

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

**КВАЛІФІКАЦІЙНА РОБОТА**

Тема: «Гнучке управління створенням застосунку для управління проектами  
для малих команд»

Ступінь вищої освіти – магістр

Спеціальність – 073 «Менеджмент»

Освітня програма «Agile-технології розробки програмного забезпечення»

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

Керівник: д.е.н., доц.,  
професор кафедри ІММС  
Ольга ОРЛОВА-КУРИЛОВА

Керівник: к.е.н., проф.,  
професор кафедри УТ  
Наталія ЛІТВІН

Виконав: здобувач  
групи МЕН/Agile-23м-о  
Антон КОВАЛЬЧУК

Київ, 2025 р.

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

ЗАТВЕРДЖУЮ:

завідувач кафедри інформаційного  
менеджменту, математики та статистики

\_\_\_\_\_ Денис БАЛДИК

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ  
КОВАЛЬЧУК АНТОН ОЛЕКСАНДРОВИЧ**

Тема роботи	Гнучке управління створенням застосунку для управління проектами для малих команд
Номер та дата наказу про затвердження теми	№ 106-1 від 11 листопада 2024 року
Коротка постановка завдання	Створення системи для організації робочого процесу та впровадження методології Agile в малих або недосвідчених командах
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	Agile Software Development Framework in a Small Project Environment– URL: <a href="https://www.researchgate.net/publication/264167913_Agile_Software_Development_Framework_in_a_Small_Project_Environment">https://www.researchgate.net/publication/264167913_Agile_Software_Development_Framework_in_a_Small_Project_Environment</a> A Software Development Process for Small Projects– URL: <a href="https://www.researchgate.net/publication/272813672_A_Software_Development_Process_for_Small_Projects">https://www.researchgate.net/publication/272813672_A_Software_Development_Process_for_Small_Projects</a>
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має містити теоретичне та/або практичне дослідження за темою роботи, яку слід розглядати як складне спеціалізоване завдання або практичну проблематику в галузі управління та адміністрування, яка характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів Agile технологій.

Дата видачі завдання «11» листопада 2024 р.

Керівник

Ольга ОРЛОВА-КУРИЛОВА

Керівник

Наталія ЛІТВІН

Здобувач

Антон КОВАЛЬЧУК

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
<b>Підготовчий етап</b>			
1	Вибір напрямку дослідження та керівника	01.11.2024 р.	<i>виконано</i>
2	Формування теми та призначення керівника	08.11.2024 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи	11.11.2024 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу	11.11.2024 р.	<i>виконано</i>
<b>Основний етап</b>			
5	Розробка концепції кваліфікаційної роботи	12.11.2024 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів.	12.11.2024 р.	<i>виконано</i>
7	Затвердження розширеної постановки завдання. Підготовка та подання керівнику розділу 1 кваліфікаційної роботи	15.11.2024 р.	<i>виконано</i>
8	Проектування інформаційної системи. Підготовка та подання керівнику розділу 2 кваліфікаційної роботи	22.11.2024 р.	<i>виконано</i>
9	Реалізація інформаційної системи. Підготовка та подання керівнику розділу 3 кваліфікаційної роботи	29.11.2024 р.	<i>виконано</i>
10	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	02.12.2024 р.	<i>виконано</i>
11	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівнику доопрацьованого варіанту кваліфікаційної роботи	09.12.2024 р.	<i>виконано</i>
<b>Завершальний етап</b>			
12	Представлення рукопису для перевірки на плагіат	09.12.2024 р.	<i>виконано</i>
13	Підготовка презентації та доповіді на передзахист	13.12.2024 р.	<i>виконано</i>
14	Передзахист кваліфікаційної роботи	23.12.2024 р.	<i>виконано</i>
15	Технічна самоекспертиза роботи на відповідність вимогам до оформлення та виправлення недоліків	29.12.2024 р.	<i>виконано</i>
16	Експертиза роботи керівником та зовнішнім експертом	06.01.2025 р.	<i>виконано</i>
17	Доопрацювання доповіді та презентації для захисту	13.01.2025 р.	<i>виконано</i>
18	Захист кваліфікаційної роботи	20-24.01.2025 р.	<i>виконано</i>

Керівник

Ольга ОРЛОВА-КУРИЛОВА

Керівник

Наталія ЛІТВІН

Здобувач

Антон КОВАЛЬЧУК

## АНОТАЦІЯ

*КОВАЛЬЧУК А. О. «ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ ЗАСТОСУНКУ ДЛЯ УПРАВЛІННЯ ПРОЄКТАМИ ДЛЯ МАЛИХ КОМАНД»*

В кваліфікаційній роботі розглянуто теоретико-методичні основи гнучкого управління, зокрема особливості Agile підходів в управлінні створенням продукту. Представлено дизайн застосунку та опис запропонованого застосунку для впровадження та організації використання методології Agile.

Розглянуті окремі аспекти гнучкого управління створенням застосунку, зокрема: цілі проєкту, план виконання та терміни проєкту, бюджет проєкту; команда проєкту, стейкхолдери проєкту, правила роботи та план комунікацій в проєкті.

Ключові слова: Agile-маніфест, Agile підходи, проєкт, продукт, команда.

## ANNOTATION

*KOVALCHUK A.O. "FLEXIBLE MANAGEMENT OF DEVELOPMENT OF A MOBILE APPLICATION FOR MANAGING AND INTEGRATION OF AGILE METHODOLOGY IN SMALL TEAMS"*

The qualification work uses the theoretical and methodological foundations of flexible management, in particular the features of Agile approaches in managing product creation. The application's proposed design incentivizes implementation and usage of the Agile methodology in small scale software development teams.

Some aspects of flexible management of application creation are considered, in particular: project goals, project implementation plan and deadlines, project budget; project team, project stakeholders, work rules and project communications plan.

Keywords: Agile manifesto, Agile approaches, project, product, team.

## ЗМІСТ

<b>ВСТУП</b> .....	<b>6</b>
<b>РОЗДІЛ 1 ХАРАКТЕРИСТИКА ПРОБЛЕМИ, ПОСТАНОВКА ЗАВДАНЬ ТА ОГЛЯД АНАЛОГІВ СТВОРЮВАНОВОГО ЗАСТОСУНКУ</b> .....	<b>9</b>
1.1. ПРОБЛЕМАТИКА ПОСТАВЛЕНОГО ЗАВДАННЯ .....	9
1.2. АНАЛІЗ КОНКУРЕНТІВ ТА АНАЛОГІВ .....	11
1.3. ПОСТАНОВКА ЦІЛЕЙ, ЗАВДАНЬ ТА ВИМОГ ДО ПРОЄКТУ.....	13
Висновки до розділу 1 .....	17
<b>РОЗДІЛ 2 ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ ЗАСТОСУНКУ</b> .....	<b>18</b>
2.1. МЕТОДИ ГНУЧКОГО УПРАВЛІННЯ ПРОЄКТАМИ.....	18
2.2. МОНИТОРИНГ ВИКОНАННЯ ПРОЄКТУ .....	19
2.3. ПЛАНУВАННЯ ПРОЦЕСІВ ПРОЄКТУ .....	20
Висновки до розділу 2 .....	22
<b>РОЗДІЛ 3 ПЛАНУВАННЯ РОБІТ НАД ЗАСТОСУНКОМ</b> .....	<b>23</b>
3.1. РОЗРОБКА ПОСЛІДОВНИХ ЕТАПІВ ТА ПЕРЕЛІКУ РОБІТ .....	23
3.2. СИТУАЦІЙНА ОЦІНКА СТЕЙКГОЛДЕРІВ ПРОЄКТУ .....	26
3.3 ЕТАПИ ТА ЗМІСТ РОБІТ ПРОЄКТУ .....	26
Висновки до розділу 3 .....	37
<b>ВИСНОВКИ</b> .....	<b>39</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	<b>41</b>

