

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

**КВАЛІФІКАЦІЙНА РОБОТА**  
на тему: «Система управління ІТ проектами на основі  
застосування генеративного штучного інтелекту  
IT project management system based on the use of generative  
artificial intelligence»

на здобуття освітнього ступеня магістра  
зі спеціальності F3 Комп'ютерні науки  
(код, найменування спеціальності)  
освітньо-професійної програми Комп'ютерні науки  
(назва)

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

\_\_\_\_\_  
(підпис) Ігор МАЛЯРЕНКО  
(Ім'я, ПРІЗВИЩЕ здобувача)

Виконав:  
здобувач вищої освіти  
група КНДМ-62

Ігор МАЛЯРЕНКО

Керівник:  
науковий ступінь,  
вчене звання

Юлія Березовська  
доктор філософії

Рецензент:  
науковий ступінь,  
вчене звання

\_\_\_\_\_  
(Ім'я, ПРІЗВИЩЕ)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Комп'ютерних наук

Ступінь вищої освіти Магістр

Спеціальність ЕЗ Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедру Комп'ютерних наук

\_\_\_\_\_ Віктор ВИШНІВСЬКИЙ

«\_\_\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

\_\_\_\_\_ Маляренку Ігорю Ігоровичу

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Система управління ІТ проектами на основі застосування генеративного штучного інтелекту

керівник кваліфікаційної роботи Юлія Березовська доктор філософії.

*(Ім'я, ПРИЗВИЩЕ науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025р. №467

2. Строк подання кваліфікаційної роботи «26» грудня 2025р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з питань управління проектами, великих мовних моделей, навчання нейронних мереж,

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз концепцій великих мовних моделей із засобів генеративного штучного інтелекту

Аналіз проблем управління ІТ-проєктами

Розробка системи управління ІТ-проєктами з використанням генеративного штучного інтелекту

5. Перелік графічного матеріалу: презентація

1. Мета, об'єкт дослідження, предмет дослідження
2. Ключові компоненти генеративного штучного інтелекту.
3. Напрями застосування генеративного штучного інтелекту в управлінні проєктами
1. Життєвий цикл розробки програмного забезпечення
2. Результати спілкування з ChatGPT
3. Висновки

6. Дата видачі завдання «15» вересня 2025 р.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи                    | Строк виконання етапів роботи | Примітка |
|-------|--------------------------------------------------------|-------------------------------|----------|
| 1     | Підбір науково-технічної літератури                    | 19.10-05.11.25                | виконано |
| 2     | Дослідження концепцій генеративного штучного інтелекту | 05.11-12.11.25                | виконано |
| 3     | Дослідження концепцій управління ІТ-проєктами.         | 13.11-19.11.25                | виконано |
| 4     | Розробка системи управління ІТ-проєктами               | 27.11-03.12.25                | виконано |
| 5     | Вступ, висновки, реферат                               | 04.12-10.12.25                | виконано |
| 6     | Розробка обов'язкових демонстраційних матеріалів       | 11.12-20.12.25                | виконано |
| 7     | Попередній захист роботи                               | 21.12-29.12.25                | Виконано |

Здобувач вищої освіти

\_\_\_\_\_

(підпис)

Ігор МАЛЯРЕНКО

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

\_\_\_\_\_

(підпис)

Юлія БЕРЕЗОВСЬКА

(Ім'я, ПРІЗВИЩЕ)





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 74 стор., 11 рис., 1 табл. 14 джерел.

Наукове завдання – дослідження основних методів керування об'єктом і можливості їх об'єднання в одній системі.

Мета роботи – Підвищення ефективності менеджерських рішень при управлінні ІТ-проєктами

Об'єкт дослідження – Процес управління створенням ІТ-проєктів.

Предмет дослідження – життєвий цикл розробки програмного забезпечення

Короткий зміст роботи: Проведено дослідження концепцій генеративного штучного інтелекту і великих мовних моделей, здійснений порівняльний аналіз сучасних чатботів з генеративним штучним інтелектом.

Запропоновано систему управління ІТ-проєктами із застосуванням генеративного штучного інтелекту.

Використання запропонованої системи дозволяє збільшити ефективність менеджерів ІТ-проєктів.

**КЛЮЧОВІ СЛОВА:** КЕРУВАННЯ ІТ-ПРОЕКТАМИ, ГЕНЕРАТИВНИЙ ШТУЧНИЙ ІНТЕЛЕКТ, CHAT GPT, МЕТОДОЛОГІЯ, SDLC

## **ABSTRACT**

Text part of the qualification work for obtaining a master's degree: 74 pages, 11 figures, 1 table. 14 sources.

Scientific task – research of the main methods of object management and the possibility of their combination in one system.

Objective of the work – Increasing the efficiency of managerial decisions in managing IT projects

Object of the research – The process of managing the creation of IT projects.

Subject of the research – the software development life cycle

Summary of the work: The concepts of generative artificial intelligence and large language models were studied, a comparative analysis of modern chatbots with generative artificial intelligence was carried out.

An IT project management system using generative artificial intelligence is proposed.

The use of the proposed system allows to increase the efficiency of IT project managers.

**KEYWORDS:** IT PROJECT MANAGEMENT, GENERATIVE ARTIFICIAL INTELLIGENCE, CHAT GPT, METHODOLOGY, SDLC

## ЗМІСТ

Стор.

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| <b>ВСТУП .....</b>                                                       | <b>9</b>  |
| <b>1 СУТНІСТЬ ТА ПРИНЦИПИ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ .....</b>     | <b>11</b> |
| 1.1 Принципи великих мовних моделей .....                                | 11        |
| 1.2 Робота з великими мовними моделями .....                             | 15        |
| <b>2 ОГЛЯД ПРОБЛЕМ В УПРАВЛІННІ ІТ ПРОЕКТАМИ .....</b>                   | <b>26</b> |
| 2.1 Проектний менеджмент .....                                           | 26        |
| 2.2 Напрями застосування GAI в управлінні ІТ-проектами .....             | 30        |
| 2.3 Аналіз наявних проблем у сфері керування проектами .....             | 35        |
| 2.4 Переваги і недоліки GAI в управлінні проектами .....                 | 38        |
| 2.5 Програмні засоби управління ІТ-проектами .....                       | 42        |
| <b>3 СИСТЕМА УПРАВЛІННЯ ПРОЄКТУВАННЯМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b> | <b>46</b> |
| 3.1 Поняття моделі життєвого циклу .....                                 | 46        |
| 3.2 Вибір моделі генеративного штучного інтелекту .....                  | 51        |
| 3.3 Опис роботи з системою .....                                         | 59        |
| 3.3.1 Етап збору та аналізу вимог .....                                  | 59        |
| 3.3.2 Етап проектування та створення дизайну .....                       | 61        |
| 3.3.3 Етап розробки продукту .....                                       | 63        |
| 3.3.4 Етап тестування .....                                              | 64        |
| 3.3.5 Етап релізу та підтримки продукту .....                            | 65        |
| <b>ВИСНОВКИ.....</b>                                                     | <b>67</b> |
| <b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>                                             | <b>68</b> |
| <b>ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....</b>                                    | <b>70</b> |

## ВСТУП

У світі інформаційних технологій конкурентоспроможність є ключовими факторами, ефективне управління ІТ-проєктами стає дедалі важливішим завданням для компаній будь-якого розміру. Розуміння ролі та переваг, які може надати генеративний штучний інтелект у цьому контексті, тому поступово перетворюється на необхідність. Лише ті, хто осягає можливості цих технологій, можуть цілеспрямовано вживати заходів для підвищення продуктивності, оптимізації витрат і тим самим забезпечення довгострокового успіху ІТ-проєктів. У цьому світлі стратегічні міркування щодо інтеграції генеративного ШІ в існуючі процеси управління проєктами набувають значення, оскільки допомагають здобути конкурентні переваги та реалізувати проєкти ефективніше.

Генеративний штучний інтелект може приносити значні переваги у підвищенні продуктивності управління ІТ-проєктами й діяти на кількох рівнях. Його практичне застосування насамперед охоплює автоматизацію рутинних завдань і процесів, пов'язане зі зниженням видатків на персонал, зменшенням кількості помилок та підвищенням точності виконання. Крім того, він сприяє прискоренню та збільшенню якості виконання завдань. Перебираючи на себе монотонні роботи, такі як складання звітів, підготовка документації або планування ресурсів і засобів, генеративний ШІ зменшує часові та кадрові витрати на ці здебільшого повторювані завдання. Це помітно звільняє проєктні команди від непродуктивних дій і створює простір для інших завдань.

Автоматизація цих рутинних операцій також знижує потребу в значній кількості працівників для простих повторюваних завдань. Замість залучення людей до монотонної роботи, кадрові ресурси можна спрямувати на складніші та стратегічні завдання, що вимагають людського інтелекту, досвіду та креативності. Таким чином генеративний ШІ дає можливість ефективного використання людських ресурсів й сприяє оптимізації витрат на працю, оскільки потужності краще розподіляються, а пріоритети – коригуються.

Ще однією важливою перевагою генеративного ШІ є його здатність зменшувати кількість помилок і покращувати показники точності виконання задач. Системи ШІ можуть швидко й майже безпомилково виконувати розрахунки, систематично аналізувати великі дані і виводити якісні результати. Це допомагає мінімізувати типові помилки, що виникають через людську втомлю або неуважність, і забезпечує точніші та більш надійні результати в управлінні ІТ-проєктами. Завдяки вищій точності ризики можна оцінювати краще, а процеси – робити стабільнішими.

Крім того, генеративний ШІ через автоматизацію та оптимізацію процесів може значно прискорити виконання завдань і процесів управління в ІТ-проєктах. Він аналізує за короткий час великі масиви даних, виявляє можливі шляхи оптимізації та надає прогнози щодо очікуваних результатів. Ці можливості дозволяють проєктним командам більше зосередитися на стратегічних аспектах проєкту й одночасно приймати швидші, більш обґрунтовані рішення. Таким чином генеративний ШІ сприяє прискоренню прийняття рішень, ранньому виявленню вузьких місць і своєчасному вжиттю заходів.

Управління проєктами має велике значення у багатьох галузях, оскільки забезпечує структуру, координацію та лідерство, необхідні для отримання позитивних результатів. З прогресом технологій зацікавленість до GAI зростає з подальшим вдосконаленням практики керування проєктами та пристосування її до зростаючих вимог. Генеративний ШІ – це підгалузь ШІ, яка використовує машинне навчання для створення нових, не стандартних рішень або підвищення ефективності існуючих процесів. Завдяки цьому підходу можна автоматизувати повторювані завдання, оптимізувати розподіл ресурсів і покращити оцінку ризиків, не відмовляючись від базових принципів управління проєктами.

# 1 СУТНІСТЬ ТА ПРИНЦИПИ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ

## 1.1 Принципи великих мовних моделей

Генеративний штучний інтелект (ШІ) став у багатьох сферах надзвичайно потужним інструментом і нині широко застосовується в різних доменах. Тому різні теорії та концепції менеджменту слід переглянути й дослідити в контексті генеративного ШІ, оскільки ця технологія може впливати на управлінську роботу на кількох рівнях: на стратегічному, функціональному та адміністративному рівнях. Було б наївно й нереалістично припускати, що саме управління проектами залишиться не торканим цими змінами. Генеративні системи ШІ, що базуються на алгоритмах і методах машинного навчання, здатні генерувати нові ідеї, ставити під сумнів існуючі процеси й оптимізувати їх, а також навіть частково їх приймати. Це може суттєво змінити способи роботи й розподіл відповідальності в управлінні.

Одним із найважливіших досягнень у сфері генеративного ШІ є розробка великих мовних моделей, також відомих як Large Language Models (LLM). Ці моделі репрезентують здатність ШІ не лише розуміти людську мову, а й генерувати її та трансформувати мовний контент. У категорії генеративного ШІ особливої уваги та визнання набув попередньо навчений генеративний трансформер (GPT), мовна модель від OpenAI. Ґрунтуючись на архітектурі трансформера, серія GPT спричинила помітну зміну парадигми в можливостях обробки природної мови (Natural Language Processing, NLP), особливо в таких завданнях, як автоматичне генерування тексту, машинний переклад та відповіді на запитання природною мовою.

Методи LLMs, як-от реалізації в ChatGPT (версії GPT-3.5 і GPT-4), можливі завдяки навчанню на обширних текстових корпусах. Ці моделі навчалися на величезних масивах текстових даних і продемонстрували видатні здібності в різноманітних задачах NLP. Зокрема, ChatGPT виявився надзвичайно універсальним і продемонстрував свою ефективність у різних сферах

застосування: наприклад у освіті, де він допомагає в створенні й поясненні навчальних матеріалів; у системі охорони здоров'я, де може надавати допоміжні тексти та інформацію; у висновках і аналізах; у безпосередньому генеруванні тексту; у взаємодії людини з комп'ютером через діалоговий інтерфейс; а також у наукових дослідженнях, де він допомагає в літературних оглядах, узагальненнях або формулюванні ідей. Крім того, розвиток і адаптація великих мовних моделей та їх спрямування до автономних агентів, прикладами чого є концепції на кшталт Auto-GPT, вказують на потенціал рухатися в бік певного ступеня загального інтелекту. Водночас виникають значні й невіддільні виклики при прямому використанні LLM як повністю автономних агентів. Запропоновані агенти часто працюють через складні робочі процеси, які, по суті, дозволяють моделі у кілька кроків взаємодіяти сама з собою та планувати дії без безпосереднього людського втручання; це, однак, породжує питання щодо надійності, контролю та відповідальності.

Впровадження ChatGPT у багатьох контекстах призвело до суттєвого підвищення якості та продуктивності продукції. Використання таких моделей о скорочує час для здійснення певних завдань, водночас у багатьох випадках підвищуючи якість виконаної роботи. Крім того, застосування ChatGPT може сприяти зменшенню нерівності серед працівників, надаючи людям із меншими професійними навичками або меншим досвідом покращення, які дають їм переваги, що зазвичай були б доступні лише досвідченим фахівцям. Дослідження показують, що з великими мовними моделями для програмування (також LLM, спеціально адаптованими для програмістських завдань) значна частина професійних задач може виконуватися значно швидше: за оцінками, приблизно 15 % усіх завдань можна виконати помітно швидше при збереженні поточного рівня якості. Якщо ж використовувати програмне забезпечення та інструменти, що широко базуються на LLM, ця доля – тобто частка завдань, які можна виконувати швидше – може зрости навіть до близько 56 %, що демонструє потенційний вплив на повсякденну роботу та ефективність.

Принципи роботи генеративного ШІ. Генеративний штучний інтелект у своїй основі базується на застосуванні складних алгоритмів глибокого навчання, які дозволяють обробляти, аналізувати дуже великі обсяги даних і генерувати з них новий, когерентний контент. Центральним елементом у цьому контексті є навчання на дуже великих наборах даних. Моделі, що працюють із текстами, зазвичай опрацьовують мільйони книг, статей, веб-сторінок та інших джерел тексту, щоб розвинути статистичне й структурне розуміння того, як побудована мова, які слова часто трапляються разом і як у тексті формуються логічні зв'язки та виразні форми. Ці широкі відомості використовуються під час навчання для подальшої генерації текстів, що звучать природно й відповідають поставленим завданням або запитам (рис. 1.1).

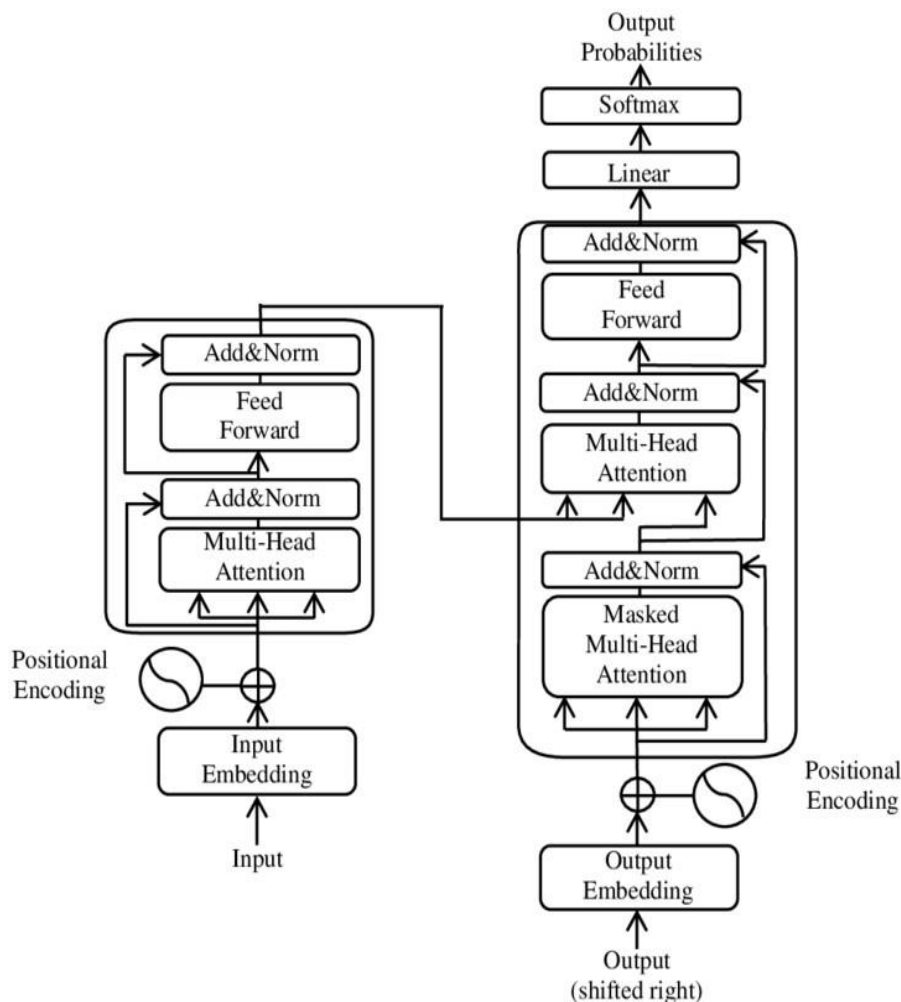


Рисунок 1.1 - Схематична ілюстрація архітектури моделі трансформера.[13]

Одним із ключових технологічних досягнень у генеративному ШІ є архітектура трансформера, схема якої, наприклад, наведена на рисунку 1.2. Ця архітектура містить увагу), що дозволяє моделі під час аналізу тексту визначати, які слова чи фрази є особливо важливими для розуміння контексту. Механізм багатоголової уваги забезпечує можливість моделі одночасно зосереджувати увагу на кількох релевантних частинах вхідного тексту, таким чином фіксуючи складні залежності й взаємозв'язки між словами. Завдяки цьому модель може цілеспрямовано концентруватися на тих частинах речення, які є вирішальними для значення. Простим прикладом є запит: «Столиця Франції?». У такому випадку модель визначає, що слова «столиця» та «Франція» містять ключову інформацію, тоді як інші слова чи фрази менш значущі. На підставі цього розпізнавання модель може згенерувати відповідну відповідь, зваживши релевантну інформацію та включивши її у вихідний текст.

Сам процес створення контенту відбувається таким чином: користувач ставить конкретний запит або називає певну тему, а модель у відповідь генерує відповідь, аналізуючи і обробляючи введені дані. Цей процес заснований на моделі, яка передбачає найбільш імовірні слова або елементи в даному контексті й поступово складає з них послідовний текст або інший формат вихідних даних. Саме завдяки цим оцінкам ймовірності виникає згенерована відповідь, яка за змістом, структурою й стилем відповідає початковому промпту.

