура формування рекомендації включає три основні етапи.

Етап 1. Для кожної технології проєктування j визначається міра подібності до технології i . Для цього може бути використана метрика центрованого косинуса кута, яка набуває вигляду:

$$\text{sim}(x, y) = \frac{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_i) \cdot (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_i)^2} \cdot \sqrt{\sum_{u \in U_{i,j}} (r_{u,j} - \bar{r}_j)^2}}, \quad (2.20)$$

де $U_{i,j}$ – множина розробників, які мають оцінки для обох технологій i та j .

Етап 2. Формується підмножина N_i найбільш подібних до i технологій. Експериментальні дослідження, зокрема наведені в [12], показують, що практично доцільною є кількість близько 30 елементів, хоча це значення може змінюватися залежно від специфіки предметної області.

Етап 3. Виконується прогнозування значення оцінки на основі оцінок подібних об'єктів:

$$\hat{r}_{u,i} = r_i + \frac{\sum_{j \in N_i} \text{sim}(i,j)(r_{u,i} - \bar{r}_i)}{\sum_{j \in N_i} |\text{sim}(i,j)|}, \quad (2.21)$$

Наведена процедура є базовою для методів колаборативної фільтрації, однак у реальних умовах вона потребує врахування додаткових чинників. Зокрема, матриця оцінок зазвичай є сильно розрідженою, а самі оцінки можуть бути суб'єктивними. Частина розробників схильна завищувати власні компетентності, тоді як інші – навпаки, занижувати їх, що спотворює реальну картину і ускладнює прийняття рішень.

Для зменшення впливу таких факторів доцільно використовувати регуляризацію, яка дозволяє обмежити появу випадкових або некоректних залежностей між розробниками та технологіями. Регуляризація реалізується шляхом введення додаткових штрафних членів із параметрами-регуляризаторами.

З метою підвищення точності прогнозування перед обчисленням подібності виконується нормалізація оцінок. Нехай множина відомих оцінок утворює набір K . Тоді прогнозування невідомих значень може бути здійснене за формулою:

$$\hat{r}_{u,i} = \mu + b_u + b_i, \quad (2.22)$$

де μ – загальне середнє значення оцінок;

b_u, b_i – зміщення (відхилення) для розробника і технології відповідно.

Для визначення параметрів b_u та b_i розв'язується задача мінімізації функціоналу найменших квадратів:

$$\min_{b_u, b_i} \sum_{(u,i) \in K} (r_{u,i} - \mu - b_u - b_i)^2 + \lambda(b_u^2 + b_i^2), \quad (2.23)$$

де λ – коефіцієнт регуляризації, що зменшує ефект перенавчання.

З урахуванням регуляризаторів міра подібності між об'єктами визначається як:

$$sim'(i, j) = \frac{|U_{i,j}|}{|U_{i,j}| + \lambda} \cdot sim(i, j), \quad (2.24)$$

де $|U_{i,j}|$ – кількість розробників, що мають оцінки для обох технологій;

$sim'(i, j)$ – центрований косинус кута.

Якість формування рекомендацій оцінюється за допомогою середньоквадратичної помилки:

$$RMSE = \sqrt{\frac{1}{|K|} \sum_{(u,i) \in K} (r_{u,i} - \hat{r}_{u,i})^2}, \quad (2.25)$$

де $r_{u,i}$ – реальне значення оцінки;

$\hat{r}_{u,i}$ — прогнозоване значення.

За результатами експериментальних досліджень значення параметрів регуляризації було обрано на рівні λ_1 та λ_2 , що забезпечило компроміс між точністю прогнозування та стійкістю моделі.

Таким чином, у роботі запропоновано модель і метод формування команд розробників комп'ютерних систем на основі колаборативної фільтрації з використанням регуляризаторів, що дозволяє підвищити достовірність

рекомендацій і формувати обґрунтовані ранжовані списки кандидатів для реалізації конкретних ІТ-проектів у межах системи підтримки прийняття рішень.

2.5. Висновки до розділу

1. Виконано системний аналіз задачі формування команди для виконання ІТ-проектів у контексті підтримки прийняття рішень, що дало змогу представити процес підбору розробників як багатокритеріальну задачу вибору альтернатив із урахуванням компетентностей фахівців, вимог проекту та обмежень організації-замовника.

2. Формалізовано модель представлення основних сутностей предметної області, зокрема розробників, ІТ-проектів та компаній, із використанням апарату теорії множин і реляційної алгебри, що дало змогу уніфікувати опис різномірних характеристик і забезпечити можливість їх подальшої автоматизованої обробки в системі підтримки прийняття рішень.

3. Розроблено метод оцінювання відповідності між компетентнісними профілями розробників та технологічними вимогами ІТ-проектів на основі метрик подібності, що дало змогу кількісно оцінювати рівень придатності кандидатів і формувати ранжовані списки потенційних учасників команди.

4. Запропоновано метод колаборативної фільтрації з використанням регуляризаторів для зменшення розрідженості та суб'єктивності даних, що дало змогу підвищити стійкість моделі до шумів, забезпечити більш достовірне прогнозування відсутніх оцінок і покращити якість рекомендацій при формуванні команд для виконання ІТ-проектів.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАСОБУ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ ФОРМУВАННІ КОМАНД ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ

3.1. Визначення функціональних вимог до засобу автоматизованого підбору команди розробників комп'ютерних систем

На основі проведеного аналізу предметної області, а також з урахуванням запропонованої математичної моделі та методів підтримки прийняття рішень на базі колаборативної фільтрації, першочерговим етапом проєктування програмного засобу є формування та деталізація його функціональних вимог. Саме функціональні вимоги визначають перелік ключових можливостей системи, логіку взаємодії користувачів із програмним засобом та межі автоматизації процесів формування команди ІТ-проєкту.

Функціональні вимоги дозволяють структуровано описати поведінку системи, визначити ролі користувачів та сценарії їх взаємодії. Для їх формалізації у роботі використано апарат об'єктно-орієнтованого моделювання UML, зокрема діаграми варіантів використання, які наочно відображають основні сценарії роботи із системою.

Для опису функціональних можливостей, доступних розробникам комп'ютерних систем, побудовано діаграму, наведену на рис. 3.1.

Згідно з діаграмою (рис. 3.1), до основних функціональних вимог, реалізованих для розробників, належать:

- реєстрація розробника в системі;
- пошук компаній і проєктів;
- формування та редагування переліку власних знань і навичок;
- надсилання запиту або резюме компанії, що пропонує вакансію.

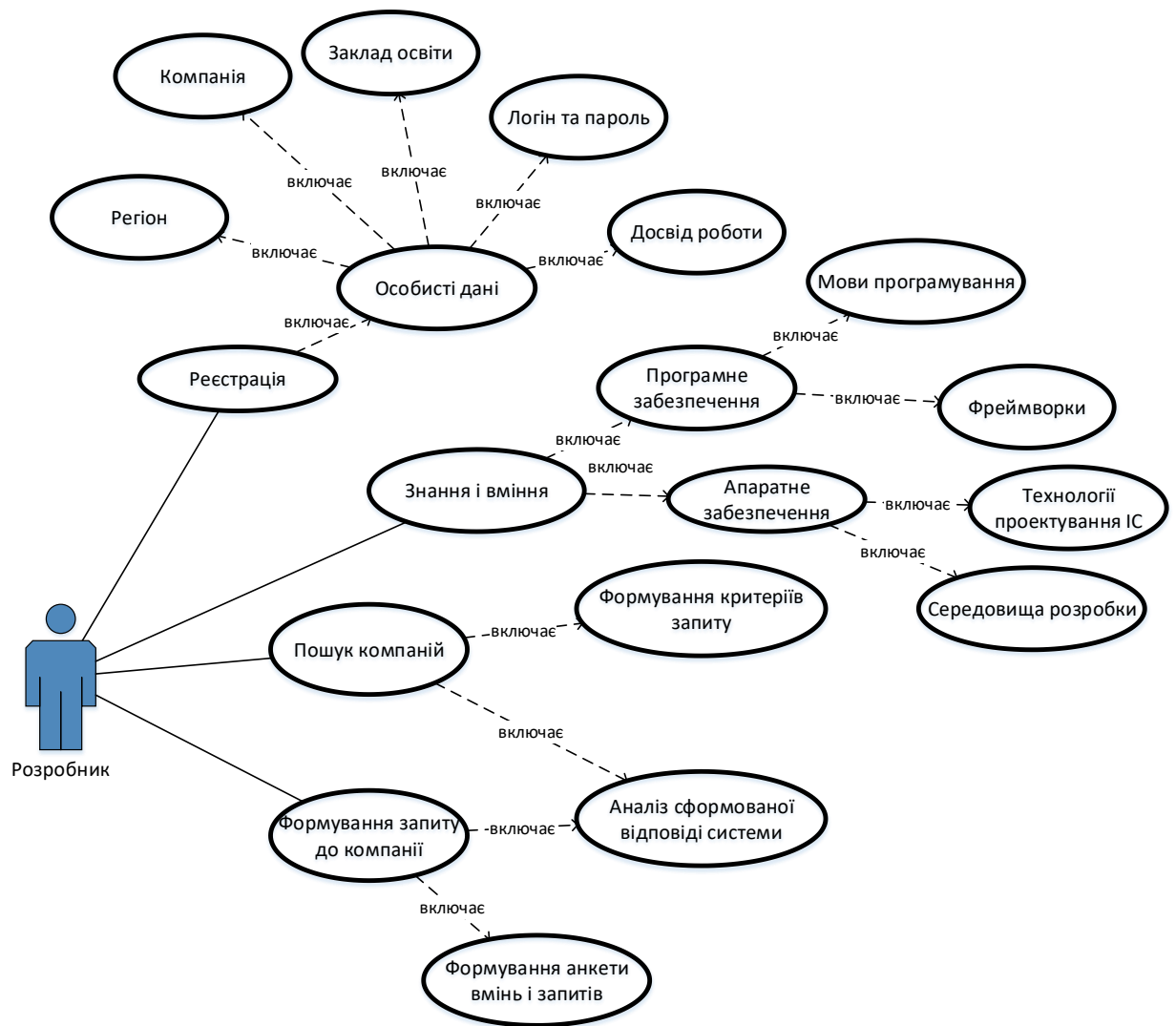


Рисунок 3.1 – Use case діаграма «Розробник»

Процедура реєстрації розробника передбачає обов'язкове введення персональних та професійних даних, зокрема:

- прізвище та ім'я;
- логін і пароль для доступу до системи;
- регіон перебування;
- заклад освіти та спеціальність;
- загальний досвід роботи в ІТ-сфері;
- попереднє місце роботи.