## ВСТУП

У сучасному IT-середовищі ефективне управління проєктами є критично важливим фактором успіху як для великих компаній, так і для невеликих команд, зокрема стартапів чи малих проєктних груп. Такі команди часто стикаються з необхідністю швидко адаптуватися до змін у вимогах, забезпечуючи при цьому прозорість процесів і злагоджену співпрацю між учасниками. Однак, незважаючи на наявність численних інструментів для управління проєктами, їх використання супроводжується певними проблемами: складністю освоєння функціоналу, високим порогом входу для новачків, відсутністю адаптивних рішень для малих колективів, а також потребою в значних ресурсах для впровадження гнучких методологій, таких як Agile.

Сьогодні ринок пропонує потужні інструменти для управління проєктами, такі як Jira чи Notion, які широко використовуються великими командами. Проте ці рішення часто не враховують специфічних потреб невеликих колективів, які лише починають знайомство з Agile-методологією. Занадто складний функціонал і перевантажений інтерфейс можуть відштовхувати початківців, сповільнюючи робочі процеси та заважаючи команді досягати поставлених цілей. Це створює значну прогалину на ринку, яку має заповнити новий інструмент, орієнтований на простоту використання та адаптивність.

Розробка інструменту, який враховує специфіку малих команд і допомагає їм освоїти Agile-методологію, є надзвичайно актуальною. У сучасних умовах постійних змін і конкуренції саме здатність до адаптації стає ключовим фактором виживання на ринку. Для стартапів, які працюють в умовах обмежених ресурсів і часу, важливо мати доступ до інструментів, що дозволяють мінімізувати складність організації робочих процесів, зберігаючи при цьому гнучкість і прозорість. Запропонований проєкт спрямований на задоволення цих потреб,

створюючи рішення, яке забезпечує поєднання простоти використання та навчального потенціалу для інтеграції Agile.

Метою цієї роботи є розробка концепції та створення легкого у використанні додатку для управління проєктами, який буде орієнтований на потреби малих команд та сприятиме впровадженню Agile-методології з нуля.

Для досягнення мети були сформульовані такі завдання:

1. Провести аналіз потреб цільової аудиторії, зокрема малих команд, які вперше впроваджують Agile.
2. Вивчити конкурентне середовище, проаналізувавши сильні та слабкі сторони існуючих рішень.
3. Розробити прототип продукту з інтуїтивним інтерфейсом та адаптивними функціями для командного управління.
4. Впровадити ключові принципи Agile у функціонал додатку, забезпечивши його доступність для новачків.
5. Тестувати продукт та отримати зворотний зв'язок від цільової аудиторії.

Об'єктом дослідження є процеси управління проєктами у малих командах, зокрема організація робочих процесів із використанням гнучких методологій.

Предметом дослідження є принципи побудови програмного забезпечення для управління проєктами, що забезпечує легке впровадження Agile у командах, які раніше не працювали з цією методологією.

Методи дослідження включають системний аналіз, порівняльний аналіз, методи проєктного менеджменту, методи календарного та ресурсного планування.

Практична значущість роботи полягає у створенні інструменту, який не лише допоможе малим командам організувати свою роботу, а й стане платформою для освоєння принципів Agile. Продукт може бути використаний стартапами, освітніми установами та іншими організаціями, які прагнуть

підвищити ефективність управління проектами та адаптуватися до сучасних вимог ринку.

Для виконання роботи застосовувалися методи аналізу літературних джерел, порівняння існуючих рішень, розробка програмного забезпечення у рамках Agile-процесів, користувацьке тестування та опитування.

Результати дослідження можуть бути використані в малих командах, які прагнуть підвищити продуктивність і впровадити Agile-процеси, а також у навчальних програмах для ознайомлення студентів з інструментами управління проектами.

Структура роботи. Робота складається зі вступу, трьох розділів і висновків, викладених на 42 сторінках тексту. Матеріали кваліфікаційної роботи містять 9 таблиць. Список використаних джерел складається із 14 найменувань, які розміщено на 2 сторінках.

# РОЗДІЛ 1

## ХАРАКТЕРИСТИКА ПРОБЛЕМИ, ПОСТАНОВКА ЗАВДАНЬ ТА ОГЛЯД АНАЛОГІВ СТВОРЮВАНОВОГО ЗАСТОСУНКУ

### 1.1. Проблематика поставленого завдання

Малі ІТ-команди та стартапи часто опиняються у ситуації, коли їм потрібно організувати свою роботу в умовах обмежених ресурсів і часу. Ефективне управління проєктами є критично важливим, але впровадження структурованих методів, таких як Agile, стає викликом через ряд специфічних проблем.

У невеликих командах часто відсутній чіткий розподіл ролей і зон відповідальності. Завдання не мають структури, а прогрес проєкту не відстежується належним чином. Це призводить до плутанини, втрати важливих деталей і зниження продуктивності.

Більшість рішень на ринку створені для великих команд і компаній, які мають усталені процеси. Для невеликих груп такі інструменти часто стають перевантаженими: Вони включають функції, які не використовуються малими командами, їхній інтерфейс вимагає тривалого освоєння, що є проблемою для команд, які потребують швидкого старту.

Хоча Agile вважається гнучким підходом, його базові принципи – спринти, беклог, постійна комунікація – потребують розуміння та практичного застосування. Команди, які тільки починають свій шлях, не завжди мають досвід або ресурси для навчання. Це створює бар'єри для впровадження Agile навіть у тих випадках, коли команда готова його використовувати.

Малі команди часто використовують кілька окремих інструментів: одні для задач, інші для комунікації, треті для документації. Такий підхід може працювати на початкових етапах, але з часом веде до фрагментованості процесів, втрати інформації та затримок у роботі. Малі команди не завжди мають змогу інвестувати в дорогі платформи чи навчання. Крім того, розробка

індивідуального рішення для управління проєктами потребує значних ресурсів, що часто є недоступним.

Фінальний продукт представлятиме інструмент, який не лише полегшує управління проєктами, але й усуває бар'єри, що заважають малим командам упроваджувати Agile.

Ключовими аспектами функціоналу стануть:

1. Проста структура для складних задач.

Усі основні компоненти (беклог, дошка завдань, спринти) організовані таким чином, щоб користувач міг швидко орієнтуватися навіть без попереднього досвіду. Інтерфейс побудований із фокусом на малі команди: мінімум зайвих функцій, акцент на інтуїтивності.

2. Вбудована підтримка Agile-процесів.

Додаток пропонує інтерактивні кроки для впровадження Agile, допомагаючи команді організувати перші спринти, розподілити завдання та розставити пріоритети. Інструменти для ретроспектив і моніторингу прогресу сприяють аналізу роботи та постійному вдосконаленню.

3. Гнучкість для різних потреб.