Наглядний приклад ілюструє цей процес: якщо запитати, наприклад, «Напиши коротку історію про космос», модель використовує свої наявні знання та вивчені зразки, щоб створити зв'язний текст. Вона при цьому стежить за тим, щоб історія мала логічну побудову і стилістично була доречною, тобто містила відповідні описи, зрозумілий сюжет і відповідний тон. Згенерована історія є результатом зіставлення між запитом користувача та мовними й знаннявими шаблонами, збереженими в моделі.

Контроль над результатами генерації можна здійснювати за допомогою різних параметрів, наприклад установкою так званої «температури» моделі, яка впливає на те, наскільки креативними або консервативними будуть відповіді, а

також обмеження максимальної довжини тексту, що визначає, наскільки докладною може бути відповідь. Такі параметри дозволяють налаштовувати баланс між передбачуваністю й різноманітністю виходів і таким чином адаптувати гнучкість моделі до конкретних потреб користувача. Варіюючи ці налаштування, можна цілеспрямовано добиватися того, щоб відповідь була або точнішою й стислою, або більш винахідливою й об'ємною.

Генеративний ШІ вирізняється здатністю адаптації до нових задач і працювати в різних системних середовищах. Наприклад, його можна інтегрувати в рішення для служби підтримки клієнтів, де він може звертатися до корпоративних баз даних, для врахування історії взаємодій з клієнтом і таким чином надавати персоналізовані, контекстно релевантні відповіді. Завдяки цьому вона може застосовуватися в багатьох сценаріях або як автономний інструмент, або як доповнення до людської праці.

Хоча базові принципи роботи генеративного ШІ технічно складні й ґрунтуються на передових статистичних моделях і методах навчання, вони створюють інструменти, що суттєво полегшують повсякденне життя людей. Такі інструменти сприяють прискоренню робочих процесів, спрощенню рутинних завдань і підвищенню ефективності в багатьох сферах. Автоматизація певних процесів і підтримці в творчих або аналітичних завданнях генеративний ШІ може зробити помітний внесок у зростання продуктивності.

Застосування генеративного ШІ різноманітне і знаходить місце в багатьох галузях. Такі функції дають змогу компаніям швидше й індивідуальніше реагувати на потреби клієнтів і робити комунікацію більш послідовною. Загалом видно, що генеративний ШІ пропонує практичні рішення в численних контекстах, які підтримують як рутинні процеси, так і збагачують творчі завдання.

## 1.2 Робота з великими мовними моделями

Генеративний штучний інтелект (ГШІ) або великі моделі навчання (LLM) став однією з найбільш обговорюваних тем у сучасній технологічній спільноті. Ці технології привертають таку велику увагу тому, що вони значно виходять за межі класичного аналізу даних або простих завдань автоматизації й натомість здатні продукувати креативні, нові результати. Це не просто алгоритм для аналізу даних або виконання заданих завдань – ГШІ здатен генерувати унікальний контент.

Цей контент я робить видимою потужність моделей: навчаючись на великих збірках прикладів, вони можуть переймати стилі й структури та створювати нові, пов'язані тексти. Або модель намалює ілюстрацію до історії за вашим описом. У таких випадках ГШІ перекладає текстовий ввід на візуальні характеристики і генерує зображення, що відповідає опису. Саме в цьому суть генеративного ШІ – створювати нове на їх основі з виявлення зв'язків у великих масивах даних і. Ця здатність базується на виявленні шаблонів, кореляцій та типової структури й подальшому творчому комбінуванні їх, щоб надати користувачу корисні результати.

Генеративний штучний інтелект ґрунтується на застосуванні складних алгоритмів глибокого навчання, які дозволяють аналізувати великі обсяги даних і генерувати новий контент. Ці алгоритми складаються з багатьох шарів нейронних мереж, що поступово опрацьовують і абстрагують інформацію. Ключовим елементом є навчання на великих датасетах. Моделі, що працюють з текстом, наприклад, опрацьовують мільйони книг, статей і вебсайтів, щоб зрозуміти, як побудована мова, які слова вживаються разом і як формується логіка вираження думки. Завдяки такому обширному навчанню моделі вивчають статистичні закономірності мови, граматичні структури, стилістичні прийоми та тематичні зв'язки. Ця інформація використовується для генерації текстів, які звучать природно і відповідають поставленому завданню. Подібним чином моделі для зображень навчають на великих парах «зображення-текст», щоб пов'язати візуальні ознаки зі словесним описом і таким чином створювати відповідні зображення.

Процес створення контенту відбувається так: користувач формулює запит або тему, модель аналізує вхідні дані і генерує відповідь. Спочатку вхідні дані переводять у форму, яку модель може опрацювати, а потім на основі набутого знання створюється послідовність виходів, що має бути найкращим продовженням або найвідповіднішим представленням. Цей процес базується на передбаченні найбільш вірогідних слів або елементів у заданому контексті. Модель обчислює ймовірність для кожного можливого продовження і вибирає вихід залежно від своїх стратегій або налаштувань. Це генеративне передбачення є ядром того, як створюються зв'язні та контекстно відповідні матеріали.

Наприклад, модель на запит: «Напиши коротку історію про космос» на основі своїх знань сформулює текст, що є логічним і стилістично коректним. Вона враховує типові мотиви й формулювання на тему космосу, упорядковує їх у наративний послідовний виклад і забезпечує мовну когерентність. Залежно від того, як сформульовано запит і які додаткові вказівки задано (наприклад, тон або довжина), модель підлаштовує свій вихід, щоб відповідати очікуванням. Завдяки цьому користувачі можуть цілеспрямовано впливати на те, який результат вони хочуть отримати.

Контроль над результатами генерації здійснюється через налаштування таких параметрів, як «температура» моделі (впливає на креативність відповідей) або максимальна довжина тексту. Температура визначає, наскільки детермінованим або варіативним є вибір наступних слів: нижча температура дає більш передбачувані, послідовні результати, тоді як вища температура може призвести до несподіваніших, різноманітніших виходів. Параметр максимальної довжини тексту обмежує, наскільки великим буде згенерований контент. Додатково можуть застосовуватися інші параметри чи техніки про-мптдизайну для впливу на стиль, фокус або рівень деталізації виходу.

Генеративний ШІ також вирізняється адаптації до нових завдань і працювати в різних системах. Трансферне навчання, донастройка та використання специфічних датасетів дозволяють пристосувати загальну модель до спеціальних випадків застосування, щоб вона була ефективною в різних контекстах.

Наприклад, її можна інтегрувати в системи обслуговування клієнтів в компаніях, щоб враховувати історію взаємодій з клієнтами і надавати персоналізовані відповіді. Інтегруючись у існуючі системи, ГШІ може отримувати й використовувати релевантні дані для формування контекстуально обґрунтованих, індивідуальних реакцій. Вона підтримує творчі процеси, постачаючи ідеї, створюючи чорнові варіанти або пропонуючи візуальні напрацювання, які художники й автори можуть використовувати як основу або джерело натхнення.

Хоча принципи роботи генеративного ШІ доволі складні, вони дають змогу розробляти інструменти, які значно полегшують життя людей і підвищують ефективність у багатьох сферах. Поєднання розпізнавання шаблонів, здатності до навчання та адаптивності призводить до застосувань, що автоматизують рутинні завдання, прискорюють творчі процеси та допомагають фахівцям (рис.1.2).

### Навчання мережі.

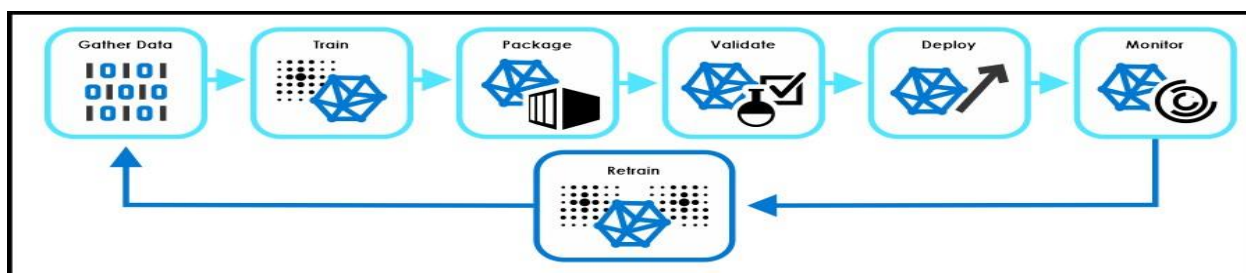


Рисунок. 1.2 - Схема роботи ГШІ, що ілюструє процес навчання моделі та генерації нового контенту

Ще один важливий аспект полягає в тому, що генеративні моделі ШІ зазвичай навчаються у двох чітко відокремлених фазах. Перша фаза називається попереднім навчанням; у цій фазі нейронна мережа аналізує великі масиви даних, щоб знайти загальні закономірності, структури та регулярності в текстах або інших видах контенту. Попереднє навчання закладає базові здібності моделі, наприклад розуміння мови, синтаксичні й семантичні шаблони, а також загальне знання й розуміння контексту.

У другій фазі відбувається цілеспрямована доопрацювання попередньо навченої моделі, під час якої його точно настроюють для виконання конкретних, спеціалізованих завдань. Це донастроювання здійснюється додатковим навчанням

на вузько спрямованих наборах даних, що відображають майбутні умови застосування. Наприклад, якщо ШІ має використовуватися в медицині, після загального попереднього навчання модель додатково тренують на медичних текстах, фахових статтях, клінічних звітах або навчальних матеріалах, щоб вона могла генерувати відповіді, які будуть релевантними для лікарів, професійно обґрунтованими та змістовно коректними. Таким чином із широко натренованої моделі виростає спеціалізований інструмент для певних сфер застосування.

Навчання нейронної мережі – це поступовий і ітеративний процес, що включає кілька послідовних кроків: підготовку даних, створення й визначення моделі, її навчання та тестування, а також подальше використання для виконання реальних завдань. Центральним елементом цього циклу є постійне, поступове поліпшення моделі з метою підвищення її точності та забезпечення здатності адаптуватися до змінних умов. Цей ітеративний характер означає, що під час і після першого впровадження постійно проводяться коригування та вдосконалення.

Процес зазвичай починається зі збору даних, оскільки якість і кількість даних становлять основу створення та продуктивності нейронної мережі. Зібрані дані мають бути достатньо об'ємними, щоб охопити різноманіття релевантних ситуацій, репрезентативними, щоб уникнути упереджень, і достатньо специфічними для завдань, які пізніше ставитимуться перед моделлю. Якщо, наприклад, цільове завдання – генерація тексту, збирають тексти з різних джерел і стилів – як-от газетні статті, наукові публікації, дописи на форумах чи художні твори – щоб забезпечити різноманітність контекстів, які допоможуть моделі робити надійні прогнози й генерувати багатофункціональні результати.

Потім модель проходить власне фазу навчання. У цьому процесі застосовуються спеціальні алгоритми навчання та процедури оптимізації, що дозволяють нейронній мережі виявляти й узагальнювати закономірності в наданих даних. Залежно від завдання це може охоплювати аналіз чисто текстових даних, зображень, аудіозаписів або мультимодального контенту. Головне

завдання цієї фази – створити модель, здатну передбачати або генерувати нові, ще не бачені результати на основі вивчених зв'язків у вхідних даних.

Під час навчання налаштовуються ваги, мінімізуються функції помилки та відстежуються метрики валідації, щоб уникнути перенавчання й забезпечити здатність до узагальнення. Після завершення основного навчання натреновану модель часто «упаковують». Цей крок включає оптимізацію структури моделі та її реалізації для ефективної роботи в практичних умовах. Упакування може означати оптимізацію під наявне апаратне забезпечення, наприклад через стиснення, квантування або обрізання (pruning), а також інтеграцію в існуючі робочі платформи або інтерфейси. Крім того, цей етап передбачає підготовку моделі для спеціальних випадків використання, щоб вона могла стабільно й продуктивно виконуватись у реальних застосуваннях із наявними обчислювальними ресурсами. Мета – знайти баланс між продуктивністю й ефективністю, щоб модель надійно працювала в продуктивних середовищах. Наступним важливим кроком є перевірка або валідація моделі.

На цьому етапі систематично оцінюють точність і відповідність прогнозів моделі вимогам. Для цього використовують окремі тестові набори даних, які не залучалися під час навчання, щоб об'єктивно встановити, чи може модель безпомилково виконувати свої функції. Валідація охоплює як кількісні оцінки – вимірювання точності, precision, recall чи інших релевантних метрик, – так і якісні оцінки, під час яких перевіряють, чи відповідають згенеровані результати бажаним вимогам щодо якості та безпеки. Ця валідація є вирішально важливою для запобігання небажаних або помилкових результатів при застосуванні до реальних даних. Без ретельного тестування існує ризик, що модель у виробничих умовах видаватиме хибні, упереджені або іншим чином невідповідні результати. Тому валідація є ключовою складовою для формування довіри до продуктивності та надійності моделі. Після підтвердження якості та досягнення прийнятних результатів валідації модель розгортають, тобто впроваджують у виробниче середовище. На цьому етапі модель фактично починає виконувати свої завдання: вона генерує тексти, зображення, аудіо або інші формати даних відповідно до

вимог конкретного застосування. Розгортання включає не лише надання технічної інфраструктури, а й встановлення процесів для логування, контролю доступу та регулярних перевірок.

Остання фаза життєвого циклу моделі – це безперервний моніторинг продуктивності моделі. Моніторинг охоплює постійне спостереження за її роботою в реальному часі, виявлення помилок або відхилень та оцінку відповідності результатів поточним вимогам. Це важливо, оскільки середовище, умови використання або базові дані можуть змінюватися з часом; через це колись добре працююча модель може втратити релевантність або точність. Тому потрібно своєчасно визначати, коли необхідні доопрацювання або коригування.

Коли виявляється потреба в покращенні, модель повторно навчають на нових даних або цілеспрямовано усувають раніше виявлені недоліки. Цей процес повторного навчання або адаптації забезпечує постійне вдосконалення і безперервну підгонку під специфічні умови експлуатації. Таким чином модель залишається працездатною та корисною протягом тривалого часу, оскільки може реагувати на змінені вимоги.

### **Донавчання нейромережі**

Донавчання нейромережі – це процес адаптації моделі до нових задач або уточнення її параметрів для покращення результатів на певних даних. Є кілька основних підходів до донавчання моделей, зокрема через System\_Message, Fine Tuning та RAG (Retrieval-Augmented Generation). Ось приклади кожного з підходів:

1. System\_Message – це метод надання контексту через системні повідомлення в межах однієї взаємодії. Йдеться про потужний інструмент для передавання підказок або контекстуальних інструкцій нейронній мережі без необхідності повторного навчання самої моделі. Цей метод застосовують, щоб уже на початку сесії задати моделі рамки або настанови щодо того, як мають виглядати наступні відповіді. System\_Message дозволяє визначати цілі, тональність або навіть конкретні поведінкові правила моделі під час взаємодії. Системне повідомлення можна використовувати для встановлення формальних

вимог, стилістичних вказівок або змістових пріоритетів, які модель повинна враховувати під час генерації. Наприклад, за допомогою системного повідомлення можна вказати, чи має модель відповідати формально й поділовому, або ж обрати більш невимушений, неформальний стиль. Також можна визначати ролі, наприклад, щоб модель діяла як експерт у певній галузі. Конкретним прикладом системного повідомлення може бути: `system_message = "Ви – висококваліфікований юрист, який надає детальні юридичні консультації. Це системне повідомлення не змінює внутрішні параметри моделі, проте передає їй контекст і очікування для поточної сесії. Завдяки цьому згенеровані відповіді зорієнтовані на відповідні цільові аудиторії та призначення. Якщо, наприклад, користувач ставить запитання про правові вимоги, модель завдяки цьому системному повідомленню буде спрямована на надання точних, професійно орієнтованих і офіційних відповідей. Перевага цього методу – адаптація моделі до різних контекстів без витратного за часом і ресурсами повторного навчання. Завдяки встановленню різних системних повідомлень її поведінку можна змінювати динамічно й залежно від ситуації, без суттєвих змін архітектури моделі або її параметрів. Недоліком є те, що системні повідомлення лише задають контекст для конкретної сесії або взаємодії, тоді як базова модель залишається незмінною. Це означає, що модель не вчиться назавжди новим концепціям і не адаптує свої внутрішні представлення відповідно. Також ускладнює роботу те, що управління великою кількістю різних системних повідомлень при великому обсязі запитів може бути складним і трудомістким, оскільки ці повідомлення необхідно постійно підтримувати й адаптувати до нових вимог, щоб запобігти виходу моделі за межі очікуваної поведінки.`

2. Fine Tuning – метод тонкого налаштування або донавчання моделі. Fine Tuning є процесом додаткового навчання попередньо навченої моделі за допомогою нових, спеціалізованих даних. Мета цього краща відповідність чітко визначеним вимогам, конкретним завданням або спеціальним доменам. У рамках донавчання початкове навчання моделі не замінюється, а продовжується; при цьому використовуються інші, для конкретного завдання більш релевантні набори

даних, які цілеспрямовано навчають модель додатковим знанням і тонкощам. Fine Tuning таким чином є цілеспрямованим кроком, що доповнює раніше здобуті загальні знання специфічною, часто фаховою інформацією, і підвищує продуктивність у вузько окреслених випадках застосування.

Наприклад: Припустимо, у вас є модель, яка спочатку була натренована на загальних текстових корпусах і тому володіє широкими базовими мовними навичками, але цю модель в майбутньому потрібно, щоб вона значно краще розуміла та обробляла медичні терміни, фахову лексику й особливості медичних текстів. У цьому випадку необхідно зібрати спеціалізований датасет, який міститиме відповідні медичні текстові документи. Такі дані можуть включати медичні наукові статті, інтерв'ю з лікарями, опис випадків, опис захворювань, звіти клінічних досліджень або інші релевантні медичні тексти. Після цього ці специфічні медичні дані використовують для подальшого навчання моделі, яка вже була попередньо натренована на загальних текстах. Завдяки цьому додатковому навчанні модель навчається правильно інтерпретувати медичну термінологію, розуміти контекст медичних висловлювань та надавати відповідні відповіді або класифікації в цій галузі. Приклад ілюструє типову послідовність: вибір і підготовка спеціалізованого корпусу, повторне навчання існуючої моделі з цими даними та подальша оцінка, чи принесла адаптація бажані покращення.

Серед переваг Fine Tuning – значне підвищення точності у конкретних завданнях, оскільки модель явно налаштовується під властивості й вимоги цих завдань. Донавчання також дозволяє моделі засвоїти нові концепти, фахові терміни та доменно-специфічні вирази, що підвищує її універсальність і придатність для складних завдань у спеціалізованих сферах. Через повторне або безперервне тонке налаштування модель може працювати ефективніше в довгостроковій перспективі, оскільки її постійно адаптують до нових даних і змінених вимог. Ще одна перевага – застосовність у спеціалізованих доменах, що вимагають високої точності та надійності, наприклад у медицині, праві чи техніці, де загальних знань зазвичай недостатньо. Така цілеспрямована адаптація суттєво

покращує продуктивність при специфічних задачах класифікації, розпізнавання або генерації.

