

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
Дослідження методів гібридного управління ІТ- проектами в системах
дистанційного навчання
(тема)

Виконав:

Студент 2 курсу, групи УПГІТМ-22-3
Калінін Денис Володимирович
(прізвище, ім'я, по батькові)


Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
Освітня програма Управління
проектами в галузі інформаційних
технологій
(повна назва освітньої програми)

Керівник
доц. каф. ІУС Марина КУДРЯВЦЕВА
(Власне ім'я ПРІЗВИЩЕ)

Допускається до захисту

Зав. кафедри


(підпис)

Костянтин ПЕТРОВ
(Власне ім'я ПРІЗВИЩЕ)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
 Кафедра Інформаційних управляючих систем
 Рівень вищої освіти другий (магістерський)
 Спеціальність 122 Комп'ютерні науки
 (код і повна назва)
 Тип програми освітньо-наукова
 (освітньо-професійна або освітньо-наукова)
 Освітня програма Управління проектами в галузі інформаційних технологій
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри



(підпис)

«01» квітня 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

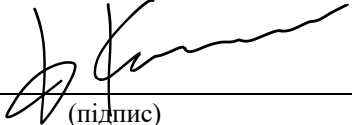
студентці Калініну Денису Володимировичу
 (прізвище, ім'я та по батькові)

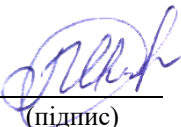
1. Тема роботи Дослідження методів гібридного управління ІТ-проектами в системах дистанційного навчання
 затверджена наказом університету від 01 квітня 2024 р. № 258 Сп
2. Термін подання студентом роботи до екзаменаційної комісії 30 червня 2024 р.
3. Вихідні дані до роботи: Аналіз підходів і методів управління проектами (Scrum, Agile, Waterfall, Stage-Gate, Kanban) та їхніх переваг і недоліків. Розгляд проблем контролю проєктів і імплементації методів у системах дистанційного навчання. Оцінка результатів впровадження гібридних методів, постановка задачі дослідження. Дослідження ефективності гібридних методів (Water-Scrum-Fall, Waterfall-Agile), критерії оцінки, аналіз ризиків, вибір ефективних елементів. Опис розробки гібридного методу, поєднання Agile, Kanban і Waterfall, новизна взаємодії з OpenAI. Реалізація гібридної системи управління проектами, впровадження в реальні проєкти, оцінка результатів і коригування методології.
4. Перелік питань, що потрібно опрацювати в роботі: аналіз існуючих підходів та методів управління проектами, вивчення монолітних методологій та парадигм управління та їх характеристик, огляд підходів та їх переваг і недоліків, дослідження гібридних методів, структурне формування універсального гібридного методу спеціалізованому для зручності навчання, розробка демонстраційної програмної системи.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз існуючих підходів та методів управління проектами	01.04.2024 – 14.04.2024	виконано
2	Вивчена монолітних методологій та парадигм управління та їх характеристик	07.04.2024 – 14.04.2024	виконано
3	Дослідження гібридних методів	14.04.2024 – 21.04.2024	виконано
4	Структурне формування універсального гібридного методу спеціалізованому для зручності навчання	21.04.2024 – 28.04.2024	виконано
5	Розробка демонстраційної програмної системи	28.04.2024 – 05.05.2024	виконано
6	Оформлення пояснювальної записки	05.05.2024 – 26.05.2024	
7	Підготовка презентації	19.05.2024 – 26.05.2024	виконано
8	Надання роботи для перевірки на плагіат	19.05.2024 – 26.05.2024	виконано
9	Попередній захист	19.05.2024 – 26.05.2024	виконано
10	Надання роботи на рецензію	26.05.2024 – 02.06.2024	виконано
11	Захист	07.06.2024	виконано

Дата видачі завдання: «01» квітня 2024 р

Студент  _____
(підпис)

Керівник роботи  _____
(підпис)

доц. каф. ІУС Марина КУДРЯВЦЕВА
(посада, власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить: 112 с., 17 рис., 4 табл., 16 джерел, 1 додатки.

ГІБРИДНИЙ МЕНЕДЖМЕНТ, ІТ ПРОЄКТИ, КАЛЕНДАРНИЙ ПЛАН, КРИТЕРІЙ ЕФЕКТИВНОСТІ ПРОЄКТУ, МЕТОДОЛОГІЯ, МОНІТОРИНГ, ПЛАНУВАННЯ, ТРИВАЛІСТЬ РОБІТ.

Об'єктом даної кваліфікаційної роботи є процес імплементації гібридних методів управління в ІТ проєктах у системах дистанційного навчання. Мета – аналіз існуючих гібридних методів управління ІТ проєктами та їх вдосконалення.

Предметом дослідження є гібридні методи управління проєктами, що поєднують елементи Agile і Waterfall підходів для оптимізації роботи над дистанційними навчальними проєктами. Робота проводить детальний аналіз існуючих гібридних методів, виявляє їхні переваги та недоліки, а також формулює проблему імплементації цих методів, яка потребує вирішення.

Завданням кваліфікаційної роботи є удосконалення ефективності гібридних методів управління для планування ІТ проєктів в рамках систем дистанційного навчання. Для досягнення мети було проведено дослідження практичного застосування гібридних методів, що дозволило реалізувати удосконалену модель управління та сформулювати оцінку її ефективності.

Після вивчення існуючих методів управління ІТ проєктами була розроблена удосконалена гібридна модель управління для малих і середніх проєктів, результати якої підтверджують її ефективність і доцільність застосування в освітніх системах.

ABSTRACT

The explanatory note to the qualification work contains: 88 pages, 17 figures, 4 tables, 16 sources, 1 appendices.

HYBRID MANAGEMENT, IT PROJECTS, CALENDAR PLAN, PROJECT EFFICIENCY CRITERIA, METHODOLOGY, MONITORING, PLANNING, WORK DURATION.

The object of this qualification work is the process of implementing hybrid management methods in IT projects in distance learning systems. The goal is to analyze the existing hybrid methods of IT project management and their improvement.

The subject of research is hybrid project management methods that combine elements of Agile and Waterfall approaches to optimize work on distance learning projects. The work conducts a detailed analysis of the existing hybrid methods, reveals their advantages and disadvantages, and also formulates the problem of the implementation of these methods, which needs to be solved.

The task of the qualification work is to improve the efficiency of hybrid management methods for planning IT projects within the framework of distance learning systems. To achieve the goal, a study of the practical application of hybrid methods was conducted, which made it possible to implement an improved management model and form an assessment of its effectiveness.

After studying existing IT project management methods, an improved hybrid management model for small and medium-sized projects was developed, the results of which confirm its effectiveness and feasibility of application in educational systems.

ЗМІСТ

Вступ.....	9
1 Аналіз існуючих підходів та методів управління проектами в системах дистанційного навчання.....	10
1.1 Загальні відомості про управління ІТ проектами.....	10
1.1.1 Характеристика сучасних ІТ проектів	11
1.1.2 Ініціація за принципом «Request for Proposal»	12
1.2 Аналіз проблем контролю проектів та проблем навчальних систем контролю проектів.....	15
1.3 Огляд та порівняння інструментальних засобів управління проектами.....	18
1.3.1 Scrum.....	18
1.3.2 Agile	21
1.3.3 Waterfall	23
1.3.4 Stage-Gate	26
1.3.5 Kanban.....	28
1.3.6 Переваги та недоліки розглянутих методів	30
1.4 Аналіз оточення розгортання програмного забезпечення.....	32
1.4.1 Аналіз імплементації програмних методів управління в систему.....	34
1.5 Постановка задачі дослідження	35
2 Дослідження ефективності методів гібридного управління	37
2.1 Критерії оцінки ефективності.....	37
2.2 Аналіз гібридних методологій.....	43
2.2.1 Water-Scrum-Fall	45
2.2.2 Waterfall-Agile	47
2.2.3 Дослідження ризиків гібридних підходів.....	49
2.3 Фільтрація елементів методів управління проектами.....	58
3 Опис і використання методу управління проектами	64

3.1 Поєднання методів Agile, таблиці Kanban та Waterfall.....	64
3.2 Новизна імплементації взаємодії із OpenAI.....	66
3.3 Опис реалізації структурних елементів.....	70
4 Реалізація гібридної системи для управління проектами	76
Висновки.....	84
Перелік джерел посилання	86
Додаток А Графічні матеріали	89

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

RFP – Request for Proposal

ISACA – Information Systems Audit and Control Association

OWASP – Open Web Application Security Project

CSV – Comma-Separated Values

SQL – Structured Query Language

ВСТУП

У сучасному світі управління проектами відіграє ключову роль у досягненні цілей організацій та забезпеченні їхнього успіху на ринку. Проте, з огляду на швидкі зміни технологій, постійно зростаючі вимоги замовників та необхідність адаптації до глобальних викликів, існуючі методології управління проектами часто зіштовхуються з проблемами ефективності та гнучкості.

Тож були розроблені гібридні методології які поєднують у собі два чи більше підходів, але це не прибрало специфічні вимоги до управління в залежності від конкретної компанії.

Також варто відмітити що системи управління не представлені у програмах чи системах по навчанню, на ринку присутні лише обмежені можливості певних сервісів управління які націлені на навчання, але функціонал часто не адаптований та не повноцінний для цих цілей.

Традиційні підходи, як-от Waterfall, хоча й забезпечують чітку структуру та передбачуваність процесів, часто виявляються занадто негнучкими для адаптації до швидких змін в умовах проекту. З іншого боку, гнучкі методології, такі як Agile, хоч і пропонують необхідну адаптивність, іноді можуть призвести до втрати контролю над довготривалим плануванням та обсягом робіт.

Таким чином, існує очевидна потреба у розробці комплексного підходу до управління проектами, який поєднував би структурованість традиційних методів із гнучкістю сучасних практик, дозволяючи командам ефективно адаптуватися до змін, водночас забезпечуючи високий рівень продуктивності та задоволення вимог замовників та адаптований до вимог навчання.

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ТА МЕТОДІВ ГІБРИДНОГО УПРАВЛІННЯ ІТ ПРОЕКТАМИ В СИСТЕМАХ ДИСТАНЦІЙНОГО НАВЧАННЯ

1.1 Загальні відомості про управління ІТ проектами

ІТ проекти у загальній характеристиці представляють собою продукт який є результатом розробки за певними правилами та вимогами, впровадження або підтримку програмного забезпечення, апаратних засобів, мережових систем або будь-яких інших компонентів, пов'язаних із обробкою та передачею інформації.

Вони включають в себе великий спектр діяльності, від створення веб-сайтів і розробки мобільних додатків до впровадження корпоративних систем управління ресурсами підприємства (ERP) чи систем електронного навчання.

ІТ проекти характеризуються певними особливостями, серед яких важливість терміновості та точності, постійна зміна вимог та технологій, а також велика кількість зацікавлених сторін. Зазвичай такі проекти вимагають великого обсягу ресурсів, таких як фінанси, персонал та технічні засоби, і проводяться в рамках певного часового обмеження. Управління ІТ проектами включає в себе планування, виконання, контроль та завершення проектів з метою досягнення поставлених цілей в рамках обмежень, таких як бюджет, час та якість.

Оскільки технологічний ландшафт швидко змінюється, а вимоги користувачів неперервно розвиваються, ефективне управління ІТ проектами стає ключовим фактором успіху для будь-якої організації. Це означає, що команди, що працюють над такими проектами, повинні бути гнучкими, швидко адаптуватися до змін та здатними ефективно співпрацювати для досягнення поставлених завдань.

У цій темі неможливо не відмітити що менеджмент людських ресурсів займає центральне місце в системах управління ІТ проектами з кількох важливих причин [1].

Люди є найважливішим активом будь-якого проекту. Їх навички, знання, досвід та мотивація безпосередньо впливають на успіх проекту. Ефективне управління людськими ресурсами забезпечує, що команда має необхідні умови для високопродуктивної роботи.

1.1.1 Характеристика сучасних ІТ проектів

ІТ проекти часто характеризуються високою складністю та необхідністю різноманітних навичок, від програмування до аналізу даних і управління проектами. Управління людьми, які працюють над проектом, вимагає точного розуміння їх сильних сторін і можливостей, щоб ефективно розподілити завдання та відповідальності.

Управління людьми також означає управління способами, якими вони спілкуються і співпрацюють. У системах управління ІТ проектами значна увага приділяється розробці ефективних каналів комунікації та забезпеченню гладкої співпраці між учасниками команди[2].

Мотивація та задоволеність працівників мають безпосередній вплив на продуктивність та якість роботи. Менеджмент людських ресурсів включає розробку ініціатив, спрямованих на підтримку високого рівня задоволеності та залученості команди, таких як програми навчання, кар'єрного росту та визнання досягнень.

Управління змінами є важливою частиною управління ІТ проектами, і люди є ключем до успішного впровадження цих змін. Розуміння реакції команди на зміни, їх адаптація та підтримка в цьому процесі є критично важливими для забезпечення неперервності проекту.

Далі варто зазначити що менеджмент необхідний в великих проектах, адже вони зазвичай є доволі великі та комплексні. Тож варто розібрати ключові із них.

1.1.2 Ініціація за принципом «Request for Proposal»

Запит на пропозицію (Request for Proposal, RFP) є ключовим елементом у плануванні та ініціації ІТ проектів, який слугує мостом між ідеєю проекту та його реалізацією. Цей документ детально описує технічні та бізнесові вимоги проекту, встановлюючи основу для залучення потенційних виконавців або постачальників послуг. Через RFP організація надає вичерпну інформацію про свої цілі, очікування та параметри проекту, щоб отримати від постачальників конкретні пропозиції з рішеннями, які відповідають зазначеним вимогам[3].

Процес RFP вимагає глибокого аналізу потреб проекту та ринку, щоб забезпечити точне та детальне формулювання запиту. Це дозволяє оцінити компетентність та досвід потенційних виконавців, їхню здатність виконати роботу в рамках визначених термінів і бюджету, а також якість запропонованих технологічних рішень. Відповіді на RFP мають містити деталізовану пропозицію з описом підходів, технологій, кошторису, плану реалізації та відповідності проектних цілей.

Важливо, щоб запит на пропозицію був складений чітко та структуровано, оскільки якість та повнота наданої інформації безпосередньо впливають на

якість отриманих пропозицій. Крім того, процес RFP повинен бути прозорим і справедливим, щоб забезпечити рівні умови для всіх учасників. У результаті, запит на пропозицію є вирішальним кроком у виборі найкращих постачальників або партнерів для реалізації ІТ проекту, сприяючи його успіху.

Обсяг проекту визначає межі та специфікації того, що має бути виконано в рамках ІТ проекту, встановлюючи чіткі критерії для успішного завершення. Цей компонент є фундаментом для планування, виконання, моніторингу та контролю проекту, оскільки він деталізує вимоги та очікування, задачі, цілі, функції, процеси та результати, які повинні бути досягнуті. Визначення обсягу проекту допомагає уникнути непорозумінь між командою проекту та зацікавленими сторонами, забезпечуючи спільне розуміння цілей і обмежень.

Ключовим елементом управління обсягом проекту є розробка структурованого документа, який включає в себе Декларацію обсягу проекту, де детально описано, що входить до проекту та що з нього виключено. Це допомагає всім учасникам проекту мати однакове розуміння задач і обмежень, а також слугує основою для розробки більш деталізованих планів проекту.

Ретельно визначений обсяг проекту є критично важливим для управління змінами, оскільки будь-які відхилення від первісно затвердженого обсягу потребують чіткого аналізу впливу, затвердження та внесення відповідних змін у проектні плани. Також, це допомагає забезпечити, що ресурси проекту використовуються ефективно, оскільки учасники проекту зосереджені на досягненні визначених цілей, мінімізуючи ризик "здуття" проекту та неочікуваних робіт.

В кінцевому підсумку, чітко визначений обсяг проекту сприяє ефективному плануванню, реалізації та контролю проекту, дозволяючи команді успішно досягати поставлених цілей у межах встановлених термінів і бюджету.

Це один з ключових аспектів управління проектами, який вимагає уваги на початкових етапах проекту для забезпечення його успішного завершення.

Команда проекту — це основа успіху будь-якого ІТ проекту, що складається з різних спеціалістів, які працюють разом над досягненням спільної мети. Вона може включати розробників, аналітиків, менеджерів проекту, тестувальників, а також фахівців із забезпечення якості, управління ризиками та інших важливих областей. Формування команди проекту є критичним етапом, на якому вирішується, наскільки ефективно ресурси будуть використані для досягнення цілей проекту.

Ключ до успішного формування команди полягає в правильному підборі кандидатів з необхідними навичками та досвідом, а також у забезпеченні їх здатності ефективно співпрацювати. Комунікація є життєво важливою для забезпечення злагодженої роботи команди, а також для вирішення конфліктів та мінімізації непорозумінь.

Менеджер проекту відіграє ключову роль у керуванні командою, відстежуючи прогрес роботи та забезпечуючи, щоб всі члени команди були націлені на спільні цілі. Мотивація та підтримка команди також є важливими аспектами її управління, оскільки вони сприяють підвищенню продуктивності та задоволеності працівників.

Управління командою проекту включає регулярні зустрічі для обговорення прогресу, вирішення проблем та адаптації планів за потреби. Розподіл відповідальності та завдань, заснований на сильних сторонах кожного члена команди, забезпечує ефективне використання ресурсів та сприяє згуртованості команди.

Врешті-решт, успіх проекту значною мірою залежить від здатності команди ефективно працювати разом, використовуючи свої унікальні навички та досвід для досягнення спільної мети.

