

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Управління проєктами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження технологій управління проєктом створення веб-платформи для громадської організації з питань урбаністики»

Студента 2-го курсу групи УПз-21

Валерія НЕЧЕПУРЕНКО

(Ім'я, ПРИЗВИЩЕ)

(підпис студента)

Науковий керівник:

К.Т.Н.

(науковий ступінь, вчене звання)

Любов КУБЯВКА

(Ім'я, ПРИЗВИЩЕ)

(дата)

(підпис)

Попередній захист:

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач
кафедри технологій
управління

Віктор МОРОЗОВ

(підпис)

(Ім'я, ПРИЗВИЩЕ)

(дата)

Київ – 2025

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій**

Кафедра технологій управління

Освітній рівень Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Управління проєктами

ЗАТВЕРДЖУЮ

Завідувач кафедри
професор Віктор МОРОЗОВ

«29» вересня 2025 року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Валерія НЕЧЕПУРЕНКО

Група: УПз-21

1. Тема кваліфікаційної роботи

«Дослідження технологій управління проєктом створення веб-платформи для громадської організації з питань урбаністики»

Затверджена протоколом №15 від «16» 06 2025 року.

2. Строк подання студентом готової роботи – «04» 12 2025 р.

3. Цільова установка та вихідні дані до роботи: дослідження характеристики об'єкту управління планування виконання проєкту (календарне планування, зміст, бюджет, ресурси) та планування управління іншими галузями управління проєктами.

4. Зміст роботи: Аналіз ринку громадських організацій, паспорт проєкту, побудова концептуальної моделі платформи, опис життєвого циклу проєкту,

архітектура системи, WBS-модель, організаційна структура, фінансове планування, управління якістю та ризиками, висновки.

5. Перелік графічного матеріалу (слайдів): титульна сторінка, актуальність, мета, вимоги, дизайн-макет, планування, процес функціонування системи, ЖЦ проекту, логічна модель БД, WBS структура, висновки.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	Виконання роботи
1	Вивчення літературних джерел з предмету дослідження	22.09.25 - 23.09.25
2	Збір і вивчення матеріалів досліджуваного підприємства	23.09.25 - 26.09.25
3	Складання розгорнутого плану кваліфікаційної роботи	26.09.25 - 30.09.25
4	Ознайомлення наукового керівника з планом кваліфікаційної роботи	01.10.25
5	Підготовка розділу 1	02.10.25 - 10.10.25
6	Підготовка розділу 2	11.10.25-16.10.25
7	Підготовка розділу 3	17.10.25-06.11.25
8	Підготовка розділу 4	07.11.25-25.11.25
9	Оформлення кваліфікаційної роботи	28.11.25-07.12.25
10	Передача кваліфікаційної роботи науковому керівникові	08.12.25
11	Передача кваліфікаційної роботи рецензенту для рецензування	12.12.25
12	Захист кваліфікаційної роботи	22.12.25 - 23.12.25

Дата видачі завдання «__» _____ 2025 р.

Керівник роботи кандидат технічних наук, Любов КУБЯВКА
(посада, ім'я, прізвище)

(підпис)

Завдання прийняв до виконання студент групи УПз-21

Валерія НЕЧЕПУРЕНКО

(ім'я, прізвище)

(підпис)

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження технологій управління проектом створення веб-платформи для громадської організації з питань урбаністики»

Студент: Нечепуренко Валерія Віталіївна

Науковий керівник: Кубявка Любов Богданівна

Рік захисту: 2025

Тема даної кваліфікаційної роботи «Дослідження технологій управління проектом створення веб-платформи для громадської організації з питань урбаністики».

Метою кваліфікаційної роботи є трансформувати ГО на цифровий стандарт ефективного управління урбаністичними проектами шляхом розробки, впровадження та управління веб-платформою, здатною централізувати комунікацію, забезпечити прозорість та масштабувати соціальний вплив ініціативи.

Ціль проекту – створення комплексної веб-платформи для громадської організації «Місто Майбутнього» (Одеса), а також створенні прототипу цієї платформи для подолання цифрових бар'єрів, забезпечення прозорості та масштабування урбаністичних ініціатив організації.

Практичне значення: Результати роботи можуть бути використані місцевою владою, громадськими та культурними організаціями для впровадження ефективних моделей урбаністичного розвитку, підвищення громадської активності та забезпечення WCAG-доступності інформаційних ресурсів.

Кваліфікаційної робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, переліку використаних інформаційних джерел.

Перший розділ містить аналіз проблеми, PEST- та SWOT-аналіз для оцінки зовнішніх та внутрішніх впливів. Обґрунтовано вибір Agile/Scrum як доцільної методології. Сформовано Дерево цілей та проведено аналіз аналогічних проектів для визначення ключового функціоналу.

У другому розділі були зроблені: постановка задачі дослідження та формулювання технічного завдання; аналіз вибору методології управління проектом; огляд та вибір інструментарію управління проектом; визначення вимог до продукту ІТ проекту; розробка та обґрунтування архітектурних моделей інформаційної системи; використання методів ШІ та моделювання розроблених моделей.

Третій розділ включає в себе розробку плану проекту, створення ієрархічною структурою робіт (WBS) та календарним плануванням за допомогою діаграми Ганта. Також розглядаються процеси управління проектними ризиками, зокрема ідентифікація та оцінка ризиків проекту, а також розробка карти протиризикових заходів.

Четвертий розділ охоплює: складові веб-розробки; алгоритмічне та інтерфейсне забезпечення; аналіз та управління ризиками проекту; кошторис проекту.

Після виконання кваліфікаційної роботи було проаналізовано результати та сформовані висновки.

Ключові слова: управління проектами, ризиками, якістю; календарне планування; концептуальні моделі; громадська організація; веб-платформа.

ТАБЛИЦЯ СКОРОЧЕНЬ ТА ПОЯСНЕНЬ

Скорочення	Пояснення
БД	База даних
ШІ	Штучний інтелект
NLP	Natural Language Processing (Обробка Природної Мови)
ML	Machine Learning (Машинне Навчання)
MVP	(Minimum Viable Product) Мінімально Життєздатний Продукт
KPI	Key Performance Indicators (Критерії Успіху)
WCAG	Web Content Accessibility Guidelines (Рекомендації щодо Доступності Веб-вмісту)
ЖЦ	Життєвий цикл
WBS	Work Breakdown Structure (Структура Декомпозиції Робіт)
WIP	Work In Progress Limit (Обмеження Незавершеної Роботи)
OBS	Organizational Breakdown Structure (Організаційна структура роботи)

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ.....	10
1.1. Характеристика об'єкту дослідження та проблеми.....	10
1.2. Аналіз літературних та інформаційних джере.....	12
1.3. Огляд методів оцінки впливів оточення ІТ-проєктів.....	13
1.4. Аналіз аналогічних проєктів та веб-платформ в Україні.....	15
1.5. Комплексний аналіз середовища проєкту та прогностичне обґрунтування стратегії.....	16
1.6. Формування дерева цілей проєкту.....	21
РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	22
2.1. Постановка задачі дослідження та формулювання технічного завдання.....	22
2.2. Дослідження та аналіз вибору методології управління проєктом.....	23
2.3. Огляд та вибір інструментарію управління проєктом.....	36
2.4. Визначення вимог до продукту ІТ проєкту.....	38
2.5. Формування Use Case елементів до функціональних вимог.....	42
2.6. Розробка та обґрунтування архітектурних моделей інформаційної системи.....	50
2.7. Використання методів ШІ та моделювання розроблених моделей.....	59
РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ...	62
3.1. Життєвий цикл проєкту.....	62
3.2. WBS робіт та віхи проєкту.....	63
3.3. Організаційна структура та розподіл ролей.....	67
3.4. Оцінка трудомісткості та ресурсів.....	69
3.5. Діаграма Ганта проєкту.....	70
3.6. Trello-дошка проєкту.....	73

РОЗДІЛ 4. Аналіз результатів та успішності проєкту.....	78
4.1. Технічна архітектура та реалізація платформи.....	78
4.1.1. Модульна архітектура та технологічний стек.....	78
4.1.2. Алгоритмічне забезпечення.....	79
4.2. Управління ризиками та критерії успішності.....	81
4.3. Фінансове планування та кошторис проєкту.....	83
4.3.1. Кошторис проєкту.....	83
4.3.2. Моделі фінансування та забезпечення сталості.....	86
ВИСНОВКИ.....	89
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....	91
Додаток А.....	93
Додаток Б.....	94
Додаток В.....	97
Додаток Г.....	98

ВСТУП

У контексті стрімкої урбанізації та зростаючих вимог до якості міського середовища, всебічний розвиток міст, як-от Одеса, є критично важливим. Ініціатива «Місто Майбутнього» є багатогранним проєктом, який вирішує актуальні проблеми, що стосуються міського розвитку, культурного життя та активної участі громадян.

Ефективність діяльності такої соціально орієнтованої ініціативи, що функціонує як культурний хаб та рушій урбаністичних змін, має значний вплив на життя міста. Хоча ініціатива не є комерційним підприємством, розуміння її ринкової позиції у сфері культурних та урбаністичних проєктів, а також її конкурентних переваг (зокрема, активне залучення громадськості та налагодження партнерств), є ключовим для оцінки її впливу та визначення стратегій подальшого розвитку.

Дослідження цієї ініціативи є актуальним для ідентифікації найкращих практик у сфері:

- Міського розвитку та урбанізму: модернізація інфраструктури, створення громадських просторів, відновлення історичних будівель, підвищення мобільності та безпеки.
- Залучення громадян, а саме: волонтерство, просування ідей сталого розвитку та екології.
- Освітніх та просвітницьких проєктів: співпраця з владою та міжнародний обмін досвідом.
- Міжорганізаційних партнерств: співпраця з владою та міжнародний обмін досвідом.

Ця кваліфікаційна робота присвячена комплексному аналізу діяльності ініціативи, що дозволить визначити її ключову роль у трансформації міського середовища Одеси.

Метою роботи є дослідження та аналіз багатогранної діяльності ініціативи «Місто Майбутнього» в Одесі, її стратегічних напрямків та конкурентних

переваг у контексті соціального та урбаністичного розвитку міста, а також рекомендацій для максимізації її соціального впливу та ефективності.

Для досягнення поставленої мети необхідно виконати наступні задачі:

1. Провести аналіз основних напрямків діяльності ініціативи «Місто Майбутнього».
2. Визначити ринкову позицію ініціативи як соціально орієнтованого проєкту та культурного хабу, а також її конкурентні переваги, що базуються на співпраці з громадами та інституціями.
3. Здійснити оцінку впливу ініціативи на покращення міської інфраструктури, культурне середовище та активність громадського суспільства Одеси.
4. Проаналізувати механізми залучення громадськості та партнерів до проєктів ініціативи, орієнтованих на сталий розвиток.
5. Розробити стратегічні рекомендації щодо посилення співпраці з місцевими та міжнародними організаціями та оптимізації реалізації ініціатив у сфері урбанізму та соціального розвитку.

Об'єктом дослідження є процеси комплексного розвитку міста Одеси, що охоплюють урбаністичні зміни, культурні ініціативи, залучення громадян та освітні проєкти.

Предметом дослідження є діяльність ініціативи «Місто Майбутнього», її стратегічні напрямки, методи роботи, ринкова позиція та конкурентні переваги у сфері сприяння соціальному та урбаністичному розвитку міста.

Практичне значення отриманих результатів полягає у тому, що вони можуть бути використані місцевими органами влади, громадськими та культурними організаціями Одеси та інших міст для впровадження ефективних моделей урбаністичного розвитку, підвищення громадської активності та налагодження міжнародного обміну досвідом у сфері інфраструктури та соціального розвитку.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ.

1.1. Характеристика об'єкту дослідження та проблеми.

Об'єктом дослідження є ГО «Місто майбутнього», яка відіграє ключову роль у трансформації міського простору Одеси, реалізуючи інноваційні та соціально значущі проєкти, наведено в таблиці 1.1.

Таблиця 1.1

Ключові напрямки діяльності ГО «Місто Майбутнього»

Напрямок діяльності	Приклади проєктів (ключові)	Вплив на функціонал веб-платформи
Створення інклюзивного середовища	Проєкти безбар'єрності, співпраця з військовим шпиталем.	Високі вимоги до WCAG-доступності та функціоналу збору потреб.
Ревіталізація міського простору	«Європейська набережна», «Військовий узвіз».	Модуль візуалізації прогресу проєкту, інтерактивні карти, галереї.
Екологічні та освітні ініціативи	Сортування сміття, освітні заходи.	Календар подій, інтерактивна карта пунктів сортування, освітні матеріали.
Залучення ресурсів та партнерство	Збір коштів на підтримку ініціатив.	Модуль донатів (безпечний та прозорий), сторінки партнерів.

Метою створення веб-платформи є перетворення цієї організації на цифровий стандарт ефективного управління урбаністичними проєктами, здатний залучати ресурси та координувати діяльність на новому рівні. Реалізація цієї мети перетворює розробку платформи зі звичайної модернізації на інструмент стратегічного прориву. Впровадження цілісної цифрової екосистеми дозволить ГО «Місто майбутнього» функціонувати як прозорий та інтерактивний комунікаційний хаб, що є ключовою умовою для масштабування урбаністичних ініціатив та повноцінного втілення місії зі створення інклюзивного середовища.

Проблемна область проєкту полягає у недостатній цифровій інтеграції громадської організації «Місто Майбутнього», що створює бар'єри для ефективної комунікації та масштабування урбаністичних проєктів.

Поточна комунікація організації зводиться переважно до використання зовнішніх соціальних платформ, зокрема соціальної мережі Instagram. Така обмеженість комунікації не відповідає потребам сучасної громадської організації, що працює з комплексними соціальними та фінансовими завданнями, і призводить до таких критичних ситуацій:

1. Бар'єри для інклюзивності та структурованої інформації: Ускладнюється надання детальної, структурованої інформації про комплексні проєкти (як-от «Європейська набережна» чи ініціативи з інклюзивності). Більш того, комунікація лише через соціальні мережі суперечить цілям безбар'єрності, оскільки такі платформи не відповідають критичним стандартам веб-доступності WCAG 2.1. Це створює цифровий бар'єр для доступу до інформації та участі для бенефіціарів (наприклад, люди з інвалідністю), підриваючи основну місію ГО.
2. Непрозорість та фінансові ризики: Ускладнюється здійснення прозорого збору коштів та публікації фінансових звітів. Відсутність централізованого, верифікованого каналу для звітності підриває інституційну довіру донорів та партнерів, що є прямим ризиком для сталості фінансування.
3. Децентралізація даних та неефективна координація: Відсутня єдина платформа для цифровізації ключових процесів, що призводить до критичної фрагментації даних. Інформація про реєстрацію на події, збір пропозицій, трекінг волонтерських годин, а також самі дані про прогрес проєктів (інтерактивні карти ініціатив) розкидана між різними зовнішніми інструментами. Це унеможлиблює:
 - Централізовану координацію волонтерів, експертів та місцевих громад.

- Автоматизований аналіз фідбеку та потреб громади, що планується інтегрувати за допомогою ШІ.
- 4. Відсутність функціонального хабу. Організація не має єдиної платформи для цифровізації ключових процесів (реєстрація на події, збір пропозицій, трекінг волонтерських годин, інтерактивні карти ініціатив).
- 5. Проблематика сталості та довіри: Залежність від зовнішніх соціальних платформ (які можуть змінити алгоритми, правила або стати недоступними) не сприяє підвищенню інституційної довіри та сталості комунікаційного каналу. Це ставить під загрозу довгострокову життєздатність організації.

Створення веб-платформи є розв'язанням цієї проблеми, перетворюючи ГО на прозорий, інтерактивний та функціональний комунікаційний хаб, що є єдиним шляхом до масштабування урбаністичних ініціатив та реалізації місії ГО «Місто Майбутнього».

Таким чином, створення цілісної веб-платформи — це не просто модернізація, а стратегічна необхідність. Це рішення перетворить ГО «Місто майбутнього» на прозорий, інклюзивний, інтерактивний та функціональний комунікаційний хаб, що є єдиним шляхом до масштабування урбаністичних ініціатив та реалізації своєї місії щодо створення інклюзивного середовища.

1.2. Аналіз літературних та інформаційних джерел.

Аналіз джерел є фундаментальним для обґрунтування технології управління та технічних рішень. Буде проведено критичну оцінку таких груп джерел:

- Технології управління проектами: дослідження стандартів PMBOK (для розуміння життєвого циклу) та, особливо, Agile Manifesto і керівництв Scrum/Kanban з метою адаптації гнучких методологій до некомерційного IT-проєкту.

- Ризик-менеджмент у громадському секторі: аналіз робіт, що описують методи ідентифікації та мінімізації соціальних, фінансових та політичних ризиків при реалізації урбаністичних ІТ-проектів в Україні.
- Розробка веб-додатків та UX/UI: вивчення стандартів веб-доступності WCAG 2.1 (Web Content Accessibility Guidelines), що є критичним для інклюзивних проектів ГО. Аналіз архітектурних патернів (наприклад, мікросервісна архітектура) для забезпечення масштабованості та стійкості.
- Кейси аналогічних проектів: огляд успішних українських урбаністичних веб-платформ та ІТ-проектів у волонтерському секторі.
- Застосування ШІ в комунікаціях: дослідження можливостей Machine Learning та NLP для персоналізації контенту, аналізу фідбеку та автоматизації взаємодії з користувачами на веб-платформі.

1.3. Огляд методів оцінки впливів оточення ІТ-проектів.

Успішна реалізація ІТ-проекту, особливо у сфері громадської урбаністики, вимагає не лише технічної досконалості, але й адекватного управління зовнішніми та внутрішніми впливами. Традиційні інструменти стратегічного планування, такі як PEST-аналіз та SWOT-аналіз, надають базове розуміння факторів впливу (політичних, економічних, соціальних, технологічних, екологічних та правових). Вони дозволяють зафіксувати поточний стан середовища та визначити потенційні ризики.

Однак для проекту створення вебплатформи ГО «Місто Майбутнього», який є динамічним і критично залежним від громадської думки, волонтерського ресурсу та змін у законодавстві, статичного аналізу недостатньо. Необхідно впроваджувати прогностичні методи, які дозволяють моделювати майбутні сценарії розвитку проекту та адаптувати стратегію ще до виникнення критичних змін. Прогностичні методи поділяють на дві основні категорії: якісні

та кількісні. Їх вибір залежить від етапу життєвого циклу проєкту та доступності аналітичних даних.

Якісні методи, наведені в таблиці 1.2, ґрунтуються на інтуїції, досвіді, знаннях та судженнях фахівців у відповідній галузі. Вони є незамінними, коли відсутня достатня кількість історичних даних або коли досліджуються унікальні, нові чи швидкозмінні процеси.

Таблиця 1.2

Якісні методи

Метод	Суть застосування в ІТ-проєкті
Метод Дельфі	Анонімне опитування групи експертів (урбаністів, ІТ-фахівців, активістів) для досягнення консенсусу щодо майбутніх тенденцій використання веб-платформи чи пріоритетів громади.
Метод сценаріїв	Розробка кількох альтернативних майбутніх (оптимістичний, песимістичний, найбільш ймовірний) для проєкту, наприклад, щодо рівня залученості громадян або змін у законодавстві.
Мозковий штурм	Колективне генерування ідей щодо можливих проблем чи нових функцій платформи на найближчі 1-2 роки.

Кількісні методи, наведені в таблиці 1.3, використовують математичні моделі та статистичний аналіз історичних даних для виявлення закономірностей і екстраполяції їх у майбутнє.

Таблиця 1.3

Кількісні методи

Метод	Суть застосування в ІТ-проєкті
Екстраполяція трендів	Прогнозування зростання кількості користувачів, обсягу даних або частоти використання певних функцій платформи на основі попередніх місяців/років.
Регресійний аналіз	Встановлення зв'язку між різними змінними (наприклад, між рівнем фінансування проєкту та кількістю реалізованих функцій) для прогнозування результату при зміні вхідних даних.
Методи часових рядів	Прогнозування короткострокових коливань (наприклад, сезонного зниження активності користувачів) на основі аналізу даних у динаміці.

Завдяки застосуванню прогностичних методів з'являється можливість для:

- Прогнозування попиту.
- Прогнозування ризиків.

- Управління ресурсами.

Таким чином, прогностичні методи є інтелектуальним інструментарієм для зниження невизначеності та забезпечення стійкості ІТ-проєкту в динамічному соціальному середовищі.

Інтеграція штучного інтелекту для оцінки впливу: Застосування інструментів Обробки Природної Мови (NLP) дозволяє аналізувати великі обсяги неструктурованих даних із соціальних мереж та медіа, ідентифікуючи реальні потреби та настрої цільових аудиторій щодо урбаністичних ініціатив. Це дає змогу не просто оцінити оточення, а й прогнозувати його реакцію на нові функції платформи.

Визначення технології управління: Результати аналізу впливів оточення, необхідність швидкої реакції на фідбек від громади та змінні пріоритети фінансування, є вирішальними для вибору методології управління. У даному дослідженні буде обґрунтовано, що Agile-методології, зокрема, Scrum є найбільш доцільними. Вони забезпечують гнучкість, ітераційну розробку MVP (Мінімально Життєздатного Продукту) та постійну співпрацю зі стейкхолдерами, що є ключовим для громадських ІТ-проєктів.

Аналіз впливів із залученням ІІІ не тільки оцінює доцільність проєкту, але й визначає ефективну технологію управління ним [14].

1.4 Аналіз аналогічних проєктів та веб-платформ в Україні.

Аналіз українських проєктів, наведений в таблиці 1.4, підтверджує доцільність створення платформи та дозволяє ідентифікувати найкращі практики управління та функціоналу. На основі проведеного порівняння було виокремлено перелік пріоритетних сервісів, які стануть фундаментом майбутнього хабу. Це дозволяє команді сфокусуватися на розробці унікального інкрементального продукту, який не лише відповідає національним стандартам цифрової трансформації, а й пропонує інноваційні підходи до координації діяльності ГО «Місто Майбутнього».

Аналіз аналогічних проєктів та веб-платформ

Проект/Платформа	Фокус діяльності	Ключові уроки для ГО «Місто майбутнього»
«ІТ Толока»	Волонтерство ІТ-спеціалістів для НУО.	Необхідність ефективного Модуля Волонтерства для координації та трекінгу робочого часу ІТ-фахівців.
«Промприлад. Реновація» (Івано-Франківськ)	Ревіталізація промислової зони.	Важливість прозорої комунікації та візуалізації складних, довгострокових проєктів (аналог «Європейська набережна»).
«Громадський бюджет»	Залучення громадян до розподілу муніципальних коштів.	Необхідність функціоналу голосування та збору пропозицій від громади, що підвищує відчуття спільної власності на проєкти.
Українські платформи для залучення ресурсів «dobro.ua»	Збір коштів для благодійних ініціатив.	Інтеграція безпечних та прозорих фінансових модулів, звітність у реальному часі.
«Доступно.UA»/ «Безбар'єрність»	Моніторинг та покращення фізичної доступності міст.	Інклюзивність як Функціонал: Платформа «Міста майбутнього» (яка має напрямок інклюзивності) повинна не лише бути WCAG-доступною, але й мати функціонал, що дозволяє вносити дані або скарги щодо стану міського середовища.
Екологічна ініціатива «Let's Do It Ukraine»	Організація масових акцій (суботників).	Логістика Подій: Необхідність географічного функціоналу для відображення локацій проєктів ГО.

Існує чіткий тренд на цифровізацію громадського сектору. Проєкт «Місто майбутнього» може інтегрувати найкращі рішення з управління волонтерами, прозорості та інклюзивного дизайну, що додатково обґрунтовує життєздатність та актуальність даного дослідження.

1.5 Комплексний аналіз середовища проєкту та прогностичне обґрунтування стратегії.

PEST-аналіз, наведений в таблиці 1.5, допомагає системно оцінити зовнішні макроекономічні фактори, що можуть суттєво вплинути на життєвий цикл та успіх реалізації ІТ-проєкту. На відміну від SWOT-аналізу, який

фокусується на внутрішніх характеристиках, PEST-аналіз дозволяє вивчити середовище «далекого оточення», що формує довгострокові тренди та обмеження для веб-платформи.

Для проєкту ГО «Місто Майбутнього» критично важливо розуміти ці вектори, оскільки урбаністична платформа функціонує на стику технологій, соціальної активності та державного регулювання.

Таблиця 1.5

PEST-аналіз

Фактор	Приклад впливу на веб-платформу	Оцінка впливу
Політичний	Змінні пріоритети фінансування та ризики: Зміна місцевої влади або пріоритетів центрального уряду може вплинути на доступ до грантів, фінансування проєктів ревіталізації та готовність органів місцевого самоврядування до співпраці.	Середній
	Нормативно-правова база для ГО та волонтерства: Спрощення чи ускладнення процедур звітності ГО або регулювання використання АІ може вплинути на функціонал платформи.	Середній
Економічний	Залежність від фінансування: Спад економіки може знизити обсяги донатів від приватного сектору та зменшити доступність міжнародних грантів.	Високий
	Вартість ІТ-ресурсів: Коливання курсу валют впливає на вартість хостингу, ліцензій та оплату праці вузькоспеціалізованих ІТ-фахівців, що посилює ризики масштабованості.	Середній
Соціальний	Актуальність міських проблем: Високий попит на вирішення проблем інклюзивності та ревіталізації в Одесі створює сприятливе середовище для залучення користувачів.	Високий
	Зростаючий попит на прозорість: Загальний соціальний тренд на довіру до прозорих ГО стимулює успіх платформи як централізованого хабу.	Високий
	Цифрова грамотність та залучення: Конкуренція за увагу та рівень цифрової грамотності цільової аудиторії впливають на швидкість адаптації до нового хабу.	Середній
Технологічний	Доступність технологій: Зростаюча доступність інструментів ШІ та хмарних сервісів знижує початкові витрати на розробку.	Високий
	Розвиток стандартів доступності: Вимоги до WCAG 2.1 стають де-факто стандартом, що вимагає постійного оновлення технологій.	Середній
	Кібербезпека: Постійні загрози кібератак вимагають значних ресурсів для захисту даних користувачів та стабільності платформи.	Високий

Для комплексної оцінки внутрішнього потенціалу створення веб-платформи та системного аналізу впливу зовнішнього середовища було проведено SWOT-аналіз, наведений в таблиці 1.6. Цей інструмент дозволяє не лише систематизувати наявну інформацію, а й виявити приховані взаємозв'язки між ресурсами організації та викликами ринку.

Метою аналізу є обґрунтування життєздатності проєкту, ідентифікація критичних точок контролю та формування стратегічних напрямків, спрямованих на мінімізацію ризиків і максимізацію конкурентних переваг.

