

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Гнучке управління створенням ігрової платформи «GameHub» із застосуванням сучасних методологій для оптимізації розробки»

Ступінь вищої освіти – магістр

Спеціальність – 073 «Менеджмент»

Освітня програма «Agile-технології розробки програмного забезпечення»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Керівники: зав. кафедри комп'ютерних наук,
к.е.н., с.н.с., доцент
Сергій МІЧКІВСЬКИЙ
старший викладач кафедри
комп'ютерних наук
Олег ЛУКУТІН

Виконав: здобувач
групи МЕН/Agile-24м
Дмитро ФЕДОРЧУК

Засвідчую, що кваліфікаційна
робота оформлена відповідно до
ДСТУ 3008:2015 та не містить
запозичень з праць інших авторів
без відповідних посилань.

Здобувач: _____
(підпис)

Київ, 2026 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
Федорчук Дмитро Олексійович**

Тема роботи	Гнучке управління створенням ігрової платформи “GameHub” із застосуванням сучасних методологій для оптимізації розробки
Номер та дата наказу про затвердження теми	№ 109-3 від 14 жовтня 2025 року
Коротка постановка завдання	Розробити та обґрунтувати систему гнучкого управління (Agile) проєктом створення універсальної ігрової платформи-агрегатора «GameHub». Метою роботи є вирішення проблеми фрагментації цифрового контенту та технічної нестабільності інтеграцій через впровадження гібридної методології управління (Scrum для ядра, Kanban для плагінів) та моделі Community-Driven.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	Про цифровий контент та цифрові послуги : Закон України від 10.08.2023 № 3321-IX. URL: https://zakon.rada.gov.ua/laws/show/3321-20#Text (дата звернення: 07.12.2025). Швабер К., Сазерленд Дж. Home Scrum Guides. URL: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Ukrainian.pdf (дата звернення: 07.12.2025).
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 Agile-технологій.

Дата видачі завдання «27» жовтня 2025 р.

Керівник

Олег ЛУКУТІН

Керівник

Сергій МІЧКІВСЬКИЙ

Здобувач

Дмитро ФЕДОРЧУК

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напрямку дослідження та керівника.	01.09.2025 р.	<i>виконано</i>
2	Формування теми та призначення керівника.	22.09.2025 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи.	09.10.2025 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу.	27.10.2025 р.	<i>виконано</i>
Основний етап			
5	Розробка концепції кваліфікаційної роботи.	06.11.2025 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів.	08.11.2025 р.	<i>виконано</i>
7	Теоретико-методичний аналіз предметної області та розширена постановка завдання. Підготовка та подання керівнику розділу 1 кваліфікаційної роботи.	13.11.2025 р.	<i>виконано</i>
8	Дослідницько-аналітична робота. Підготовка та подання керівнику розділу 2 кваліфікаційної роботи.	20.11.2025 р.	<i>виконано</i>
9	Розробка рекомендацій щодо вдосконалення управління із застосуванням Agile-технологій. Підготовка та подання керівнику розділу 3 кваліфікаційної роботи.	27.11.2025 р.	<i>виконано</i>
10	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи.	01.12.2025 р.	<i>виконано</i>
11	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівнику доопрацьованого варіанту кваліфікаційної роботи	03.12.2025 р.	<i>виконано</i>
Завершальний етап			
12	Представлення рукопису для перевірки на плагіат.	08.12.2025 р.	<i>виконано</i>
13	Підготовка презентації та доповіді на передзахист.	22.12.2025 р.	<i>виконано</i>
14	Передзахист кваліфікаційної роботи.	23-24.12.2025 р.	<i>виконано</i>
15	Технічна самоекспертиза роботи на відповідність вимогам до оформлення та виправлення недоліків.	12-16.01.2026 р.	<i>виконано</i>
16	Експертиза роботи керівником та зовнішнім експертом (рецензентом).	20.01.2026 р.	<i>виконано</i>
17	Доопрацювання доповіді та презентації для захисту.	22.01.2026 р.	<i>виконано</i>
18	Захист кваліфікаційної роботи.	26-30.01.2026 р.	<i>виконано</i>

Керівник

Олег ЛУКУТІН

Керівник

Сергій МІЧКІВСЬКИЙ

Здобувач

Дмитро ФЕДОРЧУК

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ ГНУЧКОГО УПРАВЛІННЯ ІТ-ПРОЄКТАМИ	8
1.1. Еволюція та порівняльний аналіз методологій розробки програмного забезпечення	8
1.2. Специфіка продуктового менеджменту в індустрії цифрової дистрибуції	10
1.3. Обґрунтування вибору Agile-фреймворку для розробки GameHub	11
1.4. Системи вимірювання ефективності та метрики Agile-процесів	14
Висновки до розділу 1	15
РОЗДІЛ 2. ДИЗАЙН БІЗНЕСУ ТА АНАЛІЗ СЕРЕДОВИЩА ПРОЄКТУ «GAMENUB»	16
2.1. Комплексний аналіз ринкового середовища та конкурентних сил	16
2.2. Стратегічне позиціонування та аналіз стейкхолдерів	17
2.3. Побудова бізнес-моделі	19
2.4. Формування візії продукту (Product Vision) та стратегії розвитку	21
2.4. Моделювання шляху користувача (Customer Journey Map)	22
2.5. Економічне обґрунтування та Unit-економіка проєкту.....	24
2.6. Організаційно-правове забезпечення проєкту.....	27
Висновки до розділу 2	28
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ «GAMENUB» ЗА МЕТОДОЛОГІЄЮ AGILE	29
3.1. Формування Product Backlog та планування ітерацій	29
3.2. Організаційна структура команди та ресурсне забезпечення.....	32
3.3. Розрахунок бюджету проєкту (Кошторис)	33
3.4. Аналіз метрик ефективності та звітність у Jira.....	34
3.5. Моніторинг ефективності, контроль якості та CI/CD процеси.....	38
3.6. Управління ризиками проєкту.....	40
Висновки до розділу 3	41
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТОК А	45
ДОДАТОК Б	46