Усі ці та інші фактори являють собою структурний об'єкт для управління та постійного контролю.

1.2 Аналіз проблем контролю проектів та проблем навчальних систем контролю проектів

Управління ІТ проектами, хоча і дозволяє розвивати та впроваджувати інноваційні технологічні рішення, часто зустрічається з рядом проблем, які можуть ускладнювати процес і впливати на результативність проекту. Розглянемо причини деяких проблем управління ІТ проектами:

- зміна вимог: один з найбільш поширених викликів - це постійні зміни вимог з боку замовника або користувачів. часті зміни можуть призводити до збільшення обсягу робіт, затримок у графіку та перевищення бюджету;
- неясні чи недостатньо визначені цілі: не завжди вдається чітко визначити цілі проекту з самого початку. це може призвести до непорозумінь щодо очікувань замовника та невідповідності результатів реальним потребам користувачів;
- брак ресурсів: недостатність фінансових, людських або технічних ресурсів може суттєво ускладнити виконання проекту та призвести до затримок у графіку або низької якості продукту;
- комунікаційні проблеми: недостатня чи неефективна комунікація між учасниками проекту може призвести до непорозумінь, конфліктів та незгодженості, що в свою чергу може загрожувати успішному завершенню проекту;

- технічні складнощі: у деяких випадках можуть виникати технічні проблеми, такі як несподівані труднощі під час розробки програмного забезпечення або інтеграції нових технологій, що призводять до затримок та перевищення бюджету;
- управління ризиками: недостатнє управління ризиками може призвести до непередбачених проблем, які можуть серйозно підірвати успішність проекту;
- недооцінка складності проекту: часом менеджери проекту можуть недооцінювати складність завдань або обсяг робіт, що призводить до нереалістичних планів та затримок у виконанні.

Зрозуміння цих проблем і розробка стратегій їх вирішення є важливими компонентами успішного управління ІТ проектами.

Саме тому важливо встановлювати чіткий контроль над процесами розробки подібних великий проектів.

Чіткий контроль процесу розробки ІТ проектів має вирішальне значення з кількох причин, що робить його надзвичайно актуальним у сучасному динамічному світі технологій і бізнесу.

Контроль дозволяє впевнитися, що розробка проекту відбувається у відповідності з вимогами та очікуваннями замовника. Це зменшує ризик розбіжностей між очікуваним та реальним результатами, забезпечуючи задоволення замовника.

Скрупульозний контроль за ходом проекту допомагає виявляти та вирішувати проблеми на ранніх стадіях, запобігаючи затримкам і перевитратам. Це дозволяє командам завершувати проекти вчасно та в межах затвердженого бюджету.

У сучасному швидкоплинному технологічному середовищі проекти часто зустрічаються з несподіваними викликами та змінами вимог. Чіткий контроль

дозволяє проектним командам швидко адаптуватися та реагувати на ці зміни, зберігаючи проект на правильному шляху. Слідкування за процесом розробки включає перевірку відповідності продукту встановленим стандартам якості та вимогам безпеки. Це забезпечує високу якість кінцевого продукту, зменшує кількість помилок і недоліків[4].

Ефективне керування та контроль за проектом дозволяє оптимально використовувати доступні ресурси, знижуючи витрати та збільшуючи продуктивність роботи команди.

Чіткий контроль і прозорість процесу розробки підвищують довіру з боку замовників та інших зацікавлених сторін, оскільки вони можуть бачити прогрес і бути впевненими в успішному виконанні проекту.

Тепер варто звернути увагу на важливість ефективного навчання починаючих розробників. Одним з ключових аспектів цього процесу є знайомство з управлінням проектами, адже воно лежить в основі успішної роботи в будь-якому IT-проекті. Тут на сцену виходять системи контролю проектів, які стають незамінним інструментом для забезпечення цілісного розуміння процесу розробки.

Спрощені та уніфіковані системи контролю проектів відіграють вирішальну роль у навчанні починаючих розробників. Вони не тільки допомагають зрозуміти базові принципи управління проектами, але й навчають ефективної комунікації, планування, моніторингу та аналізу процесів розробки. Через їх використання, новачки швидше адаптуються до робочого середовища, навчаються працювати в команді та розвивають навички вирішення задач, які є критичними для будь-якого IT-проекту.

Крім того, використання таких систем стимулює самостійне мислення та критичний аналіз, дозволяючи починаючим розробникам не просто слідувати інструкціям, але й активно брати участь у формуванні проекту, вносити свої ідеї

та бачення. Це формує не тільки технічні навички, але й розвиває лідерські якості, підготовлюючи ґрунт для майбутніх інновацій та успіхів.

Використання таких систем є логічним кроком на шляху до формування кваліфікованих фахівців, готових ефективно працювати над складними проектами та вносити вагомий вклад у розвиток технологій.

Вивчення та тренування на подібних, спрощених системах дасть достатній рівень для опанування більш комплексних систем та до їх розуміння на вхідному етапі.

Але перед командами розробників, та молодими спеціалістами зараз постає питання ефективного входження у область менеджменту проекту, існуючі підходи які будуть розглянуті пізніше – дають великий поріг входу, задаючи великий темп спрінтів, заплутану систему керування та не дають багато простору на здійснення помилок які притаманні саме починаючим спеціалістам та новачкам в даній галузі менеджменту.

Саме тому постає актуальне питання по знаходженню оптимальної збалансованої в плані навантаження та вихідної продуктивності системи яка б давала ефективність та можливість вчитись за для майбутнього використання цих знань у інших, більш складних та комплексних системах.

1.3 Огляд та порівняння інструментальних засобів управління проектами

1.3.1 Scrum

Scrum – це популярна методологія управління проектами, особливо в розробці програмного забезпечення, яка базується на ітеративному та інкрементальному підходах до управління. Основною одиницею роботи в Scrum

є "спринт", який представляє собою фіксований часовий проміжок (зазвичай 2-4 тижні), під час якого команда зосереджується на виконанні конкретного списку завдань. Основні особливості Scrum включають використання крос-функціональних команд, самоорганізацію та ітеративний підхід до розробки продукту.

Крім того, Scrum використовує ролі, такі як Product Owner (власник продукту), Scrum Master (фасилітатор процесу) та розробників, для забезпечення ефективного управління та співпраці. Один з ключових принципів Scrum - це регулярність: регулярні зустрічі спринтів, ретроспективи та планування допомагають зберігати стійкий ритм роботи.

Також, він акцентує на важливості сталих зворотних зв'язків з клієнтом або замовником, що дозволяє швидко адаптувати продукт до змін вимог. Ці особливості роблять потужним інструментом для розробки продуктів у вимогливих та змінюючих середовищах.

Scrum – це гнучка методологія управління проектами, яка стала особливо популярною в розробці програмного забезпечення. Вона базується на ітеративному та інкрементальному підході, акцентуючи на співпраці між членами команди та замовником. Основні будівельні елементи Scrum включають:

- власник продукту (product owner): відповідає за визначення функціоналу продукту, пріоритетів завдань і взаємодію зі стейкхолдерами;
- scrum-майстер (scrum master): фасилітує процес scrum, допомагаючи команді дотримуватися принципів scrum, вирішує перешкоди та підтримує продуктивну роботу;
- команда розробників (development team): група професіоналів, яка відповідає за виконання роботи по розробці продукту. команди зазвичай багатофункціональні та самоорганізовані;

- беклог продукту (product backlog): перелік усіх вимог до продукту, включаючи функціонал, фікси, поліпшення, з упорядкованим пріоритетом;
- беклог спринту (sprint backlog): підмножина беклогу продукту, яку команда планує реалізувати протягом наступного спринту;
- інкремент продукту (product increment): сума всіх елементів беклогу продукту, завершених під час спринтів, які досягли критеріїв прийнятності;
- планування спринту (sprint planning): зустріч на початку кожного спринту, де команда обирає роботу на наступний спринт з беклогу продукту;
- денний стендап (daily scrum): щоденна коротка зустріч для команди, щоб синхронізувати роботу та планувати дії на наступний день;
- огляд спринту (sprint review): зустріч наприкінці спринту для демонстрації зробленої роботи та отримання зворотного зв'язку від стейкхолдерів;
- ретроспектива спринту (sprint retrospective): зустріч після огляду спринту, але перед наступним плануванням спринту, де команда обговорює, що працювало добре, що могло б бути краще, і як вдосконалити наступний спринт[5].

Далі розглянемо структурну взаємодію елементів даного підходу, структура зображена на рисунку 1.3.1.

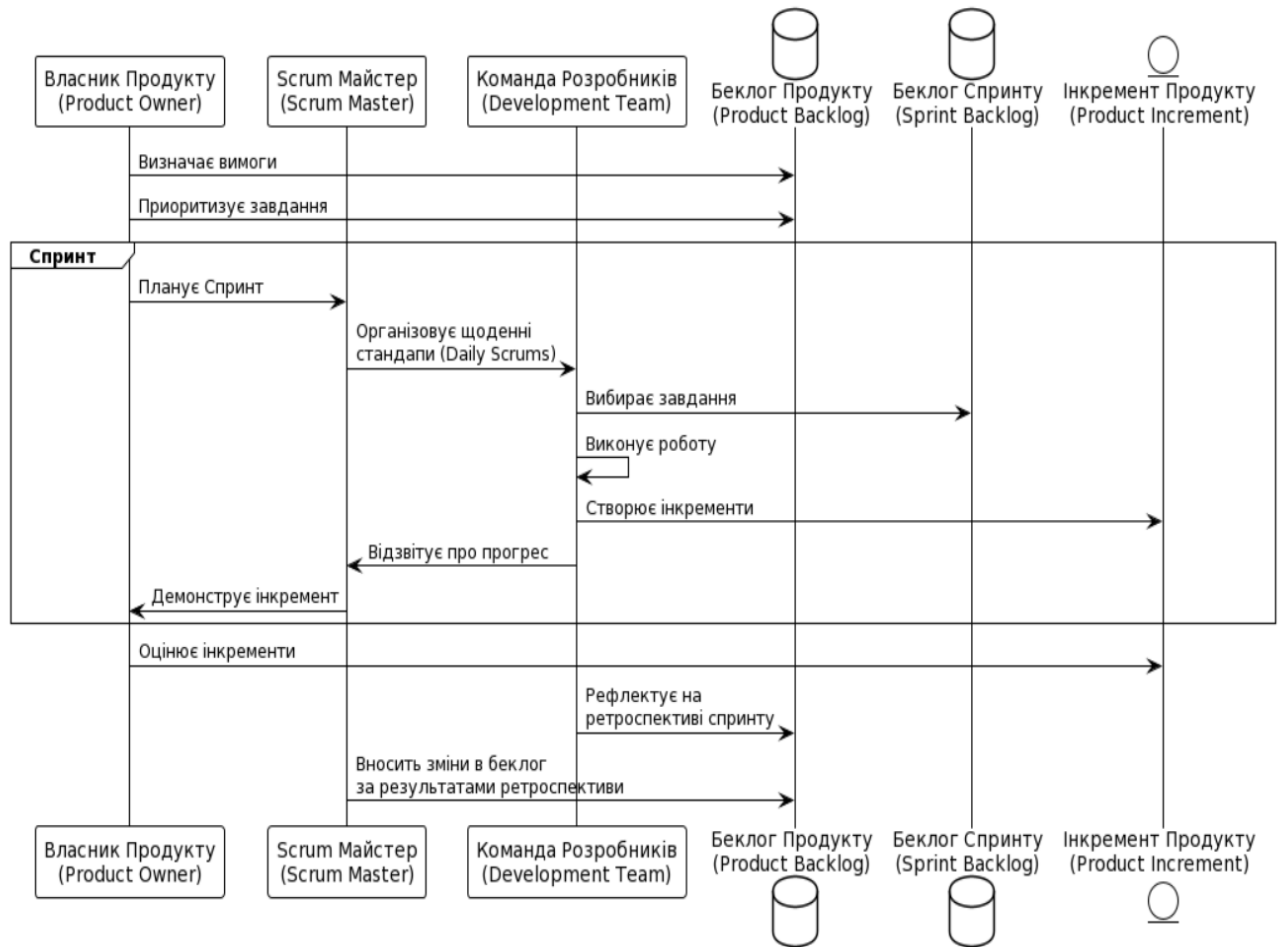


Рисунок 1.3.1 – Взаємодія елементів Scrum

1.3.2 Agile

Agile, або гнучка методологія, представляє собою підхід до управління проектами, зокрема в розробці програмного забезпечення, який акцентує на швидкій адаптації до змін, ітеративному розвитку та високому рівні співпраці між усіма зацікавленими сторонами. Agile прагне мінімізувати ризики, шляхом розбиття процесу розробки на короткі періоди, звані ітераціями або спринтами, що тривають зазвичай від одного до чотирьох тижнів.

Кожен спринт передбачає планування, аналіз вимог, проектування, кодування, тестування та оцінку, завершуючись демонстрацією проміжного продукту замовнику.

Цей підхід дозволяє замовникам та розробникам вести постійний діалог та швидко вносити корективи в проект. Agile вихваляє принципи, закладені в Agile Маніфесті, серед яких: пріоритет взаємодії між людьми над процесами та інструментами, фокус на функціонуючому програмному забезпеченні над докладною документацією, співпраця з клієнтом понад договірні угоди, та готовність до змін над слідуванням початковому плану.

Метод сприяє створенню продуктів вищої якості, оскільки залучення замовника та адаптація до змін відбуваються на ранніх стадіях розробки. В результаті, команди, які слідують Agile методології, можуть швидше відповідати на зміни ринку та більш ефективно задовольняти потреби своїх клієнтів[6].

- дошки (boards): використовуються для трекінгу прогресу спринту та роботи над функціоналом;
- беклог (backlog): для планування спринтів та визначення, які завдання будуть виконуватися наступними;
- спринти (sprints): для швидкої реалізації набору функцій та отримання зворотного зв'язку від користувачів або замовників;
- щоденні зустрічі (daily stand-ups): сприяють синхронізації дій команди та швидкому вирішенню проблем;
- демонстрації (demos): дозволяють отримати зворотний зв'язок від замовників та користувачів, що сприяє подальшому покращенню продукту;
- ретроспективи (retrospectives): зустрічі після завершення кожного спринту для обговорення того, що працювало добре, що могло бути краще, та планування змін для наступного спринту.

Далі розглянемо структурну взаємодію елементів даного підходу, структура зображена на рисунку 1.3.2.

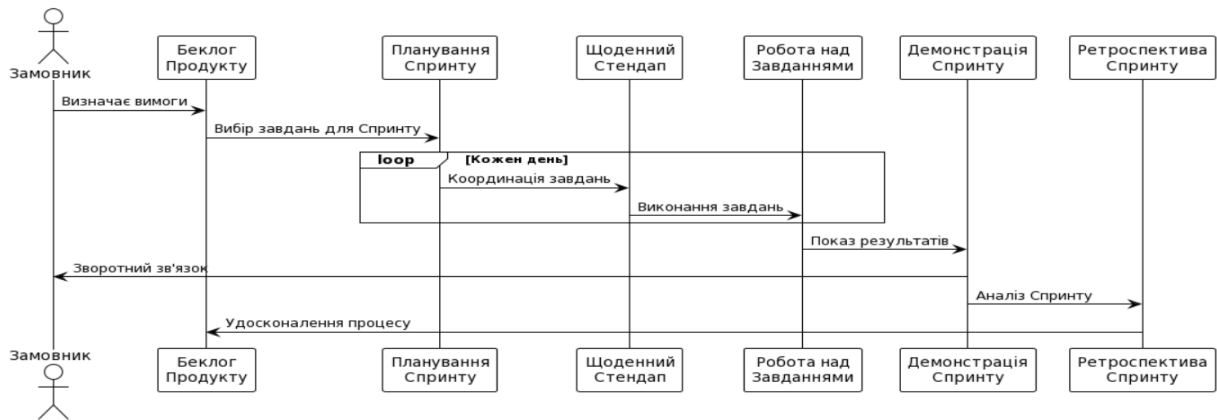


Рисунок 1.3.2 – Взаємодія елементів Agile

На цій діаграмі відображено ключові етапи Agile процесу, починаючи з визначення вимог замовником та формування беклогу продукту. Далі йде процес планування спринту, де команда вибирає завдання з беклогу для роботи в наступному спринті.

Щоденні стендапи допомагають координувати роботу команди та вирішувати поточні завдання. Після завершення спринту проводиться демонстрація досягнутих результатів замовнику, яка дозволяє отримати зворотний зв'язок. Ретроспектива спринту допомагає команді аналізувати свою роботу та шукати шляхи для подальшого удосконалення процесів.

1.3.3 Waterfall

Метод Waterfall, або водоспадний метод, є класичним підходом у керуванні проектами, зокрема в області розробки програмного забезпечення. Він

передбачає строгу послідовність етапів, де кожен наступний етап починається лише після повного завершення попереднього. Основні фази включають визначення вимог, проектування, реалізацію, тестування, впровадження та обслуговування. Цей лінійний підхід забезпечує чітку структуру та легкість управління проектом, дозволяючи ефективно планувати ресурси та час.

Однак, даний підхід має певні обмеження, особливо у випадках, коли вимоги до проекту не можуть бути повністю визначені на початковому етапі або коли вони схильні до зміни. У таких ситуаціях, жорстка послідовність Waterfall може призвести до затримок і збільшення вартості проекту, оскільки будь-які зміни вже в реалізованій частині вимагають повернення до попередніх фаз.