Професійні компетентності розробника у межах програмного засобу поділено на дві логічні групи:

- навички розроблення програмного забезпечення;

- навички проєктування апаратного забезпечення інформаційних систем.

На рис. 3.2 наведено діаграму варіантів використання для іншого типу користувачів – представників компаній, відповідальних за підбір персоналу.

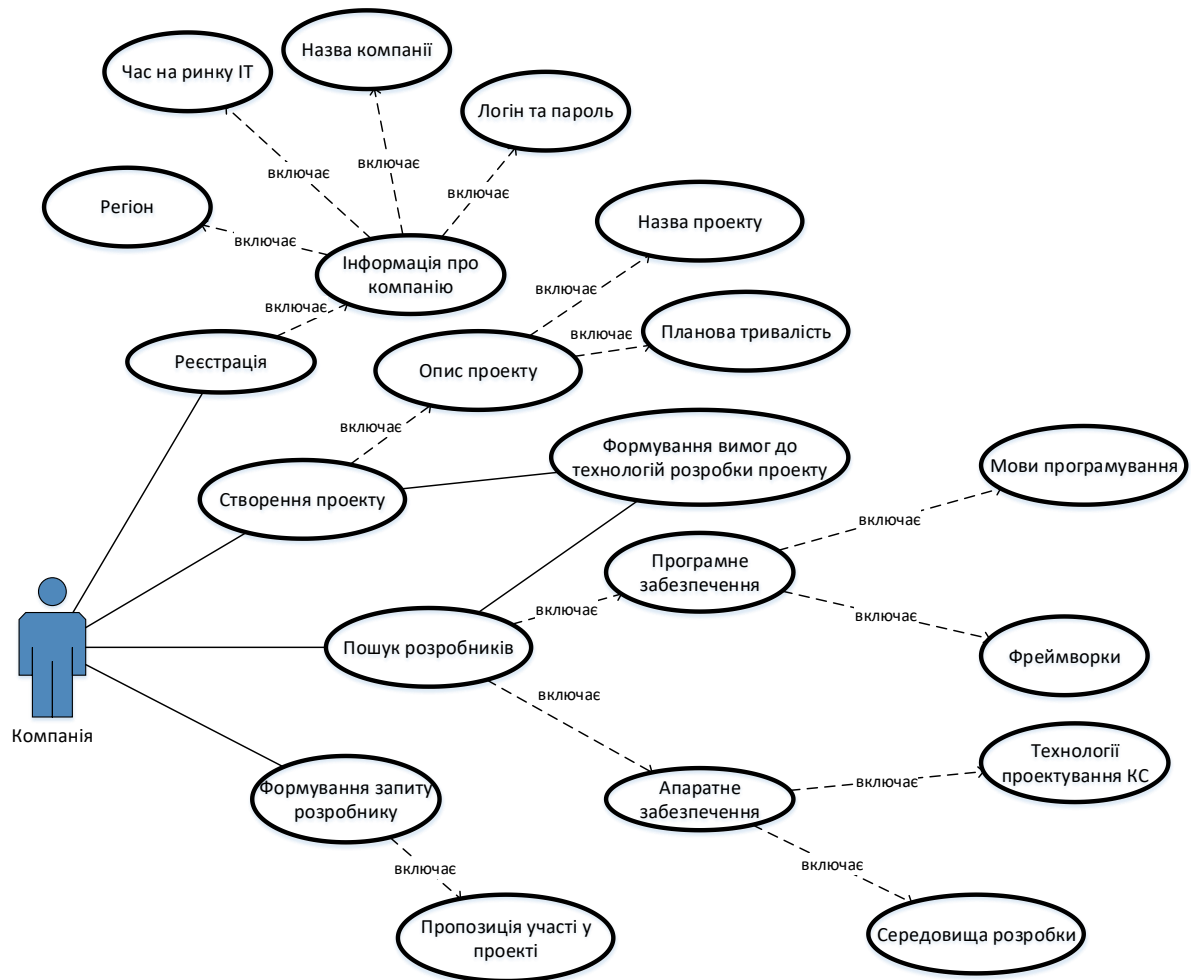


Рисунок 3.2 – Use case діаграма «Компанія»

Основні функціональні можливості представника компанії включають:

- реєстрацію в системі;
- створення та опис ІТ-проєкту;
- формування вакансій;
- пошук розробників;
- надсилання запитів потенційним кандидатам.

При реєстрації компанії обов'язковим є введення таких даних:

- назва компанії;

- регіон розташування;
- облікові дані представника компанії;
- тривалість діяльності на IT-ринку;
- засоби комунікації.

Оскільки підбір персоналу здійснюється під конкретний IT-проект, представник компанії повинен описати проект із зазначенням його назви, тривалості та переліку необхідних компетентностей. Після цього реалізується пошук розробників на основі довідників технологій і отриманих значень подібності, а також формується пропозиція щодо співпраці з урахуванням релевантності результатів.

Таким чином, основними акторами програмного засобу є розробники інформаційних систем і компанії-замовники. Після визначення та аналізу їх функціональних вимог наступним етапом є проектування структури бази даних.

3.2. Моделювання та реалізація бази даних системи підтримки прийняття рішень щодо підбору команди розробників

Аналіз предметної області та визначених функціональних вимог дозволив виокремити основні сутності та їх атрибути, які були покладені в основу реляційної моделі бази даних програмного засобу [17,18]. До ключових сутностей належать:

- розробник комп'ютерної системи;
- компанія-замовник;
- мови програмування;
- фреймворки;
- технології проектування комп'ютерних систем;
- середовища проектування апаратного забезпечення;
- заклади освіти;
- параметри IT-проекту.

Частина зазначених сутностей є довідниковими (мови програмування, фреймворки, технології, середовища розробки, заклади освіти), однак між ними існують складні зв'язки типу «багато-до-багатьох», які враховуються при нормалізації бази даних і відображаються на ER-діаграмі.

У табл. 3.1 наведено опис сутності «Мови програмування» та її атрибутів.

Таблиця 3.1 – Сутність «ProgrammingLanguage»

Назва атрибуту	Тип	Примітка
ID_PL	int	PK
PLName	varchar(100)	
Description	varchar(max)	

Проектування бази даних виконано з використанням реляційного підходу та мови SQL у середовищі Microsoft SQL Management Studio, що відповідає стандарту T-SQL і забезпечує зручні засоби адміністрування та візуалізації структури даних..

Реалізація SQL-запиту при створенні таблиці «ProgrammingLanguage» представлена на рис. 3.3, а результат – на рис. 3.4.

```
CREATE TABLE ProgrammingLanguage (
  ID_PL int primary key identity (1,1),
  PLName varchar (100),
  [Description] varchar(max)
)
```

Рисунок 3.3 – SQL-запит створення таблиці «ProgrammingLanguage»

	Column Name	Data Type	Allow Nulls
	ID_PL	int	☐
	PLName	varchar(100)	☒
	Description	varchar(MAX)	☒

Рисунок 3.4 – Результат виконання запиту

Сутність «Frameworks» з визначеними властивостями проілюстрована у табл. 3.2.

Таблиця 3.2 – Сутність «Frameworks»

Назва атрибуту	Тип	Примітка
ID_FR	int	PK
FrName	varchar(100)	
Description	varchar(max)	

SQL-скрипт, що реалізує генерацію таблиці «Frameworks» проілюстровано на рис. 3.5.

```
CREATE TABLE Frameworks (
  ID_FR int primary key identity (1,1),
  FrName varchar (100),
  [Description] varchar(max)
)
```

Рисунок 3.5 – Скрипт генерації «Frameworks»

У результаті виконання вищенаведеного скрипта, одержано таблицю БД, яка показана на рис. 3.6.

	Column Name	Data Type	Allow Nulls
🔑	ID_FR	int	☐
	FrName	varchar(100)	☒
	Description	varchar(MAX)	☒

Рисунок 3.6 – Результат створення таблиці «Frameworks»

При аналізі предметної області визначено сутність «Technologies» та її атрибути, які наведено у табл. 3.3.

Таблиця 3.3 – Сутність «Technologies»

Назва атрибуту	Тип	Примітка
ID_T	int	PK
TName	varchar(100)	
Description	varchar(max)	

Програмна реалізація сутності «Technologies» показана на рис. 3.7, а одержаний результат – на рис. 3.8.

```
CREATE TABLE Technologies (
  ID_T int primary key identity (1,1),
  TName varchar (100),
  [Description] varchar(max)
)
```

Рисунок 3.7 – Скрипт генерації таблиці «Technologies»

	Column Name	Data Type	Allow Nulls
	ID_T	int	☐
	TName	varchar(100)	☒
	Description	varchar(MAX)	☒

Рисунок 3.8 – Результат створення таблиці «Technologies»

Опис сутності «Environments», що відповідає за навички володіння середовищами проектування інформаційних систем представлена у табл. 3.4.

Таблиця 3.4 – Сутність «Environments»

Назва атрибуту	Тип	Примітка
ID_En	int	PK
EnName	varchar(100)	
Description	varchar(max)	

Практична імплементація таблиці «Environments» показана на рис. 3.9 з відповідним результатом на рис. 3.10.

```
CREATE TABLE Environments (
  ID_En int primary key identity (1,1),
  EnName varchar (100),
  [Description] varchar(max)
)
```

Рисунок 3.9 – SQL-запит створення таблиці «Environments»

	Column Name	Data Type	Allow Nulls
▶	ID_En	int	☐
	EnName	varchar(100)	☒
	Description	varchar(MAX)	☒

Рисунок 3.10 – Створена таблиця «Environments»

Для реалізації зв'язків типу «багато-до-багатьох» між мовами програмування і фреймворками, а також між технологіями проєктування і середовищами розробки, створено проміжні таблиці PrLanguage_Frameworks та Tecnologies_Environments (табл. 3.5, 3.6).

Таблиця 3.5 – «PrLanguage_Frameworks»

Назва атрибуту	Тип	Примітка
ID_PL_FR	int	PK
ID_PL	int	FK
ID_FR	int	FK
Description	varchar(max)	

Таблиця 3.6 – «Technologies_Environments»

Назва атрибуту	Тип	Примітка
ID_T_En	int	PK
ID_T	int	FK
ID_En	int	FK
Description	varchar(max)	

Програмна реалізація засобами мови SQL таблиць «PrLanguage_Frameworks» та «Technologies_Environments» проілюстровано на рис. 3.11.

```

CREATE TABLE PrLanguage_Frameworks (
  ID_PL_FR int primary key identity (1,1),
  ID_PL int foreign key references ProgrammingLanguage(ID_PL),
  ID_FR int foreign key references Frameworks(ID_FR),
  [Description] varchar(max)
)
CREATE TABLE Technologies_Environments (
  ID_T_En int primary key identity (1,1),
  ID_T int foreign key references Technologies(ID_T),
  ID_En int foreign key references Environments(ID_En),
  [Description] varchar(max)
)

```

Рисунок 3.11 – Скрипти генерації таблиць «PrLanguage_Frameworks» і «Technologies_Environments»

Структура програмно реалізованої таблиці PrLanguage_Frameworks показана на рис. 3.12.