Однак недоліки й обмеження Fine Tuning також є значними. Процедура зазвичай пов'язана з високими обчислювальними затратами та значними часовими витратами, оскільки повторне навчання складних моделей потребує інтенсивних ресурсів. Крім того, успішне донавчання часто вимагає великої кількості якісних, добре анотованих і репрезентативних даних; за відсутності достатнього обсягу даних або при недостатній їх якості Fine Tuning може бути неефективним або навіть шкідливим. Важливий ризик – переобучення (overfitting): якщо модель надто сильно підлаштувати під нерепрезентативний або занадто малий цільовий датасет, вона може втратити частину свого початкового загального знання та здатність виконувати широке коло завдань. Також Fine Tuning не забезпечує миттєвої адаптації в режимі реального часу: зміни вимог або розподілу даних не можна просто реалізувати шляхом негайної модифікації моделі «на ходу», зазвичай потрібне повторне навчання, що забирає час і ресурси. Ці недоліки необхідно ретельно зважувати при плануванні та виконанні тонкого налаштування.

3. RAG (Retrieval-Augmented Generation) – це метод, який поєднує генерацію тексту з пошуком релевантної інформації у зовнішніх базах даних або документах. За цього підходу модель не генерує текст виключно зі свого внутрішнього знання, а доповнює генерацію шляхом цілеспрямованого вилучення зовнішньої інформації з джерел, таких як документи, бази даних, архіви або вебсайти. Основний принцип полягає в тому, щоб спочатку знайти релевантні уривки або документи у зовнішній базі знань, а потім ці віднайдені матеріали використати як контекстуальну основу в процесі генерації. Таким чином модель може надавати відповіді або тексти, які базуються на конкретних зовнішніх джерелах, навіть якщо ця інформація не входила до початкового тренувального набору даних.

Це дозволяє моделі надавати точнішу й часто актуальнішу інформацію, оскільки вона за потреби звертається до зовнішніх джерел і інтегрує знайдені

факти у відповідь. Практичний приклад: ви хочете, щоб модель давала точні відомості про конкретні закони, юридичні тексти або наукові дослідження. У RAG-воркфлоу система спершу виконує пошук у відповідній базі даних або документному корбині (наприклад у збірці наукових статей або юридичних документів), визначає релевантні результати і потім використовує ці знайдені документи або витяги з них як основу для формулювання остаточної відповіді. Завдяки цьому генерація спирається на перевірені зовнішні джерела і може надавати детальніші, ближчі до джерела відповіді.

До сильних сторін RAG належить здатність надавати актуальні відповіді через використання зовнішніх інформаційних джерел, завдяки чому знання системи залишаються динамічними та релевантними без потреби в повному додатковому донавченні моделі. Система ефективно працює з великими обсягами даних, звертаючись за потреби до баз даних, документних сховищ або інтернет-ресурсів; це в багатьох випадках покращує якість і точність відповідей і одночасно зменшує потребу постійно перевчати модель. RAG також розширює ефективний обсяг знань моделі, оскільки вона отримує доступ до великих інформаційних масивів і може інтегрувати ширший контекст та оновлену інформацію у процес генерації.

Проте RAG залежить від якості та надійності використаних зовнішніх джерел: якщо посилання на документи застарілі, помилкові або погано куратовані, це може погіршити достовірність і правильність згенерованих відповідей. Також можуть виникати затримки, коли потрібні пошуки в реальному часі; особливо якщо підлягаючі бази даних або інфраструктури пошуку працюють не надто продуктивно, це може знизити швидкість відповіді. Проблеми масштабування є ще одним викликом: при дуже великому обсязі запитів або величезних корпусах даних інфраструктура може опинитися під навантаженням, що вимагатиме додаткових ресурсів і оптимізацій. Нарешті, критично важливо постійно оновлювати й підтримувати зовнішні джерела – якщо базові документи або бази даних не оновлюються, з часом існує ризик зниження надійності отриманих результатів.

## 2 ОГЛЯД ПРОБЛЕМ В УПРАВЛІННІ ІТ-ПРОЕКТАМИ

### 2.1 Проєктний менеджмент

Проєктний менеджмент (Project management) – розуміється як структурований набір знань, навичок, інструментів і методів – служить для реалізації цілей, корисних для людей та організацій. Це широке поле застосувань охоплює розробку програмного забезпечення для підвищення ефективності бізнес-процесів, будівництво будівель і інфраструктури, організацію допомоги постраждалим від природних катастроф або стратегічне розширення збуту на нові географічні ринки. Усі ці приклади можна розглядати як проєкти, у яких проєктний менеджмент відіграє центральну і часто вирішальну роль для досягнення успіху. Проєктний менеджмент суттєво відрізняється від загального менеджменту: однією з ключових ознак є фокус на досягненні конкретного кінцевого результату в межах чітко обмеженого проміжку часу. У той час як звичайне управління часто керує безперервними, постійними процесами, проєктний менеджмент спрямований спеціально на тимчасові заходи. Ця часово-обмеженість і орієнтація на ціль ставлять особливі вимоги до керівника проєкту, який має володіти широким спектром компетенцій. До них належать не лише фахові та технічні знання у відповідній домені, а й виражені лідерські навички, здатність вирішувати конфлікти та глибоке розуміння бізнес-цілей і організаційного контексту, в якому реалізується проєкт.

Проєкт сам по собі визначається як тимчасова ініціатива, що має на меті створення корисності. Це досягається шляхом розробки унікальних продуктів, надання специфічних послуг або покращення існуючих процесів. Проєкти можуть істотно відрізнятися один від одного: деякі вирішують негайну, термінову проблему і реалізуються в короткі строки, інші є складнішими і потребують тривалішого часу. Прикладом можуть бути розробка індивідуальних програмних рішень або планування й реалізація заходів із подолання наслідків природних катастроф. Іншою типовою ознакою проєкту є його чіткий життєвий цикл, який зазвичай складається з кількох послідовних фаз. Базові фази – ініціація,

планування та виконання; часто також включають стадії моніторингу та закриття, що завершують життєвий цикл. Цей цикл допомагає керівникам проєктів структурувати завдання, ефективно використовувати ресурси та систематично планувати й здійснювати робочі етапи. Завдяки фазовій моделі можна вимірювати прогрес, визначати точки передачі та краще керувати ризиками й залежностями. Підсумовуючи, проєкт – це добре спланована діяльність з визначеним початком і кінцем, яка зазвичай включає команду, бюджет і графік та спрямована на створення вимірюваної користі [1].

Отже, проєктний менеджмент – це системний підхід до планування, організації, керування і контролю виконання проєкту. Він застосовується для забезпечення досягнення цілей проєкту та виконання окремих завдань у встановлені строки і в межах запланованого бюджету. Йдеться не лише про виконання завдань, а про свідоме управління всім заходом з метою узгодження термінів, витрат і якості. Проєктний менеджмент охоплює як операційні, так і стратегічні аспекти: на операційному рівні плануються пакети робіт, розподіляються ресурси і відслідковується щоденний прогрес; на стратегічному рівні цілі проєкту зіставляються з вищими цілями організації, а очікування зацікавлених сторін регулюються.

Фундамент проєктного менеджменту – розуміння природи проєкту і факторів, що визначають його успіх. Проєктний менеджмент стосується всіх аспектів проєкту, включно з чітким визначенням цілей, обсягом робіт (scope), необхідними ресурсами, ідентифікацією та оцінкою ризиків, внутрішньою та зовнішньою комунікацією, а також іншими факторами, що можуть впливати на успіх проєкту. Для подолання цієї різноманітності використовують методики, інструменти та визначені процеси, які забезпечують ефективне планування, чіткий розподіл ролей і відповідальностей, безперервний моніторинг прогресу та цілеспрямоване вирішення проблем. Важливою складовою є також впровадження звітних і комунікаційних механізмів, що слугують для інформування проєктної команди та відповідних зацікавлених сторін про прогрес, зміни та ризики. Такі

механізми створюють прозорість, полегшують процеси прийняття рішень і сприяють своєчасному виявленню та вирішенню проблем.

Методики проєктного менеджменту – це концепції й набори знань, перевірених підходів і інструментів, які цілеспрямовано використовуються для планування, організації та керування проєктами. Вони пропонують структуровані рамки й стандарти, що підтримують команди проєктів у ефективному виконанні проєктів та досягненні визначених цілей. Ці методики дають орієнтири щодо процедур, ролей, процесів і вимог до документації і можуть, за потреби, адаптуватися до специфічних вимог проєкту або галузі [2].

Декілька найбільш популярних методологій управління проєктами:

1. Waterfall (водоспадна методика): це традиційний послідовний підхід, за якого проєкт розбивається на чітко відмежовані фази, і кожна фаза починається лише після завершення попередньої. Водоспадна методика особливо підходить для проєктів зі стабільними, добре визначеними вимогами, коли після початку проєкту мало очікується змін. Вона забезпечує структуроване планування й документацію, а також чіткі точки приймання наприкінці кожної фази.

2. Agile (гнучкий підхід): Agile – ітеративна методика, яка ділить проєкти на короткі повторювані цикли розробки – так звані спринти. Команда працює невеликими інкрементальними кроками над частинами проєкту, що дозволяє під час роботи уточнювати або змінювати вимоги. Гнучка робота сприяє тісній співпраці зі стейкхолдерами, швидким зворотним зв'язкам і високій гнучкості щодо змін. Серед популярних гнучких методів – Scrum, Kanban і Extreme Programming (XP), кожен із яких визначає спеціальні практики й ролі для підтримки ітеративного та адаптивного підходу.

3. Prince2: Prince2 – методика, розроблена у Великій Британії, яка має дуже структурований підхід і надає детальні принципи та процеси для управління проєктами. Вона чітко визначає ролі та відповідальності, описує фази проєкту і необхідні керівні документи. Prince2 модульна за структурою і може адаптуватися до різних типів проєктів і розмірів команд, завдяки чому її застосовують як у невеликих ініціативах, так і в комплексних програмах.

4. PMBOK (Project Management Body of Knowledge): PMBOK – стандарт і рамка, розроблені Інститутом управління проєктами (PMI). Він узагальнює набір процесів, знань і перевірених практик проєктного менеджменту. PMBOK пропонує загальну міжгалузеву основу, що системно описує основні області знань проєктного менеджменту – такі як інтеграція, обсяг, час, витрати, якість, персонал, комунікації, ризики і закупівлі. Ця рамка слугує орієнтиром для керівників проєктів і організацій, які прагнуть встановити стандартизовані процеси та спільну мову в проєктному менеджменті.

Це лише деякі приклади методів управління проєктами; існує багато інших підходів і варіантів, які можуть застосовуватися залежно від конкретних потреб, середовища та типу проєкту. Залежно від галузі, корпоративної культури та мети проєкту розглядаються різні методи, тож відповідальні за проєкт можуть обирати з широкого репертуару. Вибір відповідної методології тому залежить від конкретних вимог, розміру та складності проєкту, а також від уподобань, досвіду та компетенцій команди проєкту. На це рішення істотно впливають також зовнішні умови, такі як тиск часу, бюджетні обмеження і очікування зацікавлених сторін.

Під час роботи над проєктом керівник проєкту має не лише контролювати виконання завдань і дотримання графіка, а й постійно приймати важливі рішення для вирішення проблем, які виникають протягом роботи над проєктом. Окрім контролю часу, якості та витрат, це також включає управління ризиками, координацію команди та комунікацію з усіма зацікавленими сторонами. Щоб мати змогу приймати обґрунтовані рішення та ефективно долати труднощі, у розпорядженні керівника проєкту є різні допоміжні методи та інструменти; деякі з цих засобів наведені нижче:

1. Аналіз Парето – цей інструмент допомагає пріоритезувати проблеми або завдання за їхнім впливом, зосереджуючи увагу на тих небагатьох факторах, які спричиняють найбільший ефект. Аналіз Парето дозволяє з обмеженими ресурсами цілеспрямовано боротися з причинами, що приносять найбільшу користь, і таким чином підвищувати ефективність вжитих заходів.

2. «Шість капелюхів» Едварда де Боно – цей метод дає змогу систематично розглянути проблему з різних точок зору та сприяє структурованому й креативному мисленню. Приймання різних мисленнєвих позицій робить процес прийняття рішень багатограннішим і збалансованішим, що особливо корисно при складних запитаннях.

3. Майндмепінг (інтелектуальні карти) – ця техніка візуалізує думки й ідеї, завдяки чому краще видно зв'язки, пріоритети та альтернативи. Майндмепи допомагають структурувати інформацію та полегшують генерування варіантів, що, у свою чергу, сприяє прийняттю рішень.

Важливо завжди пам'ятати, що кожна ситуація й кожен проєкт унікальні, і застосування цих інструментів вимагає індивідуального, контекстуально зумовленого підходу. Жоден інструмент не підходить універсально; радше методи й засоби слід гнучко адаптувати до конкретної проблеми. Тому керівник проєкту повинен мати досвід, здатність до судження й методичні навички, щоб у конкретній ситуації визначити, який інструмент або яка комбінація інструментів буде найефективнішою. Лише так можна приймати рішення, що враховують конкретний контекст проєкту й призводять до успішних результатів.

## **2.2 Напрями застосування GAI в управлінні IT-проєктами**

Керівники проєктів використовують генеративний штучний інтелект, щоб усунути рутинні адміністративні завдання та швидше приймати рішення на основі даних. Технологія обробляє все: від створення документації до прогнозування ризиків.

Ці інструменти штучного інтелекту інтегруються безпосередньо в існуючі робочі процеси проєктів. Вони надають аналітику в режимі реального часу та автоматизують трудомісткі процеси, які раніше вимагали ручної роботи.

Основні застосування GAI подані на рис.2.1 [14].

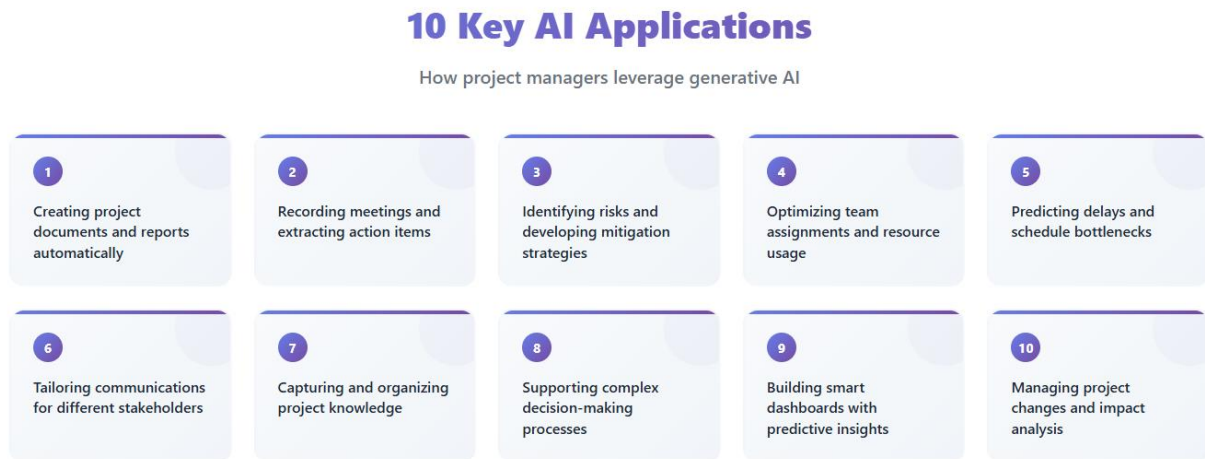


Рисунок 2.1 - Напрями застосування GAI в управлінні ІТ-проектами

### 1. Автоматизоване створення документації

Попрощайтесь з годинами, витраченими на складання проектної документації. Генеративний штучний інтелект тепер може автоматично створювати:

- Комплексні плани проектів, засновані на мінімальних вхідних параметрах
- Детальні технічні специфікації з простих технічних описів проекту
- Автоматизовані звіти про стан , які збирають дані з кількох джерел
- Користувацькі шаблони для потреб у постійній документації

Впроваджуючи інструменти, які автоматично генерують документацію, керівники проектів можуть зменшити адміністративні накладні витрати до 70%.

### 2. Підсумки зустрічей та вилучення пунктів завдання

Зустрічі є важливими, але можуть забирати дорогоцінний час. Інструменти на базі штучного інтелекту дозволяють:

- Транскрипція зустрічей проекту в режимі реального часу
- Автоматичне визначення ключових рішень, прийнятих під час обговорень
- Вилучення та призначення завдань членам команди
- Створення архівів зустрічей з можливістю пошуку

Використання штучного інтелекту в управлінні проектами для транскрипції зустрічей та відстеження дій гарантує, що жодна деталь не пройде повз увагу, і покращує злагодженість роботи команди.

### 3. Визначення ризиків та планування їх пом'якшення

Проактивне управління ризиками відділяє успішні проекти від невдалих. Генеративний штучний інтелект пропонує:

- Прогнозний аналіз потенційних ризиків на основі історичних даних проекту
- Створення комплексних реєстрів ризиків з оцінками ймовірності та впливу
- Індивідуальні стратегії пом'якшення наслідків, адаптовані до конкретних контекстів проекту
- Постійний моніторинг факторів ризику протягом усього життєвого циклу проекту

Аналізуючи закономірності тисяч подібних проектів, ці інструменти можуть виявляти ризики, які менеджери-люди можуть пропустити, забезпечуючи вирішальну систему безпеки для складних ініціатив.

### 4. Оптимізація розподілу ресурсів

Знаходження ідеального балансу ресурсів між проектами залишається одним із найбільших викликів для керівництва. Передові інструменти штучного інтелекту можуть:

- Аналізувати можливості та доступність ресурсів у вашій організації
- Рекомендувати оптимальний розподіл ресурсів на основі навичок та досвіду
- Моделювання різних сценаріїв розподілу для визначення найефективнішого підходу
- Прогнозувати вузькі місця в ресурсах, перш ніж вони вплинуть на терміни

Провідна будівельна фірма повідомила про 22% підвищення ефективності проектів після впровадження інструментів оптимізації ресурсів на основі штучного інтелекту, що призвело до мільйонної щорічної економії.

#### 5. Прогнозування розкладу та прогнозування вузьких місць

Терміни виконання проектів рідко йдуть точно за планом. Генеративний штучний інтелект може допомогти:

- Створення динамічних графіків , що адаптуються до змінних умов
- Прогнозування ймовірних затримок на основі історичних даних про ефективність
- Заздалегідь виявляйте потенційні вузькі місця
- Рекомендації щодо коригування графіка для забезпечення виконання проектів у межах графіка

Ці можливості дозволяють керувати графіком на випередження , а не реагувати на пожежогасіння, що значно знижує стрес, пов'язаний із зривом термінів.

#### 6. Покращення комунікації із зацікавленими сторонами

Чітка комунікація сприяє успішним проектам. Практичне застосування генеративного штучного інтелекту для керівників проектів включає:

- Генерація адаптованих оновлень зацікавлених сторін на основі ролей та інтересів одержувачів
- Створення переконливих візуалізацій , що відображають складні дані проєкту
- Переклад технічної інформації доступною мовою для різних аудиторій
- Аналіз настроїв відповідей зацікавлених сторін для спрямування комунікаційних стратегій

Ці інструменти гарантують, що кожна зацікавлена сторона отримує інформацію у форматі, який є для неї найбільш корисним, зміцнюючи стосунки та будуючи довіру протягом усього життєвого циклу проєкту.

#### 7. Управління знаннями та інституційна пам'ять

Проекти генерують цінні знання, які часто зникають, коли члени команди йдуть далі. Системи управління знаннями на базі штучного інтелекту можуть:

- Автоматично фіксувати та впорядковувати уроки, отримані під час виконання проекту
- Створіть репозиторії досвіду проектів та передового досвіду з можливістю пошуку
- Створюйте навчальні матеріали на основі накопичених знань про проект
- Збереження інституційної пам'яті під час організаційних змін

Цей застосунок гарантує, що важко здобуті висновки принесуть користь майбутнім ініціативам, створюючи благотворний цикл постійного вдосконалення.