Метод Waterfall найкраще підходить для проектів з чітко визначеними вимогами та низьким ризиком змін у процесі реалізації. Він дозволяє легко відстежувати прогрес та забезпечує документування на кожному етапі розробки. Незважаючи на критику за недостатню гнучкість, Waterfall продовжує бути використаним в певних проектах, де прогнозованість та структурованість мають вирішальне значення.

Далі розглянемо більш детально структурні елементи підходу.

Фази:

- визначення вимог: збір детальних вимог від замовників і стейкхолдерів;
- проектування: розробка архітектури системи та детального дизайну рішення;
- реалізація: кодування та розробка програмного забезпечення відповідно до проектних специфікацій;
- тестування: перевірка розробленого продукту на відповідність вимогам і виявлення помилок;

- впровадження: розгортання готового продукту в операційне середовище;
- підтримка та обслуговування: здійснення необхідних виправлень, оновлень та поліпшень продукту.

Кожен етап проекту генерує свою частину документації, що служить основою для наступних фаз. Документація включає технічні специфікації, тестові плани, проектні плани тощо.

Хоча Waterfall не визначає специфічні ролі так, як Scrum, в проекті зазвичай беруть участь:

- проектний менеджер: відповідає за загальне керування проектом та досягнення його цілей;
- аналітики вимог: займаються збором і аналізом вимог до проекту;
- архітектори та дизайнери систем: розробляють архітектуру і дизайн продукту;
- розробники: виконують програмування та кодування продукту;
- тестувальники: перевіряють продукт на наявність помилок і відповідність вимогам[7].

На кінці кожної фази зазвичай проводиться ревізія результатів роботи та затвердження переходу до наступної фази.

Далі розглянемо структурну взаємодію елементів даного підходу, структура зображена на рисунку 1.3.3.

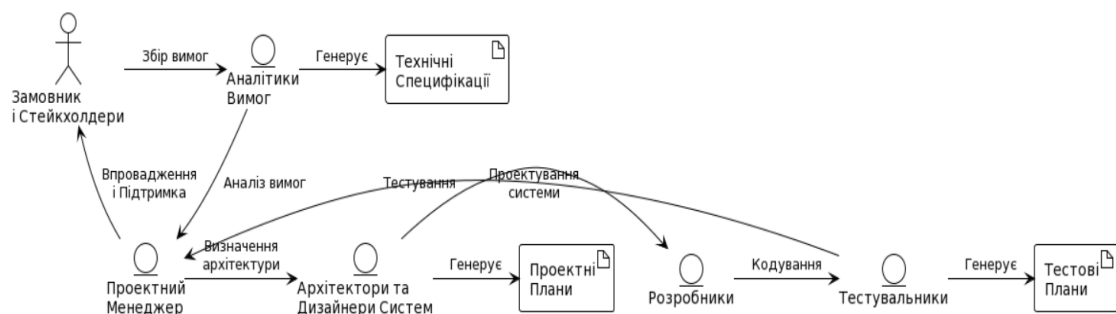


Рисунок 1.3.3 – Взаємодія елементів Waterfall

1.3.4 Stage-Gate

Stage-Gate є популярною та широко використовуваною моделлю управління проектами, особливо в області розробки нових продуктів. Ця модель, розроблена доктором Робертом Г. Купером, допомагає організаціям оптимізувати процес розробки від ідеї до запуску на ринку, зосереджуючись на стратегічному плануванні та оцінці на кожному етапі. Модель Stage-Gate розділяє процес на окремі етапи (stages), кожен з яких завершується "воротами" (gates), де приймаються рішення про подальше продовження проекту.

Кожен етап включає певну набір дій та завдань, які необхідно виконати перед переходом до наступного етапу. Етапи можуть включати:

- попереднє дослідження: оцінка ідеї, ринкові дослідження, технічне дослідження тощо;
- детальне дослідження: розробка детального плану проекту, включаючи аналіз ринку, фінансові прогнози, технічні специфікації;
- розробка: створення прототипу або бета-версії продукту, розробка маркетингової стратегії;
- тестування та валідація: випробування продукту, збір зворотного зв'язку від споживачів, оцінка виробничих процесів;
- запуск: підготовка до виробництва, розробка ринкових стратегій та планів запуску[8].

"Ворота" є контрольними пунктами, де висококваліфікована команда оцінює прогрес проекту і вирішує, чи варто продовжувати інвестувати в нього. На кожних "воротах" розглядаються:

- отримані результати етапу;
- актуальність проекту в контексті стратегічних цілей організації;

- відповідність проекту бюджету та графіку;
- ризики та можливості проекту.

Переваги моделі Stage-Gate:

- структурований підхід: модель забезпечує чітку структуру для керування процесом розробки, зменшуючи ризики та невизначеність;
- підтримка прийняття рішень: "ворота" надають механізм для оцінки та перегляду проекту на кожному етапі, допомагаючи приймати обґрунтовані рішення про продовження чи припинення розробки;
- ефективність ресурсів: дозволяє оптимізувати використання ресурсів, спрямовуючи їх на найбільш перспективні проекти;
- краще управління ризиками: регулярна оцінка проекту на різних етапах допомагає ідентифікувати та мінімізувати ризики на ранніх стадіях;
- підвищення шансів на успіх: систематичний підхід до розробки та запуску продукту підвищує ймовірність його успішного впровадження на ринку.

Модель Stage-Gate ідеально підходить для компаній, які розробляють нові продукти або технології та хочуть мінімізувати ризики, пов'язані з інвестиціями в інновації. Вона особливо корисна в індустріях з високим рівнем невизначеності та довгими циклами розробки, таких як фармацевтика, біотехнології, виробництво обладнання та ін.

Завдяки своїй структурованості та фокусу на ретельній оцінці на кожному етапі, Stage-Gate допомагає забезпечити, що лише найбільш виправдані проекти отримують необхідне фінансування та ресурси для продовження розробки, тим самим підвищуючи загальну ефективність інноваційних інвестицій компанії.

1.3.5 Kanban

Kanban є однією з гнучких методологій управління проектами, яка виникла в Японії в 1940-х роках на заводах Toyota як система для оптимізації виробничих процесів. Слово "Kanban" у перекладі з японської означає "візуальна картка" або "табличка". Ця методологія була адаптована для розробки програмного забезпечення та управління проектами в ІТ, де вона допомагає командам керувати потоком робіт з високою ступенем прозорості та гнучкості.

Основні принципи Kanban:

- візуалізація роботи: використання дошок kanban для візуалізації всіх поточних завдань на різних етапах виконання (наприклад, "до виконання", "в роботі", "виконано"). це дозволяє легко ідентифікувати поточний статус завдань та виявити будь-які затримки або проблеми в потоці роботи;
- обмеження кількості роботи у процесі (wip): встановлення лімітів на кількість завдань, які можуть бути виконані одночасно на кожному етапі. це допомагає знизити перевантаження команди та забезпечує більш рівномірний потік роботи;
- управління потоком: спостереження за потоком робіт через систему та внесення коригувань для оптимізації процесів. мета - мінімізувати час на виконання завдань і покращити продуктивність;
- постійне вдосконалення: kanban стимулює команду до постійної оцінки робочих процесів та впровадження змін для покращення ефективності та продуктивності[9].

Функціональні елементи Kanban:

- канбан-дошка: візуальне представлення потоку робіт, зазвичай поділене на колонки, які відображають різні етапи процесу;

- картки (завдання): візуальні представлення робочих елементів, що переміщуються між колонками на дошці відповідно до статусу їх виконання;
- вір ліміти: обмеження, встановлені для кожної колонки на дошці, щоб контролювати кількість роботи, яка виконується одночасно на кожному етапі;

Далі розглянемо структурну взаємодію елементів даного підходу, структура зображена на рисунку 1.3.5.

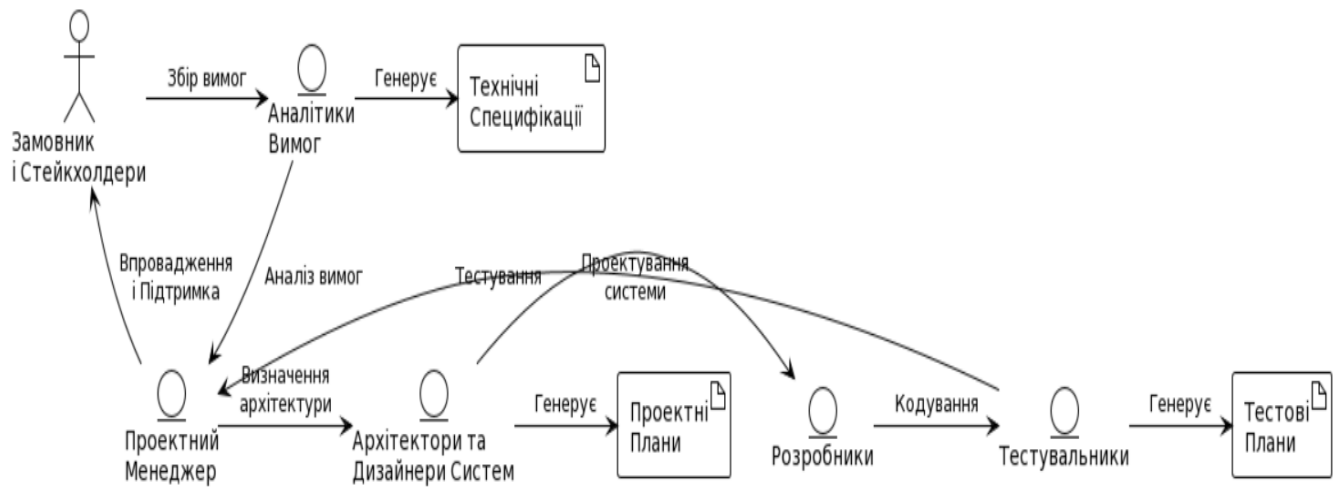


Рисунок 1.3.5 – Взаємодія елементів Kanban

1.3.6 Переваги та недоліки розглянутих методів

Проведення детального аналізу переваг та недоліків кожного з методів – допоможе виділити ті методи які будуть підходити під критерії встановлені із огляду на специфіку використання новачками для здобуття досвіду та навчання. У таблиці 1.3.6 наведено недоліки та переваги методів керування проектами[10].

Таблиця 1.3.6 – Переваги та недоліки методів керування проектами

Метод керування проектами	Переваги	Недоліки
Scrum	<p>Гнучкість та адаптивність до змін, швидка реакція на вимоги ринку та змінені потреби стейкхолдерів.</p> <p>Підвищена залученість команди та стейкхолдерів завдяки регулярному зворотному зв'язку та співпраці,</p> <p>Прозорість робочого процесу, яка дозволяє всім учасникам бути в курсі стану проекту. Раннє виявлення проблем та помилок, що сприяє їх своєчасному вирішенню,</p> <p>Постійне вдосконалення процесів та продукту через ретроспективи.</p>	<p>Вимагає високого рівня дисципліни та залученості від команди. Може бути складним для масштабування у великих організаціях. Ризик перевантаження команди через недостатньо чітко визначені рамки спринту. Потребує досвідчених Scrum-майстрів для ефективного впровадження.</p>
Waterfall	<p>Чітка структура проекту та детальне планування, які спрощують управління та контроль. Легкість у визначенні ресурсів, бюджету та часових рамок завдяки детальному плануванню на початкових етапах,</p> <p>Висока прогнозованість результату проекту. Послідовність етапів сприяє детальній документації проекту.</p>	<p>Низька гнучкість та адаптивність до змін, призводить до затримок і збільшення вартості проекту.</p> <p>Ризик розриву між очікуваннями стейкхолдерів та кінцевим продуктом через відсутність регулярного зворотного зв'язку. Зростання витрат через потребу повертатися до попередніх фаз. Затримка тестування призводить до пізнього виявлення помилок та високих витрат на їх виправлення</p>

Продовження таблиці 1.3.6

Метод керування проектами	Переваги	Недоліки
Agile	<p>Висока гнучкість та адаптивність до змін, можливість швидко реагувати на змінені вимоги та умови ринку, Тісна співпраця зі стейкхолдерами та регулярний зворотний зв'язок забезпечують високу відповідність продукту очікуванням клієнта, Раннє та часте випускання продукту або його частин покращує час виходу на ринок, Постійний фокус на цінності для клієнта, пріоритизація завдань забезпечує ефективне використання ресурсів.</p>	<p>Може бути важко прогнозувати кінцевий бюджет та часові рамки проекту через його гнучкість, Вимагає високого рівня залученості та співпраці зі стейкхолдерами, що не завжди можливо, Потребує від команди високого рівня самоорганізації та самостійності.</p>
Stage-Gate	<p>Чітка структура процесу з визначеними "воротами" дозволяє ефективно управляти ризиками та рішеннями на кожному етапі, Забезпечує систематичний підхід до інновацій та розробки продуктів з регулярною оцінкою прогресу, Допомагає оптимізувати розподіл ресурсів та фокусуватися на найбільш перспективних проектах, Підтримує ретельне планування та документування на кожному етапі.</p>	<p>Може бути надто бюрократичним та громіздким для деяких проектів або швидкоплинних галузей, Недостатня гнучкість для адаптації до змін, які виникають під час розробки проекту, Процес "ворот" може сповільнити інноваційний процес та введення продукту на ринок.</p>

Метод керування проектами	Переваги	Недоліки
Kanban	<p>Підвищує прозорість робочого процесу та допомагає ідентифікувати затримки та вузькі місця в потоці роботи,</p> <p>Гнучкий та легко адаптується під різні типи проектів і команди будь-якого розміру,</p> <p>Обмеження кількості роботи в процесі допомагає зменшити перевантаження та покращує фокус на завданні.</p>	<p>Менш структурований у порівнянні з іншими методологіями, що може призвести до невизначеності в завданнях та ролях,</p> <p>Вимагає дисципліни та залученості від усіх членів команди для ефективного використання, інакше система може не працювати належним чином,</p> <p>Може не забезпечувати достатньої підтримки для великих, складних проектів з багатьма залежностями між завданнями.</p>

Кінець таблиці 1.3.6

1.4 Аналіз оточення розгортання програмного забезпечення

Розгортання програмного забезпечення для системи управління проектами, особливо в контексті комфорту доступу та комплексного функціоналу, має враховувати різні аспекти технічного оточення, щоб забезпечити ефективну взаємодію між користувачами та системою.

Серверне ПЗ (Backend) – це основа системи, яка обробляє логіку додатку, управління базами даних, аутентифікацію користувачів, обробку запитів тощо. Серверне ПЗ забезпечує обробку даних та взаємодію з іншими сервісами та ресурсами.

Клієнтська частина (Frontend) – це інтерфейс, через який користувачі взаємодіють із системою. Вона включає веб-сторінки та/або мобільні додатки, які відображають інформацію з сервера та надсилають користувацькі дані на сервер.

Система управління проектами, це програмне забезпечення або платформа, яка інтегрує вищезазначені елементи для управління проектами. Вона включає функції для планування, відстеження, спілкування, документування та звітності по проектах.

Інтеграція з зовнішніми сервісами, додаткові інструменти та сервіси, які можуть бути інтегровані з системою управління проектами, включно з електронною поштою, календарями, інструментами для спілкування (наприклад, Slack), системами контролю версій (наприклад, Git) тощо. Ці інтеграції розширюють функціональність системи та забезпечують гнучкість у роботі команд.

Інфраструктура та Хостинг, визначає, де і як буде розгорнуто програмне забезпечення. Може включати використання хмарних сервісів для гнучкості та масштабування, фізичних серверів у власному дата-центрі або гібридних рішень.

Модуль безпеки, відповідає за захист даних та інформації в системі, забезпечуючи аутентифікацію, авторизацію, шифрування даних, захист від несанкціонованого доступу та інші заходи безпеки.

Розробка та розгортання системи управління проектами, особливо для дистанційної роботи, вимагає врахування цих складових елементів для створення ефективного, безпечного та доступного рішення, яке задовольняє потреби користувачів та організації.

1.4.1 Аналіз імплементації програмних методів управління в систему

Далі варто приділити увагу детальному аналізу процесу вибору, адаптації та імплементації програмних методів управління в майбутню систему для дистанційного навчання. Варто розглянути ключові аспекти, що впливають на вибір методології, такі як Agile, Scrum, Kanban, та їх гібридні моделі, та обговоримо, як ці методи можуть бути інтегровані в архітектуру веб-сервісу, щоб забезпечити ефективне управління проектами, високу адаптивність до змін та оптимальне задоволення потреб користувачів.

Далі варто описати кроки розробки від визначення вимог до впровадження та моніторингу впроваджених методів, з акцентом на технічні, оперативні та стратегічні аспекти інтеграції. Важливість збору та дослідження відгуків користувачів після впровадження нових методів також підкреслюється як критичний етап, що дозволяє забезпечити неперервне вдосконалення системи. Розглянемо ці етапи.

Визначення вимог:

- аналіз потреб користувачів, розробник починає з збору вимог від кінцевих користувачів або аналізу ринкових потреб, щоб зрозуміти, які функції методу управління проектами потрібні;
- специфікація вимог, на основі зібраних даних розробник формулює специфікацію вимог, де детально описується, як метод управління проектами має бути представлений у системі.

Проектування:

- архітектура системи: розробник визначає архітектуру веб-сервісу, зокрема, як новий метод управління буде інтегруватися з існуючими модулями і базою даних;

- інтерфейс користувача: проектується інтерфейс, через який користувачі взаємодіятимуть з новим методом управління, забезпечуючи інтуїтивно зрозуміле та зручне використання.

