

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-наукова програма «Управління
проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

“Дослідження методів створення проекту розробки програмного забезпечення
для оцінки продуктивності працівників”

Студента 2-го курсу групи УП-22

Науковий керівник:

Дениса ТАНАСІЙЧУКА

Тетяна ЛАТИШЕВА

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: “До захисту в Екзаменаційній комісії”)

Завідувач кафедри
технологій управління

(підпис)

Віктор МОРОЗОВ

(прізвище, ініціали)

(дата)

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

Кафедра технологій
управління Освітній рівень
Магістр
Спеціальність 122 Комп'ютерні науки
Освітня програма Управління
проектами

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Морозов В.В.

“ _____ ” _____ 20__ року

**З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Танасійчук Денис Олександрович

Група: УП-22

1. Тема кваліфікаційної роботи: “Дослідження методів створення проєкту розробки програмного забезпечення для оцінки продуктивності працівників”.
Затверджена наказом №6 від 06.11.2023.

2. Строк подання студентом готової роботи - “13” травня 2024р.

3. Цільова установка та вихідні дані до роботи: дослідження характеристики об'єкту, планування та виконання проєкту, що включає календарне планування, зміст, бюджет, ресурси.

4. Зміст роботи: аналіз предметної галузі, проведення маркетингових досліджень, побудова дерева проблем, дерева цілей, аналіз проєктних альтернатив, опис мети та цілі продукту, розробка життєвого циклу проєкту, побудова ієрархічної структури робіт проєкту, побудова організаційної структури компанії, формування складу команди проєкту, планування ресурсів проєкту, календарне планування проєкту, планування ресурсів проєкту, планування вартості проєкту, планування управління, ризиками, закупівлями, зацікавленими сторонами, моніторинг проєкту.

5. Перелік графічного матеріалу (слайдів): титульна сторінка, мета

дипломної роботи, життєвий цикл проєкту, дерево проблем, дерево цілей, ієрархічна структура робіт, управління ризиками, календарне планування, використання бюджету, висновки.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	План виконання роботи
1	Вивчення інформації з предмету.	09.11.2023 - 28.11.2023
2	Збір і вивчення матеріалів дослідження.	29.11.2023 - 18.12.2023
3	Складання плану кваліфікаційної роботи.	19.12.2023 - 07.01.2024
4	Ознайомлення наукового керівника з планом кваліфікаційної роботи магістра. Внесення змін.	08.01.2024 - 27.01.2024
5	Підготовка розділу 1.	28.01.2024 - 16.02.2024
6	Підготовка розділу 2.	17.02.2024 - 07.03.2024
7	Підготовка розділу 3.	08.03.2024 - 27.03.2024
8	Оформлення кваліфікаційної роботи.	28.03.2024 - 16.04.2024
9	Передача роботи на рецензування.	20.04.2024
10	Передача роботи науковому керівнику.	20.04.2024
11	Захист роботи.	20.04.2024

Дата видачі завдання “09” листопада 2023р.

Керівник роботи кандидат техн. наук, асистент, Латишева Т.В.

(підпис)

Завдання прийняв до виконання
студент групи УП-22 Танасійчук Д.О.

(підпис)

ЗМІСТ

АНОТАЦІЯ	6
ТАБЛИЦЯ СКОРОЧЕНЬ ТА ПОЯСНЕНЬ	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИКИ ОБ'ЄКТУ УПРАВЛІННЯ	15
1.1 Огляд об'єкту управління	15
1.1.1 Загальний опис об'єкта управління	15
1.1.2 Історичний контекст та еволюція об'єкта управління.....	15
1.1.3 Сучасне положення та значення в системі управління.....	17
1.2 Проведення аналізу предметної галузі	17
1.2.1 Визначення предметної галузі	17
1.2.2 Аналіз поточного стану предметної галузі	18
1.2.3 Аналіз існуючих методів оцінки	19
1.2.4 Аналіз конкурентів	22
1.2.5 SWOT-аналіз.....	25
1.3 Проведення маркетингових досліджень.....	26
1.3.1 STEP-аналіз.....	26
1.3.2 Аналіз внутрішнього середовища	27
1.3.3 Аналіз зовнішнього середовища	29
1.4 Інвестиційні дослідження.....	30
1.5 Дерево проблем, дерево цілей та логіко структурна схема.....	34
1.6 Мета, цілі та продукт проєкту	38
1.7 Дослідження методів розробки ПЗ.....	39
РОЗДІЛ 2 ПЛАНУВАННЯ ВИКОНАННЯ ПРОЄКТУ.....	41
2.1 Розробка життєвого циклу проєкту. Побудова ієрархічної структури робіт проєкту.	41
2.2 Склад команди проєкту та розподіл відповідальності.....	44
2.3 Планування робіт проєкту.....	48
2.4 Планування ресурсів проєкту	53

2.5 Планування вартості проєкту	57
РОЗДІЛ 3. ДОСЛІДЖЕННЯ ПРОЄКТУ НА СТАДІЇ ВИКОНАННЯ ТА	
УПРАВЛІННЯ КОМПОНЕНТАМИ.....	64
3.1 Управління якістю	64
3.2 Управління ризиками.....	77
3.3 Управління зацікавленими сторонами.....	84
3.4. Управління закупівлями	90
3.5 Моніторинг проєкту.....	95
ВИСНОВКИ.....	98
СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	99
Додаток А.....	101
Додаток Б.....	102

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

“Дослідження методів створення проєкту розробки програмного забезпечення для оцінки продуктивності працівників”

Студент: Танасійчук Денис Олександрович.

Науковий керівник: Латишева Тетяна Володимирівна.

Рік захисту – 2024.

Тема даної кваліфікаційної роботи - “Дослідження методів створення проєкту розробки програмного забезпечення для оцінки продуктивності працівників”. Мета даної наукової роботи полягає у вивченні та аналізі методів створення програмного забезпечення для ефективно оцінки продуктивності працівників. Це дослідження спрямоване на розробку і вдосконалення системи, яка автоматизує процес оцінки працівників у компаніях, роблячи його більш об'єктивним та ефективним.

Предметом дослідження є процес аналізу існуючих технологій та підходів до розробки програмного забезпечення для оцінки продуктивності працівників. Це включає в себе вивчення різних програмних рішень на ринку, ідентифікацію ключових функцій, які повинно виконувати таке програмне забезпечення, і оцінку технологічних підходів до його розробки.

Об'єктом дослідження є процеси та методології управління проєктами розробки програмного забезпечення, зосереджені на оцінці продуктивності працівників в інформаційних системах. Робота включає аналіз сучасних методик та інструментів, придатних для оптимізації робочих процесів з метою підвищення ефективності виконання завдань і взаємодії в командах .

Під час аналізу ринку вивчалась різні програмні продукти для управління людськими ресурсами, їх функціональність, зручність використання та вартість. Окрім цього, ідентифікувались ключові функції, які повинно мати програмне забезпечення для оцінки працівників, і проводиться оцінка технологічних підходів до його розробки. В даній роботі описані результати

аналізу ринку подібних технологій, можливість виходу альтернативного програмного забезпечення на ринок, побудовано дерево проблем і рішень. Розроблення життєвий цикл проєкту, побудовано ієрархічну структуру робіт, побудовано організаційну структуру, а також описано склад команди, що працюватиме над проєктом. Було проведено календарне планування, планування ресурсів та вартості проєкту. Здійснено дослідження проєкту на стадії виконання, проведено управління якістю, описані можливі ризики та запропоновані запобіжні методи та надано опис зацікавлених сторін проєкту.

Ціль проєкту - розробка та успішне впровадження функціонального та безпечного додатку

Наукова новизна роботи полягає у комплексному дослідженні технічних, безпекових, інтерфейсних та маркетингових аспектів розробки додатку для

Новизною також є впровадження та оптимізація

Кваліфікаційна робота складається з анотації, вступу, основної частини, яка включає три розділи, висновків, списку використаних джерел та додатків.

В першому розділі було проведено дослідження та аналіз характеристик об'єкту дослідження. Також було розглянуто предметну область та проаналізовано існуючі методи дослідження.

В другому розділі було проведено планування проєкту, що включало в себе створення життєвого циклу проєкту, створення команди проєкту, проведено планування робіт та вартості проєкту.

В третьому розділі наводиться опис дослідження проєкту на стадії його виконання. Висвітлені питання управління якістю, ризиками, зацікавленими сторонами та закупівляями.

За результатами роботи були зроблені наступні **висновки**:

1. В рамках дипломної роботи було розроблено комплексний проєкт програмного забезпечення.

2. В процесі розробки було створено і детально описано життєвий цикл

проєкту.

3. Було ретельно розплановано бюджет проєкту.

4. Проєкт включає обґрунтований аналіз ризиків, який допомагає ідентифікувати потенційні проблеми та запропонувати методи їх мінімізації або вирішення.

5. В дипломній роботі описано процес управління зацікавленими сторонами, що є ключовим для успішного впровадження проєкту та забезпечення його прийняття користувачами.

Робота містить 97 сторінок з урахуванням додатків, 27 рисунків, 23 таблиці та 2 формули.

Ключові слова: *оцінка продуктивності працівників, розробка програмного забезпечення, управління проєктами, SWOT-аналіз, життєвий цикл проєкту, аналіз ризиків, кількісні методи дослідження*

ТАБЛИЦЯ СКОРОЧЕНЬ ТА ПОЯСНЕНЬ

Скорочення	Пояснення
ПЗ	Програмне забезпечення
ОС	Операційна система
ШІ	Штучний інтелект
ЛСС	Логіко-структурна схема
ГБ	Гігабайт
ТЗ	Технічне завдання
IPC (WBS)	Ієрархічна структура робіт (Work Breakdown Structure)
ЗС	Зацікавлена сторона
LP	Libre project
OBS	Organizational Breakdown Structure (організаційна структура)

ВСТУП

У сучасному світі, де темпи технологічного розвитку невинно зростають, ефективне управління продуктивністю працівників стає ключовим чинником успіху для багатьох організацій. Від того, наскільки ефективно компанія може оцінювати та підвищувати продуктивність своїх співробітників, залежить її здатність досягати стратегічних цілей, оптимізувати процеси та підтримувати конкурентоспроможність на ринку. У цьому контексті розробка програмного забезпечення для оцінки продуктивності працівників набуває особливої актуальності, оскільки воно може надати організаціям потужний інструмент для збору, аналізу та інтерпретації даних про роботу співробітників.

Вступ до нашої роботи присвячений дослідженню методів створення проєкту розробки такого програмного забезпечення. Ми розглянемо різні підходи до оцінки продуктивності працівників, аналізуємо сучасні технології та методології розробки програмного забезпечення, які можуть бути застосовані для створення ефективних систем управління продуктивністю. Особливу увагу буде приділено етичним аспектам використання такого програмного забезпечення, оскільки важливо збалансувати потреби організації у контролі та аналізі продуктивності з правами та приватністю працівників.

Актуальність даної роботи зумовлена рядом ключових чинників. По-перше, у сучасному бізнес-середовищі набуває значення технологічний прогрес, що спонукає організації оновлювати свої методи і інструменти для збереження конкурентоспроможності. По-друге, здатність ефективно аналізувати та оптимізувати роботу співробітників за допомогою автоматизованих систем оцінки є важливим аспектом для підвищення продуктивності робочих процесів. Також, використання даних для обґрунтування управлінських рішень стає невід'ємною частиною ефективного управління. Крім того, сучасні робочі умови, такі як дистанційна робота, вимагають нових підходів до моніторингу діяльності працівників. Нарешті, врахування етичних і юридичних аспектів в розробці таких систем є критично

важливим для забезпечення приватності та дотримання нормативних вимог. Таким чином, дипломна робота відповідає на сучасні виклики і вносить значний вклад у розвиток інструментів управління персоналом у корпоративному середовищі.

Метою нашої роботи є не лише вивчення теоретичних аспектів розробки програмного забезпечення для оцінки продуктивності, а й аналіз реальних кейсів та розробка рекомендацій для компаній, які планують впроваджувати подібні системи. Через комплексний підхід до вивчення цієї теми ми прагнемо внести вагомий внесок у розвиток інструментів для підвищення продуктивності працівників, а також у формування бестпрактик для їх впровадження та використання в корпоративному середовищі.

Об'єктом дослідження виступає процес оцінки продуктивності працівників в організаціях за допомогою програмного забезпечення. Це включає в себе загальну структуру та динаміку процесів оцінювання, відстеження та аналізу робочої діяльності працівників, а також механізми використання отриманої інформації для управління персоналом та оптимізації бізнес-процесів.

Предметом дослідження є методи та технології розробки програмного забезпечення, спрямовані на оцінку продуктивності працівників. В рамках предмета дослідження розглядаються алгоритми обробки даних, інтерфейси користувача, системи збору та аналізу інформації, а також методи захисту даних та забезпечення приватності працівників.

Основні завдання роботи:

– Аналіз потреб та вимог користувачів: провести детальний аналіз потреб користувачів і вимог до програмного забезпечення для оцінки продуктивності працівників, враховуючи сучасні тенденції та технології.

– Дослідження життєвого циклу проєкту: описати і аналізувати всі етапи життєвого циклу розробки проєкту, від ідеї до реалізації і підтримки, з акцентом на ключові етапи та їхнє значення.

– Планування бюджету проєкту: розробити детальний фінансовий план,

включаючи оцінку витрат на ресурси, матеріали, та інші аспекти, необхідні для виконання проєкту.

- Аналіз ризиків: виконати аналіз потенційних ризиків проєкту та розробити стратегії їх мінімізації або управління.

- Управління зацікавленими сторонами: описати стратегії та методи управління зацікавленими сторонами, що сприятимуть успішному впровадженню та прийняттю проєкту.

- Оцінка і вибір методологій розробки: аналізувати існуючі методології розробки програмного забезпечення та обрати найбільш ефективні для даного проєкту, зокрема обґрунтувати вибір методу Agile.

- Оцінка результатів проєкту: оцінити, як розроблене програмне забезпечення може підвищити продуктивність працівників і оптимізувати робочі процеси, та який вплив це матиме на загальну ефективність організацій.

На першому етапі роботи був проведений глибокий аналіз існуючих публікацій, статей та наукових досліджень щодо методів оцінки продуктивності працівників, а також існуючих технологій і платформ для розробки відповідного програмного забезпечення. Це дозволило виявити прогалини в наявних підходах та технологіях.

Наступний етап включав в себе аналіз і вибір технологій. На основі літературного огляду було обрано найбільш відповідні технології та методології розробки, які могли б забезпечити гнучкість, масштабованість та безпеку розроблюваного рішення.

Розробка та тестування прототипу є третім етапом. В його рамках було створено прототип програмного забезпечення, що враховує сучасні вимоги до аналізу продуктивності та етичні стандарти. Прототип пройшов ряд тестів, які підтвердили його ефективність та відповідність поставленим задачам.

Останнім етапом була аналіз даних та отримання результатів. За допомогою прототипу було проведено аналіз продуктивності співробітників у декількох організаціях, що дозволило отримати цінні дані та інсайти.

Новизна проєкту - проєкт пропонує інноваційний підхід до оцінки

продуктивності працівників, інтегруючи новітні технології машинного навчання та штучного інтелекту для автоматизованого аналізу великих об'ємів даних про робочу активність. Окрім того, в проєкті реалізовано комплексний підхід до забезпечення приватності та етичності моніторингу.

Основні методи дослідження: у даній дипломній роботі використано комплексний підхід до дослідження, що охоплює аналіз наукових публікацій для забезпечення глибокого розуміння теми і виявлення існуючих методів оцінки продуктивності. SWOT-аналіз допомагає визначити внутрішні та зовнішні фактори, які можуть вплинути на успіх проєкту, а детальне дослідження предметної області виявляє ключові технології і підходи до розробки програмного забезпечення. Розробка структурних схем та організаційних моделей дозволяє чітко визначити процеси в проєкті, в той час як застосування кількісних методів, таких як математичне моделювання і статистичний аналіз, забезпечує оцінку ефективності рішень. Також, управління ризиками відіграє ключову роль у мінімізації потенційних перешкод для успішної реалізації проєкту, забезпечуючи його стійкість та адаптивність до можливих змін.

Практичне значення:

- Підвищення продуктивності: Розроблене програмне забезпечення дозволяє організаціям ефективніше управляти продуктивністю своїх працівників, виявляючи та оптимізуючи ключові процеси.
- Оптимізація робочих процесів: Аналіз даних про продуктивність сприяє виявленню "вузьких місць" в робочих процесах та їх подальшій оптимізації.
- Покращення робочого середовища: Рекомендації, отримані на основі аналізу даних, можуть бути використані для створення більш здорового та продуктивного робочого середовища.

Таким чином, розроблене програмне забезпечення та отримані результати мають значний потенціал для покращення управління персоналом, оптимізації процесів та підвищення загальної ефективності роботи організацій.

Вступ до нашої роботи встановлює фундаментальну потребу у розробці

програмного забезпечення для оцінки продуктивності працівників як ключового елемента в підвищенні ефективності та конкурентоспроможності сучасних організацій. Через детальний аналіз існуючих методів оцінки, а також вивчення передових технологій та етичних аспектів управління даними працівників, ми підкреслюємо важливість створення інтегрованого рішення, що спроможне відповідати потребам організацій у точній та ефективній оцінці продуктивності.

Основні завдання нашої роботи включають аналіз існуючих підходів, розробку нових методик та впровадження етичних стандартів у моніторинг продуктивності, виходячи з чого ми прагнемо внести вклад у розвиток управління персоналом і оптимізацію бізнес-процесів.

Структура роботи: робота складається з анотації, вступу, основної частини, яка включає три розділи, висновків, списку використаних джерел та додатків.

Повний обсяг дослідження: робота містить 97 сторінок з урахуванням додатків.

РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИКИ ОБ'ЄКТУ УПРАВЛІННЯ

Об'єктом управління в контексті даної роботи є процес оцінки продуктивності працівників за допомогою спеціалізованого програмного забезпечення в організаційному середовищі.[1] Цей процес охоплює різноманітні аспекти, починаючи від збору ініціативних даних про діяльність співробітників до аналізу та використання цих даних для формування стратегічних рішень управління персоналом та підвищення загальної продуктивності компанії.

1.1 Огляд об'єкту управління

Об'єктом управління в контексті даної роботи є процес оцінки продуктивності працівників за допомогою спеціалізованого програмного забезпечення в організаційному середовищі. Цей процес охоплює різноманітні аспекти, починаючи від збору ініціативних даних про діяльність співробітників до аналізу та використання цих даних для формування стратегічних рішень управління персоналом та підвищення загальної продуктивності компанії.[2]

1.1.1 Загальний опис об'єкта управління

Процес оцінки продуктивності є ключовим для забезпечення ефективної роботи організацій.[3] Він дозволяє керівництву оцінювати внесок кожного працівника у загальні цілі компанії, ідентифікувати потреби в навчанні та розвитку, а також сприяє підвищенню мотивації працівників через систематичний зворотний зв'язок та визнання їхніх досягнень.

1.1.2 Історичний контекст та еволюція об'єкта управління

Об'єкт управління нашого дослідження, процес оцінки продуктивності працівників за допомогою програмного забезпечення, має глибокі історичні корені, що сягають часів промислової революції, коли почала формуватися

потреба у вимірюванні та аналізі ефективності робочої сили.[4] Розвиток цього процесу можна розглянути через кілька ключових етапів:

Передтехнологічний етап

До виникнення комп'ютерних технологій оцінка продуктивності в основному здійснювалася через безпосереднє спостереження та ручний аналіз діяльності працівників. У цей період акцент робився на фізичній працездатності та кількості виконаної роботи, а методи оцінки були відносно примітивними та суб'єктивними.

Етап автоматизації

З появою перших комп'ютерів та програмного забезпечення в середині 20-го століття почалася ера автоматизації багатьох процесів, у тому числі і оцінки продуктивності.[5] Програмне забезпечення дозволило збільшити обсяг та точність збору даних, а також спростити їх аналіз. Однак, ці системи часто були занадто загальними та не дозволяли глибокого аналізу специфічних для кожної компанії процесів.[6]

Етап розвитку інформаційних систем управління

З розвитком IT-індустрії та підвищенням потужності обчислювальної техніки з'явилася можливість створення комплексних інформаційних систем управління, здатних інтегрувати великі обсяги даних, проводити їх багатовимірний аналіз та формувати детальні звіти про продуктивність. Це дало поштовх до розвитку ERP-систем (Enterprise Resource Planning) та систем управління відносинами з клієнтами (CRM), які включали модулі для оцінки роботи персоналу.

Етап цифрової трансформації

На сучасному етапі, в умовах цифрової трансформації, використання штучного інтелекту, машинного навчання та big data для аналізу продуктивності відкриває нові горизонти.[7] Алгоритми штучного інтелекту дозволяють не просто аналізувати наявні дані, а й прогнозувати майбутню продуктивність, ідентифікувати приховані закономірності в роботі персоналу та автоматизувати процеси управління.

Майбутнє

Перспективи розвитку об'єкта управління пов'язані з подальшим удосконаленням технологій аналізу даних, зокрема через розвиток квантових обчислень та нейромережевих технологій.[8] Це дозволить організаціям не тільки оцінювати продуктивність на основі історичних даних, але й ефективно адаптуватися до мінливих умов ринку, прогнозуючи майбутні тренди та оптимізуючи робочі процеси в реальному часі.[9]

Цей історичний огляд підкреслює, як еволюція технологій та методологій оцінки продуктивності працівників сприяла поступовому переходу від простих, часто суб'єктивних методів до складних, інтегрованих систем управління, що використовують передові технології для підвищення ефективності організацій.