Таблиця 1.6

SWOT-аналіз проєкту

Категорія	Внутрішнє середовище (Контрольовані фактори)	Зовнішнє середовище (Неконтрольовані фактори)
Сильні сторони	S1. Чіткі напрямки діяльності: Створення інклюзивного середовища, ревіталізація міського простору, еко-ініціативи, що формують багатий функціонал платформи.	O1. Зростаючий попит на прозорість: Загальний тренд в Україні та світі на підвищення прозорості некомерційного сектору (фінансові звіти, прогрес проєктів).
	S2. Інноваційний підхід: Заплановане застосування штучного інтелекту (NLP) для аналізу громадської думки та швидкого реагування.	O2. Наявність експертної бази: Можливість залучення волонтерів та експертів через централізований хаб для урбаністичних ініціатив.
	S3. Обґрунтований вибір методології: Вибір Agile забезпечує гнучкість, швидку реакцію на фідбек від громади та ітераційну розробку MVP.	O3. Актуальність міських проблем: Необхідність вирішення проблем інклюзивності та ревіталізації простору в Одесі (як об'єкті дослідження).
	S4. Дотримання стандартів: Високі вимоги до WCAG 2.1 (веб-доступність) сприяють інклюзивності та підвищують якість продукту.	
Слабкі сторони	W1. Поточна комунікаційна обмеженість: Надмірна залежність від однієї соціальної мережі (Instagram), що не дозволяє надавати структуровану інформацію та здійснювати прозорі звіти.	T1. Залежність від фінансування: Як некомерційний проєкт, критично залежний від грантів, донатів та партнерства, що може бути нестабільним.
	W2. Відсутність функціонального хабу: Немає єдиної платформи для цифровізації ключових процесів (реєстрація, пропозиції, трекінг волонтерів).	T2. Соціальні та політичні ризики: Змінні пріоритети фінансування та ризики, пов'язані із громадською думкою та політичною нестабільністю в урбаністичному секторі (як згадано у п. 1.2).

Продовження Таблиці 1.6

Слабкі сторони	W3. Потреба в навчанні: Необхідність навчити команду ГО ефективно користуватися новими інструментами ШІ та Agile-процесами.	T3. Конкуренція за увагу: Конкуренція з іншими громадськими та комерційними платформами за увагу та залучення цільової аудиторії.
	W4. Ризики масштабованості: Вибір архітектурних рішень (наприклад, мікросервісна архітектура) потребує значної початкової інвестиції часу та ресурсів для забезпечення стійкості.	

Стратегічні висновки (TOWS-матриця):

1. S-O Стратегії (використання сильних сторін для реалізації можливостей): Використовувати S2 (ШІ-аналіз) для моніторингу O1 (попиту на прозорість), автоматично генеруючи звіти та прогнозуючи реакцію спільноти.
2. W-O Стратегії (мінімізація слабких сторін, використовуючи можливості): Усунути W1 (обмеженість комунікації) за допомогою веб-платформи, щоб відповідати O1 (вимогам прозорості) та централізовано залучати O2 (експертів).
3. S-T Стратегії (використання сильних сторін для уникнення загроз): Використовувати S3 (Agile) для швидкої реакції на T2 (змінні пріоритети фінансування) та коригування функціоналу MVP відповідно до потреб.
4. W-T Стратегії (мінімізація слабких сторін та уникнення загроз): Розв'язати W2 (відсутність хабу) шляхом створення централізованого модуля донатів та звітності для підвищення T1 (сталості фінансування) та T2 (інституційної довіри).

В таблиці 1.7. наведена прогнозована метрика кількості активних волонтерів/експертів, залучених через платформу.

Кількісний прогноз

Період	Активні волонтери/міс.	Прогноз (Зростання)	Обґрунтування прогнозу
Початковий етап	50 (через стару комунікацію)	→	
Рік 1, Квартал 1 (Запуск MVP)	120	+70	Реалізація функціонального хабу усуває комунікаційну обмеженість, залучаючи наявність експертної бази.
Рік 1, Квартал 4	300	+180	Впровадження III для аналізу думок та WCAG 2.1 збільшує довіру та приваблює ширшу аудиторію.
Рік 2, Квартал 4	650	+350	Екстраполяція: Додаткове залучення експертів через актуальність проблем та стабілізація процесів.

Висновки для проекту: необхідно закласти архітектурні рішення з розрахунком на зростання бази користувачів у 13 разів протягом перших двох років, що впливає на вибір хостингу, бази даних та необхідність навчання команди для управління таким потоком.

Якісний прогноз, наведений в таблиці 1.8, для оцінки впливу основних загроз та слабких сторін. Ключові фактори невизначеності:

1. Стабільність зовнішнього фінансування.
2. Рівень прийняття платформи місцевою владою та громадою.

Якісний прогноз

Сценарій	Характеристика	Вплив на проєкт	Стратегія реагування (S-T / W-T)
1. Оптимістичний (Стійкий розвиток)	Фінансування стабільне, місцева влада активно підтримує ініціативи.	Висока можливість масштабування. Можна інвестувати у складні архітектурні рішення та розширювати функціонал III.	S-O Стратегія: Інвестувати в розробку нових модулів; використовувати Agile для запуску додаткових ітерацій.
2. Базовий (Консервативний)	Фінансування нестабільне, але достатнє для MVP. Влада нейтральна.	Середня швидкість розвитку. Проєкт критично залежить від W-T стратегій.	W-T Стратегія: Максимально сфокусуватися на підвищенні сталості фінансування через централізований модуль донатів та чіткі звіти.

<p>3. Песимістичний (Кризовий)</p>	<p>Значне падіння грантового фінансування та зміна політичних пріоритетів. Конкуренти посилюють свою присутність.</p>	<p>Існування проєкту під загрозою. Потреба у швидкій мінімізації витрат.</p>	<p>S-T Стратегія: Використовувати Agile для швидкого скорочення обсягу робіт до MVP та сфокусуватися на функціоналі, який забезпечує тільки модулі донатів та звітності.</p>
---	---	--	---

1.6 Формування дерева цілей проєкту.

Побудова Дерева цілей є ключовим етапом декомпозиції стратегічного задуму проєкту. Цей інструмент чітко візуалізує, як досягнення конкретних операційних завдань (нижчий рівень) призводить до виконання стратегічних цілей (середній рівень), які, у свою чергу, реалізують єдину головну мету проєкту (Рис. 1.1).

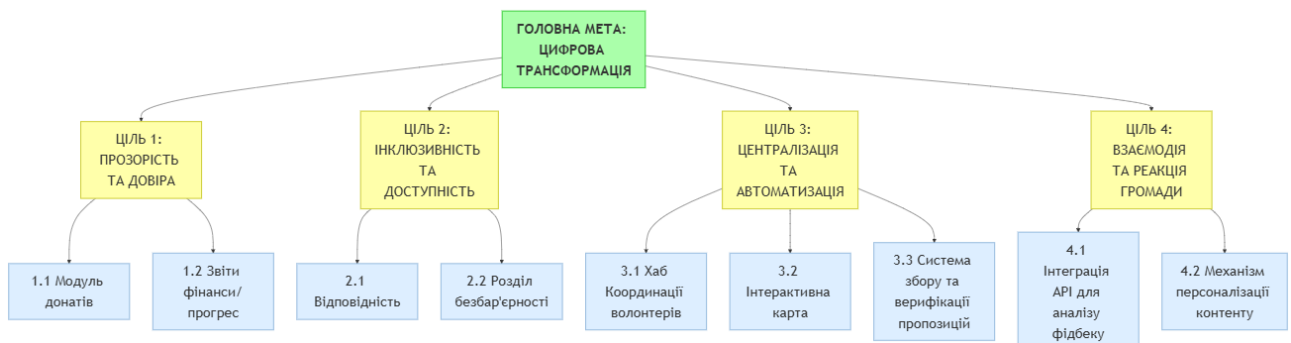


Рис. 1.1 – Дерево цілей проєкту

Сформульовані завдання прямо відповідають на проблемну область. Наприклад, завдання 3.1 та 3.2 безпосередньо усувають W2 (Відсутність функціонального хабу), а завдання 1.1 та 1.2 мінімізують T1 (Залежність від фінансування). Дане дерево цілей слугуватиме основою для формування функціональних та нефункціональних вимог до веб-платформи. Такий підхід забезпечує цілісність проєктування: від ідентифікації критичних розривів (слабких сторін та загроз) до конкретних технічних рішень.

РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1 Постановка задачі дослідження та формулювання технічного завдання.

Задача дослідження полягає у теоретичному обґрунтуванні та практичній розробці концептуальної моделі веб-платформи, яка забезпечить ефективну комунікацію та досягнення цілей ГО «Місто Майбутнього». Ключовий аспект – дослідження та обґрунтування технології управління цим проєктом, наведено в таблиці 2.1.

Таблиця 2.1

Паспорт проєкту

Назва проєкту	Розробка веб-платформи для громадської організації з питань урбаністики.
Мета проєкту	Створення ефективної веб-платформи організації «Місто майбутнього» для покращення комунікації та залучення цільових аудиторій.
Бізнес-обґрунтування	Веб-платформа є критично важливим інструментом для підвищення прозорості, цифровізації комунікацій та централізованого збору даних від громадян. Успішна реалізація дозволить значно розширити вплив ГО, збільшити волонтерську базу та забезпечити кращу якість урбаністичних проєктів.
Високорівневі вимоги	<ol style="list-style-type: none">1. Публічний функціонал: Відображення новин, аналітики, інтерактивних карт (GIS-інтеграція).2. Функціонал залучення: Реєстрація волонтерів, форми для подання пропозицій/скарг.3. Технічні: Адаптивний дизайн (Mobile-first), висока швидкість завантаження, надійний захист даних.
Обсяг	Повний цикл розробки: від детального планування та дизайну до розгортання та передачі платформи в експлуатацію. Включає: аналіз потреб, UX/UI дизайн, front-end та back-end розробку, тестування (альфа/бета), розгортання на продакшн-сервері.
Що не входить	<ol style="list-style-type: none">1. Наповнення контентом (написання всіх статей, завантаження всіх звітів) – цим займається контент-команда ГО.2. Технічне обслуговування після 12-го місяця.3. Розробка мобільного додатку (обмежується адаптивною веб-версією).4. Навчання команди розробників обраній методології.

Продовження Таблиці 2.1

Орієнтовні строки	Фаза 1: Планування та дизайн (3 місяці): Аналіз потреб, ТЗ, Архітектура, UX/UI Дизайн. Фаза 2: Розробка (6 місяців): Front-end та Back-end кодування, інтеграції. Фаза 3: Тестування та впровадження (3 місяці): Бета-тестування, виправлення помилок, розгортання, передача знань. Загальна тривалість – 12 місяців.
Бюджет (оцінка)	20 280 USD. Ключові статті витрат: фонд оплати праці.
Менеджер проєкту	Нечепуренко Валерія
Основні стейкхолдери	Керівник ГО, менеджер проєкту, команда розробки, команда ГО, експерти з урбаністики, волонтери, цільова аудиторія.
Критерії успіху	<ol style="list-style-type: none"> 2. Підвищення рівня поінформованості про діяльність організації (KPI: +30% трафіку на сайт за 3 місяці після запуску). 3. Збільшення кількості залучених волонтерів та учасників (KPI: 50+ нових реєстрацій волонтерів через платформу за півроку). 4. Технічний успіх: Безперебійна робота платформи протягом перших 6 місяців (uptime > 99%).
Критерії прийняття результату	<ol style="list-style-type: none"> 1. Виконано всі функціональні вимоги (згідно з фінальним ТЗ). 2. Успішно пройдено приймальне тестування (UAT) кінцевими користувачами. 3. Надано повну технічну документацію та інструкції з експлуатації.
Основні ризики	<ol style="list-style-type: none"> 4. Ризик зміни пріоритетів ГО: Уповільнення або призупинення проєкту. 5. Технологічний ризик: Складнощі з інтеграцією ГІС-систем. 6. Ресурсний ризик: Відтік волонтерів або ключових розробників.
Примітки	<ol style="list-style-type: none"> 1. Даний документ є основою для подальшого детального планування проєкту створення веб-платформи. 2. Успіх проєкту залежить від активної участі Керівника ГО та надання всієї необхідної вхідної інформації щодо майбутньої команди розробки.

2.2. Дослідження та аналіз вибору методології управління проєктом.

Методологія управління проєктом — це цілісна система, яка охоплює принципи, методи та правила роботи над проєктом. Її впровадження дає змогу:

- Зменшити ризики.
- Оптимізувати зусилля працівників.
- Уникнути дублювання функцій.
- Вивести на новий рівень взаєморозуміння в команді.

Методологія суттєво впливає на ефективність бізнесу та якість виконання робіт:

1. Створення культури успішного управління: Методологія стає основою, де чітко прописані кращі практики, що дають результат, і дії, яких слід уникати.
2. Чітке визначення ролей: Робить прозорими обов'язки та зони відповідальності для всіх залучених осіб, включно з топ-менеджерами, усуваючи плутанину.
3. Перехід до демократичних моделей управління: Чітко прописані ролі знижують рівень невизначеності, зменшують кількість конфліктів і сприяють внутрішній відповідальності співробітників.
4. База для врахування досвіду колег: Дозволяє систематизувати та ефективно обмінюватися набутим досвідом для використання в майбутніх проєктах.
5. Відмова від регулярного «винайдення колеса»: У методології фіксуються готові рішення та головні "лайфхаки", заощаджуючи час команд на пошук відповідей.
6. Зростання загального рівня якості роботи компанії: Забезпечує передбачуваність результатів, зменшуючи залежність від випадкового збігу обставин чи ентузіазму окремих співробітників.
7. Прогнозованість результатів: Наявність стандарту роботи дозволяє чітко формулювати терміни готовності проєкту та навіть завершувати роботу раніше запланованого.
8. Зрозумілі та контрольовані витрати ресурсів: З самого початку визначає обсяги робіт та необхідні ресурси, дозволяючи швидко виявити відхилення від плану.
9. Адекватна робота з ризиками: Містить алгоритми прогнозування ризиків та необхідних дій у разі їх настання, а також чіткі вимоги до комунікацій.
10. Прийняття ефективних рішень: Однозначне розуміння ролей, цілей та шляхів дозволяє приймати більш виважені та правильні рішення без дефіциту інформації.

11. Підвищення мотивації команди та продуктивності: Мінімум неприємних несподіванок та ясність цілей підвищують задоволення від роботи, що веде до зростання продуктивності.
12. Швидке зростання кваліфікації project-менеджерів: Менеджерам простіше взаємодіяти з командами та професійно розвиватися, отримуючи цілісне розуміння проєкту.
13. Замовник розуміє, що отримає в результаті: Методологія забезпечує коректну фіксацію вимог та погодження спільного розуміння задачі між виконавцем і клієнтом.
14. Проблеми у проєкті спливають раніше: Встановлені метрики контролю дозволяють помічати проєкти з дефектами практично одразу, даючи час на виправлення.
15. Стає зрозуміло, у що краще інвестувати: На основі даних методології можна виявити потреби (наприклад, інструменти для віддаленої роботи) та здійснити цільові інвестиції.

Методології управління проєктами поділяються на три основні категорії:

- Традиційний підхід, часто званий Waterfall (Водоспад), базується на лінійній та послідовній структурі. Головна ідея полягає в тому, що кожна наступна фаза проєкту починається лише після повного завершення попередньої. Основні характеристики:
 1. Жорстке планування: Весь обсяг робіт, бюджет та графік визначаються на самому початку (фаза планування).
 2. Документованість: Кожен етап супроводжується детальними специфікаціями та звітами.
 3. Низька гнучкість: Будь-які зміни в вимогах на пізніх етапах зазвичай коштують дорого і вимагають повернення до початкових фаз.
 4. Чіткі ролі: Учасники команди мають чітко визначені обов'язки, що мінімізує плутанину в управлінні.



Рис.2.1 – Візуальне представлення традиційної методології «Водоспадної» моделі

- Гнучкі методології є ідеальним вибором для проєктів, які характеризуються високою невизначеністю або вимогами, що постійно змінюються (Рис.2.2).



Рис.2.2 – Візуальне представлення гнучкої методології (Agile)

- Гібридні методології — це поєднання елементів традиційних та гнучких підходів. Вони ідеально підходять для великих і складних проєктів, де

певні частини мають жорстко визначені зовнішні вимоги (наприклад, регуляторні або законодавчі), а інші частини, як-от внутрішня розробка програмного забезпечення, потребують гнучкості та швидкої адаптації (Рис.2.3).

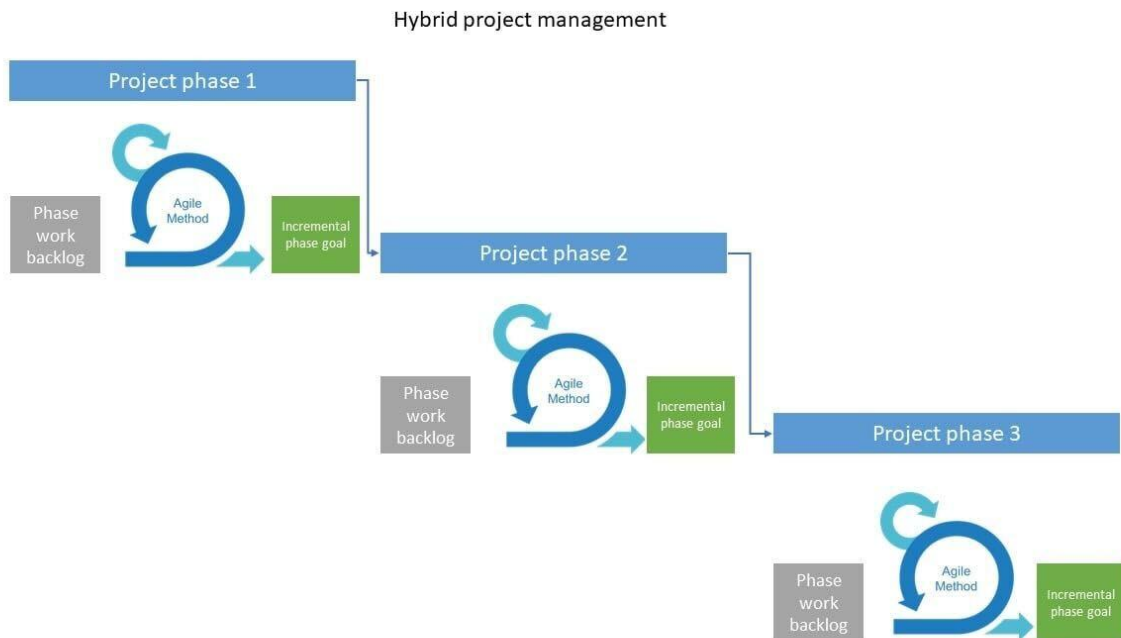


Рис.2.3 – Візуальне представлення гібридної методології

В сфері ІТ ефективне управління проєктами вимагає від менеджерів не слідувати сліпо одній методології, а набувати досвіду роботи з багатьма фреймворками та вміти поєднувати різні практики. Головна мета — максимально задовольнити потреби ІТ проєкту.

Керівники проєктів повинні обирати методи, інструменти та прийоми, які найкраще підходять для успішного завершення роботи. Важливо, що методології можуть змінюватися від одного проєкту до іншого, залежно від його розміру, результатів, складності та цілей.

За визначенням Інституту Управління Проєктами (PMI), методологія — це:

"Система практик, прийомів, процедур та правил, що використовуються тими, хто працює в тій чи іншій сфері." [4]

Згідно з опитуванням Pulse of the Profession за 2021 рік, 89% респондентів заявили, що їхні організації впровадили гнучкі методології (Agile) для управління проектами. Ця тенденція підкреслює важливість адаптивності:

Організації, які демонструють низьку ефективність управління проектами, зазнали значних збитків (близько 12% інвестицій), що підкреслює необхідність ефективних підходів [5].

Дані того ж опитування вказують, що майже 80% ІТ-компаній використовують Agile або гібридний Agile підхід, тоді як прогнозовані (традиційні) методи склали лише 10% проектів (Рис.2.4).

Agile-методологія — це не просто набір інструментів, а філософія, що передбачає певний набір цінностей та принципів для розробки програмного забезпечення.

Маніфест Agile окреслює чотири ключові цінності, які визначають пріоритети роботи:

- Люди та взаємодії важливіші за процеси та інструменти.
- Робоче програмне забезпечення важливіше за повну та вичерпну документацію.
- Співпраця із замовником важливіша за суворе дотримання контрактних зобов'язань.
- Реакція на зміни важливіша за сліпе дотримання початкового плану.

Agile-підходи забезпечують гнучкий, ітеративний процес проектування та побудови. Agile-проекти не слідують заздалегідь спланованому процесу, а складаються з низки завдань, які розробляються, виконуються та адаптуються відповідно до поточної ситуації. Це дозволяє команді оперативно реагувати на непередбачувані зміни ринку або технічні виклики, що виникають під час розробки. Замість суворого дотримання застарілих планів, акцент зміщується на створення реальної цінності в кожній ітерації, що суттєво знижує ризик розробки продукту, який не відповідає актуальним потребам користувачів.

Частка використовуваних методологій в ІТ проєктах

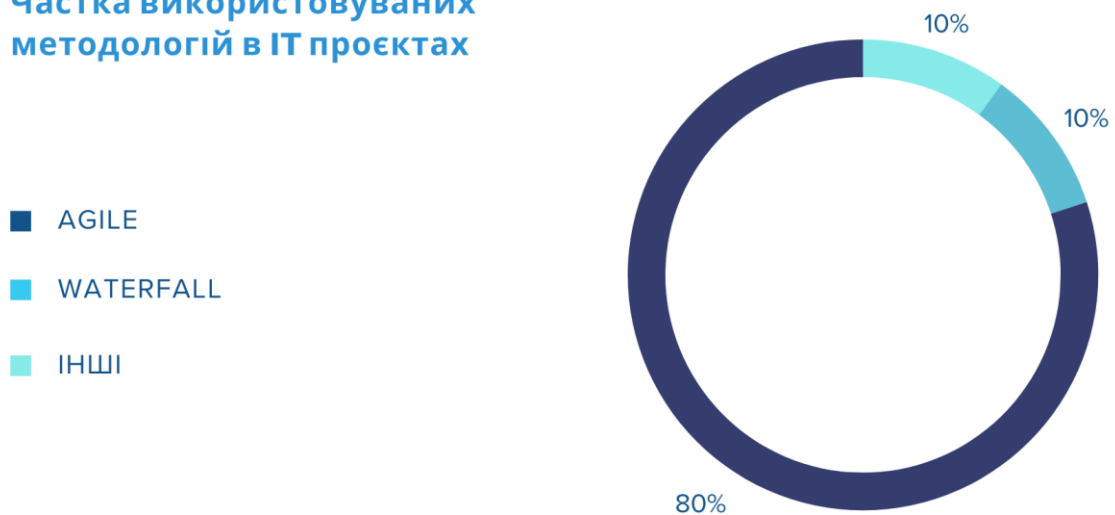


Рис.2.4 – Частка використовуваних методологій в ІТ проєктах

Основні причини запровадження згаданих методологій (Agile, Гібридні тощо) полягають у прагненні компаній збільшити продуктивність роботи над проєктами, навчитися правильно розставляти пріоритети та успішно впроваджувати інновації. (Рис. 2.5).



Рис.2.5 – Причини використання методологій

Scrum — це один із найпоширеніших фреймворків управління проектами, що пропонує чіткі принципи та процеси для покращення результатів. Він є однією з найпопулярніших та найпростіших систем для практичної реалізації принципів Agile-методології, особливо в процесі розробки програмного забезпечення. Відповідно до методології Scrum, робота поділяється на короткі, фіксовані часові інтервали, які називаються «спринтами».

Зазвичай спринт триває від 2 до 4 тижнів. Важливо, щоб тривалість була постійною протягом усього проекту — це допомагає команді вибудувати ритм і точно прогнозувати свою продуктивність. Якщо спринт триває більше місяця, цілі можуть стати занадто розмитими, складність зростає, а ризики накопичуються. Коротші цикли дозволяють швидше отримувати зворотний зв'язок від ринку.

Під час спринту не допускаються зміни, які можуть поставити під загрозу ціль спринту (Sprint Goal). Після завершення кожного спринту проводиться Sprint Review — це робоча зустріч для інспекції результатів та адаптації планів. У зустрічі беруть участь Скрам-команда та ключові стейкхолдери. Власник продукту оцінює, які елементи беклогу продукту можна вважати «Готовими» (Done). Команда демонструє функціонал, який було розроблено. Обговорюються труднощі, з якими стикнулися, та способи їх вирішення. Стейкхолдери надають відгуки: що їм подобається, а що варто змінити.

Результат засідання: оновлений беклог продукту. На основі отриманих відгуків Власник Продукту може змінити пріоритети для наступних спринтів, щоб продукт залишався конкурентоспроможним.

Опитування, проведене сайтом Scrum.org, виявило наступні тенденції використання фреймворку (Рис.2.6):

- Розмір команди: Дві третини респондентів працюють у Scrum-командах, що складаються щонайменше із семи осіб. Невеликі групи (4-5 учасників) становлять менше 20% вибірки.
- Тривалість спринту: Переважна більшість (81%) членів Scrum-команд використовують двотижневий спринт.

- Навчання: Більше 10% респондентів приділяють щонайменше 4 години на тиждень навчанню (тренінгу) інших членів команди. Водночас, 40% витрачають менше години або взагалі не займаються коучингом товаришів по команді.

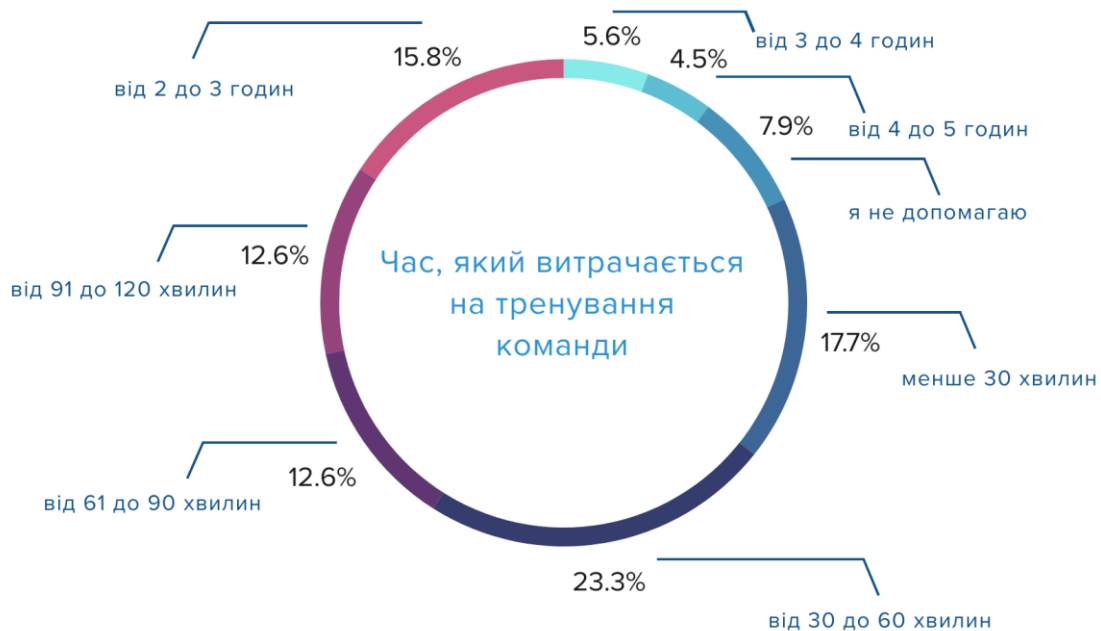


Рис.2.6 – Необхідний час на тиждень, що витрачається на тренування або підтримку окремих членів команди

Kanban — це метод управління проектами, що ґрунтується на принципах оцадливого виробництва (Lean) та суворому підвищенні ефективності. Вона має багато спільного зі Scrum, але простіша та гнучкіша. Методологія не має встановлених ролей і фокусується на покращенні пропускнуої здатності за рахунок підвищення концентрації команди. Основні практики:

- Візуалізація робочого процесу: Робота подається на Kanban-дошці (фізичній або цифровій) із простими категоріями: «To Do», «In Progress» та «Done».
- Обмеження незавершеного виробництва (WIP): Це запобігає перевантаженню команди та сприяє швидшому завершенню розпочатих завдань.