Розробка:

- імплементація функціоналу: на цьому етапі розробник пише код, що реалізує всі плановані функції методу управління в рамках системи. це може включати створення нових баз даних, розробку API для взаємодії між модулями та інтеграцію з іншими інструментами;

- інтеграція: розробник інтегрує новий модуль з існуючою системою, забезпечуючи сумісність та правильну взаємодію компонентів.

Тестування:

- юніт-тестування: перевірка окремих компонентів методу на коректність роботи;

- інтеграційне тестування: тестування взаємодії нового методу з іншими частинами системи;

- тестування користувацького інтерфейсу: переконатися, що інтерфейс є зручним і зрозумілим для кінцевих користувачів.

Впровадження:

- розгортання: виведення оновленої системи на продакшн;

- моніторинг: спостереження за роботою нового методу на ранніх етапах впровадження для виявлення та усунення потенційних проблем.

1.5 Постановка задачі дослідження

Об'єктом дослідження в рамках кваліфікаційної роботи є методи

управління IT проектами для формування API імплементації у системи дистанційного навчання.

Предметом дослідження є адаптація та інтеграція гібридних методів управління проектами, таких як Agile, Scrum, та Kanban, для ефективної роботи в умовах навчання, а саме – дистанційного навчання.

Метою даної роботи є дослідження та вибір оптимальних гібридних методів управління IT проектами, які максимально враховують специфіку дистанційного навчання, для підвищення ефективності, гнучкості та успішності проектів.

Для досягнення мети, необхідно виконати наступні задачі:

- дослідження існуючих методів управління IT проектами: оцінка основних переваг та недоліків методологій agile, scrum, kanban та інших гібридних методів;
- дослідження специфіки управління IT проектами в системах дистанційного навчання: вивчення особливостей дистанційного навчання, що впливають на вибір та адаптацію методів управління проектами;
- адаптація гібридних методів управління для дистанційного навчання: розробка підходів до інтеграції та адаптації гібридних методів управління в контексті освітніх IT проектів;
- розробка критеріїв оцінки ефективності гібридних методів управління: визначення ключових показників ефективності (кпе) для оцінки успішності впровадження гібридних методів управління в IT проекти дистанційного навчання;
- експериментальна перевірка адаптованих методів: проведення експерименту за допомогою реальних IT проектів у системах дистанційного навчання для оцінки ефективності впроваджених гібридних методів управління.

2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ ГІБРИДНОГО УПРАВЛІННЯ

У цьому розділі буде досліджено гібридні методи управління, їх структура, поєднання двох підходів в один, також їх буде порівняно на рівні функціональних можливостей які вони надають. Також буде проведено критеріальне порівняння підходів для визначення їх відповідності до наших вимог.

Далі буде проведено тестування за допомогою бібліотеки Pandas на дата-сетах для визначення ризиків за формулою агрегованого ризику. Таким чином ми встановимо – який гібридний підхід має менше ризиків та є більш стабільним за інші.

Після цих досліджень, на завершальному етапі дослідження – буде проведено систематизоване дослідження що до елементів систем управління для гібридних систем спеціалізованих на дистанційному навчанні.

2.1 Критерії оцінки ефективності

Важливість встановлення чітких критеріїв оцінки ефективності управління проектами не може бути недооцінена. Визначення цих критеріїв слугує фундаментом для вибору найбільш ефективних елементів і підходів управління, забезпечуючи здатність проекту досягати своїх цілей в умовах змінних вимог і обмежень. Особливо це стає актуальним при розробці та впровадженні ІТ-

проектів у сфері дистанційної освіти, де потреба в інноваціях і адаптивності постійно зростає.

Складність та динамічність сучасних ІТ-проектів у сфері дистанційного навчання вимагають від управлінських команд високого рівня гнучкості та здатності до швидкої адаптації. Водночас, застосування гібридних підходів до управління, що поєднують в собі елементи як гнучких, так і традиційних методологій, ставить питання про вибір оптимальної стратегії. Якість та ефективність такого вибору безпосередньо залежать від коректно визначених критеріїв оцінки, які враховують специфіку проекту, його цілі, умови виконання та очікування замовників.

Критерії ефективності дозволяють оцінити не тільки кінцеві результати проекту, але й ефективність процесів управління на всіх його етапах. Вони включають показники часу, бюджету, якості, задоволення клієнтів, адаптивності проектною командою до змін та здатності проекту відповідати зростаючим вимогам ринку. Встановлення таких критеріїв на початковому етапі проекту створює основу для об'єктивної оцінки різних методологій та інструментів управління, а також сприяє розробці більш виважених та обґрунтованих рішень щодо їх застосування.

Визначення ефективних методів управління проектами в контексті навчання новачків та забезпечення ефективного порогу входження передбачає врахування низки ключових критеріїв. Кожен з них відіграє важливу роль у створенні умов, які сприяють швидкому освоєнню новачками основ управління проектами та їхньому подальшому професійному розвитку. Далі розглянуто ці критерії:

– наявність ролей більш досвідчених менеджерів: наявність у команді досвідчених менеджерів, які можуть поділитися своїми знаннями та досвідом, є

надзвичайно важливою. вони слугують як наставники для новачків, надаючи їм необхідну підтримку та орієнтири для ефективної роботи;

- стабільність завдань на проєкті: стабільність завдань забезпечує зрозумілість і передбачуваність робочого процесу для новачків. це допомагає їм краще адаптуватися та сконцентруватися на навчанні, не відволікаючись на постійно змінювані вимоги;

- чітка послідовність без перевантаження процесу: чітко визначена послідовність дій і процедур, яка водночас не є перевантаженою надмірною кількістю етапів або бюрократії, сприяє кращому розумінню процесів управління проєктами новачками;

- швидкий фідбек: можливість отримувати швидкий та конструктивний зворотний зв'язок дозволяє новачкам ефективно навчатися на власних помилках та успіхах, прискорюючи їх професійне зростання;

- низький поріг входу: низький поріг входу, тобто відсутність складних вимог до попередніх знань для розпочати роботи над проєктом, дозволяє новачкам швидше інтегруватися в процес і почати навчання, спираючись на практичний досвід.

Усі ці критерії допоможуть обрати ті елементи управління проєктами – які будуть задовольняти їм, та загалом буде сформовано базу для змішаного методу управління.

За цими характеристиками досліджено методи контролю за проєктами. Був проведений порівняльний аналіз методів, та спів ставлено елементи методів до критеріїв. У таблиці нижче ці перетини позначені зеленим кольором.

Результат дослідження наведений в таблиці 2.1, в якій представлено п'ять основних критеріїв:

- критерій K1, в якому акцент робиться на достатній зв'язок із досвідченою командою, команда повинна приймати активну участь;

- критерій К2, в якому акцент робиться на стабільність та чітку структурованість завдань на проєкті, щоб новачки не губились в постійних змінах та розгалуженні;
- критерій К3, акцент робиться на відсутності перевантажень як процесу так і учасників;
- критерій К4, який позначає швидкий фідбек від команди, що дозволить виправляти помилки швидко;
- критерій К5, низький поріг входу у систему контролю.

Порівнюємо кожний з методів на відповідність заданим критеріям, де символ «+» – це відповідність заданому критерію.

Таблиця 2.1 – Порівняння методів контролю проєктів

Методи інтерфейсної взаємодії	Порівняльні критерії				
	1	2	3	4	5
Scrum					
Waterfall					
Agile					
Stage-Gate					
Kanban					

Методологія Agile, використовуючи елементи, такі як Task Boards, Daily Stand-Ups, адаптивне планування та Sprint Backlog, ефективно задовольняє вищезазначеним критеріям К1, К2 та К4.

Критерій К1: зв'язок із досвідченою командою та активна участь, Daily Stand-Ups, щоденні короткі зустрічі дають можливість кожному члену команди,

включно з новачками, ділитися тим, над чим вони працювали, що планують робити далі та які перешкоди виникають. Це сприяє активній участі всіх членів команди і забезпечує безпосередній зв'язок із досвідченими колегами, які можуть надати пораду або допомогу.

Task Boards, візуалізація роботи на дошці задач дозволяє новачкам легше орієнтуватися у поточних завданнях команди та бачити внесок досвідчених колег, сприяючи кращому розумінню процесів та практик.

Критерій K2: стабільність та чітка структурованість завдань, Sprint Backlog визначення завдань, які потрібно виконати протягом спринту, забезпечує чітку структуру роботи. Новачки можуть легше адаптуватися, коли точно знають, які завдання їм потрібно виконати, та бачать чіткий поділ роботи на етапи.

Адаптивне планування дозволяє команді гнучко реагувати на зміни, одночасно зберігаючи структурованість завдань. Це допомагає уникнути хаосу та забезпечує стабільність навіть у динамічному середовищі проекту.

Критерій K4: швидкий фідбек, незважаючи на початкове припущення про відсутність цих елементів, вони є важливими для Agile, оскільки дозволяють команді швидко отримувати відгуки та вносити корективи. Ретроспективи спринту дають змогу команді аналізувати, що було зроблено добре, що можна покращити, та як це вплине на наступні завдання. Це допомагає уникнути перевантаження новачків інформацією та сприяє їх швидкому розвитку.

Елементи які стали ключовими для методу Agile:

- task boards;
- дейлі мітинги;
- daily stand-ups;
- фідбек і ретроспективи;
- адаптивне планування;
- sprint backlog.

Метод Kanban і його елементи, такі як дошки завдань з обмеженнями WIP (work in progress), простота процесу, та самі обмеження WIP, ефективно задовольняють критерії K3, K4, та K5.

Критерій K3: Відсутність перевантажень як процесу, так і учасників WIP Limit (Обмеження WIP), обмеження кількості завдань, над якими може працювати команда одночасно, безпосередньо впливає на зниження рівня перевантаження учасників і процесу в цілому. Це забезпечує, що команда не бере на себе більше роботи, ніж може ефективно виконати, тим самим знижуючи стрес і збільшуючи продуктивність.

Критерій K4: Швидкий фідбек від команди, дошки завдань (Task Boards), візуалізація поточного стану проекту на дошці завдань дозволяє команді швидко виявляти проблемні місця і надавати фідбек щодо роботи. Це сприяє оперативному вирішенню проблем і корекції помилок, а також підтримує постійний зворотний зв'язок між учасниками команди.

Критерій K5: Низький поріг входу у систему контролю, Kanban відрізняється своєю простотою і зрозумілістю, що робить його доступним навіть для новачків у сфері управління проектами. Відсутність складних правил і обмеження на кількість одночасних завдань спрощують процес втілення і адаптації до методу, знижуючи поріг входу.

Метод Waterfall, з його послідовною фазовою структурою та сильним акцентом на плануванні, може задовольнити критерії K1, K2, K3 та K5.

Критерій K1: Достатній зв'язок із досвідченою командою, активна участь команди.

Planning Stand (етап планування): На початковому етапі Waterfall, де команда збирається разом для детального планування проекту, новачки можуть вивчати від досвідчених колег. Активна участь у процесі планування дає новачкам змогу краще зрозуміти проект, його цілі та вимоги.

Критерій К2: Стабільність і чітка структурованість завдань.

Структуровані фази: Waterfall відомий своєю чіткою структурою фаз, що дозволяє новачкам легше орієнтуватися в завданнях та процесах проекту. Кожна фаза (планування, дизайн, реалізація, тестування, впровадження) виконується послідовно, що знижує ризик загублення серед постійних змін.

Критерій К3: Відсутність перевантажень.

Фазовий підхід: Оскільки кожна фаза в Waterfall вимагає завершення перед переходом до наступної, це допомагає уникнути перевантаження команди, дозволяючи зосередитись на одному етапі роботи за раз.

Критерій К5: Низький поріг входу у систему контролю.

Чіткі рекомендації та документація: Висока вимога до документації в Waterfall забезпечує детальний опис проекту, що може полегшити новачкам розуміння процесу та управління проектом. Це створює низький поріг входу, оскільки все планування та вимоги чітко документовані з самого початку.

2.2 Дослідження гібридних методологій

Далі варто порівняти гібридні підходи які складаються із обраних нами самостійних методологій, таких як Scrum, Agile, Waterfall та на концептуальному рівні – Kanban.

Після детального дослідження окремих методологій управління проектами, стає очевидною необхідність розгляду гібридних підходів у менеджменті. Це зумовлено кількома ключовими факторами, які виявилися значущими в сучасному динамічному середовищі управління проектами[11].

Перш за все, жодна із класичних методологій не є універсальною та не може бути ідеальною для всіх типів проектів. Наприклад, методологія Waterfall ефективна для проектів з чітко визначеними етапами та стабільними вимогами, тоді як Agile підходить для проектів, що потребують гнучкості та швидкої адаптації до змінюваних потреб користувачів. Однак, у багатьох сценаріях проект може вигодувати від комбінації обох підходів, що спонукає до розгляду гібридних методів.

Другий аспект — це здатність гібридних методологій задовольнити різні вимоги зацікавлених сторін в рамках одного проекту. У складних проектах, які включають багатофункціональні команди та різноманітні вихідні цілі, гібридні підходи можуть забезпечити необхідний баланс між строгістю планування та гнучкістю в реалізації задач.

Третій важливий аспект полягає у здатності гібридних методологій мінімізувати ризики. Використання різних методів управління в межах одного проекту дозволяє краще контролювати ризики, які можуть виникнути на різних стадіях реалізації проекту. Наприклад, строге планування на ранніх етапах за допомогою Waterfall може забезпечити стабільність та передбачуваність, тоді як впровадження елементів Agile на пізніших етапах дозволяє адаптуватися до змін та оптимізувати процеси.

Загалом, розгляд гібридних методів управління проектами надає можливість розробити більш гнучкі та ефективні стратегії для різних типів проектів, що є особливо актуальним у умовах постійних змін та інноваційних викликів сучасного бізнесу та освітнього середовища.

2.2.1 Water-Scrum-Fall

Методологія Water-Scrum-Fall (рисунок 2.2.1) поєднує традиційну методологію Waterfall із гнучким Scrum. Він базується на думці про те, що для проекту має бути структурна основа, яка забезпечується встановленою методологією Waterfall. У цей традиційний підхід до процесів інтегровані гнучкі фази. Складна частина проекту, розробка, виконується на основі методології Scrum.

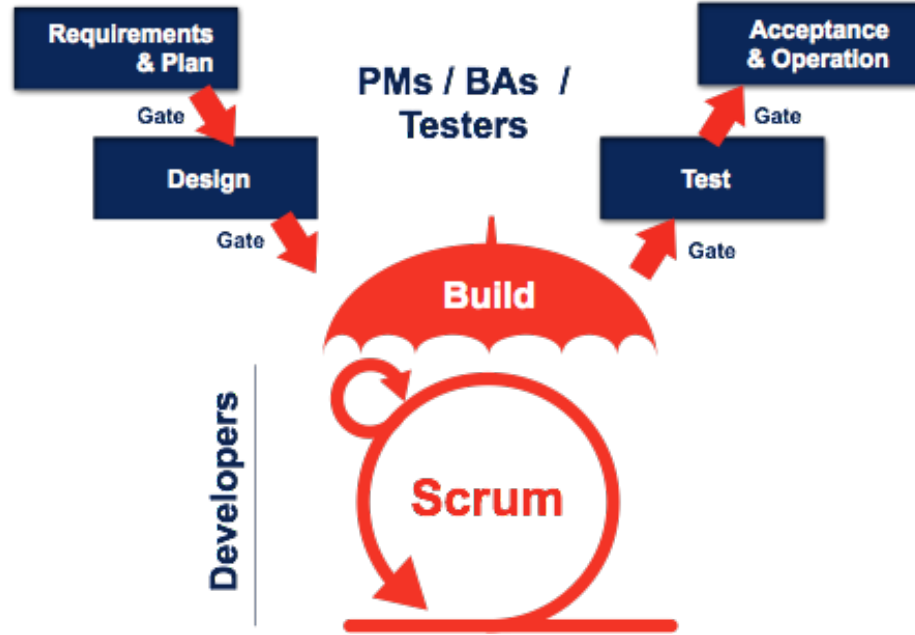


Рисунок 2.2.1 – Діаграма Water-Scrum-Fall.

Методологія Waterfall є найвідомішою та найпростішою моделлю процесу традиційного управління проектами, яка працює послідовно. Окремі етапи проекту виконуються один за одним. Щоб отримати готовий продукт у кінці, кожен етап має бути завершений, перш ніж переходити до наступного.

Також можна повернутися до попередньої завершеної фази, якщо потрібні коригування або виправлення. Scrum — одна з найвідоміших і найпопулярніших

гнучких методологій. Багато аспектів Scrum є загальними для гнучких методологій, таких як ітерації, поступовий розвиток, самокеровані команди та гнучкість перед мінливими вимогами.

Термін «ітеративний підхід» означає поділ тривалості проекту на ітерації або спринти, коли загальний проект ділиться на кілька невеликих проектів. Кожен спринт структурований таким же чином: на початку спринту команда вирішує, які функції будуть розроблені під час спринту, члени команди потім працюють над цими функціями, а в кінці спринту команда зустрічається з клієнтом, переглядати та отримувати відгуки про функції, розроблені під час спринту. Замовник бере активну участь у процесі розробки та може впливати на хід проекту протягом усього його терміну. Зазвичай спринт триває від двох до чотирьох тижнів.