#### 8. Системи підтримки рішень

Керівники проектів щодня стикаються зі складними рішеннями. Генеративний штучний інтелект може забезпечити:

- Рекомендації на основі даних для складних проектних рішень
- Моделювання кількох результатів прийняття рішень для оцінки потенційних шляхів
- Виявлення факторів, які могли бути пропущені в традиційному аналізі
- Підтримка рішень у режимі реального часу на критичних етапах проекту

Як зазначив один директор програми: «Наявність штучного інтелекту як партнера по мисленню не замінює моє судження; він посилює його, гарантуючи, що я враховую всі відповідні фактори, перш ніж приймати рішення».

#### 9. Інтелектуальні панелі інструментів проекту

Інформаційне перевантаження може паралізувати керівників проектів. Панелі інструментів на базі штучного інтелекту можуть створювати:

- Налаштовані інформаційні панелі , які виділяють найважливіші показники для кожного глядача
- Пояснення природною мовою складних тенденцій даних

- Прогностичні індикатори , що позначають потенційні проблеми ще до того, як вони з'являться в даних
- Інтерактивні візуалізації , що дозволяють інтуїтивно досліджувати стан проекту

Ці інтелектуальні інтерфейси, що стали можливими завдяки використанню штучного інтелекту в управлінні проектами, перетворюють необроблені дані на практичні висновки для швидшого прийняття рішень.

## 10. Допомога в управлінні змінами

Зміни в проектах неминучі, але ефективне управління ними є складним завданням. Генеративний штучний інтелект пропонує:

- Аналіз впливу запропонованих змін за всіма вимірами проекту
- Генерація вичерпної документації щодо запитів на зміни
- Шаблони комунікації для пояснення змін зацікавленим сторонам, які за них задіяні.
- Стратегії адаптації для мінімізації збоїв, спричинених необхідними змінами

Завдяки оптимізації процесів управління змінами, ці інструменти допомагають проектам залишатися гнучкими, не жертвуючи контролем чи прозорістю.

Пам'ятайте, що генеративний ШІ для керівників проектів розроблений для розширення людських можливостей, а не для їх заміни.

Найефективнішими керівниками проектів будуть ті, хто навчається співпрацювати зі штучним інтелектом, використовуючи його для виконання рутинних завдань, зосереджуючи свою людську креативність та судження на складних викликах, які дійсно потребують людського втручання.

### 2.3 Аналіз наявних проблем у сфері управління IT-проектами

Управління проектами вже тривалий час є дуже популярною сферою в IT; через це посада керівника проекту масово поширилася, і одночасно знизився поріг входження в цю галузь. Це має як переваги, так і недоліки, які ми розглянемо далі.

1. Останніми роками спостерігається тенденція, що багато людей із різних професій переходять в IT. Окрім класичних шляхів зміни кваліфікації, таких як здобуття вищої освіти в новій галузі, із поширенням інтернету значно збільшилася кількість онлайн-курсів та шкіл, які пропонують здобути нову кваліфікацію значно швидше. Обсяг матеріалу, що розглядається на таких курсах, однак обмежений, тому випускникам таких шкіл часто бракує глибокого розуміння обраного напрямку. Коли людина з такою підготовкою обіймає посаду керівника проекту, вона часто стикається з «бар'єрами», переважно з боку розробників.. Низький рівень кваліфікації керівника також означає, що звичні завдання будуть рутинними й одноманітними та не вимагатимуть творчого підходу; через це навіть така задача стає викликом для працівника, що безпосередньо впливає на його мотивацію до роботи та загальний емоційний стан.

2. У проектах часто бракує достатньої деталізації, зокрема. Відсутність деталізації є причиною багатьох наступних проблем. Центральною ознакою є відсутність адекватного розподілу ресурсів, що призводить до неефективності та ускладнює відстеження використання ресурсів відносно виконаної роботи. Неправильні оцінки часто є наслідком недостатньо чітко визначених робіт, що веде до нереалістичних планів і фаз проекту. [11].

3. Нереалістичні графіки виконання спричиняють значні проблеми в управлінні проектами й часто призводять до додаткових витрат, пропущених проміжних етапів проекту та постійного перегляду показників ефективності. Явною ознакою проблемного початково встановленого графіка є надмірні витрати. Пропущені проміжні дедлайни – ще одна поширена ознака того, що проект відстає від графіка. Крім того, спостерігається постійний перегляд методів

вимірювання продуктивності як наслідок затримок у графіку або необхідності модернізації функції контролю проекту.

Проблема нереалістичних графіків має кілька причин. Конкурентний тиск часто змушує учасників пропонувати надто амбітні терміни, щоб виграти замовлення. Недостатнє планування, що опирається на минулий досвід також є причиною, адже організації, які не ведуть бази даних про терміни та витрати, перебувають у невідповідній ситуації. Занадто мало часу на складання графіків і аналіз релевантних факторів може призвести до надто оптимістичного планування [11].

4. Недостатній контроль витрат і калькуляція ускладнюють порівняння бюджету та фактичних витрат, призводять до неточного розподілу витрат і спотворюють фінансову звітність. Ефективний контроль витрат необхідний для дотримання бюджету, прозорості та обґрунтованих рішень. Причинами є відсутність загальнозастосовних інформаційних систем для мультипроектного управління, слабе поєднання визначення обсягів робіт, оцінювання і накопичення витрат, низька обізнаність персоналу про методи контролю, а також недостатні механізми розподілу.

5. Генеративний ШІ може автоматизувати рутинні завдання, такі як планування, розподіл ресурсів і аналіз ризиків, оптимізуючи процеси. Водночас людське прийняття рішень залишається центральним: менеджери проєктів вносять навички вирішення проблем, креативність і соціальні компетенції, які ШІ сам по собі не замінить. Дослідження підкреслюють доповнювальну роль ШІ та людського інтелекту для стійкого управління проєктами. Наразі лише близько 35 % проєктів є успішними; однак автори очікують, що зростаюче використання ШІ, машинного навчання та інших технологій покращить продуктивність, якість даних і процеси прийняття рішень та змінить роль менеджерів проєктів.

6. Динаміка взаємодії людини та штучного інтелекту в управлінні проєктами. Для глибшого розуміння ефективності генеративного штучного інтелекту та людської оцінки в управлінні проєктами були проведені порівняльні дослідження, які представляють підхід, заснований на даних, до управління

процесами в реальному часі в складному інженерному проектуванні, враховуючи як штучний інтелект, так і залучення людини. Дослідження показує, як генеративні системи штучного інтелекту можуть сприяти прийняттю рішень та оптимізації процесів в управлінні проектами, а також визнає унікальні сильні сторони та знання, що привносить менеджер проектів-людина.

Завдяки порівняльному аналізу ми можемо краще зрозуміти контексти, в яких генеративний ШІ або людська експертиза переважають, та визначити потенційну синергію для ефективних практик управління проектами.

Хоча деякі автори стверджують, що ШІ не може замінити людський розум, довіра та співпраця між системами ШІ та менеджерами проектів-людей є критично важливими для успішної співпраці в управлінні проектами.

Розуміння динаміки довіри та співпраці між людьми та ШІ в управлінні проектами є важливим для ефективного використання генеративних технологій ШІ. Вивчаючи результати цього дослідження, ми можемо отримати уявлення про те, як можна зміцнити довіру до систем ШІ та як можна оптимізувати співпрацю між людьми та ШІ для максимізації результатів проекту.

## **2.4 Переваги і недоліки GAI в управління проектами**

В роботі [12] приводяться результати дослідження сприйняття респондентів з проблем застосування GAI в управління проектами. 52 учасники цього дослідження обіймали різноманітні посади, причому найбільшу групу становили керівники проектів (11 учасників), за ними йшли планувальники проектів (3), головні виконавчі директори (3) та керівники програм (3), серед інших. Щодо типів роботодавців, багато учасників ідентифікували себе як консультанти (23), далі йшли державні підрядники та працівники (19), галузеві підрядники (6) та виробники (4). Розміри фірм, у яких працювали учасники, варіювалися: 32 учасники працювали у великих організаціях з понад 500 співробітниками, тоді як 20 учасників працювали у відносно менших фірмах. Учасники представляли різні унікальні сектори, причому найбільше домінував сектор інформаційних

технологій (ІТ) (13), далі йшли аерокосмічна галузь (5), будівництво (4), оборона (3), наука (3) та різні комбінації цих секторів.

Оцінюючи вплив штучного інтелекту на корпоративні цінності, 58% учасників не відзначили суттєвих змін, тоді як 42% повідомили про позитивні зміни у відповіді на бінарне запитання (Так/Ні) про те, чи змінилося бачення, місія чи стратегічний план їхньої фірми завдяки розвитку штучного інтелекту. Щодо сприйняття співробітниками застосування штучного інтелекту в управлінні проектами, більшість (74%) висловили змішані або негативні почуття, і лише 26% повідомили про позитивні настрої. Ці учасники представляли різний рівень досвіду роботи в галузі, в середньому 17,12 років. Якщо врахувати різницю між високим і низьким рівнями досвіду на основі 25-річного інтервалу (Aramali et al., 2021), 58% учасників належать до категорії з високим досвідом, тоді як 42% належать до категорії з низьким досвідом. Більшість учасників, 65%, були представниками приватних організацій, тоді як решта 35% представляли державні установи. Крім того, щодо використання ChatGPT або аналогічного генеративного інструменту штучного інтелекту, ми виявили, що 73% учасників активно використовували ці інструменти на момент збору результатів опитування, тоді як решта 27% знали про них та висловлювали зацікавленість у їх використанні. Цей профіль учасників відображає різні аспекти професійного досвіду в галузі управління проектами та штучного інтелекту.

На рис. 2.2 представлені отримані результати щодо сприйняття учасників придатності ChatGPT або інших подібних генеративних інструментів штучного інтелекту для різних процесів управління проектами та галузей знань.

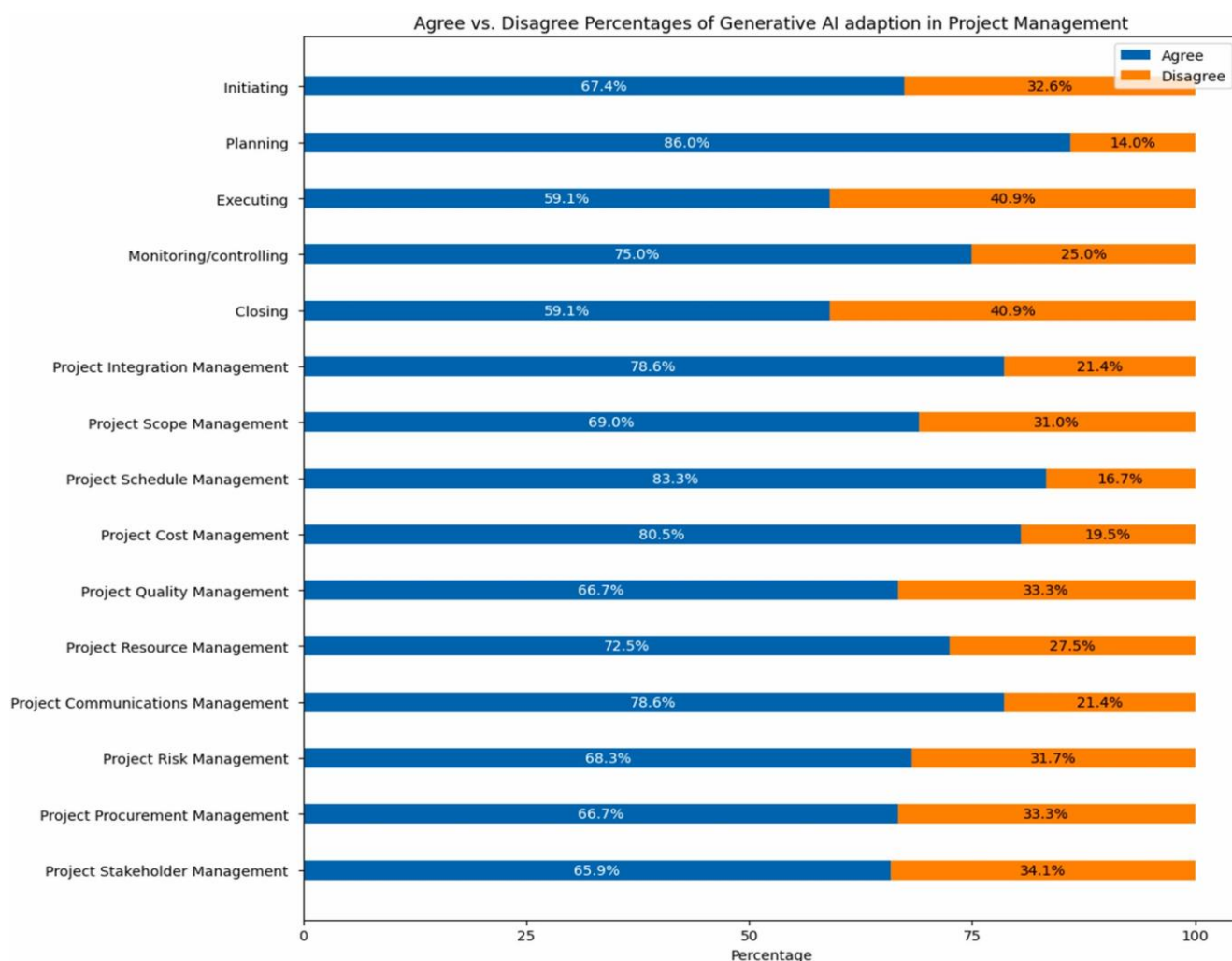


Рисунок 2.2 - Впровадження штучного інтелекту в управлінні проектами

Стовпчаста діаграма на цьому рисунку ілюструє відсотки практиків, які погоджуються із застосовністю ШІ для конкретних груп процесів управління проектами або галузей знань, порівняно з тими, хто не погоджувався. У групі ініціювання процесу 67% погодилися із застосовністю інструментів ШІ, тоді як відсотки, що спостерігалися для наступних груп процесів, становили 86% для планування, 59% для виконання, 75% для моніторингу та контролю, і, нарешті, 59% для груп процесів закриття. Ці відсотки свідчать про консенсус серед учасників щодо потенційної корисності інструментів ШІ на різних етапах управління проектами. Ця широка згода свідчить про те, що інтеграція інструментів штучного інтелекту в процеси управління проектами може підвищити ефективність та результативність. Більше того, оцінюючи конкретні сфери, такі як інтеграція проектів, обсяг, графік, вартість, якість, ресурси, комунікації, ризики, закупівлі та управління зацікавленими сторонами, більшість

учасників послідовно підтримували допомогу ШІ, з відсотками від 66% до 83%. Аналогічно, учасники висловлювали свою думку щодо того, якою мірою генеративні інструменти ШІ можуть допомогти різним посадам у сфері управління проектами (рис. 2.3).

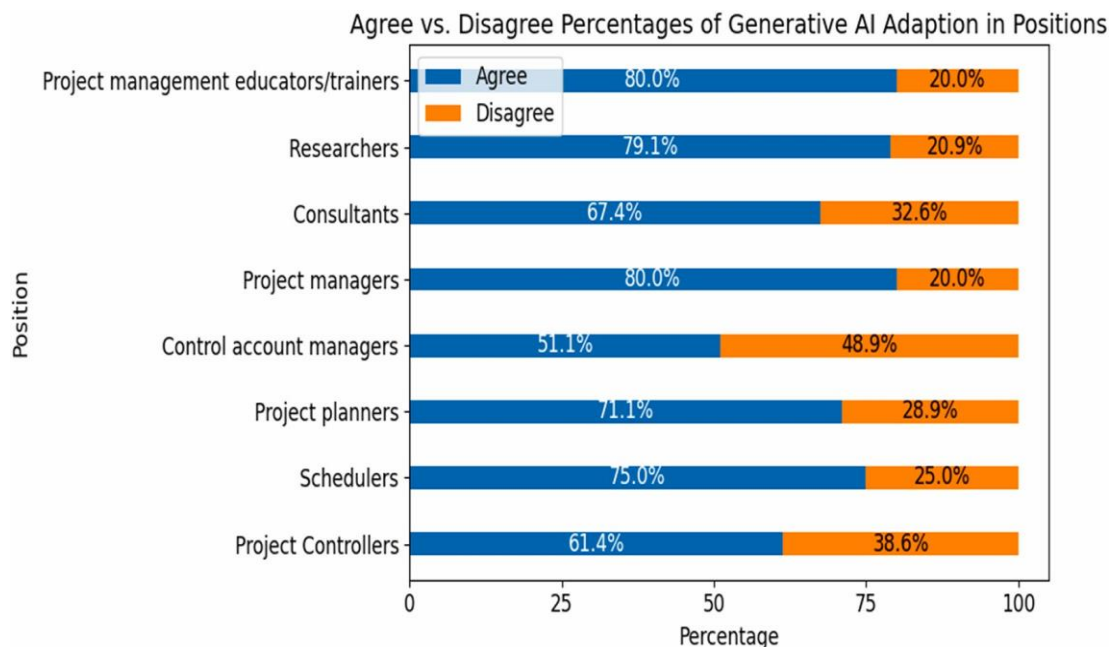


Рисунок 2.3 - Впровадження генеративного штучного інтелекту на посадах керівників проектів

Загалом, спостерігалася позитивна схильність до допомоги ШІ, з відсотками від 51% до 80%. Зокрема, керівники проектів (80%) та викладачі/тренери з управління проектами (80%) висловили найвищу підтримку допомозі ШІ, що свідчить про тверду віру в потенційні переваги ШІ на цих посадах. Нарешті, коли учасників запитали про очікуваний вплив ШІ на продуктивність, 52% вважали, що ШІ може призвести до підвищення продуктивності понад 20%, тоді як 47% були більш скептично налаштовані, передбачаючи незначне або взагалі нульове підвищення продуктивності. Цей поділ відображає спектр рівнів оптимізму серед учасників щодо трансформаційного потенціалу штучного інтелекту в практиці управління проектами. На рис. 2.4 представлено дані опитування щодо сприйняття фахівцями ШІ на робочому місці, зокрема щодо динаміки співпраці та продуктивності.