	Column Name	Data Type	Allow Nulls
	ID_PL_FR	int	☐
	ID_PL	int	☒
	ID_FR	int	☒
	Description	varchar(MAX)	☒

Рисунок 3.12 – Результат створення таблиці «PrLanguage_Frameworks»

Представлення таблиці «Technologies_Environments» наведено на рис. 3.13.

	Column Name	Data Type	Allow Nulls
	ID_T_En	int	☐
	ID_T	int	☒
	ID_En	int	☒
	Description	varchar(MAX)	☒

Рисунок 3.13 – Результат створення таблиці «Technologies_Environments»

Властивості сутності «School» визначено та продемонстровано у табл. 3.7.

Таблиця 3.7 – «School»

Назва атрибуту	Тип	Примітка
ID_Sch	int	PK
SchName	varchar (100)	
Location	varchar (150)	
Reyting	int	
Description	varchar(max)	

Для генерації таблиці «School» реалізовано відповідний скрипт, результат роботи якого представлено на рис. 3.14.

	Column Name	Data Type	Allow Nulls
	ID_Sch	int	☐
	SchName	varchar(100)	☒
	Location	varchar(150)	☒
	Reyting	int	☒
	Description	varchar(MAX)	☒

Рисунок 3.14 – Результат генерації таблиці «School»

Опис атрибутів компанії, яка виступає в якості замовника проєкту, проілюстровано у 3.8.

Таблиця 3.8 – «Company»

Назва атрибуту	Тип	Примітка
ID_Sch	int	PK
SchName	varchar (100)	
Location	varchar (150)	
Reyting	int	
Description	varchar(max)	

Програмна реалізація запиту на створення «Company» показана на рис.3.15, з результат створення наведений на рис. 3.16.

```
CREATE TABLE Company (
  ID_Company int primary key identity (1,1),
  CompanyName varchar (100),
  Location varchar (150),
  [Description] varchar(max),
)
```

Рисунок 3.15 – Програмний код реалізації «Company»

	Column Name	Data Type	Allow Nulls
	ID_Company	int	☐
	CompanyName	varchar(100)	☒
	Location	varchar(150)	☒
	Description	varchar(MAX)	☒

Рисунок 3.16 – Результат реалізації «Company»

За наявності в програмному засобі відомостей про компанію виникає потреба реалізувати механізм створення та опису відповідного ІТ-проєкту. З цією метою визначено структуру проєкту, яка наведена в табл. 3.9.

Таблиця 3.9 – «Project»

Назва атрибуту	Тип	Примітка
ID_Project	int	PK
ID_Company	Int	FK
ProjectName	varchar (100)	
Duration	Int	
Description	varchar(max)	

Таблиця «Project» імплементована із застосуванням SQL-запиту, представленого на рис. 3.17. Результат його виконання показаний на рис. 3.18.

```
CREATE TABLE Project (
  ID_Project int primary key identity (1,1),
  ID_Company int foreign key references Company(ID_Company),
  CompanyName varchar (100),
  Location varchar (150),
  [Description] varchar(max),
)
```

Рисунок 3.17 – Запит на створення «Project»

🔑	ID_Project	int	☐
	ID_Company	int	☒
	ProjectName	varchar(100)	☒
	Duration	varchar(150)	☒
	Description	varchar(MAX)	☒

Рисунок 3.18 – Результат виконання запиту

Для формалізованого подання технологій та інструментальних засобів, що планується використовувати під час реалізації ІТ-проєкту, доцільно виокремити сутність «ProjectRequirements». Перелік атрибутів цієї сутності наведено в табл. 3.10.

Таблиця 3.10 – «ProjectRequirements»

Назва атрибуту	Тип	Примітка
ID_ProjReq	int	PK
ID_Proj	int	FK
ID_PL	int	FK
ID_FR	int	FK
ID_T	int	FK
ID_En	int	FK
ID_T_En	int	FK
ID_PL_En	int	FK
Description	varchar(max)	

Програмний код реалізації опису ІТ-проєкту показано на рис. 3.19, а одержаний результат на наступному рисунку (рис. 3.20).

```
CREATE TABLE ProjectRequirements (
  ID_ProjReq int primary key identity (1,1),
  ID_Proj int foreign key references Project(ID_Project),
  ID_PL int foreign key references ProgrammingLanguage(ID_PL),
  ID_FR int foreign key references Frameworks(ID_Fr),
  ID_T int foreign key references Technologies(ID_T),
  ID_En int foreign key references Environments(ID_En),
  ID_PL_FR int foreign key references PrLanguage_Frameworks(ID_PL_FR),
  ID_T_En int foreign key references Tecnologies_Environments(ID_T_En),
  [Description] varchar(max),
)
```

Рисунок 3.19 – Скрипт генерування таблиці «ProjectRequirements»

	Column Name	Data Type	Allow Nulls
PK	ID_ProjReq	int	☐
	ID_Proj	int	☒
	ID_PL	int	☒
	ID_FR	int	☒
	ID_T	int	☒
	ID_En	int	☒
	ID_PL_FR	int	☒
	ID_T_En	int	☒
	Description	varchar(MAX)	☒

Рисунок 3.20 – Результат створення таблиці ProjectRequirements

З метою зберігання відомостей про розробників, зокрема їх персональної інформації без урахування професійних знань і навичок у сфері ІТ, у базі даних передбачено таблицю «Developer», структура якої наведена в табл. 3.11.

Таблиця 3.11 – «Developer»

Назва атрибуту	Тип	Примітка
ID_Dev	int	PK
FirstName	varchar(100)	
LastName	varchar(100)	
Login	varchar(50)	
Password	varchar(50)	
Experience	int	
Company_ID	int	FK
School_ID	int	FK
Location	varchar(100)	
Description	varchar(max)	

Скрипт генерації таблиці «Developer» показано на рис. 3.21.

```

CREATE TABLE Developer (
  ID_Dev int primary key identity (1,1),
  FirstName varchar (100),
  LastName varchar (100),
  [Login] varchar (50),
  [Password] varchar (50),
  Experience int,
  Company_ID int foreign key references Company(ID_Company),
  School_ID int foreign key references School(ID_Sch),
  Location varchar (100),
  [Description] varchar(max)
)

```

Рисунок 3.21 – Скрипт створення таблиці «Розробник»

Отриманий результат виконання скрипта 3.21 проілюстровано на рис. 3.22.

	Column Name	Data Type	Allow Nulls
►	ID_Dev	int	☐
	FirstName	varchar(100)	☒
	LastName	varchar(100)	☒
	Login	varchar(50)	☒
	Password	varchar(50)	☒
	Experience	int	☒
	Company_ID	int	☒
	School_ID	int	☒
	Location	varchar(100)	☒
	Description	varchar(MAX)	☒

Рисунок 3.22 – «Developer»

Для відтворення зв'язку між сутностями, які описують розробника і його професійні навички необхідно реалізувати таблицю, яка міститиме таку інформацію. Структуру такої таблиці під назвою «DeveloperRequirements» показано у табл. 3.12.

Таблиця 3.12 – «DeveloperRequirements»

Назва атрибуту	Тип	Примітка
ID_DevReq	int	PK
ID_Dev	int	FK
ID_PL	int	FK
ID_FR	int	FK
ID_T	int	FK
ID_En	int	FK
ID_T_En	int	FK
ID_PL_En	int	FK
Description	varchar(max)	

SQL-скрипт імплементації сутності щодо вмінь розробників та одержаний результат представлено відповідно на рис. 3.23 та рис. 3.24.

```
CREATE TABLE DeveloperRequirements (
  ID_DevReq int primary key identity (1,1),
  ID_Dev int foreign key references Project(ID_Project),
  ID_PL int foreign key references ProgrammingLanguage(ID_PL),
  ID_FR int foreign key references Frameworks(ID_Fr),
  ID_T int foreign key references Technologies(ID_T),
  ID_En int foreign key references Environments(ID_En),
  ID_PL_FR int foreign key references PrLanguage_Frameworks(ID_PL_FR),
  ID_T_En int foreign key references Tecnologies_Environments(ID_T_En),
  [Description] varchar(max),
)
```

Рисунок 3.23 – SQL-запит створення таблиці «DeveloperRequirements»

	Column Name	Data Type	Allow Nulls
🔍	ID_DevReq	int	☐
	ID_Dev	int	☒
	ID_PL	int	☒
	ID_FR	int	☒
	ID_T	int	☒
	ID_En	int	☒
	ID_PL_FR	int	☒
	ID_T_En	int	☒
	Description	varchar(MAX)	☒

Рисунок 3.24 – Результат запиту при створенні «DeveloperRequirements»

Завершальним етапом проєктування бази даних стало формування ER-діаграми, наведеної на рис. 3.25, яка відображає всі сутності та зв'язки між ними у межах предметної області.