АНОТАЦІЯ

Федорчук Д. О. «Гнучке управління створенням ігрової платформи «GAMEHUB»

В кваліфікаційній роботі розглянуто теоретико-методичні основи гнучкого управління та особливості адаптації Agile-підходів до специфіки ринку цифрової дистрибуції. Представлено дизайн бізнес-моделі та опис платформи-агрегатора «GameHub», спрямованої на вирішення проблеми фрагментації ігрового контенту та оптимізації системних ресурсів. Розглянуті прикладні аспекти проекту: формування беклогу, планування спринтів та бюджету, застосування гібридного підходу Scrumban для підтримки інтеграцій та налаштування процесів у системі Jira.

Ключові слова: Agile, Scrumban, ігрова платформа, Product Backlog, Jira, цифрова дистрибуція.

ANNOTATION

Fedorchuk D. O. «Flexible management of creating the game platform «GAMEHUB»

The qualification paper examines the theoretical and methodological foundations of flexible management and the adaptation of Agile approaches to the digital distribution market. The business model design and the description of the aggregator platform "GameHub" are presented to solve the problem of content fragmentation and PC resource optimization. Applied aspects of the project were considered: backlog formation, sprint scheduling and budgeting, application of the hybrid Scrumban approach for integration support, and process configuration in the Jira system.

Key words: Agile, Scrumban, game platform, Product Backlog, Jira, digital distribution.

ВСТУП

Актуальність теми дослідження. Сучасний світ цифрових розваг переживає фундаментальні зміни. Ринок дистрибуції відеоігор, який ще десять років тому здавався стабільним і монолітним, сьогодні перетворився на поле запеклої конкуренції, відомої як «війна лаунчерів». Гравці стикаються з явищем цифрової фрагментації: щоб пограти в улюблені ігри, доводиться встановлювати та постійно оновлювати півдюжини різних клієнтів (Steam, Epic Games Store, Battle.net тощо). Це не лише незручно, але й неефективно з точки зору використання ресурсів комп'ютера та соціальної взаємодії.

Спроби створити єдиний агрегатор (наприклад, GOG Galaxy) показали, що головною проблемою є не розробка самої програми, а підтримка її працездатності. Сторонні сервіси постійно змінюють свої API, і традиційні, неперороткі методи управління проектами (Waterfall) просто не встигають за цими змінами. Команди витрачають місяці на планування оновлень, які стають застарілими ще до релізу.

Саме тому критично важливим стає перехід до **гнучких методологій управління (Agile)**. Проєкт «GameHub», що розглядається в цій роботі, пропонує вирішення проблеми через побудову адаптивної системи, де швидкість реакції та залучення спільноти стають головними конкурентними перевагами. Тема є актуальною, оскільки поєднує вирішення реальної проблеми користувачів із застосуванням передових управлінських практик.

Зв'язок роботи з науковими програмами. Кваліфікаційна робота виконана в рамках наукових досліджень кафедри інформаційного менеджменту, математики та статистики Університету «КРОК» за спеціальністю 073 «Менеджмент» (ОП «Agile технології розробки програмного забезпечення») [9].

Мета і завдання дослідження. Метою роботи є розробка механізму гнучкого управління створенням платформи-агрегатора «GameHub», що дозволить оптимізувати процес розробки та забезпечити стійкість продукту до змін зовнішнього середовища.

Для досягнення мети необхідно вирішити такі **завдання**:

1. Проаналізувати еволюцію підходів до управління IT-проєктами та довести переваги Agile в умовах невизначеності.
2. Дослідити ринок ігрових лаунчерів, виявити слабкі сторони конкурентів та сформулювати унікальну пропозицію GameHub.
3. Побудувати бізнес-модель проєкту, використовуючи інструменти Lean Canvas та візуалізувати стратегію розвитку (Roadmap).
4. Сформувати структуру продукту та розробити пріоритезований беклог (Product Backlog).
5. Обґрунтувати вибір гібридного Agile-фреймворку та описати процеси командної взаємодії.
6. Розрахувати необхідні ресурси, бюджет проєкту та визначити ключові метрики успіху.

Об'єктом дослідження є процеси управління IT-проєктами в умовах динамічного ринкового середовища.

Предметом дослідження є методи та інструменти гнучкого управління (Agile) при створенні ігрової платформи «GameHub».

Методи дослідження. У роботі використано комплекс методів: *аналіз та синтез* (для опрацювання теоретичної бази), *порівняльний аналіз* (для вивчення конкурентів), *моделювання* (для побудови бізнес-процесів та архітектури), *експертні оцінки* (для планування бюджету).

Наукова новизна полягає у запропонованій адаптивній моделі управління розробкою мета-платформи, яка поєднує Scrum для розвитку ядра продукту та Kanban для підтримки інтеграцій, що дозволяє мінімізувати технічний борг.

Практичне значення. Розроблений план проєкту, включаючи дорожню карту, бюджет та беклог, може бути використаний як основа для запуску реального стартапу.

РОЗДІЛ 1

ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ ГНУЧКОГО УПРАВЛІННЯ ІТ-ПРОЄКТАМИ

1.1. Еволюція та порівняльний аналіз методологій розробки програмного забезпечення

Сучасний ландшафт індустрії розробки програмного забезпечення (Software Engineering) характеризується високим рівнем невизначеності та динамікою змін вимог. Проєкт – це діяльність, за якої матеріальні, фінансові та людські ресурси організовано новаторським шляхом для виконання унікальної роботи при обмеженні у часі та витратах, щоб досягти позитивних змін, визначених кількісними та якісними параметрами [14]. Вибір правильної методології управління стає критичним фактором успіху, особливо коли йдеться про створення складних інтеграційних платформ, таких як «GameHub» [1].