- Вимірювання часу виконання (Lead Time): Вимірюється час, що минає від моменту проходження інструктажу (початку) до завершення роботи.

Scrumban — це відносно нова гібридна методологія, яка функціонально поєднує елементи Scrum та Kanban. На відміну від традиційних спринтів Scrum із фіксованим часом, Scrumban працює на основі поповнення черги завдань на вимогу. Завдання виконуються командою в режимі потоку з обов'язковим обмеженням обсягу незавершеної роботи (WIP), що є ключовою практикою Kanban.

Такий підхід підвищує концентрацію на виконанні пріоритетних завдань і зменшує накладні витрати, пов'язані з плануванням та оглядами фіксованих спринтів (Рис.2.7).

Методологія управління проєктами Lean— це підхід, орієнтований на ефективність та мінімізацію втрат. Ця методологія допомагає підприємствам вирішувати ключові завдання, зокрема:

- Створювати високоякісну продукцію при низьких затратах.
- Скорочувати термін створення продукту.

Бар'єри, що перешкоджають запровадженню методологій

- Брак знань в даній сфері
- Небажання змін у процесі роботи
- Відсутність підтримки з боку управління
- Відсутність інструментів
- Брак часу
- Страх

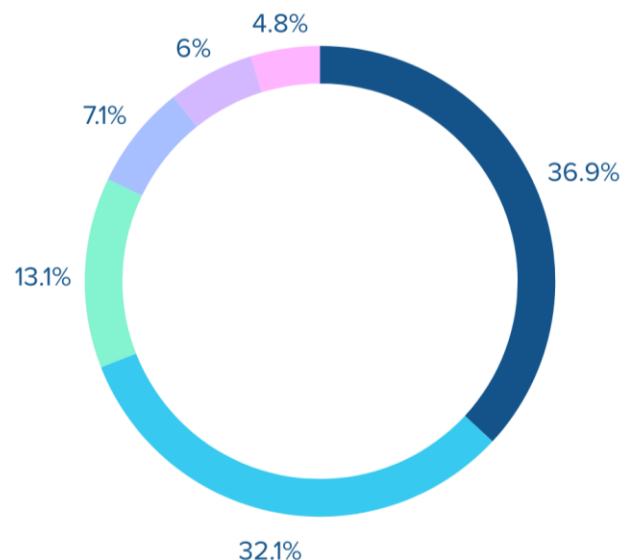


Рис.2.7 – Бар'єри, що перешкоджають запровадженню методологій

Незважаючи на доведену ефективність сучасних методологій управління проектами, їхнє повноцінне впровадження в різних галузях суспільної діяльності все ще стикається зі значними викликами. Основними перешкодами є:

- Недостатній рівень обізнаності в цій сфері.
- Опір змінам, викликаний небажанням відмовлятися від вже усталених і звичних методів роботи.

Проте, спостерігається позитивна тенденція: рівень знань та обізнаності у сфері управління проектами поступово зростає з кожним роком. В таблиці 2.2 наведено порівняння основних методологій.

Таблиця 2.2

Порівняння основних методологій

Критерій	Waterfall (Традиційна)	Scrum (Гнучка)	Kanban (Гнучка)	Scrumban (Гібридна)	Lean	PRINCE2
Основна філософія	Послідовне виконання фаз, фіксація вимог на початку.	Ітеративна та інкрементальна поставка продукту.	Безперервний потік, візуалізація та обмеження роботи.	Поєднання структури Scrum та потоку Kanban.	Максимізація цінності через усунення всіх втрат.	Чітке управління, фокус на бізнес-обґрунтуванні та продуктах.
Жорсткість вимог	Висока (зміни дорогі).	Низька (зміни вітаються).	Низька (легко адаптуватися до змін).	Низька (гнучкість).	Низька (фокус на цінності для клієнта, а не на фіксованому ТЗ).	Середня (вимоги фіксуються на початку кожної стадії).
Планування та цикли	Одна довга фаза планування.	Короткі, фіксовані часові Спринти (1-4 тижні).	Постійний потік, планування на вимогу.	Планування на вимогу, використовує певні події Scrum.	Фокус на швидкому потоці та тягучій системі (pull system).	Планування поділене на керовані стадії.

Продовження Таблиці 2.2

Пріоритет	Дотримання початкового плану та бюджету.	Поставка цінності клієнту та швидкий зворотний зв'язок.	Швидкість проходження завдання через систему.	Зосередженість на завершенні поточного завдання.	Ефективність та уникнення зайвих дій.	Контроль та управління ризиками.
Зворотний зв'язок	Рідкісний (наприкінці проєкту).	Частий (після кожного Спринту: Review).	Постійний (візуальна дошка завжди актуальна).	Частий (залежно від використання подій Scrum).	Постійний (фокус на навчанні та покращенні).	Наприкінці кожної стадії.
Командні ролі	Менеджер Проєкту, функціональні фахівці.	Власник Продукту (PO), Scrum-майстер (SM), Команда Розробки.	Відсутні формальні ролі (команда самоорганізована).	Гібрид ролей Scrum, але з меншим акцентом на SM.	Лідер (Leader), Команда (Team) – фокус на самовдосконаленні.	Виконавчий, Менеджер Проєкту, Старший Користувач.
Підходить для:	Невеликі, регульовані проєкти; будівництво.	Створення ПЗ, розробка нових продуктів, R&D.	Підтримка, обслуговування, процеси з постійним потоком завдань.	Зрілі команди, яким потрібен потік, але збереження ітеративності.	Будь-яка галузь, де необхідно оптимізувати процеси та скоротити витрати.	Великі, складні проєкти (часто державні), де потрібен високий рівень контролю та аудиту.
Ключові недоліки	Негнучкість, високий ризик.	Ризик "розповзання" обсягу, потреба у високій самоорганізації.	Відсутність фіксованого терміну може сповільнювати проєкт.	Може бути нечіткою, якщо принципи не дотримуються суворо.	Культурний зсув (важко змінити мислення), вимагає глибокого аналізу процесів.	Бюрократія, великий обсяг документації.

Для обґрунтованого вибору методології управління, необхідно врахувати ключові особливості проєкту, наведені в таблиці 2.3, розробки веб-платформи для громадської організації з питань урбаністики:

1. Нечіткість вимог: Громадські організації часто мають динамічні потреби. Вимоги до функціоналу платформи (наприклад, інтерактивні

карти, форми збору даних) можуть змінюватися або уточнюватися після перших тестів та отримання зворотного зв'язку від громади та стейкхолдерів.

2. Обмежені ресурси та бюджет: Проєкти ГО зазвичай мають фіксований або обмежений бюджет та невелику команду. Це вимагає максимальної ефективності та мінімізації втрат.
3. Потреба у швидкій цінності: Існує потреба у швидкому запуску мінімально життєздатного продукту (MVP), щоб почати збирати дані та залучати волонтерів якомога раніше.
4. Фокус на процесі: Успіх залежить не тільки від фінального продукту, а й від ефективності процесу розробки та комунікації.

Таблиця 2.3

Дані на основі специфіки проєкту

Методологія	Відповідність Специфіці Проєкту	Обґрунтування
Waterfall	Низька	Непридатна через високу ймовірність зміни вимог. Фіксація ТЗ на початку призведе до дорогих переробок.
Scrum	Висока	Підходить завдяки ітеративному підходу, дозволяє гнучко реагувати на зміни та швидко постачати MVP.
Kanban	Висока	Ідеальна для процесів, орієнтованих на потік завдань та мінімізацію WIP (втрат), що відповідає принципам Lean.
Lean	Висока	Основні принципи (мінімізація втрат, максимізація цінності) ідеально відповідають бюджетним обмеженням ГО.

Для проєкту створення веб-платформи для ГО була обрана гібридна модель управління Scrumban, як найбільш адаптивний та ефективний підхід. Аргументація вибору Scrumban:

1. Гнучкість та адаптивність (Agile-складова): Scrumban, як і Scrum, дозволяє гнучко інтегрувати нові вимоги та зворотний зв'язок від громадськості без необхідності перезатвердження великих планів.
2. Ефективність та усунення втрат (Kanban-складова): Використання принципів Kanban та обмеження незавершеної роботи (WIP) гарантує,

що команда фокусується на завершенні поточних завдань, мінімізуючи «втрати» від перемикання контексту та незавершеного коду (відповідність принципам Lean).

3. Планування на вимогу (Pull System): Відсутність жорстких спринтів дозволяє команді тягнути нові завдання з бек логу тоді, коли вона готова, а не за фіксованим календарем. Це особливо корисно для невеликих команд, які можуть мати нерівномірне навантаження.
4. Прозорість: Візуальна дошка (як у Kanban) забезпечує повну прозорість для всіх стейкхолдерів щодо поточного статусу розробки.

Методологія Scrumban найкраще відповідає потребам проєкту, оскільки поєднує гнучкість Scrum (для реагування на змінні вимоги) та ефективність Kanban/Lean (для мінімізації витрат та оптимізації потоку роботи), що є критично важливим для успішної реалізації ІТ-проєкту в умовах обмежених ресурсів громадської організації.

2.3. Огляд та вибір інструментарію управління проєктом.

Вибір інструментів управління проєктом (ПМ) має ґрунтуватися на специфіці проєкту та обраній методології Scrumban, як зазначено в пункті 1.7, як найбільш доцільну для ІТ-розробки з гнучкими вимогами. Ключові вимоги до ПМ-інструментарію, наведено в таблиці 2.4:

1. Підтримка Agile: Можливість створення дощок (Kanban/Scrum), управління спринтами, та ведення бек логу (Backlog).
2. Прозорість та візуалізація: Забезпечення легкої візуалізації статусу завдань (WIP), прогресу та виявлення вузьких місць.
3. Співпраця: Наявність функцій для ефективної командної комунікації, призначення завдань та відстеження часу.
4. Доступність та вартість: Оскільки проєкт реалізується для громадської організації, перевага надається інструментам, які мають безкоштовний

або доступний тарифний план для малих команд/некомерційних організацій.

Таблиця 2.4

Основні категорії інструментів

Інструмент	Основний Фокус	Ключові Переваги	Обмеження
Jira	Комплексний Agile-інструмент	Глибока інтеграція зі Scrum/Kanban, потужне звітування, кастомізація робочих процесів.	Складна для швидкого освоєння, платні тарифи швидко зростають зі збільшенням команди.
Trello	Візуальний Kanban-інструмент	Надзвичайно простий інтерфейс (drag-and-drop), ідеально підходить для Kanban та особистого управління.	Обмежені функції звітності, слабко підходить для суворого Scrum-управління.
Asana	Управління завданнями та портфоліо	Добре поєднує списки завдань та візуалізацію (дошки), має чистий інтерфейс, сильні функції для управління портфоліо проєктів.	Безкоштовна версія має суттєві обмеження, фокус на завданнях, а не на коді.
Confluence	Управління знаннями та документацією	Тісна інтеграція з Jira, потужні функції для спільного створення документації (wiki-стиль), шаблони для вимог та протоколів.	Вимагає окремої підписки від Jira, може стати неорганізованим без чіткої структури, є платним.
Notion	Єдиний робочий простір	Висока гнучкість, поєднує можливості Wiki, таблиць (баз даних) та завдань, має щедрий безкоштовний план.	Може бути складно стандартизувати робочі процеси, деякі команди вважають його занадто гнучким, менш потужне звітування, ніж у Jira.

Для проєкту розробки веб-платформи для ГО, інструменти Jira/Trello/Asana використовуються для виконання (трекінг завдань), тоді як Confluence або Notion є життєво необхідними для управління знаннями: зберігання фінальних вимог (ТЗ), протоколів зустрічей, дизайну архітектури та інструкцій для користувачів. Враховуючи вимоги до гнучкості, необхідність підтримки Scrum/Scrumban та бюджетні обмеження громадської організації, найбільш збалансованим вибором є Trello та Notion, завдяки чому буде отримано:

1. Простота та швидкість впровадження: Команда може почати працювати негайно без тривалого навчання, що критично для ГО.

2. Підтримка Kanban/Scrumban: Trello ідеально підходить для візуалізації робочого потоку (наприклад, колонки "Backlog", "To Do", "In Progress", "Code Review", "Done").
3. Вартість: Безкоштовний тарифний план Trello покриває базові потреби малої команди розробників веб-платформи
4. Візуалізація WIP: Чітка дошка Trello забезпечує миттєву прозорість щодо того, хто і над чим працює, що відповідає принципам Lean та Kanban.

Якщо в майбутньому команда розробки збільшиться або проєкт потребуватиме суворішого контролю та деталізованого звітування (наприклад, графіків Burndown Chart, аналізу швидкості команди), організація має бути готовою до переходу на Jira.

Для проєкту розробки веб-платформи для громадської організації, зважаючи на пріоритет швидкого старту, гнучкості та економічності, основним інструментом управління проєктом обирається Trello як платформа для візуалізації завдань, підтримки Scrumban та ефективної командної співпраці.

2.4. Визначення вимог до продукту IT проєкту.

Вимоги до інформаційної системи розподіляють на функціональні та нефункціональні [10].

Функціональні вимоги — це конкретні вказівки, які описують поведінку, функції та операції, які має виконувати програмне забезпечення або система. У розробці програмного забезпечення та систем ці вимоги окреслюють те, що має робити система, щоб задовольнити потреби користувачів і бізнес-цілі.

Функціональні вимоги зазвичай детально описуються в документації, такій як специфікація вимог, щоб забезпечити ясність і узгодженість між командами розробників, зацікавленими сторонами та кінцевими користувачами.

У розробці програмного забезпечення та систем функціональні вимоги визначають основні функції або дії, які повинна виконувати система. Вони можуть варіюватися від взаємодії з користувачем, обробки даних, обчислень та інтеграції з іншими системами до кроків, пов'язаних із завершенням робочого процесу. Вони відрізняються від нефункціональних вимог, які зосереджуються на таких атрибутах системи, як продуктивність, зручність використання та безпека. Функціональні вимоги відповідають на те, «що» і «як» поведінки системи, щоб забезпечити схему розробки та тестування.

Функціональні вимоги мають вирішальне значення для успіху продукту, оскільки вони служать основою для процесу проектування та розробки. Чіткі, чітко визначені функціональні вимоги допомагають:

- a) Об'єднати зацікавлених сторін для створення кращого розуміння між розробниками, менеджерами продуктів і зацікавленими сторонами, гарантуючи, що кінцевий продукт відповідає очікуванням.
- b) Мінімізація ризиків розвитку чітко визначає, що повинна робити система, функціональні вимоги зменшують неоднозначність, знижуючи ризик дорогих помилок і переробок.
- c) Точне тестування та перевірку вимог пропонують вимірні критерії для тестування, гарантуючи, що кожна функція працює належним чином.
- d) Підвищення задоволеності користувачів завдяки чітко визначеним вимогам. Користувачі, швидше за все, знайдуть програмне забезпечення інтуїтивно зрозумілим і узгодженим з їхніми потребами, що веде до більшого задоволення.

Таким чином, функціональні вимоги є життєво важливими для створення продуктів, які не тільки правильно функціонують, але й відповідають очікуванням користувачів, забезпечуючи успішний результат проектів розробки програмного забезпечення.

При цьому функціональні вимоги визначають що система повинна робити, опис нефункціональних вимог як система повинна працювати.

- Функціональні вимоги: Зосередженість на діях, поведінці та результатах роботи системи.
- Нефункціональні вимоги: Зосередженість на якості та стандартах продуктивності, таких як безпека, зручність використання, надійність і масштабованість.

Розуміння цих характеристик функціональних вимог і їх відмінностей від нефункціональних вимог, наведених в таблицях 2.5 та 2.6, допомагає командам розробників створити рішення, яке не тільки відповідає функціональним потребам, але й працює ефективно та надійно.

Таблиця 2.5

Функціональні вимоги проєкту

№	Назва	Опис
FR001	Управління контентом проєктів	Авторизовані менеджери повинні мати змогу створювати, редагувати та публікувати детальні сторінки проєктів (текст, зображення, відео, файли).
FR002	Перегляд інформації про проєкти	Відвідувачі сайту повинні мати змогу переглядати список активних проєктів та деталі кожного з них.
FR003	Автентифікація адміністраторів	Система повинна забезпечувати безпечний вхід для адміністраторів та контент-менеджерів.
FR004	Функція пошуку проєктів	Користувачі повинні мати змогу шукати проєкти за ключовими словами.
FR005	Модуль збору пожертв (Інтеграція)	Реалізація безпечної інтеграції з платіжним шлюзом для прийому благодійних внесків.
FR006	Модуль новин та анонсів	Повинен бути розділ для публікації новин та анонсів подій організації.
FR007	Фільтрація проєктів	Користувачі повинні мати змогу фільтрувати проєкти за статусом (активний, завершений) та тематикою (урбаністика, освіта, допомога).
FR008	Керування заявками волонтерів	Менеджери повинні мати змогу переглядати та обробляти заявки на волонтерство.
FR009	Модуль підписки на розсилку	Інтеграція форми для збору email-адрес з метою регулярної розсилки новин.

Продовження Таблиці 2.5

FR010	Публікація офіційних звітів	Окремий розділ для розміщення фінансових та річних звітів організації у форматі PDF.
FR011	Автоматична зміна статусу проєкту	Система має автоматично змінювати статус проєкту на "Завершений" після настання кінцевої дати.
FR012	Кнопки для шерингу у соцмережах	Наявність кнопок для поширення сторінок проєктів та новин у соціальних мережах.
FR013	Система реєстрації на події	Модуль для реєстрації відвідувачів на анонсовані заходи (з підтвердженням на email).
FR014	Багатомовність (UA/EN)	Підтримка двох мовних версій (українська та англійська) для ключового контенту.
FR015	Форма зворотного зв'язку для партнерів	Спеціалізована форма для подання пропозицій щодо співпраці з організацією.
FR016	Система коментарів	Можливість додавання коментарів до новин чи проєктів (з обов'язковою модерацією).
FR017	Галерея зображень та відео	Модуль для відображення медіа-контенту у форматі галереї.
FR018	Карта локацій проєктів	Інтерактивна карта (на базі Google Maps API) для відображення географії реалізації проєктів.
FR019	Управління мета-тегами для SEO	Адміністратор повинен мати змогу редагувати Title, Description та ключові слова для кожної сторінки.
FR020	Модуль "Наша команда"	Розділ для представлення членів команди з їхніми ролями та контактами.

Таблиця 2.6

Нефункціональні вимоги проєкту

№	Назва	Опис
NFR001	Адаптивність (Responsive Design)	Платформа повинна коректно відображатися та функціонувати на всіх типах пристроїв (мобільні, планшети, десктопи).
NFR002	Безпека (SSL/HTTPS)	Весь трафік повинен бути зашифрований за допомогою SSL-сертифіката.
NFR003	Надійність (Uptime)	Доступність системи повинна становити не менше 99.5% протягом місяця.
NFR004	Продуктивність (Час завантаження)	Час завантаження основних сторінок не повинен перевищувати 3 секунди при типовому інтернет-з'єднанні.
NFR005	Зручність користування (Usability)	Інтерфейс повинен бути інтуїтивно зрозумілим та простим для цільової аудиторії (відвідувачів та адміністраторів).
NFR006	Резервне копіювання	Система повинна забезпечувати щоденне автоматичне резервне копіювання бази даних та медіа-контенту з можливістю відновлення.

Продовження Таблиці 2.6

NFR007	Кросбраузерність	Система повинна коректно функціонувати в останніх двох актуальних версіях основних браузерів (Chrome, Firefox, Safari).
NFR008	Захист від атак	Код повинен бути стійким до типових веб-уразливостей.
NFR009	Доступність (Accessibility)	Інтерфейс повинен відповідати базовим стандартам WCAG 2.1 для забезпечення доступності для людей з порушеннями зору/слуху.
NFR010	Простота управління контентом (CMS)	CMS для адміністраторів має бути WYSIWYG і дозволяти публікацію без знання коду.
NFR011	Сумісність з Google Analytics	Легка інтеграція з Google Analytics 4 для відстеження статистики.
NFR012	Масштабованість навантаження	Архітектура має витримувати одночасне навантаження до 100 користувачів без зниження швидкості.
NFR013	Оптимізація для SEO	Дотримання технічних вимог SEO (чисті URL, швидкість, мета-теги) для високої видимості в пошукових системах.
NFR014	Технічна документація	Надання повної інструкції з розгортання та інструкції користувача для бек-енду.
NFR015	Конфіденційність (GDPR-сумісність)	Зберігання персональних даних (email, імена волонтерів) повинно відповідати вимогам конфіденційності.
NFR016	Продуктивність (Оптимізація медіа)	Система повинна автоматично оптимізувати та стискати завантажені зображення для зменшення часу завантаження.
NFR017	Система сповіщень	Надсилання автоматичних email-сповіщень адміністраторам про нові заявки волонтерів або пожертви.
NFR018	Модульність та Довговічність	Код повинен бути модульним та коментованим, щоб полегшити майбутні доопрацювання та інтеграції.
NFR019	Сумісність з API (Maps)	Легка інтеграція з API третіх сторін (наприклад, Google Maps API для відображення локацій).
NFR020	Економічність хостингу	Система повинна працювати на стандартних комерційних хостингових рішеннях для мінімізації операційних витрат.

2.5. Формування Use Case елементів до функціональних вимог.

Кожен Use Case елемент визначає конкретний аспект взаємодії користувача (актора) із системою, що дозволяє чітко сформулювати необхідну функціональність, наведено в таблицях 2.7-2.26.

Наступний рис. 2.8 графічно ілюструє повний набір Use Case, їх зв'язок між собою та з відповідними акторами, слугуючи основою для подальшого проєктування та реалізації системи.

Варіантами використання (Use Cases) є одиниці функціональності, які приносять вимірюваний результат для актора. Вони допомагають уникнути надмірної деталізації коду на ранніх етапах, фокусуючись на тому, що система має робити, а не як.

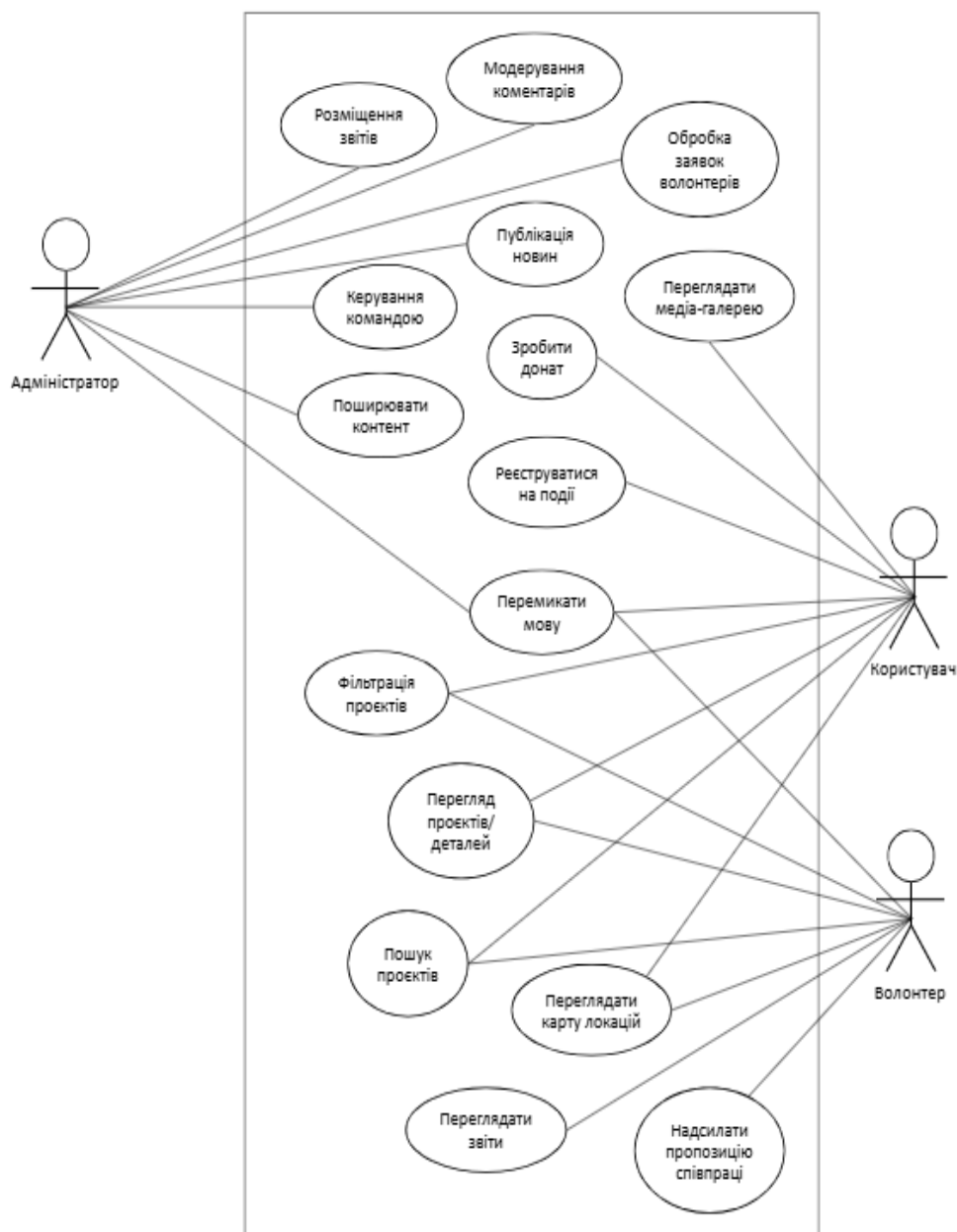


Рис. 2.8. – Use Case діаграма розроблюваного проєкт

Use Case: FR001 - Управління контентом проєктів

Категорія	Опис
Summary	Управління життєвим циклом контенту проєктів (створення, редагування, публікація, архівування).
Primary Actor	Авторизований користувач (адміністратор/контент-менеджер)
Precondition	Користувач успішно авторизований та має права доступу.
Postcondition	Контент проєкту успішно створено, змінено, опубліковано або архівовано.
Main Flow	<ol style="list-style-type: none"> 1: Користувач переходить до розділу управління проєктами. 2: Система відображає інтерфейс управління. 3: Користувач обирає дію (Створити/Редагувати/Архівувати). 4: Користувач вводить/змінює дані та натискає "Зберегти" або "Опублікувати". 5: Система зберігає зміни та підтверджує успішне виконання.
Alternative Flow	4a: Помилка введення даних (наприклад, пропущено назву). Система блокує збереження та відображає повідомлення про помилку.