Налаштовувані шаблони дозволяють адаптувати систему під специфіку команди або проєкту. Кожен користувач може налаштовувати відображення інтерфейсу залежно від своєї ролі: розробник, менеджер, тестувальник тощо.

4. Єдиний центр співпраці.

Замість використання кількох платформ, продукт об'єднує задачі, комунікацію та документацію в одному середовищі. Інтеграція з популярними інструментами (наприклад, Google Calendar, Slack) забезпечує зручність у використанні вже існуючих процесів.

5. Доступність і навчання.

Додаток містить інтерактивні підказки та пояснення, які дозволяють навіть новачкам швидко зрозуміти основи Agile. Для команд із мінімальним бюджетом пропонується безкоштовний план із усіма необхідними функціями.

Продукт розробляється з урахуванням особливостей і можливостей малих IT-команд, які часто працюють у нестабільних умовах із жорсткими дедлайнами та обмеженими ресурсами. Основна цінність додатку полягає в його здатності не лише впорядковувати робочі процеси, але й допомагати командам поступово освоювати Agile, зменшуючи рівень стресу та підвищуючи ефективність роботи.

Простий, інтуїтивний і адаптований до потреб малих груп, цей додаток стане універсальним інструментом для оптимізації роботи, створення структури та досягнення результатів навіть у найскладніших умовах.

## **1.2. Аналіз конкурентів та аналогів**

На ринку управління проєктами домінують кілька популярних платформ, ці інструменти мають значні переваги та забезпечують широкий функціонал, проте не завжди відповідають потребам малих команд або новачків у роботі за Agile-методологією.

Однією з головних проблем є складність використання цих платформ новачками. Продукти, як-от Jira, пропонують розширений функціонал для глибокого управління процесами, проте їхній інтерфейс і багатошаровість нерідко викликають труднощі у користувачів, які не мають досвіду роботи з подібними системами. Це може стати бар'єром для впровадження Agile у командах, які прагнуть почати з малого, але стикаються з перевантаженням інструментами, що потребують значного часу для налаштування та навчання.

Інші інструменти, як Notion, відрізняються простотою й гнучкістю, проте не мають вбудованої підтримки процесів Agile. Хоча вони можуть слугувати стартовими рішеннями, користувачі змушені витратити час на адаптацію

функціоналу під свої потреби, що уповільнює роботу команди та відволікає від основних цілей проекту.

Ще однією суттєвою перешкодою є фрагментованість рішень, коли командам доводиться використовувати декілька різних платформ для управління задачами, комунікації та документування. Це створює хаос у робочих процесах, знижує продуктивність і ускладнює синхронізацію між членами команди.

Варто також врахувати фінансовий аспект. Більшість провідних платформ мають безкоштовні версії з суттєвими обмеженнями, а повний доступ до функціоналу потребує значних інвестицій. Для невеликих команд і стартапів це може стати вирішальним фактором при виборі інструмента.

*Таблиця 2.3 - Аналіз конкурентів*

Критерії оцінки	Оцінка конкурентів	
	Jira	Notion
Досвід роботи	9	7
Імідж	9	7
Тенденції розвитку	5	9
Потенціал розвитку	6	10
Доля на ринку	10	5
Всього балів:	39	38

*Джерело: розроблено автором [1]*

Запропонований продукт вирішує ці проблеми, створюючи баланс між простотою, доступністю та функціональністю. Його концепція побудована навколо потреб малих команд, які прагнуть швидко адаптувати Agile у своїй роботі без складних налаштувань чи великих витрат.

На відміну від існуючих конкурентів, додаток поєднує кілька ключових переваг:

- простий інтерфейс, розроблений для користувачів без досвіду роботи з Agile;
- інтеграція основних процесів Agile, таких як спринти, беклоги та ретроспективи, із мінімумом налаштувань;
- єдине середовище для управління задачами, обговорення та документування, що усуває потребу в багатьох інструментах;
- доступність для малих команд завдяки безкоштовному базовому пакету, що не поступається якістю платним рішенням.

Цей підхід не тільки спрощує впровадження Agile, але й усуває бар'єри, які зазвичай виникають у новачків. У той час як великі платформи зосереджені на масштабованості та широкій функціональності, цей проєкт орієнтований на легкість використання та адаптацію до конкретних потреб невеликих колективів.

Таким чином, він стає не просто ще одним інструментом на ринку, а рішенням, яке враховує реальні виклики, з якими стикаються команди в умовах обмежених ресурсів і часу.

### **1.3. Постановка цілей, завдань та вимог до проєкту**

Даний проєкт спрямований на створення додатку, який буде надавати організаційну допомогу командам чи організаціям малого розміру для покращення процесу організації та управління робочим процесом. Особливістю цієї системи буде її орієнтація на проєкти або команди невеликих масштабів з фокусом на максимальне спрощення взаємодії користувача і системи, максимальну видимість і прозорість організованих процесів та інструменти для планування і моніторингу.

### Цілі проєкту:

- проведення аналізу потреб сегменту ринку та загальної цільової аудиторії з метою формулювання чітких вимог до функціоналу системи;
- проведення аналізу сектора ринку щодо існуючих альтернатив, їх сильних та слабких сторін та особливостей їх функціоналу або методу впровадження з метою побудови плану дій щодо аспектів або особливостей, на яких варто капіталізувати при виконанні проєкту;
- розробка MVP: створення прототипу з основними функціями, які можна оперативнo тестувати на користувачах;
- підвищення продуктивності команди: отримання результатів від впровадження системи що покаже ефективність її роботи у вигляді зменшення часу, витраченого на управління проєктами, і покращення якості та швидкості виконання завдань;
- підвищення якості комунікацій всередині команди: отримання результатів від впровадження системи що покаже ефективність її роботи у вигляді збільшення рівню взаємодії між членами команди;
- розробка стратегії впровадження: Створення плану впровадження системи в реальних командах, включаючи навчання та підтримку користувачів.

### Обґрунтування проєкту:

До актуальності цього проєкту відносяться фактори розвитку структури сучасного бізнесу - підвищення важливості ефективного управління проєктами, потреби в активній оптимізації робочих процесів, необхідність покращення комунікацій та підтримки темпу виконання завдань. Розробка системи управління проєктами що представить інструменти для вирішення цих проблем і при цьому буде адаптованою під потреби малих команд є актуальним напрямком діяльності.

Ключові позиції включають:

- зростаючі вимоги до продуктивності: завдяки структурованому підходу та централізації інформації, команди зможуть виконувати більше завдань у коротші терміни;
- необхідність покращення комунікації: збільшення обсягу внутрішніх комунікацій і зменшення кількості непорозумінь;
- впровадження Agile-методів: забезпечення команди інструментами для реалізації Agile-підходів, що дозволить легко адаптуватися до змін у вимогах і пріоритетах.

Основною інсентивою для створення такого продукту є надання альтернатив для починаючих команд або стартапів, що хочуть застосувати методологію Agile для виконання своїх проєктів, але мають обмежені ресурси або технічну компетенцію. Виходячи з аналізу середовища та альтернатив (представлено в наступному розділі) можна скласти список особливостей, на яких потрібно сконцентруватись:

- продукт повинен бути легким у освоєнні без необхідності проходити складні навчання;
- користувачі повинні мати можливість швидко почати роботу, не витрачаючи час на налаштування;
- можливість адаптувати функціонал під конкретні потреби команди;
- підтримка інтеграції з іншими інструментами, в врахуванням масштабів таких проєктів;
- продукт повинен бути доступним для малих команд, з привабливими тарифами, можливо, з безкоштовним базовим планом;
- адаптивність до специфічних потреб.