1.1.3 Сучасне положення та значення в системі управління

На сьогоднішній день, використання програмного забезпечення для оцінки продуктивності стало невід'ємною частиною стратегічного управління в багатьох організаціях.[10] Ці системи забезпечують автоматизацію збору даних, аналітику, візуалізацію KPI (ключових показників ефективності) та інші функції, що допомагають оптимізувати процеси та підвищити загальну продуктивність.[11] Розвиток штучного інтелекту та машинного навчання відкриває нові перспективи для подальшого удосконалення цих систем, зокрема, за рахунок автоматизованого розпізнавання патернів продуктивності та адаптивного управління робочими процесами.[12]

1.2 Проведення аналізу предметної галузі

1.2.1 Визначення предметної галузі

Аналіз предметної галузі, а саме процесів оцінки продуктивності працівників через програмне забезпечення, є ключовим кроком у розробці ефективних методів та інструментів управління.[13] Цей пункт детально описує механізми вивчення та аналізу основних компонентів, технік і технологій, задіяних у процесі оцінки продуктивності.[14]

Предметна галузь охоплює системи і процеси, що дозволяють організаціям вимірювати, аналізувати та оптимізувати продуктивність своїх співробітників. Це включає різноманітні методики оцінки, від кількісних метрик до якісних оцінок, а також використання програмного забезпечення для автоматизації цих процесів.[15]

1.2.2 Аналіз поточного стану предметної галузі

Поточний стан предметної галузі, пов'язаної з оцінкою продуктивності працівників за допомогою програмного забезпечення, характеризується швидким розвитком та інтеграцією новітніх технологій.[16] Розглядаючи цей процес, основну увагу зосередимо на наступних аспектах:

1. Методики оцінки продуктивності

Сучасні методики оцінки включають в себе широкий спектр підходів, починаючи від традиційних ревію і самооцінок до використання комплексних програмних рішень, що забезпечують кількісний аналіз продуктивності на основі великих об'ємів даних. Цифрові інструменти та платформи дозволяють автоматизувати процес збору даних і проводити глибокий аналіз продуктивності, включаючи відстеження часу, аналіз завдань і проектів.

2. Технологічні рішення

В області технологічних рішень останніми роками спостерігається значний прогрес. Ринок пропонує широкий спектр програмного забезпечення для оцінки продуктивності, яке інтегрується з іншими системами управління в організації, такими як ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) та HRM (Human Resource Management).[17] Такі системи забезпечують не лише збір та аналіз даних, але й пропонують інструменти для планування ресурсів, управління проектами та комунікації в команді.

3. Проблеми та виклики

Незважаючи на технологічний прогрес, існують певні виклики та проблеми, з якими стикаються організації при оцінці продуктивності працівників. Однією з основних проблем є забезпечення точності та

об'єктивності даних, що вимагає розробки комплексних алгоритмів аналізу. Крім того, питання конфіденційності та етики стає все більш актуальним у зв'язку зі збором та аналізом великих об'ємів інформації про діяльність співробітників.[9]

4. Вплив на ринок праці та корпоративну культуру

Оцінка продуктивності має великий вплив не тільки на управління ресурсами компанії, але й на корпоративну культуру та мотивацію співробітників. З одного боку, це може сприяти підвищенню загальної продуктивності та ефективності роботи.[18] З іншого боку, неправильне використання даних або надмірний контроль може призвести до зниження довіри та задоволення роботою серед співробітників.

Поточний стан предметної галузі демонструє, що оцінка продуктивності працівників за допомогою програмного забезпечення є важливим інструментом для сучасних організацій. Проте, для досягнення оптимальних результатів, необхідно враховувати ряд викликів та тенденцій, які формують цю галузь.

1.2.3 Аналіз існуючих методів оцінки

Системи контролю якості працівників використовуються в організаціях для моніторингу, оцінки та підвищення продуктивності персоналу.[19] Вони варіюються від традиційних методів оцінки до сучасних програмних рішень, кожен з яких має свої переваги та недоліки.

Створення таблиці переваг та недоліків існуючих методів є важливим інструментом у процесі планування та виконання проектів. Ця таблиця дозволяє не тільки порівняти різні підходи, але й допомагає ідентифікувати найефективніші методи, які можуть бути адаптовані до специфіки проекту. Вона надає змогу проектним командам враховувати різноманітні фактори, такі як час, вартість, ресурси та можливі ризики, пов'язані з кожним методом. Завдяки цьому, команда може ухвалювати обґрунтовані рішення, оптимізувати процеси та мінімізувати потенційні проблеми, покращуючи загальну продуктивність та якість кінцевого продукту. Це також сприяє кращому

управлінню очікуваннями зацікавлених сторін та підвищує прозорість проектного процесу.

Таблиця 1.1

Переваги та недоліки існуючих методів

Метод	Переваги	Недоліки
360-градусна зворотній зв'язок	<ul style="list-style-type: none"> • Забезпечує всебічний огляд продуктивності працівника, оскільки зворотній зв'язок надходить від різних джерел, включаючи керівників, колег по роботі, підлеглих та навіть клієнтів. • Сприяє розвитку корпоративної культури зворотного зв'язку та самовдосконалення. 	<ul style="list-style-type: none"> • Може сприйматися як суб'єктивний та упереджений, особливо якщо інформатори не надають чесний зворотній зв'язок. • Організація та аналіз зворотного зв'язку вимагають значних часових та адміністративних ресурсів.
Ключові показники ефективності (КРІ)	<ul style="list-style-type: none"> • Чіткі та об'єктивні критерії оцінки, що засновані на кількісних даних, як-от досягнення продажів, якість роботи, терміни виконання проектів тощо. • Сприяє цілеспрямованому управлінню та плануванню, оскільки працівники знають, що від них очікують. 	<ul style="list-style-type: none"> • Може не враховувати не кількісні аспекти роботи, такі як творчий підхід, командна взаємодія та ініціативність. • Надмірна фокусація на КРІ може спонукати працівників до маніпулювання результатами або ігнорування довгострокових цілей на користь короткострокових показників.

<p>Системи управління проектами</p>	<ul style="list-style-type: none"> • Забезпечують централізоване місце для моніторингу ходу роботи, розподілу завдань та спілкування всередині команд. <p>Дозволяють керівництву відстежувати прогрес проєктів в реальному часі та виявляти затримки або проблеми на ранніх стадіях.</p>	<ul style="list-style-type: none"> • Вимагають від працівників додаткового часу на ведення документації та оновлення статусів завдань, що може сприйматися як відволікання від основної роботи. <p>Залежні від якості вхідних даних; неточності у введеній інформації можуть призвести до неправильного розуміння загального прогресу.</p>
<p>Автоматизовані системи моніторингу часу</p>	<ul style="list-style-type: none"> • Забезпечують точний облік часу, проведеного над конкретними завданнями або проєктами. <p>Можуть автоматично генерувати звіти про продуктивність, допомагаючи ідентифікувати області для підвищення ефективності.</p>	<ul style="list-style-type: none"> • Сприйняття нагляду може негативно вплинути на моральний дух та відносини в команді. <p>Може не враховувати якість виконаної роботи, зосереджуючись лише на кількості витраченого часу.</p>

Проаналізувавши існуючі методи оцінки якості можна чітко виокремити необхідні критерії, які має враховувати високоякісна система.

1.2.4 Аналіз конкурентів

Програмне забезпечення для оцінки продуктивності працівників допомагає керівництву компаній збирати, аналізувати та використовувати дані для покращення робочих процесів та ефективності персоналу. Нижче наведено

кілька прикладів таких систем:

1. BambooHR

Опис: BambooHR є комплексним рішенням для управління персоналом, що включає функції для відстеження часу та відпусток, автоматизації HR-процесів та проведення оцінки продуктивності. [20]

Переваги: забезпечує легку інтеграцію з іншими інструментами, має зручний інтерфейс, підтримує різноманітні HR-функції в одному місці.

Недоліки: може бути дорогим для дуже малих компаній, обмеження на кількість користувачів у безкоштовній версії.

2. Workday

Опис: Workday пропонує широкий спектр інструментів для управління персоналом та фінансами, зокрема для планування, аналітики та оцінки продуктивності працівників.

Переваги: інтегрує управління персоналом з фінансовим плануванням, має гнучкі інструменти для аналітики та звітності.

Недоліки: конфігурація та налаштування системи можуть вимагати часу та ресурсів, висока вартість.

3. Performance Pro

Опис: Performance Pro спеціалізується на автоматизації процесів оцінки продуктивності та управлінні талантами, забезпечуючи інструменти для налаштування цілей та відстеження досягнень працівників.[21]

Переваги: гнучкі можливості для налаштування процесів оцінки, підтримка цілей розвитку працівників, зручний інтерфейс.

Недоліки: може вимагати певного часу для повного освоєння всіх функцій, орієнтований більше на середні та великі компанії.

4. 15Five

Опис: 15Five - це платформа для зворотного зв'язку та оцінки продуктивності, яка дозволяє працівникам щотижня ділитися оновленнями та досягненнями, а керівникам - відслідковувати прогрес та визначати області для

розвитку.[10]

Переваги: сприяє регулярному зворотному зв'язку та комунікації між керівниками та працівниками, інтуїтивно зрозумілий інтерфейс.

Недоліки: фокусується переважно на зворотному зв'язку та щотижневих оновленнях, може не покривати всі потреби у комплексній оцінці продуктивності.

5. UltiPro

Опис: UltiPro надає комплексне рішення для управління персоналом, включаючи модулі для оцінки продуктивності, які дозволяють компаніям автоматизувати процес оцінки та розвитку співробітників.

Переваги: Інтеграція з іншими HR-функціями, високий рівень кастомізації, зручний користувацький інтерфейс.

Недоліки: Висока вартість рішення, складність у впровадженні та налаштуванні системи, потребує тривалого навчання персоналу.

6. Zoho People

Опис: Zoho People — це частина більшого набору інструментів Zoho для управління бізнесом, що включає функціонал для оцінки та аналізу продуктивності працівників.

Переваги: легкість інтеграції з іншими продуктами Zoho, доступна ціна, хороша підтримка користувачів.

Недоліки: обмежені можливості для великих підприємств, іноді зустрічаються проблеми зі швидкістю системи.

7. SAP SuccessFactors

Опис: SAP SuccessFactors є частиною широкої лінійки ERP-рішень, пропонуючи розширені можливості для оцінки та управління продуктивністю працівників на всіх рівнях організації.

Переваги: глибока інтеграція з бізнес-процесами компанії, масштабованість, потужні аналітичні інструменти.

Недоліки: висока вартість, складність конфігурації та потреба в професійній підтримці, може бути надто складним для малих підприємств.

1.2.5 SWOT-аналіз

Для демонстрації SWOT аналізу проєкту "Розробка програмного забезпечення для оцінки продуктивності працівників", розглянемо ключові сильні та слабкі сторони проєкту, а також можливості та загрози, які він може зустріти на ринку. [22]

Таблиця 1.2

Матриця здійсненого SWOT-аналізу проєкту

Категорія	Опис
Сильні сторони (Strengths)	<ul style="list-style-type: none">– Інноваційний підхід до оцінки продуктивності.– Гнучкість та масштабованість рішення.– Інтеграція з існуючими корпоративними системами.– Підтримка рішень на основі даних для керівництва.
Слабкі сторони (Weaknesses)	<ul style="list-style-type: none">– Висока вартість розробки та впровадження.– Потреба в тренуванні користувачів.– Відсутність ринкового досвіду або визнання.– Потенційна складність інтеграції з деякими старими системами.
Можливості (Opportunities)	<ul style="list-style-type: none">– Зростаючий попит на інструменти для підвищення продуктивності.– Розвиток технологій штучного інтелекту та аналітики даних.– Можливість розширення на нові ринки та індустрії.– Співпраця з навчальними закладами та науково-дослідними інститутами.
Загрози (Threats)	<ul style="list-style-type: none">– Сильна конкуренція на ринку ПЗ– Швидкі зміни в технологічних стандартах та практиках.– Можливий опір з боку співробітників та профспілок

Цей SWOT аналіз допомагає ідентифікувати ключові аспекти проєкту, які потрібно врахувати для успішного розвитку та реалізації. Він також надає важливу інформацію для розробки стратегій, спрямованих на мінімізацію ризиків і максимізацію успіху проєкту на ринку.

1.3 Проведення маркетингових досліджень

1.3.1 STEP-аналіз

STEP (або PEST) аналіз — це інструмент стратегічного планування, який дозволяє оцінити зовнішні фактори, що впливають на проєкт або бізнес. Він включає аналіз Соціальних (Social), Технологічних (Technological), Економічних (Economic) та Політичних (Political) аспектів.[23] Давайте проведемо STEP аналіз для проєкту розробки програмного забезпечення для оцінки продуктивності працівників.

Таблиця 1.3

Фактори впливу

Соціальні фактори	Технологічні фактори
<ul style="list-style-type: none"> • Зростаючий попит на прозорість робочих процесів з боку як співробітників, так і керівництва може сприяти впровадженню системи. • Зміна робочих умов, таких як збільшення дистанційної роботи, вимагає більш гнучких підходів до оцінки продуктивності. • Етичні занепокоєння щодо моніторингу та збору даних про працівників. 	<ul style="list-style-type: none"> • Швидкий розвиток технологій, зокрема штучного інтелекту та машинного навчання, надає нові можливості для аналізу продуктивності. • Збільшення доступності та зниження вартості хмарних технологій робить програмне забезпечення для оцінки продуктивності більш доступним для широкого кола організацій. • Покращення інструментів збору та дозволяє збирати більш точні дані

Економічні фактори	Політичні фактори
<ul style="list-style-type: none"> • Економічна нестабільність може збільшити потребу в оптимізації ресурсів та підвищенні продуктивності. • Бюджетні обмеження в організаціях можуть вплинути на рішення про впровадження нових систем. <p style="text-align: center;">Рентабельність інвестицій в програмне забезпечення для оцінки продуктивності є критичним фактором для більшості організацій.</p>	<ul style="list-style-type: none"> • Законодавство щодо приватності та захисту даних (наприклад, GDPR в ЄС) встановлює строгі вимоги до збору та обробки персональних даних. • Державні та міжнародні регуляції можуть впливати на доступність та використання певних технологій у оцінці продуктивності. <p style="text-align: center;">Політична стабільність в країні та на міжнародному рівні може впливати на інвестиційний клімат і готовність компаній інвестувати в нові технології.</p>

Розуміння цих зовнішніх факторів є важливим для стратегічного планування та успішного впровадження проєкту розробки програмного забезпечення для оцінки продуктивності працівників.

1.3.2 Аналіз внутрішнього середовища

Аналіз внутрішнього середовища проєкту зосереджується на визначенні основних учасників, їх ролях, впливі на проєкт, а також потенційних сильних та слабких сторонах в контексті проєкту розробки програмного забезпечення для оцінки продуктивності працівників.[11] Основні учасники проєкту включають:

1. Команда розробників:

- Роль: Розробка, тестування та впровадження програмного забезпечення.
- Вплив на проєкт: Ключовий вплив на якість, технічні характеристики та відповідність програмного забезпечення вимогам користувачів.
- Сильні сторони: Технічна експертиза, знання новітніх технологій.

- Слабкі сторони: Обмеження в ресурсах, можливі затримки в розробці через неочікувані технічні проблеми.

2. Керівництво проєкту:

- Роль: Планування, координація роботи команди, контроль виконання проєкту.

- Вплив на проєкт: Визначення стратегічних цілей, розподіл ресурсів, управління ризиками.

- Сильні сторони: Досвід управління проєктами, знання ринку.

- Слабкі сторони: Можливе недостатнє розуміння деяких технічних аспектів проєкту.

3. Відділ маркетингу та продажів:

- Роль: Промоція продукту, залучення клієнтів, збір зворотного зв'язку від ринку.

- Вплив на проєкт: Впливає на розробку функціоналу на основі потреб ринку, формування цінової політики.

- Сильні сторони: Розуміння потреб клієнтів, ефективні стратегії продажу.

- Слабкі сторони: Можливе неповне розуміння технічних обмежень продукту.

4. Клієнти (бета-тестери):

- Роль: Тестування програмного забезпечення, надання зворотного зв'язку про його функціональність та користувацький досвід.

- Вплив на проєкт: Важливі для виявлення помилок, оцінки зручності користування та визначення потреб користувачів.

- Сильні сторони: Забезпечують реальне розуміння того, як продукт використовується в житті.

- Слабкі сторони: Обмежене розуміння можливостей розвитку продукту, фокус на конкретних, іноді незначних, проблемах.

5. Відділ підтримки та обслуговування:

- Роль: Надання технічної підтримки користувачам, збір зворотного зв'язку та запитів на нові функції.

- Вплив на проект: Важливі для вдосконалення користувацького досвіду та задоволеності клієнтів.
- Сильні сторони: Безпосереднє спілкування з користувачами, швидке виявлення та вирішення проблем.
- Слабкі сторони: Може зіткнутися з великою кількістю запитів, що вимагають швидкого реагування.
-

1.3.3 Аналіз зовнішнього середовища

Аналіз зовнішнього середовища проекту розробки програмного забезпечення для оцінки продуктивності працівників зосереджується на ідентифікації факторів, які можуть вплинути на його успіх ззовні.[24] Для цього ми використовуємо STEP аналіз, який був описаний раніше, щоб розглянути соціальні, технологічні, економічні та політичні фактори.[25]

1. Соціальні фактори:

- Зміна в робочих умовах: Пандемія COVID-19 сприяла переходу багатьох компаній на дистанційну роботу, що збільшило потребу в інструментах для моніторингу та оцінки продуктивності дистанційних працівників.
- Зростання уваги до робочого балансу та добробуту співробітників: Компанії все більше зосереджуються на створенні здорового робочого середовища, що може впливати на підходи до оцінки продуктивності.

2. Технологічні фактори:

- Швидкий розвиток технологій: Нові технології, такі як штучний інтелект та машинне навчання, відкривають нові можливості для більш ефективного аналізу продуктивності.
- Популяризація хмарних рішень: Хмарні платформи спрощують доступ до програмного забезпечення для оцінки продуктивності та його масштабування.

3. Економічні фактори:

- Глобальна економічна нестабільність: Економічні коливання можуть вплинути на бюджети компаній та їх готовність інвестувати в нове програмне

забезпечення.

- Ринкова конкуренція: Сильна конкуренція в галузі програмного забезпечення може впливати на ціноутворення та маркетингові стратегії.

4. Політичні фактори:

- Регулювання приватності даних: Законодавство, таке як GDPR у Європейському Союзі, встановлює строгі вимоги до обробки персональних даних, що може вплинути на розробку та впровадження програмного забезпечення.

- Політична стабільність: Нестабільність у регіоні або зміни в урядовій політиці можуть впливати на економічне середовище та ринкові можливості.

Розуміння цих зовнішніх факторів є критично важливим для успішного планування та реалізації проекту. Воно дозволяє проєктній команді адаптуватися до зовнішніх умов, використовувати можливості та мінімізувати ризики, пов'язані з зовнішнім середовищем.

1.4 Інвестиційні дослідження

Щоб провести інвестиційне дослідження для проєкту розробки програмного забезпечення для оцінки продуктивності працівників, ми могли б розглянути такі ключові показники як внутрішня норма прибутковості (ВНП, англ. IRR), чистий приведений дохід (ЧПД, англ. NPV) та період окупності. Однак, для цього нам потрібно б мати конкретні цифри, включаючи очікувані витрати на розробку, операційні витрати, прогнозовані доходи від продажів чи ліцензування, а також шкалу часу проєкту.[12]

Початкові інвестиції: \$500,000

Очікуваний дохід за рік 1: \$100,000

Очікуваний дохід за рік 2: \$200,000

Очікуваний дохід за рік 3: \$300,000

Очікуваний дохід за рік 4 та далі: \$400,000 на рік

Оцінка тривалості проєкту: 5 років

Дисконтна ставка (ставка дисконтування): 10%

На основі цих даних ми можемо розрахувати NPV і IRR, а також намалювати графік очікуваних грошових потоків.

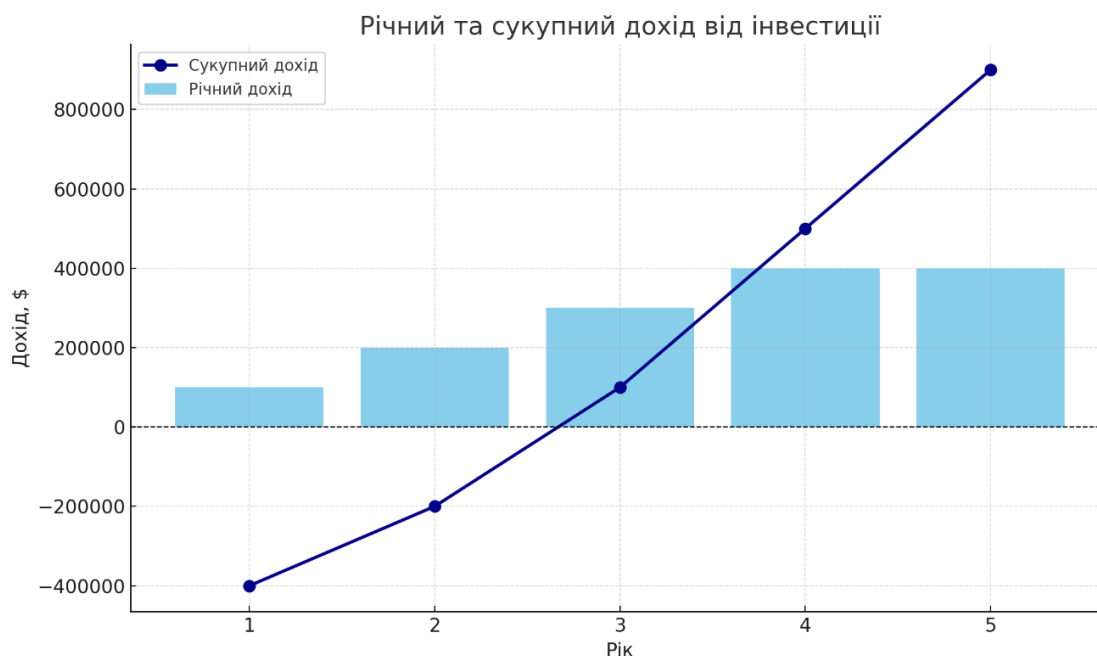
Розрахуємо NPV та IRR на основі наданих даних.

На основі проведеного аналізу, результати є наступними:

Чистий приведений дохід (NPV): \$503,166.70

Внутрішня норма прибутковості (IRR): 36.34%

Ці результати вказують на те, що проект є фінансово привабливим, оскільки NPV є позитивним, а IRR перевищує дисконтну ставку (10%), що свідчить про високу очікувану прибутковість інвестицій у проект. Тепер, враховуючи ці дані, ми можемо намалювати графіки очікуваних грошових



потоків.