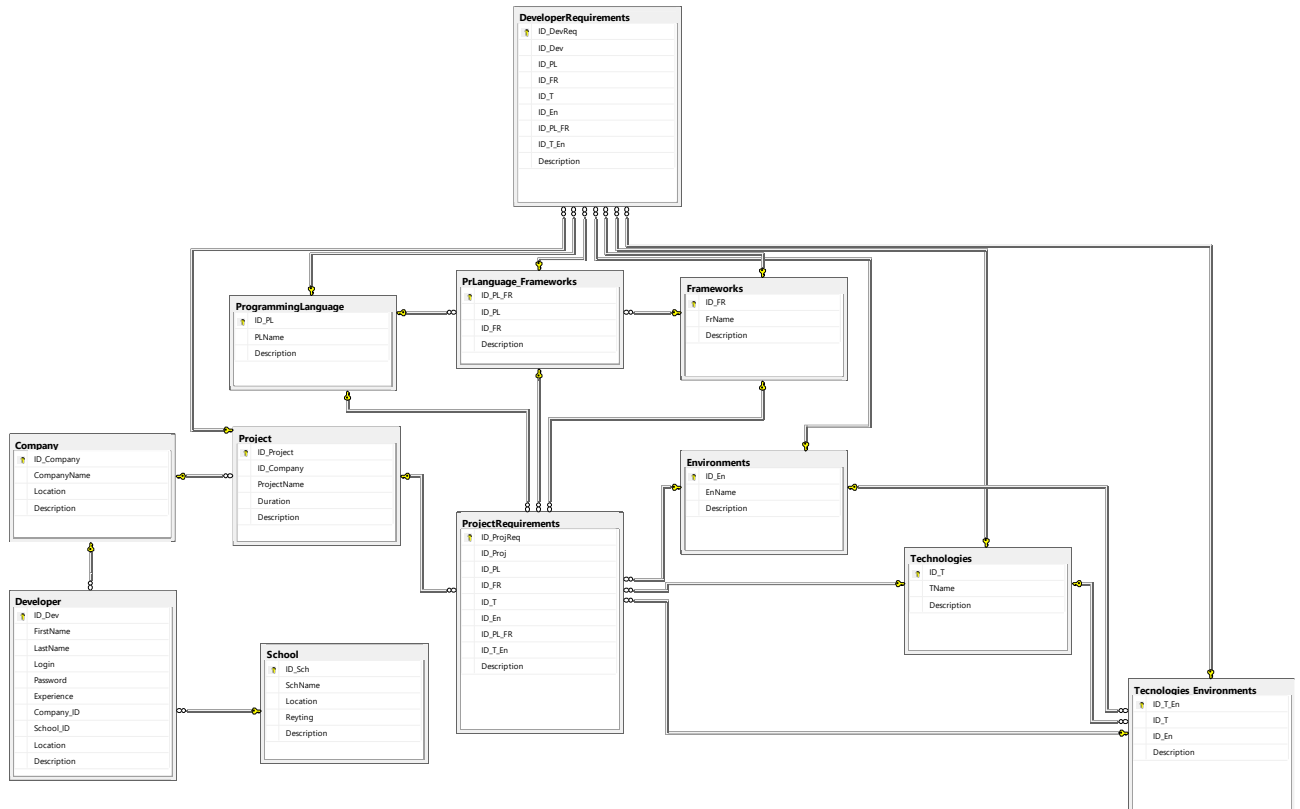


Рисунок 3.25 – ER-діаграма бази даних програмного засобу

Таким чином, у результаті аналізу предметної області, визначення функціональних вимог та врахування особливостей моделей оцінювання подібності компетентностей розробників до вимог ІТ-проєктів, розроблено реляційну схему бази даних із дотриманням вимог нормалізації та цілісності. Наступним етапом є проєктування архітектури програмного засобу та реалізація алгоритмів його функціонування.

3.3. Побудова моделі архітектури програмного засобу підтримки прийняття рішень

Архітектура програмного забезпечення може бути визначена як узагальнена модель, що відображає склад системи, її основні структурні елементи та взаємозв'язки між ними. Для опису архітектурних рішень використовуються різні графічні нотації, серед яких найбільш поширеними є:

- діаграми класів;
- діаграми пакетів класів;
- діаграми компонентів;
- багаторівнева (шарова) архітектура за Фаулером.

Діаграми класів застосовуються для деталізованого подання структури програмного забезпечення на рівні окремих програмних модулів. Вони відображають атрибути, методи та параметри класів, що дозволяє розробникам зрозуміти принципи реалізації базових функціональних можливостей системи.

Діаграми пакетів класів забезпечують більш узагальнене уявлення про архітектуру, групуючи взаємопов'язані класи у логічні пакети. Такий підхід дозволяє представити каркас програмного засобу на абстрактному рівні та є корисним на етапах проєктування, інтеграції компонентів і проведення інтеграційного тестування.

Діаграми компонентів використовуються для опису архітектури на рівні підсистем. Вони можуть включати в себе як діаграми пакетів, так і діаграми класів, і застосовуються на початкових етапах архітектурного проєктування або під час системного тестування, коли необхідно оцінити цілісність та ефективність програмної системи в цілому.

Одним із найбільш поширених архітектурних підходів є шарова архітектура за Фаулером, яка передбачає розподіл програмного забезпечення на три базові рівні:

- рівень представлення;
- рівень бізнес-логіки;

- рівень доступу до даних.

Рівень представлення охоплює компоненти, відповідальні за взаємодію з кінцевими користувачами та реалізацію користувацького інтерфейсу. Саме на цьому рівні здійснюється введення даних і відображення результатів роботи системи.

Рівень бізнес-логіки реалізує основну функціональність програмного засобу. Він відповідає за обробку запитів, реалізацію алгоритмів, координацію взаємодії між компонентами та забезпечує зв'язок між рівнем представлення і рівнем даних.

Рівень даних включає компоненти доступу до бази даних та саму базу даних, забезпечуючи збереження, оновлення і вибірку інформації.

Таке багаторівневе подання архітектури дозволяє чітко розмежувати відповідальність між компонентами, підвищити масштабованість і спростити супровід програмного забезпечення. З огляду на це, для опису архітектури програмного засобу автоматизованого підбору команди розробників комп'ютерних систем у роботі використано поєднання шарової архітектури за Фаулером і діаграм пакетів класів. Архітектурну модель розроблюваної системи наведено на рис. 3.26.

Рівні представлення у системі реалізовано за допомогою трьох основних пакетів класів:

- інтерфейс авторизації;
- інтерфейс розробника комп'ютерних систем;
- інтерфейс представника компанії.

Інтерфейс авторизації є спільним для всіх категорій користувачів і забезпечує виконання процедур реєстрації та входу до системи шляхом перевірки облікових даних.

Інтерфейс розробника надає доступ до функціональних можливостей програмного засобу, визначених у підрозділі 3.1 та відображених на рис. 3.1, зокрема до формування профілю, пошуку компаній і надсилання запитів.

Інтерфейс представника компанії реалізує функціональні вимоги, наведені на рис. 3.2, і дозволяє здійснювати опис проєктів, формування вакансій, пошук кандидатів та ініціювання взаємодії з розробниками.

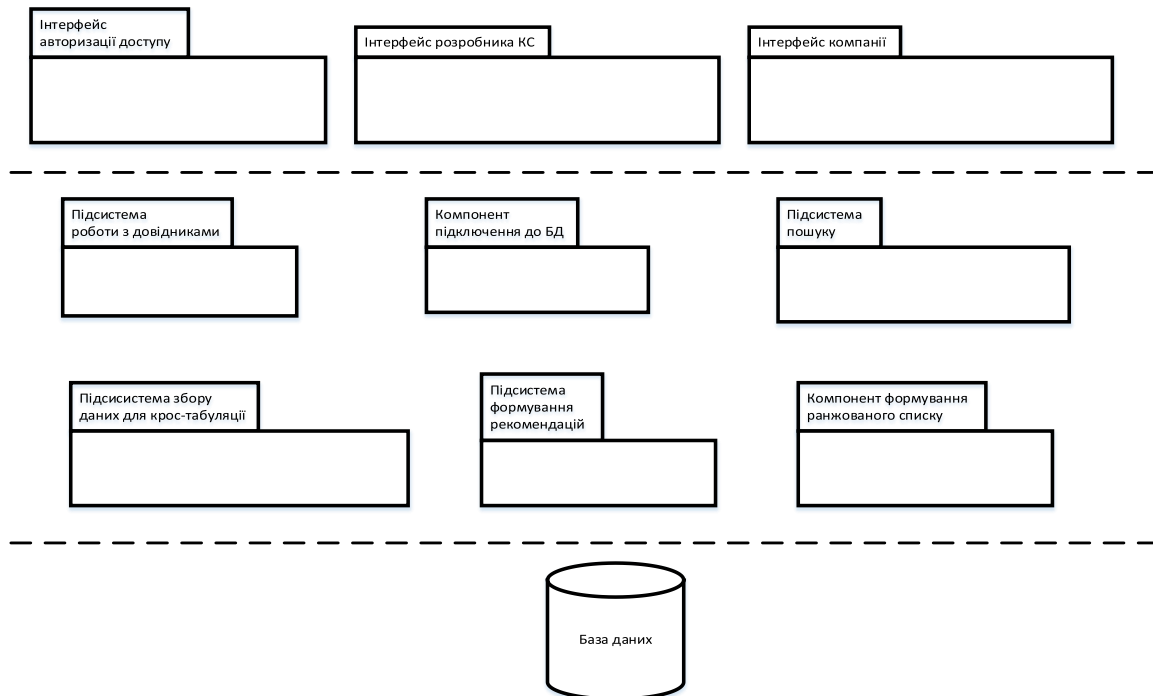


Рисунок 3.26 – Архітектура програмного засобу підтримки прийняття рішень

На рівні бізнес-логіки програмного засобу автоматизованого підбору команди визначено такі основні компоненти:

- підсистема керування довідниками;
- компонент взаємодії з базою даних;
- підсистема пошуку;
- підсистема збору даних для формування крос-табуляцій;
- підсистема формування рекомендацій;
- компонент побудови ранжованих списків.

Підсистема роботи з довідниками відповідає за обробку даних, що надходять із рівня представлення, та їх збереження у відповідних таблицях бази даних. Вона забезпечує підтримку актуальності довідників мов програмування, фреймворків, технологій проєктування та середовищ розробки.

Компонент підключення до бази даних реалізує контроль доступу до СУБД, відстежує стан з'єднання та забезпечує журналювання дій користувачів.

Підсистема пошуку виконує зв'язуючу роль між інтерфейсами користувачів і даними, що зберігаються в базі даних, надаючи можливість здійснювати пошук за визначеними критеріями, перелік яких відповідає сценаріям, наведеним на рис. 3.1 і рис. 3.2.

Підсистема збору даних для крос-табуляції забезпечує підготовку та агрегацію інформації, необхідної для подальшого обчислення подібності між компетентностями розробників і вимогами ІТ-проектів.

Підсистема формування рекомендацій реалізує алгоритми оцінювання подібності мов програмування, фреймворків, технологій і середовищ, формує відповідні вектори та визначає рівень відповідності кандидатів проектним вимогам.

Компонент формування ранжованого списку отримує результати з підсистеми рекомендацій, виконує їх обробку та передає узагальнену інформацію до відповідного користувацького інтерфейсу – розробника або представника компанії.

У результаті архітектурного проектування програмного засобу підтримки прийняття рішень при формуванні команди для виконання ІТ-проектів визначено ключові модулі, їх функціональне призначення та взаємодію. Реалізацію програмного сервісу виконано з використанням мови програмування C#, технології ASP.NET MVC та системи керування базами даних MS SQL Server, що забезпечує масштабованість, розширюваність і зручність подальшого супроводу системи.