Кожен спринт базується на резерві спринту, який описує набір пріоритетних функцій (або приростів продукту), які потрібно розробити в поточному спринті, вибраних тому, що вони мають високий пріоритет і можуть бути завершені протягом зазначеного періоду часу спринту. Під час спринту відставання від спринту не можна змінювати.

Scrum є адаптивним, оскільки дозволяє команді реагувати на постійно мінливі вимоги, ринкові ситуації, зміни в команді проекту та інші фактори. Зміна плану та обсягу проекту не потребує перегляду договору; зміни постійно коригуються. На початковому етапі методології Water-Scrum-Fall відбувається попередній етап планування проекту, під час якого готується гнучка реалізація та встановлюються плани щодо часу, бюджету та управління обсягом. Крім того, на цьому етапі створюються вимоги користувача та системні вимоги.

Фаза водоспаду вище за течією дає змогу мінімізувати початкові сумніви, оскільки створюється початкова документація та беруться початкові договірні зобов'язання. На другому етапі розробка здійснюється за гнучкими

методологіями. Реалізація відбувається в ітераційних кроках розробки. Це призводить до короточасних часткових результатів. Гнучка фаза мінімізує ризик переробки, затримок, перепланування та пропуску термінів.

У процесі Water-Scrum-Fall може сприяти розділенню між тестуванням і розробкою, щоб тестування стало частиною процесу випуску. Як тільки всі вимоги будуть реалізовані командою розробників, гнучка частина проекту закінчується. На завершальному етапі проекту гнучке рішення знову постачається за допомогою традиційної процедури, заснованої на встановленні контрольних воріт для зменшення частоти випусків програмного забезпечення. На даний момент продукт, як правило, уже певною мірою перевірений.

Це допомагає керувати тестами високого рівня та критеріями прийняття проекту. Water-Scrum-Fall — гарний вступ для компаній, які використовували традиційний підхід і зараз роблять перший крок до гнучкості. Це також вимагає нових ролей, яких раніше не існувало в компанії. Scrum вимагає, крім команди розробників, виконання ролей «Scrum Master» і «Product Owner», а також призначення конкретних прав і обов'язків. Сама команда є багатофункціональною, але немає чіткої чи чітко описаної архітектури команди.

2.2.2 Waterfall-Agile

На перший погляд методологія моделі Waterfall-Agile (рисунок 2.2.2) здається дуже схожою на методологію Water-Scrum-Fall від West. Однак ці дві методології відрізняються на кінцевій стадії. Наприклад, заключна фаза West базується на традиційному проектному підході. Остання фаза проекту, який використовує методологію Waterfall-Agile, все ще належить до гнучкого підходу.

План проекту визначений, а перший гнучкий спринт запланований до початку проекту. Для цього потрібен повний план проекту, але конкретні деталі кожного спринту не визначаються, доки не буде завершено перший спринт.

Розробка, проектування та впровадження базуються на гнучких методологіях. На кожній ітерації визначаються вимоги та запитуються відгуки клієнтів. Проводяться випробування та вносяться корективи, щоб забезпечити постійне вдосконалення. Вміст окремих фаз проекту розумно вибирається та розподіляється перед початком проекту, але може бути замінений у подальшому ході проекту відповідно до визначеної кількості зусиль.

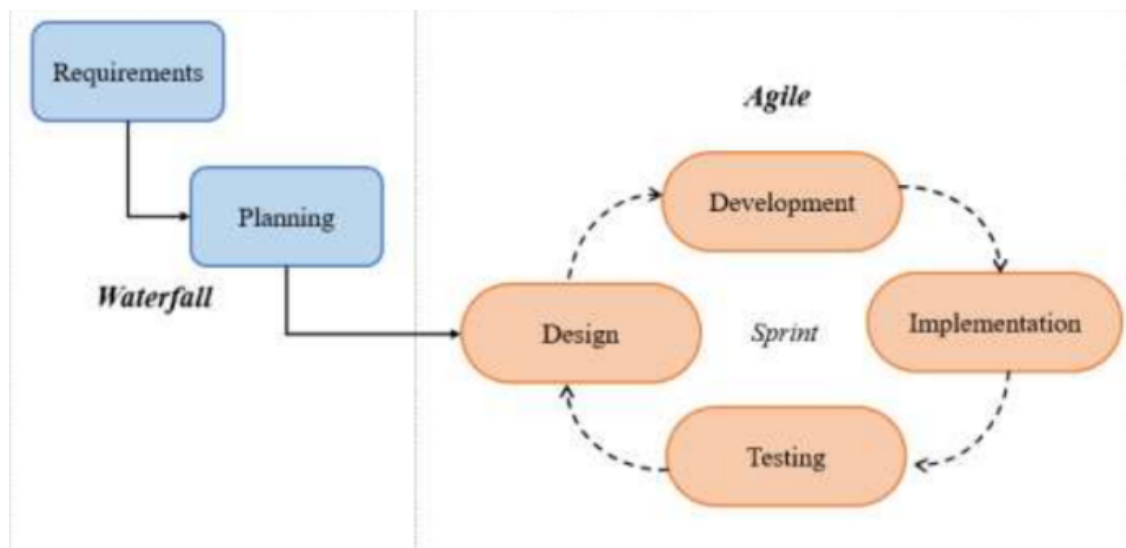


Рисунок 2.2.2 – Діаграма Waterfall-Agile.

Формальний і традиційний підхід використовується для визначення результату для кожної ітерації. Реалізація є дуже гнучкою на окремих етапах. Agile скорочує час доставки та полегшує збір відгуків на ранніх стадіях, щоб краще відповідати вимогам клієнтів. Розробка та тестування відбуваються за допомогою коротких швидких спринтів, які часто реалізуються за допомогою Scrum. За словами Хассані та інших, у Waterfall-Agile є різні ролі, які виконує команда проекту. У проектах традиційної форми менеджер проекту несе

відповідальність за планування та управління проектом, щоб він був успішно завершений, а цілі проекту були досягнуті. Керівник проекту забезпечує дотримання визначених робочих процесів і може реалізувати вимоги, зафіксовані на початку проекту, за визначених витрат і термінів. Розробники відповідають за обробку окремих робочих пакетів.

Тестер відповідає за моніторинг досягнення цілей якості та тестування продукту. Методологія Waterfall-Agile унікальна тим, що немає чіткої специфікації щодо того, де, коли або як традиційне планування переходить у гнучку реалізацію. Це те, що можна вирішити окремо для кожного окремого проекту[12]. Цілком можливо також, що проект визначається та планується за допомогою гнучкого підходу, а потім розробляється та реалізується відповідно до традиційних процедур.

2.2.3 Дослідження ризиків гібридних підходів

Загалом було досліджено переваги та недоліки складових елементів розглянутих вище комплексних підходів, тож надалі варто дослідити ризики які можуть нести в своєму процесі дані методи.

Після розгляду ризиків можна буде остаточно встановити кращий комплексний підхід. Оцінка ризиків буде проведена за допомогою математичного аналізу та побудови математичної моделі. Таким чином буде прораховано на коефіцієнтному рівні ризики кожного підходу.

Надалі варто створити модель, яка оцінює ймовірність неуспішного завершення проекту з урахуванням різних ризиків, пов'язаних із кожним підходом. Модель може включати такі параметри, як зміна вимог, проблеми з

ресурсами, затримки у графіках тощо. Використання статистичних методів або імітаційного моделювання дозволить оцінити, як різні стратегії управління реагують на ці зміни.

Для розробки моделі оцінки ризиків гібридних методологій управління проектами, таких як Waterfall+Agile та Waterfall+Kanban, можемо використовувати комбінацію кількісних і якісних методів. Ось основні кроки для розробки такої моделі:

Визначення ризиків – спочатку необхідно ідентифікувати ризики, які можуть впливати на проекти, керовані за допомогою обраних методологій. Ці ризики можуть включати:

- технологічні ризики: зміни в технологіях, залежності від стороннього пз;
- ризики управління: недостатня комунікація, невизначеність управлінських рішень;
- ринкові ризики: зміни у вимогах замовників, зміни в ринковому середовищі;
- оперативні ризики: затримки в проекті, проблеми з якістю робіт.

Оцінка ймовірності і впливу – кожен ризик оцінюється за двома основними параметрами. Ймовірність виникнення ризику (наприклад, від 0 до 1, де 0 - неможливо, 1 - неодмінно відбудеться).

Вплив ризику на проект (може бути оцінений за шкалою від мінімального до катастрофічного).

Розробка математичної моделі – далі варто використати модель оцінки ризиків, яка враховує сукупний вплив всіх ризиків на проект. Одна з можливих моделей - це використання моделі PERT (Program Evaluation Review Technique) для розрахунку очікуваного часу завершення проекту з урахуванням ризиків:

$$E(T) = \frac{(0+4m+p)}{6}, \quad (2.2.3)$$

де o – оптимістичний час завершення (без урахування ризиків);
 m – найімовірніший час завершення;
 p – песимістичний час завершення (з урахуванням можливих затримок через ризики).

Розрахунок агрегованого ризику – варто також розрахувати агрегований ризик для проекту, склавши ймовірності та впливи всіх ризиків:

$$R = \sum_{i=1}^n p_i \times u_i, \quad (2.2.4)$$

де p_i – вплив i -го ризику на проект;
 u_i – ймовірність i -го ризику.

Надалі варто провести дослідження та вивести рішення на основі отриманих результатів ви можете визначити, які ризики потребують негайних дій для мінімізації або ж які аспекти управління проектом потребують змін.

Ця модель буде адаптована і деталізована залежно від конкретних умов та потреб певного проекту у навчальних цілях чи дослідженнях.

Для проведення оцінки ризиків використаю модель, описану вище.

Дані для проведення дослідження та оцінки ризиків було взяті дані із статистичних сетів даних з досліджень ISACA[13], дослідження OWASP[14] та була зібрана статистика із досвіду компанії X1Group[15], хоча дані із компанії не є публічними та будуть опубліковані чи представлені, буде представлено тільки результати статистичних критеріїв.

Розглянемо типові ризики в IT-проектах, використовуючи гібридні методології Waterfall+Agile та Waterfall+Scrum-Fall.

Параметри для дослідження:

- технологічні ризики;
- організаційні ризики;
- комунікаційні ризики;
- ринкові ризики;
- оперативні ризики.

Ймовірність і вплив ризиків:

- ймовірність розглядаємо від 0.1 (дуже низька) до 1.0 (неодмінно відбудеться);
- вплив оцінюємо від 1 (мінімальний) до 5 (критичний).

Надалі було проведено дослідження дата сетів, та отримана статистична оцінка у вигляді коефіцієнтів ризику та імовірності. Дослідження даних було проведено за допомогою проекту з відкритим вихідним кодом з можливістю модифікації Pandas-dev[16].

Pandas — це відкритий проект програмного забезпечення, що надає бібліотеку на мові Python для аналізу даних.

Основні структури даних у Pandas включають DataFrame та Series. DataFrame – це двовимірна мутабельна таблична структура з потенційно гетерогенними стовпцями, а Series — це одновимірний масиво-подібний об'єкт.

Маніпуляція даними дозволяє виконувати широкий спектр операцій з ними, включаючи зміну форми дата-сетів, сортування, фільтрацію, агрегацію та більш складні обчислення.

Проект підтримує читання та запис у різноманітні формати даних, включаючи CSV, Excel, SQL бази даних, JSON та багато інших.

Обробка часових рядів: Бібліотека має розширені можливості для аналізу часових рядів, включаючи перескакування по часових рядах, періодичність і конвертацію часових зон.

Pandas активно використовується у фінансовому аналізі для обробки і дослідження фінансових даних, в наукових дослідженнях для обробки експериментальних даних, а також у веб-аналітиці для обробки і візуалізації даних з веб-ресурсів.

Pandas є проектом з відкритим вихідним кодом, і він розвивається за підтримки активної спільноти розробників та користувачів. Бібліотека постійно оновлюється і вдосконалюється, що робить її одним з найпопулярніших інструментів для роботи з даними у Python.

За результатами дослідження було виведено середні значення ризиків для кожного із гібридних підходів. Отже на рисунку 2.2.3 зображено коефіцієнти ризиків Water-Scrum-Fall.

```

Program build started...
Program build executed successfully.

Technological Risk Analysis initialized...
Technological Risk Analysis executed successfully:
  Probability - 0.40
  Impact Level - 4
  Description - Changes in the technological stack of the project

Organizational Risk Analysis initialized...
Organizational Risk Analysis executed successfully:
  Probability - 0.30
  Impact Level - 3
  Description - Changes in management structure

Communication Risk Analysis initialized...
Communication Risk Analysis executed successfully:
  Probability - 0.50
  Impact Level - 3
  Description - Problems with internal communication

Market Risk Analysis initialized...
Market Risk Analysis executed successfully:
  Probability - 0.20
  Impact Level - 4
  Description - Changes in market conditions

Operational Risk Analysis initialized...
Operational Risk Analysis executed successfully:
  Probability - 0.60
  Impact Level - 5
  Description - Delays in development or testing

Final results:
Technology Risk Probability: 0.40
Technology Risk Impact: 4
Organizational Risk Probability: 0.30
Organizational Risk Impact: 3
Communication Risk Probability: 0.50
Communication Risk Impact: 3
Market Risk Probability: 0.20
Market Risk Impact: 4
Operational Risk Probability: 0.60
Operational Risk Impact: 5

Program executed successfully.
Total errors : 0
turn go(f, seed, [])

```

Рисунок 2.2.3 – Результат дослідження ризиків Water-Scrum-Fall

Технологічні ризики зосереджені навколо можливих змін у технологічному стеку проекту, що може вимагати адаптації під час розробки чи навіть відкладення проектних планів. Зміни можуть бути викликані новими технологічними відкриттями або змінами у вимогах до продукту, що може вплинути на весь хід проекту.

Організаційні ризики пов'язані зі змінами у структурі управління, що може включати зміни в керівництві або реструктуризацію компанії.

Недоліки у комунікації можуть вплинути на узгодженість розуміння проектних цілей.

Оперативні ризики включають затримки у розробці або тестуванні, що є особливо критичними у проектах, де дотримання графіка є ключовим. Такі затримки можуть серйозно вплинути на загальний успіх проекту та його фінансові показники.

Для більшого розуміння надамо результати у таблиці 2.2.3

Таблиця 2.2.3 – Результат дослідження ризиків Water-Scrum-Fall

Ризики	Ймовірність	Вплив	Опис
Технологічні	0.4	4	Зміни в технологічному стеку проекту
Організаційні	0.3	3	Зміни в структурі управління
Комунікаційні	0.5	3	Проблеми із внутрішньою комунікацією
Ринкові	0.2	4	Зміни в ринкових умовах
Оперативні	0.6	5	Затримки в розробці або тестуванні

Далі було проведено розрахунки за формулою агрегованого ризику:

$$R = (0.4*4)+(0.3*3)+(0.5*3)+(0.2*4)+(0.6*5)$$

$$R = 7.8$$

Загальний ризик проекту складає 7.8, що є не досить значним показником і вимагає додаткових заходів щодо мінімізації ризиків.

Для методології Waterfall-Agile ми досліджуємо ті самі категорії ризиків, що і в Waterfall-Scrum-Fall. Результати цього підходу наведено на рисунку 2.2.4.

```

Program build started ...
Program build executed successfully.

Technological Risk Analysis initialized...
Technological Risk Analysis executed successfully:
  Probability - 0.30
  Impact Level - 3
  Description - Changes in the technological stack of the project

Organizational Risk Analysis initialized...
Organizational Risk Analysis executed successfully:
  Probability - 0.20
  Impact Level - 2
  Description - Changes in management structure

Communication Risk Analysis initialized...
Communication Risk Analysis executed successfully:
  Probability - 0.40
  Impact Level - 2
  Description - Problems with internal communication

Market Risk Analysis initialized...
Market Risk Analysis executed successfully:
  Probability - 0.10
  Impact Level - 3
  Description - Changes in market conditions

Operational Risk Analysis initialized...
Operational Risk Analysis executed successfully:
  Probability - 0.50
  Impact Level - 4
  Description - Delays in development or testing

Final results:
Technology Risk Probability: 0.30
Technology Risk Impact: 3
Organizational Risk Probability: 0.20
Organizational Risk Impact: 2
Communication Risk Probability: 0.40
Communication Risk Impact: 2
Market Risk Probability: 0.10
Market Risk Impact: 3
Operational Risk Probability: 0.50
Operational Risk Impact: 4

Program executed successfully.
Total errors : 0
turn go(f, seed, [])

```

Рисунок 2.2.4 – Результат дослідження ризиків Water- Agile

Для більшого розуміння надамо результати у таблиці 2.2.4

Таблиця 2.2.4 – Результат дослідження ризиків Water-Scrum-Fall

Ризики	Ймовірність	Вплив	Опис
Технологічні	0.3	3	Зміни в технологічному стеку проекту
Організаційні	0.2	2	Зміни в структурі управління
Комунікаційні	0.4	2	Проблеми із внутрішньою комунікацією
Ринкові	0.1	3	Зміни в ринкових умовах
Оперативні	0.5	4	Затримки в розробці або тестуванні

Далі було проведено розрахунки за формулою агрегованого ризику:

$$R = (0.3*3)+(0.2*2)+(0.4*2)+(0.1*3)+(0.5*4)$$

$$R = 4.4$$

Загальний ризик проекту складає 4.4, що є значно нижче за попередній підхід, а значить – значно ефективніший.