Use Case: FR002 - Перегляд інформації про проєкти

Категорія	Опис
Summary	Забезпечення доступу відвідувачів до списку та детальної інформації про активні проєкти.
Primary Actor	Відвідувач сайту (будь-який користувач)
Precondition	Платформа доступна для перегляду. Є хоча б один активний проєкт.
Postcondition	Відвідувач успішно переглянув список та деталі обраного проєкту.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач заходить на сторінку "Проєкти". 2: Система відображає список усіх активних проєктів. 3: Відвідувач обирає проєкт. 4: Система завантажує та відображає детальну сторінку проєкту.
Alternative Flow	2a: Немає активних проєктів. Система відображає інформаційне повідомлення замість списку.

Use Case: FR003 - Автентифікація адміністраторів

Категорія	Опис
Summary	Безпечний вхід адміністратора або контент-менеджера в систему.
Primary Actor	Авторизований користувач (адміністратор/контент-менеджер)
Precondition	Користувач має дійсні облікові дані (логін/пароль).
Postcondition	Користувач успішно ввійшов у систему та перенаправлений у панель управління.
Main Flow	<ol style="list-style-type: none"> 1: Користувач вводить логін та пароль. 2: Система перевіряє облікові дані. 3: Якщо дані вірні, система створює сесію та надає доступ до панелі управління.
Alternative Flow	2a: Невірні облікові дані. Система відображає повідомлення про помилку "Невірний логін або пароль".

Use Case: FR004 - Функція пошуку проєктів

Категорія	Опис
Summary	Дозволяє користувачам швидко знаходити проєкти за ключовими словами.
Primary Actor	Відвідувач сайту
Precondition	На сайті є активні проєкти.
Postcondition	Відвідувач бачить список проєктів, що відповідають його пошуковому запиту.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач вводить ключові слова у рядок пошуку. 2: Користувач натискає "Пошук". 3: Система сканує вміст проєктів та відображає сторінку з результатами пошуку.
Alternative Flow	3a: Не знайдено результатів. Система відображає повідомлення "За вашим запитом нічого не знайдено".

Use Case: FR005 - Модуль збору пожертв (Інтеграція)

Категорія	Опис
Summary	Забезпечення безпечного прийому благодійних внесків через платіжний шлюз.
Primary Actor	Відвідувач сайту (людина, яка хоче зробити донат)
Precondition	Модуль пожертв налаштований та інтегрований з платіжним шлюзом.
Postcondition	Транзакція успішно завершена. Кошти перераховано.
Main Flow	<ol style="list-style-type: none"> 1: Користувач обирає суму пожертви та натискає "Задонатити". 2: Система перенаправляє користувача на сторінку платіжного шлюзу. 3: Користувач вводить платіжні дані та підтверджує транзакцію. 4: Платіжний шлюз підтверджує успішне списання коштів. 5: Система повертає користувача на сайт з повідомленням "Дякуємо за донат!".
Alternative Flow	4a: Помилка транзакції. Система повертає повідомлення про помилку ("Платіж не пройшов") та пропонує повторити спробу.

Use Case: FR006 - Модуль новин та анонсів

Категорія	Опис
Summary	Перегляд актуальних новин та анонсів подій.
Primary Actor	Відвідувач сайту
Precondition	Розділ "Новини/Події" має опублікований контент.
Postcondition	Відвідувач переглянув список новин та/або деталі конкретної новини чи події.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач переходить у розділ "Новини" або "Події". 2: Система відображає хронологічний список усіх публікацій. 3: Відвідувач обирає новину. 4: Система відображає повний текст та медіа-контент новини.
Alternative Flow	2a: Відвідувач використовує фільтр за датою/категорією події. Система відображає відфільтрований список.

Use Case: FR007 - Фільтрація проєктів

Категорія	Опис
Summary	Звуження списку проєктів за заданими критеріями (статус, тематика).
Primary Actor	Відвідувач сайту
Precondition	На сторінці проєктів відображаються фільтри.
Postcondition	На сторінці відображається лише список проєктів, що відповідають критеріям фільтрації.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач переходить на сторінку проєктів. 2: Відвідувач обирає критерій фільтра. 3: Система миттєво або після натискання кнопки "Застосувати" оновлює список проєктів.
Alternative Flow	2a: Скинути фільтри. Відвідувач натискає "Скинути фільтри". Система повертає повний список проєктів.

Use Case: FR008 - Керування заявками волонтерів

Категорія	Опис
Summary	Обробка, перегляд та зміна статусу заявок на волонтерство менеджером.
Primary Actor	Менеджер
Precondition	Менеджер авторизований та має права доступу до розділу "Заявки". Є нові/необроблені заявки.
Postcondition	Заявка оброблена, її статус оновлено, волонтер сповіщений про рішення.
Main Flow	<ol style="list-style-type: none"> 1: Менеджер переходить у розділ "Заявки волонтерів". 2: Менеджер обирає заявку для перегляду деталей. 3: Менеджер змінює статус (Затверджено/Відхилено) та додає коментар (якщо потрібно). 4: Менеджер зберігає зміни. 5: Система надсилає волонтеру email-сповіщення про зміну статусу.
Alternative Flow	3a: Відхилення заявки. Менеджер встановлює статус "Відхилено" та обов'язково вводить причину у текстове поле.

Use Case: FR009 - Модуль підписки на розсилку

Категорія	Опис
Summary	Збір email-адрес відвідувачів для регулярної розсилки новин.
Primary Actor	Відвідувач сайту
Precondition	На сайті відображається форма підписки.
Postcondition	Email-адреса додана до бази даних розсилки. Користувач отримує підтвердження.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач вводить свій email у форму підписки. 2: Відвідувач натискає "Підписатися". 3: Система зберігає email та надсилає його до інтегрованого сервісу розсилки. 4: Система відображає повідомлення "Дякуємо за підписку!".
Alternative Flow	1a: Невірний формат email. Система відображає повідомлення про помилку валідації.

Таблиця 2.16

Use Case: FR010 - Публікація офіційних звітів

Категорія	Опис
Summary	Забезпечення доступу до офіційних та фінансових документів організації для прозорості.
Primary Actor	Відвідувач сайту
Precondition	На сайті є розділ "Прозорість/Звіти".
Postcondition	Відвідувач успішно переглянув/завантажив обраний звіт.
Main Flow	1: Відвідувач переходить до розділу "Звіти". 2: Система відображає список доступних звітів. 3: Відвідувач натискає на посилання на звіт. 4: Браузер завантажує або відкриває PDF-документ звіту.
Alternative Flow	3a: Документ тимчасово недоступний. Система відображає повідомлення про помилку.

Таблиця 2.17

Use Case: FR011 - Автоматична зміна статусу проєкту

Категорія	Опис
Summary	Система автоматично змінює статус проєкту після настання кінцевої дати, зазначеної адміністратором.
Primary Actor	Система
Precondition	Проєкт має заповнену кінцеву дату реалізації. Настала дата, що перевищує кінцеву дату.
Postcondition	Статус проєкту змінено на "Завершений" або "В архіві".
Main Flow	1: Система запускає щоденну перевірку статусу проєктів. 2: Система ідентифікує проєкт, кінцева дата якого минула. 3: Система оновлює статус проєкту на "Завершений". 4: Проєкт більше не відображається у списку "Активні проєкти".
Alternative Flow	2a: Автоматична зміна статусу ініціює надсилання email-сповіщення адміністратору про факт завершення проєкту.

Таблиця 2.18

Use Case: FR012 - Кнопки для шерингу у соцмережах

Категорія	Опис
Summary	Надання можливості користувачам легко ділитися контентом у соціальних мережах.
Primary Actor	Відвідувач сайту
Precondition	На сторінці проєкту/новини відображаються кнопки для шерингу.
Postcondition	Користувач успішно поширив посилання на контент у обраній соціальній мережі.
Main Flow	1: Відвідувач переглядає сторінку проєкту. 2: Відвідувач натискає на іконку соціальної мережі. 3: Система відкриває нове вікно/вкладку з діалогом для публікації посилання. 4: Користувач публікує посилання.
Alternative Flow	3a: У користувача не виконано вхід у соцмережу. Система пропонує йому авторизуватися.

Таблиця 2.19

Use Case: FR013 - Система реєстрації на події

Категорія	Опис
Summary	Реєстрація відвідувачів на анонсовані заходи (школи, зустрічі).
Primary Actor	Відвідувач сайту
Precondition	На сторінці події відображається активна форма реєстрації.
Postcondition	Користувач зареєстрований, його дані збережено, він отримує підтвердження на email.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач переходить на сторінку події. 2: Відвідувач заповнює реєстраційну форму (Ім'я, Email) та погоджується на обробку даних. 3: Користувач натискає "Зареєструватися". 4: Система зберігає дані та надсилає email-підтвердження про реєстрацію.
Alternative Flow	4a: Досягнуто ліміту учасників. Система відображає повідомлення "Реєстрацію закрито" та блокує кнопку.

Таблиця 2.20

Use Case: FR014 - Багатомовність (UA/EN)

Категорія	Опис
Summary	Перемикання між українською та англійською версіями контенту.
Primary Actor	Відвідувач сайту
Precondition	Контент має бути введений в обох мовних версіях.
Postcondition	Інтерфейс та контент сторінки відображається обраною мовою.
Main Flow	<ol style="list-style-type: none"> 1: Користувач переглядає сторінку українською мовою. 2: Користувач натискає на перемикач мови (наприклад, "EN"). 3: Система відображає той самий контент, перекладений англійською мовою.
Alternative Flow	3a: Контент відсутній обраною мовою. Система відображає наявний контент із повідомленням про те, що переклад цієї сторінки відсутній.

Таблиця 2.21

Use Case: FR015 - Форма зворотного зв'язку для партнерів

Категорія	Опис
Summary	Подання пропозицій щодо співпраці від юридичних осіб чи організацій.
Primary Actor	Потенційний партнер
Precondition	На сайті доступна спеціалізована форма "Партнерство".
Postcondition	Запит партнера успішно надісланий на відповідну корпоративну пошту.
Main Flow	<ol style="list-style-type: none"> 1: Партнер переходить у розділ "Співпраця". 2: Партнер заповнює форму (Назва організації, Контактна особа, Мета співпраці). 3: Партнер натискає "Надіслати пропозицію". 4: Система надсилає email-сповіщення адміністраторам про новий запит.
Alternative Flow	2a: Не заповнено обов'язкові поля. Система відображає повідомлення про помилку.

Use Case: FR016 - Система коментарів

Категорія	Опис
Summary	Додавання коментарів до публікацій та їх модерація адміністратором.
Primary Actor	Відвідувач сайту, адміністратор
Precondition	На сторінці (новини/проекту) активована функція коментарів.
Postcondition	Коментар додано до бази даних та відображається на сторінці після модерації.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач вводить текст коментаря. 2: Відвідувач натискає "Надіслати". 3: Система зберігає коментар зі статусом "На модерації" та сповіщає Адміністратора. 4: Адміністратор затверджує коментар у панелі управління. 5: Система відображає коментар на сторінці.
Alternative Flow	4a: Адміністратор відхиляє коментар. Система видаляє коментар з бази даних.

Use Case: FR017 - Галерея зображень та відео

Категорія	Опис
Summary	Перегляд медіа-контенту проекту в інтерактивному форматі галереї.
Primary Actor	Відвідувач сайту
Precondition	Сторінка проекту містить кілька зображень/відео.
Postcondition	Відвідувач може переглянути всі медіа-файли у повноекранному режимі.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач переглядає сторінку проекту. 2: Відвідувач натискає на мініатюру зображення/відео. 3: Система відкриває інтерактивну галерею у повноекранному режимі. 4: Відвідувач перемикається між медіа-файлами за допомогою навігації галереї.
Alternative Flow	3a: Відео не завантажується. Система відображає повідомлення про помилку.

Use Case: FR018 - Карта локацій проектів

Категорія	Опис
Summary	Відображення географії реалізації проектів на інтерактивній карті.
Primary Actor	Відвідувач сайту
Precondition	На сайті інтегровано картографічний сервіс та до проектів додано географічні мітки.
Postcondition	Відвідувач бачить місце розташування проекту на карті.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач переходить на сторінку "Локації" або деталі проекту. 2: Система відображає інтерактивну карту з міткою проекту. 3: Відвідувач взаємодіє з картою (масштабує, переміщує). 4: При натисканні на мітку відображається інформація про проект.
Alternative Flow	2a: Немає дозволу API на використання карт. Система відображає статичне зображення або повідомлення про помилку.

Use Case: FR019 - Управління мета-тегами для SEO

Категорія	Опис
Summary	Адміністратор керує ключовими елементами для пошукової оптимізації (SEO).
Primary Actor	Авторизований користувач (адміністратор/SEO-менеджер)
Precondition	Користувач має права доступу до налаштувань контенту.
Postcondition	Мета-теги оновлені та збережені у вихідному коді сторінки.
Main Flow	<ol style="list-style-type: none"> 1: Адміністратор редагує сторінку проєкту. 2: Адміністратор переходить до спеціального розділу "SEO". 3: Користувач вводить або змінює Title, Description та Keywords. 4: Користувач зберігає сторінку. 5: Система оновлює дані у базі та HTML-код сторінки.
Alternative Flow	3a: Перевищено ліміт символів у Description. Система відображає попередження про перевищення ліміту.

Use Case: FR020 - Модуль "Наша команда"

Категорія	Опис
Summary	Перегляд інформації про ключових членів команди організації.
Primary Actor	Відвідувач сайту
Precondition	На сайті є розділ "Команда" з наповненим контентом.
Postcondition	Відвідувач успішно переглянув профілі членів команди та їхні ролі.
Main Flow	<ol style="list-style-type: none"> 1: Відвідувач переходить до розділу "Наша команда". 2: Система відображає список членів команди, їхні фотографії, імена та посади. 3: Відвідувач може натиснути на профіль для отримання більш детальної інформації.
Alternative Flow	2a: Профіль команди тимчасово приховано. Система перенаправляє на сторінку "Про нас".

2.6. Розробка та обґрунтування архітектурних моделей інформаційної системи.

Розробка концептуальних моделей є ключовим етапом проєктування інформаційної системи. Вона дозволяє наочно представити структуру, функції та взаємозв'язки між компонентами системи, а також визначити вимоги до її реалізації (Рис.2.9) [11].

На основі цих моделей формується архітектурне рішення, яке гарантує відповідність системи цілям організації «Місто Майбутнього» щодо ефективної взаємодії з користувачами, волонтерами та ЗМІ.

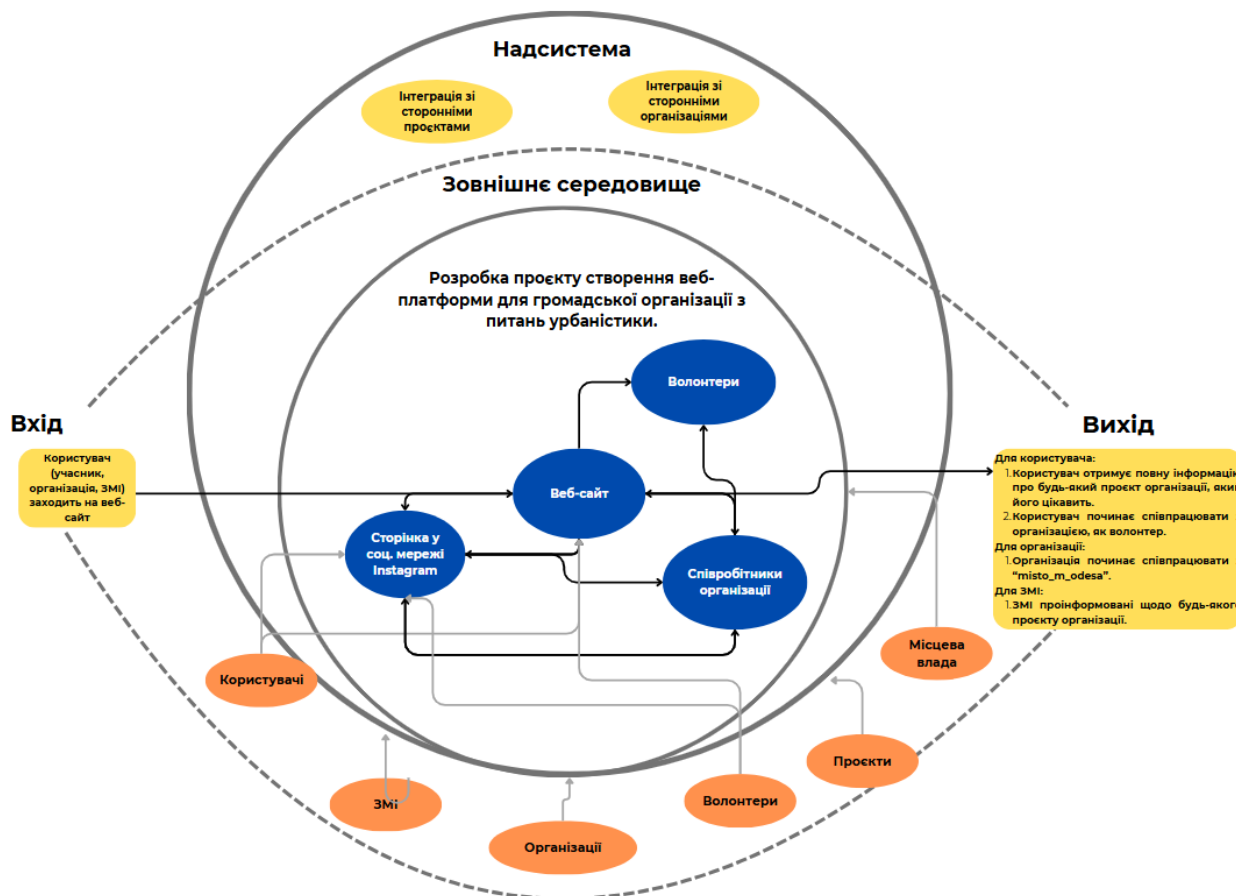


Рис. 2.9. - Процес функціонування системи

Система може бути представлена як множина взаємопов'язаних підпроцесів $P = \{p_1, p_2, \dots, p_n\}$, де кожен підпроцес p_i має вхід $I(p_i)$ та вихід $O(p_i)$. Функціонування системи описується потоками інформації $F_{ij}: O(p_i) \rightarrow I(p_j)$ між підпроцесами.

Підпроцеси як елементи системи:

- Доступ користувача до веб-сайту (Вхід I_{user}). Початкова ініціація взаємодії, що вимагає аутентифікації/авторизації або анонімного доступу до публічного контенту.

- Поширення інформації для користувачів (Вихід O_{info}). Дисемінація контенту (новини, проекти), що є ключовою функцією платформи.
- Залучення волонтерів (Вихід $O_{volunteer}$). Процес збору заявок та координації волонтерської діяльності.
- Організаційне партнерство (Вихід $O_{partner}$). Формування комунікаційних каналів та обмін даними з партнерами.
- Взаємодія зі ЗМІ/інформування (Вихід O_{media}). Спеціалізований вихід інформації для медіа-ресурсів та прес-релізів.
- Потік інформації між веб-сайтом та волонтерами/персоналом F_{web_staff} . Внутрішній цикл обробки даних: модерація контенту, управління заявками.
- Інтеграція з соціальними мережами F_{social} . Автоматизований обмін даними для розширення охоплення та залучення.

Мета математичної постановки задачі оптимізації функціонування полягає в максимізації коефіцієнта залученості η , який може бути визначений як функція від кількості успішно виконаних цільових виходів:

$$\eta = f(N_{users}, N_{volunteers}, N_{partners}) \rightarrow \max$$

де N – кількість залучених користувачів, волонтерів та партнерів відповідно, при обмеженнях на ресурсні витрати C_{total} (обчислювальні ресурси, час персоналу).

Модель предметної області є ключовим артефактом об'єктно-орієнтованого аналізу. Вона відображає основні сутності (класи) предметної області та зв'язки (асоціації) між ними, слугуючи основою для проектування бази даних та архітектури застосунку. На Рис. 2.10 представлена концептуальна схема, яка абстрагується від технічних деталей реалізації (таких як типи полів бази даних) і зосереджується на бізнес-логіці.

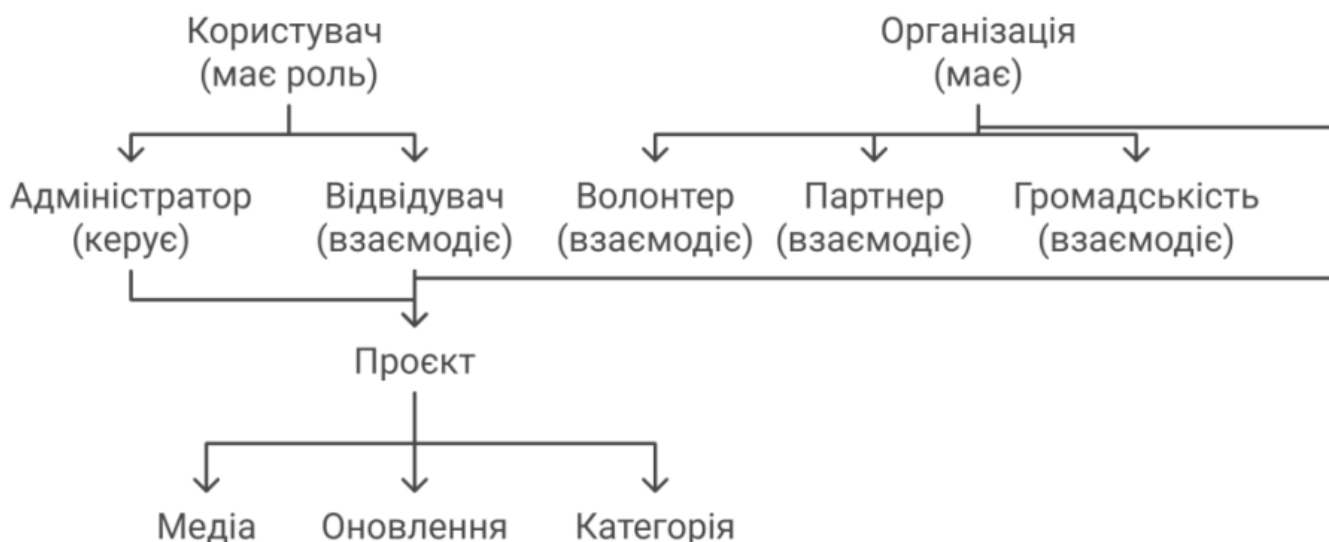


Рис. 2.10 – Модель предметної області проєкту

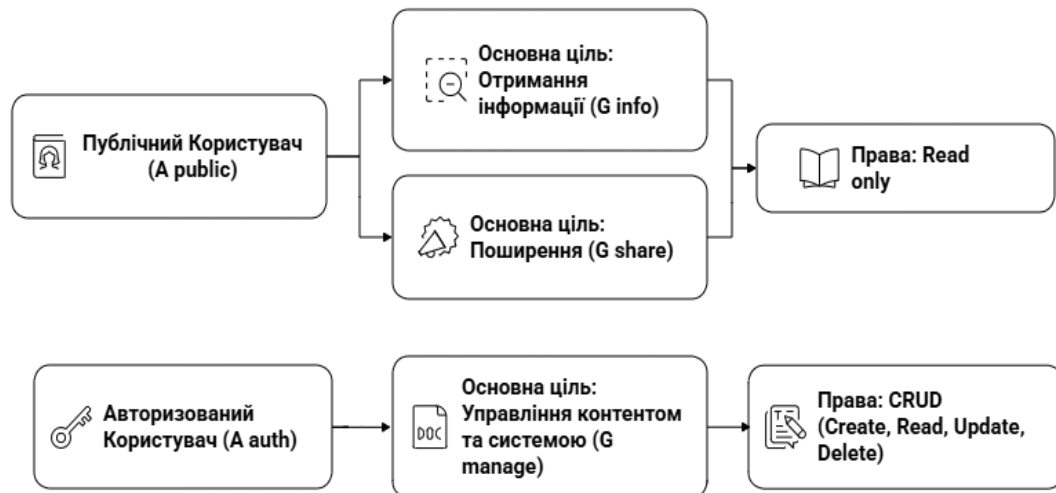
Ключові сутності моделі:

- Користувач: Головна сутність, яка може бути поділена на підтипи (Публічний, Авторизований, Адміністратор).
- Проєкт: Центральна одиниця контенту, що містить деталі про ініціативу, статус, необхідні ресурси та географічне розташування.
- Новина: Хронологічна інформаційна одиниця.
- Волонтерська заявка: Сутність для збору даних про потенційних волонтерів, пов'язана з конкретним проєктом або загальною діяльністю.

Модель взаємодії з користувачем описує поведінковий аспект системи, формалізуючи діалог між користувачем та веб-платформою [11].

1. Типи користувачів та їх інтерактивні ролі:

- а) Актори: На Рис. 2.11 представлено ієрархію акторів, кожен з яких має чітко визначений набір сценаріїв взаємодії. В основі моделі лежить принцип мінімальних привілеїв, де кожен тип користувача отримує доступ лише до необхідного інструментарію.



Made with Napkin

Рис. 2.11 – Актори

б) Інтерактивні цілі: Кожен тип користувача має набір цілей, які він прагне досягти через взаємодію з платформою. $G = \{G_{info}, G_{share}, G_{manage}\}$. Кожен тип користувача взаємодіє з платформою з метою вирішення конкретних прагматичних завдань.

2. Компоненти взаємодії: Система S в кожен момент часу перебуває в певному стані $S_i \in \Sigma$, де Σ – множина всіх можливих конфігурацій інтерфейсу та даних (наприклад, «Сторінка Проєкту», «Опитування»).

а) Інформаційний простір платформи (Рис. 2.12) — це багаторівнева структура, що організовує контент таким чином, щоб забезпечити його легку доступність та логічну зв'язність.