3.4. Результати апробації запропонованого методу і засобу

Для наочного представлення та оцінювання ефективності результатів автоматизованого підбору команди розробників комп'ютерних систем було використано програмну утиліту GraphLab Canvas.

На рис. 3.27 продемонстровано процес формування та візуалізації даних на основі двох фреймів даних – «item_sim_recomm» та «popularity_recomm».

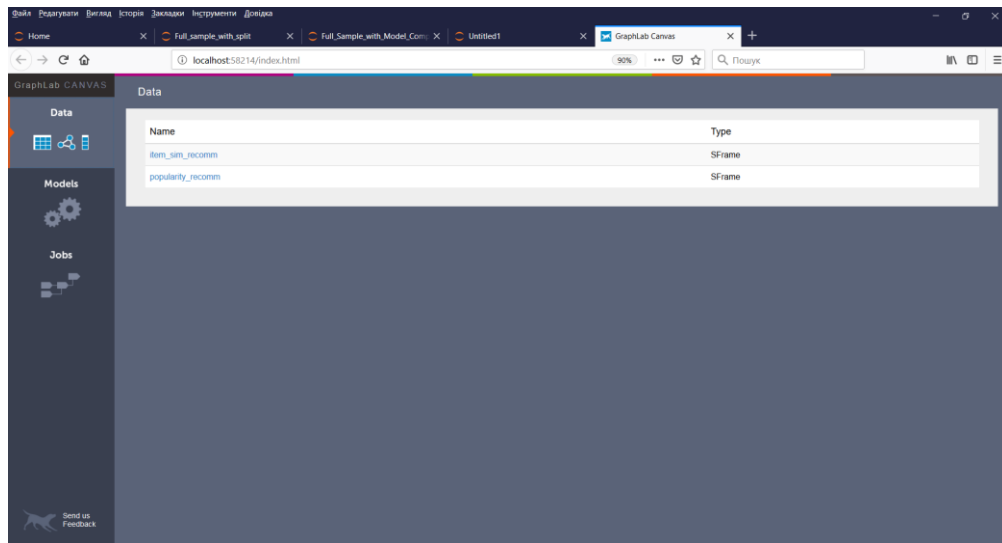


Рисунок 3.27 – Генерація даних у GraphLab Canvas

Застосування GraphLab Canvas дозволяє виконати детальний огляд структури та характеристик обраного фрейму даних. Зокрема, на рис. 3.28 наведено узагальнену статистичну інформацію для фрейму item_sim_recomm.

Подана статистика відображає відомості про розробників комп'ютерних систем за їх ідентифікаторами, включаючи кількість унікальних записів, максимальні та середні значення ідентифікаторів, а також характер розподілу даних у фреймі. Аналогічні показники формуються і для технологій розробки, що представлені в наборі даних.

Окрім цього, засобами Canvas виконано візуалізацію найбільш поширених мов програмування, фреймворків та інших технологічних елементів. На вкладці Summary зосереджено узагальнену інформацію щодо найбільш популярних технологій та їх відносного рейтингу.

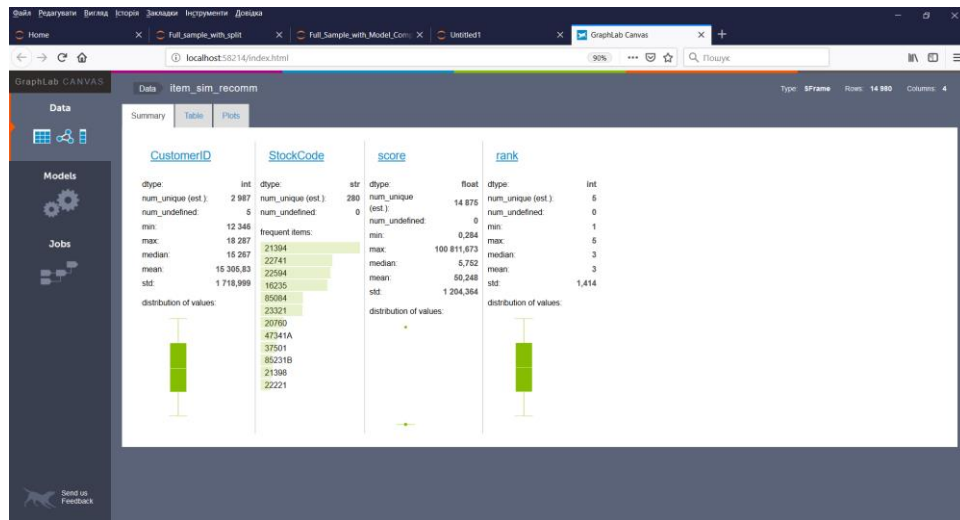


Рисунок 3.28 – Загальна статистика у фреймі даних *item_sim_recomm*

Табличне подання статистичних характеристик фрейму *item_sim_recomm* представлено на рис. 3.29, що дає змогу детальніше проаналізувати структуру даних та окремі записи.

row #	CustomerID	StockCode	score	rank
0	17 850	21394	7,154	1
1	17 850	16235	3,7	2
2	17 850	23321	3,292	3
3	17 850	21398	2,739	4
4	17 850	20760	2,663	5
5	13 047	21394	15,366	1
6	13 047	22594	6,685	2
7	13 047	23321	4,97	3
8	13 047	16235	4,756	4
9	13 047	21398	3,314	5
10	12 583	23321	27,937	1
11	12 583	17091A	20,518	2
12	12 583	16235	18,721	3
13	12 583	22594	15,915	4

Рисунок 3.29 – Табличний вигляд фрейму даних «Item_sim_recomm»

На рис. 3.30 наведено графічну інтерпретацію подібності між технологіями проектування комп'ютерних систем, сформовану на основі частоти їх появи у відповідному фреймі даних. Таке подання дозволяє виявити групи близьких за використанням технологій і оцінити їх взаємозв'язки.

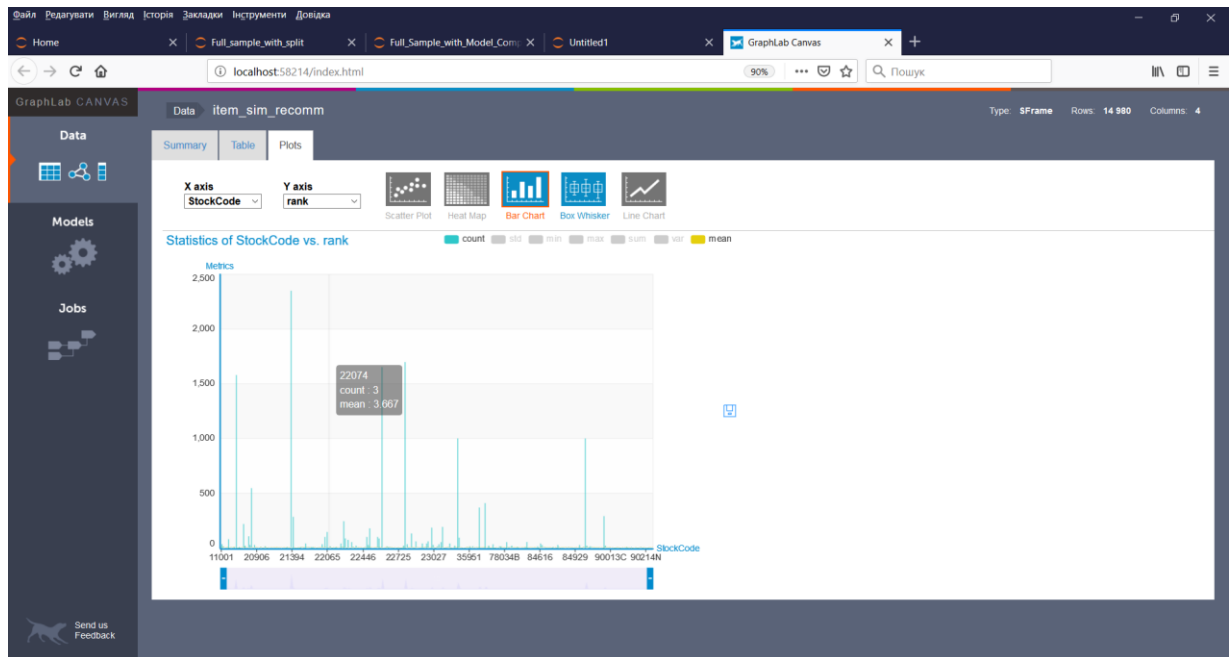


Рисунок 3.30 – Графік подібності елементів у фреймі «Item_sim_recomm»

Статистичні характеристики та опис моделі Popularity_model, яка використовується для побудови рекомендацій на основі популярності елементів, наведено на рис. 3.31.

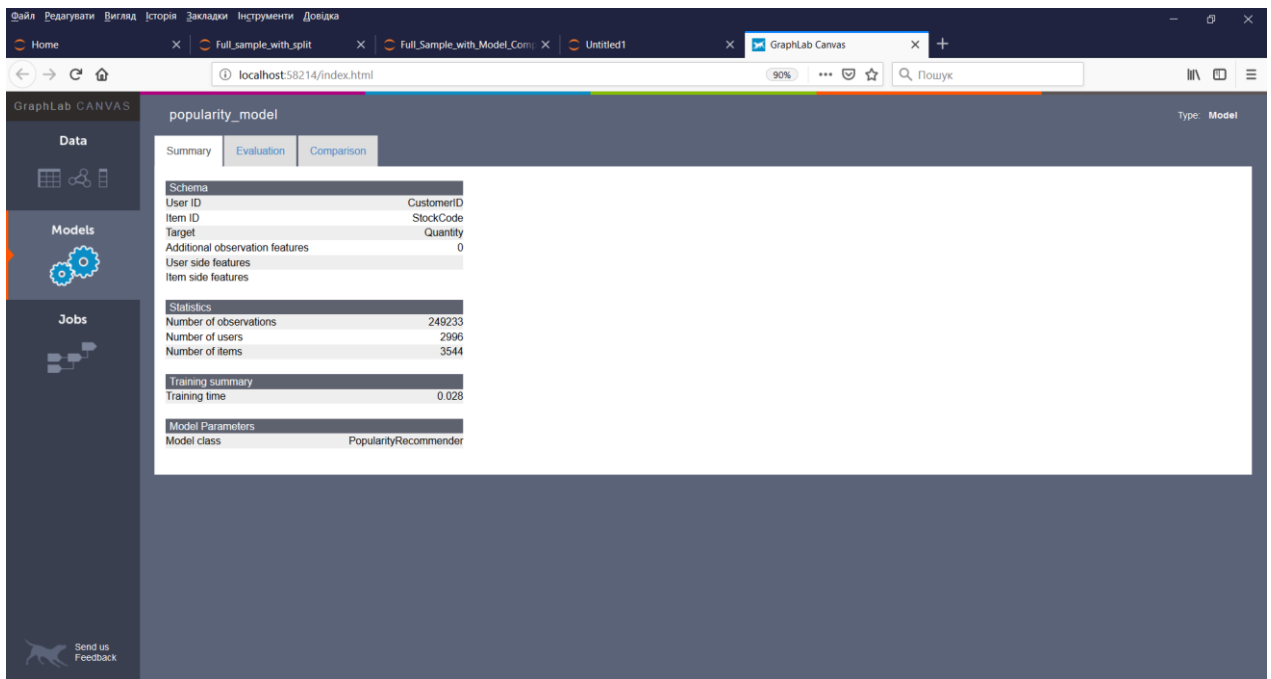


Рисунок 3.31 – Статистичні дані про модель «Popularity_model»

У результаті проведених експериментальних досліджень встановлено, що точність формування рекомендацій відповідно до запропонованої моделі та реалізованого методу становить 84,65 %. Отримане значення свідчить про достатній рівень ефективності розробленого підходу та підтверджує доцільність його використання у задачах підтримки прийняття рішень при підборі команд для виконання ІТ-проектів.