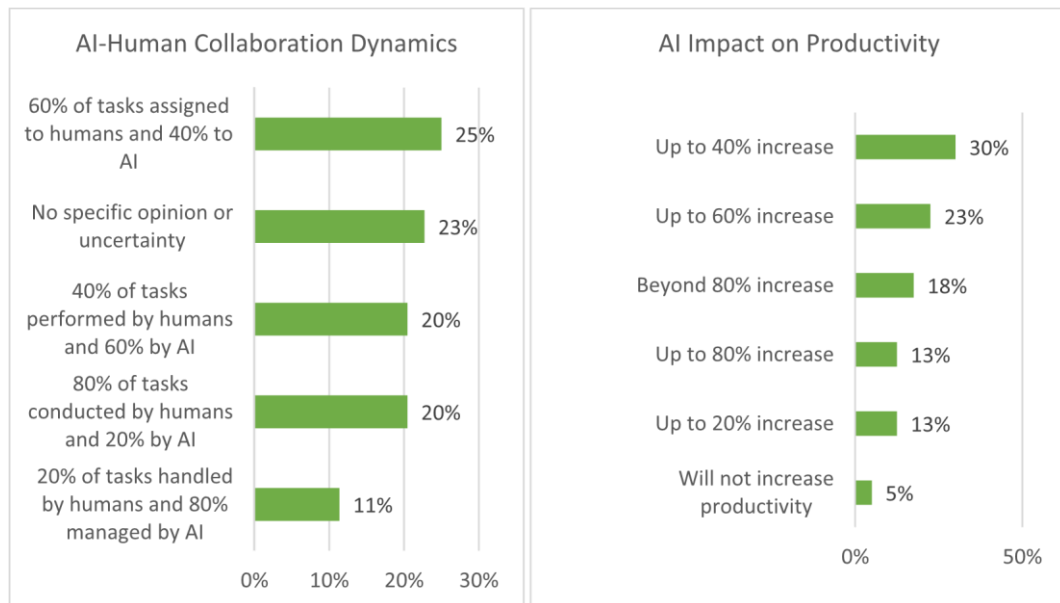


Рисунок 2.4 - Генеративний ШІ проти людей

Перша діаграма (праворуч) демонструє збалансований погляд на співпрацю за допомогою ШІ, з перевагою помірному розподілу завдань між людиною та ШІ (25%) та значним відсотком респондентів (23%), які не визначилися щодо ролі ШІ. Друга діаграма (ліворуч) відображає цю помірковану позицію, причому найбільша група (30%) очікує, що ШІ сприятиме збільшенню продуктивності до 40%. Загалом, обидві діаграми демонструють оптимізм, але також обережне ставлення до генеративного використання ШІ в процесах управління проектами, визнаючи його потенціал для підвищення ефективності робочого місця, а також визнаючи поточні обмеження та невизначеності, пов'язані з його інтеграцією. На рис. 5 представлено дані опитування щодо сприйняття фахівцями ШІ на робочому місці, зокрема щодо динаміки співпраці та продуктивності. Перша діаграма (праворуч) демонструє збалансований погляд на співпрацю за допомогою ШІ, з перевагою помірному розподілу завдань між людиною та ШІ (25%) та значним відсотком респондентів (23%), які не визначилися щодо ролі ШІ. Друга діаграма (ліворуч) відображає цю помірковану позицію, причому найбільша група (30%) очікує, що ШІ сприятиме збільшенню продуктивності до 40%. Загалом, обидві діаграми демонструють оптимізм, але також обережне ставлення до генеративного використання ШІ в процесах управління проектами, визнаючи його

потенціал для підвищення ефективності робочого місця, а також визнаючи поточні обмеження та невизначеності, пов'язані з його інтеграцією.

## 2.5 Програмні засоби управління ІТ-проектами

Штучний інтелект призначено для того, щоб сприяти прийняттю рішень, особливо коли залучені дані, параметри та змінні настільки великі або складні, що вони виходять за межі безпосереднього людського розуміння. У таких випадках ШІ може допомогти виявляти зв'язки та давати рекомендації, які людям самотійно було б важко вивести. Зазвичай системи ШІ в межах заданих обмежень приймають правильні або принаймні дуже корисні рішення, оскільки вони послідовно й швидко обробляють великі обсяги даних. Засоби на базі ШІ допомагають менеджерам та керівникам визначати пріоритети та приймати обґрунтовані рішення на кожному етапі проєкту – від першого планування через оперативну реалізацію до підсумкового контролю. Вони особливо корисні для обробки великих обсягів даних про проєкт, виявлення шаблонів і залежностей та отримання практичних висновків, які можна застосувати в повсякденній роботі над проєктом. З огляду на це варто розглянути існуючі розробки в галузі управління проєктами, що базуються на генеративному ШІ або інтегрують відповідні функції ШІ.

1. **Wrike** Типові можливості включають управління завданнями, облік робочого часу, а також різні інструменти для сприяння командній роботі та комунікації. Wrike також використовує штучний інтелект, щоб надавати проєктним керівникам конкретну допомогу, наприклад автоматизуючи рутинні й повторювані завдання, покращуючи внутрішню комунікацію та роблячи робочі процеси загалом ефективнішими. Завдяки так званому рішенню Work Intelligence керівники проєктів можуть пріоритезувати найважливішу роботу, адже Wrike надає автоматичні рекомендації щодо пріоритезації завдань і зосередження на пріоритетних робочих пакетах. До функцій, що підтримуються ШІ у Wrike, належать, зокрема, інтелектуальна пріоритизація завдань, управління

навантаженням для збалансування командних потужностей та розподіл ресурсів, що має допомагати уникати вузьких місць. Завдяки цим автоматизаціям і аналізам проєктні керівники у Wrike можуть заощаджувати час і зменшувати ймовірність помилок, оскільки системно виконуються рутинні завдання, такі як планування термінів, встановлення дедлайнів або відправлення нагадувань, і тому потрібно менше ручного втручання.

2. **Proggio** - пропонує численні подібні функції. Стандартні можливості Proggio включають управління завданнями, облік робочого часу, а також різні інструменти для командної співпраці та комунікації. Proggio також використовує штучний інтелект для автоматизації повторюваних рутинних завдань, покращення внутрішньої комунікації та оптимізації робочих процесів. За допомогою функцій ШІ Proggio керівники проєктів можуть ідентифікувати та пріоритезувати найважливіші завдання, забезпечуючи, щоб їхні команди спрямовували зусилля на найцінніші види діяльності. Цим слугують автоматизовані рекомендації Proggio щодо пріоритезації завдань, які дають керівникам проєктів конкретні пропозиції [6].

AI-функції Proggio можуть допомогти керівнику проєктів економити час та зменшувати помилки за рахунок автоматизації рутинних завдань, таких як планування зустрічей та нагадування [7]. Функція інтелектуального пріоритезування завдань Proggio допомагає керівникам проєктів визначати найважливішу роботу, аналізуючи дані з минулих проєктів та надаючи уявлення про те, які завдання найцінніші. Ця функція може допомогти менеджерам проєктів гарантувати, що їхня команда націлена на найціннішу роботу з рекомендаціями щодо визначення пріоритетів автоматизованих завдань Proggio.

3. **Forecast** – надає можливості співробітникам і використовувати агреговані дані з метою сталого покращення як бізнес-результатів, так і успіху клієнтів. Платформа застосовує ШІ для «читання» та інтеграції великих обсягів інформації з різних корпоративних застосунків. Завдяки цій інтеграції формується розширена загальна картина конкретного проєкту та можливих результатів і сценаріїв, що дає проєктним командам більш обґрунтовану основу для ухвалення рішень [8].

Forecast обробляє дані з різних джерел у межах компанії та пов'язує цю інформацію так, щоб приховані зв'язки й релевантні взаємозв'язки між завданнями, графіками та ресурсами ставали видимими.

До функцій Forecast на базі ШІ належить підтримка в реальному часі при плануванні й виконанні проєктів, здатність раннього виявлення потенційних ризиків і точного інформування про те, коли щось іде не так, а також механізми навчання на основі минулих проєктів для постійного покращення виконання майбутніх ініціатив. Ці функції мають на меті не лише надавати менеджерам проєктів підказки й попередження, а й конкретні пропозиції щодо коригування планів або перерозподілу ресурсів. Автоматизувавши ці адміністративні дії, керівники проєктів можуть більше зосередитися на стратегічних завданнях.

Ще одне важливе поле застосування ШІ в Forecast – пріоритизація завдань. Аналізуючи дані попередніх проєктів, Forecast може надавати менеджерам проєктів рекомендації щодо того, які завдання мають найбільший вплив на успіх проєкту і тому повинні виконуватися першочергово [9]. Ці дані щириво-обґрунтовані рекомендації надаються у вигляді автоматизованого списку завдань, який допомагає менеджерам переконатися, що їхня команда працює над найважливішими та найбільш ціннісними завданнями. Таким чином Forecast підтримує не лише операційне виконання, а й сприяє фокусуванню на стратегічно важливих завданнях.

4. **Asana** – пропонує широкий набір функцій, зокрема управління завданнями, облік часу та різні інструменти командної взаємодії. Функції на базі ШІ в Asana спрямовані на спрощення процесів і автоматичне надання менеджерам пріоритетних рекомендацій, щоб їхня увага була скерована на найважливіші завдання [10].

До конкретних ШІ-підтримуваних функцій Asana належать інтелектуальна пріоритизація завдань, управління навантаженням і розподіл ресурсів. Ці аналізи дозволяють виявляти закономірності у витратах часу, вузьких місцях і повторюваних проблемах, що сприяє уникненню аналогічних труднощів у нових проєктах. Автоматизуючи рутинні операції, такі як планування зустрічей і

надсилання нагадувань, менеджери проєктів також економлять час і зменшують кількість помилок.

На основі цих аналізів Asana генерує рекомендації щодо того, які дії слід виконувати першими, щоб команда ефективно досягала своїх цілей. Додатково Asana пропонує ряд інших корисних функцій, таких як діаграми Ганта, подання календаря, звіти та аналітична панель, які дозволяють менеджерам проєктів всебічно розуміти та відстежувати прогрес проєкту й ефективність команди. Діаграми Ганта в Asana дають змогу організовувати завдання вздовж часової осі та візуально відстежувати залежності й прогрес. Вбудований календар підтримує планування й відстеження дедлайнів для завдань і подій, що полегшує координацію термінів і віх.

Звіти Asana дають менеджерам проєктів змогу збирати та аналізувати структуровані дані про прогрес проєкту, зокрема час, який команда витрачає на окремі завдання, і дотримання встановлених дедлайнів. Аналітична панель Asana, крім того, дозволяє отримувати уявлення про кількість завдань, над якими наразі працюють команди, а також про кількість уже виконаних завдань. Ця інформація дає менеджерам проєктів змогу ефективно використовувати ресурси команди, коригувати пріоритети та розуміти рівень продуктивності команди, щоб ухвалювати обґрунтовані рішення щодо покращення її роботи.

## 3 СИСТЕМА УПРАВЛІННЯ ПРОЄКТУВАННЯМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Поняття моделі життєвого циклу розробки програмного забезпечення

Підхід має базуватися на застосуванні генеративних моделей на кожному етапі життєвого циклу розробки програмного забезпечення (SDLC – Software Development Life Cycle). Послідовна інтеграція генеративних моделей у всі фази дозволяє усунути суттєвий дефіцит багатьох із раніше розглянутих рішень у галузі управління проєктами. Існуючі рішення часто не враховують можливостей, які надають генеративні моделі для автоматизації процесів, покращення підстав для прийняття рішень та підвищення якості результатів у всіх фазах SDLC.

Розглянемо тепер, що саме розуміється під життєвим циклом розробки програмного забезпечення та які конкретні завдання виконує керівник проєкту в кожній з фаз (див. рис. 3.1).

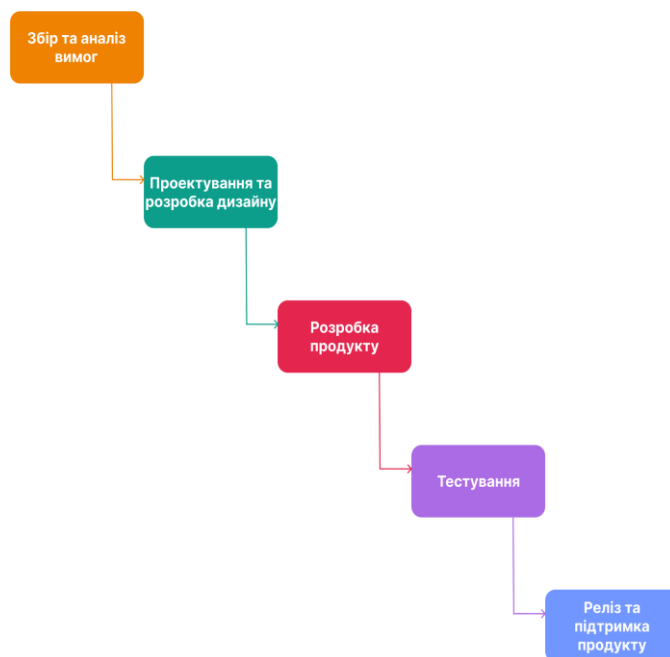


Рисунок 3.1 - Етапи життєвого циклу розробки ПЗ

Життєвий цикл розробки програмного забезпечення (SDLC) охоплює сукупність окремих робочих фаз, які виконуються у визначеній послідовності та протягом визначеного періоду. Цей період починається з рішення про розробку певного програмного продукту і завершується припиненням його використання. У

цьому рамках структуруються та керуються всі діяльності – від початкового аналізу потреб до остаточного виведення з експлуатації [12]. Керування розробкою програмного забезпечення часто є складним, оскільки вимоги можуть змінюватися в ході проєкту, технічні середовища зазнають оновлень, і необхідно координувати співпрацю між різними відділами. З цієї причини методологія життєвого циклу розробки програмного забезпечення надає структуру управління, яка для кожної фази визначає конкретні очікувані результати. Ця структура забезпечує узгодження цілей і вимог усіма зацікавленими сторонами на початковому етапі та наявність чіткого плану для досягнення цих цілей. Вона створює прозорість щодо відповідальностей, віх та критеріїв приймання і полегшує контроль та управління проєктом.

Розглянемо основні етапи SDLC, що наведено на рисунку 3.1.

1. **Збір та аналіз вимог** – На першому етапі менеджер проєкту тісно співпрацює з усіма релевантними зацікавленими сторонами, до яких належать замовники, кінцеві користувачі, бізнес-аналітики, предметні експерти та, за потреби, зовнішні партнери або постачальники. Метою цієї фази є систематичне виявлення потреб, очікувань та обмежень щодо планованого програмного продукту. Це включає активну комунікацію у формі бесід, структурованих інтерв'ю, воркшопів, спостережень та інших технік збору вимог, щоб отримати максимально повний і надійний набір вимог. Після збору слідує фаза аналізу: тут керівник проєкту разом із бізнес-аналітиками та технічними експертами перевіряє зібрані вимоги на зрозумілість, конкретність, повноту, а також на технічну й організаційну здійсненність. Менеджер проєкту ідентифікує та оцінює потенційні ризики й обмеження, досліджує залежності між вимогами та оцінює різні варіанти реалізації. Крім того, на нього покладається пріоритизація вимог, у ході якої він виділяє найважливіші функціональні вимоги та релевантні нефункціональні вимоги, такі як продуктивність, масштабованість, безпека, надійність і підтримуваність. На основі цієї пріоритизації готуються матеріали для прийняття рішень щодо подальших дій. Після завершення аналізу керівник проєкту активно

підтримує офіційну документацію виявлених вимог. Ця документація зазвичай містить детальні описи функціональностей, інтерфейсів, потоків даних, а також вимог до продуктивності, безпеки, відповідності нормативам та інших критеріїв якості. Менеджер проєкту забезпечує, щоб ця інформація була зафіксована у форматі, зрозумілому та доступному для всіх зацікавлених сторін, щоб вона слугувала обов'язковою основою для наступних фаз розробки.

**2. Проектування та створення дизайну** – На цьому етапі створюється дизайн і розробляється архітектура майбутнього застосунку. Керівник проєкту тісно співпрацює з розробниками, дизайнерами та іншими фахівцями з різних дисциплін, щоб детально вивчити вимоги і на цій основі визначити оптимальні рішення як з технічної, так і з дизайнерської точки зору. Надається значення тому, щоб були враховані всі релевантні аспекти, такі як зручність для користувача, технічна здійсненність та естетичні критерії. Керівник забезпечує ефективну комунікацію та тісну співпрацю всього проєктного колективу, координує узгоджувальні наради і стежить за тим, щоб інформація передавалася своєчасно та повністю. Він також допомагає визначити загальну архітектуру системи, включно зі структурою, окремими компонентами та їхніми взаємодіями, а також інтерфейсами між компонентами, і слідкує за тим, щоб ця архітектура була масштабованою, ефективною і добре структурованою. Це включає вибір відповідних технологій і патернів, визначення меж модулів, а також документування архітектурних рішень, щоб полегшити подальшу розробку та супровід. Керівник проєкту встановлює стандарти та процедури для контролю якості проєктування та дизайну, впроваджує підходи для перевірки якості дизайну й проводить регулярні рев'ю та аудити дизайну, щоб виявляти і усувати потенційні проблеми на ранніх стадіях. У цьому контексті він організовує сесії рев'ю, збирає відгуки зацікавлених сторін і забезпечує відстеження виявлених питань. Він ідентифікує потенційні ризики, пов'язані з проєктуванням і дизайном, аналізує їхні причини та наслідки, розробляє стратегії їхнього управління і готує конкретні плани щодо мінімізації та уникнення ризиків. Крім того, він забезпечує постійний зворотний зв'язок і тісну координацію з замовником щодо проєктного

дизайну, представляє замовникові результати проєктування та дизайну у відповідній формі і враховує його зауваження та бажані зміни, щоб гарантувати відповідність очікуванням замовника. При цьому він документує результати погоджень і забезпечує впровадження узгоджених коригувань у процесі дизайну.

**3. Розробка продукту** – На цьому етапі керівник проєкту організовує і керує роботою команди, що складається з розробників, програмістів та інших фахівців, чітко та прозоро розподіляє завдання, встановлює пріоритети та безперервно контролює виконання робіт. Він дбає про те, щоб ролі та відповідальності були чітко визначені, і підтримує команду у плануванні робочих пакетів та етапів. Він також керує необхідними ресурсами, такими як бюджет, матеріали, обладнання, інфраструктура та програмне забезпечення, забезпечує їх своєчасне придбання та ефективне використання протягом проєкту. До цього належить моніторинг використання ресурсів, коригування планів ресурсів за потреби, а також забезпечення наявності потрібних інструментів і ліцензій. Крім того, керівник проєкту встановлює чіткі стандарти якості та процедури контролю для створення високоякісного продукту, організовує та проводить тестування і оцінювання якості продукту, щоб переконатися, що функціональні та нефункціональні вимоги виконані. Він виявляє й керує змінами, що виникають під час розробки продукту, аналізує їхній вплив на графік, витрати та якість, оцінює їхню значимість для проєкту і приймає рішення щодо їх впровадження з метою мінімізації втрат часу та ресурсів. Цей процес змін документується і прозоро комунікується, щоб усі зацікавлені сторони були поінформовані. Також керівник проєкту сприяє ефективній комунікації всередині команди та із замовником, встановлює регулярні наради, механізми звітування та зворотного зв'язку, забезпечує збір і опрацювання відгуків, підтримує вирішення конфліктів і стимулює співпрацю всіх учасників проєкту для забезпечення безперебійного просування робіт з розробки.

**4. Етап тестування** – Ця фаза охоплює комплекс завдань, пов'язаних із тестуванням продукту, і є ключовою для забезпечення якості продукту. Керівник проєкту готує план тестування, визначає цілі та встановлює обсяг робіт,

розподіляє завдання між тестовими фахівцями і встановлює конкретні часові рамки для проведення тестів та оцінки їх результатів. Крім того, керівник забезпечує ефективне керування тестовими процесами відповідно до встановлених стандартів якості, контролює виконання планів тестування і координує необхідні заходи для усунення виявлених проблем. Він координує роботу тестової команди, надає необхідну підтримку та ресурси, забезпечує відповідні тестові середовища та тестові дані і сприяє взаємодії між командами розробки та тестування для швидкого вирішення проблем і постійного вдосконалення продукту. Крім того, керівник відповідає за забезпечення високої якості продукту після завершення тестів, організовуючи виправлення помилок, проведення повторних тестів і супровід покращення функціональності відповідно до вимог замовника та стандартів якості. Він документує результати тестів, відстежує виправлення дефектів і проводить фінальні перевірки, щоб переконатися, що виявлені проблеми задовільно усунено, перш ніж продукт буде передано або випущено.