Навігаційна архітектура: Визначає шляхи переміщення користувача між різними розділами системи. Вона включає глобальне меню, «хлібні крихти» (breadcrumbs) та контекстні посилання.

Семантична організація: Весь контент групується за змістом (категорії проєктів, типи документів, архівні записи). Це дозволяє реалізувати ефективні алгоритми пошуку та фільтрації.

Візуальна ієрархія: Використання принципів дизайну для акцентування уваги на найважливіших елементах (наприклад, кнопка «Взяти участь» повинна бути помітнішою за допоміжну інформацію).

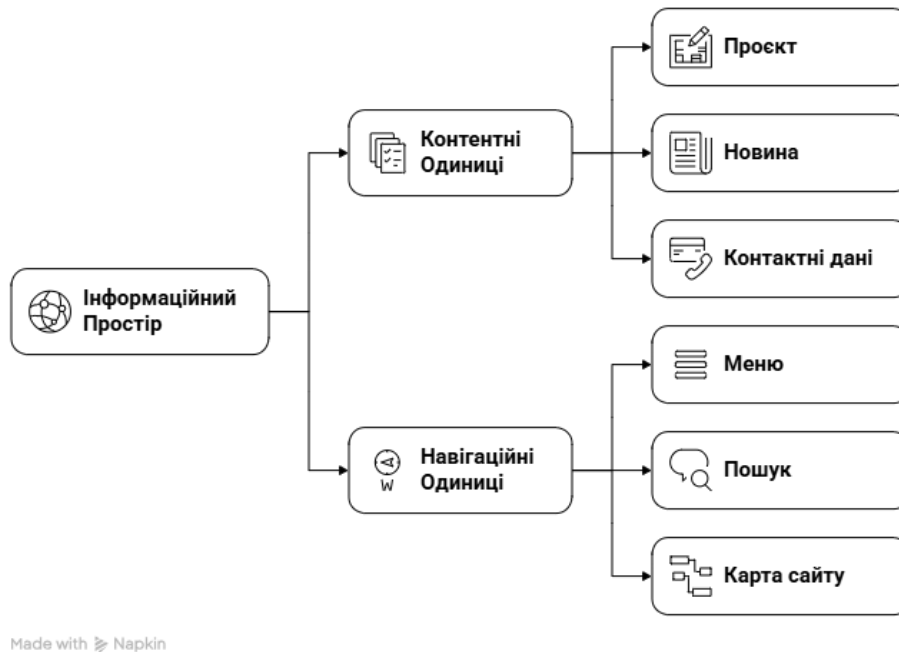


Рис. 2.12 – Інформаційний простір

а) Інтерфейсні елементи: Абстрактні візуальні та інтерактивні компоненти, через які користувач взаємодіє з системою (Рис.2.13).



Рис. 2.13 – Інтерфейсні елементи

б) Системні Стани: Відповідь системи на взаємодію користувача (Рис.2.14).



Рис. 2.14 – Системні стани

3. Моделі навігації: Навігація може бути представлена як орієнтований граф $G_{nav} = (V, E)$, де вершини V – це інформаційні та функціональні одиниці, а ребра E – це можливі переходи (посилання, кнопки).

а) Глобальна навігація: Забезпечує доступ до основних розділів платформи з будь-якої точки.

- *Структура:* Ієрархічна (головне меню).
- *Механізми:* Постійні меню, панелі.

б) Контекстна навігація: Дозволяє переміщатися між пов'язаними елементами або повертатися до попередніх станів.

- *Структура:* Зв'язки "батько-дитина", навігація "наступний/попередній".
- *Механізми:* Посилання, кнопки.

в) Пошук: Механізм для прямого доступу до конкретної інформації.

4. Модель обробки подій та зворотного зв'язку: Взаємодія є послідовністю подій E_k :

$$Interaction = \{(A_i, E_k, S_j) \rightarrow S_{j+1}\}$$

де A_i – актор, E_k – ініційована подія (клік, введення), S_j – початковий стан, S_{j+1} – новий стан системи. Зворотний зв'язок FB є частиною переходу, що інформує користувача про результат дії.

а) Ініціація події: Дії користувача, які викликають реакцію системи.

- *Пряма маніпуляція*: Клік, введення тексту, вибір елемента.
- *Непряма ініціація*: Наведення курсора, використання клавіатурних скорочень.

б) Системна реакція: Як система реагує на ініційовані події.

- *Візуальний зворотний зв'язок*: Зміна стану елемента (підсвічування кнопки, зміна курсора), оновлення контенту.
- *Зворотний зв'язок*.
- *Повідомлення про стан*: Підтвердження дії, повідомлення про помилку, індикатори завантаження.

в) Валідація та обробка помилок:

- *Валідація вводу*: Перевірка коректності даних перед їх обробкою.
- *Чіткі повідомлення про помилки*: Інформування користувача про причину помилки та можливі шляхи її виправлення.

5. Принципи інтеракції, а саме: правила та керівні ідеї, які використовуються в дизайні для створення ефективних, інтуїтивно зрозумілих та приємних для користувача цифрових продуктів. На Рис. 2.15 наведено ієрархію цих принципів, які використовуються як контрольні показники під час тестування та приймання системи. Для об'єктивної оцінки відповідності реалізованої веб-платформи принципам інтеракції (згідно з Рис. 2.15), використовуються такі показники:

- *Ефективність*: Кількість часу або кроків, необхідних користувачеві для досягнення конкретної цілі.
- *Легкість навчання*: Наскільки швидко нові користувачі можуть опанувати основні функції платформи без додаткових інструкцій.

- Задоволеність користувача: Суб'єктивна оцінка комфорту роботи з інтерфейсом.

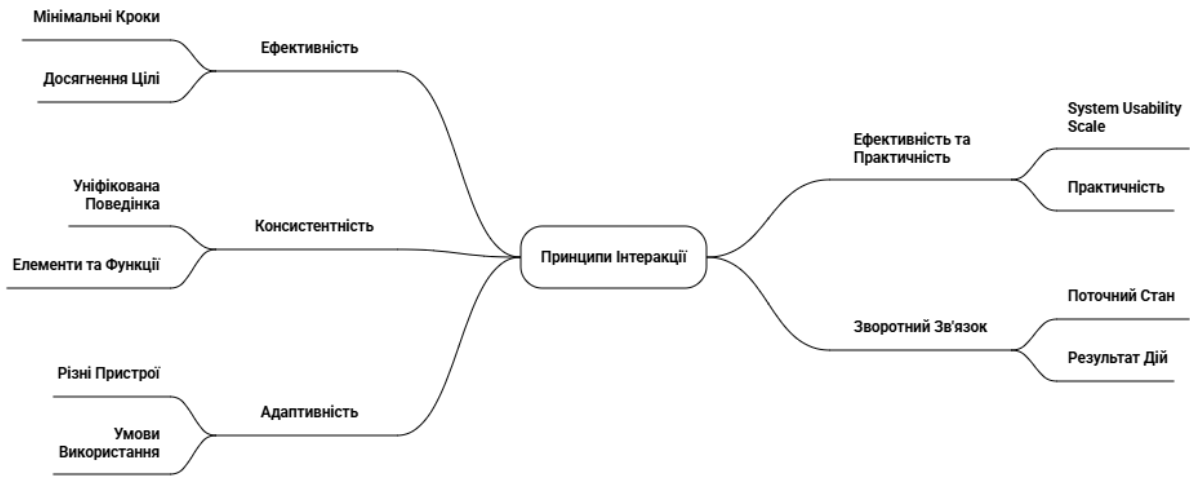


Рис. 2.15 – Принципи Інтеракції

Концептуальна модель інформаційної системи є абстрактним представленням структури та функцій системи, орієнтованим на користувача. Вона описує, які дані обробляються, які функції виконуються, та які взаємозв'язки існують між різними компонентами системи (Рис.2.16) [10-12].

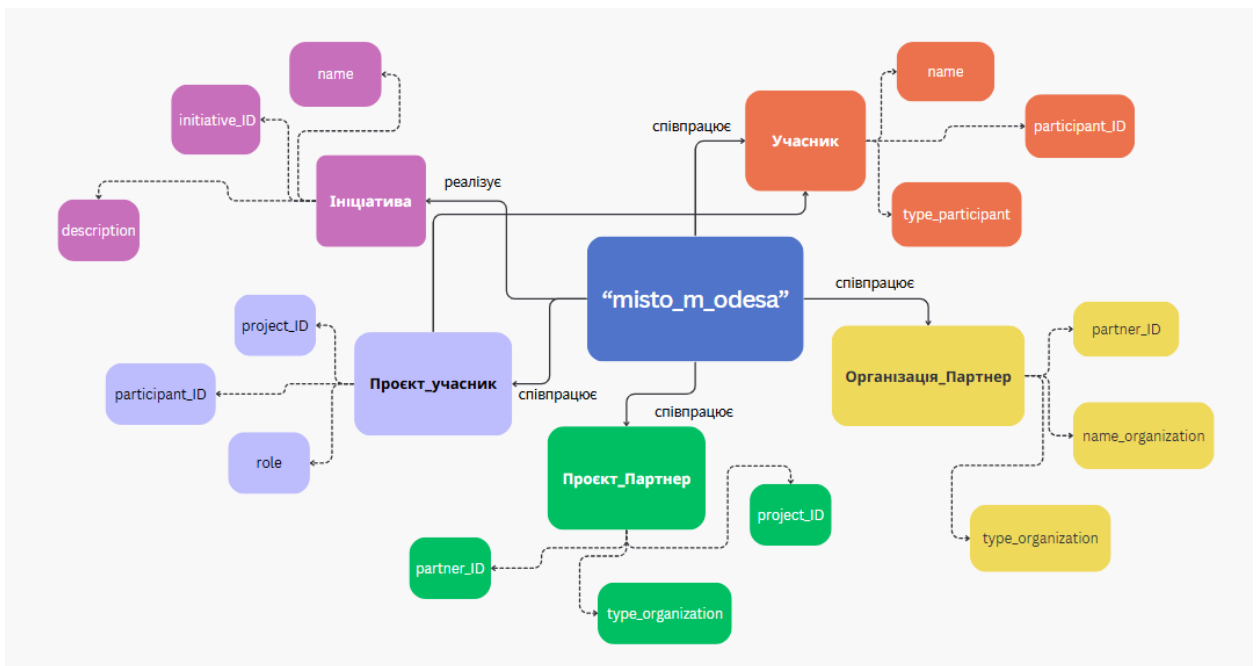


Рис. 2.16 – Концептуальна модель проекту

Логічна модель бази даних - деталізує структуру бази даних, включаючи таблиці, стовпці, типи даних, ключі та зв'язки між таблицями (Рис.2.17).

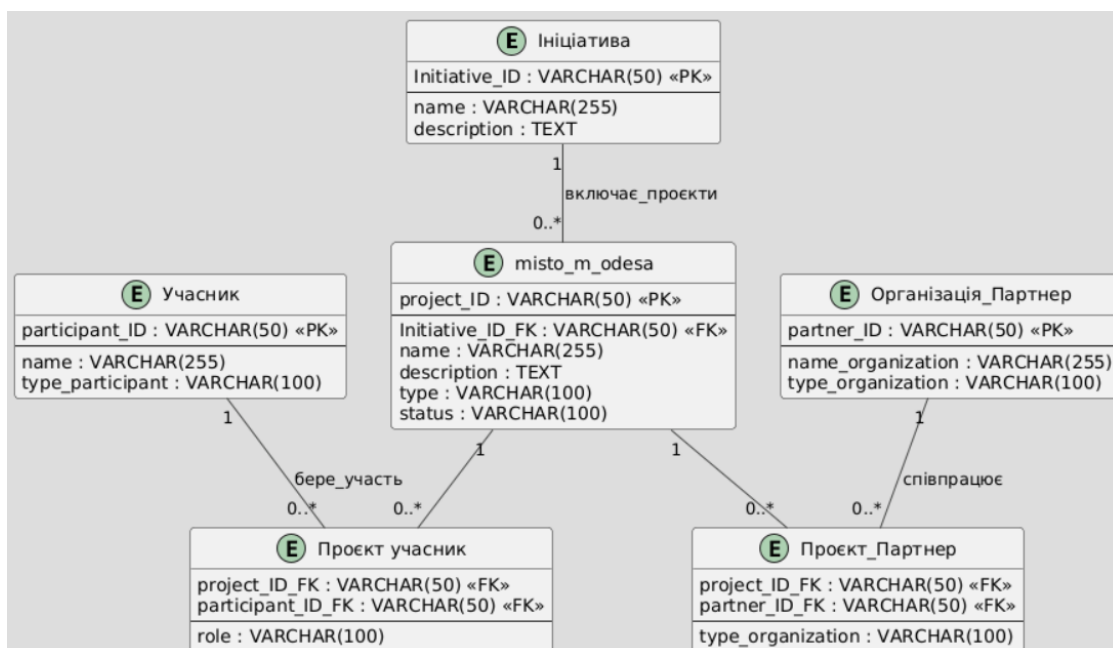


Рис. 2.17 – Логічна модель бази даних

2.7. Використання методів ШІ та моделювання розроблених моделей.

Впровадження інтелектуальних рішень у життєвий цикл розробки веб-платформи дозволяє перейти від статичного управління до динамічної оптимізації процесів. Використання методів штучного інтелекту (ШІ) на різних етапах — від аналізу вимог до розгортання та підтримки — забезпечує вищу точність прогнозування та автоматизацію рутинних операцій.

Ключовим аспектом є попереднє моделювання розроблених моделей. Перш ніж інтегрувати ШІ-алгоритми у реальне середовище, проводиться їх тестування на архітектурних прототипах. Це дозволяє:

1. Оцінити продуктивність: визначити навантаження на сервери при використанні складних нейромережових обчислень.
2. Валідувати точність: перевірити релевантність вихідних даних (наприклад, рекомендацій хабу) на основі контрольних вибірок.

3. Мінімізувати ризики: виявити потенційні "вузькі місця" в обробці даних ще до стадії фінального програмування.

Застосування методів ШІ не є ізольованим процесом; воно інтегрується в загальну Agile-структуру проєкту, де кожна ітерація моделювання дозволяє уточнити параметри системи. Нижче наведена таблиця 2.27 з детальним оглядом методів ШІ, класифікованих за етапами життєвого циклу платформи.

Таблиця 2.27

Методи штучного інтелекту (ШІ) в життєвому циклі платформи

	Аналіз даних та оптимізація взаємодії з користувачами	Пошук інформації та семантична релевантність	Моделювання сценаріїв функціонування та прогнозування поведінки
Опис	Методи машинного навчання, зокрема кластеризація та регресійний аналіз, можуть бути застосовані для збору, агрегації та глибокого аналізу даних про користувачів та їхню поведінку на сайті. Це включає аналіз шляхів навігації, часу, проведеного на сторінках проєктів, частоти переглядів конкретних категорій або оновлень, а також взаємодії з елементами зворотного зв'язку.	Методи обробки природної мови, включаючи векторизацію слів та семантичний пошук, можуть бути використані для розробки більш інтелектуальної системи пошуку, яка виходить за рамки простого співпадіння ключових слів.	Методи імітаційного моделювання з елементами агентного моделювання (Agent-based Modeling) та системної динаміки, можуть бути використані для побудови прогнозних моделей поведінки веб-платформи та її користувачів.
Користь для проєкту	1.Персоналізація контенту: Рекомендаційні системи на базі Collaborative Filtering або Content-Based Filtering ідентифікують закономірності у вподобаннях відвідувачів, пропонуючи найбільш релевантні проєкти	1.Розуміння запитів користувачів: NLP дозволяє системі розуміти синоніми, контекст запиту та наміри користувача, навіть при використанні неточних формулювань.	1.Прогнозування навантаження: Моделі можуть симулювати поведінку великої кількості одночасних користувачів для прогнозування пікових навантажень на сервер, що дозволить оптимізувати інфраструктуру та забезпечити стабільність роботи платформи.

Продовження Таблиці 2.27

Користь для проєкту	2.Оптимізація структури сайту: Аналіз "теплових карт" та асоціативних правил виявляє неефективні елементи інтерфейсу, що дозволить дизайнерам та розробникам покращити практичність та навігацію.	2.Надання найбільш релевантних результатів: Використання ранжування на основі семантичної схожості замість точного співпадіння.	2.Оцінка ефективності нових функцій: Моделі симулюють вплив нових функцій на метрики (наприклад, коефіцієнт конверсії).
Користь для проєкту	3.Прогнозування популярності контенту: Часові ряди (Time Series Analysis) та ML-моделі прогнозують, які типи проєктів зацікавлять більшу аудиторію в майбутньому, підтримуючи стратегічне планування.	3.Кластеризація та тематичне моделювання: Автоматичне групування проєктів за тематикою або виявлення прихованих зв'язків між ними, покращать функціонал категоризації та пошуку.	3.Виявлення аномалій та безпека: Алгоритми машинного навчання виявляють аномальні патерни поведінки та сигналізують про потенційні загрози безпеці.

Таким чином, інтеграція методів ШІ та комплексне моделювання дозволять не лише створити функціональну веб-платформу для організації «Місто Майбутнього», але й забезпечити її ефективну, надійну та адаптивну роботу в довгостроковій перспективі, постійно покращуючи взаємодію з користувачами та максимізуючи вплив організації [12-13]. Використання інтелектуальних агентів та прогнозних моделей перетворює платформу з простого інформаційного ресурсу на активну екосистему, здатну самостійно ідентифікувати тренди у розвитку громади. Це закладає фундамент для масштабування проєкту: від локального хабу до загальнонаціональної мережі «розумних міст». Таким чином, технологічний стек, підкріплений методами ШІ, стає не просто допоміжним інструментом, а стратегічним активом, що гарантує життєздатність концепції «Міста Майбутнього» в умовах швидкої цифровізації суспільства.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

Успішна реалізація IT-проєкту для ГО «Місто Майбутнього» вимагає гнучкого підходу, який дозволить швидко реагувати на зміни вимог та зворотню відповідь громади. Саме тому обрано гібридну методологію Scrumban, як найбільш адаптивний та ефективний підхід. Цей вибір обґрунтовано необхідністю поєднання гнучкості та швидкості реакції на змінні вимоги Agile/Scrum з ефективністю та мінімізацією втрат Lean/Kanban.

3.1 Життєвий цикл проєкту.

П'ять традиційних фаз життєвого циклу проєкту, наведених в таблиці 3.1, реалізуються з використанням ітераційного підходу Scrumban, де фази 2-4 є безперервним потоком роботи.

Таблиця 3.1

Життєвий цикл проєкту

Фаза Життєвого Циклу	Опис	Ключові Активності
Фаза 1: Ініціалізація та Планування	Визначення цілей проєкту, обґрунтування його необхідності, формування команди (для реального проєкту), збір та аналіз початкових вимог, розробка загального плану проєкту.	<ul style="list-style-type: none">- Визначення мети та завдань веб-платформи.- Аналіз цільової аудиторії та її потреб.- Формулювання ключових функціональних та нефункціональних вимог.- Оцінка ресурсів (час, потенційні технології).- Розробка попереднього плану проєкту.
Фаза 2: Аналіз та Проєктування (Дизайн)	Детальне вивчення вимог, створення архітектури системи, розробка інтерфейсів користувача та бази даних. Ця фаза є критично важливою, оскільки саме тут закладаються основи майбутньої платформи.	<ul style="list-style-type: none">- Глибокий аналіз функціональних вимог.- Розробка Use Case діаграм.- Проєктування структури бази даних.- Розробка дизайн-макетів інтерфейсу користувача (UI/UX).- Вибір технологічного стеку.- Створення технічної специфікації.

Продовження Таблиці 3.1

<p>Фаза 3: Реалізація (Розробка)</p>	<p>Безпосереднє кодування та розробка всіх компонентів веб-платформи згідно з проектними рішеннями.</p>	<ul style="list-style-type: none"> - Розробка фронтенду (клієнтська частина). - Розробка бекенду (серверна частина та бізнес-логіка). - Розробка та інтеграція бази даних. - Розробка модулів для управління контентом, користувачами, подіями тощо.
<p>Фаза 4: Тестування та Впровадження</p>	<p>Перевірка функціональності, продуктивності та безпеки системи, виправлення виявлених помилок, підготовка до запуску та розгортання платформи.</p>	<ul style="list-style-type: none"> - Модульне тестування. - Інтеграційне тестування. - Системне тестування. - Виправлення дефектів. - Розгортання платформи на сервері.
<p>Фаза 5: Експлуатація та Супровід</p>	<p>Підтримка функціональності платформи після запуску, оновлення, усунення нових проблем, додавання нового функціоналу за потребою.</p>	<ul style="list-style-type: none"> - Моніторинг продуктивності та безпеки. - Регулярне оновлення контенту та програмного забезпечення. - Виправлення помилок, що виникають під час експлуатації. - Запровадження нових функцій та покращень.

3.2. WBS робіт та віхи проєкту.

WBS (Work Breakdown Structure) – це ієрархічна декомпозиція робіт проєкту на менші, керовані компоненти. WBS дозволяє чітко визначити обсяг робіт, розподілити відповідальність та оцінити тривалість і вартість кожної задачі. Головна мета WBS — перетворити загальну ціль проєкту (створення веб-платформи) на конкретний, структурований набір результатів. Він являє собою деревоподібну структуру, де на першому рівні знаходиться сам проєкт, а на найнижчому рівні — пакети робіт, які є найменшими, повністю керованими елементами.

Основні переваги використання WBS:

1. Точне визначення обсягу забезпечує, що всі роботи, необхідні для завершення проєкту, включені, і, навпаки, виключає роботи, що не стосуються проєкту.
2. Основа для планування слугує базисом для створення графіку, оцінки ресурсів та розробки бюджету проєкту.

3. Чітко прив'язує кожен пакет робіт до конкретної команди чи виконавця, спрощуючи контроль та звітність.
4. Допомагає ідентифікувати потенційні ризики на рівні окремих, невеликих завдань.

Деталізація основних переваг:

- Усунення розмиття меж: Чітка структура WBS фіксує межі проєкту. Будь-яка нова вимога, що не вписується в ієрархію, автоматично ідентифікується як зміна обсягу, що потребує додаткового узгодження та бюджетування.
- Точність бюджетування: Оскільки WBS розбиває проєкт на дрібні пакети, вартість всього проєкту розраховується шляхом підсумовування витрат на найнижчих рівнях. Це значно точніше, ніж загальна експертна оцінка «зверху вниз».
- Оптимізація комунікацій: Кожен елемент WBS отримує свій унікальний ідентифікатор (код), що дозволяє команді однозначно посилатися на конкретний функціонал у технічній документації, звітах та баг-трекерах.
- Візуалізація критичних вузлів: Завдяки деревоподібній структурі (Рис. 3.1) стає очевидним, які компоненти є найбільш складними та потребують залучення вузькопрофільних спеціалістів.

На Рис. 3.1 відображено логічний перехід від абстрактної ідеї веб-платформи до конкретних технічних модулів. Така декомпозиція дозволяє команді уникати «хаосу розробки», коли складність системи заважає бачити загальний прогрес.

Сформована WBS слугує вхідним документом для побудови Діаграми Ганта, де пакети робіт отримують часову прив'язку та встановлюються логічні зв'язки (залежності) між ними. Без якісно опрацьованої структури робіт, представленої в цьому розділі, подальше календарне планування втрачає точність.

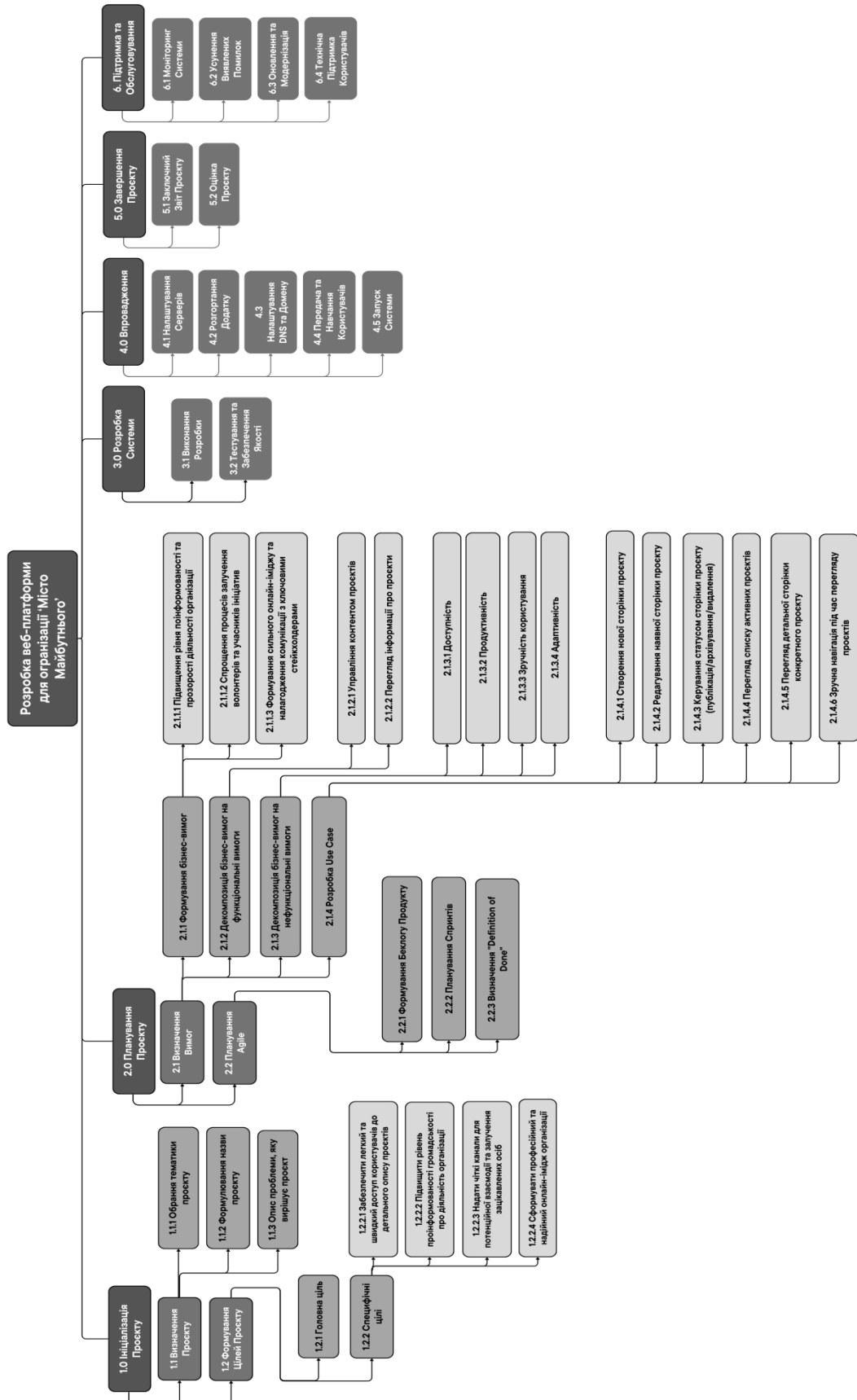


Рис. 3.1 – WBS проєкту

Кожен пакет робіт на найнижчому рівні WBS автоматично стає User Story або завданням у беклозі, яке команда "тягне" на Scrumban-дошку.