Рис. 1.1. Графік доходу

Нехай, що чистий обіговий капітал (ЧОК) та щомісячні платежі за обслуговування заборгованості (ОЗ) для проекту виглядають наступним чином протягом 5 років:

Чистий обіговий капітал (ЧОК) змінюється від \$50,000 у перший рік до \$150,000 у п'ятий рік, збільшуючись на \$25,000 щороку.

Обслуговування заборгованості (ОЗ) складає \$20,000 щорічно протягом

перших 3 років та зменшується до \$10,000 на 4 та 5 рік.

Для оцінки фінансової стійкості та ефективності управління ресурсами проекту побудуємо графік обслуговування заборгованості та графік чистого обігового капіталу.

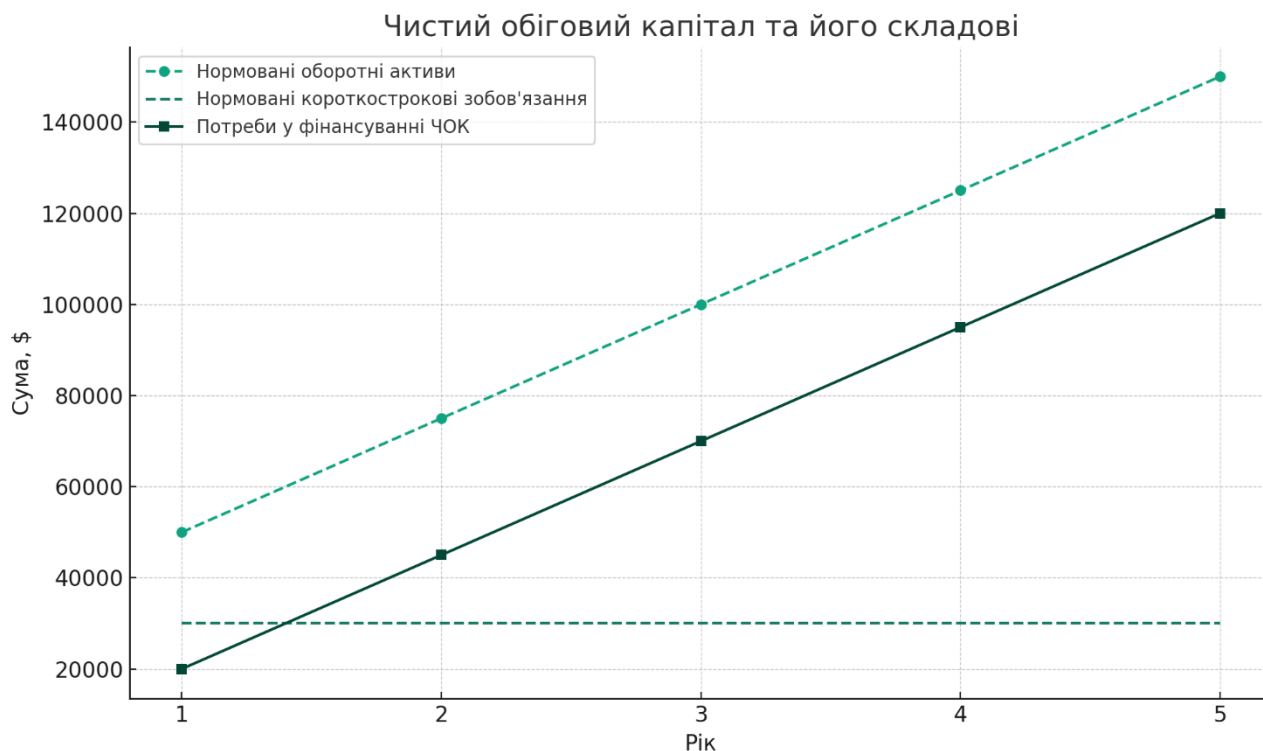


Рис. 1.2. Чистий обіговий капітал

Графік чистого обігового капіталу відображає баланс між оборотними активами та короткостроковими зобов'язаннями проекту, демонструючи його ліквідність та фінансову гнучкість. Нормовані оборотні активи (наприклад, готівка, дебіторська заборгованість) порівнюються з нормованими короткостроковими зобов'язаннями (наприклад, кредиторська заборгованість), щоб визначити потреби у фінансуванні чистого оборотного капіталу. Це критично важливо для забезпечення плавної операційної діяльності проекту.

На графіку представлено динаміку чистого обігового капіталу (ЧОК) та його ключових складових протягом 5 років:

Нормовані оборотні активи зростають щороку, що свідчить про збільшення ресурсів компанії, доступних для підтримки її операційної діяльності.

Нормовані короткострокові зобов'язання залишаються стабільними протягом усього періоду, відображаючи постійний рівень зобов'язань, які потрібно покрити в короткостроковій перспективі.

Потреби у фінансуванні чистого обігового капіталу показують збільшення потреби в додаткових ресурсах для підтримки зростання активів, виходячи з припущення, що обсяг короткострокових зобов'язань залишається незмінним.

Цей графік корисний для оцінки фінансової здатності проєкту покривати свої короткострокові зобов'язання за рахунок оборотних активів та визначення загальних потреб у фінансуванні для забезпечення стійкого зростання.

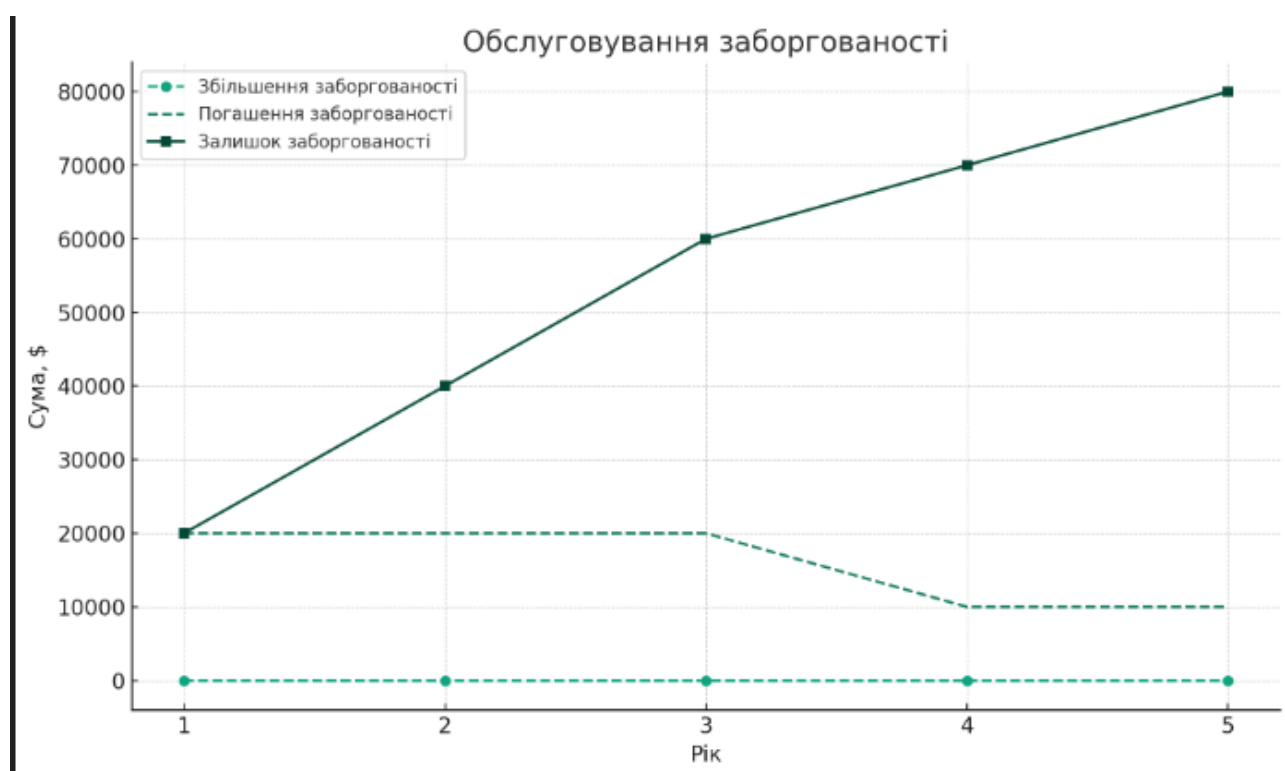


Рис. 1.3. Обслуговування заборгованості

Графік обслуговування заборгованості дозволяє візуалізувати структуру та динаміку заборгованості проєкту, включаючи збільшення заборгованості через нові позики, погашення основної суми та відсотків, а також залишок заборгованості на кінець кожного періоду. Це допомагає зрозуміти, як проєкт управляє своїм боргом та його спроможність генерувати достатньо коштів для покриття зобов'язань.

На даному графіку представлено динаміку збільшення заборгованості,

погашення заборгованості та залишку заборгованості для проєкту протягом 5 років.

Збільшення заборгованості залишається на нульовому рівні, оскільки, згідно з припущеннями, нові борги не набираються.

Погашення заборгованості відображає річні виплати за обслуговування заборгованості, які становлять \$20,000 на рік протягом перших трьох років і знижуються до \$10,000 у четвертому та п'ятому роках.

Залишок заборгованості демонструє сукупний ефект збільшення та погашення заборгованості, що в цьому випадку відображає загальну суму погашення заборгованості через відсутність нового боргу.

Цей графік корисний для візуального аналізу зобов'язань проєкту та оцінки його фінансової здатності обслуговувати та погашати заборгованість у встановлені терміни.

1.5 Дерево проблем, дерево цілей та логіко структурна схема

Для розробки дерева проблем для проєкту розробки програмного забезпечення для оцінки продуктивності працівників було обрано основну проблему та розглянуто причини, що призводять до неї, а також наслідки, які ця проблема спричиняє.



Рис. 1.4. Дерево проблем

Це візуалізація демонструє як основну проблему, так і зв'язані з нею причини та наслідки на трьох рівнях, допомагаючи краще зрозуміти структуру та зв'язки між різними аспектами ситуації.

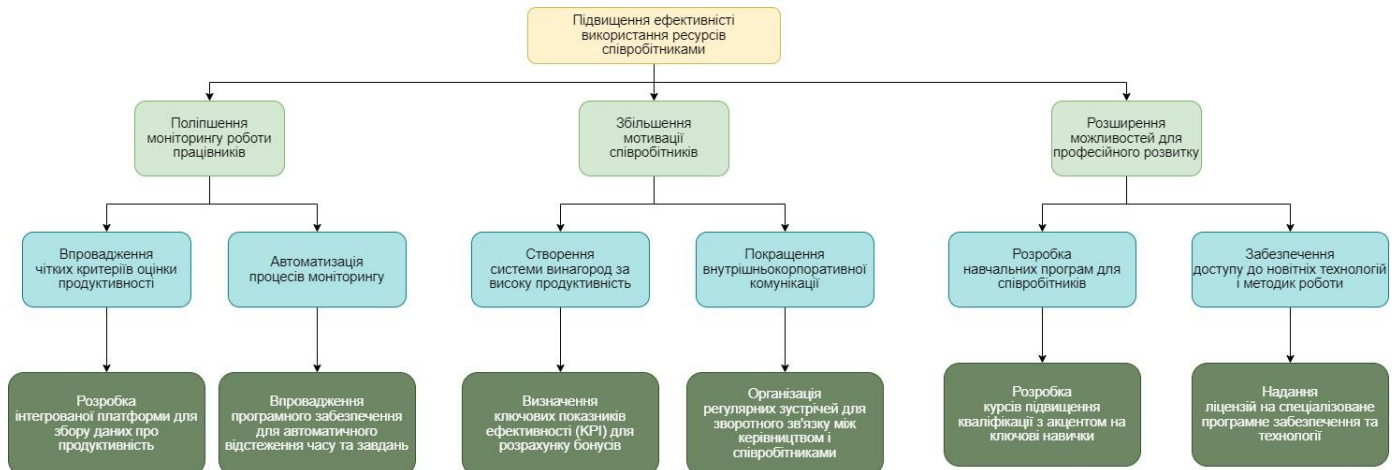


Рис. 1.5. Дерево цілей

Дерево цілей є важливим інструментом у процесі стратегічного планування проєктів, оскільки воно дозволяє не тільки структурувати і візуалізувати основні цілі проєкту, але й деталізувати їх до конкретних завдань і підзавдань.[26] Така ієрархічна структура сприяє кращому розумінню того, як досягнення окремих завдань веде до реалізації загальної мети проєкту. Крім того, дерево цілей значно полегшує комунікацію між усіма зацікавленими сторонами, включаючи керівництво, інвесторів та виконавців проєкту, надаючи чітке бачення цілей та шляхів їх досягнення.

Використання дерева цілей також сприяє більш ефективному плануванню проєкту, дозволяючи точно розподіляти ресурси та визначати терміни виконання окремих завдань. Це стає основою для моніторингу прогресу проєкту та оцінки його успішності на кожному етапі виконання. Завдяки визначенню пріоритетів, команда проєкту може зосередитися на ключових аспектах, що мають найбільше значення для досягнення кінцевої мети.

Не в останню чергу, дерево цілей забезпечує гнучкість управління проєктом, дозволяючи адаптувати стратегію та тактику виконання до змін у

внутрішньому та зовнішньому середовищі. Таким чином, цей інструмент не лише сприяє ефективній реалізації проєкту, але й підвищує його стійкість до різноманітних викликів, забезпечуючи досягнення поставлених цілей в умовах невизначеності.

Для створення логіко структурної схеми проєкту розробки програмного забезпечення для оцінки продуктивності працівників, ми розробимо структуру, яка включає ключові компоненти проєкту, їх взаємозв'язки та вихідні результати. Схема буде представлена у вигляді таблиці, яка демонструє як вхідні дані, процеси, вихідні дані, так і залучених стейкхолдерів.

Таблиця 1.4

Логіко структурна схема

Компонент	Опис	Вхідні дані	Процеси	Вихідні дані	Стейкхолдери
Аналіз вимог	Визначення функціональних та нефункціональних вимог до системи	Звіти про продуктивність, Відгуки користувачів	Збір вимог, Інтерв'ю з керівництвом та співробітниками	Специфікація вимог	Керівництво, Співробітники
Розробка дизайну	Створення архітектури системи та дизайну інтерфейсу	Специфікація вимог	Розробка концепту дизайну, Прототипування	Дизайн-макети, Архітектурна система	Дизайнери, Розробники
Реалізація	Кодування та інтеграція компонентів системи	Дизайн-макети, Архітектура системи	Написання коду, Інтеграція модулів	Програмне забезпечення	Розробники

Тестування	Перевірка функціональності та виявлення помилок	Програми забезпечення	Юніт-тестування, Інтеграційне тестування, Тестування користувачького інтерфейсу	Звіти про тестування, Версія для релізу	Тестувальники
Впровадження	Запуск системи для кінцевих користувачів	Версія для релізу	Налаштування середовища, Розгортання системи	Робоча система	ІТ-адміністратори
Навчання та підтримка	Організація навчальних сесій для користувачів, забезпечення технічної підтримки	Робоча система, Посібники користувача	Проведення тренінгів, Забезпечення підтримки	Задоволені користувачі, Звіти про використання	Користувачі, Техпідтримка
Моніторинг та оцінка	Постійний аналіз ефективності системи та її впливу на продуктивність	Звіти про використання, Відгуки користувачів	Збір даних, Аналітика	Рекомендації для покращення	Керівництво, Аналітики

1.6 Мета, цілі та продукт проєкту

Метою проєкту є: розробити програмне забезпечення (ПЗ), що дозволить компаніям ефективно оцінювати, моніторити та підвищувати продуктивність своїх працівників, використовуючи автоматизовані інструменти для збору даних, аналізу та відстеження ключових показників ефективності (КРІ).

Цілі проєкту:

1. Розробка IT-рішення: Створити інтуїтивно зрозуміле, гнучке та масштабоване програмне забезпечення, що відповідає потребам бізнесу різного масштабу у сфері управління персоналом та оцінки їх продуктивності.

2. Інтеграція з існуючими системами: Забезпечити можливість легкої інтеграції програмного забезпечення з вже використовуваними в компанії HRM-системами, ERP-системами та іншим корпоративним програмним забезпеченням.

3. Доступність аналітики та звітності: Розробити функціонал для генерації звітів і аналітики щодо продуктивності працівників на основі зібраних даних, з можливістю кастомізації показників та звітів відповідно до конкретних потреб компанії.

4. Забезпечення приватності та захисту даних: Гарантувати високий рівень безпеки зберігання та обробки персональних даних працівників у відповідності до міжнародних стандартів та законодавства.

Продуктом проєкту є: програмне забезпечення для оцінки продуктивності працівників – це комплексне рішення, що дозволяє автоматизувати процеси моніторингу, оцінки та управління продуктивністю персоналу. Продукт надає інструменти для збору даних про робочу активність, аналізу продуктивності на основі встановлених КРІ та відстеження динаміки змін продуктивності в часі.

Функціонал ПЗ:

– Автоматизований збір даних: відстеження часу, завдань, проєктів, які виконують працівники.

– Гнучкі налаштування КРІ: можливість встановлення індивідуальних критеріїв оцінки для різних підрозділів або ролей в компанії.

- Інтеграція: сумісність з іншими системами управління ресурсами компанії.
- Модуль аналітики та звітності: генерація звітів, аналіз трендів продуктивності, візуалізація даних.
- Безпека даних: захист інформації з використанням сучасних методів шифрування та автентифікації.

Цей продукт розроблено з метою допомогти компаніям підвищити ефективність використання людських ресурсів, покращити управління персоналом і стимулювати зростання продуктивності праці на основі даних та аналітики.

1.7 Дослідження методів розробки ПЗ

У процесі розробки програмного забезпечення важливо вибрати методологію, яка найкраще відповідає вимогам проекту, структурі команди та кінцевим цілям продукту. Існує кілька основних методологій, таких як водоспад (Waterfall), гнучка (Agile), спритна (Scrum), канбан (Kanban) та інші. Кожна з них має свої переваги та недоліки, в залежності від специфіки проекту.[27]

Для проекту розробки програмного забезпечення для оцінки продуктивності працівників, я рекомендував би використовувати методологію Scrum.

Обґрунтування вибору Scrum

Scrum — це гнучка методологія, яка підходить для проектів, де важко повністю визначити вимоги заздалегідь, і де можливі часті зміни протягом процесу розробки.[28] Ось декілька ключових причин, чому Scrum є оптимальним вибором для цього проекту:

Гнучкість до змін: Scrum забезпечує високу адаптивність до змінюваних вимог користувачів або ринку. Це критично важливо для проектів, які розробляються у швидко змінюваних умовах.

Ітераційний підхід: Розробка ведеться короткими спринтами, що тривають зазвичай 2-4 тижні. Кожен спринт завершується створенням робочого продукту, що дозволяє регулярно перевіряти прогрес та вносити необхідні корективи.

Залучення зацікавлених сторін: Scrum фокусується на тісній співпраці між усіма учасниками проекту, включаючи регулярні наради (щоденні стендапи), перегляд спринтів та ретроспективи. Це сприяє кращому розумінню потреб клієнта та ефективній комунікації.[29]

Прозорість процесу: Всі ключові зацікавлені сторони мають постійний доступ до інформації про статус проекту, що підвищує довіру та забезпечує чітке уявлення про прогрес.

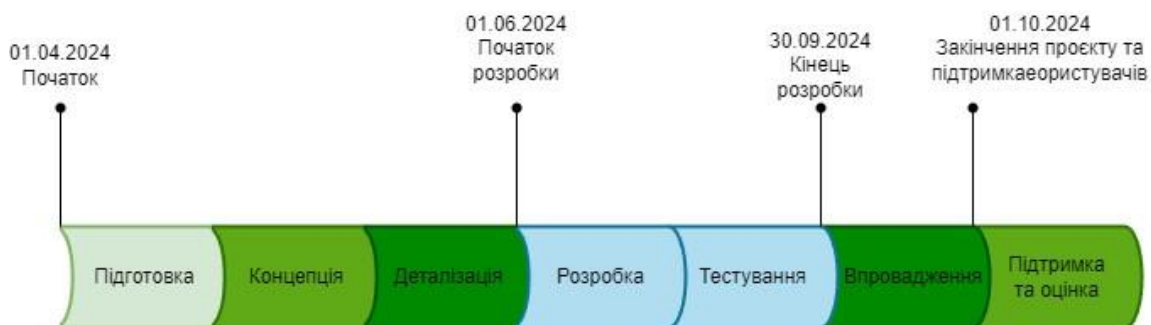
Покращення продуктивності команди: Scrum допомагає підвищити продуктивність завдяки чіткій ролі та відповідальності кожного члена команди, що важливо для успішної реалізації проекту.

Застосування методології Scrum дозволить збільшити шанси на успіх проекту за рахунок адаптивного підходу до управління змінами та високого рівня залучення всіх учасників проекту.[30]

РОЗДІЛ 2 ПЛАНУВАННЯ ВИКОНАННЯ ПРОЄКТУ

2.1 Розробка життєвого циклу проєкту. Побудова ієрархічної структури робіт проєкту.

Життєвий цикл проєкту розробки програмного забезпечення для оцінки продуктивності працівників можна поділити на кілька ключових етапів, кожен з



яких має визначені цілі та результати.

Рис. 2.1. Життєвий цикл проєкту

Таблиця 2.1

Опис фаз життєвого циклу проєкту

Фази ЖЦ	Опис
Підготовка	Визначення цілей проєкту, аналіз вимог, планування ресурсів.
Концепція	Розробка концептуального дизайну, структури програмного забезпечення.
Деталізація	Детальне проектування архітектури ПЗ, інтерфейсів користувача.
Розробка	Кодування, тестування модулів, інтеграція компонентів.
Тестування	Тестування ПЗ, виявлення та усунення помилок, оцінка якості продукту.
Впровадження	Розгортання ПЗ у середовищі замовника, навчання користувачів.
Підтримка та	Моніторинг роботи ПЗ, збір зворотного зв'язку,

оцінка	внесення покращень.
--------	---------------------

Планом проєкт передбачено 4 основні віх проєкту:

1. Планування проєкту: 01.04.2024 - 31.05.2024.
2. Розробка проєкту: 01.06.2024 - 30.09.2024.
3. Впровадження: 01.10.2024 - 30.11.2024.
4. Підтримка: 01.01.2025 – 01.12.2024.