Можна зазначити, що кожен етап адаптований під принципи Agile [2].

Завдання:

1. Формування команди та розподіл ролей - на старті проєкту необхідно зібрати команду фахівців, які будуть відповідальні за кожен аспект реалізації продукту.

2. Визначення цілей і вимог продукту - цей етап зосереджений на з'ясуванні потреб кінцевих користувачів і специфікації вимог до продукту.

3. Розробка дорожньої карти - цей етап включає планування всього процесу розробки, встановлення пріоритетів завдань і визначення часових рамок.

4. Створення дизайну продукту - на основі отриманих вимог команда дизайнерів працює над візуальним оформленням і прототипами продукту.

5. Розробка MVP (Minimum Viable Product) - основна мета цього етапу – створення базової версії продукту, що містить мінімально необхідний функціонал для тестування.

6. Тестування та зворотний зв'язок - на цьому етапі команда тестувальники перевіряють продукт на відповідність вимогам, виправляють помилки та пропонують можливі покращення.

7. Впровадження покращень і реліз - після аналізу зворотного зв'язку, команда впроваджує оновлення, додає новий функціонал або покращує існуючий.

8. Підтримка та розвиток продукту - після релізу команда продовжує працювати над покращенням продукту, зважаючи на нові запити користувачів та зміни на ринку.

З кожним етапом проєкту команда керується Agile-підходом, що передбачає ітераційний характер розвитку. Це означає, що на будь-якому етапі процесу можна зробити корективи, змінити пріоритети або адаптувати план відповідно до нових умов або зворотного зв'язку від користувачів. Такий підхід забезпечує швидке реагування на зміни і створення продукту, який точно відповідає очікуванням та потребам кінцевих користувачів [3].

## **Висновки до розділу 1**

У цьому розділі було здійснено всебічний аналіз проблематики поставленого завдання, що включав дослідження сучасних викликів, з якими стикаються малі команди під час впровадження Agile-методології у свої робочі процеси. Було визначено ключові аспекти, що потребують вирішення, зокрема складність використання існуючих інструментів, висока вартість навчання персоналу та адаптації, а також недостатня увага до специфічних потреб малих команд.

Під час аналізу були сформовані конкретні вимоги та норми для розробки програмного продукту, що забезпечують досягнення поставлених цілей. Особливу увагу приділено інтуїтивності інтерфейсу, адаптивності функціоналу та спрощенню процесу впровадження Agile для новачків.

Результати аналізу дозволили уточнити та скоригувати цілі й мету проєкту, визначивши пріоритети та основні напрями роботи. Було визначено оптимальні методи, якими ці цілі можуть бути досягнуті, зокрема розробка легкого у використанні інструменту, що поєднує простоту, доступність і навчальну функцію.

Таким чином, перший розділ заклав фундамент для подальшої роботи над проєктом, визначивши проблемні аспекти, потреби цільової аудиторії та шляхи їх вирішення.

## РОЗДІЛ 2

### ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ ЗАСТОСУНКУ

#### 2.1. Методи гнучкого управління проєктами

Для керування проєктами існують різні підходи, що мають застосовуватись під час виконання робіт, і поділяються вони на гнучкі та каскадні:

- *гнучке управління* проєктами представляє собою сучасний підхід до організації робочого процесу і базується на інкрементальному розвитку продукту;
- *каскадне управління* є застарілим методом, головним недоліком якого є необхідність великого обсягу істотного планування перед початком виконання проєктних робіт і полягає в максимально чіткому послідовному виконанню планів.

Гнучкі методи дозволяють швидко адаптуватись до змін умов виконання робіт або вимог до фінального продукту та передбачає більш ефективну взаємодію і в середині і на зовні команди.

До основних методів гнучкого управління відносяться:

1. Scrum – методологія, що передбачає розбиття проєкту на спринти (короткі ітерації) з тривалістю в декілька тижнів. Кожен спринт включає планування, щоденні зустрічі, виконання робіт, демонстрацію результатів та ретроспективу.

Ключовими ролями в Scrum є Product Owner, Scrum Master та Development Team.

2. Kanban – метод, що фокусується на візуалізації робочого процесу та обмежені кількості завдань при створенні плану робіт. Використання Kanban – дошки дозволяє оптимізувати робочі процеси та легко виявляти ті, що затримують загальний прогрес.

3. Екстремальне програмування – це методологія, що акцентується на технічній досконалості продукту та якості виконання робіт. Цей метод включає в

себе такі практики як парне програмування, постійна інтеграція та розробка через тестування.

4. Lean Software Development – це підхід, запозичений з виробничої системи Toyota, спрямований на максимальне значення витрат при розробці для підвищення фінальної цінності для користувача [4].

Найважливішими аспектами гнучкого управління є:

- постійне оновлення беклогу робочих процесів та вимог - дозволяє команді оперативно реагувати на зміни у пріоритетах або вимогах і забезпечує гнучкість у реалізації проєкту;
- постійні комунікації всередині команди та назовні, з зацікавленими сторонами - регулярні зустрічі, стендапи, демо-версії та ретроспективи сприяють обміну інформацією, вирішенню проблем та підтримці довіри серед усіх учасників проєкту;
- постійне вдосконалення процесів на основі зворотного зв'язку - забезпечує еволюцію команди та проєкту, дозволяючи виправляти помилки та оптимізувати процеси в реальному часі;
- адаптивне планування - дає змогу створювати короткострокові плани, які враховують поточну ситуацію та пріоритети, залишаючи можливість для змін;
- самоорганізація команди - члени команди беруть на себе відповідальність за виконання завдань, розподіл ролей і прийняття рішень, що сприяє ефективності та мотивації.

## **2.2. Моніторинг виконання проєкту**

Моніторинг виконання проєкту означає постійний контроль за прогресом, забезпечення прозорості роботи та адаптацію до змін. Використання таких інструментів, як Notion, допомагає організувати цей процес, але основна увага приділяється методологічним принципам Agile. Планування сприяє ефективному

використанню ресурсів, зменшує кількість переробок і дозволяє фокусуватися на цінності для кінцевого користувача.

Перед початком планування потрібно створити дошку проєкту, що відстежуватиме:

- *беклог продукту* – який містить весь запланований функціонал з детальними описами окремих деталей, і фактично являє собою список вимог, характеристик та специфікацій;
- *спринти* - де розташовані завдання, які мають бути виконані найближчим часом, тобто на протязі декількох тижнів. Також тут вказується опис, виконавець та дедлайн виконання;
- *виконання роботи над окремими елементами, як фронтенд та бекенд* - для зручності відстеження варто поділити завдання на фронтенд та бекенд;
- *завдання в процесі виконання* - для зручності відстеження варто сортувати картки завдань відповідно до їх прогресу, щоб виконавці могли бачити поточний стан виконання робіт;
- *виконані завдання в процесі тестування* - для контролю якості варто створити окремий розділ спеціально під QA, де тестувальник(и) можуть систематизувати знайдені недоліки, для їх подальшого усунення;
- *завершені завдання; Заархівовані завдання* [3].