**5. Реліз та підтримка продукту** – На останньому етапі керівник проєкту бере на себе низку ключових і різнопланових завдань, що є вирішальними для успішного завершення проєкту та для довгострокового супроводу продукту. До його обов'язків належить розробка і опрацювання детальної стратегії релізу. Це включає чітке визначення функцій, які увійдуть до випуску, пріоритезацію фіч та встановлення конкретних дат публікації. Він також планує окремі етапи та координаційні кроки, щоб забезпечити своєчасне проведення запланованих релізів і їх відповідність очікуваній якості.

Керівник проєкту координує роботу розробників, тестувальників та інших зацікавлених сторін і забезпечує ефективну взаємодію всіх учасників, щоб реліз пройшов безперебійно. У цьому контексті він керує процесом затвердження продукту, документує необхідні кроки та гарантує належну реалізацію й розгортання. Він стежить за дотриманням процесів, перевіряє критерії якості перед випуском і при потребі організовує управління залежностями між командами та зовнішніми підрядниками.

Після релізу керівник проєкту продовжує нести відповідальність за підтримку продукту. Він забезпечує структури підтримки, усуває виявлені проблеми та помилки і координує оперативні реакції на збої, аби зберегти задоволення користувачів. Додатково він піклується про належну підтримку користувачів, відповідає на запити або забезпечує наявність команд і ресурсів служби підтримки для допомоги користувачам у використанні продукту.

Важливою складовою його завдань є постійний моніторинг функціональності продукту та взаємодії з користувачами. Він систематично збирає зворотний зв'язок, аналізує реакції користувачів і документує пропозиції щодо покращення, що виникають з практичного використання. На основі цих спостережень він оцінює необхідні коригування і за потреби планує оперативні виправлення помилок або оптимізації.

Крім того, керівник проєкту розробляє стратегію подальшого розвитку продукту. Ця стратегія базується на аналізі потреб ринку, зібраного зворотного зв'язку від користувачів і відповідних технологічних трендів. З отриманих висновків він виводить напрями дій для розширення та покращення продукту і пріоритезує можливі подальші розробки. Завдяки цим заходам керівник проєкту не лише забезпечує успішну реалізацію продукту, а й гарантує його довгострокову підтримку й розвиток, що врешті-решт сприяє досягненню бізнес-цілей і стійкому задоволенню потреб користувачів.

### **3.2 Вибір моделі генеративного штучного інтелекту**

Враховуючи величезну кількість наявних рішень на основі GAI, кілька тисяч базуються на популярних моделях, наприклад GPT 3.5 чи GPT 4. Не менш важливим також є те, чи актуальні дані використовувались для навчання моделі, адже саме від цього залежить наскільки точними і релевантними будуть згенеровані відповіді. Ще одним досить вагомим аспектом є готовність платформи до використання середньостатистичним користувачем Інтернет – тобто людиною, яка не має навичок програмування і хоче швидко у форматі чату

чи іншого зручного інтерфейсу отримати відповіді на свої запитання чи бажаний контент.

Генеративні платформи штучного інтелекту – це сфера, яка швидко розвивається, і постійно з'являються нові платформи. Нижче наведено огляд деяких найпопулярніших генеративних платформ ШІ сьогодні, їх переваг та недоліків:

- **ChatGPT від OpenAI (GPT 3.5/GPT 4):** це велика мовна модель від OpenAI, яка здатна генерувати текст, перекладати текст на безліч мов, створювати різні типи контенту та інформативно відповідати на запитання. ChatGPT все ще знаходиться в стадії розробки, але його вже використовують для створення широкого спектру вражаючих програм, включаючи чат-боти, генератори коду та інструменти для копірайтингу.

#### ***Переваги:***

1) ChatGPT дуже добре розуміє природну (людську) мову, що робить його особливо ефективним для застосунків, які вимагають вільної форми комунікації, наприклад діалогових систем, інтелектуальних помічників або інтерактивних FAQ-рішень. Завдяки цьому розумінню можна ставити складні запити природною мовою і отримувати змістовні відповіді;

2) Платформа придатна для широкого спектра завдань, включно з відповідями на загальні запитання, створенням спеціалізованих документів, творчих текстів або генерацією фрагментів коду. Ця універсальність робить ChatGPT гнучким інструментом у різних сценаріях застосування.

3) Оскільки ChatGPT було натреновано на різноманітних і великих наборах даних, він має значні переваги в наданні відповідей у різних шаблонах, стилях і мовах. Це полегшує адаптацію вихідних текстів до конкретного запиту або до певних вимог щодо тональності й формату.

#### ***Недоліки:***

1) Через залежність від стандартизованих зразків у доступних тренувальних даних ChatGPT інколи може проявляти обмежену чутливість до специфічного контексту. Це може призводити до того, що згенеровані відповіді

іноді виглядають незв'язними або в певних ситуаціях не зовсім відповідають реальному контексту.

2) Як і багато інших великих мовних моделей, ChatGPT має ризик непередбаченої передачі упереджень або викривлень із тренувальних даних. Ці потенційні упередження можуть відобразитися в результуючих відповідях і спричинити односторонні або проблемні відповіді на певні запити.

- **Bard від Google (LaMDA)** Це велика мовна модель від Google AI, яка в багатьох аспектах схожа на ChatGPT. Bard, проте, спеціально розроблено так, щоб надавати особливо інформативні та вичерпні відповіді, тож видача часто містить докладні пояснення та фонову інформацію. Додатково модель навчена на величезній кількості прикладів коду, що дозволяє їй цілеспрямовано й вправно допомагати користувачам у розробці програмного забезпечення або надавати практичну підтримку при програмуванні. Завдяки цьому Bard підходить як для загальних розмовних завдань, так і для технічних питань у сфері розробки ПЗ [13].

### ***Переваги:***

- Bard – універсальний текстогенератор, який відмінно підходить для створення текстів у різних форматах і для різних цілей, наприклад для пояснювальних текстів, резюме чи творчих матеріалів;
- модель демонструє високий рівень креативності при створенні оповідань і художніх текстів, що робить її цінним інструментом для авторів, контент-творців і творчих проєктів;
- Bard відносно гнучкий у створенні різноманітних текстових відповідей у різних жанрах і стилях письма; крім того, модель у відповідях часто одразу надає посилання на первинне джерело і може доповнювати зміст відповідними зображеннями, що підвищує корисність і простежуваність відповідей;
- система часто пропонує кілька варіантів відповіді одночасно, тож користувачі можуть обирати між різними формулюваннями або підходами до вирішення.

### ***Недоліки:***

1) хоча Bard здатний створювати креативні тексти, при дуже довгих відповідях інколи може виникати складність у збереженні загальної когерентності та структурної послідовності на всю довжину тексту; це може проявлятися у частково фрагментованих або менш організованих фрагментах;

2) як і будь-яка система, що генерує текст, Bard час від часу може надавати результати з контекстуальними невідповідностями або небажаними змінами тону, що обмежує його надійність у чутливих контекстах.

- **Claude від Anthropic (Claude/Claude 2):** Це нова генеративна платформа III від Anthropic, яка навчається на великій кількості тексту та коду і наразі ще перебуває у стадії розробки. Незважаючи на цей етап розвитку, Claude уже продемонстрував здатність генерувати реалістичні та креативні результати, зокрема фрагменти коду, поетичні тексти, сценарії та музичні композиції. Практично Claude особливо підходить для завдань на кшталт розробки програмного забезпечення та творчого письма, оскільки модель у цих сферах може давати переконливі результати. Однак у деяких випадках Claude може виглядати менш креативним, ніж типові користувачі, тож створений контент іноді здається більш консервативним або стриманим [14].

### ***Переваги:***

- Claude генерує візуально описові та творчо насичені текстові результати, що вирізняються наочними формулюваннями та образними описами;
- модель може створювати текстові описи з чітко впізнаваними художніми стилями, тому вона добре підходить для творчих проєктів письма, стилізованих описів або художніх завдань

### ***Недоліки:***

1) на відміну від багатофункціональних загального призначення моделей, спеціалізація Claude на створенні художніх текстів може призводити до того, що він менш придатний для завдань поза сферою творчого письма; така фокусування звужує діапазон застосування у некреативних або дуже утилітарних завданнях

- **AI21 Studio від AI21 Labs (Jurassic-1 Jumbo):** Це потужна модель штучного інтелекту для генерації тексту, розроблена AI21 Labs. Jurassic-1 Jumbo

вважається найбільшою та найпотужнішою моделлю цієї серії для автоматичного створення текстів на сьогодні. Вона була навчена на дуже великому корпусі текстів і коду, і завдяки цьому здатна генерувати різні формати тексту. Можливі застосування включають генерацію програмного коду, написання віршів, створення сценаріїв та композицію текстів пісень. Завдяки всебічному навчанню на різноманітних джерелах даних модель може надавати гнучкі та форматонезалежні виходи [15].

### ***Переваги:***

- Модель від AI21 Labs вирізняється розширеними можливостями розуміння природної мови, завдяки чому вона може генерувати когерентні та контекстно відповідні тексти. Це покращене розуміння мови допомагає створювати тематично узгоджені і змістовно послідовні тексти.

- Jurassic-1 Jumbo також продемонструвала виражену креативність у виробництві різноманітного та якісного контенту. Ця творча сила робить модель особливо цінною для численних застосувань, зокрема для допомоги у письмі, генерації ідей або створенні різного роду інформаційних продуктів. Здатність надавати оригінальні та привабливі матеріали підвищує її корисність у креативних та редакторських контекстах.

- Як розвиток попередньої моделі Jurassic-1, Jurassic-1 Jumbo використовує масштабну архітектуру та додатково оптимізовані методи навчання. Це сприяє кращому розумінню та генерації контенту з широкого кола тем і тим самим підвищує її застосовність у різних доменах та сценаріях використання

### ***Недоліки:***

- 1) Великі мовні моделі, такі як ті, що розроблені AI21 Labs, часто дуже ресурсомісткі. Експлуатація та навчання таких моделей вимагають значних обчислювальних ресурсів, що може супроводжуватися високими витратами. Ці вимоги до ресурсів впливають як на розробку, так і на розгортання та поточне використання .

2. Як і для багатьох великих мовних моделей, для Jurassic-1 Jumbo існує занепокоєння щодо можливих упереджень у згенерованому контенті.

Забезпечення справедливості та зменшення або усунення системних викривлень залишається постійним викликом у розвитку та застосуванні таких моделей ШІ. Ідентифікація та корекція таких упереджень вимагає безперервних досліджень і заходів контролю якості

- **BLOOM від Hugging Face (BLOOM):** BLOOM – це ще одна потужна модель ШІ для генерації тексту, розроблена організацією Hugging Face. Вона була навчена на великих обсягах даних із залученням ресурсів у промислових масштабах, що надало їй широку базу знань та різноманітні генераційні можливості. BLOOM створено, щоб вирішувати широкий спектр мовних завдань, демонструючи високу гнучкість у стилі та формі.

#### ***Переваги:***

1. Модель навчена на величезному наборі даних, що дає їй широкий спектр знань і здатність генерувати тексти в різних стилях і реєстрах. Це всебічне навчання сприяє тому, що BLOOM може надавати контекстно відповідні та інформаційно насичені тексти.

2. BLOOM дуже універсальна і охоплює різні завдання. Серед них – створення текстів, переклад на 46 мов, підтримка в розробці програмного забезпечення на 13 мовах програмування, а також генерація контенту і відповідей на запитання в інформаційній формі. Така універсальність робить модель корисною для багатьох різних сфер застосування.

3. Hugging Face має активну спільноту, яка долучається до розвитку та покращення моделей. Залучення спільноти сприяє інноваціям, постійному вдосконаленню та адаптації моделей, що може підвищувати їхню якість і придатність.

4. Hugging Face також дотримується принципів відкритого коду і робить моделі та інструменти доступними для розробників і зацікавлених осіб. Така відкритість полегшує дослідження, перевірку та подальший розвиток моделей широкою громадськістю і підтримує прозорі процеси розробки.

#### ***Недоліки:***

1. Без детального знання внутрішньої роботи BLOOM важко цілеспрямовано виявляти та виправляти конкретні вразливості. Загальні проблеми, характерні для великих мовних моделей – наприклад, високі вимоги до ресурсів і можливі упередження – також потенційно релевантні для BLOOM і потребують уваги.

2. Оскільки це порівняно велика та потужна модель, її розгортання у середовищах з обмеженими ресурсами може бути ускладнене. Вимоги до обчислювальної потужності та пам'яті можуть ускладнювати використання в певних сценаріях і обмежувати застосування в менших або чутливих до витрат середовищах (таб. 3.1).

Таблиця 3.1 – Порівняльна оцінка ГШІ

| Модел<br>ь/<br>платф<br>орма | Готовність до<br>використання<br>(інтерфес<br>користувача)                                                                                                             | Відповідність до EU Artificial<br>Intelligence Act (AIA)* |             |       |            | Загальна<br>оцінка |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|-------------|-------|------------|--------------------|
|                              |                                                                                                                                                                        | Data                                                      | Comput<br>e | Model | Deployment |                    |
| ChatG<br>PT                  | Чатбот, інтуїтивно<br>зрозумілий<br>функціонал (3)                                                                                                                     | 3                                                         | 0           | 14    | 8          | <b>30</b>          |
| Bard                         | Чатбот, зрозумілий<br>функціонал, може<br>генерувати у певних<br>форматах, які<br>автоматично можна<br>перенести в інше<br>середовище (напр.<br>Таблиці<br>Google) (3) | 5                                                         | 0           | 11    | 11         | <b>33</b>          |

Продовження таблиці 3.1

|             |                                                                                                                |    |   |    |   |           |
|-------------|----------------------------------------------------------------------------------------------------------------|----|---|----|---|-----------|
| Claude      | Чатбот, інтуїтивно зрозумілий функціонал, проте наразі недоступний у ЄС (потрібне VPN-з'єднання) (2)           | 0  | 0 | 2  | 5 | <b>11</b> |
| AI21 Studio | API, вимагає вміння програмування та іншого продукту для інтеграції (0)                                        | 0  | 0 | 4  | 4 | <b>11</b> |
| BLOOM       | Доступний вихідний код, до якого можна вносити зміни для кастомізації моделі, вимагає вміння програмування (0) | 11 | 8 | 10 | 7 | <b>39</b> |

З результатів цього аналізу чітко випливає, що найпопулярніше не обов'язково є найкращим. Те, що модель особливо широко розповсюджена або популярна, ще не означає, що вона у всіх випадках застосування дає найкращі результати. Щоб отримати справді цінні та надійні результати згенерованого вмісту, необхідно переконатися, що використовувана модель дійсно враховує всі релевантні аспекти роботи з вихідними даними. До цього належить: щоб самі дані були актуальними та коректними, щоб вони були належним чином підготовлені, і щоб модель була здатна надавати контекстно-адекватну й точну інформацію. Усі ці фактори та ще низка інших мають бути перевірені, щоб вивід був не лише зовні правдоподібним, а й змістовно обґрунтованим.

Аналіз також показав, що не варто обмежуватися виключно готовими, популярними рішеннями. Натомість може бути корисно звертати увагу на менш поширені варіанти моделей і за потреби активно сприяти їхньому подальшому розвитку або адаптації. Внесок у існуючу, але менш популярну модельну

варіанту може бути виправданим, оскільки це дозволяє досягти оптимізацій, які зроблять модель значно придатнішою для конкретного застосування. Такі налаштування можуть призвести до того, що менш відоме рішення в певних контекстах дає кращі результати, ніж лідери ринку.

Для цілей цього дослідження і для спрощення проведення експерименту свідомо було обрано для аналізу вимог лише ті моделі, які не потребують додаткових налаштувань і одразу можуть бути використані у вигляді чат-бота. Конкретно йшлося про ChatGPT (GPT-4), Bard (LaMDA) і Claude (Claude). Цей вибір дозволив відносно нескладну оцінку, оскільки всі три моделі без додаткових модифікацій готові до застосування і таким чином забезпечують порівнянні початкові умови для дослідження.

### **3.3 Опис роботи з системою**

Візьмемо за основу запит від клієнтів на розробку інтернет-магазину з продажу товарів.

Розглянемо як генеративна модель може допомогти керівнику проекту на кожному етапі моделі життєвого циклу проекту:

#### **3.3.1 Етап збору та аналізу вимог**

На початку проекту відбуваються перші переговори з замовником. На цих ранніх зустрічах розробник або проектна команда збирають першу інформацію про вимоги до продукту, побажання клієнтів, бізнесові умови та технічні вимоги. Сьогодні розвиток технології «speech-to-text» значною мірою просунувся. Тому отримати протокол розмови, що відбулася в одній із поширених онлайн-конференційних програм, таких як Zoom або Google Meet, сьогодні доволі просто. Як тільки є письмовий запис розмови з клієнтом, генеративна модель може обробити ці дані у великому обсязі й з високою швидкістю. Модель ШІ здатна з довгих транскриптів створювати стислий резюме, відфільтровувати важливі висловлювання і таким чином надавати компактний огляд ключових пунктів розмови. Крім того, модель може згенерувати структурований список основних

вимог, скласти пріоритезовані списки функцій і виділити будь-які неясності або відкриті питання, які є актуальними для подальших переговорів. Таким чином модель виступає інструментом для швидкої обробки й категоризації сирих даних з розмов із клієнтом. На рисунках 3.2 і 3.3 наведено відповідні прикладні запити та їхні результати щодо вимог проєкту та подальшої структуривання вимог.