Історично першою системною спробою впорядкувати хаос розробки стала каскадна модель (Waterfall model), яка домінувала в індустрії протягом останніх десятиліть ХХ століття. Ця модель базується на послідовному, лінійному виконанні етапів життєвого циклу: від аналізу вимог до проєктування, кодування, тестування та впровадження. Кожна наступна фаза розпочинається виключно після повного завершення та документального затвердження попередньої.

У контексті великих інфраструктурних проєктів (будівництво, авіабудування) Waterfall демонструє свою ефективність завдяки чіткій структурі та передбачуваності бюджету. Проте, при перенесенні цієї моделі на розробку користувацького ПЗ виявилися її критичні недоліки, відомі як «криза програмного забезпечення». Головна проблема полягає у так званому «ефекті тунелю»: замовник і розробники фіксують вимоги на старті, і команда зникає в «тунелі» розробки на кілька місяців. Коли ж продукт виходить на світло (етап релізу), часто виявляється, що ринкова ситуація змінилася, або початкові гіпотези були хибними.

Для проєкту «GameHub» використання каскадної моделі несе неприпустимі ризики. Специфіка агрегатора ігор полягає у залежності від третіх сторін (Steam, Epic Games). Якщо під час фази проєктування, яка у Waterfall може тривати місяць, компанія Valve змінить протокол авторизації, все написане технічне завдання стане неактуальним ще до початку написання коду. Виправлення такої помилки на пізніх етапах у каскадній моделі коштує експоненціально дорого.

Відповіддю на ці виклики стала поява гнучких методологій (Agile). Офіційним народженням цього напрямку вважається 2001 рік, коли було сформульовано «Agile Manifesto» [10]. Це не просто набір інструкцій, а зміна світогляду, що базується на чотирьох цінностях:

1. Люди та взаємодія важливіші за процеси та інструменти. Успіх GameHub залежить не від дорогого софту для трекінгу задач, а від комунікації між розробниками ядра та спільнотою, що пише плагіни.

2. Працюючого продукту важливіші за вичерпну документацію. Користувачу не цікава архітектурна схема агрегатора, йому важливо, щоб кнопка «Грати» працювала тут і зараз.

3. Співпраця із замовником важливіша за узгодження умов контракту. У нашому випадку замовником виступає спільнота геймерів, чії відгуки формують вектор розвитку.

4. Готовність до змін важливіша за слідування плану [12]. Це ключовий принцип для виживання в умовах «війни лаунчерів».

Сутність Agile полягає в ітеративно-інкрементальному підході.

- *Ітеративність* означає розбиття процесу на короткі цикли (спринти), що дозволяє регулярно звіряти курс із реальністю.

- *Інкрементальність* означає, що в кінці кожного циклу ми отримуємо не напівфабрикат, а готовий до використання шматок продукту (Increment).

Таким чином, перехід від Waterfall до Agile (рис 1.1) – це перехід від спроби передбачити майбутнє до здатності швидко адаптуватися до нього. Для GameHub це означає можливість випускати оновлення щотижня, реагуючи на зміни в ігровій індустрії миттєво.

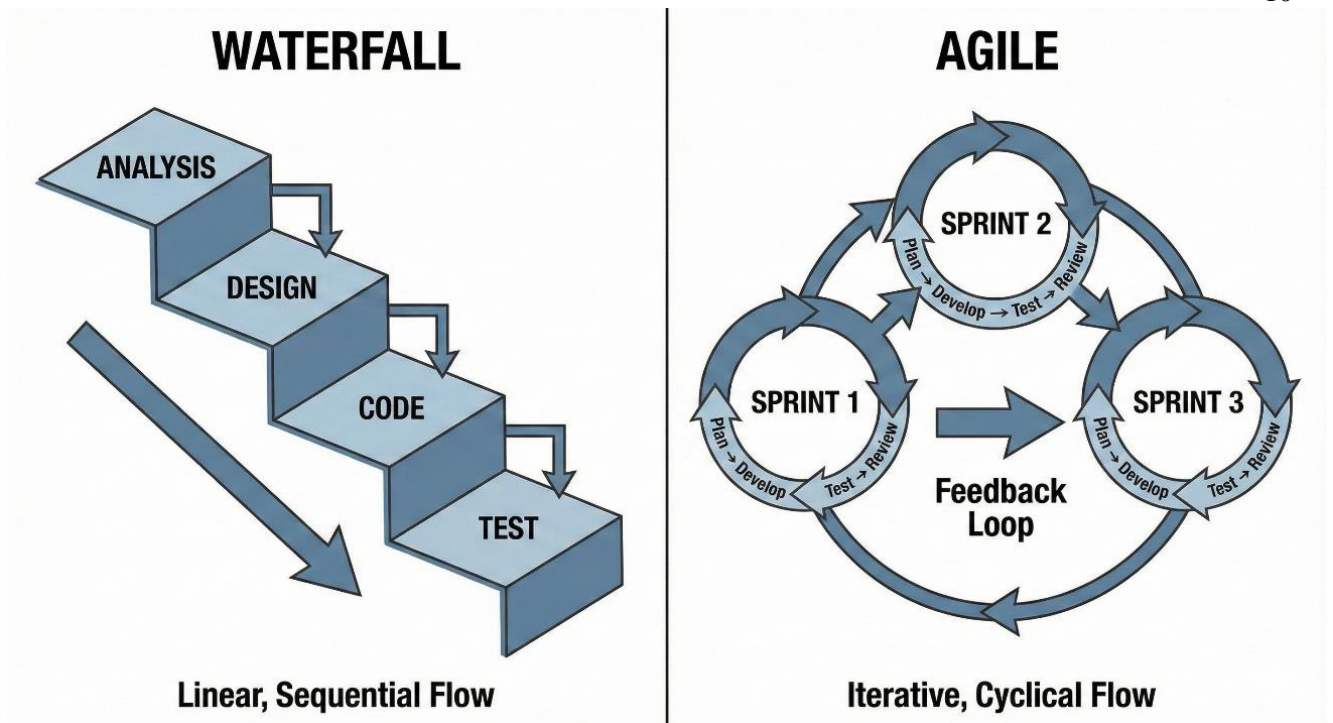


Рисунок 1.1 – Порівняльна характеристика Waterfall та Agile

Джерело: розроблено автором.