### **2.3. Планування процесів проєкту**

Система планування робочих процесів проєкту складається з взаємопов'язаних компонентів, які разом створюють комплексний підхід до організаційної роботи:

- *стратегічне планування* є фундаментом такої системи. На рівні стратегічного планування визначаються цілі проєкту, його мета та місія. Також тут проводиться аналіз сектору та сегменту ринку, потенційної цільової

аудиторії, оцінка мікро- та макро- середовища та формування стратегії реалізації. Для успіху проекту важливо визначити чіткі цілі та критерії виконання робочих процесів;

- *планування ресурсів* полягає в визначенні необхідній ресурсній базі у вигляді людських та матеріальних ресурсів, необхідних для виконання робочих процесів. Результатом є чіткі вимоги та стандарти щодо розподілення та використання ресурсів;

- *часове планування* передбачає створення календарного плану проекту. Створюється структура робіт, в якій визначено необхідні завдання, їх взаємозв'язки та витрати у вигляді часу;

- при *плануванні якості* визначаються стандарти та метрики, яким мають відповідати кінцеві результати. Виконується розробка процедури забезпечення якості, методи тестування та перевірки результатів. Встановлюються критерії прийняття робіт та процеси управління змінами;

- *планування комунікацій* охоплює всі головні аспекти взаємодії між учасниками проекту. Визначаються канали комунікації, графік зустрічей, формати документації та звітностей. Також визначається матриця комунікацій, де визначається які виконавці мають отримувати конкретну інформацію та за який час;

- *планування ризиків* передбачає ідентифікацію потенційних загроз та ризиків, що притаманні проекту. Виконується розробка та аналіз стратегії врегулювання ризиків, резервні плани та випадки, в яких вони використовуватимуться. Визначається регулярність перегляду реєстру ризиків;

- *фінансове планування* включає в себе розробку бюджетного плану проекту, планування використання джерел фінансування та створення системи фінансового контролю. Виконується аналіз прямих та непрямих витрат і планування на випадок непередбачених ситуацій [7].

## **Висновки до розділу 2**

В цьому розділі було проведено детальний аналіз існуючих гнучких методологій розробки продуктів, розглянуто приклади застосування цих методологій, проаналізовано їх переваги та недоліки, доцільність використання під час роботи над проєктами і особливості цього процесу.

Були розглянуті канбан дошки, завдяки яким виконується керування роботами на проєкті, доцільність їх використання, особливості їх застосування під час роботи, окремі функції та функціонал, що вони представляють.

Були розібрані процеси планування проєкту, розглянуті взаємопов'язані компоненти, що створюють комплексний підхід до організації роботи.

Виходячи з висновків, зроблених в цьому розділі, в наступному створено на представлено відповідні таблиці.

## РОЗДІЛ 3

### ПЛАНУВАННЯ РОБІТ НАД ЗАСТОСУНКОМ

#### 3.1. Розробка послідовних етапів та переліку робіт

*Зовнішнє середовище* (макро- та мікро- маркетингове середовище) організації формується чинниками, що є об'єктивними та впливають на досягнення поставленої в проєкті мети.

Після проведення аналізу чинників зовнішнього середовища організації можна побудувати список критеріїв впливу факторів на проєкт і його виконання.

На рівні мікросередовища аналіз охоплює внутрішні аспекти проєкту, які безпосередньо впливають на його виконання. Сюди входять ресурси команди, компетенції учасників, доступні технології та інструменти. Наприклад, для розробки застосунку потрібно оцінити, чи вистачає команди досвіду у створенні UI/UX-дизайнів, чи є доступ до потужних засобів розробки, таких як сучасні фреймворки, та чи достатньо виділено фінансування для підтримки ключових етапів. Цей аналіз також включає вивчення внутрішніх процесів команди, ефективність комунікації та здатність до впровадження методології Agile.

Особлива увага приділяється цільовій аудиторії. Для мікросередовища це означає вивчення потреб малих команд розробників, які є основними користувачами продукту. Аналіз їхніх запитів допомагає визначити, які саме функції – наприклад, простота у використанні, інтуїтивний інтерфейс або можливість кастомізації робочих процесів – мають бути в пріоритеті. Цільова аудиторія активно залучається до процесу, забезпечуючи постійний зворотний зв'язок на всіх етапах розробки.

Окремо варто зазначити регуляторні аспекти. Для продуктів, які працюють із даними команд, необхідно враховувати правила захисту даних, такі як GDPR. Це вимагає створення системи, яка забезпечує безпеку і конфіденційність інформації.

Аналіз мікро- та макросередовища забезпечує всебічний погляд на проєкт, допомагає виявити потенційні проблеми та створити стратегії для їх вирішення. Врахування цих факторів є основою для розробки успішного продукту, який відповідатиме очікуванням як внутрішніх стейкхолдерів, так і кінцевих користувачів.

Критерії впливу факторів визначаються наступним чином:

- 1-3 бали – не впливають фактори
- 4-6 бали – мінімальний вплив факторів
- 7-8 бали – впливають фактори
- 9-10 бали – максимальний вплив факторів

Проведемо оцінку впливу факторів макросередовища та результати сформуємо в таблиці 3.1.

*Таблиця 3.1 - Оцінка впливу факторів макросередовища*

Чинники маркетингового середовища	Максимальний вплив, 10 балів
Соціально-демографічні:	
Рівень стану соціальної сфери	6
Рівень народжуваності	6
Національний склад населення	2
Суспільні традиції і норми поведінки	2
Соціальна стабільність	8
Щільність населення	9
Економічні:	
Рівень доходів населення	6
Рівень безробіття	5
Вплив валютних курсів	4
Попит – пропозиція запропонованих послуг	10
Екологічні:	
Стан довкілля	4
Забезпеченість ресурсами	8
Кліматичні умови	4
Чинники маркетингового середовища	Максимальний вплив, 10 балів

Технологічні:	
Рівень розвитку науки	6
Інноваційний рівень	7
Рівень розвитку цифровізації	9
Захист інтелектуальної власності	6
Політико-правові:	
Наявність законодавчих актів	5

*Джерело: розроблено автором*

На основі проведеного аналізу в табл. 2.1 можна зробити висновки, що найбільшим фактором впливу на цей проект буде маркетингове середовище. Продукту проекту доведеться змагатись з альтернативами, що мали час гарно себе зарекомендувати. Протистояти цьому можна буде завдяки орієнтації на специфічну частину цього сегменту ринку і акценту на функціонал, що покаже себе найкраще саме у цих умовах.

Після проведення оцінки впливу факторів макросередовища можна провести оцінку мікросередовища застосунку (Таблиця 3.2).

*Таблиця 3.2 – Оцінка макросередовища*

Чинники мікро-маркетингового середовища	Кількісне значення
Споживачі (цільові групи):	
Кількість цільових груп	9
Кількість запитів	6
Кількісна оцінка потреби	7
Ступінь забезпеченості даною послугою	10
Посередники:	
Кількість посередників	0
Постачальники:	
Наявність ресурсів	3
Кількість матеріальних ресурсів	8
Контактні аудиторії:	
Органи самоврядування	7
ЗМІ	5
Громадські організації	8

*Джерело: розроблено автором*

Оцінка впливу факторів мікросередовища застосунку показує, що проєкт може мати функціональну компетенцію бути конкурентноздатним, проте має орієнтуватись на специфічну клієнтуру.