2.3 Фільтрація елементів методів управління проектами

Було порівняно переваги та недоліки складових, ключових елементів методів управління проектами:

- task boards;
- дейлі мітинги;
- daily stand-ups;
- фідбек і ретроспективи;

- адаптивне планування;
- sprint backlog;
- task boards(wip limit);
- planing stand;
- gates.

Основна ідея KANO- дослідження у полягає в тому, щоб класифікувати атрибути продукту або функції за тим, як вони впливають на задоволення користувачів і як їх відсутність чи наявність впливає на сприйняття продукту.

Ключовими елементами KANO- дослідження є:

- базові (Must-be Quality Attributes) – це атрибути, які користувачі приймають як само собою зрозумілі, і їх відсутність призводить до незадоволеності. Наприклад, якщо у смартфоні відсутня можливість здійснювати телефонні дзвінки, користувачі будуть дуже незадоволені;

- перформанс-атрибути (Performance Quality Attributes) – це атрибути, які прямо впливають на рівень задоволеності користувача. Їх наявність призводить до підвищення задоволеності, а відсутність – до зменшення задоволеності. Наприклад, якість камери в смартфоні;

- захоплюючі атрибути (Excitement Quality Attributes) – це атрибути, які можуть призвести до несподіваного підвищення задоволеності користувача, коли вони присутні, але не призводять до незадоволення, коли їх немає. Наприклад, додаткові функції або особливості, які приємно вражають користувачів.

KANO-дослідження також враховує залежності між атрибутами і задоволенням користувачів. Наприклад, з плином часу, атрибути, які раніше були захоплюючими, можуть стати базовими, оскільки користувачі розвивають нові очікування.

Для проведення KANO- дослідження можуть використовуватися різні методи, включаючи опитування користувачів та дослідження їх відгуків, опитувальники та експертні оцінки.

У проекті в опитуванні приймали участь робітники, переважно із менеджерських посад із досвідом щоб встановити відповідність елементів для комфортного прогу входження для навчання та необхідність тих чи інших елементів для загальної цілісності проекту.

Далі необхідно сформулювати по 2 питання до кожного елементу, користувач буде оцінювати кожний метод за шкалою зацікавленості у реалізації того чи іншого методу від 1 до 10, де: 1 – це абсолютно не зацікавлений, 10 – наймовірно зацікавлений:

- чи вважаєте ви, що дошки завдань дозволяють новим членам команди легше адаптуватися та швидше вчитися;
- як впливають дошки завдань на загальну цілісність і ефективність системи управління проектами;
- чи сприяють щоденні мітинги швидкому залученню новачків до процесу роботи команди;
- як щоденні мітинги впливають на цілісність управління проектами в організації;
- чи допомагають щоденні стендапи новачкам швидше інтегруватися в команду;
- який вплив щоденних стендапів на ефективність та синхронізацію роботи команди;
- чи забезпечує процес збору фідбеку та проведення ретроспектив легкий вхід та навчальну підтримку для новачків;
- як впливають фідбек та ретроспективи на підвищення цілісності процесу управління проектами;

- чи полегшує адаптивне планування процес залучення та навчання нових членів команди;
- як адаптивне планування впливає на гнучкість та ефективність системи управління проектами;
- чи спрощує спринт-беклог вхід у проект для нових учасників та їх орієнтування у завданнях;
- як наявність спринт-беклогу впливає на організаційну цілісність проектних процесів;
- чи допомагають обмеження вір на дошках завдань новачкам легше орієнтуватися у поточних задачах;
- який вплив мають обмеження вір на дошках завдань на загальну продуктивність та управління проектом;
- чи полегшує стенд для планування процес адаптації для нових членів команди та їх залучення в планування проекту;
- як стенд для планування впливає на ефективність управління проектами та їх цілісне виконання;
- чи спрощують ворота проекту процес входження для нових учасників, надаючи чіткі етапи та перевірки;
- який вплив мають ворота проекту на загальну цілісність і ефективність системи управління проектами.

Опитування користувачів проводилось через соціальні мережі (Twitter, Facebook) та інші платформи та сервіси.

Ви можете знайти посилання на форму з опитуваннями тут:

https://docs.google.com/forms/d/e/1FAIpQLSfJIbO2zN6_CPQPWurzHnQkbrRX46XpSjbQK2blM2JDMuOwrw/viewform?usp=sf_link

Нижче наведені гістограми(див. рис. 2.3) з результатами проведення опитувань для усіх питань де можна побачити тенденції важливості того чи іншого функціоналу за вибором користувачів.

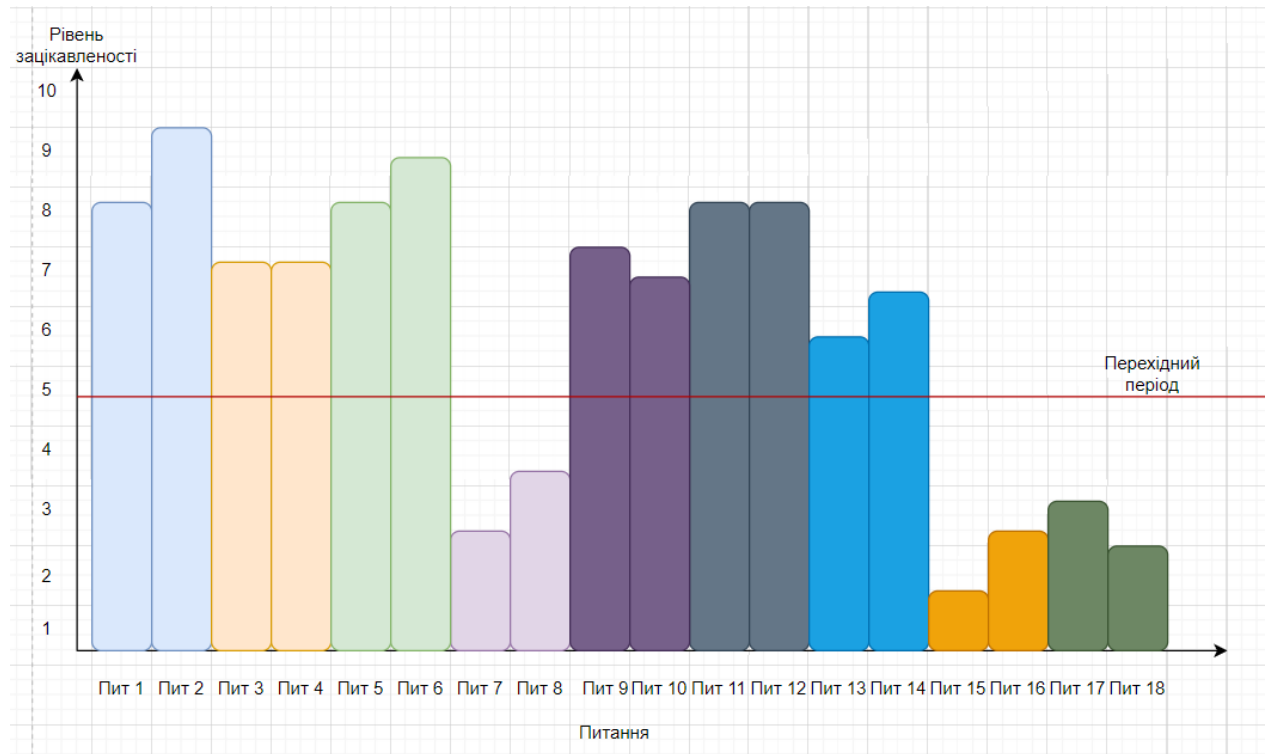


Рисунок 2.3 – Гістограма оцінки користувачами елементів

За результатами опитування та дослідження результатів було встановлено що ключовими елементами методів управління які задовольняють встановленим вимогам є наступні елементи:

- task boards(wip limit);
- wip limit;
- daily stand-ups;
- дейлі мітинги;
- sprint backlog;
- planing stand.

За загальними результатами опитування можна встановити що всі ключові елементи співпадають або є частиною методологій Agile та Waterfall, також варто зазначити що нами було обрано візуальну складову методу Kanban, а саме його таблиці для більшої наочності та структурованості робочого процесу, також варто відмітити що даний підхід збігається із гібридною методологією Waterfall-Agile.

Надалі буде розроблено модель системи із урахуванням досліджень проведених раніше.

3 ОПИС І ВИКОРИСТАННЯ МЕТОДУ УПРАВЛІННЯ ПРОЕКТАМИ

У даному розділі буде розібрано поєднання обраних методологій та обраних структурних елементів в один гібридний метод управління.

Далі буде описано інноваційний підхід взаємодії із штучним інтелектом через застосування API для впровадження інтерактивних посилань на чати із штучним інтелектом для більш зручної взаємодії між цими ресурсами для отримання інформації у системі керування проектом.

По завершенню буде описано планування для розробки системи та її складових частин.

3.1 Поєднання методів Agile, таблиці Kanban та Waterfall

Поєднання Agile, Kanban та Waterfall у єдину гібридну систему управління проектами може пропонувати унікальну суміш гнучкості, візуалізації та структурованості. Таке поєднання здатне задовольнити різні потреби проекту на різних його етапах і пропонує наступні переваги:

- візуалізація робочого процесу (Kanban). Kanban відомий своєю здатністю візуалізувати поточні завдання та процеси за допомогою дошки завдань. Це дозволяє командам швидко оцінити стан проекту, визначити затори в робочих процесах і відповідно розподілити ресурси. Візуалізація робочого процесу сприяє кращій координації дій і ефективності роботи команди;

- гнучкість і адаптивність (Agile). Agile пропонує швидку адаптацію до змін у проекті, забезпечуючи високу реактивність команди на нові вимоги або

проблеми. Ітеративні спринти та постійний зворотний зв'язок зі стейкхолдерами дозволяють постійно вдосконалювати продукт і оптимізувати робочий процес;

– структуроване планування та впровадження (Waterfall). Waterfall вносить чіткість і передбачуваність у початкові етапи планування та завершальні етапи впровадження проекту. Визначення обсягу робіт, вимог і цілей проекту на ранньому етапі, а також структуроване впровадження допомагає забезпечити, що ключові цілі проекту будуть досягнуті.

Гнучкість Agile дозволяє швидко реагувати на зміни, підтримуючи інноваційність та креативність команди.

Візуалізація Kanban підтримує ефективне управління завданнями і ресурсами, спрощуючи координацію між учасниками проекту.

Структурованість Waterfall забезпечує чітке розуміння цілей проекту та вимог до нього з самого початку, а також забезпечує якісне завершення проекту.

Таким чином, гібридний підхід, що об'єднує Agile, Kanban і Waterfall, може забезпечити баланс між потребою в швидкій адаптації до змін, ефективною координацією завдань і структурованим плануванням та впровадженням. Цей підхід дозволяє використовувати сильні сторони кожної методології в залежності від поточних потреб проекту та стадії його розвитку:

На початковому етапі Waterfall допомагає чітко визначити обсяг проекту і розробити детальний план, що знижує ризик непорозумінь і невідповідності у майбутньому.

Протягом проекту Kanban забезпечує неперервну візуалізацію прогресу і допомагає в управлінні пріоритетами та ресурсами, а також підвищує прозорість процесів для всієї команди.

Agile, в свою чергу, робить акцент на гнучкості та швидкості адаптації до змін, включаючи регулярний зворотний зв'язок від замовника, що дозволяє оперативно вносити корективи і оптимізувати робочий процес.

В результаті, поєднання Agile + Kanban + Waterfall може створити міцну основу для управління різноманітними проектами, від простих і передбачуваних до складних і динамічних. Цей гібридний підхід дозволяє ефективно масштабувати управління проектами, адаптувати його під конкретні потреби та особливості проекту, забезпечуючи при цьому високу якість виконання та задоволення вимог замовника.

3.2 Новизна імплементації взаємодії із OpenAI

Варто відзначити та описати новітнє рішення для систем управління проектами, яке дозволяє користувачам вносити у базу даних посилання на конкретний чат ChatGPT, або будь якого іншого джерела подібного функціоналу що після цього прив'язується до певного об'єкту на фронт-енді. Це рішення стало значущим кроком вперед у забезпеченні індивідуальної інтеграції між користувачами та системою, а також у персоналізації їхнього досвіду користування системою управління проектами.

Основною ідеєю цього підходу є забезпечення високого рівня контекстуальної взаємодії з користувачами, що є особливо важливим у складних та динамічних проектах. Завдяки прив'язці посилань на діалоги з ChatGPT до конкретних елементів фронт-енду, користувачі мають можливість отримувати миттєвий доступ до відповідної інформації, необхідної для виконання завдань або прийняття рішень. Це не тільки покращує ефективність робочих процесів, але й значно зменшує час на пошук потрібних даних.

Крім того, такий підхід значно підвищує рівень безпеки інформації, оскільки доступ до посилань на чати мають лише ті користувачі, яким це

дозволено відповідно до їх ролей та прав у проекті. Це забезпечує додатковий рівень контролю над тим, хто може бачити чи редагувати ключову інформацію.

Загалом, реалізація цього рішення в системах управління проектами відкриває нові перспективи для оптимізації управлінських процесів і створення більш адаптивних і інтуїтивно зрозумілих інструментів, що відповідають унікальним вимогам користувачів у широкому спектрі індустрій. Ми вважаємо, що це рішення має великий потенціал для подальшого розвитку та імплементації в різних секторах, де потрібне високоефективне управління проектами.

Для реалізації системи, де користувачі можуть вносити у базу даних посилання на чат ChatGPT, яке прив'язується до певного об'єкту на фронт-енді, потрібно створити декілька ключових компонентів, включаючи бекенд логіку, базу даних, інтеграцію з ChatGPT, та користувацький інтерфейс. Далі наведемо покроковий план реалізації.

Модель бази даних має включати наступні елементи:

- user: таблиця для зберігання інформації про користувачів;
- project: таблиця для проектів, які міститимуть посилання на чати;
- chatlink: таблиця, яка зберігає посилання на чати, пов'язані з проектами, ідентифікатор користувача, що додав посилання, і мітки доступу.

Створення RESTful API для управління даними:

- endpoints для створення, читання, оновлення та видалення (crud) для кожного з ресурсів (користувачі, проекти, посилання на чати);
- аутентифікація та авторизація для забезпечення безпечного доступу до арі.

Інтеграція з ChatGPT:

- інтеграція з openai api для забезпечення можливості створення чатів;
- реалізація логіки для генерації та зберігання унікальних url посилань на чати.

Розробка фронт-енду:

- створення інтерфейсу для введення посилань на чати та їх прив'язки до проектів;
- реалізація функціоналу для перегляду та управління посиланнями на чати в контексті проектів;
- захист об'єктів таким чином, щоб доступ до них мав лише авторизований користувач.

На рисунку 3.1 наведено діаграму діяльності користувача для взаємодії із цим функціоналом.

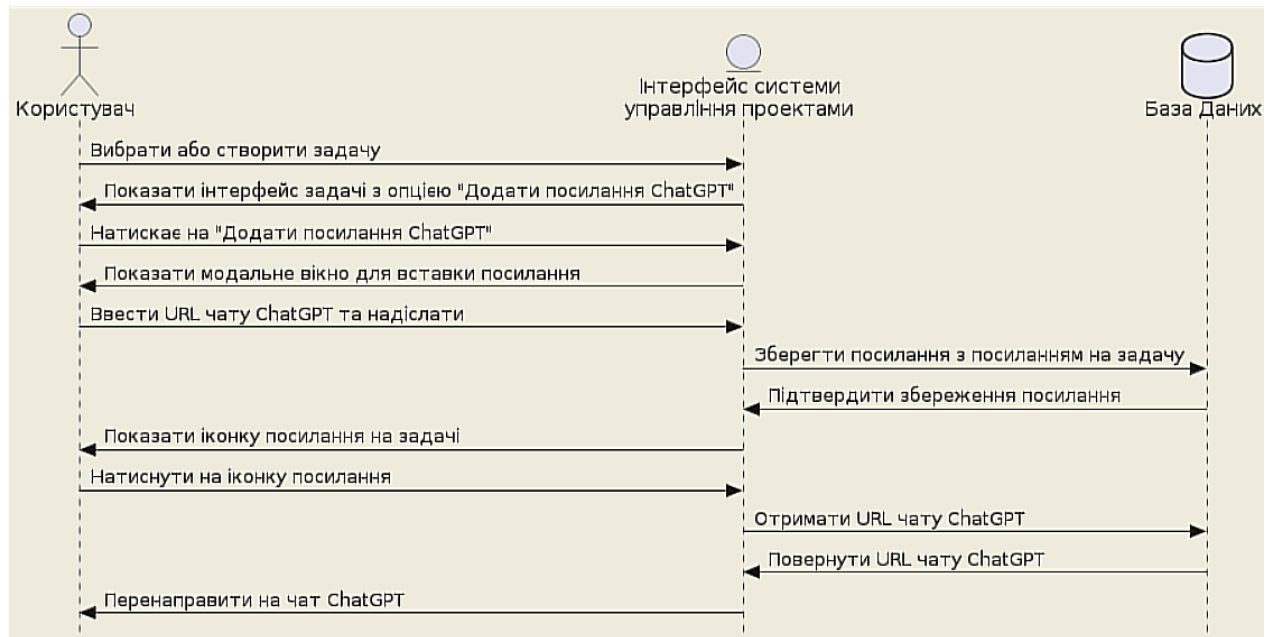


Рисунок 3.1 – Діяльність користувача із новітнім функціоналом посилань

На рисунку 3.2 наведено приклад інтерфейсу реалізації цього функціоналу.