3.5. Висновки до розділу

У даному розділі одержано наступні практичні результати:

1. Визначено та формалізовано функціональні вимоги до програмного засобу підтримки прийняття рішень при підборі команд для виконання ІТ-проектів, що дало змогу чітко окреслити ролі користувачів, сценарії їх взаємодії з системою та межі автоматизації процесів підтримки прийняття рішень.

2. Спроектовано та реалізовано реляційну базу даних програмного засобу з урахуванням вимог нормалізації та цілісності, що дало змогу забезпечити структуроване зберігання даних про розробників, проекти, технології та їх взаємозв'язки, а також створити інформаційну основу для реалізації рекомендаційних алгоритмів.

3. Розроблено архітектуру програмного засобу на основі моделі Фаулера, що дало змогу розмежувати рівні представлення, бізнес-логіки та доступу до даних, підвищити масштабованість системи та спростити її подальший супровід і розширення.

4. Проведено експериментальну апробацію запропонованого методу та програмного засобу з використанням інструментів аналітичної візуалізації, що дало змогу підтвердити ефективність реалізованої технології підтримки прийняття рішень і досягти точності формування рекомендацій на рівні 84,65 %, достатньому для практичного використання у задачах підбору команд для виконання ІТ-проектів.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Розроблення та використання технології підтримки прийняття рішень при формуванні команд для виконання ІТ-проектів пов'язані з інтенсивною інформаційно-аналітичною діяльністю, роботою з комп'ютерною технікою, програмними засобами обробки даних і візуалізації результатів. У процесі виконання кваліфікаційної роботи основна діяльність фахівця зосереджена на аналізі інформації, моделюванні, інтерпретації результатів рекомендаційних алгоритмів та прийнятті управлінських рішень у цифровому середовищі. За таких умов особливого значення набуває дотримання вимог охорони праці, техніки та пожежної безпеки, спрямованих на збереження здоров'я та працездатності виконавців [28].

Охорона праці в галузі інформаційних технологій і системного аналізу розглядається як комплекс правових, організаційних, технічних і санітарно-гігієнічних заходів, що забезпечують безпечні та нешкідливі умови праці під час експлуатації електронно-обчислювальної техніки та засобів відображення інформації. Державна політика України у сфері охорони праці базується на пріоритеті життя і здоров'я людини та закріплена в Конституції України, Законі України «Про охорону праці», Кодексі законів про працю України, а також у спеціальних нормативно-правових актах, що регламентують умови праці з екранними пристроями та електричним обладнанням [28].

Під час виконання робіт, пов'язаних із розробленням і апробацією технології підтримки прийняття рішень, використовуються персональні комп'ютери, серверні ресурси, мережеве обладнання та периферійні пристрої. Відповідно до вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», організація робочого місця повинна забезпечувати достатню площу та об'єм приміщення для

одного працівника, що сприяє зниженню психофізіологічного навантаження та створенню комфортних умов праці. Робочі приміщення мають відповідати санітарно-гігієнічним нормам, а матеріали внутрішнього оздоблення повинні бути дозволені органами державного санітарного нагляду [29].

Особлива увага приділяється електробезпеці. Усі елементи, які можуть перебувати під напругою або мають заземлення, зокрема труби опалення, водопровідні комунікації та кабелі з відкритими екранами, повинні бути надійно ізольовані. Електричне обладнання має експлуатуватися відповідно до вимог чинних правил, а в приміщеннях з інтенсивним використанням комп'ютерної техніки повинні бути передбачені аварійні вимикачі для оперативного знеструмлення.

Організація робочого місця фахівця з аналізу даних або менеджера ІТ-проектів повинна відповідати ергономічним вимогам. Правильне розташування монітора, клавіатури, маніпуляторів введення та робочого столу зменшує ризик виникнення захворювань опорно-рухового апарату та зорового перенапруження. Важливим є також регулювання режимів праці та відпочинку, оскільки тривала робота з інформаційно-аналітичними системами супроводжується підвищеним навантаженням на зір і нервову систему, що може негативно впливати на якість прийнятих рішень [29].

Освітлення робочих приміщень має відповідати вимогам ДБН В.2.5-28:2018 «Природне і штучне освітлення». Недостатнє або надмірне освітлення, а також наявність прямих чи відбитих відблисків на екранах моніторів призводять до швидкої втоми та зниження концентрації уваги. Для усунення негативного впливу сонячного світла вікна повинні бути обладнані жалюзі або шторами, а штучне освітлення — рівномірно розподілене в межах робочої зони [29].

Під час експлуатації систем підтримки прийняття рішень необхідно враховувати вплив шкідливих і небезпечних факторів виробничого середовища, таких як шум, вібрація, електромагнітне та електростатичне випромінювання. Робочі приміщення повинні бути забезпечені належною вентиляцією для підтримання оптимальних параметрів мікроклімату та якості повітря, що є

важливою умовою збереження працездатності працівників, зайнятих інтелектуальною працею.

Важливим аспектом забезпечення безпеки життєдіяльності є пожежна безпека. Приміщення, у яких розміщується комп'ютерна та офісна техніка, повинні відповідати вимогам ДБН В.1.1-7-2016 «Пожежна безпека об'єктів будівництва» та Правил пожежної безпеки в Україні. Суб'єкти господарювання зобов'язані забезпечити наявність справних вогнегасників, їх своєчасне технічне обслуговування та правильне розміщення з урахуванням шляхів евакуації. Персонал має бути ознайомлений з правилами пожежної безпеки та порядком дій у разі виникнення надзвичайних ситуацій [29].

Таким чином, під час розроблення та використання технології підтримки прийняття рішень при формуванні команд для виконання ІТ-проектів враховано основні вимоги охорони праці, електро- та пожежної безпеки, санітарно-гігієнічні та ергономічні норми. Дотримання чинних нормативних актів забезпечує створення безпечних і комфортних умов праці, що є необхідною передумовою ефективної аналітичної діяльності та прийняття обґрунтованих управлінських рішень у сфері ІТ-проектів.

4.2. Розроблення та впровадження режимів радіаційного захисту працюючих і службовців у виробничій діяльності об'єкту в умовах радіоактивного забруднення місцевості

Радіація на сьогодні є чи не найнебезпечнішим фактором впливу не тільки на людину, але й на усі живі організми на планеті. Неконтрольовані ядерні реакції, ядерні війни та ряд інших можливих результатів людської діяльності, пов'язаних із радіацією негативно впливають на здоров'я людини та навколишнє середовище [30].

Основою розробки заходів захисту населення в умовах радіоактивного забруднення є Закон України «Про захист людини від іонізуючого

випромінювання», рекомендації Міжнародного агентства з атомної енергії, а також норми радіаційної безпеки України [31].

Враховуючи рівень радіації, а також прогноз можливих аварійних викидів радіоактивних речовин та метеорологічні дані, приймається рішення про проведення таких термінових і невідкладних заходів захисту в умовах ранньої фази радіаційної аварії:

- укриття населення;
- обмеження перебування населення на відкритій місцевості;
- евакуація у разі загрози здоров'ю;
- проведення йодової профілактики;
- тимчасова заборона вживання продуктів харчування і води із зони радіоактивного забруднення.

Крім цих заходів у період ранньої і пізньої фази проводяться довгострокові заходи [32]:

- тимчасове відселення;
- евакуація — переселення на постійне місце проживання;
- обмеження вживання води і продуктів харчування забруднених радіоактивними речовинами;
- заходи захисту при виробництві продукції тваринництва, рослинництва і лісогосподарської діяльності;
- дезактивація території і будівель;
- інші заходи: гідрологічні, протиповіневі, обмеження лісокористування, полювання, рибної ловлі, перебування у полі при проведенні сільськогосподарських робіт.

Критерієм для прийняття рішення про заходи захисту населення на ранній і середніх фазах після аварії є дози зовнішнього і внутрішнього опромінення (табл. 4.1) з установленими двома рівнями радіаційного впливу – нижнім і верхнім, згідно з рекомендацією МАГАТЕ і НРБУ.

При прогнозованому опроміненні, що не перевершує нижнього рівня, заходи, перелічені в табл. 4.1 не проводяться. Якщо прогнозоване опромінення перевищує нижній рівень, але не досягає верхнього рівня, то проведення вказаних заходів може бути відкладене [30].

Якщо прогнозоване опромінення досягає або перевищує верхній рівень, то обов'язково необхідно проводити заходи, наведені в табл. 4.1, навіть якщо вони пов'язані з порушенням нормальної життєдіяльності населення і об'єктів.

Підтвердженням цього є аварія на Чорнобильській АЕС, наслідки якої відчують до сьогодні як в Україні, так і за її межами.