### 3.2. Ситуаційна оцінка стейкгольдерів проєкту

Стейкгольдерами такого проєкту можуть виступати сторони, що зацікавлені у новому погляді на існуючу екосистему (табл 3.3)

*Таблиця 3.3 - Ситуаційна оцінка стейкгольдерів*

Критерії оцінки	Оцінка стейкгольдерів	
	стейкгольдер 1 – зацікавлені сторони індустрії	стейкгольдер 2 – зацікавлені сторони сектору ринку
Досвід роботи	10	5
Імідж	7	9
Тенденції розвитку	6	9
Потенціал розвитку	8	6

*Джерело: розроблено автором*

Згідно до інформації, представленої в таблиці 2.4 можна зробити висновок, що проєкт може зацікавити стейкгольдерів, що розглядають експлуатацію специфічної ніші ринку або впровадження додаткової конкуренції в існуючу екосистему.

### 3.3 Етапи та зміст робіт проєкту

Розробка застосунку передбачає чітко визначену послідовність етапів, що ґрунтується на принципах гнучкої методології Agile. Цей підхід дозволяє

адаптувати робочі процеси до змін і забезпечує стабільний прогрес навіть у випадку виникнення нових вимог.

Проект розпочинається з етапу дослідження та підготовки, який включає аналіз потреб цільової аудиторії, визначення основного функціоналу та вивчення конкурентного середовища. Цей етап є критично важливим для формування бачення продукту, розробки дорожньої карти проекту та створення backlog'у із завданнями, впорядкованими за пріоритетами.

Після завершення підготовчого етапу проект переходить до фази планування та створення прототипу. Тут команда створює початковий дизайн інтерфейсу, моделює користувацький досвід (UX) і розробляє прототип, який демонструє ключові функції майбутнього застосунку. Цей прототип активно тестується та обговорюється з цільовою аудиторією для отримання зворотного зв'язку.

Основна частина розробки виконується під час ітераційної реалізації функціоналу, що поділяється на спринти. Кожен спринт триває від 1 до 2 тижнів і зосереджений на створенні чітко визначеного набору функцій, таких як створення канбан-дошки, налаштування командного календаря чи інтеграція інструментів сповіщення. На кінець кожного спринту команда проводить демонстрацію результатів та аналізує, чи відповідає створений функціонал початковим вимогам.

Тестування та вдосконалення виконаного функціоналу відбувається постійно протягом усього життєвого циклу проекту. Цей етап включає як технічне тестування (перевірку коду, пошук багів), так і користувацьке тестування, яке дозволяє оцінити зручність і практичність застосунку. На основі отриманих результатів вносяться необхідні корективи та вдосконалення.

Завершальним етапом є реліз продукту та підтримка після запуску. Команда готує стабільну версію застосунку для публічного використання, проводить презентацію можливостей продукту та забезпечує його запуск на

цільових платформах. Після релізу проєкт переходить у фазу технічної підтримки, яка включає моніторинг роботи застосунку, вирішення проблем та оновлення функціоналу відповідно до нових потреб користувачів [7].

Такий підхід (табл 3.4) дозволяє організувати роботу команди послідовно, чітко дотримуючись поставлених цілей, та залишатися гнучкими у відповідь на змінні умови проєкту. Це забезпечує максимальну ефективність роботи й створення продукту, що відповідає потребам ринку та очікуванням користувачів.

*Таблиця 3.4 - Етапи та зміст робіт проєкту*

Етапи роботи	Код, назва (зміст робіт)
<b>1 етап</b> Розробка концепції проєкту	1.1 Створення технічного завдання 1.2 Аналіз створених вимог ТЗ
<b>2 етап</b> Попередні дослідження	2.1 Аналіз чинників, які можуть впливати на застосунок та його розробку. 2.2 Оцінка конкретних ризиків
<b>3 етап</b> Організація процесів з виконання проєктних робіт	3.1 Структуризація робочих процесів 3.2 Створення графіку 3.3 Розподілення задач 3.4 Створення календарного плану 3.5 Контроль реалізації
<b>4 етап</b> Реалізація проєкту	4.1 Оцінка проведених етапів проєкту 4.2.1 Програмування 4.2.2 Дизайн 4.3.1 Тестування 4.4 Поширення застосунку на релевантних платформах, таких як AppStore та PlayMarket
<b>5 етап</b> Контроль та оцінка результатів роботи	5.1 Оцінка успішності реалізації проєкту.

*Джерело: розроблено автором*

Після цього можна починати продумувати склад майбутньої команди, що і буде виконувати проєкт.

Організація виконання проєктних робіт. Команда створюється на час виконання робіт по проєкту та невелика кількість людей після закінчення робіт залишається, аби підтримувати стан роботи застосунку, робота над оновленнями застосунку тощо.

Команда проєкту – це певна чисельність людей, які працюють разом для досягнення спільної мети. Вони безпосередньо працюють над здійсненням проєкту і підпорядковані керівникові (менеджеру) проєкту. Команда проєкту створюється на період реалізації проєкту і після його завершення розпускається, або фахівці створюють нову команду під новий проєкт.

Формування команди для ІТ-стартапу за методологією Agile вимагає чіткого визначення ролей, підтримки ефективної комунікації та створення позитивної внутрішньої політики [11].

Сформовану команду представлено в наступній таблиці 3.5.

*Таблиця 3.5 - Організація виконання проєктних робіт*

Код роботи	Перелік робіт	Організаційна одиниця (виконавець)
1.1 Створення технічного завдання	1. Оцінка функцій, які варто реалізувати у застосунку 2. Визначення плану виконання робіт	Керівник проєкту, проєктний менеджер
1.2 Аналіз створених вимог ТЗ	1. Аналіз доцільності завдань відтворенню у ТЗ 2. Створення детального плану робіт згідно з затвердженим ТЗ	Керівник проєкту
2.1 Аналіз чинників, які можуть впливати на застосунок та	1. Аналіз та розробка критеріїв впливу зовнішнього та внутрішніх чинників середовища на розробку проєкту 2. Проведення аналізу	Керівник проєкту

його розробку.	розроблених критеріїв	
2.2 Оцінка конкретних ризиків	1. Оцінка ризиків, які можуть впливати на проєкт	Керівник проєкту, проєктний менеджер
3.1 Структуризація робочих процесів	1. Побудова структури робіт	Проєктний менеджер
3.2 Створення графіку	1. Розробка графіку виконання завдань проєкту	Проєктний менеджер
3.3 Розподілення задач	1. Видання сформованих завдань конкретним співробітникам.	Проєктний менеджер
3.4 Створення календарного плану	1. Постановка дедлайнів	Проєктний менеджер
3.5 Контроль реалізації	1. Зустрічі з виконавцями проєкту 2. Контроль виконання завдань	Проєктний менеджер
4.1 Оцінка проведених етапів проєкту	1. Оцінка відповідності завдань до поставлених вимог	Керівник проєкту, проєктний менеджер
4.2.1 Програмування	1. Виконання завдань розробниками 2. Керування робочим процесом проєктним менеджером	Розробники, проєктний менеджер
4.2.2 Дизайн	1. Виконання завдань дизайнерами 2. Керування робочим процесом проєктним менеджером	Дизайнери
4.3.1 Тестування	1. Проведення тестів функціональності застосунку	Тестувальники
4.4 Поширення застосунку на релевантних платформах, таких як AppStore та PlayMarket	1. Викладення застосунку у відкритий доступ	Керівник проєкту, проєктний менеджер
5.1 Оцінка успішності реалізації проєкту	1. Оцінка успішності виконаних робіт 2. Аналіз відгуків користувачів	Керівник проєкту, проєктний менеджер

Виходячи з даних, що представлені в таблиці, можна розробити концепцію невеликої команди, що впорається з виконанням проєкту.