Розділімо проєкт розробки програмного забезпечення для оцінки продуктивності працівників на три основних етапи, визначивши ієрархічну структуру робіт для кожного етапу:

1. Етап: Підготовка та планування

На цьому етапі встановлюються основи проєкту, визначаються його цілі та завдання.

Визначення вимог

- Збір існуючих вимог від зацікавлених сторін
- Аналіз ринку та конкурентів

Розробка плану проєкту

- Створення графіка виконання робіт
- Розподіл ресурсів та визначення бюджету

Підготовка технічної документації

- Розробка технічного завдання
- Створення архітектурних схем

2. Етап: Розробка та тестування

Центральний етап проєкту, присвячений безпосередньо створенню програмного продукту та його перевірці.

Програмування

- Розробка алгоритмів та написання коду
- Інтеграція модулів та компонентів системи

Тестування

- Розробка тестових сценаріїв
- Виконання модульного, інтеграційного та приймального тестувань

Оптимізація та налагодження

- Виявлення та усунення помилок
- Оптимізація продуктивності та безпеки програмного забезпечення

3. Етап: Впровадження та підтримка

Останній етап проекту, на якому програмне забезпечення розгортається в середовищі замовника та забезпечується його стале функціонування.

Розгортання

- Підготовка середовища користувача
- Встановлення та налаштування ПЗ

Навчання користувачів

- Розробка навчальних матеріалів та керівництв
- Проведення тренінгів для кінцевих користувачів

Моніторинг та підтримка

- Збір зворотного зв'язку від користувачів
- Внесення необхідних змін та оновлень до ПЗ

Кожен з цих етапів містить чітко визначені завдання, що дозволяють крок за кроком рухатися до успішного завершення проекту.

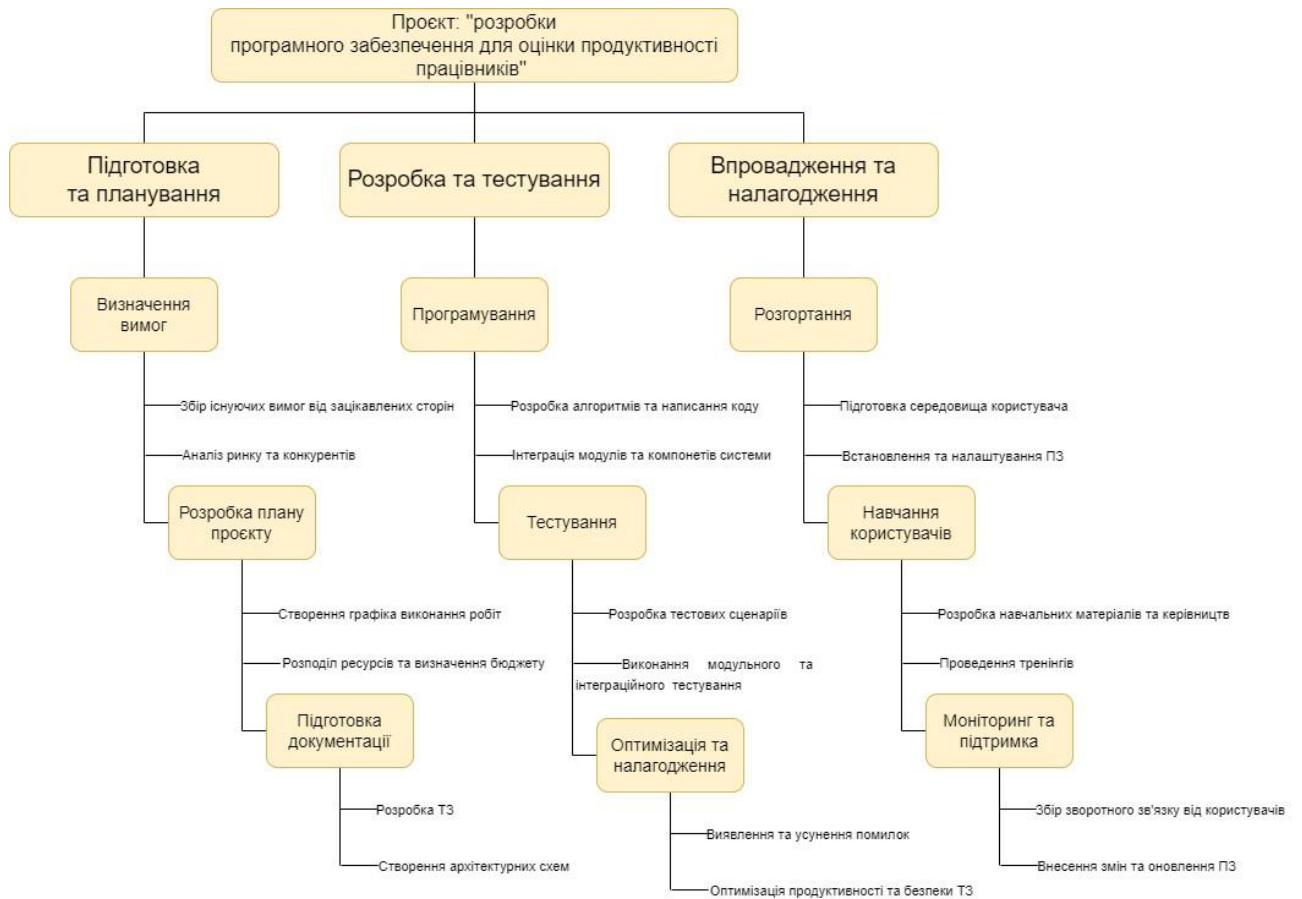


Рис. 2.2. WBS проєкту по фазах життєвого циклу.

2.2 Склад команди проєкту та розподіл відповідальності

Проектна команда для розробки програмного забезпечення для оцінки продуктивності працівників складається з наступних ролей та відповідальностей:

1. Керівник проєкту (Project Manager)

Відповідальність: загальне керівництво проєктом, планування, розподіл ресурсів, комунікація зі стейкхолдерами, вирішення проблем, моніторинг прогресу та звітність.

Деталі: веде документацію проєкту, забезпечує дотримання графіку та бюджету, проводить регулярні зустрічі з командою та стейкхолдерами.

2. Аналітик системи (System Analyst)

Відповідальність: збір та аналіз вимог до ПЗ, розробка технічного завдання, взаємодія з кінцевими користувачами для уточнення їх потреб.

Деталі: створює детальні специфікації вимог, організовує обговорення з

командою розробників для забезпечення точного розуміння вимог.

3. Лідер команди розробників (Development Team Lead)

Відповідальність: координація роботи команди розробників, технічне керівництво проектом, рев'ю коду, забезпечення високої якості та безпеки продукту.

Деталі: відповідає за архітектуру системи, розподіляє завдання між розробниками, надає технічну підтримку команді.

4. Розробники ПЗ (Software Developers)

Відповідальність: написання коду згідно з технічним завданням та специфікаціями, участь у код-рев'ю, тестування модулів.

Деталі: спеціалізація може включати фронтенд, бекенд, бази даних, залежно від структури проекту.

5. QA-інженери (Quality Assurance Engineers)

Відповідальність: планування та виконання тестів для забезпечення якості продукту, ідентифікація та документація помилок, співпраця з розробниками для їх усунення.

Деталі: використовують різні види тестування, включаючи автоматизоване та ручне.

6. Дизайнер інтерфейсів (UI/UX Designer)

Відповідальність: розробка дизайну інтерфейсу користувача, створення прототипів, тестування зручності користування.

Деталі: працює над забезпеченням інтуїтивно зрозумілого та естетичного дизайну продукту, проводить А/В тестування дизайну.

7. Технічна підтримка (Technical Support)

Відповідальність: надання допомоги користувачам після впровадження системи, збір зворотного зв'язку, усунення технічних проблем.

Деталі: організація гарячої лінії для користувачів, робота з звітами про помилки, надання оновлень.

Кожна роль у команді відіграє ключову роль у успішному виконанні проекту, починаючи від підготовчої стадії і закінчуючи постійною підтримкою

та оновленням програмного продукту.

Матриця відповідальності за методикою RACI (Responsible, Accountable, Consulted, Informed) дозволяє чітко визначити роль кожного учасника проекту на різних етапах його реалізації.

- R (Responsible) - особа, відповідальна за виконання роботи.
- A (Accountable) - особа, яка несе кінцеву відповідальність і має право затвердження.
- C (Consulted) - особи, які консультуються під час процесу виконання роботи та чиї думки враховуються.
- I (Informed) - особи, які інформуються про хід виконання та результати.

Ця матриця дозволяє уникнути непорозумінь щодо розподілу відповідальності та забезпечити ефективну комунікацію між усіма учасниками проекту. Кожен учасник чітко розуміє свою роль та відповідальність на різних стадіях реалізації проекту, що сприяє більш гладкому та ефективному процесу розробки та впровадження програмного забезпечення.

Таблиця 2.2

Матриця відповідальності RACI

Учасник и Етапи	Керівник проекту	Аналітик системи	Лідер команди	Розробники ПЗ	QA-	Дизайнер інтерфейсів	Технічна підтримка
Підготов ка	A	R	C	I	I	C	I
Концепці я	A	R	R	C	I	R	I
Деталізац ія	A	R	R	R	C	R	I
Розробка	A	C	A	R	C	C	I
Тестуван	A	C	C	C	I	C	I

ня							
Впровад ження	A	C	R	C	C	C	R
Підтримк а та оцінка	A	C	C	I	C	C	R

2.3 Планування робіт проєкту

Для максимально детального планування робіт проєкту розробки програмного забезпечення для оцінки продуктивності працівників, структуруємо завдання у вигляді таблиці. Таблиця включатиме номер завдання, назву роботи та зазначення попередніх робіт, необхідних для початку кожного наступного етапу.

Таблиця 2.3

Номер	Назва роботи	Попередні роботи
1	Визначення обсягу проєкту	
2	Розробка плану проєкту	1
3	Визначення технічних вимог	1
4	Створення команди проєкту	2
5	Розробка концептуального дизайну	3
6	Погодження концептуального дизайну з замовником	5
7	Детальне проектування архітектури ПЗ	6
8	Розробка алгоритмів	7
9	Написання коду	8
10	Інтеграція модулів	9
11	Розробка тестових сценаріїв	7
12	Тестування модулів	10, 11
13	Виявлення та усунення помилок	12
14	Підготовка середовища користувача	13
15	Розгортання системи	14
16	Навчання користувачів	15
17	Збір зворотного зв'язку	16
18	Внесення покращень	17

Перелік робіт проєкту

Цей план робіт представляє послідовність виконання завдань, від визначення обсягу та плану проєкту до його впровадження та подальшого удосконалення на основі зворотного зв'язку від користувачів. Кожне завдання має чітко визначені попередні роботи, які необхідно виконати, щоб можна було перейти до наступного етапу, забезпечуючи логічний та ефективний прогрес у реалізації проєкту.

Опишемо детально, що відбувається на кожному з робіт проєкту розробки програмного забезпечення для оцінки продуктивності працівників, які ми визначили раніше а також побудуємо діаграму ганта за допомогою застосунку Project Libre.

Старт проєкту запланований на 01.04.2024.

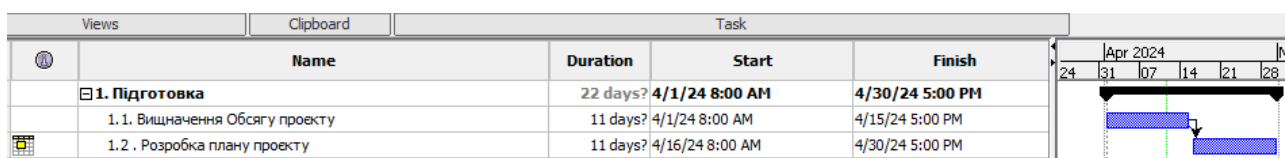
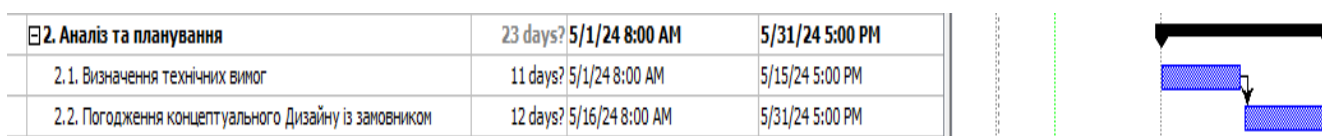


Рис. 2.2. Фрагмент календарного плану. Перший етап проєкту

На цьому етапі команда проєкту аналізує та встановлює основні цілі проєкту, його межі, визначає ключові завдання та вимоги. Здійснюється попереднє планування ресурсів, часові рамки та бюджет. Складається детальний план проєкту, що включає розподіл завдань, встановлення часових рамок для кожного етапу, розподіл ресурсів та розробку стратегії управління



ризиками.

Рис. 2.3. Фрагмент календарного плану. Другий етап проєкту.

На даному етапі аналітик системи збирає детальні технічні вимоги від зацікавлених сторін, включаючи функціональні, нефункціональні вимоги та вимоги до інтерфейсу користувача. Також Формується команда проєкту, яка

включає всі необхідні ролі (розробники, дизайнери, аналітики, тестувальники тощо). Здійснюється розподіл відповідальності та завдань між членами команди.

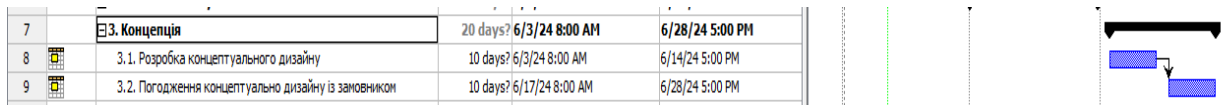


Рис. 2.4. Фрагмент календарного плану. Третій етап проекту

На наступному етапі дизайнери та аналітики працюють над створенням концептуального дизайну програмного забезпечення, що включає загальну структуру, логіку роботи системи та інтерфейсу користувача. Представляється та обговорюється концептуальний дизайн з замовником або ключовими зацікавленими сторонами для отримання зворотного зв'язку та його подальшого узгодження.



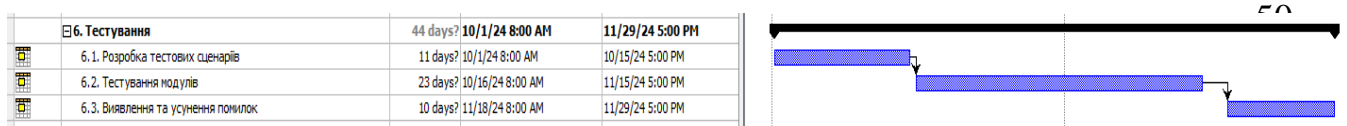
Рис. 2.5. Фрагмент календарного плану. Четвертий етап проекту

Лідер команди розробників та архітектори ПЗ розробляють детальну архітектуру системи, включаючи визначення технологічного стеку, структури баз даних, схеми інтеграції з іншими системами. Розробники працюють над створенням ключових алгоритмів та логіки роботи програмного забезпечення, що лягають в основу функціональності системи.



Рис. 2.6. Фрагмент календарного плану. П'ятий етап проекту

На п'ятому етапі розробники пишуть код програмного забезпечення, використовуючи визначені технології та мови програмування, і здійснюють первинну інтеграцію компонентів системи. Здійснюється інтеграція окремих модулів та компонентів ПЗ в єдину систему, перевірка їх взаємодії та



сумісності.

Рис. 2.7. Фрагмент календарного плану. Шостий етап проекту

QA-інженери розробляють тестові сценарії на основі технічних вимог, щоб забезпечити повне покриття функціоналу тестуванням. Виконується тестування окремих модулів та компонентів системи, щоб виявити та виправити помилки на ранніх етапах розробки. На основі результатів тестування здійснюється виявлення помилок, їх аналіз та усунення розробниками.

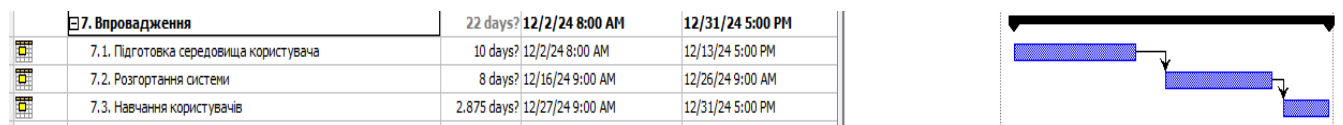


Рис. 2.8. Фрагмент календарного плану. Сьомий етап проекту

ІТ-спеціалісти та адміністратори готують середовище для впровадження ПЗ, включаючи налаштування серверів, баз даних та іншої інфраструктури. Проводиться власне розгортання системи в середовищі замовника, включаючи встановлення ПЗ, його налаштування та інтеграцію з існуючими системами. Організовується навчання кінцевих користувачів, щоб забезпечити їх знайомство з функціоналом та можливостями нової системи.



Рис. 2.9. Фрагмент календарного плану. Фінальний етап проекту

Після впровадження системи збирається зворотний зв'язок від користувачів для оцінки її ефективності та виявлення можливих проблем. На основі зібраного зворотного зв'язку команда проекту вносить необхідні покращення та оновлення до системи, щоб забезпечити її відповідність потребам користувачів та забезпечити її неперервне вдосконалення.

Даний проєкт, що починається 1 квітня 2024 року та завершується 31 січня 2025 року, триватиме приблизно 10 місяців. Це означає, що від початку до кінця проєкту пройде 305 днів. Ця тривалість дає команді проєкту достатньо

часу для планування, виконання та оцінки всіх запланованих завдань і етапів, що включають розробку, тестування та впровадження програмного

	📌	Name	Duration	Start	Finish	Predecessors
1		☑️ 1. Підготовка	22 days?	4/1/24 8:00 AM	4/30/24 5:00 PM	
2		1.1. Вищначення Обсягу проєкту	11 days?	4/1/24 8:00 AM	4/15/24 5:00 PM	
3		1.2. Розробка плану проєкту	11 days?	4/16/24 8:00 AM	4/30/24 5:00 PM	2
4		☑️ 2. Аналіз та планування	23 days?	5/1/24 8:00 AM	5/31/24 5:00 PM	
5		2.1. Вищначення технічних вимог	11 days?	5/1/24 8:00 AM	5/15/24 5:00 PM	3
6		2.2. Погодження концептуального Дизайну із замовником	12 days?	5/16/24 8:00 AM	5/31/24 5:00 PM	5
7		☑️ 3. Концепція	20 days?	6/3/24 8:00 AM	6/28/24 5:00 PM	
8		3.1. Розробка концептуального дизайну	10 days?	6/3/24 8:00 AM	6/14/24 5:00 PM	6
9		3.2. Погодження концептуально дизайну із замовником	10 days?	6/17/24 8:00 AM	6/28/24 5:00 PM	8
10		☑️ 4. Деталізація	34 days?	7/1/24 8:00 AM	8/15/24 5:00 PM	
11		4.1. Детальне проєктування архітектури ПЗ	23 days?	7/1/24 8:00 AM	7/31/24 5:00 PM	9
12		4.2. Розробка алгоритмів	11 days?	8/1/24 8:00 AM	8/15/24 5:00 PM	11
13		☑️ 5. Розробка	32.125 day...	8/16/24 8:00 AM	10/1/24 9:00 AM	
14		5.1. Написання коду	32 days?	8/16/24 8:00 AM	9/30/24 5:00 PM	12
15		5.2. Інтеграція модулів	21 days?	9/2/24 9:00 AM	10/1/24 9:00 AM	
16		☑️ 6. Тестування	44 days?	10/1/24 8:00 AM	11/29/24 5:00 PM	
17		6.1. Розробка тестових сценаріїв	11 days?	10/1/24 8:00 AM	10/15/24 5:00 PM	14
18		6.2. Тестування модулів	23 days?	10/16/24 8:00 AM	11/15/24 5:00 PM	17
19		6.3. Виявлення та усунення помилок	10 days?	11/18/24 8:00 AM	11/29/24 5:00 PM	18
20		☑️ 7. Впровадження	22 days?	12/2/24 8:00 AM	12/31/24 5:00 PM	
21		7.1. Підготовка середовища користувача	10 days?	12/2/24 8:00 AM	12/13/24 5:00 PM	19
22		7.2. Розгортання системи	8 days?	12/16/24 9:00 AM	12/26/24 9:00 AM	21
23		7.3. Навчання користувачів	2.875 days?	12/27/24 9:00 AM	12/31/24 5:00 PM	22
24		☑️ 8. Підтримка та оцінка	22 days?	1/1/25 9:00 AM	1/31/25 9:00 AM	
25		8.1. Збір зворотного зв'язку	10 days?	1/1/25 9:00 AM	1/15/25 9:00 AM	23
26		8.2. Внесення покращень	11 days?	1/16/25 9:00 AM	1/31/25 9:00 AM	25

забезпечення, а також подальшу підтримку та внесення покращень на основі зворотного зв'язку від користувачів.

Рис. 2.10. Загальний вигляд календарного клану проєкту

Діаграма Ганта є незамінним інструментом у сфері управління проєктами, який надає візуальне уявлення про всі заплановані завдання та активності в рамках проєкту протягом певного часового проміжку. Вона допомагає визначити структуру проєкту, включаючи ключові етапи, завдання та підзавдання, що є необхідними для досягнення його цілей. Ця діаграма чітко показує послідовність виконання завдань, їхні взаємозв'язки та залежності, дозволяючи планувати роботу таким чином, щоб ефективно використовувати час та ресурси.

Однією з ключових переваг діаграми Ганта є можливість відстеження часових рамок кожного елемента проєкту. Вона відображає плановані дати початку та завершення кожного завдання, а також дозволяє оцінити їхню тривалість. Крім того, діаграма може надати інформацію про розподіл ресурсів, вказуючи, хто відповідає за виконання кожного завдання, що сприяє кращій

організації роботи команди.