1.2. Специфіка продуктового менеджменту в індустрії цифрової дистрибуції

Ринок відеоігор та супутнього програмного забезпечення є унікальним середовищем, яке диктує свої жорсткі умови для менеджменту проєктів. Щоб зрозуміти, як ефективно управляти створенням GameHub, необхідно проаналізувати контекст так званої «Війни Лаунчерів» (Launcher Wars).

Ще 10-12 років тому ринок ПК-геймінгу був фактично монополізований сервісом Steam. Однак, бажання видавців максимізувати прибутки призвело до появи власних екосистем: спочатку Origin (EA) та Uplay (Ubisoft), згодом Battle.net (Blizzard) та агресивний гравець Epic Games Store.

Це створило явище фрагментації користувацького досвіду. Сучасний геймер змушений оперувати 5-7 різними клієнтами. Це породжує специфічні вимоги до продукту-агрегатора, які менеджер проєкту повинен врахувати:

1. Технологічна залежність та управління ризиками. GameHub – це паразитарна система в хорошому сенсі цього слова. Вона будується "поверх" існуючих платформ. Але ці платформи не завжди зацікавлені у співпраці.

Управління таким проєктом нагадує гру в кішки-мишки: як тільки Epic Games оновлює захист API, інтеграція «відвалюється».

- *Вимога до менеджменту:* процеси мають бути налаштовані на Adaptive Maintenance (адаптивну підтримку). У беклозі завжди має бути зарезервований ресурс (buffer) на термінові хотфікси, які неможливо запланувати заздалегідь.

2. Вимогливість аудиторії до продуктивності (Performance Anxiety). Цільова аудиторія продукту – геймери. Це технічно грамотні люди, які інвестують кошти в «залізо» заради високого FPS (кадрів в секунду). Будь-який фоновий процес, що споживає зайві 100 МБ оперативної пам'яті або 5% процесора, сприймається як ворожий.

- *Вимога до менеджменту:* пріоритет нефункціональних вимог (Non-functional requirements). Задачі з оптимізації коду та рефакторингу в GameHub мають такий самий пріоритет, як і нові функції. Це відрізняє цей проєкт від класичного бізнес-софту, де швидкодія часто є вторинною.

3. Роль спільноти як співрозробника (Community-Driven Development). Аналіз провалів конкурентів (зокрема, повільний розвиток GOG Galaxy) показує, що закрита корпоративна модель розробки тут не працює. Жодна компанія не може тримати штат програмістів, які б відслідковували зміни в сотнях ігрових сервісів.

- *Вимога до менеджменту:* менеджер проєкту GameHub має будувати процеси не лише для внутрішньої команди, а й для зовнішніх контриб'юторів (волонтерів). Це вимагає прозорої документації, чітких гайдлайнів для Pull Requests та культури відкритого коду.

1.3. Обґрунтування вибору Agile-фреймворку для розробки GameHub

Серед широкого спектру гнучких методологій для реалізації проєкту «GameHub» доцільно обрати не один «чистий» метод, а гібридний підхід, що поєднує сильні сторони Scrum та Kanban. Такий вибір зумовлений різномірністю задач, які стоять перед командою.

Scrum (рис 1.2) – це легковаговий фреймворк для створення продуктів у середовищі з високою невизначеністю [15]. Він базується на чіткому ритмі роботи.

- *Для чого підходить у GameHub: Розробка Ядра (Core).* Це архітектура додатку, дизайн інтерфейсу, система локальної бази даних, налаштування. Ці задачі є прогнозованими. Ми можемо оцінити, скільки часу займе малювання нового дизайну бібліотеки, і запланувати це в Спринт.

- *Основні елементи:* Ролі (Scrum Master, Product Owner, Developers), Події (Planning, Daily, Review, Retro) та Артефакти (Backlog, Increment) [8].