Календарне планування. Створення календарного плану проєкту.

Календарне планування – це процес, що включає визначення строків виконання завдань і етапів проєкту, а також розробку графіку, який відображає, коли кожне з цих завдань має бути виконане.

При календарному плануванні виконується аналіз всіх етапів проєкту і розбиття їх на конкретні завдання, загальна організація роботи та розподіл ресурсів.

У ході реалізації проєкту можуть використовуватись Календарні або Функціональні (ФКПР) плани. Функціональні плани поділяються за типами робіт, глибиною планування та формою подання. Можна зазначити, що створення коректного плану робіт вплине в першу чергу на координацію процесів при виконанні проєкту.

Створення календарного плану допомагає точно структурувати усі майбутні роботи, які буде виконано задля реалізації проєкту [3].

Календарний план проєкту представлений в наступній таблиці 4.1.

*Таблиця 3.6 -Календарний план проєкту*

Код роботи	Види робіт	Тривалість, дні
1.1	Створення технічного завдання	2
1.2	Аналіз створених вимог ТЗ	2
2.1	Аналіз чинників, які можуть впливати на застосунок та його розробку.	2
2.2	Оцінка конкретних ризиків	2
3.1	Структуризація робочих процесів	3

3.2	Створення графіку	1
3.3	Розподілення задач	1
3.4	Створення календарного плану	2
3.5	Контроль реалізації	4
4.1	Оцінка проведених етапів проєкту	1
4.2.1	Програмування	20
4.2.2	Дизайн	10
4.3.1	Тестування	5
4.4	Поширення застосунку на релевантних платформах, таких як AppStore та PlayMarket	5
5.1	Оцінка успішності реалізації проєкту.	2

*Джерело: розроблено автором*

Відповідно до представленого календарного плану, загальний час, що має бути витрачений на реалізацію проєкту становитиме 62 дні.

Ресурсне забезпечення. Визначення потреби у людських ресурсах по проєкту.

Ресурсне забезпечення проєкту має на увазі робочу силу, обладнання, матеріали та витратні кошти, що необхідні для виконання робіт, що і є умовою для здійснення проєкту. Щоб робочий процес виконувався згідно до графіку та з належною якістю, ресурси повинні бути правильно розподілені з урахуванням пріоритетів.

Для забезпечення належного постачання відповідних ресурсів для кожного робочого процесу календарне планування пов'язується з виконанням робіт, і конкретними цілями є:

- забезпечити вчасне фінансування;

- координувати використання ресурсів;
- забезпечити постійну наявність потрібних ресурсів;
- передбачити потреби щодо фінансових витрат і ресурсів та їх раціональний розподіл;
- забезпечити вчасне виконання робочого плану

У випадку якщо потреба в ресурсах перевищує можливості, існують такі шляхи врегулювання:

- призупинити або уповільнити роботу в межах критичного часу;
- скоригувати темп виконання робіт відповідно обмеженням;
- скоригувати інтенсивність використання ресурсів [11].

Необхідні та наявні людські ресурси для реалізації проєкту визначені в таблиці нижче (табл 3.7).

*Таблиця 3.7 - Визначення потреби у людських ресурсах*

Вид ресурсу	Необхідна кількість, чол.	Тривалість використання ресурсу	Термін початку використання ресурсу
Керівник проєкту	1	Постійно	02.01.2025
Проектний менеджер	1	Постійно	02.01.2025
Тестувальник	1	Постійно	01.03.2025
Розробник	3	Постійно	01.01.2025

*Джерело: розроблено автором*

Отже, для реалізації проєкту необхідна кількість людських ресурсів становитиме 6 осіб.

Кошторис та економічна ефективність. Кошторис проєкту – це документ, який містить детальний перелік витрат, пов'язаних з реалізацією проєкту. Він використовується для планування та контролю бюджету проєкту, а також для прийняття рішень щодо його доцільності.

Створення кошторису проєкту є ключовим етапом планування, адже саме він дозволяє оцінити необхідні ресурси, час та витрати для реалізації поставлених цілей. Для розробки застосунку з управління проєктами, орієнтованого на малі команди, формування кошторису базується на детальному аналізі задач, визначенні ресурсів та врахуванні потенційних ризиків.

Перш за все, основою кошторису стає чітко сформульований план проєкту, в якому зазначені всі етапи розробки. Цей план розбивається на окремі ітерації або спринти, як це передбачає методологія Agile. Для кожного етапу визначаються конкретні задачі, які мають бути виконані, та оцінюється час, необхідний для їх реалізації. На основі цього визначаються години роботи команди, що є основою для подальших фінансових розрахунків.

Наступним важливим кроком є оцінка людських ресурсів. Для реалізації проєкту необхідно зібрати команду, яка включає розробників, дизайнера, тестувальника та проджект-менеджера. Заробітна плата кожного члена команди стає значним компонентом загального бюджету. Тут важливо врахувати не лише базову оплату праці, але й можливі витрати на залучення зовнішніх фахівців, якщо у команди бракує певних компетенцій, наприклад, для реалізації специфічного функціоналу.

Технічні ресурси також становлять важливу частину кошторису. Хоча для проєктів такого типу зазвичай використовуються популярні інструменти з безкоштовними планами, як-от Notion для організації роботи, можуть виникати додаткові витрати на підписки на більш розширені пакети, які забезпечують доступ до необхідних функцій. Крім того, слід врахувати витрати на

інфраструктуру, зокрема на оренду серверів або використання хмарних сервісів для тестування та деплою продукту.

Окремим пунктом є витрати на дослідження і аналіз. Перед початком розробки важливо провести аналіз потреб цільової аудиторії та конкурентного середовища. Це може включати витрати на опитування, фокус-групи або залучення експертів. Ці дослідження дозволяють краще розуміти запити користувачів, що в майбутньому сприяє створенню більш ефективного продукту.

Не варто забувати і про витрати, пов'язані з забезпеченням якості. Сюди входить тестування продукту, як внутрішнє, так і зовнішнє, на базі реальних користувачів. Витрати на тестування включають оплату роботи тестувальників, організацію тестових середовищ та виправлення виявлених багів.

Важливо врахувати також ризики, які можуть виникнути під час виконання проєкту. Наприклад, затримки через технічні складнощі, зміни в команді або додаткові вимоги клієнтів можуть вплинути на фінальний бюджет. У кошторисі слід передбачити певний резерв на непередбачені витрати, що допоможе уникнути перевищення бюджету у разі виникнення складнощів.

Остаточний кошторис формується шляхом підсумовування усіх зазначених статей витрат. Він регулярно переглядається та коригується залежно від прогресу проєкту, що є важливим аспектом роботи за методологією Agile. Такий підхід забезпечує гнучкість у розподілі ресурсів і дозволяє адаптувати бюджет до змін, що виникають під час розробки.

Таким чином, побудова кошторису є не лише фінансовим, але й стратегічним процесом, що охоплює всі аспекти проєкту: від планування і організації роботи команди до забезпечення якості кінцевого продукту. Це дозволяє не тільки уникнути перевитрати бюджету, але й створити основу для успішної реалізації проєкту [4].