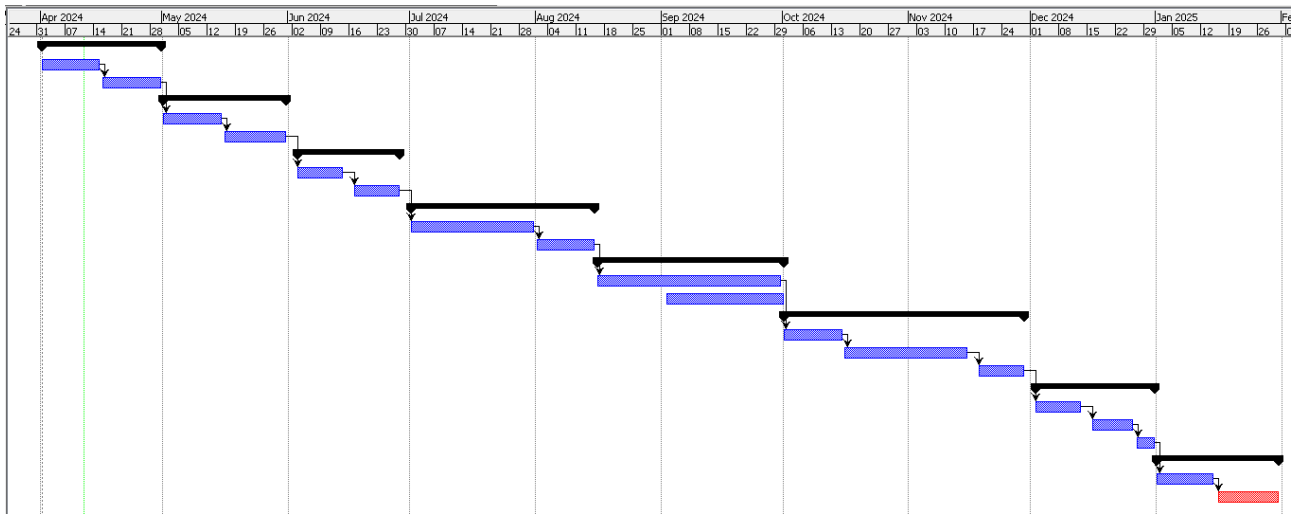


Рис. 2.11. Загальний вигляд діаграми Ганта

Важливим аспектом є також можливість порівняння запланованого прогресу з фактичним, що дозволяє виявити затримки або випередження графіка та вжити відповідних заходів для корекції. Визначення критичного шляху проєкту допомагає зосередити увагу на найважливіших завданнях, від яких залежать загальні терміни реалізації проєкту. Також діаграма Ганта відображає, як зміни в термінах одного завдання можуть вплинути на інші завдання та загальний графік проєкту.

В цілому, діаграма Ганта є ефективним засобом для планування, моніторингу та звітності, який сприяє ефективному управлінню проєктом. Вона дозволяє командам, керівникам проєктів та зацікавленим сторонам тримати проєкт під контролем, адаптуватися до змін та вирішувати проблеми на шляху до його реалізації.

2.4 Планування ресурсів проєкту

Для успішної реалізації проєкту розробки програмного забезпечення для оцінки продуктивності працівників знадобляться наступні трудові та матеріальні ресурси:

Трудові ресурси:

1. Керівник проєкту - для загального керівництва, планування, моніторингу прогресу та зв'язку зі стейкхолдерами.
2. Аналітик системи - для збору та аналізу вимог, розробки специфікацій.
3. Лідер команди розробників - для координації роботи розробників та технічного лідерства.
4. Розробники ПЗ (Frontend, Backend) - для кодування та розробки програмного забезпечення.
5. QA-інженери - для планування та виконання тестування, забезпечення якості продукту.
6. Дизайнер інтерфейсів (UI/UX) - для розробки дизайну користувацького інтерфейсу.
7. Технічна підтримка - для навчання користувачів, підтримки після впровадження.

Матеріальні ресурси:

1. Комп'ютерне обладнання - потужні робочі станції для розробників, тестувальників та дизайнерів.
2. Серверне обладнання - для розгортання та тестування програмного забезпечення, баз даних.
3. Ліцензійне програмне забезпечення - IDE для розробки, системи управління базами даних, інструменти для проектування та тестування.
4. Хмарні сервіси - для хостингу, зберігання даних, резервного копіювання, а також для використання додаткових обчислювальних ресурсів.
5. Офісне приміщення - простір для роботи команди, зустрічей, навчань.
6. Канцелярські товари та офісне обладнання - для підтримки щоденних офісних потреб команди.
7. Комунікаційні засоби - мобільні телефони, системи відеоконференцій для забезпечення зв'язку між членами команди та зі стейкхолдерами.

Ці ресурси забезпечують основу для розробки, тестування, впровадження

та підтримки програмного продукту, а також сприяють ефективній роботі

Name	RBS	Type	E-mail Address	Material Label	Initials	Group	Max. Units	Standard Rate
Керівник проекту		Work			K		100%	\$2000.00/hour
Аналітик системи		Work			A		200%	\$1800.00/hour
Лідер команди розробки		Work			L		100%	\$2000.00/hour
Розробник ПЗ		Work			P		600%	\$1600.00/hour
QA Інженер		Work			Q		300%	\$1300.00/hour
UX/UI дизайнери		Work			U		200%	\$1600.00/hour
Тех. Підтримка		Work			T		200%	\$0.00/hour
Комп'ютерне обладнання		Material			K			\$1000.00
Серверне обладнання		Material			C			\$4000.00
Ліцензійне ПЗ		Material			L			\$500.00
Хмарні сервіси		Material			X			\$300.00
Офісне приміщення		Material			O			\$900.00
Офісне обладнання		Material			O			\$250.00
Комунікаційні засоби		Material			K			\$200.00

команди та взаємодії зі стейкхолдерами.

Рис. 2.12. Лист ресурсів проекту

Також ресурси були розподілені по задачах за допомогою можливостей Project Libre.

ID	Name	Duration	Start	Finish	Predecessors	Resource Names
1	Б1. Підготовка	22 days?	4/1/24 8:00 AM	4/30/24 5:00 PM		
2	1.1. Вищачення Обсягу проекту	8.25 days?	4/1/24 8:00 AM	4/11/24 10:00 AM		Керівник проекту;Аналітик системи;Комунікаційні засоби
3	1.2. Розробка плану проекту	11 days?	4/16/24 8:00 AM	4/30/24 5:00 PM	2	Керівник проекту;Офісне обладнання
4	Б2. Аналіз та планування	23 days?	5/1/24 8:00 AM	5/31/24 5:00 PM		
5	2.1. Визначення технічних вимог	11 days?	5/1/24 8:00 AM	5/15/24 5:00 PM	3	Аналітик системи[200%]
6	2.2. Створення команди проекту	12 days?	5/16/24 8:00 AM	5/31/24 5:00 PM	5	Керівник проекту[50%]
7	Б3. Концепція	20 days?	6/3/24 8:00 AM	6/28/24 5:00 PM		
8	3.1. Розробка концептуального дизайну	10 days?	6/3/24 8:00 AM	6/14/24 5:00 PM	6	UX/UI дизайнери[200%];Комп'ютерне обладнання[2]
9	3.2. Погодження концептуально дизайну із замовником	10 days?	6/17/24 8:00 AM	6/28/24 5:00 PM	8	Керівник проекту
10	Б4. Деталізація	34 days?	7/1/24 8:00 AM	8/15/24 5:00 PM		
11	4.1. Детальне проектування архітектури ПЗ	4.6 days?	7/1/24 8:00 AM	7/5/24 1:48 PM	9	Лідер команди розробки;Розробник ПЗ[400%];Серверне обладнання;Ліцензійне ПЗ
12	4.2. Розробка алгоритмів	11 days?	8/1/24 8:00 AM	8/15/24 5:00 PM	11	
13	Б5. Розробка	32.125 days?	8/16/24 8:00 AM	10/1/24 9:00 AM		
14	5.1. Написання коду	32 days?	8/16/24 8:00 AM	9/30/24 5:00 PM	12	Розробник ПЗ[600%];Серверне обладнання;Ліцензійне ПЗ
15	5.2. Інтеграція модулів	21 days?	9/2/24 9:00 AM	10/1/24 9:00 AM		
16	Б6. Тестування	44 days?	10/1/24 8:00 AM	11/29/24 5:00 PM		QA Інженер[300%];Хмарні сервіси
17	6.1. Розробка тестових сценаріїв	11 days?	10/1/24 8:00 AM	10/15/24 5:00 PM	14	
18	6.2. Тестування модулів	23 days?	10/16/24 8:00 AM	11/15/24 5:00 PM	17	
19	6.3. Виявлення та усунення помилок	10 days?	11/18/24 8:00 AM	11/29/24 5:00 PM	18	
20	Б7. Впровадження	22 days?	12/2/24 8:00 AM	12/31/24 5:00 PM		Тех. Підтримка[200%];Офісне приміщення
21	7.1. Підготовка середовища користувача	10 days?	12/2/24 8:00 AM	12/13/24 5:00 PM	19	
22	7.2. Розгортання системи	8 days?	12/16/24 9:00 AM	12/26/24 9:00 AM	21	
23	7.3. Навчання користувачів	2.875 days?	12/27/24 9:00 AM	12/31/24 5:00 PM	22	
24	Б8. Підтримка та оцінка	22 days?	1/1/25 9:00 AM	1/31/25 9:00 AM		Тех. Підтримка
25	8.1. Збір зворотного зв'язку	10 days?	1/1/25 9:00 AM	1/15/25 9:00 AM	23	
26	8.2. Внесення покращень	11 days?	1/16/25 9:00 AM	1/31/25 9:00 AM	25	

Рис. 2.13. Фрагмент календарного плану із розподіленими ресурсами.

Дані зміни також відображені на діаграмі Ганта, нижче продемонстрований кінцевий варіант діаграми Ганта разом із ресурсами.

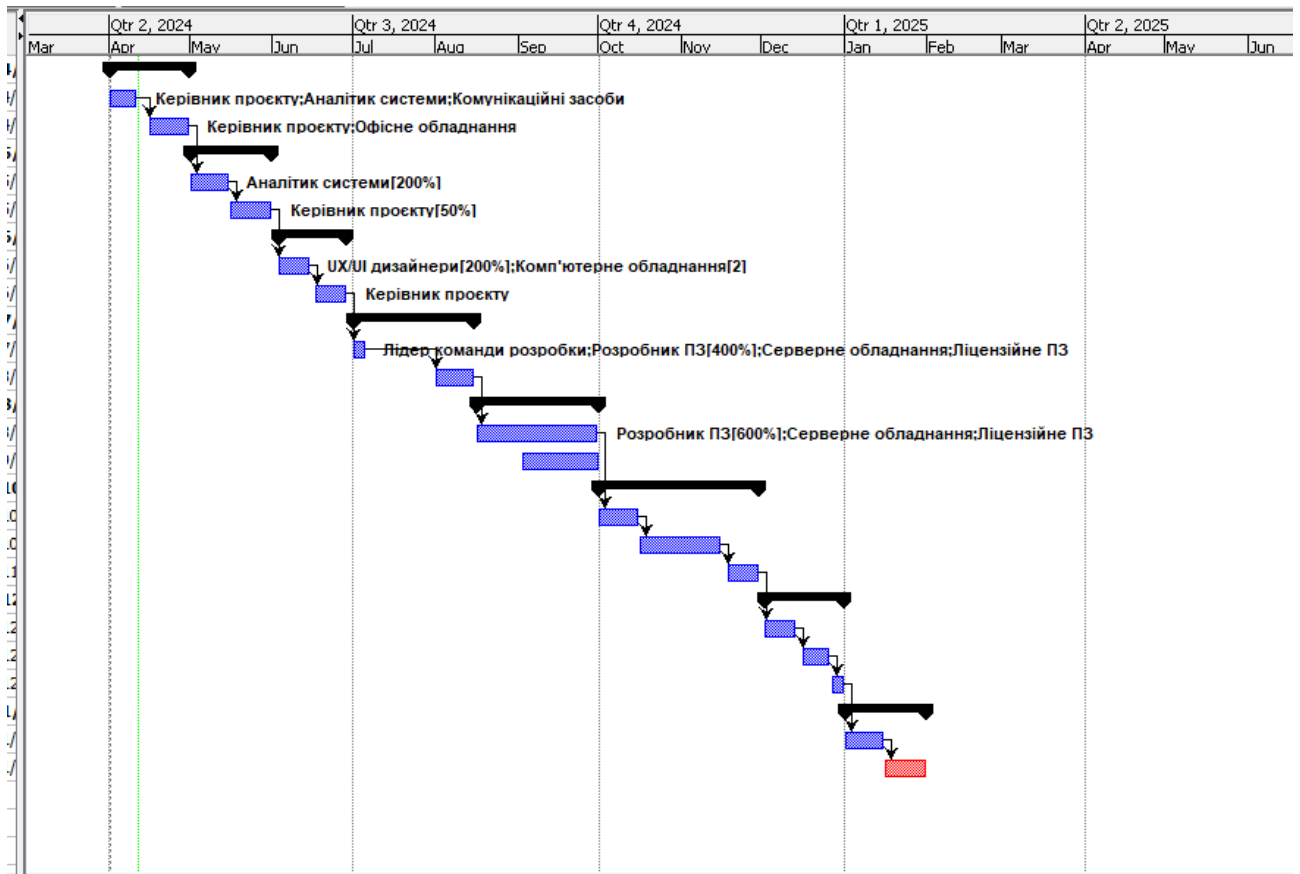
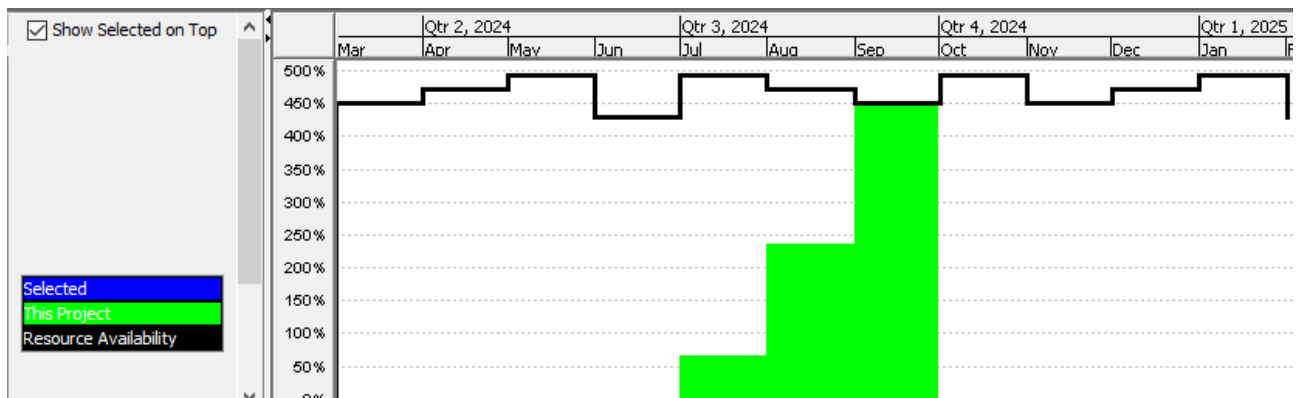


Рис. 2.13. фрагмент діаграми Ганта із розподіленими ресурсами.

Важливим етапом планування є перевірка доступності ресурсів, яку можна



провести наступним чином.

Рис. 2.14. Діаграма доступності ресурсів.

Як видно з рисунку вище, використання ресурсу Розробник ПЗ не перетинає лінію графіку доступності цього ресурсу. Відповідно ресурс не є перевантаженим.

Для того щоб перевірити використання кожного ресурсу, необхідно

відкрити відповідну таблицю. Таблиця використання ресурсів надає чітке розуміння ступеню використання кожного ресурсу, його приналежність до

Name	Work	Work Contour	Assignmen...	Leveling D...	Cost Rate Table
Керівник проекту	238 hours				
1.1. Вищачення Обсягу проекту	22 hours Flat		0 days	0 days Rate A	
1.2. Розробка плану проекту	88 hours Flat		0 days	0 days Rate A	
2.2. Створення команди проекту	48 hours Flat		0 days	0 days Rate A	
3.2. Погодження концептуально дизайну із замовником	80 hours Flat		0 days	0 days Rate A	
Аналітик системи	242 hours				
1.1. Вищачення Обсягу проекту	66 hours Flat		0 days	0 days Rate A	
2.1. Вищачення технічних вимог	176 hours Flat		0 days	0 days Rate A	
Лідер команди розробки	36.8 hours				
4.1. Детальне проектування архітектури ПЗ	36.8 hours Flat		0 days	0 days Rate A	
Розробник ПЗ	1,683.2 hours				
4.1. Детальне проектування архітектури ПЗ	147.2 hours Flat		0 days	0 days Rate A	
5.1. Написання коду	1,536 hours Flat		0 days	0 days Rate A	
QA Інженер	24 hours				
6. Тестування	24 hours Flat		0 days	0 days Rate A	
UX/UI дизайнери	160 hours				
3.1. Розробка концептуального дизайну	160 hours Flat		0 days	0 days Rate A	
Тех. Підтримка	16 hours				
7. Впровадження	8 hours Flat		0 days	0 days Rate A	
8. Підтримка та оцінка	8 hours Flat		0 days	0 days Rate A	
Комп'ютерне обладнання	0 hours				
3.1. Розробка концептуального дизайну	2 Flat		0 days	0 days Rate A	
Серверне обладнання	0 hours				
4.1. Детальне проектування архітектури ПЗ	1 Flat		0 days	0 days Rate A	
5.1. Написання коду	1 Flat		0 days	0 days Rate A	
Ліцензійне ПЗ	0 hours				
4.1. Детальне проектування архітектури ПЗ	1 Flat		0 days	0 days Rate A	
5.1. Написання коду	1 Flat		0 days	0 days Rate A	
Хмарні сервіси	0 hours				

робіт проекту та кількість годин запланованої роботи.

Рис. 2.15. фрагмент таблиці використаних ресурсів.

2.5 Планування вартості проекту

Для планування вартості проекту нам потрібно визначити кількість годин роботи кожного ресурсу. Використаємо приблизну кількість годин на місяць для кожного типу ресурсу в залежності від їхнього відсотка залучення, як описано раніше. Загальна кількість годин роботи за місяць припускається бути 160 годин (при 8-годинному робочому дні та 20 робочих днях на місяць).

Дані для розрахунків:

- Керівник проекту: 60,000 грн/міс, 100% залученість
- Аналітик системи: 50,000 грн/міс, 100% залученість
- Лідер команди розробників: 55,000 грн/міс, 75% залученість
- Розробники ПЗ: 45,000 грн/міс, 75% залученість (6 осіб)
- QA-інженери: 40,000 грн/міс, 100% залученість (3 особи)

- Дизайнер інтерфейсів: 45,000 грн/міс, 100% залученість (2 особи)
- Технічна підтримка: 35,000 грн/міс, 50% залученість (2 особи)

Розрахунок вартості на годину:

Таблиця 2.4

Таблиця вартості трудових ресурсів

Номер	Назва ресурсу	Кількість людей	Вартість на годину (грн)	Кількість годин	Загальна вартість (грн)
1	Керівник проєкту	1	375	160	60,000
2	Аналітик системи	1	313	160	50,000
3	Лідер команди розробників	1	344	120	41,250
4	Розробники ПЗ	6	281	120	202,500
5	QA-інженери	3	250	160	120,000
6	Дизайнер інтерфейсів	2	281	160	90,000
7	Технічна підтримка	2	219	80	35,000

Загальна вартість вказана для всієї кількості людей і відпрацьованих годин упродовж місяця. Ці значення допоможуть забезпечити точне бюджетування і розподіл фінансових ресурсів проєкту.

Для аналізу витрат на матеріальні ресурси проєкту, спершу визначимо їх вартість та передбачувану кількість, яка знадобиться на проєкт. Наведемо прикладну таблицю з відповідними значеннями.