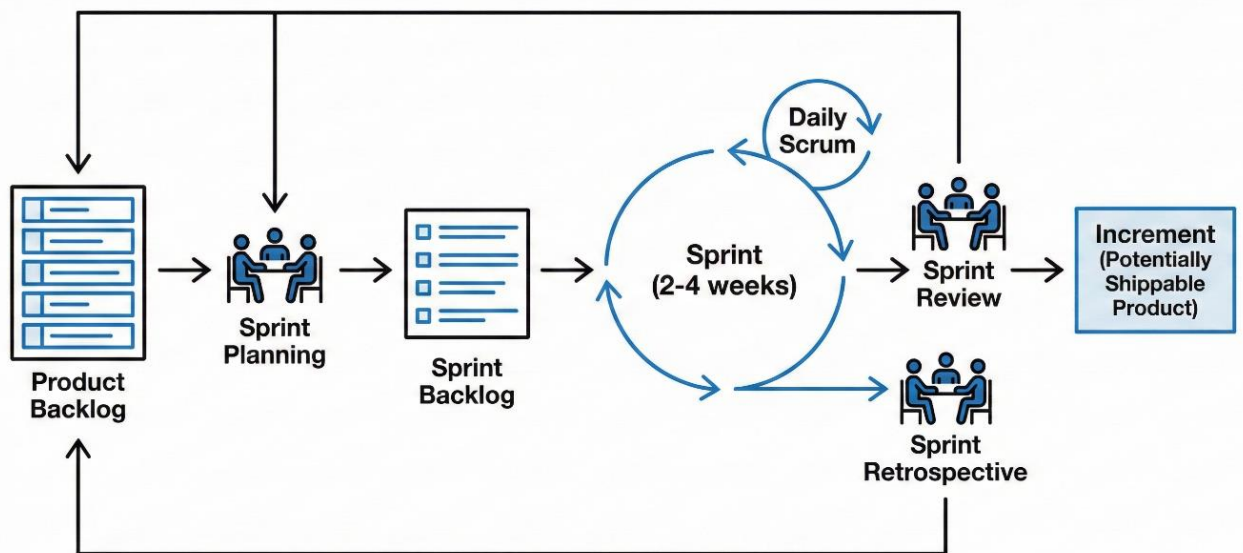


Рисунок 1.2 – Схема фреймворку Scrum

Джерело: розроблено автором.

Kanban – це метод керування потоком роботи з фокусом на візуалізації, обмеженні незавершеної роботи (WIP) та оптимізації часу циклу [2].

- *Для чого підходить у GameHub: Підтримка Інтеграцій (Integrations).* Поломки зовнішніх сервісів стаються хаотично. Ми не можемо запланувати на Sprint Planning, що «в середу зламається вхід у Steam». Коли це стається, це блокуюча проблема. Kanban дозволяє взяти цю задачу в роботу миттєво, не чекаючи початку нового спринту, пропустити її через потік (Flow) і випустити оновлення.

Також варто згадати XP (Extreme Programming) – це набір інженерних практик [13]. Для GameHub критично важливими є такі елементи XP, як:

- *Continuous Integration (CI)*: Автоматична збірка та тестування коду при кожному внесенні змін. Це гарантує, що новий плагін не зламає всю програму;
- *Code Review*: Перехресна перевірка коду для забезпечення високої продуктивності та безпеки.

Окрім базового опису інструментарію, вибір методології доцільно обґрунтувати через призму фреймворку Кеневін (Cynefin Framework), розробленого Дейвом Сноуденом. Ця модель класифікує системи на прості, складні (Complicated), заплутані (Complex) та хаотичні. Проєкт розробки платформи-агрегатора «GameHub» відноситься до заплутаних систем (Complex context). У такій системі причинно-наслідкові зв'язки стають зрозумілими лише ретроспективно. Ми не можемо заздалегідь точно знати, як поведе себе API Steam після чергового оновлення або як користувачі відреагують на новий інтерфейс. У зоні Complex єдиною ефективною стратегією є цикл «Probe – Sense – Respond» (Спробувати – Відчути – Відреагувати). Саме цю стратегію реалізує Scrum через короткі ітерації. Традиційний Waterfall, який намагається застосувати підхід «Sense – Analyze – Respond» (проаналізувати все наперед), у зоні Complex неминуче призводить до аналітичного паралічу та помилкових прогнозів.

Також варто зазначити психологічний аспект впровадження Agile. Перехід до гнучкого управління вимагає трансформації менталітету команди (Agile Mindset). Це перехід від культури «Command and Control» (наказуй і контролюй), притаманної ієрархічним структурам, до культури «Servant Leadership» (лідерство-служіння) [7]. У цій парадигмі роль менеджера проєкту (або Scrum-майстра) полягає не в роздачі завдань, а в створенні середовища, де команда може працювати максимально ефективно, та усуненні організаційних перешкод. Для GameHub, де значна частина роботи залежить від креативності розробників та їхньої ініціативи у вирішенні нестандартних проблем інтеграції, такий підхід до лідерства є критично необхідним фактором успіху [17].

Висновок щодо вибору методології: Для проєкту «GameHub» пропонується модель "Scrumban":

1. Стратегічний розвиток та розробка нових фіч (Features) відбувається за Scrum (2-тижневі ітерації, чіткий план, демо в кінці) [11].

2. Операційна підтримка, багфіксинг та оновлення плагінів відбуваються за Kanban (безперервний потік, релізи за вимогою).

Такий дуалізм дозволяє поєднати стратегічну дисципліну з тактичною гнучкістю, що є єдиним шляхом до виживання продукту в агресивному середовищі цифрової дистрибуції.

1.4. Системи вимірювання ефективності та метрики Agile-процесів

Впровадження гнучких методологій вимагає зміни підходів не лише до планування, а й до контролю. Традиційні метрики, такі як «відхилення від бюджету» або «відсоток виконання плану», є недостатньо інформативними в умовах ітеративної розробки. Для проєкту «GameHub» пропонується використовувати систему метрик, що базується на принципах Evidence-Based Management (EBM).

Ключові групи метрик для оцінки ефективності:

1. Метрики потоку (Flow Metrics):

- Lead Time (Час виконання). Загальний час від моменту появи ідеї (або запиту користувача) до моменту її реалізації у продакшені. Для GameHub критично важливо мінімізувати цей показник при виправленні помилок інтеграції;

- Cycle Time (Час циклу). Час, що витрачається безпосередньо на розробку задачі (від статусу «In Progress» до «Done»). Стабільність цього показника свідчить про зрілість команди;

- Work In Progress (WIP). Кількість задач, що знаходяться в роботі одночасно. Обмеження WIP дозволяє виявити вузькі місця (bottlenecks) у процесі розробки.

2. Метрики цінності (Value Metrics):

- Current Value (CV). Цінність, яку продукт надає користувачеві прямо зараз (наприклад, кількість підтримуваних платформ);
 - Time-to-Market (TTM). Швидкість виходу нових фіч на ринок.
3. Метрики якості (Quality Metrics):
- Change Failure Rate (CFR). Відсоток релізів, що призвели до збоїв або вимагали гарячих виправлень (hotfixes);
 - Defect Density. Кількість дефектів на тисячу рядків коду.