Таблиця 4.1 - Критерії для прийняття рішень на ранній фазі розвитку аварії

Захисні заходи	Дозові критерії (прогнозована доза за перші 10 діб), мЗв			
	Все тіло		Окремі органи (легені, щитовидна залоза, шкіра)	
	Нижній рівень	Верхній рівень	Нижній рівень	Верхній рівень
Укриття, захист органів дихання і шкіри	5	50	50	500
Йодова профілактика:				
дорослі	—	—	50*	500*
діти, вагітні жінки	—	—	50*	250*
Евакуація:				
дорослі	50 10	500 50	500 200*	5000 500*
діти, вагітні жінки				

Радіаційний захист населення включає в себе:

— організацію безперервного контролю, виявлення та оцінку радіаційної та хімічної обстановки в районах розміщення радіаційно-небезпечних об'єктів;

- завчасне накопичення, підтримання в готовності і використання при необхідності засобів індивідуального захисту, приладів радіаційної розвідки і контролю;
- створення, виробництво та застосування уніфікованих засобів захисту, приладів і комплектів радіаційної розвідки і дозиметричного контролю;
- придбання населенням у встановленому порядку в особисте користування засобів індивідуального захисту та контролю за використанням їх за призначенням;
- своєчасне впровадження і застосування засобів і методів виявлення та оцінки масштабів і наслідків аварій на радіаційно-небезпечних об'єктах;
- створення і використання на радіаційно-небезпечних об'єктах систем (переважно автоматизованих) контролю обстановки і локальних систем оповіщення;
- розробку і застосування, за необхідності, режимів радіаційного захисту населення і функціонування об'єктів економіки та інфраструктури в умовах забрудненості (зараженості) місцевості;
- завчасне пристосування об'єктів комунально-побутового обслуговування і транспортних підприємств для проведення спеціальної обробки одягу, майна і транспорту, проведенням цієї обробки в умовах аварій;
- навчання населення використання засобів індивідуального захисту і правилам поведінки на забрудненій (зараженій) території.

До числа основних заходів щодо захисту населення від радіаційного впливу під час радіаційної аварії, належать:

- виявлення факту радіаційної аварії та оповіщення про неї;
- виявлення радіаційної обстановки в районі аварії;
- організація радіаційного контролю;
- встановлення та підтримання режиму радіаційної безпеки;

- проведення, при необхідності, на ранній стадії аварії йодної профілактики населення, персоналу аварійного об'єкта, учасників ліквідації наслідків аварії;
- забезпечення населення, персоналу аварійного об'єкта, учасників ліквідації наслідків аварії засобами індивідуального захисту та використання цих коштів;
- укриття населення, яке опинилося в зоні аварії, в притулках і укриттях, що забезпечують зниження рівня зовнішнього опромінення і захист органів дихання від проникнення в них радіонуклідів, які опинилися в атмосферному повітрі;
- санітарна обробка населення, персоналу аварійного об'єкта, учасників ліквідації наслідків аварії;
- дезактивація аварійного об'єкта, об'єктів виробничого, соціального, житлового призначення, території, сільськогосподарських угідь, транспорту, інших технічних засобів, засобів захисту, одягу, майна, продовольства і води;
- евакуація або відселення громадян із зон, в яких рівень забруднення перевищує допустимий для проживання населення.

Дотримання рекомендацій щодо захисту населення від впливу радіації, які проаналізовано вище, дає змогу мінімізувати ризики, пов'язані із загибеллю великої кількості людей, а також зберегти їхнє здоров'я.

ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

1. Обґрунтовано доцільність застосування системного підходу до задачі формування команд для виконання ІТ-проектів, що дало змогу розглядати команду як цілісну динамічну систему, у якій ефективність визначається не лише рівнем компетентностей окремих учасників, а й характером їх професійної та міжособистісної взаємодії, а також формалізувати процес прийняття рішень у вигляді багатокритеріальної задачі.

2. Проаналізовано сучасні моделі компетентностей і методи оцінювання учасників ІТ-проектів, що дало змогу структурувати профіль кандидата за професійними, поведінковими, рольовими та досвідними характеристиками і створити інформаційну основу для подальшого використання методів підтримки прийняття рішень, багатокритеріального аналізу та машинного навчання.

3. Досліджено вимоги та тенденції ІТ-ринку в контексті підготовки фахівців закладами вищої освіти, що дало змогу виявити розбіжність між фундаментальною підготовкою випускників і практичними потребами ІТ-проектів, а також обґрунтувати доцільність урахування характеристик освітнього середовища та репутації закладів освіти як додаткових факторів при формуванні команд.

4. Виконано системний аналіз задачі формування команди для виконання ІТ-проектів у контексті підтримки прийняття рішень, що дало змогу представити процес підбору розробників як багатокритеріальну задачу вибору альтернатив із урахуванням компетентностей фахівців, вимог проекту та обмежень організації-замовника.

5. Формалізовано модель представлення основних сутностей предметної області, зокрема розробників, ІТ-проектів та компаній, із використанням апарату теорії множин і реляційної алгебри, що дало змогу уніфікувати опис різномірних характеристик і забезпечити можливість їх подальшої автоматизованої обробки в системі підтримки прийняття рішень.

6. Розроблено метод оцінювання відповідності між компетентнісними профілями розробників та технологічними вимогами ІТ-проектів на основі метрик подібності, що дало змогу кількісно оцінювати рівень придатності кандидатів і формувати ранжовані списки потенційних учасників команди.

7. Запропоновано метод колаборативної фільтрації з використанням регуляризаторів для зменшення розрідженості та суб'єктивності даних, що дало змогу підвищити стійкість моделі до шумів, забезпечити більш достовірне прогнозування відсутніх оцінок і покращити якість рекомендацій при формуванні команд для виконання ІТ-проектів.

8. Визначено та формалізовано функціональні вимоги до програмного засобу підтримки прийняття рішень при підборі команд для виконання ІТ-проектів, що дало змогу чітко окреслити ролі користувачів, сценарії їх взаємодії з системою та межі автоматизації процесів підтримки прийняття рішень.

9. Спроектовано та реалізовано реляційну базу даних програмного засобу з урахуванням вимог нормалізації та цілісності, що дало змогу забезпечити структуроване зберігання даних про розробників, проекти, технології та їх взаємозв'язки, а також створити інформаційну основу для реалізації рекомендаційних алгоритмів.

10. Розроблено архітектуру програмного засобу на основі моделі Фаулера, що дало змогу розмежувати рівні представлення, бізнес-логіки та доступу до даних, підвищити масштабованість системи та спростити її подальший супровід і розширення.

11. Проведено експериментальну апробацію запропонованого методу та програмного засобу з використанням інструментів аналітичної візуалізації, що дало змогу підтвердити ефективність реалізованої технології підтримки прийняття рішень і досягти точності формування рекомендацій на рівні 84,65 %, достатньому для практичного використання у задачах підбору команд для виконання ІТ-проектів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Макарова О. Рейтинг вишів DOU 2019: у Могілянки з'явився конкурент за перше місце, а КПІ за межами 10-ки лідерів. URL: https://dou.ua/lenta/articles/ukrainian-universities-2019/?from=salary_report (дата звернення 11.10.2025 р.)
2. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 2nd ed. Springer, 2015. 1003 p. DOI: <https://doi.org/10.1007/978-1-4899-7637-6>.
3. Aggarwal C. C. Recommender Systems: The Textbook. Springer, 2016. 498 p. DOI: <https://doi.org/10.1007/978-3-319-29659-3>.
4. Jannach D., Zanker M., Felfernig A., Friedrich G. Recommender Systems: An Introduction. Cambridge University Press, 2016. DOI: <https://doi.org/10.1017/9781316672315>.
5. Adomavicius G., Tuzhilin A. Context-aware recommender systems. ACM Transactions on Information Systems, 2015. DOI: <https://doi.org/10.1145/2699740>.
6. Wu L., He X., Wang X., Zhang K., Wang M. A Survey on Accuracy-Oriented Neural Recommendation. arXiv, 2021. URL: <https://arxiv.org/abs/2104.13030> (дата звернення: 10.11.2025 р.).
7. Çano E., Morisio M. Hybrid recommender systems: A systematic literature review. Artificial Intelligence Review, 2019. DOI: <https://doi.org/10.1007/s10462-018-9654-y>.
8. Nunes I., Jannach D. A systematic review and taxonomy of explanations in decision support systems. User Modeling and User-Adapted Interaction, 2020. DOI: <https://doi.org/10.1007/s11257-020-09256-0>.
9. Shani G., Gunawardana A. Evaluating Recommendation Systems. In: Recommender Systems Handbook. Springer, 2015.
10. Koren Y., Bell R., Volinsky C. Matrix factorization techniques for recommender systems. IEEE Computer, 2015. DOI: <https://doi.org/10.1109/MC.2009.263>.

11. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. URL: <https://www.deeplearningbook.org> (дата звернення: 12.11.2025 р.).
12. Domingos P. The Master Algorithm. Basic Books, 2015. ISBN 9780465065707.
13. Dinov I. D. Data Science and Predictive Analytics. Springer, 2018. DOI: <https://doi.org/10.1007/978-3-319-72347-1>.
14. Burke R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 2016. DOI: <https://doi.org/10.1007/s11257-016-9168-9>.
15. Yatsyshyn V., Pastukh O., Kukharska V., Palamar A., Kulikov S. Method and tool of detecting software architecture patterns in the process of computer systems development. CEUR Workshop Proceedings, Volume 3896, 2024 4th International Workshop on Information Technologies: Theoretical and Applied Problems, ITTAP 2024 Ternopil 23 October 2024 through 25 October 2024 Code 206051. Ukraine and Opole, Poland. Pages 12 - 24.
16. Yatsyshyn V., Pastukh O., Lutskevych A., Tsymbalistyy V., Martsenko N. A Risks management method based on the quality requirements communication method in agile approaches. Information technologies: theoretical and applied problems, 2022. P. 1-10
17. Pastukh O., Yatsyshyn V., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU.Tern.: TNTU. 2023. Vol 109. No 1. P. 54–65.
18. Pastukh O., Yatsyshyn V., Zharovskyi R., Shablii N. Software tool for productivity metrics measure of relational database management system. Mathematical Modeling. No 1 (48). 2023. P. 7-17.
19. Pastukh O., Yatsyshyn V. (2024) Development of software for neuromarketing based on artificial intelligence and data science using high-performance computing and parallel programming technologies. Scientific Journal of TNTU (Tern.). Vol 113. No 1. pp. 143–149.