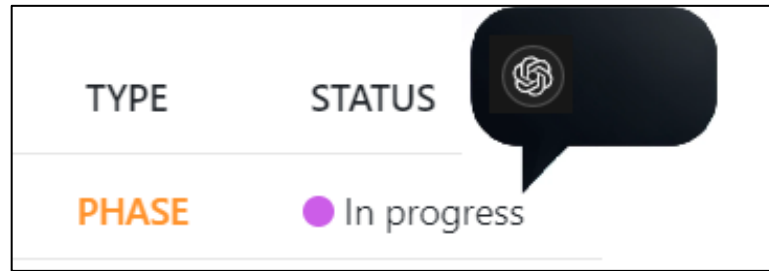


Рисунок 3.2 – UI реалізації взаємодії із чатами

Спосіб прикріплення посилання на чат ChatGPT до ресурсів у системі управління проектами є інноваційним і значно вдосконалює традиційні Kanban-дошки з кількох важливих причин:

Традиційні Kanban-дошки забезпечують візуалізацію завдань і потоку робіт, але не завжди мають вбудовані інструменти для інтерактивного спілкування. Наш підхід інтегрує посилання на чати ChatGPT безпосередньо в завдання, що дозволяє користувачам миттєво переходити до відповідних діалогів і обговорень, які можуть допомогти вирішити питання або уточнити деталі завдання.

Додавання посилань на чати безпосередньо до завдань або інших елементів на Kanban-дошці забезпечує зберігання контексту всередині системи управління проектами. Користувачі можуть легко отримати доступ до історії обговорень і збережених рішень, що значно підвищує ефективність роботи і знижує час на пошук необхідної інформації.

У висновку можна сказати що впровадження функціоналу прикріплення посилань на чати ChatGPT до елементів на Kanban-дошці є значущим вдосконаленням, яке додає інтерактивність, покращує колаборацію, забезпечує контекстуальність інформації та підвищує безпеку та ефективність управління проектами.

Цей підхід дозволяє зробити процес управління проектами більш динамічним і адаптованим до потреб сучасних команд.

3.3 Опис реалізації структурних елементів

Система з Task Boards та WIP (Work In Progress) лімітами може бути реалізована за допомогою цифрової платформи, яка дозволяє командам управляти своїми проектами в режимі реального часу.

Структура Task Board:

- створення колонок: task board поділяється на декілька колонок, які відображають різні стадії процесу роботи (наприклад, "до виконання", "в роботі", "перевірка", "завершено"). користувачі можуть налаштувати назви та кількість колонок відповідно до своїх потреб;
- додавання завдань: у кожену колонку можна додавати картки завдань, що містять опис завдання, відповідальних осіб, терміни виконання та іншу релевантну інформацію;
- переміщення завдань: картки можна легко переміщувати між колонками, що відображає прогрес у виконанні завдань.

Реалізація WIP лімітів, налаштування WIP Лімітів: Для кожної колонки можна встановити максимальну кількість завдань, які можуть бути в роботі одночасно. Це допомагає уникнути перевантаження команди та фокусує увагу на завершенні розпочатих завдань.

Візуальні Попередження: система автоматично відображає візуальні попередження або повідомлення, коли кількість активних завдань у колонці

перевищує встановлений WIP ліміт, нагадуючи команді про необхідність зосередитись на завершенні поточних завдань.

Додаткові Функції:

- фільтрація та пошук: команда може використовувати фільтри для відображення лише тих завдань, які відповідають певним критеріям (наприклад, за відповідальною особою, терміном виконання тощо);

- інтеграція з календарем: завдання можуть бути синхронізовані з календарем команди для кращого планування робочого часу та візуалізації майбутніх дедлайнів;

- аналітика та звіти: система може генерувати звіти про прогрес проекту, продуктивність команди та дослідити час виконання завдань, допомагаючи ідентифікувати потенційні зони для покращення та оптимізації робочих процесів;

- зворотний зв'язок та коментарі: кожна картка завдань може мати розділ для коментарів, де учасники команди можуть залишати зворотний зв'язок, задавати питання або обговорювати деталі завдань, сприяючи кращій комунікації в команді;

- історія змін: система веде лог усіх змін для кожної картки завдань, дозволяючи відстежити хто та коли зробив певні зміни чи оновлення, що забезпечує прозорість процесу роботи.

Реалізація WIP (Work In Progress) лімітів у системі управління проектами може включати обмеження не тільки за кількістю завдань, які можуть виконуватися одночасно, але й за кількістю завдань, призначених конкретним членам команди, та за пріоритетами завдань.

WIP ліміти для робітників:

- персональні обмеження: кожен член команди має власний wip ліміт, який визначає максимальну кількість завдань, над якими він може працювати

одночасно. це допомагає запобігти перевантаженню працівників та сприяє зосередженості на завершенні розпочатих завдань;

- автоматичні сповіщення: коли кількість призначених завдань наближається до wip ліміту члена команди, система автоматично відправляє сповіщення як цьому користувачу, так і менеджеру проекту. це дозволяє своєчасно внести корективи в розподіл роботи;

- налаштування лімітів: менеджер проекту або самі працівники можуть налаштувати свої WIP ліміти в залежності від їхньої робочої спроможності та поточного навантаження, що забезпечує гнучкість та адаптацію до змінних умов проекту.

WIP ліміти за пріоритетами:

- визначення пріоритетів: завдання в системі класифікуються за пріоритетами (наприклад, високий, середній, низький). wip ліміти можуть бути різними для кожної категорії пріоритету, що дозволяє зосередитися на найважливіших завданнях;

- динамічне налаштування: у разі зміни пріоритетів завдань або необхідності адаптації до критичних ситуацій у проекті, wip ліміти за пріоритетами можуть динамічно змінюватися. це дозволяє ефективно реагувати на поточні потреби проекту.

Реалізація Sprint Backlog, Sprint Backlog є ключовим елементом в Agile та Scrum методологіях, який представляє список завдань, обраних для реалізації в наступному спринті. Виділимо ключові елементи:

- інтерактивна дошка завдань: розробити інтерактивну дошку завдань, де можна буде створювати, редагувати та переміщувати картки завдань. картки можуть містити детальну інформацію про завдання, включаючи опис, відповідальних осіб, естимейт трудовитрат і пріоритет;

- планування спринту: інтегрувати функціонал для проведення

планувальних засідань, на яких команда обирає завдання з product backlog та переносить їх до sprint backlog. цей процес може бути підтриманий функцією "тягни-та-пускай" для легкого перенесення завдань між беклогами;

- визначення лімітів: надати можливість встановлення обмежень за обсягом роботи для спринту на основі величини естимейтів та поточної продуктивності команди (velocity);

- моніторинг прогресу: вбудувати інструменти для відстеження прогресу виконання завдань спринту в реальному часі, дозволяючи команді оперативно реагувати на затримки або проблеми. Реалізація Planning Stand, Planning Stand є частиною процесу планування в Scrum, який включає обговорення і вибір завдань на наступний спринт. Для його реалізації у системі можна використати наступний підхід:

- інтерфейс для планувальних зустрічей: розробити спеціалізований інтерфейс для проведення планувальних зустрічей, який дозволяє команді взаємодіяти з product backlog та формувати sprint backlog. цей інтерфейс може включати функції голосового та відео зв'язку для дистанційної роботи;

- естимейти та пріоритизація: інтегрувати інструменти для оцінки завдань (наприклад, planning poker) та механізми для пріоритизація завдань на основі їх важливості та терміновості, Реалізація Planning Stand, Planning Stand є частиною процесу планування в Scrum, який включає обговорення і вибір завдань на наступний спринт;

- запис рішень та змін: забезпечити можливість запису обговорень, рішень та будь-яких змін у завданнях підчас планувальної зустрічі. це може включати автоматичне створення протоколів зустрічей, які можуть бути доступні для перегляду всім членам команди для забезпечення прозорості та відповідальності;

- інтеграція з календарем: автоматично створювати події в календарі

команди для планованих планувальних зустрічей, нагадувань про наближення термінів завершення спринту та інших важливих дат, пов'язаних з проектом. це допоможе забезпечити, що всі члени команди своєчасно готові до зустрічей та важливих подій;

– відкритий доступ до документації: забезпечити легкий доступ до всієї документації, створеної під час планувальної зустрічі, включаючи естимейти, пріоритети, записи змін у завданнях, та інше. це дозволить команді мати одне джерело правди щодо планів і завдань спринту, Реалізація Planning Stand, Planning Stand є частиною процесу планування в Scrum, який включає обговорення і вибір завдань на наступний спринт.

Далі наведемо діаграму дій користувача у подібній системі для більшого розуміння внутрішніх процесів, діаграма наведена на рисунку 3.3.

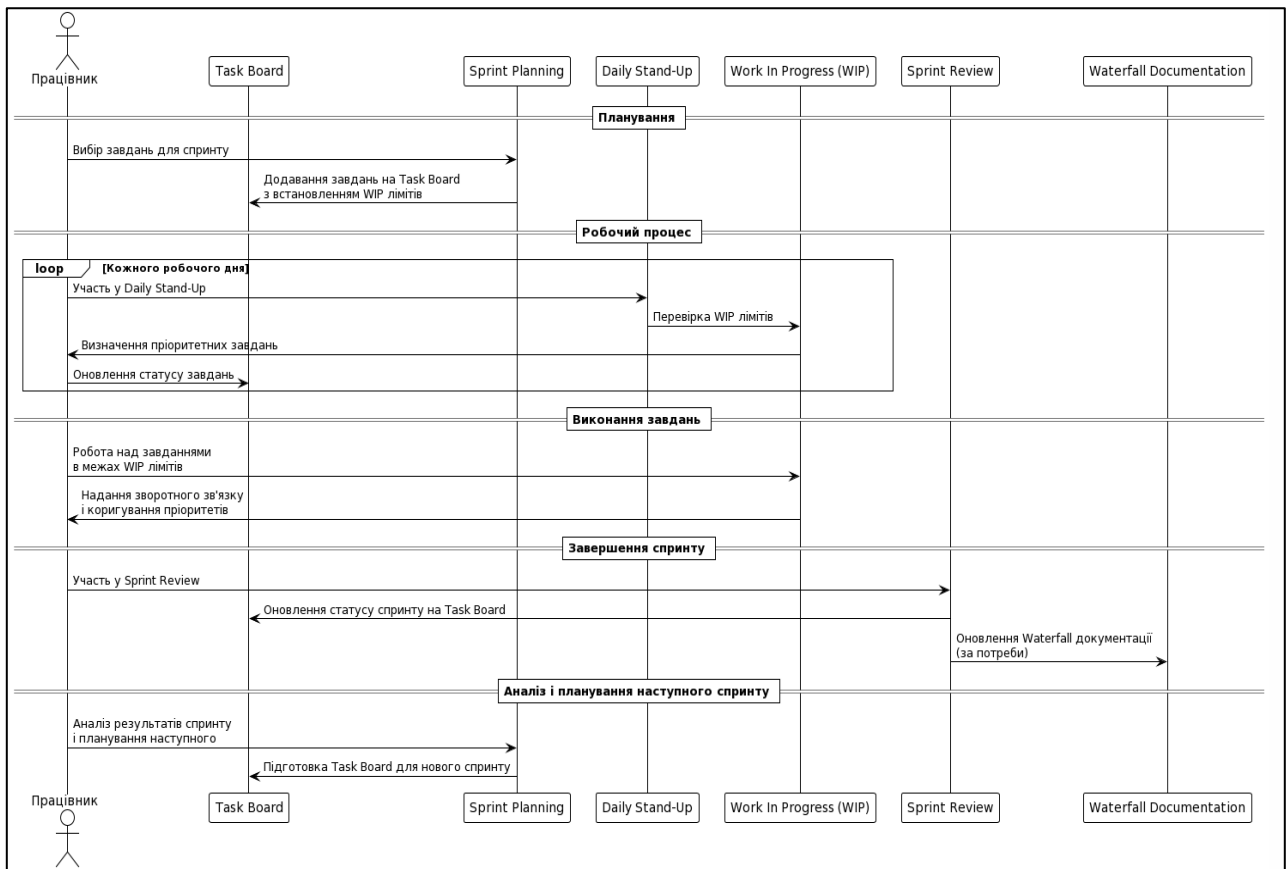


Рисунок 3.3 – Діаграма дій користувача

Сценарій починається з планування спринту, продовжується щоденною роботою над завданнями в межах встановлених WIP лімітів, включає участь у щоденних стендапах і завершується участю в sprint review та оновленням документації за методологією Waterfall. Планування спринтів проводиться з урахуванням принципів Agile, де команда спільно визначає обсяг роботи на наступний спринт, обираючи завдання з загального беклогу проекту.

4 РЕАЛІЗАЦІЯ ГІБРИДНОГО МЕТОДУ СИСТЕМИ УПРАВЛІНЯ ПРОЕКТАМИ

Далі нам необхідно розглянути реалізацію обраних та детально описаних елементів методів контролю проектів, але спочатку варто навести архітектурний план та зноску подібної системи.

Наша гібридна система управління проектами інтегрує елементи Agile, Kanban та Waterfall, створюючи гнучке, але структуроване середовище для управління різноманітними проектами. Система організована навколо центрального Task Board, який відображає всі поточні завдання проекту, розподілені за стадіями виконання. Кожне завдання може містити детальну інформацію, включаючи опис, пріоритет, відповідальних виконавців та естимовані терміни виконання.

Планування спринтів проводиться з урахуванням принципів Agile, де команда спільно визначає обсяг роботи на наступний спринт, обираючи завдання з загального беклогу проекту. Під час цього процесу використовуються WIP ліміти для кожної колонки на Task Board, що дозволяє контролювати навантаження на команду та забезпечувати зосередження на важливих задачах без перевантаження.

Щоденні стендапи (Daily Stand-Ups) допомагають команді синхронізувати свої дії та швидко вирішувати виникаючі проблеми, в той час як регулярні ретроспективи і огляди спринтів сприяють дослідженню продуктивності команди та вдосконаленню робочих процесів.

Для задач, що вимагають більшої структурованості та документації, застосовуються елементи Waterfall, особливо на ранніх стадіях планування

проекту та на етапі здачі готового продукту. Це забезпечує чітке визначення обсягу робіт, вимог до проекту та критеріїв приймання.

Адаптивне планування та пріоритизація завдань в системі дозволяють команді гнучко реагувати на зміни в проекті, забезпечуючи високу адаптивність без втрати контролю над кінцевим результатом. Інтеграція з іншими інструментами, такими як електронна пошта, календарі та системи документообігу, спрощує комунікацію всередині команди та зі стейкхолдерами, а також полегшує управління документацією та термінами.

Така архітектура системи поєднує в собі гнучкість Agile зі структурованістю Waterfall та візуалізацією робочих процесів Kanban, створюючи універсальне рішення для управління проектами. Вона забезпечує командам можливість ефективно адаптуватися до мінливих умов роботи, при цьому зберігаючи чіткий контроль над прогресом проекту та якістю кінцевого продукту. Централізоване управління завданнями через Task Board дозволяє відслідковувати стан усіх завдань у реальному часі, сприяє оптимізації розподілу ресурсів і допомагає уникнути зайвого навантаження на команду.

Гнучкість системи полягає у можливості швидко переходити між різними методологіями в залежності від потреб етапу проекту. Наприклад, на початковому етапі може бути корисною детальна планова структура Waterfall для визначення обсягу проекту та основних вимог, тоді як у фазі розробки ефективніше застосовувати Agile для гнучкого управління змінами та Kanban для візуалізації поточних задач та пріоритизації.

Застосування WIP лімітів та регулярне дослідження продуктивності команди допомагають забезпечити баланс між швидкістю виконання робіт та якістю, запобігаючи вигоранню співробітників та підвищуючи загальну задоволеність роботою в проекті. Ретроспективи та оцінка ефективності на кінці

кожного спринту сприяють постійному вдосконаленню процесів та методів роботи команди.

Інтеграція зовнішніх інструментів та сервісів надає додаткові можливості для автоматизації рутинних процесів, спрощення комунікації з клієнтами та стейкхолдерами, а також підвищення загальної ефективності управління проектом.

Далі буде розглянуто сценарії та приклади системного UI.

Коли користувач має всі завдання, вам потрібно їх систематизувати. Це підводить нас до діаграми Ганта, яка є фундаментальним інструментом водоспаду, але також корисна у багатьох методологіях проектів. У ньому перераховані ваші завдання та показаний весь проект на часовій шкалі. Ви можете призначати членів команди та описувати завдання, встановлювати пріоритети та додавати теги, щоб їх було легко знайти (див. рис. 4.1.1).