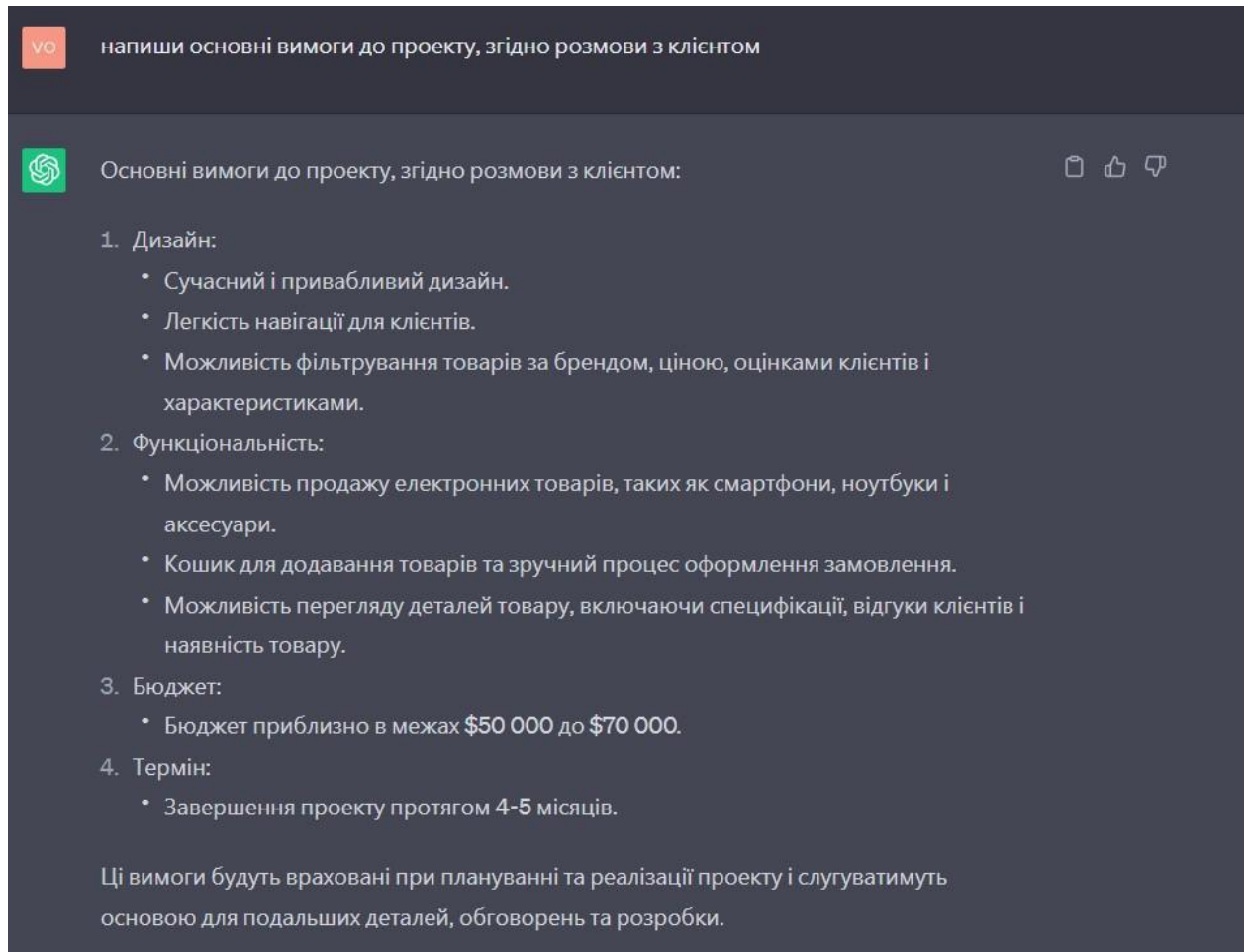


Рисунок 3.2 - Результати запиту до основних вимог проєкту

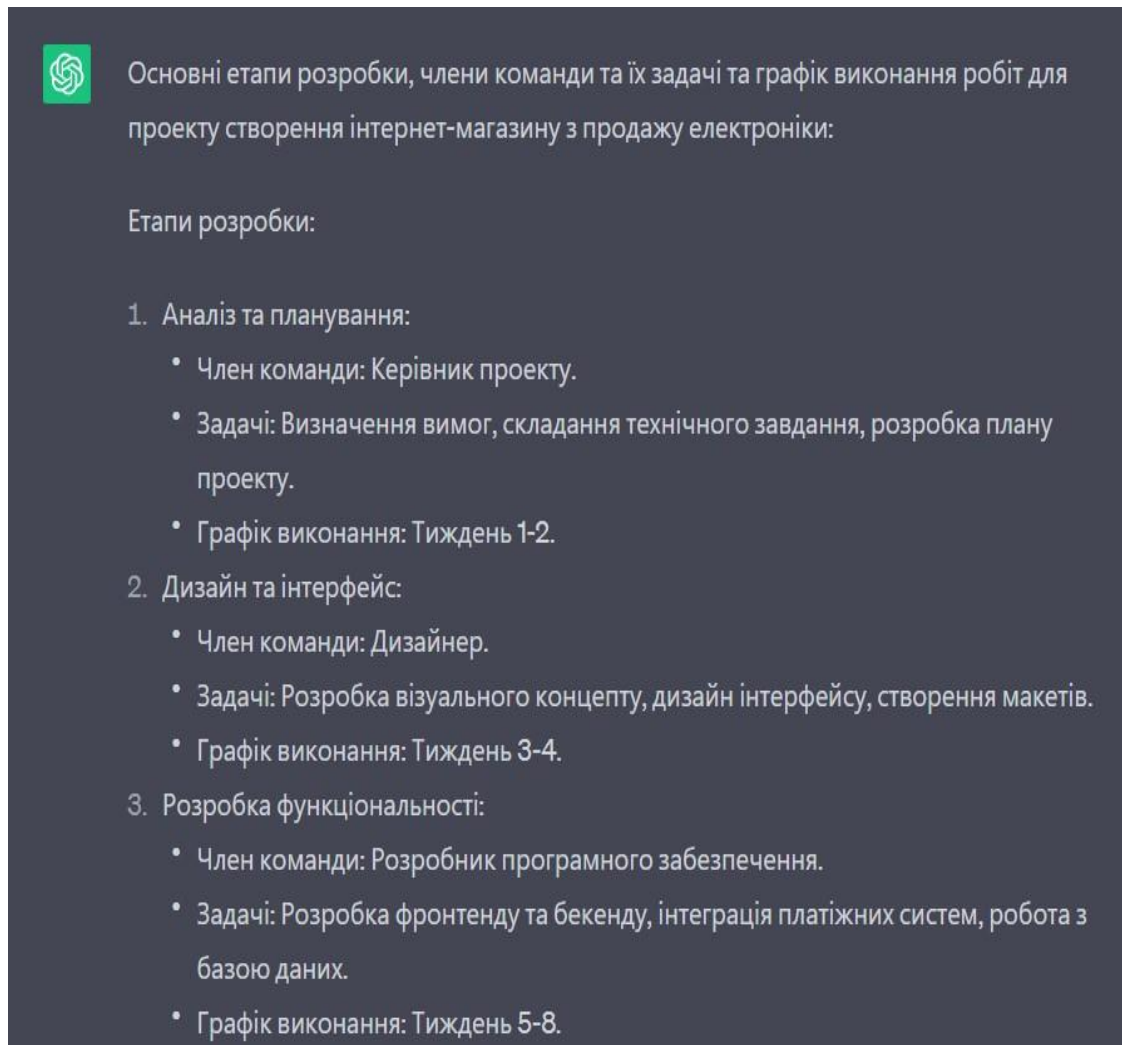


Рисунок 3.3 - Результати запиту до етапів розробки проекту

### 3.3.2 Етап проектування та створення дизайну

На другому етапі життєвого циклу проекту до керівника приєднуються додаткові члени команди – зазвичай розробники та дизайнери – щоб описати й спланувати майбутню архітектуру проекту, дизайн користувацького інтерфейсу, інформаційну архітектуру та інші необхідні завдання. Центральним завданням проектного менеджера є координація цих діяльностей і підтримка виконання окремих завдань. Тут генеративна модель може надати кілька конкретних допомог: вона може запропонувати варіанти архітектурних рішень, створювати шаблони для документації та допомагати у формулюванні завдань і специфікацій. Конкретним прикладом є створення різних UML-(Unified Modeling Language) діаграм, таких як діаграми прецедентів (Use Case Diagrams), послідовні діаграми (Sequence Diagrams), діаграми класів (Class Diagrams) та інших діаграмних

візуалізації структури системи й послідовностей взаємодії. Важливо зауважити, що ChatGPT як мовна модель самостійно не генерує зображення; проте він може створювати текстові описи та код діаграм, які зовнішні застосунки перетворюють на графічні представлення. Для роботи з діаграмами можна використовувати додаткові інструменти, що приймають текстовий код діаграми, у якому описані елементи, актори та залежності, і на основі цього генерують файл зображення. Прикладом такого інструмента є Mermaid.js – поширений інструмент для створення діаграм за допомогою текстової синтаксичної мови і JavaScript. ChatGPT може згенерувати відповідний код у синтаксисі цього інструмента, який потім вставляють у Mermaid.js для отримання фінальної діаграми. Повертаючись до прикладу інтернет-магазину з продажу електроніки: модель може спочатку надати код діаграми, який потім використовується в програмі для побудови готової UML-діаграми прецедентів. Приклад такої отриманої діаграми наведено на рис. 3.4.

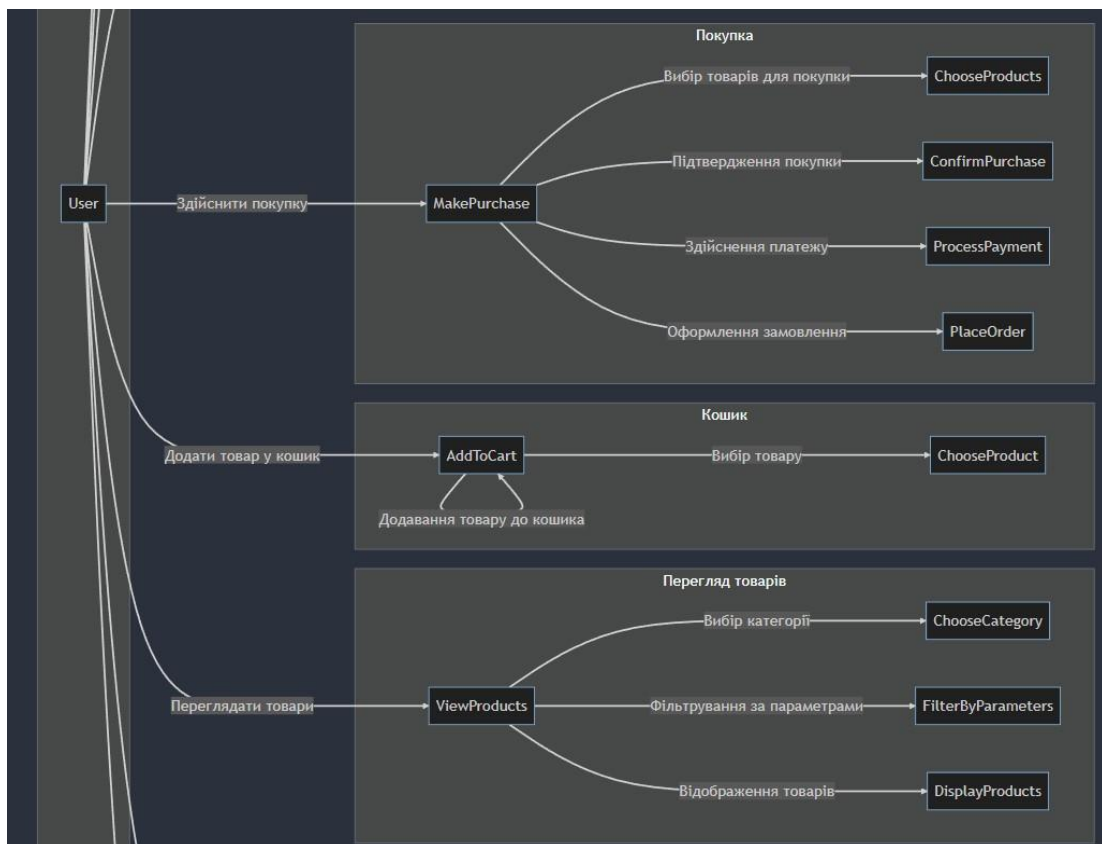
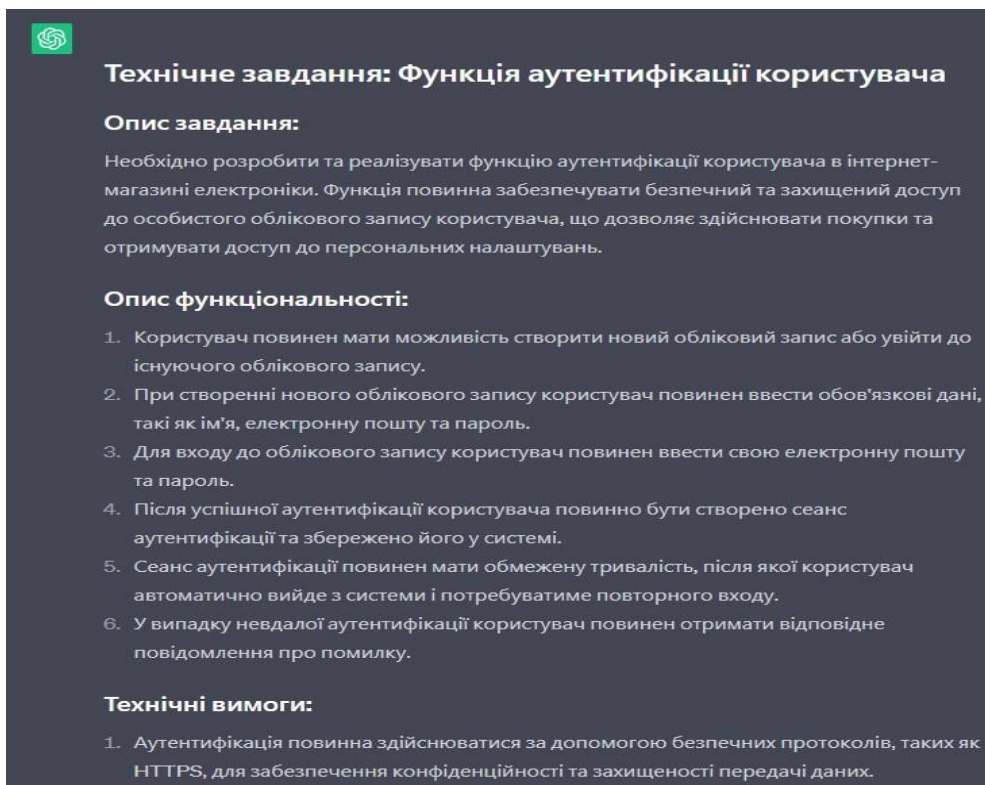


Рисунок 3.4 - Приклад отриманої Use Case Diagrams

### 3.3.3 Етап розробки продукту

На етапі розробки команда стоїть перед конкретним завданням створити працездатний продукт відповідно до документації, підготовленої на попередніх етапах. Розробники пишуть прикладний код, реалізують інтерфейси, втілюють дизайн-вимоги та інтегрують необхідні сервіси. Завдання керівника проєкту полягає в полегшенні роботи команди, контролі виконання та забезпеченні того, щоб завдання виконувалися згідно з затвердженим планом. У цьому контексті доцільно цілеспрямовано перевіряти й застосовувати можливості генеративної ШІ. Конкретні сфери застосування – автоматичне створення технічних завдань або деталізованих тікетів для розробників, формулювання чітких критеріїв приймання, а також підтримка у оцінці ризиків і заходів для запобігання проблемам у проєкті. Генеративна ШІ також може надавати стандартні фрагменти коду, шаблонні структури або пропозиції щодо підходів до реалізації, які розробники можуть використовувати як вихідну точку. На рис. 3.5 представлений приклад технічного завдання, згенерованого моделлю, яке може служити основою для тікета розробника.



**Технічне завдання: Функція аутентифікації користувача**

**Опис завдання:**

Необхідно розробити та реалізувати функцію аутентифікації користувача в інтернет-магазині електроніки. Функція повинна забезпечувати безпечний та захищений доступ до особистого облікового запису користувача, що дозволяє здійснювати покупки та отримувати доступ до персональних налаштувань.

**Опис функціональності:**

1. Користувач повинен мати можливість створити новий обліковий запис або увійти до існуючого облікового запису.
2. При створенні нового облікового запису користувач повинен ввести обов'язкові дані, такі як ім'я, електронну пошту та пароль.
3. Для входу до облікового запису користувач повинен ввести свою електронну пошту та пароль.
4. Після успішної аутентифікації користувача повинно бути створено сеанс аутентифікації та збережено його у системі.
5. Сеанс аутентифікації повинен мати обмежену тривалість, після якої користувач автоматично вийде з системи і потребуватиме повторного входу.
6. У випадку невдалої аутентифікації користувач повинен отримати відповідне повідомлення про помилку.

**Технічні вимоги:**

1. Аутентифікація повинна здійснюватися за допомогою безпечних протоколів, таких як HTTPS, для забезпечення конфіденційності та захищеності передачі даних.

Рисунок 3.5 - Приклад технічного завдання

### 3.3.4 Етап тестування

На етапі тестування розроблений продукт необхідно системно перевірити на помилки, невідповідності та відхилення від затверджених вимог. Оцінюють, чи реалізовані функції відповідають специфікаціям і чи взаємодія користувача працює так, як передбачено. Основну роботу на цьому етапі виконують тестувальники в команді, які виконують тест-кейси, документують помилки та повідомляють результати розробникам і керівництву проєкту. Роль проєктного менеджера полягає в підтримці тестувальників у їхній роботі, наприклад, шляхом надання стандартизованих тест-скриптів. Такі тест-скрипти описують послідовність взаємодій, які користувач має виконати з продуктом, і визначають очікувані результати. Завдання тестувальника – перевірити, чи всі дії, описані в скрипті, у розробленому продукті працюють коректно і дають очікувані результати. Модель може згенерувати кілька тест-сценаріїв і охопити різні варіанти тест-кейсів, наприклад звичайні сценарії використання, граничні випадки або варіанти з помилками. Вона створює структуровані тест-скрипти, які містять кроки, введення та очікувані результати, а також вказівки щодо пріоритетів і тестових даних. У поточному проєкті модель створила кілька сценаріїв; один із таких прикладів наведено на рис. 3.6. Завдяки цій автоматизованій підтримці підготовку тестування можна прискорити й краще забезпечити повноту тестування.

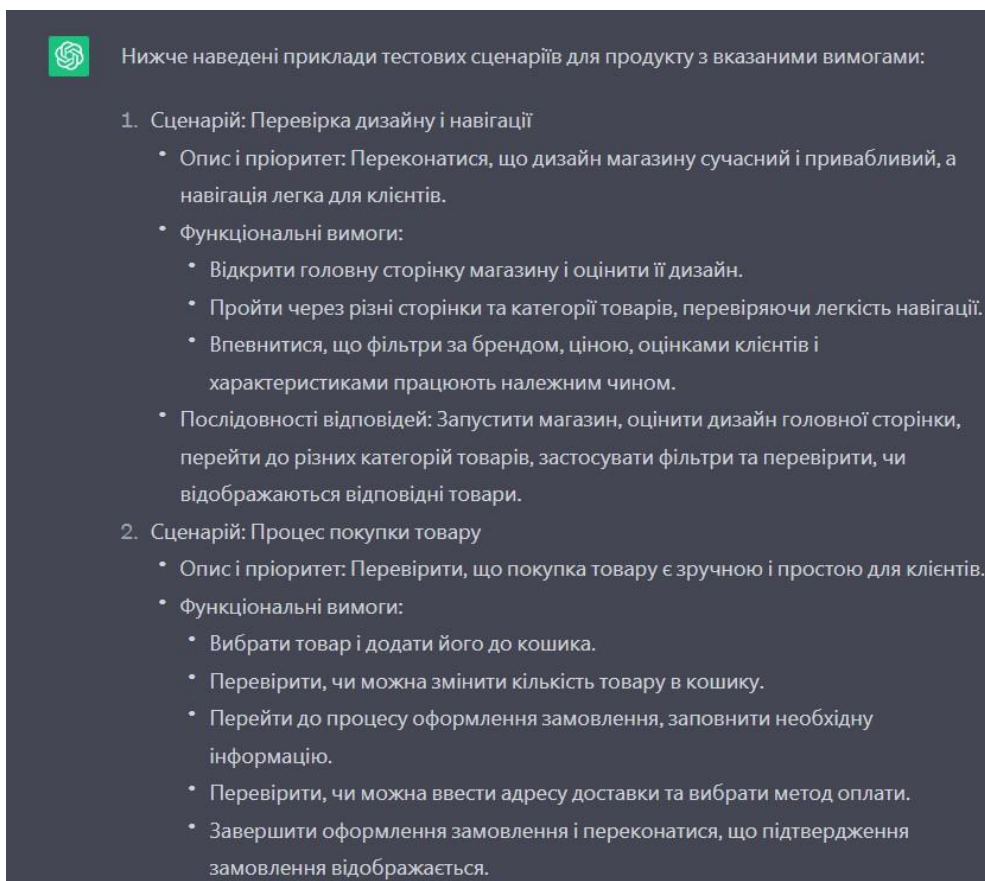


Рисунок 3.6 - Приклад сценарію

### 3.3.5. Етап релізу та підтримки продукту

Останній етап життєвого циклу продукту охоплює випуск розробленого продукту для використання кінцевими користувачами, а також постійну підтримку продукту шляхом оновлень, виправлення помилок і адаптацій, що базуються на отриманому зворотному зв'язку від користувачів. Після релізу центральними стають моніторинг, процеси підтримки й організація подальшого розвитку. Генеративні моделі можуть також підтримувати управління проектом на цьому етапі, автоматизуючи повторювані завдання та спрощуючи комунікацію з користувачами. Прикладом є створення готових відповідей на часто поставлені запитання (FAQ). Такі підготовлені відповіді розміщують на вебсайті або в центрі допомоги, щоб адресувати запити користувачів і зменшити навантаження на службу підтримки. FAQ, згенеровані моделлю, можуть бути сформульовані у стандартизованому вигляді, містити варіанти для різних цільових груп і за потреби оновлюватися при появі нових запитань. Результат відповідного запиту до моделі наведено на рис. 3.7. Загалом можна сказати, що генеративні моделі

протягом усього життєвого циклу – від збору вимог через планування, розробку й тестування до випуску та супроводу – пропонують різноманітні й конкретні можливості підтримки для прискорення процесів і покращення якості проєктної документації.