Для забезпечення контролю над розробкою та вчасного реагування на ризику, життєвий цикл проєкту розподілено на стратегічні віхи (milestones). Кожна віха являє собою завершений етап, що закінчується отриманням конкретного результату, який підлягає перевірці та затвердженню.

Нижче наведено таблицю 3.2, де визначені ключові контрольні точки, що визначають шлях від концептуалізації до повного розгортання платформи «Місто Майбутнього».

Таблиця 3.2

Віхи проєкту

Віха Проєкту (Milestone)	Опис	Очікуваний Результат
Віха 1: Затвердження концепції та вимог проєкту	Завершення первинного аналізу, узгодження основних функціональних та нефункціональних вимог до платформи.	Сформований документ з вимогами та загальною архітектурою.
Віха 2: Затвердження дизайн-макетів та структури бази даних	Повне завершення етапу проєктування, затвердження візуального дизайну та логіки зберігання даних.	Фінальні дизайн-макети (як Додаток А), схеми бази даних, технічна специфікація.
Віха 3: Завершення розробки основного функціоналу	Основна частина кодування завершена, базовий функціонал платформи реалізований.	Працюючий прототип платформи з ключовими можливостями (управління контентом, користувачами, події).
Віха 4: Завершення тестування та готовність до впровадження	Всі модулі протестовані, виявлені критичні помилки виправлені, система готова до розгортання.	Протокол тестування, відсутність критичних помилок.
Віха 5: Впровадження та запуск веб-платформи	Веб-платформа розгорнута та доступна для цільових користувачів.	Функціонуюча веб-платформа.
Віха 6: Завершення проєкту	Завершення проєкту.	Успішне завершення проєкту.

Використання методології Agile дозволяє поєднувати чіткі терміни основних віх із гнучкістю всередині кожної ітерації. Такий підхід гарантує, що архітектурні рішення, закладені на початкових етапах, будуть коректно

імplementовані в програмний код, а інтеграція методів штучного інтелекту відбудеться після стабілізації базового функціоналу.

3.3. Організаційна структура та розподіл ролей.

Успішна реалізація веб-платформи вимагає чіткої організаційної структури та розподілу відповідальності. Для цього проекту, що використовує Scrumban, пропонується компактна та ефективна команда, що функціонує за принципами Lean та крос-функціональності [16].

Команда розробки є крос-функціональною і включає необхідний мінімум фахівців для покриття повного циклу розробки та управління:

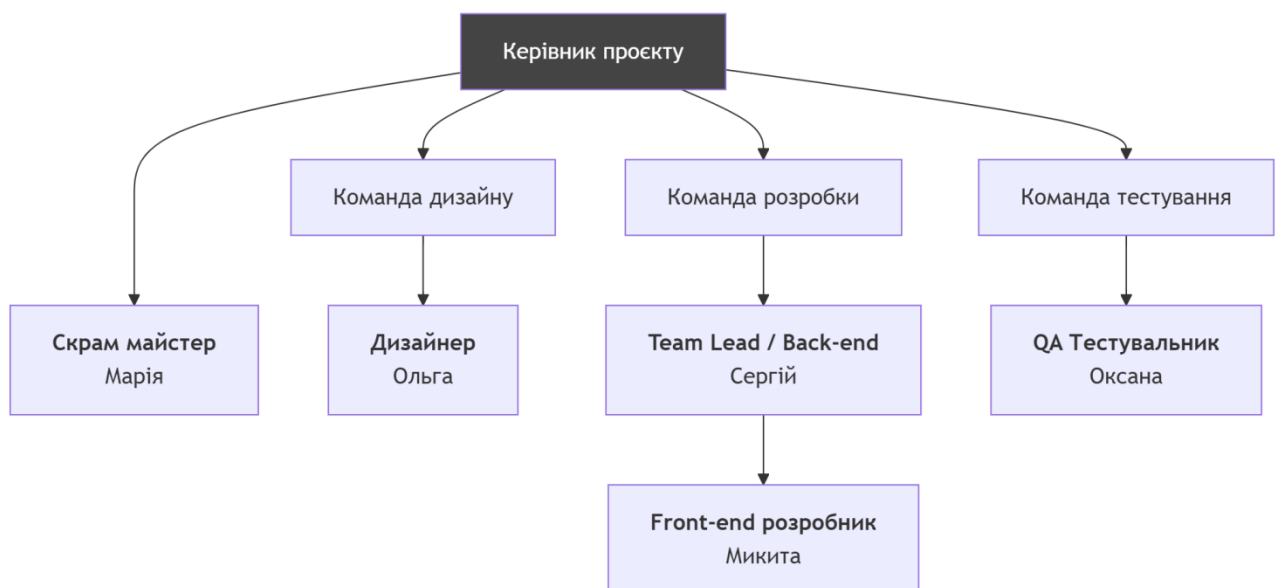


Рис. 3.2 – Проектна OBS проекту

- Менеджер проекту (PM/Product Owner): Відповідає за комунікацію з ГО "Місто майбутнього", управління беклогом, забезпечення пріоритетів та контроль термінів (Діаграма Ганта).

- Бекенд-розробник: Відповідає за розробку серверної логіки, API, інтеграцію з базою даних та платіжним шлюзом (FR005).
- Фронтенд-розробник: Відповідає за реалізацію інтерфейсу користувача згідно з дизайн-макетами (додатки Б, В, Г) та забезпечення адаптивності (NFR001).
- QA-інженер (Тестувальник): Відповідає за контроль якості, розробку тест-кейсів та верифікацію виконання всіх Функціональних Вимог (FR).
- Скрам-майстер: Проводить зустрічі та контролює дотримання методології; стежить, щоб задачі не «зависали» на дошці.

Для однозначного визначення ролей та обов'язків при виконанні ключових управлінських та технічних завдань використовується Матриця RACI (Responsible, Accountable, Consulted, Informed) наведено в таблиці 3.3 [23].

Таблиця 3.3

Матриця RACI

Завдання	Менеджер Проєкту	Бекенд-розробник	Фронтенд-розробник	QA-інженер
Розробка FR001 (Управління контентом)	A	R	C	I
Інтеграція FR005 (Платіжний шлюз)	A	R	C	I
Забезпечення NFR001 (Адаптивність)	C	C	R, A	I
Тестування FR003 (Автентифікація)	C	C	C	R, A
Затвердження фінальних дизайн-макетів	A	I	R	C

R (Responsible) – Виконавець: виконує роботу.

A (Accountable) – Відповідальний: приймає рішення та несе кінцеву відповідальність (лише один на завдання).

C (Consulted) – Консультує: надає інформацію для виконання завдання.

I (Informed) – Інформується: отримує інформацію про завершення завдання.

3.4. Оцінка трудомісткості та ресурсів.

Оцінка трудомісткості необхідна для визначення тривалості проєкту, обґрунтування графіку (Діаграма Ганта) та планування ресурсів.

Для визначення обсягу робіт використовується метод Експертної оцінки (Expert Judgment), який базується на досвіді аналогічних проєктів та деталізованій структурі WBS та Use Case. Оцінка виконана у людино-годинах (люд/год).

Нижче наведено таблицю 3.4 з оцінку трудомісткості для ключових, найбільш комплексних пакетів робіт, які безпосередньо впливають з функціональних вимог (FR).

Таблиця 3.4

Оцінка трудомісткості

Код WBS	Назва пакету робіт (Use Case)	Трудомісткість (люд/год)	Коментар
3.2.1	Розробка модуля управління контентом (FR001)	120	Створення/редагування проєктів, завантаження медіа, SEO-налаштування.
3.2.3	Розробка модуля пожертв (FR005)	90	Інтеграція зі стороннім платіжним шлюзом (висока складність).
3.3.1	Розробка публічного інтерфейсу (FR002, FR004, FR007)	160	Списки, фільтри, адаптивний дизайн.
3.4.1	Розробка адмін-панелі (FR003, FR008)	110	Безпечна автентифікація та інтерфейс для керування заявками волонтерів.
3.5.1	Розробка системи коментарів (FR016)	70	Інтерфейс коментування та механізм модерації.
3.6.1	Системне тестування та виправлення дефектів (Фаза 4)	180	Покриття 100% критичних FR та NFR.
Разом (загальна оцінка)	Основні роботи	730	

Загальна трудомісткість: Загальна оцінка обсягу основних робіт становить близько 730 людино-годин.

Тривалість проєкту: При роботі команди у складі 4 осіб (враховуючи паралельність процесів, згідно з Діаграмою Ганта) загальна тривалість проєкту становить 12 місяців (включаючи планування та тестування) [18].

Використання Scrumban: Застосування Scrumban дозволить ефективно управляти цим обсягом, забезпечуючи гнучкість пріоритетів та візуальний контроль потоку робіт, що є важливим для своєчасного виконання кожного пакету робіт.

3.5. Діаграма Ганта проєкту.

Діаграма Ганта – це графічне представлення плану проєкту, яке відображає тривалість, залежності та терміни виконання задач. Діаграма дозволить візуалізувати графік проєкту, відстежувати прогрес та управляти ресурсами (Рис.3.2, рис.3.3).

Для цього проєкту діаграма Ганта відіграє роль стратегічного планування на високому рівні, що відображає: загальний план, часові рамки для ключових milestones та великих модулів (Рис.3.4, рис.3.5, рис.3.6, рис.3.7, рис.3.8), а також потрібна для звітності перед керівництвом ГО «Місто Майбутнього». Діаграма була розроблена за допомогою ProjectLibre.

Використання цього інструменту дозволило чітко детермінувати критичний шлях проєкту, виділивши послідовність задач, від яких безпосередньо залежить дата релізу веб-платформи. Крім того, візуалізація часових розривів та накладень етапів розробки допомагає уникнути перевантаження команди, забезпечуючи рівномірний розподіл ресурсів на кожному спринті. Це створює прозоре середовище для моніторингу, де кожен учасник ГО «Місто Майбутнього» може оцінити поточний стан виконання робіт відносно запланованого графіку.

	🕒	Name	Duration	Start	Finish
1		📁 Ініціація Проекту	27 days?	5/1/25 8:00 AM	6/6/25 5:00 PM
2		📁 1.1 Визначення Проек	10 days?	5/1/25 8:00 AM	5/14/25 5:00 PM
3		1.1.1 Обрання тематики	3 days?	5/1/25 8:00 AM	5/5/25 5:00 PM
4	📅	1.1.2 Формулювання на	2 days?	5/6/25 8:00 AM	5/7/25 5:00 PM
5	📅	1.1.3 Опис проблеми/ці	5 days?	5/8/25 8:00 AM	5/14/25 5:00 PM
6	📅	📁 2 Формування Цілей П	17 days?	5/15/25 8:00 AM	6/6/25 5:00 PM
7		2.1.1 Визначення цілюв	3 days?	5/15/25 8:00 AM	5/19/25 5:00 PM
8	📅	2.1.2 Специфікація ціле	5 days?	5/20/25 8:00 AM	5/26/25 5:00 PM
9	📅	📁 2.1.3 Визначення лег	5 days?	5/26/25 8:00 AM	5/30/25 5:00 PM
10		2.1.3.1 Принципи піде	2 days?	5/26/25 8:00 AM	5/27/25 5:00 PM
11	📅	2.1.3.2 Надати чіткі в	3 days?	5/28/25 8:00 AM	5/30/25 5:00 PM
12	📅	2.1.4 Сформувати проф	5 days?	6/2/25 8:00 AM	6/6/25 5:00 PM
13	📅	📁 Планування Проекту	57 days?	6/9/25 8:00 AM	8/26/25 5:00 PM
14		📁 2.1 Визначення Вимог	33 days?	6/9/25 8:00 AM	7/23/25 5:00 PM
15		2.1.1 Формування бізне	5 days?	6/9/25 8:00 AM	6/13/25 5:00 PM
16	📅	📁 2.1.2 Декомпозиція (15 days?	6/16/25 8:00 AM	7/4/25 5:00 PM
17		2.1.2.1 FRB001: Плану	5 days?	6/16/25 8:00 AM	6/20/25 5:00 PM
18	📅	2.1.2.2 FRB002: Залуч	5 days?	6/23/25 8:00 AM	6/27/25 5:00 PM
19	📅	2.1.2.3 FRB003: Оцінк	5 days?	6/30/25 8:00 AM	7/4/25 5:00 PM
20	📅	📁 2.1.3 Декомпозиція (13 days?	7/7/25 8:00 AM	7/23/25 5:00 PM
21		2.1.3.1 NFRB001: Зруч	5 days?	7/7/25 8:00 AM	7/11/25 5:00 PM
22	📅	2.1.3.2 NFRB002: Про	3 days?	7/14/25 8:00 AM	7/16/25 5:00 PM
23	📅	2.1.3.3 NFRB003: Зруч	3 days?	7/17/25 8:00 AM	7/21/25 5:00 PM
24	📅	2.1.3.4 NFRB004: Ада	2 days?	7/22/25 8:00 AM	7/23/25 5:00 PM
25	📅	📁 2.2 Планування Agile	25 days?	7/23/25 8:00 AM	8/26/25 5:00 PM
26		2.2.1 Формування Бекл	7 days?	7/23/25 8:00 AM	7/31/25 5:00 PM
27	📅	2.2.2 Планування Спри	5 days?	8/1/25 8:00 AM	8/7/25 5:00 PM
28	📅	2.2.3 Визначення "Defin	3 days?	8/8/25 8:00 AM	8/12/25 5:00 PM
29	📅	2.2.4 Розробка User Sto	10 days?	8/13/25 8:00 AM	8/26/25 5:00 PM

Рис. 3.2 – Задачі проекту

	🕒	Name	Duration	Start	Finish
30	📅	📁 Розробка Системи	101.125 da...	8/27/25 8:00 AM	1/15/26 9:00 AM
31		3.1 Виконання Розробки	80 days?	8/27/25 8:00 AM	12/16/25 5:00 PM
32	📅	📁 3.2 Тестування та Заб	21 days?	12/17/25 9:00 AM	1/15/26 9:00 AM
33		3.2.1 Розробка Середов	5 days?	12/17/25 9:00 AM	12/24/25 9:00 AM
34	📅	3.2.2 Написання Тест-К	5 days?	12/26/25 9:00 AM	1/2/26 9:00 AM
35	📅	3.2.3 Виконання Тестув	8 days?	1/5/26 9:00 AM	1/15/26 9:00 AM
36	📅	📁 Впровадження	29 days?	1/16/26 9:00 AM	2/26/26 9:00 AM
37		4.1 Налаштування Серве	5 days?	1/16/26 9:00 AM	1/23/26 9:00 AM
38	📅	4.2 Розгортання Додатку	5 days?	1/26/26 9:00 AM	2/2/26 9:00 AM
39	📅	4.3 Налаштування DNS та	3 days?	2/3/26 9:00 AM	2/6/26 9:00 AM
40	📅	4.4 Передача та Навчанн	7 days?	2/9/26 9:00 AM	2/18/26 9:00 AM
41	📅	4.5 Запуск Системи	5 days?	2/19/26 9:00 AM	2/26/26 9:00 AM
42	📅	📁 Завершення Проекту	13 days?	2/27/26 9:00 AM	3/18/26 9:00 AM
43		5.1 Завершальні Дії Проек	7 days?	2/27/26 9:00 AM	3/10/26 9:00 AM
44	📅	5.2 Оцінка Проекту	5 days?	3/11/26 9:00 AM	3/18/26 9:00 AM
45	📅	📁 Підтримка та Обслугов	24.875 days?	3/19/26 9:00 AM	4/22/26 5:00 PM
46		6.1 Моніторинг Системи	7 days?	3/19/26 9:00 AM	3/30/26 9:00 AM
47	📅	6.2 Усунення Виявлених І	7 days?	3/31/26 8:00 AM	4/8/26 5:00 PM
48	📅	6.3 Оновлення та Модерн	5 days?	4/9/26 8:00 AM	4/15/26 5:00 PM
49	📅	6.4 Технічна Підтримка К	5 days?	4/16/26 8:00 AM	4/22/26 5:00 PM

Рис. 3.3 – Задачі проекту (продовження)

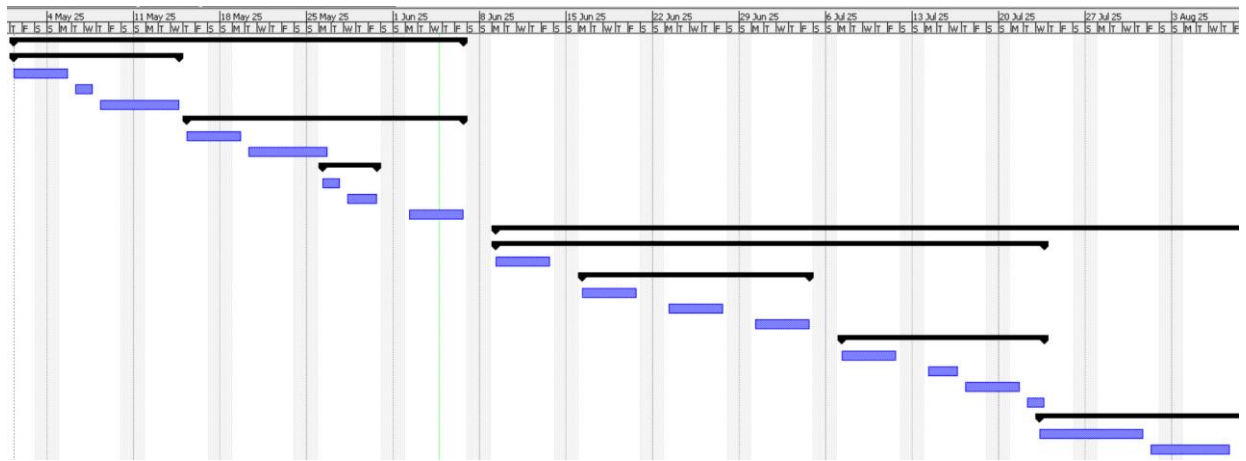


Рис. 3.4 – Діаграма Ганта в період з 01.05.2025 по 03.08.2025



Рис. 3.5 – Діаграма Ганта в період з 03.08.2025 по 09.11.2025

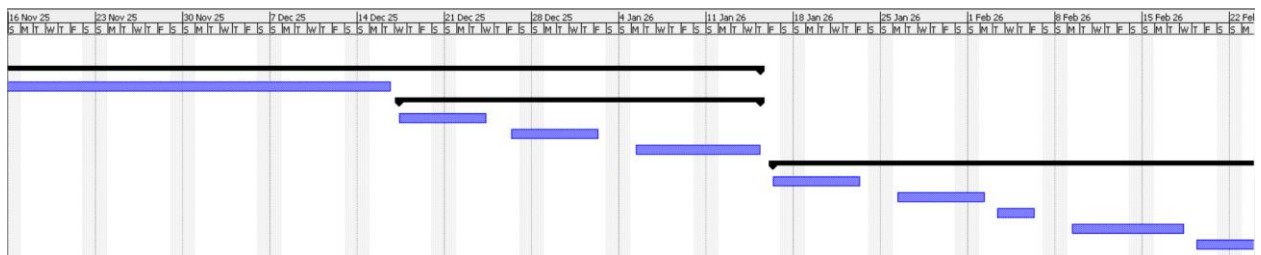


Рис. 3.6 – Діаграма Ганта в період з 16.11.2025 по 22.02.2026

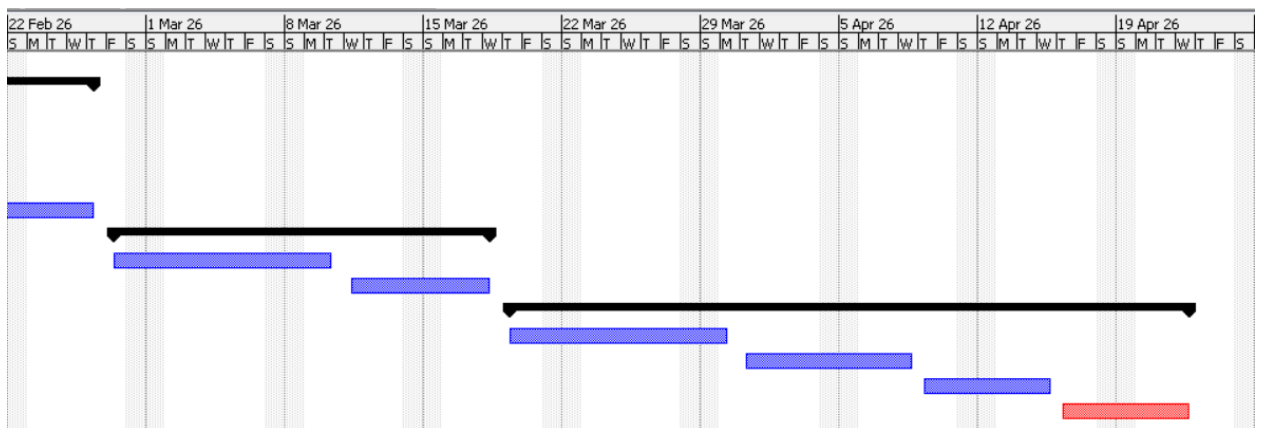


Рис. 3.7 – Діаграма Ганта в період з 22.02.2026 по 22.04.2026

	④	Name	RBS	Type	Initials	Group	Max. Units	Standard Rate
1		Valeriia	Project manager	Work	V	Managment	100%	USD 25.00/hour
2		Mykyta	Frontend developer	Work	M	Development	100%	USD 20.00/hour
3		Serhii	Backend developer	Work	S	Development	100%	USD 20.00/hour
4		Olha	Designer	Work	O	Designer	100%	USD 20.00/hour
5		Oksana	QA	Work	O	Development	100%	USD 15.00/hour
6		Maria	Scrum-master	Work	M		100%	USD 15.00/hour

Рис. 3.8 – Команда проєкту

3.6. Trello-дошка проєкту.

Дошка Trello є ключовим інструментом для операційного управління та контролю потоку роботи у Scrumban-процесі. Вона слугує візуалізацією всього процесу, від ідеї до завершення, забезпечуючи прозорість та злагодженість у роботі команди [18]. Trello-дошка дає змогу команді:

- Візуалізувати Scrumban-процес: кожна колонка на дошці відповідає певному статусу або етапу роботи. Типовий Scrumban-процес може включати такі колонки, як: загальний перелік завдань, «Task» (завдання, треба зробити), «In progress» (в процесі), «Test» (тестування) та «Done» (зроблено).
- Операційне управління: картки (завдання) переміщуються по дошці, що відображає поточний статус кожного завдання в реальному часі. Це дозволяє керівнику проєкту та членам команди швидко оцінити загальний стан проєкту та виявити потенційні «вузькі місця».
- Контроль та прозорість: дошка чітко показує, хто над чим працює прямо зараз (завдяки призначенню виконавців на картках). Це підвищує підзвітність і дозволяє швидко ідентифікувати, чи не перевантажений хтось із членів команди.

Використання Trello в контексті Scrumban гарантує дотримання ключових принципів Kanban:

- Обмеження роботи в процесі: Trello дозволяє встановлювати WIP-ліміти на колонки «In progress» та «Test». Це означає, що команда може працювати лише над обмеженою кількістю завдань одночасно.

Коли ліміт досягнуто, нові завдання не можуть бути переміщені в цю колонку, доки не буде завершено інше завдання. Це змушує команду фокусуватися на завершенні поточної роботи перед початком нової, що значно підвищує ефективність та швидкість потоку.

- Управління потоком: візуалізація дозволяє команді постійно аналізувати потік роботи та шукати способи його оптимізації. Завдання мають рухатися до колонки «Done» з постійною, передбачуваною швидкістю.
- Постійне покращення: регулярний аналіз дошки (наприклад, під час щоденних стендапів або Scrumban-оглядів) допомагає виявити, які етапи уповільнюють роботу, і відповідно скоригувати процес.

Для великих проєктів доцільно використовувати кілька спеціалізованих дощок Trello, об'єднаних в один робочий простір, щоб забезпечити сфокусованість окремих команд і, водночас, прозорість для загального управління.

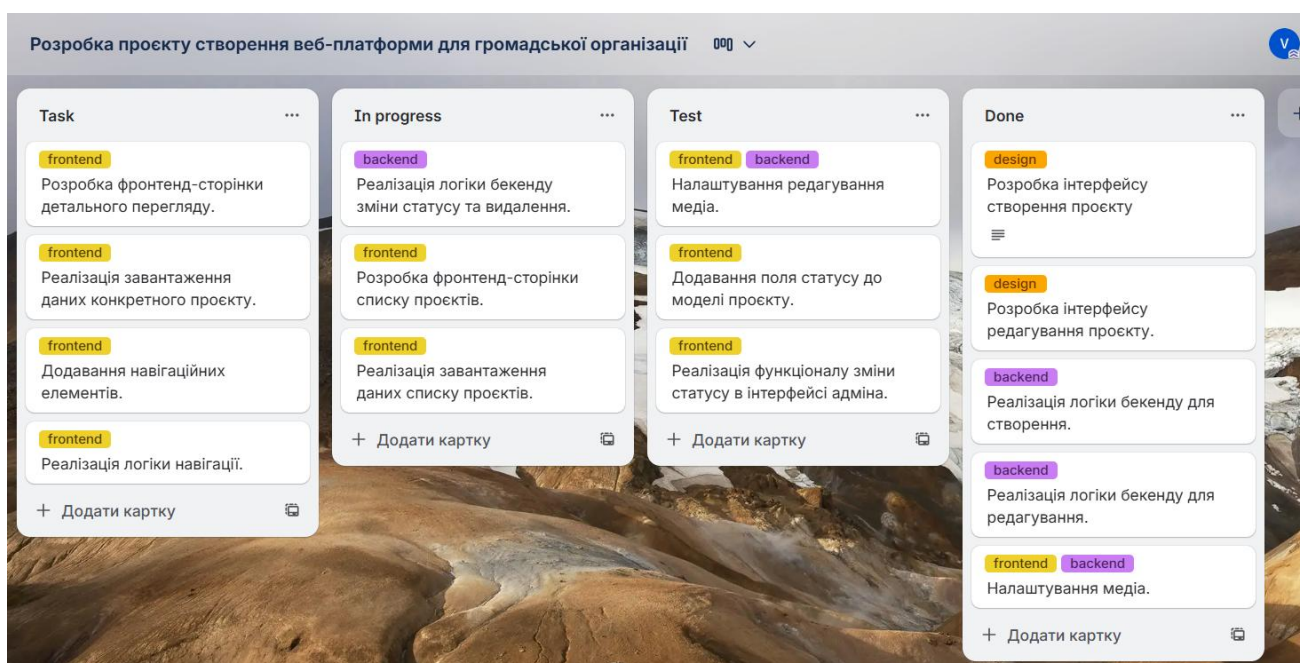


Рис. 3.8 – Загальна дошка проєкту

Ця дошка слугує, як головна дорожня карта, де відображаються великі ініціативи (Рис. 3.8). Вона дає загальне бачення прогресу проєкту для всіх зацікавлених сторін.