Кошторис проєкту представлений у вигляді таблиці 3.8.

Таблиця 3.8 – Кошторис проекту з розробки застосунку

Код роботи	Опис робіт	Час на реалізацію	Заплановані витрати за одиницю, грн.	Заплановані витрати загалом сума, грн.
1.1	Створення технічного завдання	2	1000	2000
1.2	Аналіз створених вимог ТЗ	2	500	1000
2.1	Аналіз чинників, які можуть впливати на застосунок та його розробку.	2	500	1000
2.2	Оцінка конкретних ризиків	2	300	600
3.1	Структуризація робочих процесів	3	0	0
3.2	Створення графіку	1	200	200
3.3	Розподілення задач	1	300	300
3.4	Створення календарного плану	2	200	400
3.5	Контроль реалізації	4	1000	4000
4.1	Оцінка проведених етапів проекту	1	300	300
	<b>Основна розробка застосунку</b>			
4.2.1	Програмування	20	1000	20000
4.2.2	Дизайн	10	800	8000
	<b>Тестування</b>			

Таблиця 3.8 – Кошторис проєкту з розробки застосунку

Код роботи	Опис робіт	Час на реалізацію	Заплановані витрати за одиницю, грн.	Заплановані витрати загалом, сума, грн.
4.3.1	Основне тестування	5	700	3500
4.4	Поширення застосунку на релевантних платформах, таких як AppStore та PlayMarket	2	1000	2000
5.1	Оцінка успішності реалізації проєкту	3	200	600
Загальні витрати				43900

*Джерело: розроблено автором*

З розрахунків можна зробити висновок, що приблизний капітал проєкту складатиме 43900 гривень.

### **Висновки до розділу 3**

В цьому розділі було застосовано навички організації та планування робіт по проєкту. Було складено календарний план робіт з оцінкою часу, який буде використано для виконання повного циклу робіт.

Було розроблено послідовний план виконання робіт, були розглянуті потенціальні стейкхолдери, систематизовано склад людських ресурсів.

Також було складено таблицю ресурсного забезпечення, що відповідатиме за розподіл ресурсів та витрат, що у свою чергу призведе до збалансованого циклу виконання робіт.

Було створено кошторис та розрахунок економічної цінності проекту, з чого можна представити детальний опис необхідних ресурсів у вигляді ресурсів, часу і людей.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було здійснено дослідження методології Agile та її застосування при розробці програмного забезпечення. Було проаналізовано структуру Agile-методології, принципи її впровадження та особливості використання у формуванні нових команд або адаптації вже існуючих. Особлива увага приділялася вивченню способів інтеграції гнучких методів у робочі процеси малих команд, що мають обмежений досвід у використанні подібних підходів.

Робота також включала вивчення принципів правильного планування виконання робочих процесів, ефективного розподілення ресурсів і створення планів, що забезпечують досягнення поставлених цілей у задані строки. Було розглянуто не лише теоретичні основи Agile, а й практичні аспекти: організацію спринтів, управління беклогом, проведення щоденних стендапів і постійного вдосконалення на основі зворотного зв'язку.

Окремим напрямом дослідження стало вивчення особливостей впровадження Agile у малі команди. Проаналізовано переваги, які дає така трансформація, зокрема підвищення прозорості, гнучкості та ефективності роботи, а також потенційні недоліки, як-от труднощі адаптації учасників без досвіду роботи з Agile. Було ідентифіковано ключові виклики, з якими стикаються починаючі команди, такі як обмежений доступ до ресурсів і знань, а також необхідність швидкого освоєння нових методів роботи.

На практичному рівні виконання проєкту сприяло здобуттю цінних навичок у плануванні та організації роботи команди, зокрема:

- створення загального плану робіт і календарного плану;
- складання кошторису, що враховує специфіку проєкту;
- формування команди для розробки продукту та управління нею на різних етапах реалізації.

Завдяки роботі над проєктом також було відпрацьовано механізми комунікації як всередині команди, так і з зацікавленими сторонами, розроблено підхід до моніторингу виконання завдань і оцінювання прогресу. Цей досвід дозволяє використовувати напрацьовані методики для розробки програмних продуктів, орієнтованих на малі команди, і формує основу для подальшого вдосконалення навичок у галузі управління проєктами.

Результати дослідження та виконаної роботи мають практичне значення, оскільки можуть бути використані для ефективного управління проєктами в малих командах, що прагнуть впровадити Agile. Вони також можуть слугувати основою для навчання студентів або молодих спеціалістів, які лише починають свій шлях у сфері ІТ.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дослідження конкурентів. Електронний ресурс. URL: <https://www.nuclino.com/solutions/notion-vs-jira>
2. Дослідження про Agile в секторі ринку. Електронний ресурс. URL: [https://www.researchgate.net/publication/264167913\\_Agile\\_Software\\_Development\\_Framework\\_in\\_a\\_Small\\_Project\\_Environment](https://www.researchgate.net/publication/264167913_Agile_Software_Development_Framework_in_a_Small_Project_Environment)  
URL: <https://medium.com/the-liberators/heres-why-you-should-use-agile-for-small-projects-b84883dd506b>
3. Дослідження про керування малими проєктами. Електронний ресурс. URL: [https://www.researchgate.net/publication/334966543\\_The\\_Role\\_of\\_Project\\_Management\\_in\\_the\\_success\\_of\\_Small\\_Businesses](https://www.researchgate.net/publication/334966543_The_Role_of_Project_Management_in_the_success_of_Small_Businesses)  
URL: <https://www.pmi.org/learning/library/managing-small-projects-9035>
4. Дослідження про побудову додатків у малих проєктах. Електронний ресурс. URL: <https://medium.com/@suefletcherwatson/how-to-design-a-small-research-project-1d061a43e352>  
URL: [https://www.researchgate.net/publication/272813672\\_A\\_Software\\_Development\\_Process\\_for\\_Small\\_Projects](https://www.researchgate.net/publication/272813672_A_Software_Development_Process_for_Small_Projects)
5. Agile software development. Електронний ресурс. URL: [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)
6. Agile Model. Електронний ресурс. URL: <https://www.javatpoint.com/software-engineering-agile-model>
7. Agile software development: Best practices, methodologies, and templates. Електронний ресурс. URL: <https://www.aha.io/roadmapping/guide/agile/agile-software-development>
8. Sprint Planning. Електронний ресурс. URL: <https://www.atlassian.com/agile/scrum/sprint-planning>

9. Full Comparison: Agile vs Scrum vs Waterfall vs Kanban. Електронний ресурс. URL: <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>
10. The Do's & Don'ts of Resource Planning in Agile Organizations. Електронний ресурс. URL: <https://www.runn.io/blog/agile-resource-planning>
11. Agile teams in action: from formation to high performance. Електронний ресурс. URL: <https://www.atlassian.com/agile/teams>
12. What Is Agile Transformation? . Електронний ресурс. URL: <https://www.wrike.com/agile-guide/agile-transformation/>
13. Agile manifesto як статут інноваційної методології розробки програмного забезпечення. Електронний ресурс. URL: <https://conf.krok.edu.ua/ММО/ММО-2024/paper/view/2056>
14. Адаптація Agile-методологій для цифровізації процесів управління проектами. Електронний ресурс. URL: <https://conf.krok.edu.ua/ММО/ММО-2024/paper/view/2128>