Використання цих метрик дозволяє перейти від суб'єктивних відчуттів («ми працюємо швидко») до об'єктивних даних, що є основою для прийняття управлінських рішень.

Висновки до розділу 1

У першому розділі роботи було проведено глибокий аналіз теоретико-методологічних засад управління ІТ-проектами. По-перше, встановлено, що еволюція від Waterfall до Agile є закономірною реакцією на зростання невизначеності в індустрії. Для продуктів, які залежать від зовнішніх екосистем, жорстке планування є фактором ризику. По-друге, виявлено специфіку предметної області (мета-лаунчери), яка вимагає від менеджменту фокусу на швидкодії, адаптивності та роботі зі спільнотою. По-третє, обґрунтовано доцільність застосування гібридної методології (Scrum + Kanban + елементи XP). Саме такий комбінований підхід дозволить команді GameHub ефективно розробляти новий функціонал, одночасно підтримуючи стабільність існуючих інтеграцій в умовах постійних змін зовнішнього середовища.

РОЗДІЛ 2

ДИЗАЙН БІЗНЕСУ ТА АНАЛІЗ СЕРЕДОВИЩА ПРОЄКТУ «GAMEHUB»

2.1. Комплексний аналіз ринкового середовища та конкурентних сил

Розробка стратегії виведення на ринок нового продукту «GameHub» вимагає багаторівневого аналізу зовнішнього середовища. Окрім виявлення макроекономічних факторів (через PESTEL-аналіз), критично важливим є розуміння інтенсивності конкуренції та привабливості галузі. Для цього в роботі застосовано модель п'яти сил конкуренції Майкла Портера.

1. Загроза появи нових гравців (Середня).

- *Бар'єри входу:* технічний бар'єр є відносно низьким – написати простий лаунчер може невелика команда. Проте, бар'єр утримання користувачів є високим. Гравці неохоче встановлюють нове ПЗ, якщо воно не пропонує революційного досвіду.

- *Реакція існуючих гравців:* великі корпорації (Microsoft, Valve) можуть ігнорувати стартапи, поки ті не наберуть критичної маси, після чого можуть спробувати придбати їх або заблокувати доступ до своїх API.

2. Ринкова влада постачальників (Висока).

- *Специфіка:* постачальниками для «GameHub» є платформи-донори (Steam, Epic, GOG), які надають дані про ігри.

- *Залежність:* GameHub критично залежить від відкритості цих платформ. Якщо Valve вирішить закрити публічний доступ до профілів користувачів, основна функція агрегації перестане працювати. Це вимагає від менеджменту стратегії диверсифікації джерел даних (парсинг локальних файлів замість API-запитів).

3. Ринкова влада покупців (Висока).

- *Чутливість до ціни:* оскільки базовий продукт є безкоштовним, користувачі мають нульову вартість перемикання (Switching cost). Якщо GameHub почне працювати повільно або показуватиме нав'язливу рекламу, користувач миттєво повернеться до конкурентів або видалить програму.

- *Вимогливість*: геймери – це технічно освічена аудиторія, яка вимагає ідеальної оптимізації та відсутності багів.

4. Загроза товарів-замінників (Висока).

- *Прямі замінники*: GOG Galaxy 2.0, Playnite, LaunchBox.
- *Непрямі замінники*: ярлики на робочому столі, функція «Додати сторонню гру» в Steam.
- *Майбутні замінники*: хмарні сервіси (GeForce Now), які взагалі знімають потребу в локальному управлінні бібліотекою.

5. Рівень конкурентної боротьби (Високий). Ринок насичений. Гравці змагаються не ціною, а зручністю (UX) та екосистемністю. Перемога в цій боротьбі можлива лише за рахунок надання унікального досвіду, який не можуть запропонувати конкуренти – наприклад, повної кастомізації через відкритий код.

PESTEL-аналіз макросередовища. *(Доповнення до попереднього варіанту)*
Окрему увагу слід приділити технологічному фактору, а саме тренду на *децентралізацію (Web3)*. Хоча GameHub є Web2-продуктом, зростання популярності блокчейн-ігор створює попит на агрегацію не лише традиційних ліцензій, а й NFT-активів. Це потенційна точка росту, яку слід врахувати в Roadmap.

2.2. Стратегічне позиціонування та аналіз стейкхолдерів

Для успішного управління проектом необхідно чітко ідентифікувати всі зацікавлені сторони та їхній вплив. Використано інструмент Матриця влади/інтересу (Power/Interest Grid).

1. Ключові гравці (Висока влада / Високий інтерес):
 - *core-розробники (Team)*. Від їхньої мотивації та кваліфікації залежить якість продукту;
 - *активні контриб'ютори спільноти (Plugin Creators)*. Люди, які пишуть інтеграції. Якщо вони втратять інтерес, платформа стане «мертвою»;
 - *стратегія*. Тісна співпраця, регулярні дзвінки, система винагород.

2. Суб'єкти, яких треба задовольняти (Висока влада / Низький інтерес):
 - *платформи-донори (Steam, Epic). Вони можуть заблокувати GameHub, хоча й не цікавляться його розвитком;*
 - *регулятори (GDPR). Вимагають дотримання норм приватності;*
 - *стратегія. Дотримання правил, уникнення агресивних дій (наприклад, скрапінгу серверів), юридична чистота.*
3. Суб'єкти, яких треба інформувати (Низька влада / Високий інтерес):
 - *ранні користувачі (Early Adopters). Вони створюють шум навколо проєкту, пишуть відгуки, повідомляють про баги;*
 - *техно-блогери. Впливають на громадську думку;*
 - *стратегія. Активне ведення соцмереж, публічний Roadmap, швидка реакція на фідбек.*
4. Спостерігачі (Низька влада / Низький інтерес):
 - *казуальні геймери. Потенційна масова аудиторія в майбутньому;*
 - *стратегія. Моніторинг їхніх потреб, але без активних дій на етапі MVP.*

Окремої уваги в рамках стратегічного аналізу заслуговує психологічний аспект поведінки користувачів, який в маркетингу називається «**Subscription Fatigue**» (**Втома від підписок**). Сучасний геймер стикається з парадоксом вибору: маючи доступ до сотень ігор через Game Pass, EA Play та власні бібліотеки, він витрачає більше часу на вибір гри, ніж на сам ігровий процес.