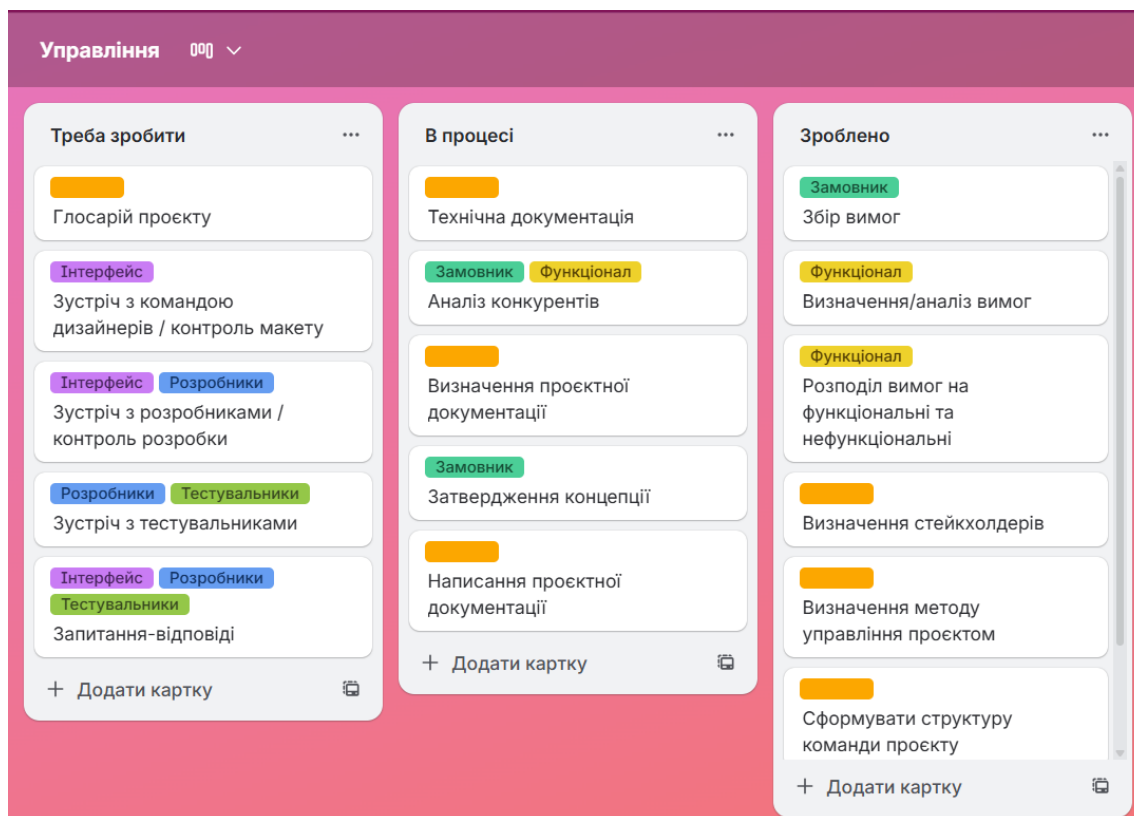


Рис. 3.9 – Дошка з управління проєкту

Призначена для менеджера проєкту та власника продукту. Містить пріоритезацію вимог, деталізацію завдань та планування релізів. Це місце, де відбувається «мікс» Scrumban – планування, подібне до Scrum, та управління потоком, подібне до Kanban.

Окрема дошка для команди дизайну (UX/UI), яка відображає їхній специфічний потік: концепція, макети, прототипування, фінальний Дизайн (Рис. 3.10). Завдання звідси передаються на дошку розробки.

Відокремлення дизайнерського потоку дозволяє команді UX/UI працювати з випередженням розробки, зазвичай на 1–2 спринти. Це створює стабільний потік підготовлених завдань для розробників, мінімізуючи простой через відсутність макетів або неясність інтерфейсних рішень.

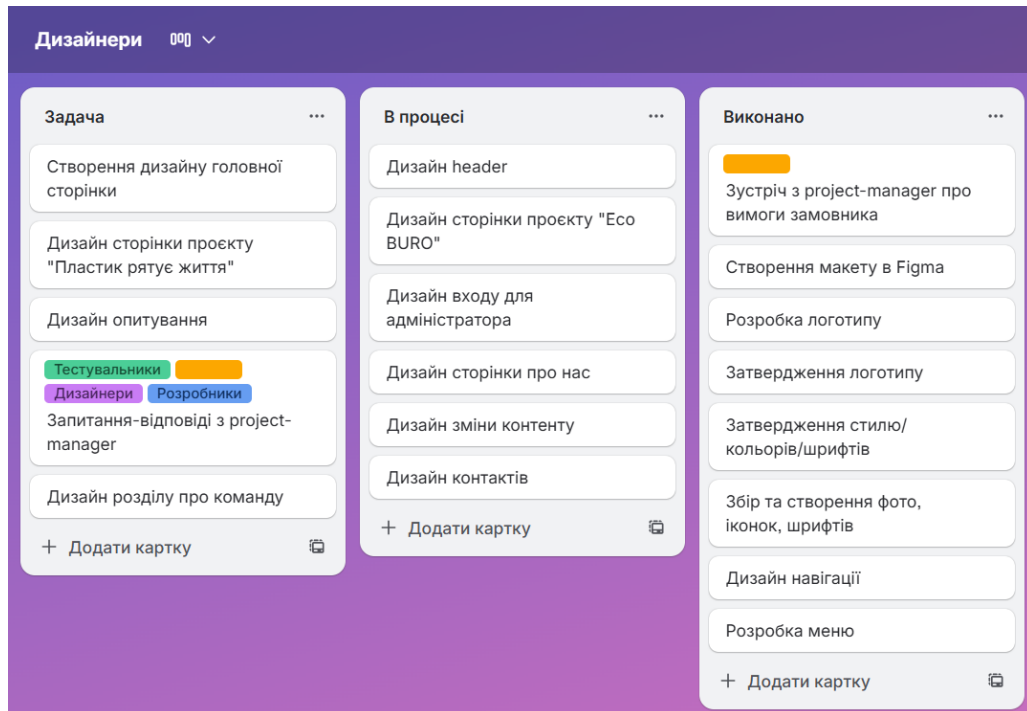


Рис. 3.10 – Дошка дизайнерів проєкту

Командна дошка, представлена на Рис. 3.11, є основним інструментом візуалізації потоку створення цінності. На відміну від стратегічних дошок високого рівня, вона сфокусована на щоденних операціях розробників та забезпечує максимальну прозорість виробничих процесів. Класичний потік, реалізований на дошці, дозволяє відстежувати динаміку кожної робочої одиниці:

- «Задача» (To Do / Backlog): Пул пріоритезованих завдань, готових до взяття в роботу. Наявність задачі в цій колонці означає, що вона пройшла етап уточнення вимог (Definition of Ready).
- «В процесі» (In Progress): Активна фаза розробки. Тут зосереджена основна активність команди.
- «Тестування / Перевірка» (Testing / Quality Assurance): Критичний етап валідації. Задача вважається завершеною розробником, але потребує верифікації на відповідність критеріям прийняття (Acceptance Criteria).

- «Виконано» (Done): Фінальна точка, що означає повну готовність функціоналу до релізу (Definition of Done).

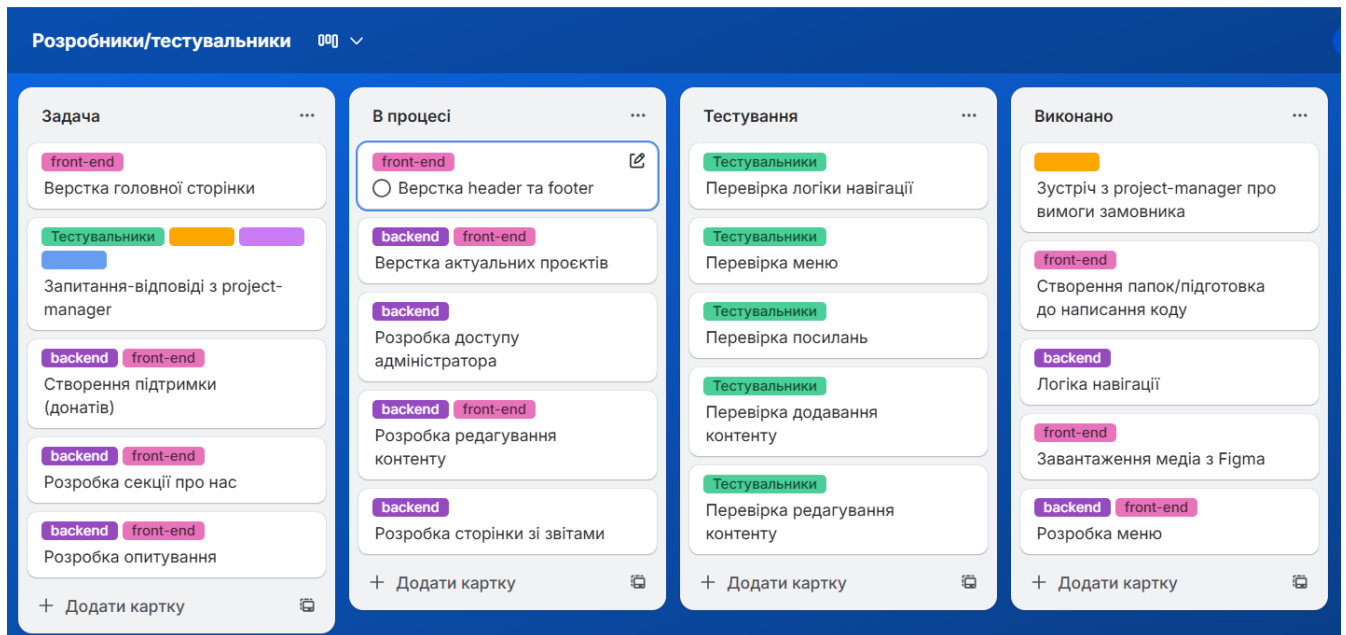


Рис. 3.11 – Дошка з розробки та тестування проєкту

Окрім візуалізації стадій, командна дошка виконує функцію інструменту виявлення вузьких місць. Завдяки обмеженню кількості одночасних завдань у колонці «В процесі» (WIP-ліміти), команда уникає розпорошення ресурсів і фокусується на завершенні розпочатого функціоналу. Це дозволяє Скрам-майстру оперативно ідентифікувати блоки, що заважають просуванню задач, а Керівнику проєкту — отримувати об'єктивні дані про реальну швидкість розробки. Такий підхід гарантує, що кожна задача на дошці проходить через системний контроль якості, мінімізуючи накопичення технічного боргу ще до моменту інтеграції в основний реліз.

Використання Trello як єдиного простору для управління задачами дозволяє інтегрувати стратегічне планування з щоденною активністю команди. Завдяки системі взаємопов'язаних дошок, проєкт отримує необхідну прозорість: відстеження кожної задачі — від ідеї до фінального тестування — стає автоматизованим процесом, що мінімізує ризики втрати інформації.

РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ ТА УСПІШНОСТІ ПРОЄКТУ.

4.1. Технічна архітектура та реалізація платформи.

Програмне забезпечення веб-платформи для громадської організації «Місто Майбутнього» буде розроблено з використанням модульної архітектури (також відомої як багат шарова або мікросервісна архітектура у більш складних випадках). Цей підхід є фундаментальним для забезпечення гнучкості, масштабованості, надійності та зручності підтримки системи в довгостроковій перспективі.

4.1.1. Модульна архітектура та технологічний стек.

Модульна архітектура дозволяє розділити систему на незалежні компоненти, кожен з яких відповідає за певний функціонал. Це спрощує розробку, тестування та подальше оновлення окремих частин без впливу на всю платформу. Нижче наведено таблицю 4.1, яка містить детальний опис ключових компонентів програмного забезпечення, їхнє призначення та обрані технологічні рішення [18].

Таблиця 4.1

Компоненти програмного забезпечення

Компонент	Призначення	Технології
Фронтенд (Front-end)	Відповідає за відображення інформації користувачеві та забезпечення взаємодії з системою.	Сучасний JavaScript-фреймворк: Vue.js (для створення динамічного та зручного інтерфейсу).
Бекенд (Back-end)	Обробка запитів, реалізація бізнес-логіки та безпека.	Серверна платформа: Node.js/Express (для швидкої розробки та масштабованості).
База даних	Зберігає всі дані системи (проєкти, події, волонтери, звіти).	PostgreSQL (забезпечує надійність та складні запити)
API	Забезпечує стандартизовану взаємодію між фронтендом та бекендом, а також можливість інтеграції з іншими системами (наприклад, платіжними шлюзами, соціальними мережами).	RESTful API

Використання розподіленого стека (Vue.js + Node.js/Express + PostgreSQL) з чітким RESTful API приносить такі ключові переваги:

- Розділення відповідальності: фронтенд і бекенд можуть розроблятися і тестуватися незалежно, що прискорює загальний процес розробки та спрощує виявлення помилок.
- Масштабованість: бекенд (Node.js) і база даних (PostgreSQL) можуть бути масштабовані окремо один від одного, дозволяючи системі ефективно справлятися зі зростаючим навантаженням (збільшенням кількості користувачів та даних).
- Гнучкість та майбутні інтеграції: завдяки стандартизованому API, у майбутньому можна легко розробити мобільний додаток (iOS/Android), використовуючи той самий Бекенд, або інтегрувати платформу з новими сервісами.
- Зручність обслуговування: чітка структура та сучасні технології спрощують підтримку, оновлення та додавання нового функціоналу для команди розробників.

4.1.2. Алгоритмічне забезпечення.

Розробка алгоритмів (Рис. 4.1.) ґрунтується на функціональних вимогах (Use Case). Ключові алгоритми включатимуть:

1. Алгоритм автентифікації: Забезпечення безпечного входу (FR003) та управління правами доступу (ролі Адміністратора, Менеджера).
2. Алгоритм обробки донатів: Послідовність дій від вибору суми до підтвердження транзакції через платіжний шлюз (FR005).
3. Алгоритм пошуку та фільтрації: Ефективна обробка запитів користувачів за ключовими словами та критеріями (FR004, FR007).
4. Алгоритм зміни статусу проєкту: Автоматична перевірка та оновлення статусу проєкту після настання кінцевої дати (FR011).

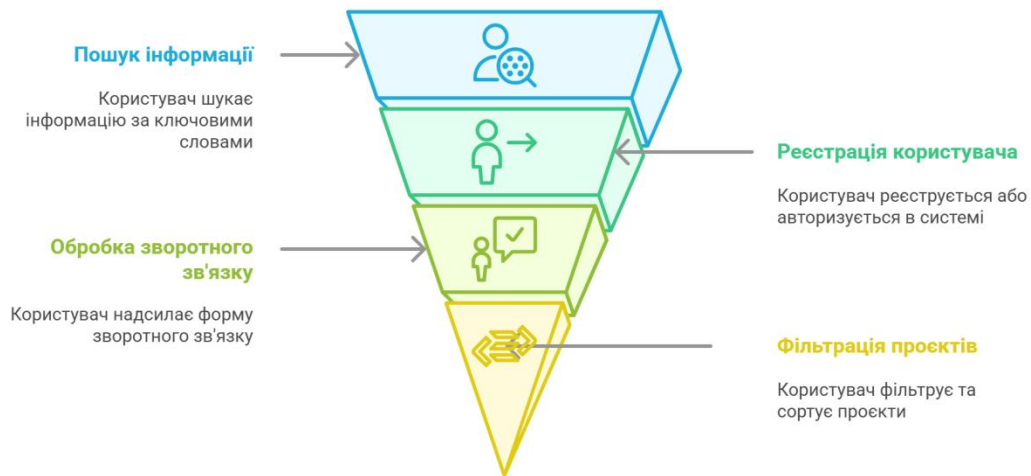


Рис. 4.1 – Алгоритми ПЗ

Ефективний веб-сайт (Рис. 4.2) поєднує функціональну надійність та інтуїтивно зрозумілий дизайн. Інтерфейси (UI/UX) проекту мають бути розроблені з акцентом на доступність (WCAG 2.1) та прозорість:

1. Панель управління: Інтерфейс для Адміністраторів/Менеджерів (FR001, FR008), що забезпечує інтуїтивне керування контентом та обробку заявок.
2. Публічний інтерфейс: Чистий, адаптивний дизайн, що забезпечує легкий перегляд інформації про проекти, новини та фінансові звіти (FR002, FR010, FR006).

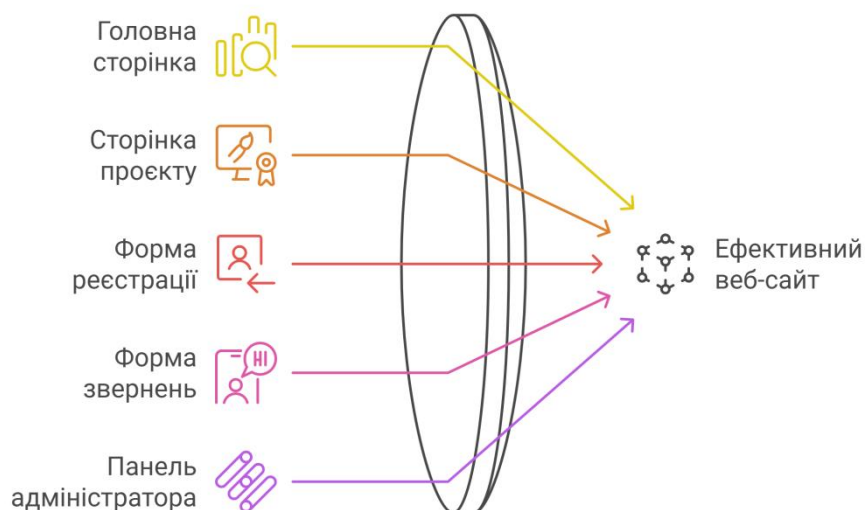


Рис. 4.2 – Візуалізація складових ефективного веб-сайту

4.2. Управління ризиками та критерії успішності.

Управління ризиками є критично важливим для забезпечення досягнення критеріїв успіху та критеріїв прийняття результату, а також, невід'ємною частиною гібридної методології Scrumban, оскільки дозволяє мінімізувати ймовірність негативних подій та їх вплив на цілі проєкту [19].

Для кожного виявленого ризику було визначено рівень ймовірності та ступінь впливу, що дозволило ранжувати їх за пріоритетністю. Матриця ризиків, представлена нижче в таблиці 4.2, слугує дорожньою картою для превентивного реагування, гарантуючи, що функціональні можливості платформи будуть реалізовані з дотриманням усіх вимог безпеки.

Таблиця 4.2

Ключові ризики проєкту

ІД	Ризик	Категорія	Ймовірність	Вплив	Стратегія реагування
P01	Ризик зміни пріоритетів ГО	Управління/Бізнес	Середня	Високий	Запобігання/Уникнення: Регулярна синхронізація з керівництвом ГО, чітке визначення та затвердження MVP для забезпечення фокусу. Підтвердження фінансування на весь термін проєкту.
P02	Технологічний ризик: Складнощі з інтеграцією ГІС-систем	Технічні	Середня	Середній	Зменшення/Запобігання: Проведення пілотного проєкту для інтеграції картографічного API на ранніх етапах. Використання стандартизованих, добре задокументованих сервісів (Google Maps API, OpenStreetMap).
P03	Ресурсний ризик: Відтік волонтерів або ключових розробників	Ресурсні	Середня	Високий	Зменшення/Перенесення: Розробка детальної технічної документації в Notion для спрощення передачі знань. Забезпечення регулярних винагород/визнання для волонтерів.
P04	Низький трафік та залученість	Експлуатація	Середня	Середній	Запобігання: Активна SEO-оптимізація (FR019), інтеграція соціальних мереж (FR012). Запуск маркетингової кампанії після Віхи 5 для досягнення KPI: +30% трафіку.

Продовження Таблиці 4.2

P05	Помилки при прийомі донатів (FR005)	Технічні /Безпека	Низька	Високий	Запобігання: Ретельне тестування модуля пожертв. Використання лише сертифікованих платіжних шлюзів та протоколу HTTPS.
P06	Провал приймального тестування (UAT)	Якість/Тестування	Середня	Високий	Запобігання: Залучення кінцевих користувачів до тестування на ранніх етапах. Чітке узгодження Критеріїв прийняття (наприклад, відповідність FR001-FR020).
P07	Проблеми з доступністю (WCAG)	Юридичні/Технічні	Середня	Середній	Запобігання: Включення WCAG 2.1 (AA) до Definition of Done кожного завдання. Використання автоматизованих інструментів перевірки доступності.

Критерії успіху демонструють, чи досягнуто головної мети проєкту — цифрової трансформації та масштабування діяльності ГО «Місто Майбутнього».

1. Підвищення рівня поінформованості про діяльність організації: KPI: +30% трафіку на сайт за 3 місяці після запуску.
2. Збільшення кількості залучених волонтерів та учасників: KPI: 50+ нових реєстрацій волонтерів через платформу за півроку (успішність FR008).
3. Технічний успіх (надійність): KPI: Безперебійна робота платформи протягом перших 6 місяців (uptime > 99%).

Критерії прийняття результату використовуються для підтвердження того, що розробка завершена відповідно до технічного завдання та готова до передачі в експлуатацію.

1. Виконання функціональних вимог: Успішно реалізовано всі функціональні вимоги (FR001 – FR020), згідно з фінальним технічним завданням.
2. Проходження приймального тестування: Успішно пройдено тестування кінцевими користувачами.
3. Надання документації: Передано повну технічну документацію та інструкції з експлуатації (Notion).

4. Відповідність стандартам: Платформа відповідає нефункціональним вимогам, включаючи WCAG 2.1.

4.3. Фінансове планування та кошторис проєкту.

Для забезпечення успішної реалізації проєкту, незалежно від його масштабу та складності, критично важливим є встановлення чітких фінансових рамок та ефективне управління ресурсами [20].

Цей розділ присвячений детальному фінансовому плануванню, яке є основою для контролю витрат та оцінки економічної доцільності проєкту. Тут буде розглянуто процес складання детального кошторису, який охоплює всі очікувані витрати — від оплати праці команди до придбання необхідного програмного забезпечення та обладнання.

Мета фінансового планування полягає у мінімізації ризиків перевищення бюджету та забезпеченні прозорості використання інвестицій, гарантуючи, що проєкт буде завершено в межах встановлених фінансових обмежень.

4.3.1. Кошторис проєкту.

В даному кошторисі прямі та непрямі витрати рахуються окремо. Розділення витрат на прямі та непрямі є критично важливим для розуміння точки беззбитковості проєкту та реальної вартості розробки. Такий підхід дозволяє менеджменту бачити, яка частина бюджету безпосередньо створює цінність продукту, а яка — забезпечує життєдіяльність проєкту та супутню інфраструктуру [21]. Після чого, наведений загальний, зведений, кошторис проєкту:

1. Прямі витрати (оплата праці команди), наведені в таблиці 4.3, становлять найбільшу частку бюджету і безпосередньо корелюють обсягом робіт, визначеним у WBS. Ці витрати ґрунтуються на загальній оцінці

трудомісткості проекту (980 люд/год для основних робіт) та додатковий час на управління та дизайн.

Таблиця 4.3

Прямі витрати проекту

Категорія витрат	Склад команди	Оцінка часу (люд/год)	Рейтова ставка (USD/год)	Сума витрат (USD)	Обґрунтування
1.1 Розробка (Backend/Frontend)	2 Розробники	450	20	9 000	Реалізація всіх функціональних вимог (FR001-FR020), включаючи інтеграції та UI.
1.2 Тестування (QA)	1 QA	180	15	2 700	Модульне, інтеграційне та системне тестування (Фаза 4).
1.3 Управління та аналіз	1 РМ/Аналітик	100	25	2 500	Керування WBS, комунікації, управління ризиками (P01), контроль якості.
1.4 Дизайн (UI/UX)	1 Дизайнер	50	20	1 000	Розробка макетів та затвердження (Віха 2).
1.5 Скрам-майстер	Скрам-майстер	100	15	1 500	Фасилітація Scrumban-процесів, усунення перешкод (blockers), моніторинг WIP-лімітів та швидкості команди.
1.6 Резерв на ризики	Уся команда	100	20	2 000	Покриття непередбачених завдань або виправлення критичних помилок (P06).
РАЗОМ ПРЯМІ ВИТРАТИ		980		18 700 USD	

2. Непрямі витрати (інфраструктура та ліцензії), наведено в таблиці 4.4. Ці витрати забезпечують відповідність нефункціональним вимогам (NFR003, NFR002, NFR020). Вибір ліцензійних рішень та гнучкої інфраструктури дозволяє забезпечити масштабованість веб-платформи у довгостроковій перспективі. Це гарантує, що при збільшенні обсягів даних для ШІ-моделей або

зростанні трафіку, система зберігатиме свою цілісність та швидкість відгуку, мінімізуючи технічний борг та витрати на подальшу підтримку коду.

Таблиця 4.4

Непрямі витрати

Категорія витрат	Призначення	Вартість (USD/міс)	Тривалість	Сума витрат (USD)	Обґрунтування / Посилання на FR/NFR
2.1 Хостинг (VPS/Cloud)	Розгортання ПЗ (Node.js/PostgreS QL)	50	12 міс.	600	Забезпечення масштабованості та надійності (NFR012, NFR003).
2.2 Домен та SSL-сертифікат	Річна реєстрація домену та SSL	20	1 рік	20	Безпека трафіку (NFR002).
2.3 API-ключі (Google Maps)	Платне використання API для відображення карти	10	12 міс.	120	Виконання FR018 (Карта локацій проєктів).
2.4 Сервіс розсилки (Email)	Інтеграція з Mailchimp/SendGrid	20	12 міс.	240	Виконання FR009 (Модуль підписки).
2.5 Платіжний шлюз	Комісія (умовно)	50	12 міс.	600	Обслуговування інтеграції (FR005).
РАЗОМ НЕПРЯМІ ВИТРАТИ				1 580 USD	

3. Зведений кошторис проєкту, наведений в таблиці 4.5. Загальна сума кошторису відображає всі витрати на розробку, тестування, управління та річну підтримку інфраструктури, необхідної для успішного запуску та експлуатації веб платформи "Місто Майбутнього".