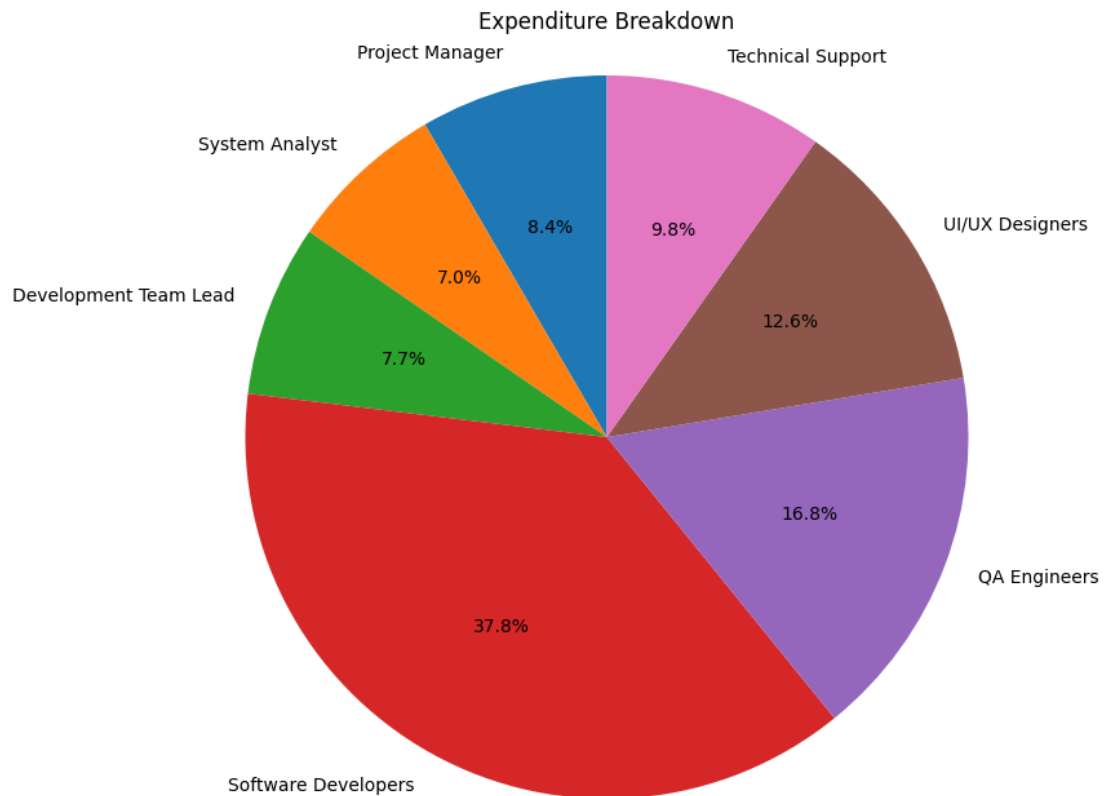


Рис. 2.16. Діаграма витрат на трудові ресурси

Таблиця 2.5

Таблиця вартості матеріальних ресурсів

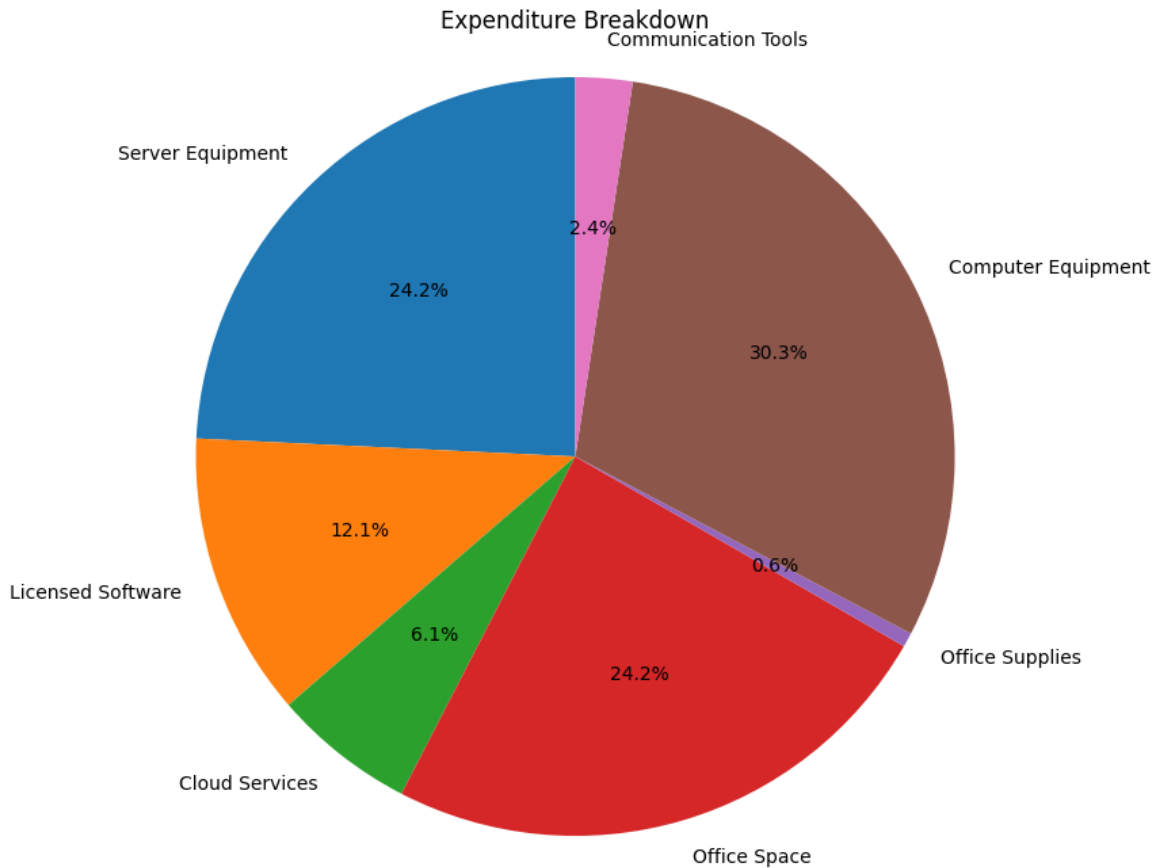
Номер	Назва ресурсу	Кількість одиниць	Вартість за одиницю (грн)	Загальна вартість (грн)
1	Комп'ютерне обладнання	10	25,000	250,000
2	Серверне обладнання	2	100,000	200,000
3	Ліцензійне програмне забезпечення	10	10,000	100,000

4	Хмарні сервіси	(місячна плата)	5,000	5,000 x 10 міс. = 50,000
5	Офісне приміщення	(місячна плата)	20,000	20,000 x 10 міс. = 200,000
6	Канцелярські товари та офісне обладнання	1	5,000	5,000
7	Комунікаційні засоби	10	2,000	20,000

Пояснення:

- Комп'ютерне обладнання: необхідно для роботи розробників, дизайнерів та інших співробітників. Припускається купівля 10 одиниць.
- Серверне обладнання: використовується для розміщення серверів, додатків і баз даних. Потрібно дві машини для редундантності та високої доступності.
- Ліцензійне програмне забезпечення: передбачається купівля ліцензій для розробки та тестування програмного продукту.
- Хмарні сервіси: щомісячна оплата за хмарні обчислювальні ресурси і зберігання даних протягом 10 місяців проєкту.
- Офісне приміщення: вартість оренди офісу на 10 місяців.
- Канцелярські товари та офісне обладнання: включає в себе придбання меблів, канцтоварів та іншого офісного обладнання.

- Комунікаційні засоби: передбачає купівлю телефонів або інших засобів



зв'язку для команди проєкту.

Рис. 2.17. Діаграма витрат на матеріальні ресурси

Цей аналіз дозволяє оцінити загальні матеріальні витрати проєкту, що складають 825,000 гривень. Ці ресурси забезпечать належні умови для реалізації проєкту і забезпечать команду необхідним обладнанням, програмним забезпеченням та робочим простором.

Для аналізу адміністративно-господарських витрат, витрат на послуги, витрат на забезпечення команди необхідними ресурсами та непередбачуваних витрат, наведемо прикладну оцінку вартості для кожної категорії. Також наведемо таблицю для кращого візуального представлення даних.

Оцінка витрат:

Адміністративно-господарські витрати:

• Включають витрати на утримання офісу, адміністративний персонал, утиліти (вода, електрика, інтернет), охорону та інше.

• Щомісячна оцінка: 10,000 грн

• Загальна вартість за 10 місяців: 100,000 грн

Витрати на послуги:

• Включають юридичні послуги, консалтинг, бухгалтерські послуги, маркетинг та рекламу.

• Щомісячна оцінка: 15,000 грн

• Загальна вартість за 10 місяців: 150,000 грн

Витрати на забезпечення команди необхідними ресурсами:

• Включають закупівлю програмного забезпечення, ліцензій, спеціалізованого обладнання для розробників, а також тренінги та курси для підвищення кваліфікації.

• Щомісячна оцінка: 20,000 грн

• Загальна вартість за 10 місяців: 200,000 грн

Непередбачувані витрати:

Резерв на непередбачувані витрати, такі як затримки, технічні проблеми, зміни у вимогах проєкту.

Загальна сума резерву: 50,000 грн

Таблиця витрат проєкту

Категорія витрат	Щомісячна оцінка (грн)	Загальна вартість за 10 місяців (грн)
Трудові ресурси		6,150,000
Матеріальні ресурси		825,000
Адміністративно- господарські витрати	10,000	100,000
Витрати на послуги	15,000	150,000
Витрати на забезпечення команди необхідними ресурсами	20,000	200,000
Непередбачувані витрати	-	50,000
Загальна сума		8,825,000

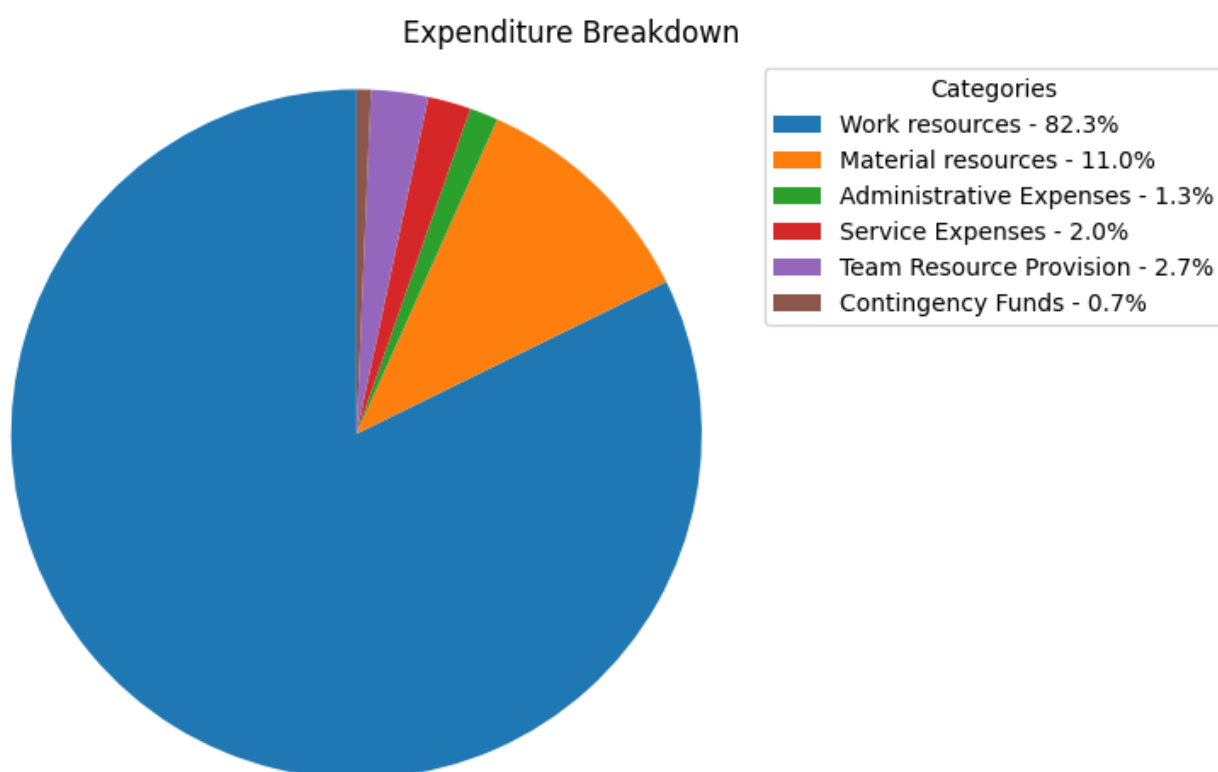


Рис. 2.18. Діаграма витрат проєкту

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ПРОЄКТУ НА СТАДІЇ ВИКОНАННЯ ТА УПРАВЛІННЯ КОМПОНЕНТАМИ

3.1 Управління якістю

Управління якістю є ключовим аспектом успішного виконання будь-якого проєкту, зокрема в розробці програмного забезпечення для оцінки продуктивності працівників. Цей процес включає визначення стандартів якості відповідно до вимог зацікавлених сторін та постійний моніторинг ефективності процесів на кожному етапі проєкту для забезпечення їх відповідності цим стандартам. Основна мета управління якістю полягає в тому, щоб проєкт відповідав очікуванням зацікавлених сторін та стандартам індустрії, забезпечуючи високу задоволеність користувачів кінцевим продуктом.

Вимоги зацікавлених сторін:

1. **Замовники:** основні зацікавлені сторони, які фінансують проєкт. Вимагають високої рентабельності інвестицій і забезпечення, що кінцевий продукт повністю відповідає зазначеним специфікаціям і функціональним вимогам.

2. **Користувачі програмного забезпечення:** потребують інтуїтивно зрозумілого, зручного і надійного продукту, який ефективно допомагає у вимірюванні та аналізі продуктивності працівників.

3. **Команда проєкту:** включає розробників, дизайнерів, тестувальників і аналітиків, які прагнуть чітких та досяжних цілей, ефективних процесів та достатніх ресурсів для виконання своєї роботи.

4. **Регулюючі органи:** вимагають дотримання встановлених законодавчих і нормативних вимог, зокрема у сферах конфіденційності даних та інформаційної безпеки.

5. **Постачальники технологій і сервісів:** цікавляться стабільним партнерством, своєчасною оплатою і чіткими технічними вимогами до продуктів і послуг, які вони постачають.

Стратегії управління якістю:

- Розробка політики якості: Визначення стандартів якості, які відповідають вимогам усіх зацікавлених сторін.
- Планування якості: встановлення процедур і критеріїв для досягнення цих стандартів.
- Забезпечення якості: регулярні перевірки та оцінка робочих процесів на предмет їх відповідності планам і стандартам.
- Контроль якості: проведення тестувань і перевірок кінцевого продукту для забезпечення його відповідності вимогам.

Ці стратегії допоможуть забезпечити, що кінцевий продукт не лише задовольнить функціональні потреби користувачів, але й буде виготовлений згідно із найвищими стандартами якості, забезпечуючи його ефективність і надійність.

Організувати структуру проєкту розробки програмного забезпечення для оцінки продуктивності працівників дозволить створення таблиці формалізації проєкту по модулях, визначивши ключові модулі та їхні функціональності. Така структура сприятиме кращому розумінню завдань, які потрібно виконати на різних стадіях розробки.

Таблиця 3.1

Таблиця формалізації проєкту по модулях

№	Модуль	Опис модуля	Основні завдання
1	Модуль авторизації	Дозволяє користувачам увійти в систему	Розробка системи аутентифікації та безпеки
2	Модуль управління користувачами	Керування профілями користувачів	Додавання, видалення та редагування користувачів
3	Модуль збору даних	Збирає дані про продуктивність працівників	Інтеграція з внутрішніми системами для збору даних

4	Модуль аналітики	Обробляє зібрані дані та виводить аналітику	Розробка алгоритмів обробки та аналізу даних
5	Модуль звітності	Створює звіти за вказаними параметрами	Генерація персоналізованих звітів
6	Модуль сповіщень	Відправляє автоматичні сповіщення користувачам	Розробка системи нагадувань та оповіщень
7	Модуль налаштувань	Дозволяє користувачам налаштовувати систему	Інтерфейс налаштувань для користувачів

Деталізація таблиці:

1. Модуль авторизації: має забезпечувати безпечний вхід у систему, використовуючи сучасні методи аутентифікації.

2. Модуль управління користувачами: організація бази даних користувачів, їх ролей та доступу до різних частин системи.

3. Модуль збору даних: інтеграція з існуючими ІТ-інфраструктурами для автоматичного збору необхідних даних.

4. Модуль аналітики: використання статистичних та інших алгоритмів для обробки вхідних даних та виведення корисної інформації.

5. Модуль звітності: динамічне створення звітів на основі даних, доступних у системі.

6. Модуль сповіщень: автоматизація процесу нагадування про важливі події або крайні терміни.

7. Модуль налаштувань: надання користувачам можливості налаштовувати аспекти системи для особистих потреб.

Ця структуризація по модулях дозволяє чітко визначити відповідальності в рамках команди розробників, спрощує управління проектом та підвищує його ефективність.

Для створення вимог до продукту за кожним модулем проекту програмного забезпечення для оцінки продуктивності працівників, ми розробимо детальний опис функціональних і нефункціональних вимог до кожного з модулів. Це допоможе забезпечити чітке розуміння того, що система повинна робити, а також які стандарти якості та безпеки вона має відповідати.

Вимоги до продукту по кожному модулю:

1. Модуль авторизації:

- Функціональні вимоги:
 - Аутентифікація користувачів через логін і пароль.
 - Підтримка двофакторної аутентифікації.
 - Відновлення забутих паролів через електронну пошту.
 - Обмеження спроб входу після невдалих спроб аутентифікації.
- Нефункціональні вимоги:
 - Висока безпека зберігання та обробки користувацьких даних.
 - Швидкість відгуку системи на запити авторизації не більше 1 секунди.

2. Модуль управління користувачами:

- Функціональні вимоги:
 - Створення, редагування та видалення профілів користувачів.
 - Надання різних рівнів доступу користувачам в залежності від їх ролей.
 - Відстеження активності користувачів в системі.
- Нефункціональні вимоги:
 - Інтуїтивно зрозумілий користувацький інтерфейс.
 - Скальованість для підтримки великої кількості користувачів.

3. Модуль збору даних:

- Функціональні вимоги:
 - Автоматичний збір даних з різних внутрішніх джерел.
 - Зберігання даних в зашифрованому вигляді.
 - Підтримка регулярних оновлень даних без перерви в роботі системи.
- Нефункціональні вимоги:
 - Надійність та точність збору даних.
 - Висока доступність та ві дна архітектура для запобігання втрати даних.

4. Модуль аналітики:

- Функціональні вимоги:
 - Обробка зібраних даних для генерації аналітичних звітів.
 - Підтримка різних типів аналітики, включаючи реальний час та історичну аналітику.
 - Можливість налаштування параметрів аналітики користувачами.
- Нефункціональні вимоги:
 - Висока швидкість обробки даних.
 - Точність обчислень, мінімізація помилок у звітах.

5. Модуль звітності:

- Функціональні вимоги:
 - Генерація звітів на основі введених користувачем критеріїв.
 - Можливість експорту звітів у різних форматах, таких як PDF, Excel.
 - Збереження історії генерованих звітів для подальшого доступу.
- Нефункціональні вимоги:
 - Безпека доступу до звітів, обмеження доступу в залежності від ролі користувача.
 - Зручність та інтуїтивність інтерфейсу.

6. Модуль сповіщень:

- Функціональні вимоги:
 - Автоматична відправка сповіщень користувачам за визначеними

критеріями.

- Конфігурація типів сповіщень, які можуть включати електронні листи, SMS, або push-повідомлення.
- Відстеження статусу відправлення та доставки сповіщень.
- Нефункціональні вимоги:
 - Надійність системи сповіщень для гарантування своєчасної доставки.
 - Скальованість для обробки великої кількості одночасних сповіщень.

7. Модуль налаштувань:

- Функціональні вимоги:
 - Дозволяє користувачам змінювати параметри системи відповідно до особистих переваг.
 - Запам'ятовування індивідуальних налаштувань для кожного користувача.
 - Можливість налаштування інтерфейсу, включаючи теми, мову інтерфейсу тощо.
- Нефункціональні вимоги:
 - Гнучкість і простота зміни налаштувань.
 - Збереження налаштувань на сервері для доступності з будь-якого пристрою.

Ці вимоги до кожного модулю забезпечують зрозумілі критерії для розробки та тестування, гарантуючи, що кінцевий продукт задовольнить потреби всіх зацікавлених сторін і відповідатиме високим стандартам якості.

Для того, щоб зазначити пріоритети вимог, які відображають важливість цих вимог для різних зацікавлених сторін, а також для окремих модулів продукту, було створено таблицю "Вимоги до якості 'Зацікавлені сторони - продукт'"

Таблиця: Вимоги до якості "Зацікавлені сторони - продукт"

Вимоги до зацікавлених сторін	Вимоги до якості продукту	Пріоритет по модулях продукту	Модуль продукту	Вимоги до якості продукту	
Високий	Користувачі	Інтуїтивно зрозумілий інтерфейс	Високий	Модуль авторизації	Швидка та безпечна авторизація
Високий	Замовники	Надійність та стабільність	Високий	Модуль збору даних	Точний збір даних, висока доступність
Середній	Регулюючі органи	Дотримання законодавства	Середній	Модуль аналітики	Захист даних, анонімізація інформації
Низький	Постачальники	Сумісність із зовнішніми системами	Низький	Модуль сповіщень	Інтеграція з електронними поштовими системами
Високий	Керівництво компанії	Рентабельність інвестицій	Високий	Модуль звітності	Генерація звітів з точними метриками
Середній	Технічна підтримка	Легкість в обслуговуванні	Середній	Модуль налаштувань	Гнучкість налаштувань

Пояснення таблиці:

- Пріоритет по зацікавленим сторонам: вказує на важливість певної вимоги для зацікавлених сторін, відображає, як критична вимога є для задоволення їх потреб.
- Зацікавлені сторони: це особи або групи, які мають інтерес до проекту та його результатів.
- Вимоги до якості продукту: опис того, що потрібно зробити для забезпечення задоволення потреб зацікавлених сторін.
- Пріоритет по модулях продукту: вказує на важливість вимоги у контексті конкретного модулю продукту.
- Модуль продукту: окремі компоненти або частини програмного забезпечення, кожен з яких виконує певні функції.
- Вимоги до якості продукту (у контексті модуля): специфічні вимоги якості, які мають бути виконані в рамках даного модулю.

Дана таблиця дозволяє систематизувати управління якістю проєкту, забезпечуючи збалансоване врахування потреб усіх зацікавлених сторін та адаптацію до конкретних аспектів проєкту.

Далі була створена таблиця пріоритетності вимог. Вона оформлена так, щоб кожна вимога була віднесена до відповідної групи зацікавлених сторін з урахуванням її значущості для цих сторін. Це допоможе оцінити, які вимоги є найкритичнішими для задоволення потреб зацікавлених сторін.

Таблиця 3.3

Таблиця пріоритетності вимог за зацікавленими сторонами

Зацікавлені сторони	Вимога	Пріоритет
Користувачі	Інтуїтивно зрозумілий інтерфейс	Високий
Користувачі	Надійність та стабільність системи	Високий
Замовники	Висока рентабельність інвестицій	Високий
Замовники	Дотримання запланованих бюджетів і термінів	Високий

Керівництво компанії	Забезпечення звітності для внутрішнього аналізу	Середній
Технічна підтримка	Легкість обслуговування і підтримки	Середній
Розробники	Чіткі технічні специфікації	Високий
Регулюючі органи	Відповідність законодавству та нормам безпеки	Високий
Постачальники	Інтеграція з зовнішніми системами	Низький
Постачальники	Своєчасна оплата та чіткі контракти	Середній
Зацікавлені сторони	Вимога	Пріоритет

Пояснення таблиці:

- Зацікавлені сторони: Хто є ключовим бенефіціаром або має інтерес у виконанні вимоги.
- Вимога: Конкретне очікування або стандарт, який повинен бути задоволений у проекті.
- Пріоритет: Вказує на важливість вимоги для проекту та зацікавлених сторін. Варіюється від "Високий" до "Низький", де "Високий" означає, що невиконання цієї вимоги може мати серйозні наслідки.

Ця таблиця дозволяє проектному менеджеру і команді зосередитись на критичних аспектах проекту, забезпечуючи відповідність роботи потребам ключових зацікавлених сторін і оптимізуючи ресурси для досягнення найважливіших цілей.

Наступним кроком буде таблиці пріоритетності вимог до продукту за модулями. Таблиця пріоритетності вимог до продукту за модулями важлива для ефективного управління проектом, оскільки допомагає визначати ключові вимоги, оптимізувати розподіл ресурсів, управляти очікуваннями зацікавлених сторін і мінімізувати ризики. Цей інструмент забезпечує зосередження на найважливіших аспектах проекту, сприяє гнучкості під час змін і допомагає підтримувати стратегічне планування впродовж усієї розробки.