«GameHub» позиціонує себе як інструмент **цифрового детоксу та впорядкування**. Ціннісна пропозиція виходить за межі простої технічної агрегації; вона пропонує емоційну цінність – відчуття контролю над цифровим хаосом. Це дозволяє сформувати додатковий вектор конкурентної боротьби, де GameHub не намагається перевершити Steam за кількістю функцій магазину, а конкурує за "спокій користувача". Такий підхід дозволяє залучити сегмент дорослих платоспроможних геймерів, які цінують свій час вище, ніж вартість

підписки на сервіс, що безпосередньо впливає на розрахунок LTV (пожиттєвої цінності клієнта) у бізнес-моделі.

Стратегія «Блакитного океану» (Blue Ocean Strategy). Більшість конкурентів (GOG, Steam) грають у «червоному океані», намагаючись продати ігри. GameHub створює «блакитний океан» – простір, де продаж ігор не є метою.

- *усунути*: необхідність купувати ігри всередині агрегатора (магазин як основна функція);
- *зменшити*: споживання системних ресурсів (порівняно з «важкими» клієнтами на electron);
- *підсилити*: роль спільноти у розробці функціоналу;
- *створити*: єдиний простір для керування *підписками*, а не лише купленими іграми (актуально в епоху game pass).

Таке позиціонування дозволяє GameHub не конкурувати з гігантами (Steam) напряму, а доповнювати їх, стаючи надбудовою над існуючим ринком.

2.3. Побудова бізнес-моделі

В умовах високої невизначеності, яка є характерною для технологічних стартапів на ранніх стадіях, використання класичного об'ємного бізнес-планування є малоефективним. Ринкова ситуація змінюється швидше, ніж реалізується довгострокова стратегія. Тому для моделювання та верифікації бізнес-логіки проекту «GameHub» обрано інструмент Lean Canvas. Цей підхід дозволяє на одній сторінці відобразити ключові гіпотези бізнесу та сфокусуватися на проблемах користувачів [4].

Розроблена бізнес-модель проекту базується на дев'яти ключових блоках, детальна характеристика яких наведена нижче:

1. Проблема (Problem): Головним болем цільової аудиторії є «хаос у бібліотеці» – геймери забувають, на якій платформі придбали гру, що призводить до втрати часу. Другою проблемою є ресурсна неефективність: фонові лаунчері декількох лаунчерів споживає до 1.5 ГБ оперативної пам'яті, знижуючи продуктивність ігор.

2. Рішення (Solution): Проєкт пропонує створення єдиного дашборду з автоматичним імпортом усіх ігор, а також функцію «Smart Kill Switch» для автоматичного завершення зайвих процесів.

3. Унікальна ціннісна пропозиція (UVP): «GameHub – твій єдиний командний центр. Всі ігри, всі друзі, максимальна швидкодія в одному додатку».

4. Нечесна перевага (Unfair Advantage): На відміну від корпоративних рішень, GameHub використовує архітектуру *Community-First*. Відкритий API дозволяє спільноті створювати та підтримувати плагіни швидше, ніж це роблять штатні команди конкурентів.

5. Сегменти клієнтів (Customer Segments): Основним сегментом є «Early Adopters» – технічно грамотні гравці з бібліотекою понад 100 ігор у різних магазинах, які критично ставляться до оптимізації ПК.

6. Потоки доходів (Revenue Streams): Обрано модель Freemium. Базовий функціонал є безкоштовним, тоді як монетизація відбувається за рахунок Premium-підписки (хмарна синхронізація, теми) та партнерських відрахувань (CPA) від магазинів ігор.

7. Структура витрат (Cost Structure): Основні витрати спрямовані на R&D (зарплати Core-команди) та оплату хмарної інфраструктури для API.

8. Ключові метрики (Key Metrics): Успішність проєкту оцінюється за кількістю активних користувачів (MAU) та стабільністю роботи інтеграцій.

9. Канали (Channels): Просування здійснюється через SEO-оптимізацію, співпрацю з техно-блогерами та спільноти Reddit/Discord.

Візуалізація розробленої бізнес-моделі представлена на рис. 2.1 нижче.

Аналіз даної моделі підтверджує, що обрана стратегія дозволяє створити стійкий продукт з чіткою цінністю для користувача, мінімізуючи ризики розробки непотрібного функціоналу.