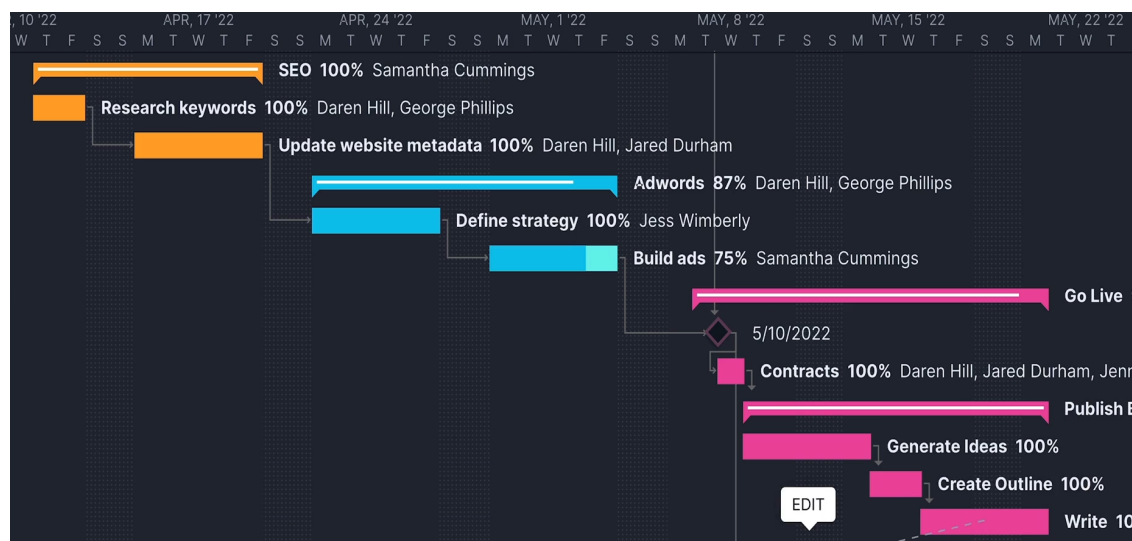


Рисунок 4 – UI ітераційного планування

Далі розглянемо дошки команди можуть організувати свій беклог на дошці Kanban, що візуалізує робочий процес, розміщуючи завдання у стовпцях, що

позначають етапи виробничого циклу. Команди можуть використовувати подання дошки для планування своїх спринтів, а менеджери отримують уявлення про процес своєї команди. Якщо вони виявлять вузьке місце, менеджери можуть перерозподілити ресурси, ніж блокувати роботу своїх команд(див. рис. 5).

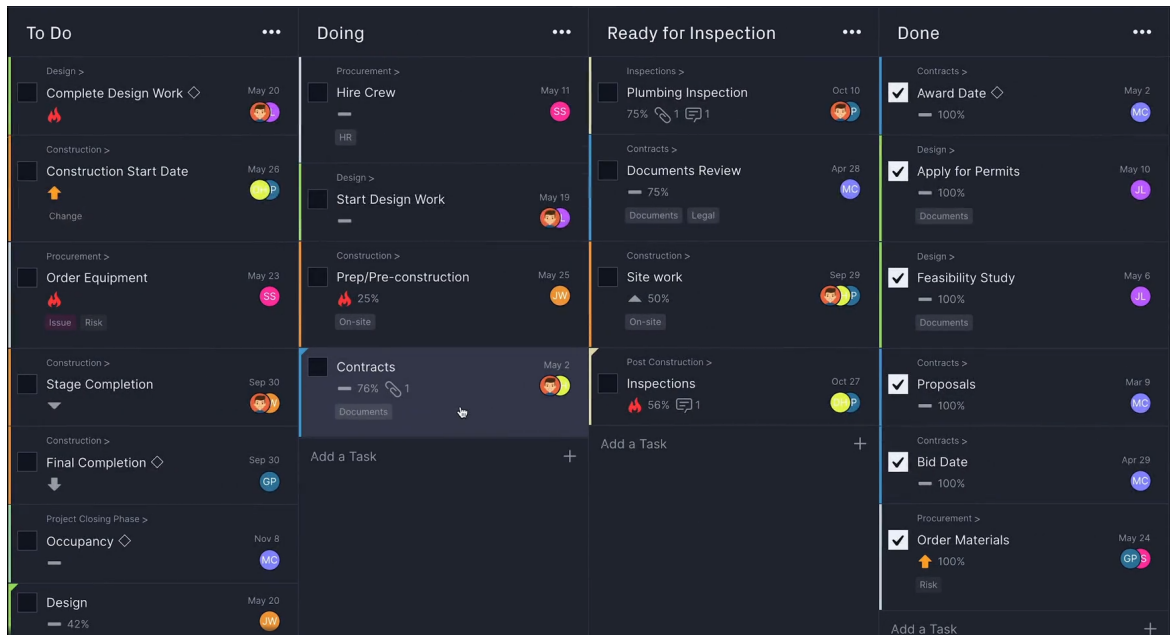


Рисунок 5 – UI дошок завдання

Далі розглянемо ліміти команди та ліміти пріоритету, надання командам свободи виконання завдань – це частина роботи у гнучкому середовищі. Менеджери хочуть допомагати, не заважаючи їм, тому є інструменти управління ресурсами, такі як діаграма робочого навантаження. Відразу дізнавайтесь, хто перерозподілений, а потім перерозподіляйте завдання прямо з діаграми робочого навантаження, щоб збалансувати робоче навантаження команди та забезпечити її продуктивність(див. рис. 6).

Ліміти WIP чітко вказані на кожній колонці дошки. Це може бути представлено числовим значенням, яке показує максимальну кількість задач, дозволених на даному етапі процесу.

Користувачі бачать ці ліміти як числові індикатори поруч із заголовками колонок, наприклад, "Doing (3/5)", де 3 — поточна кількість задач, а 5 — встановлений ліміт.

Якщо ліміт перевищено, система може блокувати можливість додавання нових задач до цієї колонки, доки кількість задач не зменшиться.

Ліміти пріоритету є індивідуальними та встановлюються працівником (див. рис. 7), також на цьому рисунку зображено коли кількість задач у колонці наближається до встановленого ліміту, система може надавати візуальні сигнали, наприклад, змінювати кольори або відображати попередження, щоб привернути увагу команди.

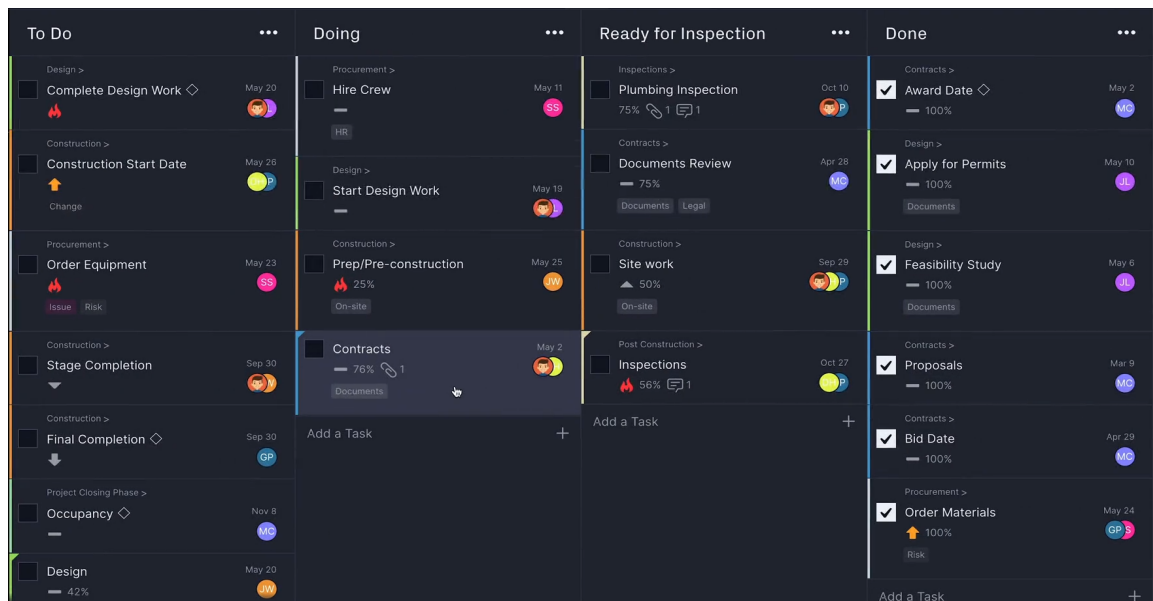


Рисунок 6 – UI wip limit команди

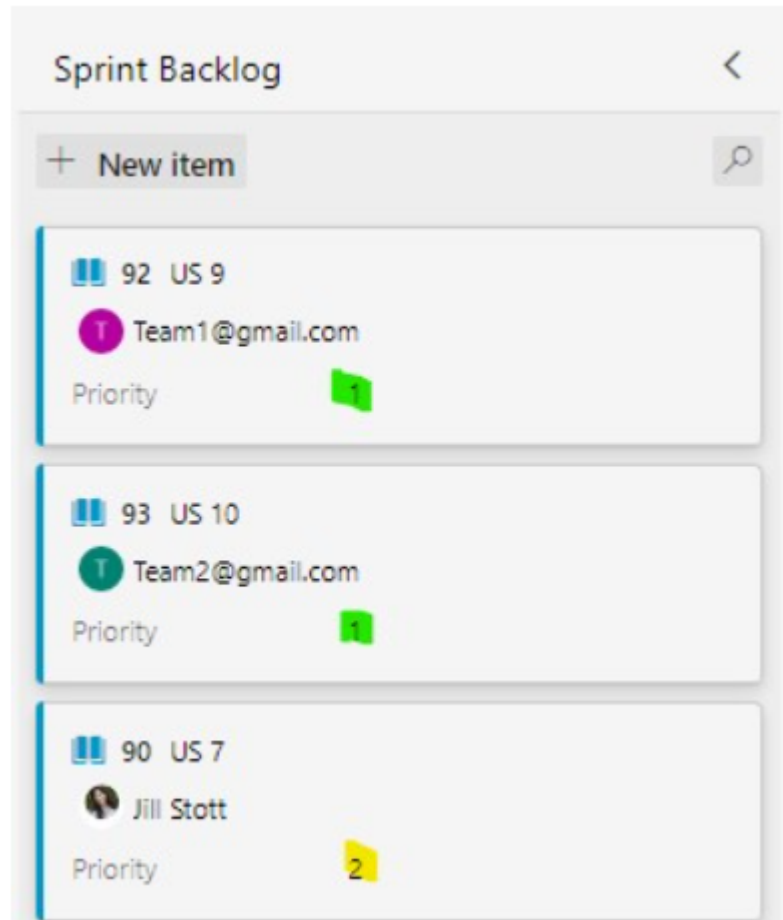


Рисунок 7 – UI wip limit пріоритету

Також варто зазначити planing stand, хоча планування етапів розробки і є фактично у результаті графіком але також має свій UI, на якому відображені завдання їх групи та графіки виконання (див. рис. 8). На інтерфейсі можна бачити розділення завдань кольорами та нумерування завдань в послідовності.

ALL	TASK NAME	WBS
1	<input type="checkbox"/> Ideation	1
2	Market Research	1.1
3	Product Requireme...	1.2
4	Feasibility Analysis	1.3
5	Stakeholder Feedba...	1.4
6	<input type="checkbox"/> Design	2
7	Engineering Drawings	2.1
8	3D Rendering	2.2
9	3D Printed Prototype	2.3
10	Stakeholder Feedba...	2.4
11	<input type="checkbox"/> Prototyping	3
12	CNC Part Creation	3.1
13	Product Assembly	3.2
14	Durability & Stress T...	3.3
15	<input type="checkbox"/> Pre-production	4
16	New Production Eq...	4.1
17	Supply Chain Sourci...	4.2
18	Product Documenta...	4.3
19	Translation Services	4.4
20	Production Operato...	4.5
21	<input type="checkbox"/> Compliance	5
22	Safety & Regulatory...	5.1
23	<input type="checkbox"/> Product Marketing	6
24	Pricing & Packaging	6.1
25		

Рисунок 8 – UI planing stand

Також варто ще раз зазначити новизну яку ми впровадили в метод управління дошками, а саме – можливість якорно прикріпляти посилання на певний чат із штучним інтелектом для зручного та швидкого отримання інформації пов’язаної із певною задачею(див. рис. 9). На рисунку можна бачити ресурс із якірною приміткою для розгортаючогося списку посилань на конкретній чат із подальшою можливістю переходу.

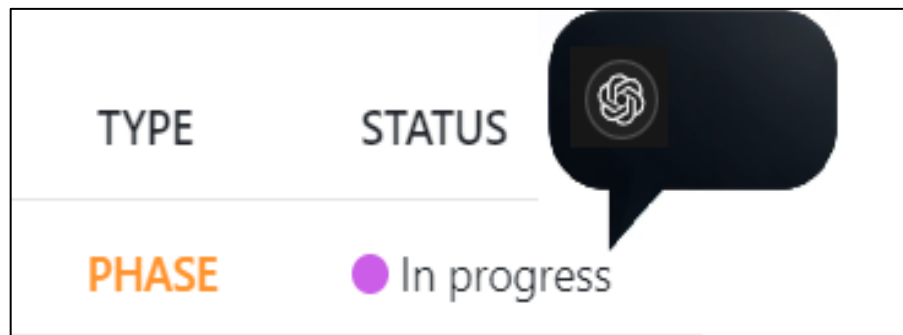


Рисунок 9 – UI реалізації взаємодії із чатами

ВИСНОВКИ

У ході роботи було встановлено задачу глибоко дослідити предметну область управління проектами та дослідити існуючі підходи до цього процесу. Метою було виявлення ключових методологій, які дозволяють ефективно керувати проектами, враховуючи специфіку роботи, потреби команд та очікування стейкхолдерів. Особливу увагу було приділено методологіями Agile, Kanban та Waterfall, кожна з яких має свої переваги та особливості застосування в різних умовах.

Після детального дослідження існуючих підходів ми обрали гібридний метод, що інтегрує сильні сторони Agile, Kanban та Waterfall, створюючи гнучку, але структуровану систему, здатну адаптуватися до різноманітних вимог та змінних умов проектного середовища. Таке поєднання дозволяє використовувати Agile для забезпечення адаптивності та швидкої реакції на зміни, таблиці Kanban, відмітимо додатково що не брався увесь підхід а тільки візуальні таблиці, для візуалізації робочих процесів та оптимізації потоку завдань, а Waterfall — для чіткого планування на початкових та завершальних етапах проекту.

На основі цього вибору було сплановано розробку програмної системи, яка б втілювала обрані підходи. Система має включати інтерактивні Task Boards з можливістю встановлення WIP лімітів, інструменти для організації планувальних засідань, щоденних стендапів, ретроспектив, а також механізми для адаптивного планування та ведення проектної документації.

Також нами було імплементовано новий функціонал зв'язку користувача із персональним чатом ChatGPT, подібний функціонал не зустрічається у подібних

системах та є дуже зручним, додаючи інтерактивності, гнучкості та зручності у доступі до інформації.

Така система має забезпечити команді високий рівень організованості, прозорості та контролю над проектним процесом, а також можливість ефективно реагувати на будь-які зміни.

ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАНЬ

1. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи другого (магістерського) рівня вищої освіти за освітньо-науковою програмою «Управління проектами в галузі інформаційних технологій» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2024. – 24 с.
2. Загальна характеристик ІТ проектів // What are the Characteristics of a Project. 2024. URL: https://www.researchgate.net/publication/322879065_General_Characteristics_and_Common_Practices_for_ICT_Projects_Evaluation_Perspective (дата звернення: 25.04.2024).
3. Ініціація за принципом «Request for Proposal» // Mastering the RFP 2024. URL: <https://attractgroup.com/blog/write-a-request-for-proposal/> (дата звернення: 25.04.2024).
4. Проблеми управління ІТ проектами // Overcoming the Top Challenges of IT Project Management 2023. URL: <https://www.wrike.com/blog/top-challenges-it-project-management/> (дата звернення: 25.04.2024).
5. Огляд підходу Scrum // A Visual Guide to Scrum Methodology. 2021. URL: <https://intive.com/insights/a-visual-guide-to-scrum-methodology> (дата звернення: 25.04.2024).
6. Огляд підходу Agile // What is Agile methodology 2023. URL: <https://asana.com/ru/resources/agile-methodology> (дата звернення: 25.04.2024).

7. Огляд підходу Waterfall // waterfall model 2020. URL: <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model> (дата звернення: 25.04.2024).
8. Огляд підходу Stage-Gate // Stage Gate Process: All You Need to Know 2023. URL: <https://www.intellectsoft.net/blog/stage-gate-process/> (дата звернення: 25.04.2024).
9. Огляд підходу Kanban, URL: <https://www.atlassian.com/agile/kanban#:~:text=A%20kanban%20team%20focus%20only,don%27t%20impact%20the%20team.> (дата звернення: 25.04.2024).
10. Порівняння методологій управління // Hybrid project management 2023. URL: https://www.researchgate.net/publication/361812857_Hybrid_project_management_-_a_systematic_literature_review (дата звернення: 25.04.2024).
11. Аналіз гібридних метадологій // Hybrid Project Management Models: A Systematic Literature Review 2023. URL: https://www.researchgate.net/publication/370801369_Hybrid_Project_Management_Models_A_Systematic_Literature_Review (дата звернення: 25.04.2024).
12. Water-Scrum-Fall// Hybrid Project Management Models: A Systematic Literature Review 2023. URL: https://www.researchgate.net/publication/370801369_Hybrid_Project_Management_Models_A_Systematic_Literature_Review (дата звернення: 25.04.2024).
13. Аналіз ISACA // Get data set 2023. URL: <https://www.isaca.org/credentialing/data-science-fundamentals-certificate> (дата звернення: 25.04.2024).
14. Аналіз OWASP // Get data set 2023. URL: <https://www.isaca.org/credentialing/data-science-fundamentals-certificate> (дата звернення: 25.04.2024).

15. Аналіз CSV // Get data set 2023.

URL: <https://www.isaca.org/credentialing/data-science-fundamentals-certificate>

(дата звернення: 25.04.2024).

16. Робота із Pandas-dev // Open source project 2023.

URL: <https://github.com/pandas-dev/pandas> (дата звернення: 25.04.2024).