Таблиця пріоритетності вимог до продукту за модулями

№	Модуль продукту	Вимога до якості продукту	Пріоритет
1	Модуль авторизації	Швидкість відгуку системи	Високий
2	Модуль авторизації	Забезпечення безпеки даних користувачів	Високий
3	Модуль управління користувачами	Надійне управління профілями	Високий
4	Модуль збору даних	Точність збору та обробки даних	Високий
5	Модуль аналітики	Точність аналітичних обчислень	Високий
6	Модуль звітності	Генерація звітів з точними метриками	Високий
7	Модуль сповіщень	Надійність системи сповіщень	Середній
8	Модуль налаштувань	Гнучкість налаштувань, простота управління	Середній

Така структурована таблиця дозволяє розробникам та тестувальникам зосередитись на ключових аспектах кожного модуля, що в свою чергу сприятиме створенню високоякісного продукту, який відповідатиме всім вимогам та очікуванням зацікавлених сторін.

Створення таблиці "Контроль якості проєкту" допоможе визначити які заходи необхідно вжити для задоволення вимог зацікавлених сторін щодо кожного модулю продукту. Така таблиця допомагає систематизувати процес контролю якості і забезпечує чіткість у виконанні проєктних цілей. Таблиця контролю якості проєкту використовується для систематичного моніторингу та оцінювання відповідності продукту встановленим стандартам та вимогам, що допомагає забезпечити високу якість результату та виявити потенційні проблеми на ранніх стадіях їх розвитку.

Контроль якості проєкту

Зацікавлені сторони	Модуль продукту	Вимоги до якості продукту	Заходи необхідні для задоволення вимог
Користувачі	Модуль авторизації	Забезпечення безпеки даних користувачів	Реалізація шифрування даних, двофакторна аутентифікація
Користувачі	Модуль інтерфейсу	Інтуїтивно зрозумілий інтерфейс	UX/UI тестування, залучення UX консультантів
Замовники	Модуль звітності	Точність звітів	Автоматичні тести точності даних, ревізія коду
Технічна підтримка	Модуль налаштувань	Простота управління налаштуваннями	Інтерактивні воркшопи для кінцевих користувачів
Регулюючі органи	Всі модулі	Відповідність законодавству та нормам безпеки	Періодичні правові аудити, впровадження GDPR
Розробники	Модуль збору даних	Надійність збору даних	Імплементация фейловер та бекап систем
Керівництво компанії	Модуль аналітики	Стабільність та швидкість обробки даних	Стрес-тестування, оптимізація запитів до баз даних

Опис таблиці:

Зацікавлені сторони: це ті, хто впливає на проект або відчуває вплив від його результатів. Вони встановлюють критерії та стандарти для різних аспектів продукту.

- Модуль продукту: конкретний компонент системи, до якого відносяться специфічні вимоги.
- Вимоги до якості продукту: деталізований опис того, що система повинна виконувати або яким чином повинна поводитися, щоб задовольнити потреби зацікавлених сторін.
- Заходи необхідні для задоволення вимог: конкретні дії або процеси, які мають бути реалізовані для досягнення вимог.

Така таблиця стане фундаментом для планування та виконання процесів контролю якості в проекті, забезпечуючи систематичний підхід до досягнення якості на всіх рівнях системи.

Наступним кроком було створення таблиці бюджету на забезпечення якості

Створення таблиці "Бюджет забезпечення якості" дозволить систематизувати та визначити фінансові ресурси, необхідні для впровадження та забезпечення якості кожного модуля продукту. Вона допоможе планувати бюджети для різних заходів, спрямованих на виконання вимог до якості, відповідно до потреб зацікавлених сторін.

Таблиця 3.6

Бюджет забезпечення якості

Зацікавлені сторони	Модуль продукту	Вимоги до якості	Заходи для задоволення вимог	Бюджет	Бюджет інкриментально
Користувачі	Модуль авторизації	Безпека даних	Імплементация SSL, двофакторна аутентифікація	\$20,000	\$20,000
Користувачі	Модуль	Інтуїтивно	UX дизайн,	\$15,000	\$5,000

	інтерфейс у	зрозумілий інтерфейс	тестування з користувачами		
--	----------------	-------------------------	-------------------------------	--	--

Продовження таблиці 3.6

Замовники	Модуль звітності	Точність звітів	Розробка звітних модулів, перевірка даних	\$25,000	\$10,000
Технічна підтримка	Модуль налаштувань	Легкість управління	Створення документації та навчальних матеріалів	\$5,000	\$1,000
Регулюючі органи	Всі модулі	Відповідність нормативам	Юридичний аудит, впровадження відповідності	\$30,000	\$10,000
Розробники	Модуль збору даних	Надійність збору даних	Розробка системи резервного копіювання, тестування	\$18,000	\$3,000
Керівництво компанії	Модуль аналітики	Висока продуктивність обробки даних	Оптимізація процесів, використання потужного обладнання	\$40,000	\$15,000

Опис таблиці:

- Зацікавлені сторони: особи або організації, чий інтерес впливає на проект або які відчувають вплив від його результатів.
- Модуль продукту: специфічний компонент або частина продукту, що має власні вимоги до якості.
- Вимоги до якості: детальний опис очікувань зацікавлених сторін щодо

якості конкретного модуля.

- Заходи для задоволення вимог: конкретні дії або технічні рішення, що необхідні для виконання вимог до якості.
- Бюджет: попередньо розрахована вартість виконання заходів для задоволення вимог.
- Бюджет інкриментально: додаткові кошти, що можуть бути потрібні для впровадження додаткових заходів або для покриття непередбачених витрат.

Ця таблиця дозволяє не тільки планувати витрати, але й визначати потенційні фінансові ризики, а також забезпечувати адекватне фінансування для всіх аспектів забезпечення якості проекту.

3.2 Управління ризиками

Управління ризиками є критично важливим аспектом управління будь-яким проектом, особливо коли мова йде про розробку програмного забезпечення для оцінки продуктивності працівників. Цей процес дозволяє проектній команді ідентифікувати, оцінювати та пріоритезувати ризики, з метою мінімізації можливості негативного впливу на успішність проекту.

Ризики можуть включати технічні, юридичні, фінансові аспекти, а також ризики, пов'язані зі змінами у вимогах зацікавлених сторін або ринкових умовах. Процес управління ризиками включає систематичне виявлення ризиків, їх оцінку з погляду ймовірності та потенційного впливу, планування відповідних заходів реагування, реалізацію цих заходів та моніторинг ефективності здійснених дій.

Основна мета управління ризиками у контексті розробки програмного забезпечення полягає у забезпеченні стабільності процесу розробки та запуску продукту, зниженні можливості затримок та збільшенні шансів на виконання проекту в рамках встановлених бюджетів і термінів.

Для цього проекту, підходи до управління ризиками включатимуть, але не обмежуватимуться, визначенням ключових точок контролю, впровадженням гнучких методик управління проектом, регулярними перевірками і

коригуванням плану проекту, а також залученням всіх зацікавлених сторін у процес ідентифікації та вирішення потенційних проблем.

Такий підхід дозволяє не тільки адекватно реагувати на ризики, але й передбачати їх, тим самим знижуючи загальну невизначеність і забезпечуючи більш високий рівень готовності команди до можливих викликів.

Таблиця 3.7

Ідентифікація ризиків

Тип ризику	Ризикова подія	Сила впливу	Керованість
Технічний	Неспроможність інтегрувати нові модулі	Висока	Середня
	Помилки в коді, що призводять до збоїв	Висока	Середня
	Невідповідність технічної документації	Середня	Висока
Технологічний	Застаріння використовуваної технології	Середня	Висока
	Залежність від одного постачальника технологій	Середня	Середня
Операційний	Втрата даних через системні збої	Висока	Середня
	Порушення процесу розробки	Висока	Середня
Фінансовий	Перевищення бюджету	Висока	Середня
	Зміни у фінансуванні проекту	Висока	Низька
Правовий	Невідповідність нових функцій законодавству	Висока	Низька
	Затримки з ліцензуванням технологій	Середня	Середня
Ринковий	Зміни у вимогах клієнтів	Середня	Висока
	Втрата клієнта	Середня	Середня

Кадровий	Відтік висококваліфікованих спеціалістів	Висока	Середня
	Проблеми з наймом спеціалістів	Середня	Висока

Продовження таблиці 3.7

Управлінський	Недоліки у плануванні проекту	Висока	Висока
	Недостатнє спілкування між командами	Висока	Висока
Вимоги	Нечіткість специфікацій	Висока	Висока
	Часті зміни вимог	Висока	Середня

Для проведення оцінювання ризиків створимо таблицю, яка включає всі ідентифіковані ризикові події та оцінює кожну подію за різними параметрами. Це допоможе виявити, які ризики можуть мати найбільший вплив на проект, з метою прийняття відповідних заходів реагування.

Таблиця 3.8

Шкала оцінювання ризиків

Проста якісна оцінка	Деталізована якісна оцінка	Шифр оцінки	Відповідна квазі-кількісна одиниця
	Відсутній	немає	0
Низький	Низько-низький	нн	1
	Низько-середній	нс	2
	Низько-високий	нв	3
Середній	Середньо-низький	сн	4
	Середньо-середній	сс	5
	Середньо-високий	св	6
Високий	Високо-низький	вн	7
	Високо-середній	вс	8
	Високо-високий	вв	9

	Катастрофічний	к	10
--	----------------	---	----

Таблиця 3.9

Оцінювання ризиків

Ризикова подія	Затримки у часі		Фінансові витрати		Ймовірність		Частота		Важливість
	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	
Неспроможність інтегрувати нові модулі	вс	8	к	8	св	6	нв	3	48
Помилки в кодї, що призводять до збоїв	сс	5	сс	5	вс	8	вн	7	40
Невідповідність технічної документації	нв	3	нс	2	сс	5	нс	2	10
Застаріння використовуваної технології	св	6	вс	7	вн	7	сн	4	49
Залежність від одного постачальника технологій	сн	4	вс	7	сс	5	нв	3	35
Втрата даних через системні збої	вв	9	вв	9	вв	8	нв	3	72
Порушення процесу	сс	6	сс	6	вн	7	вн	7	42

розробки									
Перевищення бюджету	ВН	7	К	10	ВН	7	СС	5	70

Продовження таблиці 3.9

Зміни у фінансуванні проекту	ВС	8	ВВ	9	ВС	8	СС	5	72
Невідповідність нових функцій законодавству	ВС	8	ВВ	9	НВ	3	НС	2	18
Затримки з ліцензуванням технологій	СС	5	СВ	6	СС	5	СН	4	30
Зміни у вимогах клієнтів	ВН	7	ВН	7	ВВ	9	ВС	8	63
Втрата клієнта	ВВ	9	К	10	ВВ	9	НВ	3	90
Відтік висококваліфіко ваних спеціалістів	СВ	6	ВС	8	СС	5	СС	5	40
Проблеми з наймом необхідних спеціалістів	СС	5	ВН	7	СВ	6	СН	4	42
Недоліки у плануванні проекту	ВС	8	ВВ	9	ВС	8	СС	5	72
Недостатнє спілкування між	СВ	6	СВ	6	ВН	7	ВН	7	42

командами									
Нечіткість специфікацій	вс	8	вс	8	вс	8	вс	8	64

Управління ризиками є ключовим елементом будь-якого проекту, особливо коли це стосується розробки складного програмного забезпечення. Ідентифікація ризиків дає можливість не тільки зрозуміти потенційні загрози, але й розробити ефективні стратегії для їх мінімізації або уникнення. Протиризові заходи мають бути сплановані та впроваджені з метою зменшення ймовірності негативного впливу цих ризиків на проект.

Протиризові заходи можуть включати різні стратегії, такі як перерозподіл ресурсів, зміна графіків, використання запасних технологій, створення резервних планів та покращення комунікації між учасниками проекту.

Таблиця 3.10

Таблиця протиризових заходів

Ризикова подія	Протиризові заходи	Відповідальний	Термін впровадження
Неспроможність інтегрувати нові модулі	Підготовка плану інтеграції, тестування інтеграції до розгортання	Відділ розробки	До Q3 2024
Помилки в кодї, що призводять до збоїв	Збільшення обсягу модульного та інтеграційного тестування	QA команда	Постійно
Невідповідність технічної документації	Оновлення та стандартизація технічної	Технічний письменник	До Q2 2024

	документації		
Застаріння використовуваної технології	Впровадження політики регулярного оновлення технологій	ІТ відділ	До Q4 2024

Залежність від одного постачальника технологій	Диверсифікація постачальників	Відділ закупівель	До Q1 2025
Втрата даних через системні збої	Розробка та впровадження комплексної стратегії резервного копіювання	ІТ відділ	До Q3 2024
Перевищення бюджету	Перегляд та оптимізація бюджету, регулярний фінансовий аудит	Фінансовий директор	Постійно
Зміни у фінансуванні проекту	Розробка стратегії фінансового резервування	Фінансовий відділ	До Q2 2024
Невідповідність нових функцій законодавству	Посилення юридичного моніторингу нових розробок	Юридичний відділ	Постійно
Зміни у вимогах клієнтів	Регулярні зустрічі з клієнтами для уточнення вимог	Керівник проекту	Постійно

3.3 Управління зацікавленими сторонами

Управління зацікавленими сторонами є ключовим елементом успішної реалізації проекту розробки програмного забезпечення. Цей процес включає ідентифікацію осіб або груп, які мають прямий або непрямий інтерес до проекту, та розробку стратегій для забезпечення їх підтримки та задоволення їх

потреб та очікувань.

Цілі управління зацікавленими сторонами:

Ідентифікація та аналіз: розпізнавання всіх зацікавлених сторін, аналіз їх впливу та інтересів відносно проекту.

Комунікація: забезпечення ефективного обміну інформацією з зацікавленими сторонами.

Враховання інтересів: планування дій для враховання різноманітних інтересів та потреб зацікавлених сторін у всіх аспектах проекту.

Моніторинг та адаптація: регулярний перегляд впливу проекту на зацікавлені сторони і налаштування стратегії управління за потребою.

Розробка стратегій:

Регулярні зустрічі: проведення зустрічей з ключовими зацікавленими сторонами для обговорення прогресу проекту та збору їх відгуків.

Інформаційні бюлетені: регулярна розсилка оновлень про стан проекту, щоб зацікавлені сторони були поінформовані.

Персоналізовані плани дій: розробка індивідуальних планів дій для важливих зацікавлених сторін, з урахуванням їх специфічних очікувань і потреб.

Таблиця 3.12

Зацікавлені сторони та їх очікування

Зацікавлені сторони	Очікування	Стратегії взаємодії
Клієнти	Висока якість продукту, дотримання термінів	Регулярні звіти про прогрес, демо
Команда проекту	Чітке керівництво, визнання роботи	Тімбілдинги, воркшопи, обговорення
Інвестори	ROI, зростання вартості бізнесу	Звіти про фінансовий стан і ROI
Керівництво компанії	Дотримання бюджету, успішний запуск	Щомісячні стратегічні зустрічі

Регулятори	Дотримання законодавства	Консультації з юридичним відділом
Кінцеві користувачі	Легкість використання, підтримка	Бета-тестування, зворотний зв'язок
Постачальники	Стабільні контракти, своєчасна оплата	Регулярні перегляди контрактів

За допомогою цих стратегій можна ефективно управляти очікуваннями та впливом зацікавлених сторін, що забезпечує більшу підтримку проекту та зменшує ризики, пов'язані з його виконанням.

Оцінити роль та значення кожної зацікавленої сторони в контексті розробки проекту допоможе таблиця "Вплив зацікавлених сторін на проект" (3.13). Така таблиця дозволяє систематично аналізувати, як взаємодії з різними зацікавленими сторонами можуть впливати на хід проекту, а також на його успіх або невдачу.

Таблиця 3.13

Вплив зацікавлених сторін на проект

Зацікавлені сторони	Вплив на проект	Опис впливу
Клієнти	Високий	Клієнти визначають основні вимоги та очікування до продукту, їх задоволення є ключовим для прийняття продукту ринком.
Команда проекту	Високий	Від успішної роботи команди залежить якість та своєчасність розробки продукту.
Інвестори	Середній	Впливають на фінансування проекту і мають інтерес у його фінансовому успіху.

Керівництво компанії	Високий	Відповідають за стратегічне планування та ресурси, можуть змінювати пріоритети проекту.
Регулятори	Середній до високий	Законодавчі вимоги можуть вимагати значних змін у проекті, впливаючи на його вартість та терміни.
Кінцеві користувачі	Середній	Відгуки користувачів важливі для покращення продукту, їх залучення може суттєво змінити напрямок розробки.
Постачальники	Низький до середнього	Забезпечують необхідні компоненти та послуги, затримки або проблеми з якістю можуть затримати проект.

Дана таблиця допомагає розуміти, які зацікавлені сторони мають найбільший вплив на проект і яким чином. Завдяки цьому можна краще сконцентрувати зусилля на управлінні відносинами з ключовими зацікавленими сторонами та пріорітезації їхніх вимог і очікувань.

Розуміння того, як різні зацікавлені сторони можуть вплинути на проект, є важливим для розробки ефективних стратегій управління ризиками та забезпечення успішного завершення проекту. Відповідне управління взаємодіями з ними може забезпечити не тільки успіх поточного проекту, а й позитивно вплинути на майбутні ініціативи компанії.

Наступним кроком було розробка плану комунікацій між зацікавленими сторонами. Дана таблиця допомагає забезпечити ефективну комунікацію між всіма зацікавленими сторонами проекту, враховуючи їхні потреби та очікування. Ретельно спланована комунікаційна стратегія сприяє збереженню взаєморозуміння, підтримує відкритий обмін інформацією та сприяє вирішенню проблем. Результати представлені у таблиці нижче.

План комунікацій зацікавлених сторін

Зацікавлена сторона	Вид комунікації	Частота	Питання для обговорення
Клієнти	Зустрічі	Щомісяця	Оновлення щодо прогресу, вирішення важливих питань
Команда проекту	Засідання	Щотижня	Порядок денний, прогрес робіт, ідентифікація ризиків
Інвестори	Звіти	Щокварталу	Фінансовий стан проекту, відмінності від плану
Керівництво компанії	Електронна пошта	Щотижня	Ключові рішення, стратегічні зміни
Регулятори	Листування	Потреба	Зміни в законодавстві, вимоги до проекту
Кінцеві користувачі	Форуми	Потреба	Враження від продукту, пропозиції щодо покращень
Постачальники	Засідання	Потреба	Стан поставок, якість матеріалів та послуг
Робітники	Спільна площадка	Потреба	Виконання завдань, проблеми та пропозиції
Менеджмент проекту	Особисті зустрічі	Тижнево	Прогрес проекту, розподіл завдань, рішення

Дана таблиця допомагає забезпечити ефективну комунікацію між всіма зацікавленими сторонами проекту, враховуючи їхні потреби та очікування. Ретельно спланована комунікаційна стратегія сприяє збереженню взаєморозуміння, підтримує відкритий обмін інформацією та сприяє вирішенню проблем.

Взаємодія зацікавлених сторін у проекті є ключовим аспектом успішної реалізації. Розглянемо кожен елемент плану взаємодії зацікавлених сторін та

пояснимо їх детальніше:

1. Клієнти та Команда проєкту: клієнти регулярно зустрічаються з командою проєкту для обговорення поточного прогресу робіт, вирішення важливих питань та змін у вимогах.

2. Команда проєкту та Інвестори: щокварталу команда проєкту надає звіти інвесторам про фінансовий стан проєкту, відмінності від плану та інші ключові показники.

3. Керівництво компанії та Регулятори: керівництво компанії регулярно листується з регуляторами щотижня, щоб обговорити зміни в законодавстві та вимоги до проєкту та приймати відповідні рішення.

4. Кінцеві користувачі та Постачальники: кінцеві користувачі обмінюються враженнями від продукту та пропозиціями щодо його покращення на форумах, в той час як постачальники зустрічаються з командою проєкту для обговорення стану поставок та якості матеріалів та послуг.

5. Робітники та Менеджмент проєкту: робітники взаємодіють на спільній площадці, де вони обговорюють виконання завдань, проблеми та пропозиції, в той час як менеджмент проєкту проводить особисті зустрічі щотижня для обговорення прогресу проєкту, розподілу завдань та прийняття важливих рішень.

Цей план взаємодії дозволяє всім зацікавленим сторонам бути на одній хвилі, забезпечує ефективний обмін інформацією та сприяє успішному завершенню проєкту.

3.4. Управління закупівлями

Управління закупівлями включає в себе процес планування, оцінки, здійснення та контролю закупівельних операцій.

Управління закупівлями є важливою складовою управління проєктом та діловою діяльністю в цілому. Основна мета управління закупівлями - забезпечення доступності необхідних ресурсів (товарів, послуг, матеріалів,

обладнання тощо) у відповідній якості та вчасності для виконання проєкту чи бізнес-процесу.

Головні цілі управління закупівлями включають:

1. Забезпечення якості: вибір правильних постачальників та укладання вигідних угод допомагає забезпечити високу якість продуктів і послуг, що використовуються в проєкті.

2. Ефективне використання ресурсів: управління закупівлями дозволяє оптимізувати витрати на закупівлю ресурсів, мінімізувати витрати та знижувати ризики фінансових втрат.

3. Вчасна поставка: забезпечення вчасності поставок є важливим аспектом управління закупівлями, оскільки це дозволяє уникнути затримок у проєкті та забезпечити його успішне завершення.

4. Ризик менеджмент: управління закупівлями допомагає ідентифікувати та управляти ризиками, пов'язаними з недостатньою якістю продуктів, затримками у поставках або збільшенням витрат.

5. Стратегічне партнерство: вибір правильних постачальників може сприяти розвитку стратегічних партнерства, що в свою чергу може призвести до покращення умов поставок та інших переваг для бізнесу.

План управління закупівлями:

1. Визначення потреб: аналіз матеріалів, обладнання або послуг, що потрібні для проєкту. Визначаються кількість, якість та терміни.

2. Вибір постачальників: проводиться оцінка потенційних постачальників з урахуванням їхньої репутації, досвіду, цін та умов поставки.

3. Укладання угод: обговорюються та укладаються угоди з постачальниками щодо цін, умов поставки, термінів оплати та інших умов.

4. Виконання контрактів: здійснюється контроль за виконанням угод, включаючи вчасну доставку, якість та відповідність умовам контракту.