Таблиця 4.5

Зведений кошторис проєкту

Категорія	Сума витрат (USD)	% від загальної суми
1. Прямі Витрати (оплата праці)	18 700	91.6%
2. Непрямі Витрати (інфраструктура)	1 580	8.4%
ЗАГАЛЬНИЙ КОШТОРИС ПРОЄКТУ	20 280 USD	100%

Загальний кошторис проєкту складає 20 280 USD. Це фінансова оцінка необхідна для реалізації всіх 20 функціональних вимог (FR), забезпечення відповідності критичним нефункціональним вимогам (NFR) та мінімізації ключових ризиків. Найбільшу частку витрат (91.6%) складають прямі витрати на оплату праці кваліфікованої команди розробників, що є типовим для ІТ-проєктів.

4.3.2. Моделі фінансування та забезпечення сталості.

Враховуючи, що веб-платформа розробляється для громадської організації «Місто Майбутнього», основною метою якої є не отримання прибутку, а виконання соціальної місії, фінансування проєкту та його подальша підтримка базуватимуться на моделі сталого розвитку. Це поєднання прямих пожертв, грантового фінансування та створення додаткової вартості, що опосередковано підтримує діяльність ГО [22].

Фінансове забезпечення платформи буде здійснюватися за рахунок таких ключових джерел:

1. Пряма підтримка:

- Механізм: вбудований модуль підтримки (FR005) дозволяє користувачам швидко та безпечно робити внески на конкретні проєкти або на загальну діяльність організації.
- Призначення: ці кошти покриватимуть переважно прямі витрати на реалізацію проєктів ГО. Частина донатів (наприклад, 10-15%) може бути цільово спрямована на покриття непрямих витрат платформи (хостинг, API-ключі, комісії платіжних шлюзів).

2. Грантове фінансування:

- Механізм: платформа, як інструмент цифрової трансформації та підвищення прозорості, є потужним аргументом для залучення коштів від міжнародних фондів та місцевих

грантодавців, які підтримують цифрові ініціативи та розвиток громад.

- Призначення: гранти будуть залучатися як для фінансування розробки (що покриває основну частину кошторису), так і для подальшого розвитку функціоналу платформи (наприклад, інтеграція нових модулів).

3. Корпоративне соціальне партнерство:

- Механізм: залучення соціально відповідального бізнесу до фінансування конкретних проєктів, розміщених на платформі, або прямої підтримки технічного функціонування сайту.
- Призначення: підприємства можуть спонсорувати хостинг, розробку нових функцій або надавати послуги pro bono (наприклад, SEO-оптимізацію, консультації).

Окрім прямого фінансування, платформа створить додаткові можливості для генерації доходу та підтримки сталого розвитку:

- Продаж звітів та аналітики для партнерів (Data as a Service): Платформа збиратиме унікальні дані про потреби громади, ефективність проєктів, та активність волонтерів. Ці анонімізовані та агреговані дані можуть бути продані дослідницьким інститутам, аналітичним центрам або партнерам, зацікавленим у соціальних дослідженнях.
- Організація платних онлайн-заходів: Використання платформи для організації вебінарів, тренінгів або майстер-класів з соціальної активності, волонтерства чи екологічних питань. Це може бути невеликий, але стабільний потік доходу.
- Членські внески для партнерів: Введення опційної плати за розширений доступ для партнерських організацій або шкіл, які бажають розміщувати власні міні-проєкти на піддомені ГО або отримувати розширену аналітику залученості своїх співробітників/студентів у волонтерську діяльність.

Критерієм фінансового успіху (успішності FR005 та KPI: > 99%) буде забезпечення фінансової самокупності непрямих витрат платформи протягом першого року експлуатації, що наведено в таблиці 4.6.

Таблиця 4.6

Прогнозовані джерела доходу

Джерело доходу	Цільове призначення	Очікуваний % покриття непрямих витрат
Цільові Донати	Хостинг, API, Комісія платіж. шлюзу	40%
Гранти	Оплата праці розробників/техніків	50%
Платні заходи	Резервний фонд на оновлення та ризику	10%

Для громадської організації критично важливо досягти стану фінансової стійкості, за якого підтримка цифрової платформи не залежить від нестабільних благодійних внесків. Розрахункова сума у 1 580 USD на рік охоплює витрати на хостинг, безпеку даних, ліцензії на ПЗ та технічну підтримку.

Для покриття операційних витрат передбачено впровадження каналів отримання доходу, що не суперечать неприбутковому статусу організації:

- Залучення бізнес-секторів, які готові спонсорувати окремі розділи платформи або надавати сервіси в обмін на логічне розміщення їхнього бренду як «технологічного партнера».
- Прозорий розподіл ресурсів, за якого весь обсяг благодійних внесків спрямовується виключно на потреби бенефіціарів, дозволяє суттєво підвищити показник соціальної ефективності проєкту.

Така фінансова архітектура створює фундамент для довгострокового масштабування ініціатив ГО «Місто Майбутнього». Чітке розмежування операційного бюджету та благодійних фондів мінімізує стратегічний ризик залежності від фінансування, про який йшлося раніше. Це дозволяє організації фокусуватися на стратегічному розвитку, а не на «виживанні» цифрової інфраструктури.

ВИСНОВКИ

У кваліфікаційній роботі проведено комплексне дослідження діяльності ініціативи «Місто Майбутнього» та розроблено проектне рішення для її цифрової трансформації. Відповідно до поставлених завдань, було отримано наступні результати:

1. Проаналізовано основні напрямки діяльності ініціативи: У ході роботи детально вивчено ключові вектори «Міста Майбутнього», зокрема урбаністичну модернізацію, культурні проекти та екологічні ініціативи. Це дозволило сформулювати 20 функціональних вимог до платформи, які відображають реальні потреби організації у сферах мобільності, безпеки та збереження історичної спадщини.
2. Визначено ринкову позицію та конкурентні переваги: Встановлено, що статус соціально орієнтованого проекту та роль культурного хабу є ключовими перевагами ініціативи. Для посилення цієї позиції обґрунтовано впровадження гібридної методології Scrumban, яка дозволяє організації гнучко адаптуватися до змін, зберігаючи високу швидкість реалізації соціальних проектів.
3. Здійснено оцінку впливу на міське середовище Одеси: Доведено, що цифровізація ініціативи безпосередньо корелює з покращенням міської інфраструктури. Побудовані архітектурні моделі та логічна схема бази даних (проекти, волонтери, донати) забезпечують надійний фундамент для масштабування урбаністичних змін. Очікуваний соціальний ефект підтверджено встановленими KPI: збільшення активності громади на 30% протягом перших 6 місяців.
4. Проаналізовано механізми залучення громадськості та партнерів: Розроблено модель взаємодії, що базується на Use Case сценаріях для різних типів акторів (волонтери, меценати, представники влади). Окрему увагу приділено інноваційним інструментам — ШІ (NLP) для

інтелектуального пошуку та персоналізації контенту, що значно підвищує рівень залученості громадян до ідей сталого розвитку.

5. Розроблено стратегічні рекомендації та план реалізації: Сформовано повний пакет управлінської та фінансової документації для посилення співпраці з партнерами:
 - WBS та Діаграма Ганта: визначено етапи реалізації проєкту до квітня 2026 року.
 - Кошторис: обґрунтовано бюджет у розмірі 20 280 USD, що включає прямі витрати та інвестиції в інфраструктуру (1 580 USD/рік).
 - Ризик-менеджмент: визначено стратегії реагування на загрози, що гарантує технологічну довговічність системи.

Практична цінність для стейкхолдерів:

1. Для донорів та партнерів: він слугує гарантом прозорості та професійного підходу до управління ресурсами.
2. Для розробників: є чітким технічним завданням, що мінімізує кількість ітерацій та помилок при реалізації.
3. Для громади Одеси: є запорукою створення зручного та безпечного інструменту впливу на розвиток рідного міста.

Результатом роботи є цілісний пакет проєктно-технічної документації, який забезпечує перехід ГО «Місто Майбутнього» на новий рівень цифрової взаємодії. Практичне впровадження розробленої веб-платформи дозволить оптимізувати процеси урбаністичного розвитку Одеси, підвищити прозорість діяльності організації та створити ефективний інструмент для міжнародного обміну досвідом у сфері соціальних інновацій.

Реалізація даного проєкту дозволить перетворити соціальний капітал ініціативи «Місто Майбутнього» у вимірюваний цифровий результат. Платформа стане каталізатором для нових партнерств, об'єднуючи зусилля влади, бізнесу та активних громадян навколо ідеї сталого та сучасного розвитку Одеси.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Морозов В.В., Зюсюн В.І., Єгорченков О.В. Курсова робота [Електронний ресурс]: методичні вказівки до виконання курсової роботи для студентів освітньо-професійної програми «Управління проектами» спеціальності «Комп'ютерні науки» для заочної форми навчання, К.: КНУ імені Тараса Шевченка, 2025. – 46 с.
2. PMBOK Guide: Project Management Institute. [Електронний ресурс]: *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) (7th ed.)*. Project Management Institute. 2022.
3. Russell, S. J., & Norvig, P. [Електронний ресурс]: *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson, 2020.
4. World Wide Web Consortium (W3C) Official Documentation. [Електронний ресурс]: *Web Content Accessibility Guidelines (WCAG) 2.1*.
5. Переваги використання методології управління проектами [Електронний ресурс]. – <https://worksection.com/ua/blog/benefits-of-project-management-methodology.html>
6. Agile Manifesto. Manifesto for Agile Software Development [Електронний ресурс]. – <https://agilemanifesto.org/>
7. Scrum Guide. The Scrum Guide [Електронний ресурс]. – <https://scrumguides.org/>
8. Все про Scrum. [Електронний ресурс]. – <https://www.scrum.org/resources/what-scrum-module>
9. Інформація та порівняння методологій управління проектами. Водоспадна модель VS Agile. [Електронний ресурс]. – <https://www.geeksforgeeks.org/software-engineering/agile-vs-waterfall/>
10. Якісні та кількісні методи прогнозування в стратегічному аналізі [Електронний ресурс]. – <https://studfile.net/preview/5265215/page:7/>
11. Костенко, Ю.А. Методи та інструменти управління проектами. – Харків: ХНУ

12. Kalbach, J. Mapping Experiences: A Complete Guide to Customer Alignment Through Journeys, Blueprints, and Diagrams. – O'Reilly Media
13. Nielsen, J. Usability Engineering. – San Francisco: Morgan Kaufmann
14. Як ШІ змінює управління проєктами [Електронний ресурс]. – <https://tra.cy/uk/blog/how-ai-is-changing-project-management-in-2025-5-key-trends>
15. Project Management within start-ups: Literary review and case studies in Stockholm, Sweden. [Електронний ресурс]. – <https://www.semanticscholar.org/paper/Project-Management-within-start-ups-%3A-Literary-and-Adnot/670ad1d419cfc319d4b78147882dd073243e3f65>
16. 5 фаз життєвого циклу управління проєктами [Електронний ресурс]. – <https://www.ranktracker.com/uk/blog/the-5-phases-of-theproject-management-lifecycle-in-2023/>
17. Методологія управління проєктами. [Електронний ресурс]. – <https://nt.ua/blog/methodology-of-project-management>
18. Vue.js. [Електронний ресурс]. – <https://vuejs.org/>
19. PostgreSQL. [Електронний ресурс]. – <https://uk.wikipedia.org/wiki/PostgreSQL>
20. RACI Chart. [Електронний ресурс]. – <https://www.atlassian.com/work-management/project-management/raci-chart>
21. Redis. [Електронний ресурс]. – <https://uk.wikipedia.org/wiki/Redis>
22. Статистика зарплат. [Електронний ресурс]. – <https://www.work.ua/stat/>
23. Аналіз критичного шляху [Електронний ресурс]. – https://stud.com.ua/21062/menedzhment/analiz_kritichnogo_shlyahu

ДОДАТКИ

Додаток А

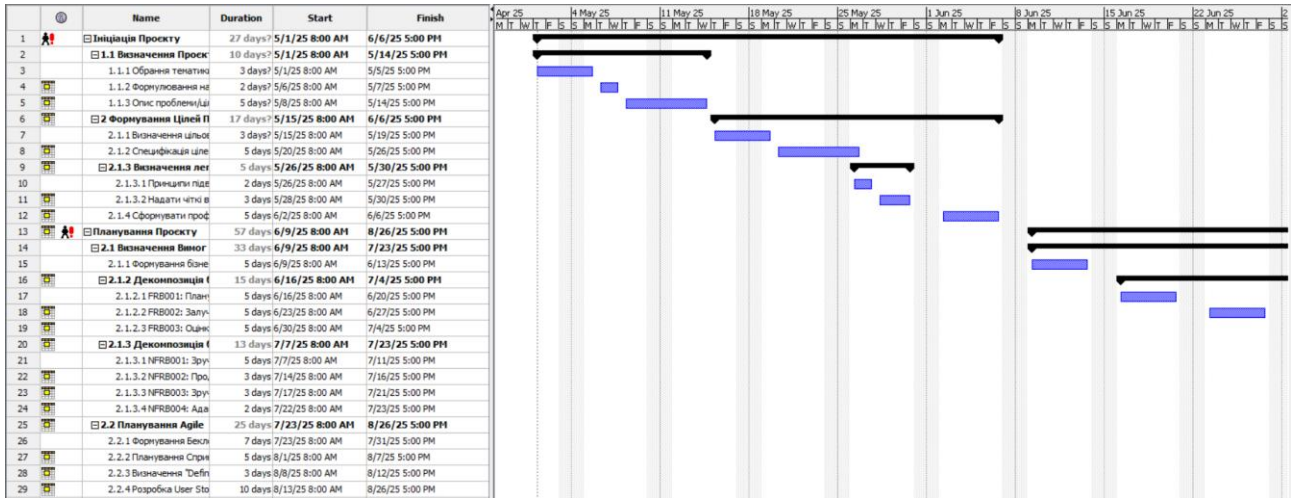


Рис.А.1 – Діаграма Ганта проєкту

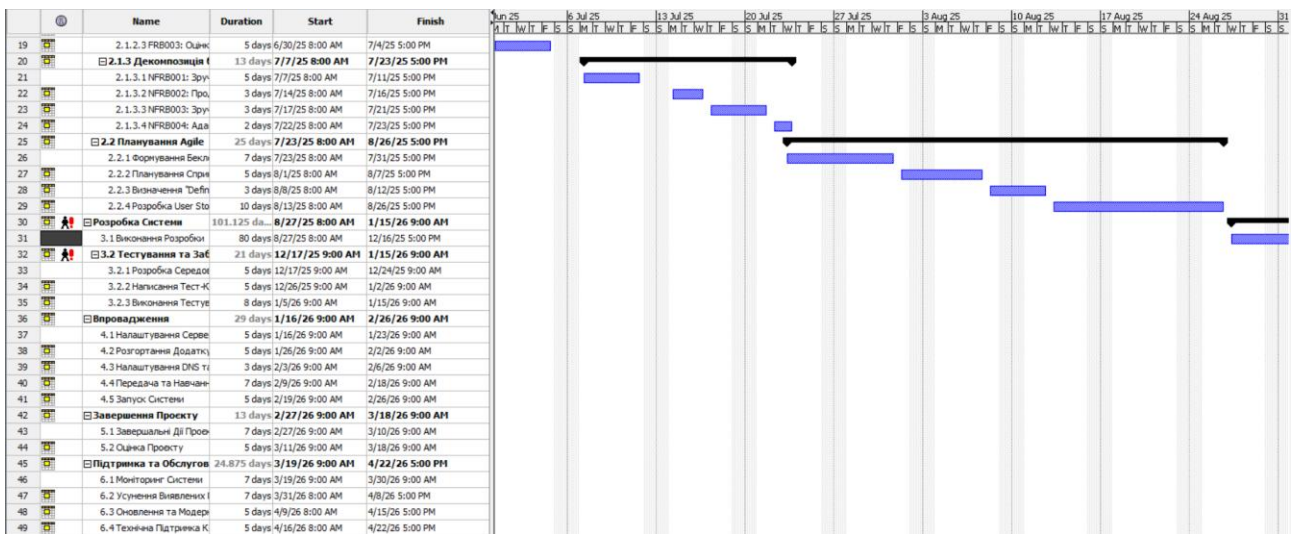


Рис.А.2 – Діаграма Ганта проєкту

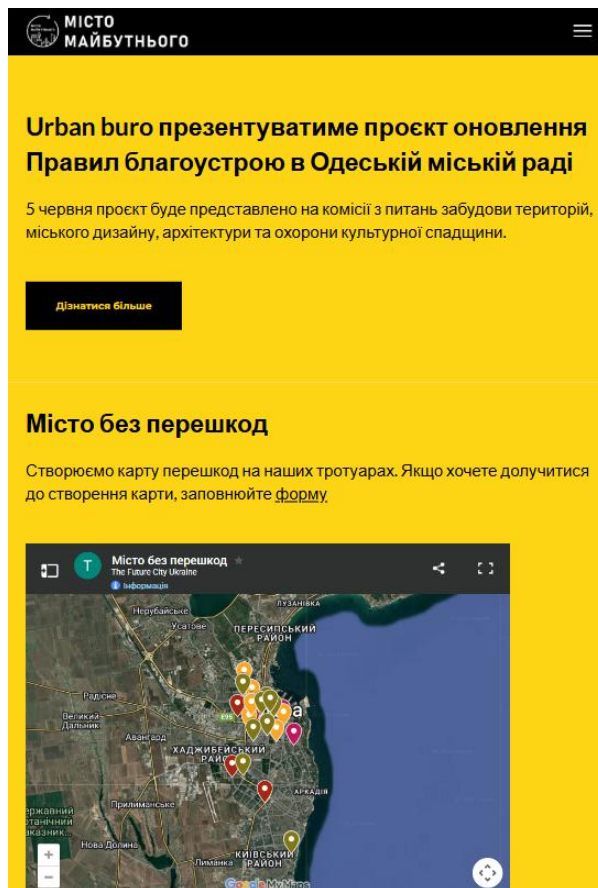


Рис.Б.1 – Дизайн-макет головної сторінки сайту

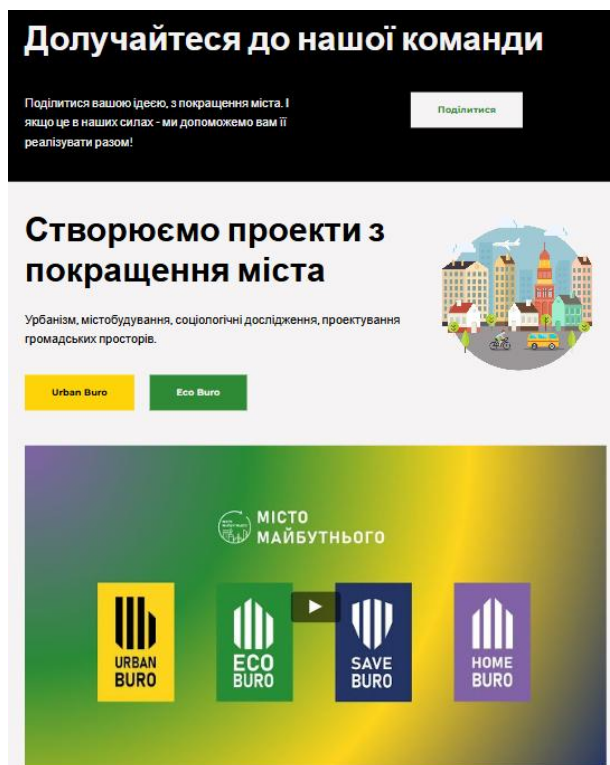


Рис.Б.2 –Дизайн-макет головної сторінки сайту (продовження)



Рис.Б.3 – Дизайн-макет головної сторінки сайту (продовження)

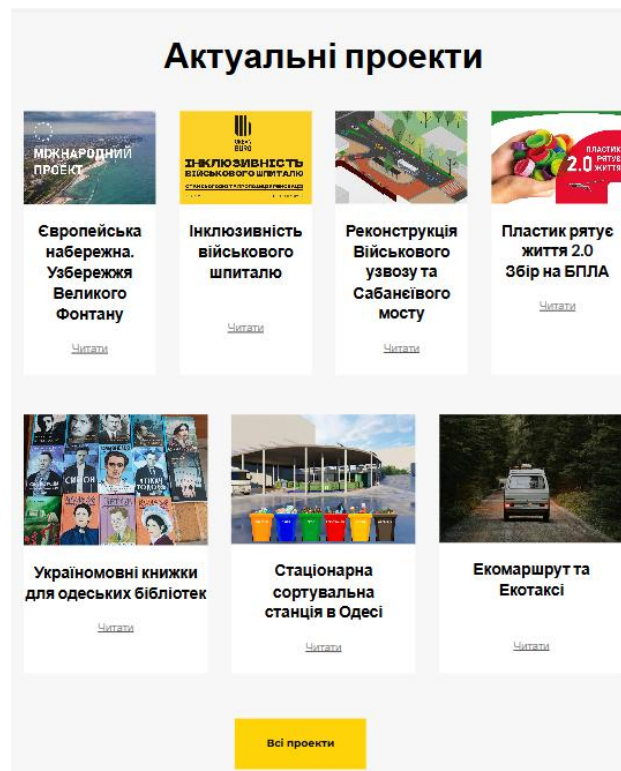


Рис.Б.4 – Дизайн-макет головної сторінки сайту (продовження)

Досягнення

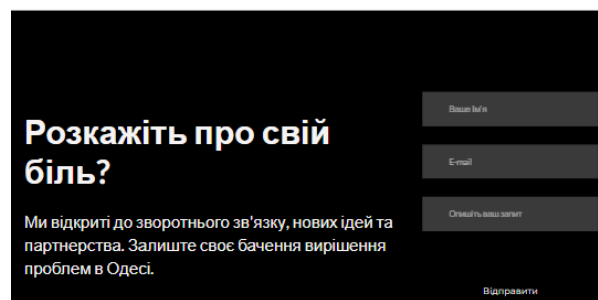
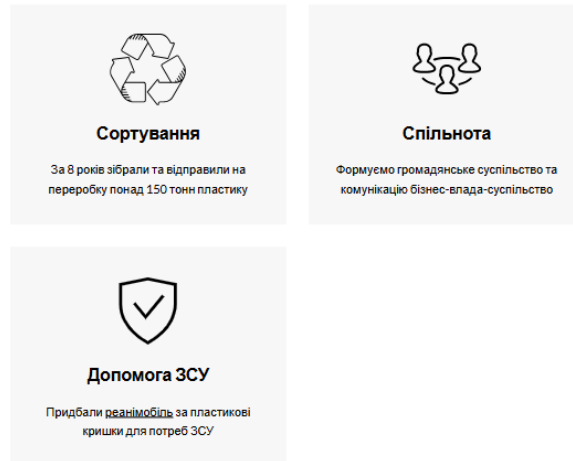


Рис.Б.5 – Дизайн-макет головної сторінки сайту (продовження)

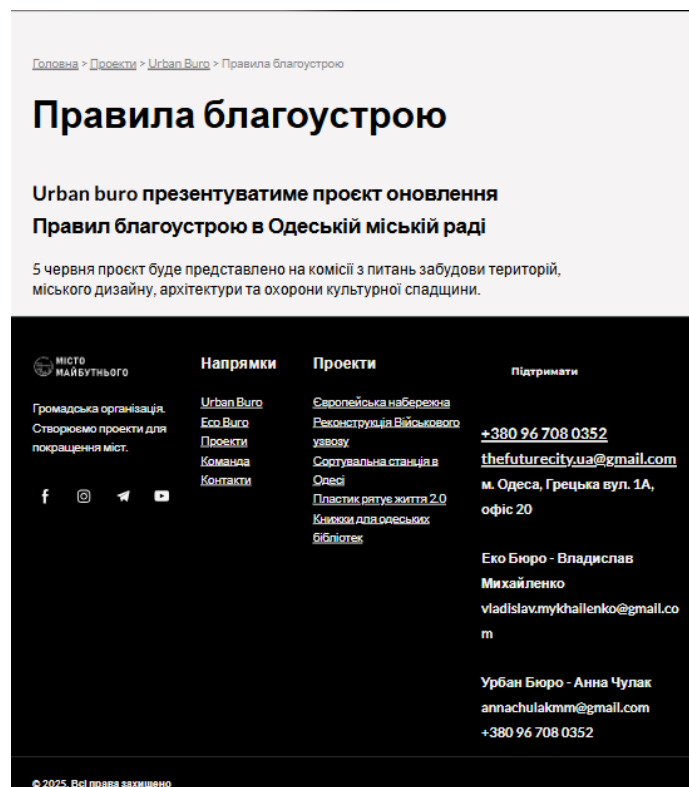


Рис.Б.6 – Дизайн-макет головної сторінки сайту (продовження)



Рис.В.1 – Дизайн-макет сторінки проєкту «Пластик рятує життя»

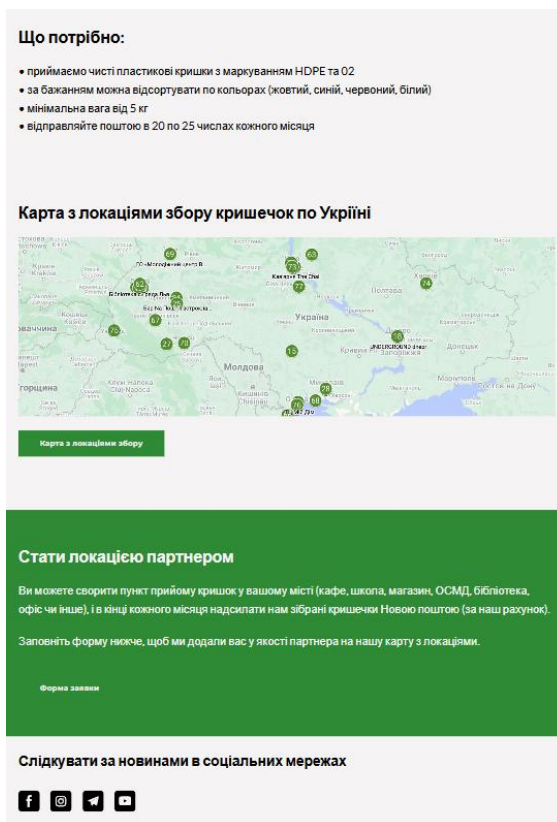



Рис.В.2 – Дизайн-макет сторінки проєкту «Пластик рятує життя»
(продовження)

 **МІСТО
МАЙБУТНЬОГО**

Підтримати

Щоб зробити внесок використайте реквізити нижче.

*Ми є некомерційною та неприбутковою організацією. Всі кошти спрямовані на підтримку ГО "Місто Майбутнього" будуть витрачені на забезпечення адміністративної діяльності організації та підтримку актуальних проектів.

Банківський переказ за реквізитами

Переказ у гривні Переказ у USD Переказ у EUR

Код ЄДРПОУ _____ 45050451

Рахунок в ПриватБанк (IBAN) _____
UA383052990000026001044921932

Отримувач _____ ГО "Місто Майбутнього Україна"

Призначення платежу _____ Благодійна допомога

Підписатись на нас в соціальних мережах





   

Рис.Г.1 – Дизайн-макет сторінки про підтримку організації