20. Lops P., Gemmis M., Semeraro G. Content-based recommender systems: State of the art. Springer, 2015.
21. Resnick P., Varian H. Recommender systems. Communications of the ACM, 2015. DOI: <https://doi.org/10.1145/1864708>.
22. Zhang S., Yao L., Sun A., Tay Y. Deep learning based recommender system: A survey. ACM Computing Surveys, 2019. DOI: <https://doi.org/10.1145/3285029>.
23. ISO/IEC 25010:2011 (reaffirmed 2017). Systems and software quality models. URL: <https://www.iso.org/standard/35733.html> (дата звернення: 12.10.2025 р.).
24. IEEE Std 42010-2011 (reaffirmed 2017). Systems and software engineering — Architecture description. URL: <https://standards.ieee.org/standard/42010-2011.html> (дата звернення: 12.11.2025 р.).
25. Saaty T. L. Decision making with the analytic hierarchy process. International Journal of Services Sciences, 2016. DOI: <https://doi.org/10.1504/IJSSCI.2008.017590>.
26. Power D. J. Decision Support Systems: Concepts and Resources. Journal of Decision Systems, 2015. DOI: <https://doi.org/10.1080/12460125.2015.994329>.
27. Python-recsys on Github. URL: <https://github.com/ocelma/python-recsys> (дата звернення 01.12.2025 р.).
28. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Київ. 2018. URL: http://sop.zp.ua/norm_праор_0_00-7_15-18_01_ua.php (дата звернення: 10.12.2025 р.).
29. Про затвердження Правил експлуатації та типових норм належності вогнегасників. URL: <https://zakon.rada.gov.ua/laws/show/z0225-18> (дата звернення 12.12.2025 р.).
30. Стеблюк М.І. Цивільна оборона та цивільний захист: Підручник. К.: Знання, 2010. 487 с.

31. Стручок В.С. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «Безпека в надзвичайних ситуаціях» / Тернопіль: ФОП Паляниця В. А. 156 с.

32. Стручок В.С. Навчальний посібник «Техноекологія та цивільна безпека. Частина «цивільна безпека»» / Тернопіль: ФОП Паляниця В. А. 156 с.

Додаток А
Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

ХІІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



17-18 грудня 2025 року

ТЕРНОПІЛЬ
2025

А. Станько, С. Козак, С. Кульчицький ЗАХИЩЕНА ТЕЛЕМЕТРІЯ В ІОТ: ПОРІВНЯЛЬНИЙ АНАЛІЗ MQTT OVER TLS, HTTPS ТА COAP/DTLS ДЛЯ ДОМАШНІХ І ПРОМИСЛОВИХ ЗАСТОСУВАНЬ A. Stanko, S. Kozak, S. Kulchytskyi SECURE TELEMETRY IN THE IOT: A COMPARATIVE ANALYSIS OF MQTT OVER TLS, HTTPS AND COAP/DTLS FOR HOME AND INDUSTRIAL APPLICATIONS	214
М. Трембач, Р. Павелко, Д. Биць ІОТ-СИСТЕМА БЕЗПЕЧНОГО МОНІТОРИНГУ ТА КЕРУВАННЯ ДОМАШНІМИ ПРИСТРОЯМИ З ПІДТРИМКОЮ OPENHAB M. Trembach, R. Pavelko, D. Byts IOT SYSTEM FOR SECURE MONITORING AND CONTROL OF HOME DEVICES WITH OPENHAB SUPPORT	215
В. Яцишин, М. Кіт СИСТЕМНИЙ ПІДХІД У ФОРМУВАННІ КОМАНДИ ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ V. Yatsyshyn, M. Kit SYSTEM-BASED APPROACH TO IT PROJECT TEAM FORMATION	216
В. Яцишин, М. Кіт ФАКТОРИ ВПЛИВУ НА ЕФЕКТИВНІСТЬ ВИКОНАННЯ ІТ-ПРОЄКТІВ V. Yatsyshyn, M. Kit KEY FACTORS INFLUENCING IT PROJECT PERFORMANCE	217
Я. Гриневич, Я. Шкіра АНАЛІЗ АВТОМАТИЗОВАНИХ МЕТОДІВ УПРАВЛІННЯ МЕТАЛУРГІЙНИМ ВИРОБНИЦТВОМ НА ОСНОВІ ВИКОРИСТАННЯ MES-СИСТЕМ Ya. Hrynevych, Ya. Shkira ANALYSIS OF AUTOMATED METHODS OF METALLURGICAL PRODUCTION MANAGEMENT BASED ON THE USE OF MES SYSTEMS	218
Н. Вітвіцький Н. МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЯ БАГАТОРІВНЕВОЇ СИСТЕМИ БЕЗПЕКИ ВЕБ-РЕСУРСУ N. Vitvitskyi MODELING AND IMPLEMENTATION OF A MULTI-LAYER WEB RESOURCE SECURITY SYSTEM	220
Н. Гашин, К. Макаричева ПРОГНОЗУВАННЯ ЕКОНОМІЧНИХ ПОКАЗНИКІВ ТРАНСПОРТНИХ ПЕРЕВЕЗЕНЬ З ВИКОРИСТАННЯМ ЕЛЕКТРОННИХ ТАБЛИЦЬ MS EXCEL N. Gashchyn, K. Makarycheva FORECASTING ECONOMIC INDICATORS OF TRANSPORTATION USING MS EXCEL SPREADSHEETS	221
ЗМІСТ	222

УДК 004.9

В. Яцишин, к. т. н., доц.; М. Кіт

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

СИСТЕМНИЙ ПІДХІД У ФОРМУВАННІ КОМАНДИ ДЛЯ ВИКОНАННЯ ІТ-ПРОЄКТІВ

UDC 004.9

V. Yatsyshyn, PhD., Assoc. Prof.; M. Kit

SYSTEM-BASED APPROACH TO IT PROJECT TEAM FORMATION

Формування команди для виконання ІТ-проєкту є складною багатофакторною задачею, у якій поєднуються технічні, організаційні та людські аспекти. На якість роботи команди впливають компетентності фахівців, їхній досвід, ролі у проєктній структурі, здатність до взаємодії, психологічні особливості, а також фактори зовнішнього середовища. Така багатовимірність робить процес формування команди типовою проблемою системного аналізу, де необхідно не лише оцінити окремі характеристики кандидатів, але й визначити інтегральну ефективність їхньої взаємодії.

У традиційних підходах вибір складу команди здебільшого базується на досвіді менеджера, експертних оцінках та попередніх результатах роботи фахівців. Проте суб'єктивність таких рішень призводить до значної варіативності результатів та підвищує ризики формування неефективних груп.

Системний підхід дозволяє розглядати команду як цілісну структуру, у якій взаємодія між учасниками команди має не менше значення, ніж властивості кожного елемента окремо. Відповідно, критерії оцінювання повинні охоплювати не тільки професійний рівень розробників, а й параметри командної динаміки, конфліктність, комунікаційні особливості, стиль роботи та потенційний вплив учасників один на одного. Різні дослідження в галузі управління ІТ-проєктами показують, що успіх команди визначається поєднанням професійних компетентностей та «м'яких» навичок. Саме тому формування команди повинно враховувати як технічні компетентності (володіння мовами й технологіями програмування), так і фактори психологічної та комунікативної сумісності.

У сучасних ІТ-компаніях активно розвиваються моделі компетентностей, що структурують вимоги до фахівців різних ролей. Зокрема, поширеними є T-shaped та π-shaped моделі, що відображають необхідність поєднання глибоких спеціалізованих знань із широким розумінням суміжних сфер. До того ж стандарти, такі як SFIA (Skills Framework for the Information Age), надають формальні описи професійних ролей і компетентностей, що дозволяє більш об'єктивно оцінювати кандидатів під час формування команди.

Системний підхід також передбачає декомпозицію задачі формування команди на підзадачі, серед яких: визначення ролей у проєкті, формування вимог до кожної ролі, зіставлення профілів кандидатів із вимогами, оцінювання сумісності між кандидатами, моделювання ризиків та прогнозування потенційної продуктивності команди. Кожна з цих підзадач може бути формалізована та описана математично, що відкриває можливість використання методів підтримки прийняття рішень, аналізу даних і оптимізації. Застосування системного підходу також дозволяє враховувати динамічний характер ІТ-проєктів. Команди не є статичними структурами: вони змінюються в залежності від етапу життєвого циклу проєкту, появи нових вимог або ризиків. Тому моделі формування команд повинні бути адаптивними та враховувати можливість перебудови ролей та функцій у процесі виконання завдань.

УДК 004.9

В. Яцишин, к. т. н., доц.; М. Кіт

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ФАКТОРИ ВПЛИВУ НА ЕФЕКТИВНІСТЬ ВИКОНАННЯ ІТ-ПРОЄКТІВ

UDC 004.9

V. Yatsyshyn, PhD., Assoc. Prof.; M. Kit

KEY FACTORS INFLUENCING IT PROJECT PERFORMANCE

Ефективність виконання ІТ-проектів залежить від багатьох чинників і пов'язана з великою кількістю ризиків. Однак найвагомішим фактором є людський. Команда, яка виконує ІТ-проект, розглядається як динамічна соціотехнічна система, у якій характеристики кожного учасника та їх взаємодія визначають загальний результат проекту. Фактори, які впливають на ефективність команди, представлені у табл. 1.

Таблиця 1. Фактори впливу на ефективність команди

Група факторів	Приклади	Вплив на команду
Технічні компетентності	Мови програмування, фреймворки, DevOps, архітектура	Визначають здатність виконувати технічні задачі
М'які навички	Комунікація, емоційний інтелект, лідерство	Формують командну динаміку та атмосферу
Проектні ролі	Developer, Architect, QA, PM, BA, DevOps	Забезпечують розподіл відповідальності
Психологічна сумісність	Темперамент, стиль роботи, конфліктність	Впливає на стабільність та продуктивність
Організаційні обмеження	Бюджет, терміни, віддаленість	Формують рамки прийняття рішень

Така класифікація показує, що завдання формування команди не може вирішуватися лише за одним критерієм – воно є багатокритеріальним, що вимагає методу підтримки прийняття рішень. У системному аналізі команда розглядається як сукупність елементів (учасників), які взаємодіють між собою. Її ефективність визначається не лише якістю елементів, але і якістю зв'язків між ними. Для проведення системного аналізу ефективності команди пропонується використовувати модель, до складу якої входять:

- вхідні дані: компетентності, досвід, поведінкові характеристики;
- процеси аналізу: оцінювання кандидатів, прогнозування сумісності, багатокритеріальний вибір;
- обмеження: бюджет, ролі, вимоги проекту;
- вихідні дані: оптимальна команда;
- зворотний зв'язок: адаптація команди у процесі виконання проекту.

Традиційно менеджери формують команду суб'єктивно, на основі попереднього досвіду, рекомендацій, інтуїції. Системний підхід дозволяє: формалізувати вимоги, оцінювати кандидатів за інтегральними метриками, моделювати сумісність, прогнозувати ризики. Тому формування команди є багатокритеріальною системною задачею, де важливі як індивідуальні компетентності, так і взаємодія учасників. Системний підхід дозволяє побачити команду як цілісну структуру, оптимізація якої можлива лише через моделювання та аналіз даних.