5. Моніторинг та контроль: проводиться постійний моніторинг витрат на закупівлі, якість отриманих товарів та послуг, а також вчасність поставок.

За допомогою Project Libre був створений детальний календарний план управління закупівлями.

	⊞	Name	Duration	Start	Finish
1		Визначення потреб	3 days?	4/24/24 8:00 AM	4/26/24 5:00 PM
2		Визначення необхідних матеріалів, обладнання та послуг для проєкту.	2 days?	4/24/24 8:00 AM	4/25/24 5:00 PM
3		Оцінка кількості, якості та термінів, необхідних для виконання проєкту.	1 day?	4/26/24 8:00 AM	4/26/24 5:00 PM
4		Вибір постачальників	7 days?	4/29/24 8:00 AM	5/7/24 5:00 PM
5		Пошук та аналіз потенційних постачальників.	5 days?	4/29/24 8:00 AM	5/3/24 5:00 PM
6		Перемовини та укладання угод.	3 days?	5/3/24 8:00 AM	5/7/24 5:00 PM
7		Укладання угод	5 days?	5/8/24 8:00 AM	5/14/24 5:00 PM
8		Укладання контрактів з постачальниками щодо цін, умов поставки та інших деталей.	3 days?	5/8/24 8:00 AM	5/10/24 5:00 PM
9		Підписання угод.	2 days?	5/13/24 8:00 AM	5/14/24 5:00 PM
10		Виконання контрактів	20 days?	5/15/24 8:00 AM	6/11/24 5:00 PM
11		Забезпечення виконання умов контрактів по доставці, якості та іншим аспектам.	10 days?	5/15/24 8:00 AM	5/28/24 5:00 PM
12		Контроль якості та вчасності поставок.	10 days?	5/29/24 8:00 AM	6/11/24 5:00 PM
13		Моніторинг та контроль	15 days?	5/13/24 8:00 AM	5/31/24 5:00 PM
14		Постійний моніторинг витрат на закупівлі та якості отриманих товарів і послуг	7 days?	5/13/24 8:00 AM	5/21/24 5:00 PM
15		Прийняття заходів для виправлення випадків порушення умов контрактів.	8 days?	5/22/24 8:00 AM	5/31/24 5:00 PM

Рис 3.1 Календарний план управління закупівлями

Деталізація календарного плану:

Визначення потреб:

- Визначення необхідних матеріалів, обладнання та послуг для проєкту.
- Оцінка кількості, якості та термінів, необхідних для виконання проєкту.
- (Кількість днів: 3, Початок: День 1)

Вибір постачальників:

- Пошук та аналіз потенційних постачальників.
- Переговори та укладання угод.
- (Кількість днів: 7, Початок: День 4)

Укладання угод:

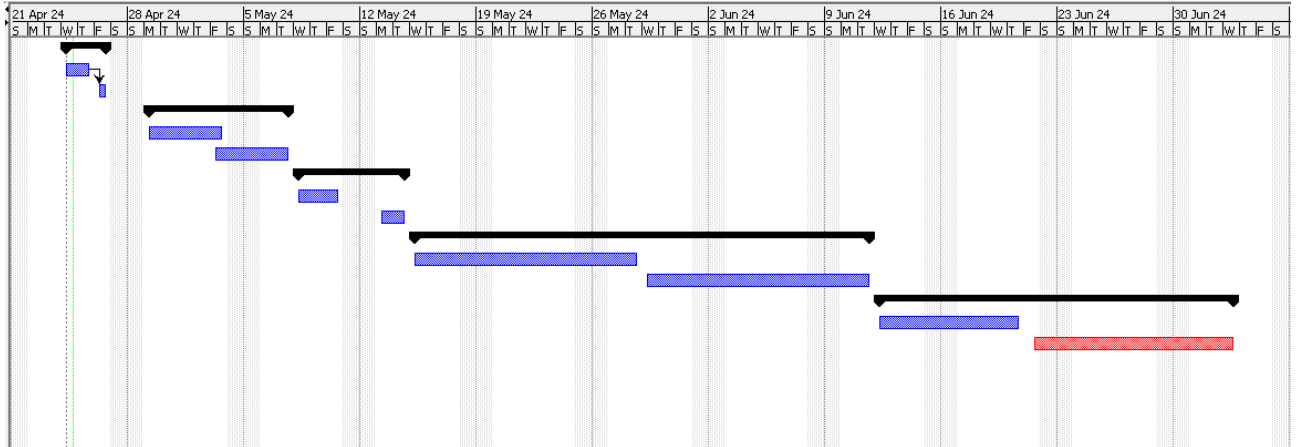
- Укладання контрактів з постачальниками щодо цін, умов поставки та інших деталей.
- Підписання угод.
- (Кількість днів: 5, Початок: День 11)

Виконання контрактів:

- Забезпечення виконання умов контрактів по доставці, якості та іншим аспектам.
- Контроль якості та вчасності поставок.
- (Кількість днів: 20, Початок: День 16)

Моніторинг та контроль:

- Постійний моніторинг витрат на закупівлі та якості отриманих товарів і послуг.
- Прийняття заходів для виправлення випадків порушення умов контрактів.



- (Кількість днів: 15, Початок: День 36)

Рис. 3.2 Графічне представлення календарного плану закупівель

Закупівлі є ключовою складовою успішного виконання проєкту, оскільки вони забезпечують необхідні ресурси для його реалізації. Для реалізації проєкту розробки програмного забезпечення для оцінки якості роботи працівників, необхідно буде організувати закупівлю наступного обладнання:

1. Комп'ютерне обладнання: комп'ютерне обладнання є необхідним для розробки програмного забезпечення, тестування продукту та інших технічних завдань, пов'язаних з проєктом. Це включає в себе комп'ютери, ноутбуки, сервери, монітори та інше обладнання.

2. Програмне забезпечення: програмне забезпечення потрібне для розробки, тестування та впровадження програмного продукту. Це можуть бути різноманітні інструменти розробки, середовища тестування, системи керування версіями та інші програмні засоби.

3. Офісне обладнання: офісне обладнання, таке як принтери, сканери, факси та інше обладнання, необхідне для організації робочого простору та

вирішення адміністративних завдань.

4. Канцелярські товари: канцелярські товари, такі як папір, ручки, канцелярські ножиці та інше, потрібні для проведення різних адміністративних та організаційних завдань.

5. Послуги з аутсорсингу: послуги з аутсорсингу можуть бути необхідними для доповнення внутрішніх ресурсів команди проєкту. Це можуть бути послуги з розробки програмного забезпечення, тестування, дизайну, маркетингу та інше.

6. Офісна меблі: офісні меблі необхідна для створення комфортного та ефективного робочого середовища для команди проєкту.

7. Інше обладнання та послуги: інші ресурси, такі як спеціалізоване обладнання, послуги з технічної підтримки, навчання персоналу та інше, які необхідні для успішного виконання проєкту.

Кожен елемент з таблиці необхідних закупівель має свою роль у забезпеченні потреб проєкту та є важливим для його успішного виконання. Без цих ресурсів проєкт може стикнутися з різними труднощами та затримками

Таблиця 3.15

Таблиця необхідних закупівель

№	Назва товару/послуги	Кількість	Одиниця виміру	Орієнтовна ціна	Постачальник
1	Система контролю версій	2	Пакет	\$500	Company A
2	Високоєфективний сервер	1	Штука	\$3000	Company B
3	Ліцензія на програмне забезпечення	5	Штука	\$200	Company C
4	Консультаційні послуги з впровадження	10 годин	Година	\$50/год	Consultant D

3.5 Моніторинг проєкту

Моніторинг проєкту було проведено за допомогою методу освоєного обсягу (Earned Value Management, EVM) - це система управління проєктами, що використовується для вимірювання продуктивності та стану проєкту шляхом порівняння фактичних результатів з планованими. Використання EVM дозволяє керівникам проєкту отримати чітку картину стану проєкту і вчасно реагувати на будь-які відхилення від плану.

Основні компоненти методу освоєного обсягу:

1. Планований обсяг робіт (PV): цей показник вказує на плановану вартість робіт на певний момент у часі. Для кожного етапу проєкту потрібно скласти графік виконання робіт та визначити плановану вартість на кожен момент часу.

2. Фактичний обсяг робіт (EV): кількість робіт, яка фактично була завершена на даний момент, вимірюється вартістю. Для кожного етапу проєкту слід визначити фактичну вартість виконаних робіт.

3. Витрачений обсяг робіт (AC): фактичні витрати на роботи, які були виконані до певного моменту у часі. Для кожного етапу проєкту необхідно зібрати інформацію про фактичні витрати.

4. Прогнозований обсяг робіт (ETC): прогнозована сума, яка ще потрібна для завершення проєкту. Щоб розрахувати це значення, можна використовувати наступну формулу:

$$ETC = EAC - AC$$

Де EAC - прогнозована кінцева вартість (Estimate at Completion), а AC - фактичні витрати.

5. Прогнозована кінцева вартість (EAC): прогнозована кінцева вартість проєкту, яка базується на даних з поточного моменту. Для розрахунку можна використовувати різні методи, наприклад, метод прямої вартості (Direct Costing), метод прогнозування на основі продуктивності (Performance-based forecasting) та інші.

6. Варіант вартості (Variance at Completion, VAC): різниця між

прогнозованою кінцевою вартістю і вартістю, яка була запланована на початку проєкту. Для розрахунку цього показника можна використовувати наступну формулу:

$$VAC = BAC - EAC$$

Де BAC - планована вартість на завершення проєкту (Budget at Completion).

Після збору всієї необхідної інформації і розрахунку цих показників, можна провести моніторинг стану проєкту, визначити відхилення від плану і розробити стратегії коригування, які допоможуть досягти поставлених цілей.

Для розрахунків з управління за методом освоєного обсягу (EVM) на повний 12-місячний період проєкту, ми використаємо ті ж базові дані, але розширимо їх до кінця проєкту:

BAC (Загальна вартість проєкту): 8,825,000 грн

PV (Запланована вартість за 12 місяців): Оскільки припускаємо лінійний розподіл бюджету, PV до кінця проєкту буде дорівнювати BAC.

PV: 8,825,000 грн

EV (Фактична вартість за 12 місяців): Припустимо, що роботи виконані на 100%.

EV: 8,825,000 грн

AC (Фактичні витрати): Якщо фактичні витрати виявляться на 20% вищими від запланованих на весь період через несподівані витрати та інші чинники.

AC: $8,825,000\text{грн} \times 1.20 = 10,590,000\text{грн}$

Тепер розрахуємо ключові показники EVM на кінець 12-го місяця:

Індекс продуктивності за графіком (SPI): $\frac{EV}{PV} = \frac{8,825,000\text{грн}}{8,825,000\text{грн}} = 1$ – проєкт точно відповідає графіку

Індекс продуктивності за вартістю (CPI) : $\frac{EV}{AC} = \frac{8,825,000\text{грн}}{10,590,000\text{грн}} \approx 0.83$ – проєкт перевищує запланований бюджет на перший рік, це виправдано планом отримання суттєвого прибутку протягом 5 років після розробки.

Відхилення за графіком (SV): $EV - PV = 8,825,000\text{грн} - 8,825,000\text{грн} = 0$ – немає відставання за графіком.

Відхилення за вартістю (CV): $EV - AC = 8,825,000\text{грн} - 10,590,000\text{грн}$ – виявлення відхилення від бюджету.

ВИСНОВКИ

1. В рамках дипломної роботи було розроблено комплексний проєкт програмного забезпечення, що включає детальний аналіз потреб та вимог користувачів. Проєкт зосереджений на покращенні систем оцінки продуктивності працівників з використанням сучасних технологій.

2. В процесі розробки було створено і детально описано життєвий цикл проєкту, що дозволяє зрозуміти його етапи, від початкового аналізу та планування до розробки, впровадження та підтримки.

3. Було ретельно розплановано бюджет проєкту, що включає розрахунки витрат на ресурси, матеріали та інші аспекти, необхідні для реалізації проєкту.

4. Проєкт включає обґрунтований аналіз ризиків, який допомагає ідентифікувати потенційні проблеми та запропонувати методи їх мінімізації або вирішення.

5. В дипломній роботі описано процес управління зацікавленими сторонами, що є ключовим для успішного впровадження проєкту та забезпечення його прийняття користувачами.

6. У дипломній роботі також було здійснено ретельний аналіз існуючих методів розробки програмного забезпечення для оцінки продуктивності працівників. Було розглянуто різноманітні технології та підходи, що використовуються на ринку, з метою вибору найбільш ефективного для даного проєкту. На основі аналізу, було вибрано метод Agile, який виявився найбільш відповідним з огляду на його гнучкість у плануванні та відповідність швидкозмінним вимогам проєкту. Вибір методу Agile дозволив максимально адаптувати процес розробки до специфічних потреб проєкту, забезпечити високий рівень залучення зацікавлених сторін та ефективно управляти змінами протягом усього життєвого циклу розробки.

7. Результати проєкту демонструють розробку ефективного програмного забезпечення, що може значно підвищити продуктивність працівників, оптимізувати робочі процеси та підвищити загальну ефективність організацій.

СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Boehm, B.W. (1981). Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall.
2. Brooks, F.P. (1975). The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley.
3. McConnell, S. (2004). Code Complete: A Practical Handbook of Software Construction. Microsoft Press.
4. Pressman, R.S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
5. Sommerville, I. (2016). Software Engineering. Pearson Education Limited.
6. IEEE Computer Society. (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK). IEEE Press.
7. Humphrey, W.S. (1989). Managing the Software Process. Addison-Wesley.
8. DeMarco, T., Lister, T. (2013). Peopleware: Productive Projects and Teams. Addison-Wesley.
9. Beck, K. (2000). Extreme Programming Explained: Embrace Change. Addison-Wesley.
10. Cockburn, A. (2001). Agile Software Development: The Cooperative Game. Addison-Wesley.
11. Larman, C. (2004). Agile and Iterative Development: A Manager's Guide. Addison-Wesley.
12. Cohn, M. (2005). Agile Estimating and Planning. Prentice Hall.
13. Martin, R.C. (2009). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
14. Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.
15. Ambler, S.W. (2002). Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. John Wiley & Sons.
16. McConnell, S. (2006). Rapid Development: Taming Wild Software

Schedules. Microsoft Press.

17. Pressman, R.S. (2005). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
18. Davis, A.M. (2002). *Just Enough Requirements Management: Where Software Development Meets Marketing*. Dorset House.
19. Bass, L., Clements, P., Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley.
20. Leffingwell, D. (2010). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley.
21. Poppendieck, M., Poppendieck, T. (2007). *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley.
22. Schwaber, K., Beedle, M. (2001). *Agile Software Development with Scrum*. Prentice Hall.
23. Beck, K. (1999). *Test Driven Development: By Example*. Addison-Wesley.
24. Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
25. Humphrey, W.S. (2000). *Introduction to the Personal Software Process*. Addison-Wesley.
26. Ambler, S.W. (2006). *The Object Primer 3rd Edition: Agile Model-Driven Development with UML 2.0*. Cambridge University Press.
27. Kruchten, P. (1999). *The Rational Unified Process: An Introduction*. Addison-Wesley.
28. Larman, C., Vodde, B. (2009). *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Addison-Wesley.
29. Schwaber, K. (2004). *Agile Project Management with Scrum*. Microsoft Press.
30. Weisert, C. (2001). *Extreme Programming Perspectives*. Addison-Wesley.

Організаційна структура команди проєкту

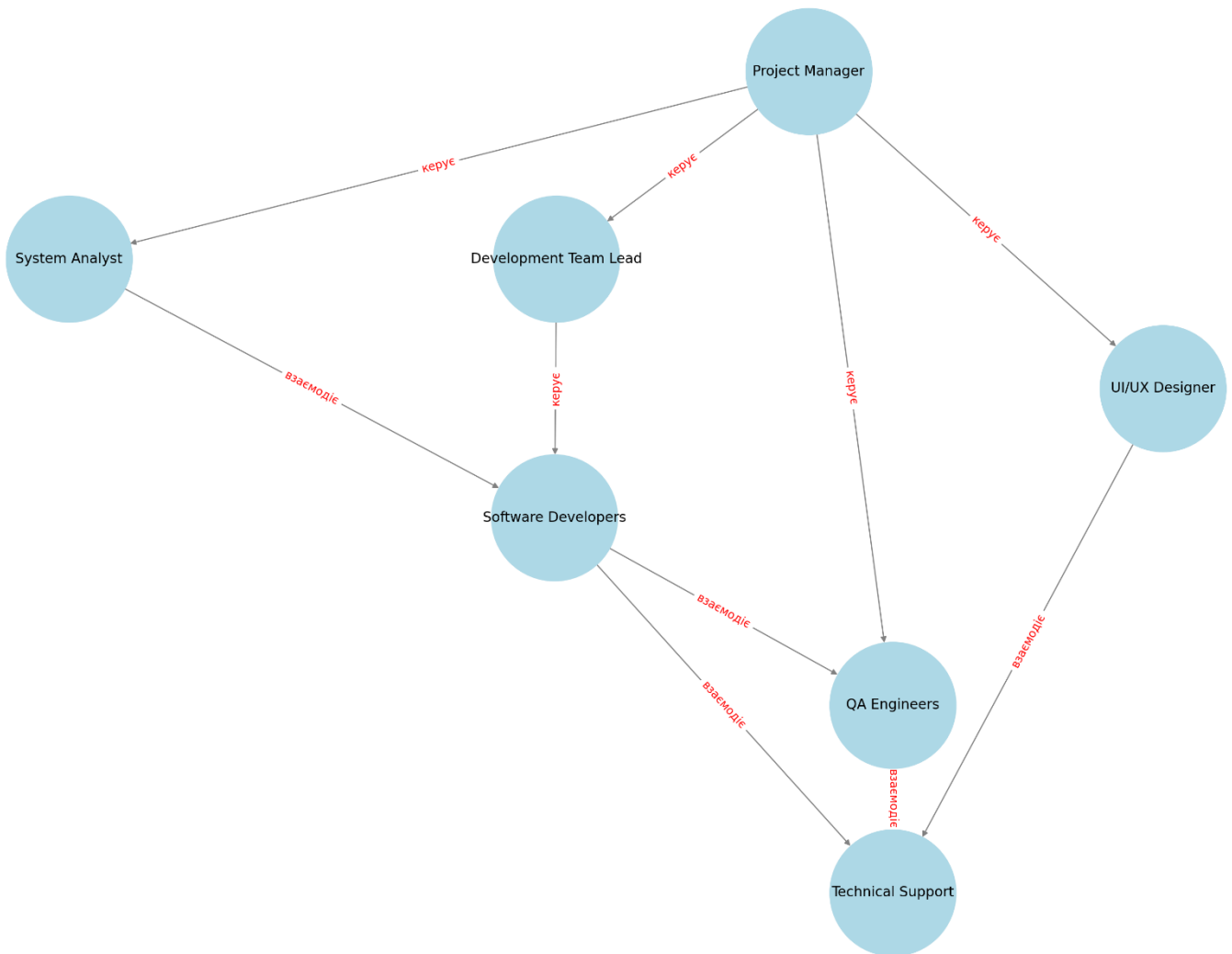


Рис. А.1 Організаційна структура команди проєкту (збалансована)

WBS проєкту

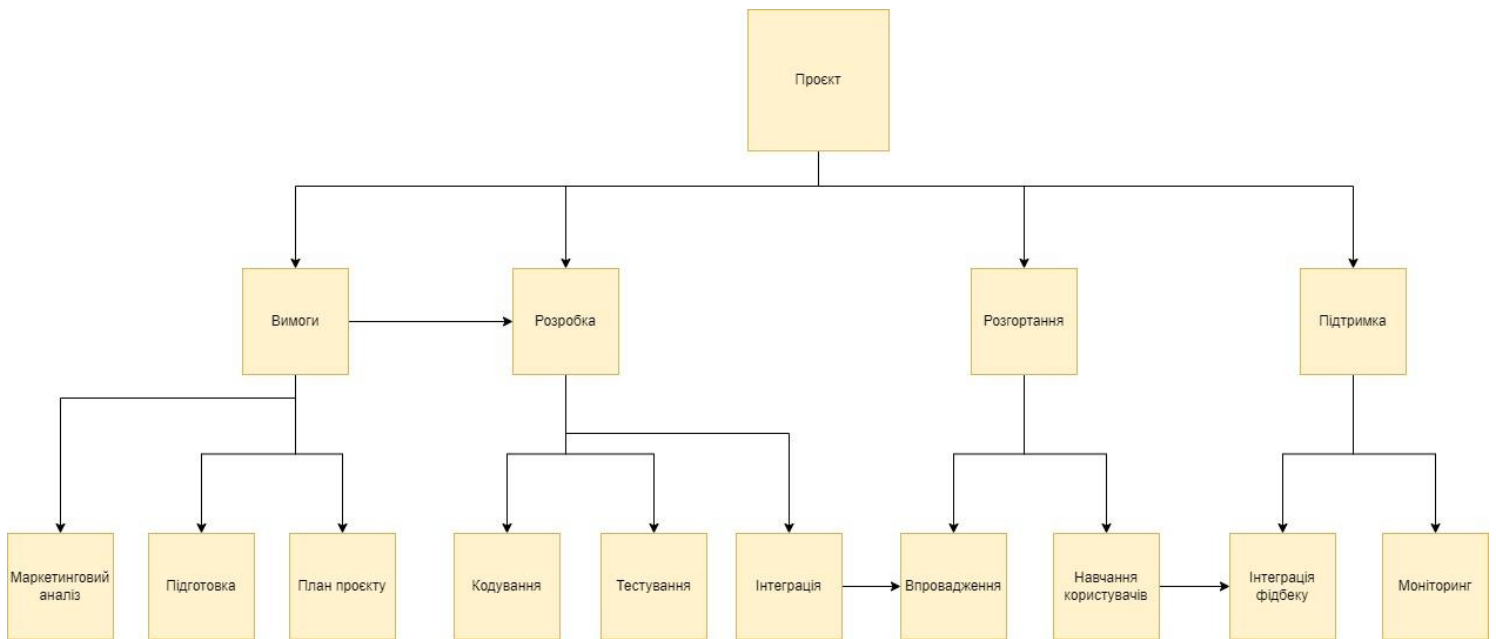


Рис. Б.1 WBS проєкту за процесами