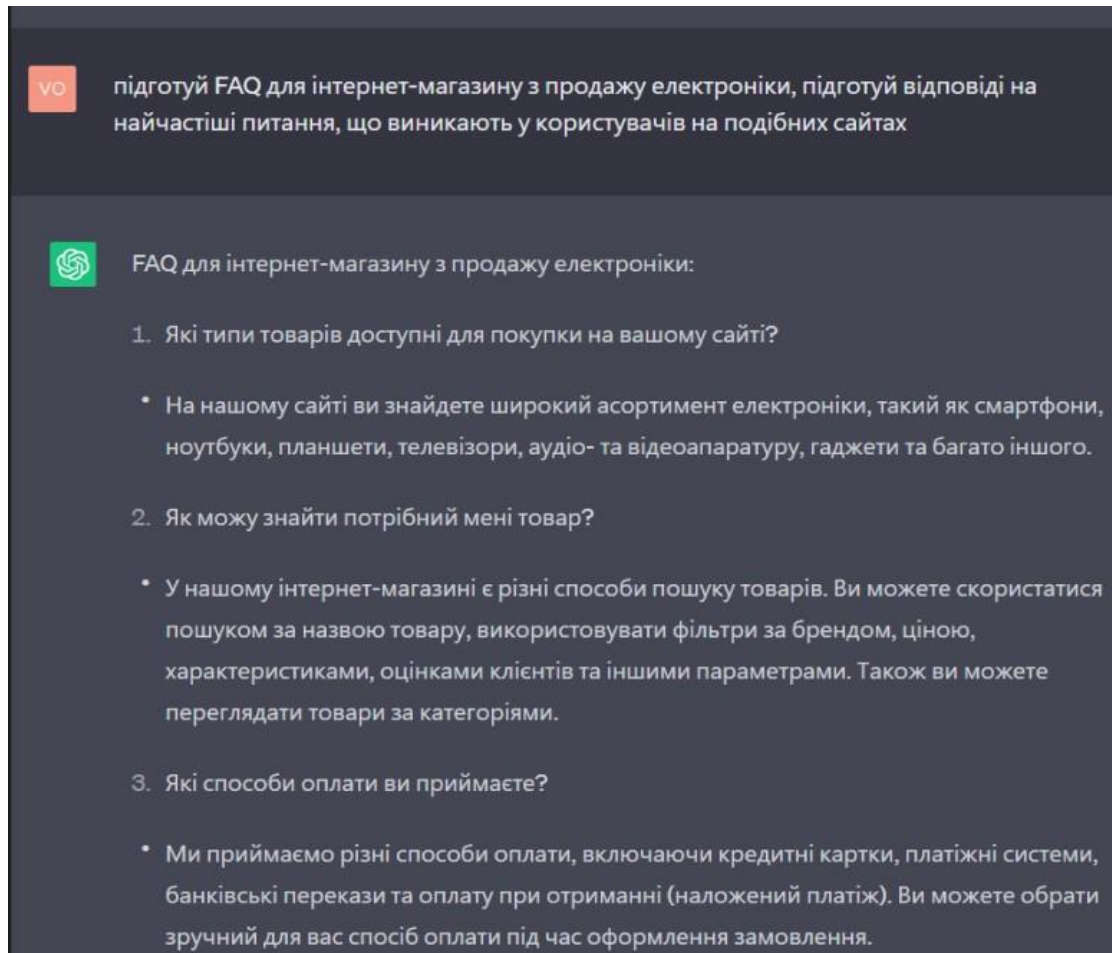


Рисунок 3.7 - Приклад FAQ

## ВИСНОВКИ

Проведений аналіз проблем в управлінні ІТ-проектами показав, що генеративний штучний інтелект є перспективним шляхом збільшення ефективності проектування. Актуальність проблеми пояснюється тим, що керування проектами відіграє вагомую частку в успішності виконання будь-якого проекту, тому впроваджуючи нові інноваційні рішення у цю сферу, можна досягнути покращення ефективності досягнення поставлених цілей.

У роботі розглянуто наявні проблеми у сфері управління проектами та недоліки існуючих рішень, що надають інструменти генеративного штучного інтелекту для керування ІТ-проектами. Враховувавши ці проблеми, було розроблено систему керування ІТ-проектами по створенню програмного продукту по моделі життєвого циклу з використанням генеративного ШІ.

Після практичного застосування та аналізу отриманих результатів було підтверджено думку, що ШІ є гарним інструментом, що позитивно впливає на результат роботи проектного менеджера, і в той же час, дозволяє людині працювати з творчими завданнями, що значно підвищує задоволення від робочого процесу.

## ПЕРЕЛІК ПОСИЛАНЬ

1. PMI | Project Management Institute [Електронний ресурс]: - Режим доступу до ресурсу: <https://www.pmi.org/about/learn-about-pmi/what-is-project-management>. Дата доступу: 25.04.2025 р.
2. Westland J. Top 10 Project Management Methodologies: An Overview [Електронний ресурс] / Jason Westland // ProjectManager. – Режим доступу: <https://www.projectmanager.com/blog/project-management-methodology> Дата доступу: 27.11.2025 р.
3. Generic AI: Artificial General Intelligence - Generic AI Home Page, [Електронний ресурс]: - Режим доступу: <http://www.generic-ai.co.uk/> , Дата доступу: 10.11.2025 р.
4. Making The Generic AI Trend Work For Your Business, Режим доступу: <https://www.forbes.com/sites/forbestechcouncil/2021/06/18/making-the-generic-ai-trend-work-for-your-business/?sh=e94230537682> , Дата доступу: 11.11.2025 р.
5. AI for Work and Project Management | Wrike, Режим доступу: <https://www.wrike.com/features/work-intelligence/> , Дата доступу: 13.10.2025 р.
6. Proggio Review - Project Management Tool Reviews, Pricing & Features, [Електронний ресурс] : - Режим доступу: <https://productivityland.com/proggio/> , Дата доступу: 14.05.2023 р.
7. Proggio - Project Portfolio Management Solution For Enterprises,[Електронний ресурс] : - Режим доступу: <https://www.proggio.com/how-it-works/> , Дата доступу: 14.05.2023 р.
8. Forecast nabs \$19M for its AI-based approach to project management and resource planning | TechCrunch,[Електронний ресурс] : -Режим доступу: <https://techcrunch.com/2021/05/19/forecast-nabs-19m-for-its-ai-based-approach-to-project-management-and-resource-planning/> , Дата доступу: 15.11.2025р.
9. How AI Will Transform Project Management,[Електронний ресурс] : Режим доступу: <https://hbr.org/2023/02/how-ai-will-transform-project-management> , Дата доступу: 15.11.2025 р.

10. Asana AI: Revolutionizing Project Management with Artificial Intelligence DotCom Magazine-Influencers And Entrepreneurs Making News,[Электроний ресурс] : -Режим доступу: <https://www.dotcommagazine.com/2023/04/asana-ai-revolutionizing-project-management-with-artificial-intelligence/> , Дата доступу:16.11.2025 р.

11. Project Management Institute (PMI),[Электроний ресурс] : - Режим доступу: <https://www.pmi.org/learning/library/project-management-problems-anticipating-conflict-5769> , Дата доступу: 20.11.2025.

12. Vartenie Aramali, Namho Chob, Falguni Pandec, M.K.S. Al-Mhdawie, Udechukwu Ojiakoh, Abroon Generative AI in project management: Impacts on corporate values, employee perceptions, and organizational practices Project Leadership and Society 6 (2025) 12p. Journal homepage: [www.sciencedirect.com/journal/project-leadership-and-society](http://www.sciencedirect.com/journal/project-leadership-and-society)

13. Robert Haase. Large Language Models: An Introduction [https://zenodo.org/files/LLMs\\_Intro\\_v22](https://zenodo.org/files/LLMs_Intro_v22)

14. Emmanuel Acquah Generative AI in project management: Leveraging AI power - Режим доступу: <https://www.dartai.com/blog/generative-ai-for-project-managers> Дата доступу: 20.11.2025.

## ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ  
ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

### КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система управління ІТ-проєктами із застосуванням  
генеративного штучного інтелекту»

Виконав: студент групи КНДМ-63

Ігор МОЛЯРЕНКО

Керівник: Юлія Березовська

2026

#### Мета та завдання

**Мета роботи** – Підвищення ефективності менеджерських рішень при управлінні ІТ-проєктами.

**Об'єкт дослідження** – Процес управління створенням ІТ-проєктів.

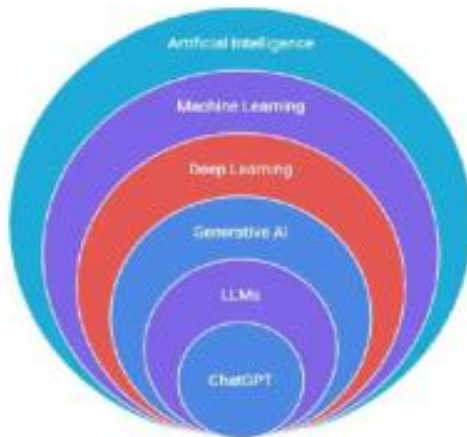
**Предмет дослідження** – життєвий цикл розробки програмного забезпечення

Завдання дослідження:

- ❖ Аналіз концепцій великих мовних моделей (LLMs).
- ❖ Аналіз напрямів застосування генеративного штучного інтелекту в управлінні ІТ-проєктами
- ❖ Аналіз і вибір засобів генеративного штучного інтелекту для системи.
- ❖ Розробка системи підтримки прийняття рішень при управлінні ІТ-проєктом

## Ключові компоненти генеративного штучного інтелекту

Hierarchy of AI Technologies



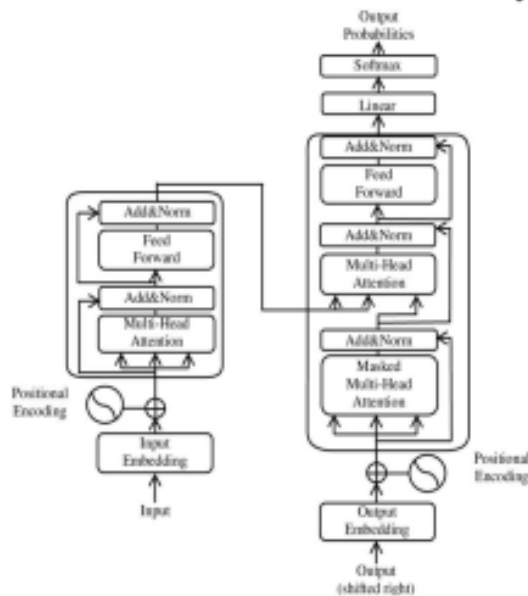
**1.Фундаментальні моделі:** Великі нейронні мережі, навчені на величезних обсягах даних, які можна адаптувати під різні завдання (наприклад, GPT-4, Claude 3.5, Gemini).

**2.Архітектура Transformer:** Основна структура більшості сучасних GAI, що використовує механізм **Self-Attention** для розуміння контексту та зв'язків між елементами в послідовності даних.

**3.Великі мовні моделі (LLM):** Компонент, що спеціалізується на розумінні та генерації тексту на основі ймовірного передбачення наступного слова.

3

## Ключові компоненти генеративного штучного інтелекту



### Ілюстрація архітектури трансформера

4

## Напрями застосування генеративного штучного інтелекту в управлінні проектами

### 10 Key AI Applications

How project managers leverage generative AI

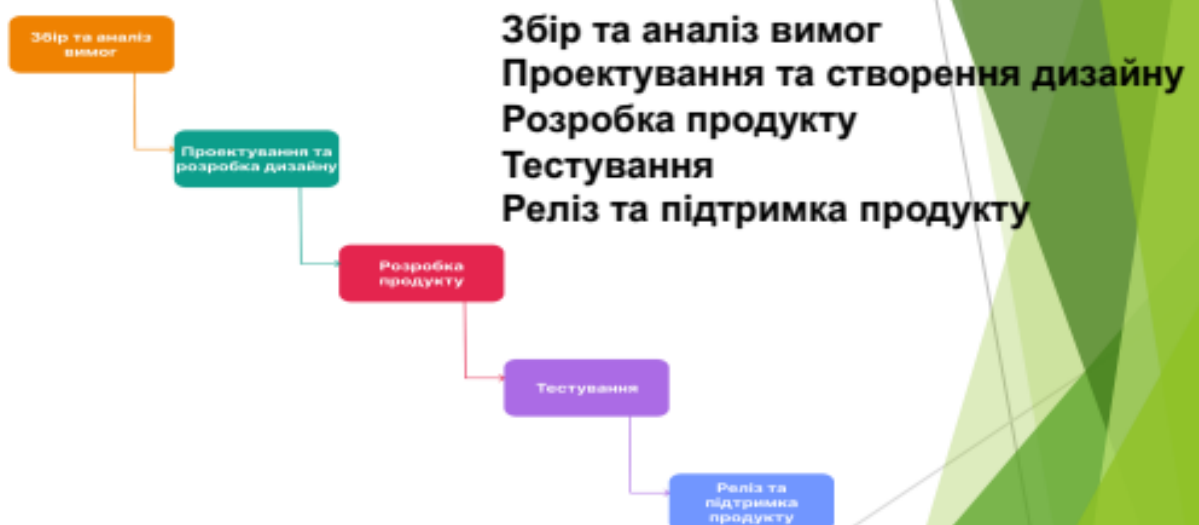


#### Допомога в управлінні змінами

1. Автоматизоване створення документації
2. Підсумки зустрічей та вилучення пунктів завдання
3. Визначення ризиків та планування їх пом'якшення
4. Оптимізація розподілу ресурсів
5. Прогнозування розкладу та прогнозування вузьких місць
6. Покращення комунікації із зацікавленими сторонами
7. Управління зустрічей та вилучення пунктів завдання
8. Системи підтримки рішень
9. Інтелектуальні панелі інструментів проекту
10. Допомога в управлінні змінами

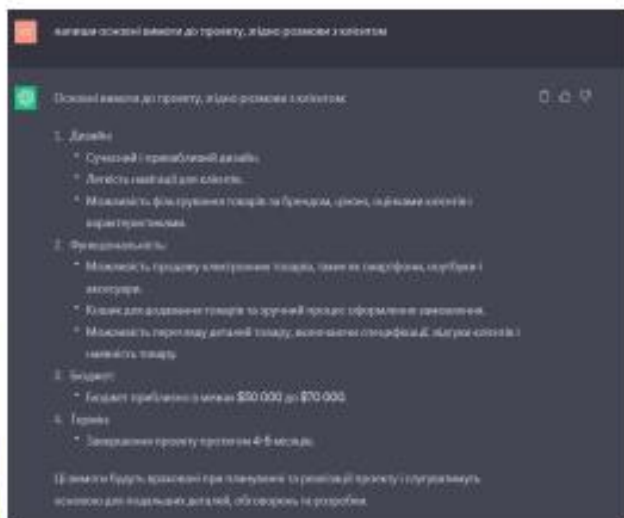
5

## Життєвий цикл розробки програмного забезпечення

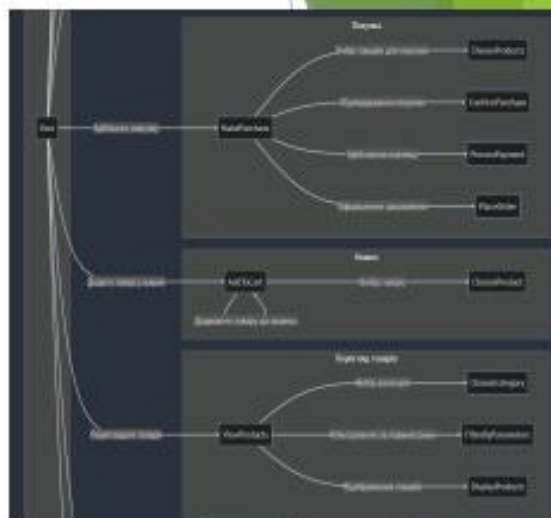


6

## Результати спілкування з ChatGPT



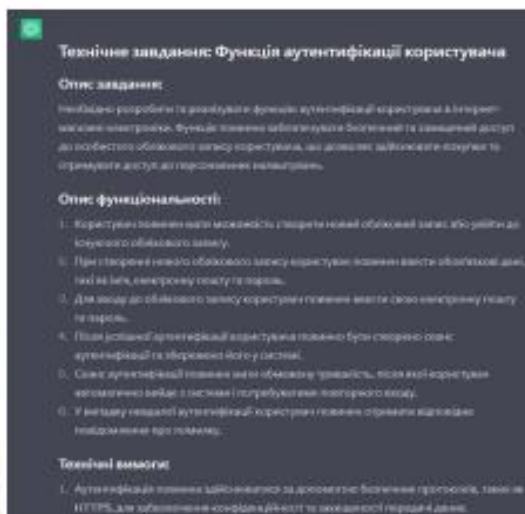
1 Етап. Генерація вимог до проекту



2 Етап. Генерація UML діаграм

7

## Результати спілкування з ChatGPT



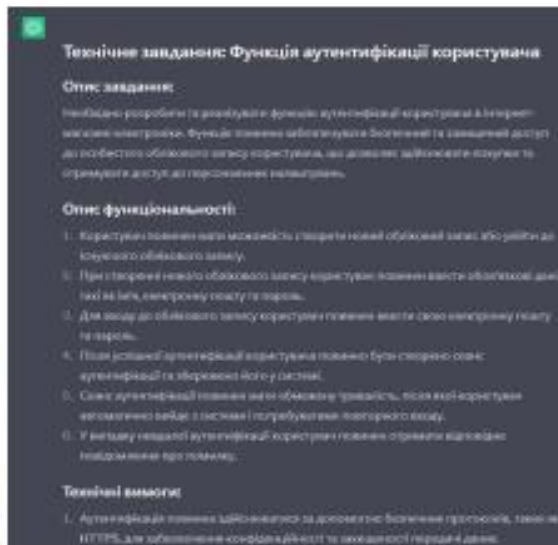
3 Етап. Розробка продукту (технічне завдання)



4 Етап. Тестування

8

## Результати спілкування з ChatGPT



5 Етап. Реліз та підтримка продукту

9

## ВИСНОВКИ

Проведений аналіз проблем в управлінні IT-проектами показав, що генеративний штучний інтелект є перспективним шляхом збільшення ефективності проектування.

Серед сучасних засобів генеративного штучного інтелекту ChatGPT має найкращі показники для задачі управління проектами.

Після практичного застосування та аналізу отриманих результатів було підтверджено, що генеративний штучний інтелект позитивно впливає на результат роботи проектного менеджера, і в той же час, дозволяє людині працювати з творчими завданнями, що значно підвищує задоволення від робочого процесу

10