<p>PROBLEM</p> <ul style="list-style-type: none"> Хаос у бібліотеці (не пам'ятають де ігри); Ресурсна неефективність (багато лаунчерів, гальмує ПК); Соціальна ізоляція (друзі в різних екосистемах). <p>EXISTING ALTERNATIVES</p> <ul style="list-style-type: none"> Steam, Epic Games, GOG Galaxy 2.0, Playnite. 	<p>SOLUTION</p> <ul style="list-style-type: none"> Єдиний інтерфейс (Universal Dashboard); Smart Kill Switch (оптимізація процесів); Агрегатор чатів та соц. стрічок. 	<p>UNIQUE VALUE PROPOSITION</p> <ul style="list-style-type: none"> GameHub – твій єдиний командний центр. Всі ігри, всі друзі, максимальна швидкодія. Платформа, яка адаптується до змін швидше за всіх. 	<p>UNFAIR ADVANTAGE</p> <ul style="list-style-type: none"> Відкрита архітектура плагінів; Community-Driven (спільнота лагодить інтеграції швидше за корпорації). 	<p>CUSTOMER SEGMENTS</p> <ul style="list-style-type: none"> Хардкорні геймери (PC Master Race, 4+ магазини); Користувачі підписок (Game Pass, EA Play); Bargain Hunters (шукачі знижок).
<p>KEY METRICS</p> <ul style="list-style-type: none"> MAU (Monthly Active Users); Retention Rate; Integration Stability Score. 	<p>HIGH-LEVEL CONCEPT</p> <ul style="list-style-type: none"> Операційна система для геймінгу, що руйнує бар'єри між магазинами. 	<p>CHANNELS</p> <ul style="list-style-type: none"> Inbound Marketing (SEO, блог); Influencer Marketing (YouTube, Twitch); Communities (Reddit, Discord). 	<p>COST STRUCTURE</p> <ul style="list-style-type: none"> R&D (зарплати команди); Хмарна інфраструктура (сервери); Маркетинг (CAC). <p>REVENUE STREAMS</p> <ul style="list-style-type: none"> Freemium (база безкоштовна); Premium підписка (\$3/міс, хмара, теми); Affiliate CPA (комісія за переходи). 	

*Рисунок 2.1 – Lean Canvas
Джерело: розроблено автором.*

2.4. Формування візії продукту (Product Vision) та стратегії розвитку

Наявність чітко сформульованої візії є критичною умовою для синхронізації Agile-команди. Для проєкту «GameHub» візію сформовано за шаблоном Джеффри Мура:

Для активних ПК-геймерів та користувачів підписних сервісів, які страждають від фрагментації бібліотек та зниження продуктивності ПК, продукт «GameHub» є універсальною операційною системою ігрового простору. Що забезпечує миттєвий доступ до всіх цифрових активів з єдиного інтерфейсу. На відміну від закритих корпоративних аналогів (GOG Galaxy), наш продукт пропонує повністю відкриту архітектуру (Open Source) та модульну систему плагінів, що підтримується спільнотою.

Ця стратегія реалізується через Дорожню карту (Roadmap) (рис 2.2), розбиту на 4 квартали:

- Q1 (MVP): Реліз ядра на Rust, інтеграція Steam/Еріс, локальна БД;
- Q2 (Optimization): Функція "Smart Kill Switch", темна тема, оптимізація пам'яті;
- Q3 (Community): Запуск магазину плагінів, відкрите SDK для розробників;

- Q4 (Growth): Соціальний хаб, агрегатор знижок, система досягнень.

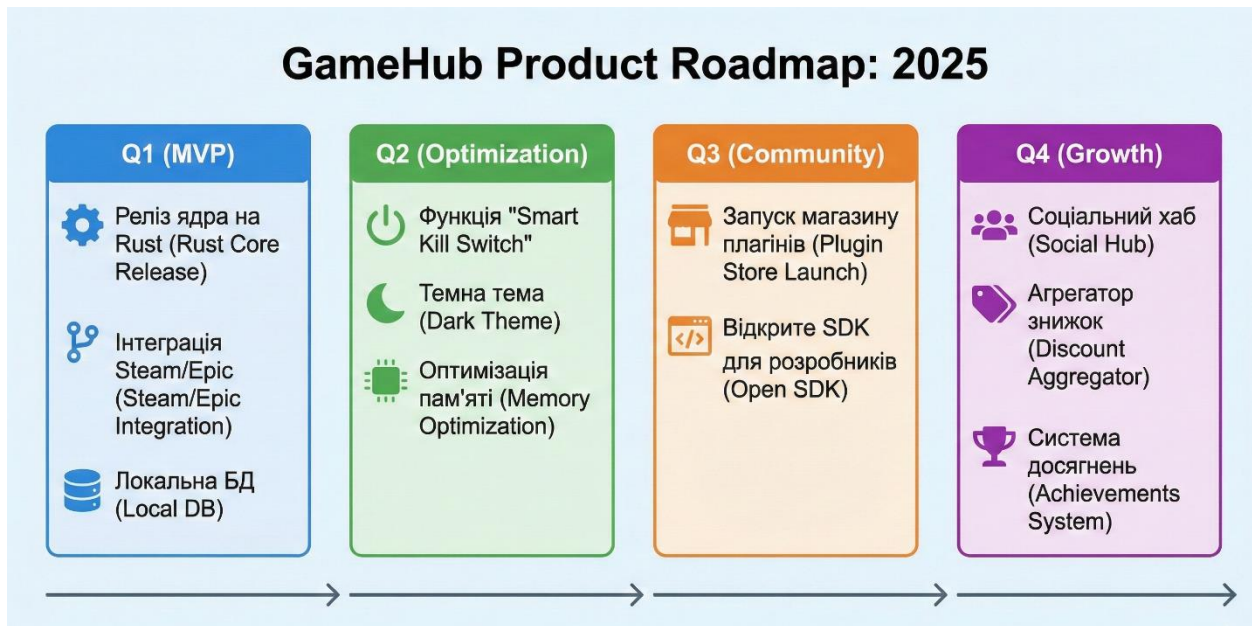


Рисунок 2.2 – Стратегічна дорожня карта (Product Roadmap) розвитку GameHub

Джерело: розроблено автором.

2.4. Моделювання шляху користувача (Customer Journey Map)

В умовах клієнтоцентричного підходу, який є основою Agile-філософії, розуміння потреб користувача не обмежується лише статичним портретом (Persona). Критично важливим є аналіз динаміки взаємодії клієнта з продуктом у часі. Для цього в роботі розроблено Карту подорожі користувача (Customer Journey Map – CJM). Цей інструмент дозволяє візуалізувати бар'єри, з якими стикається геймер, та ідентифікувати точки контакту (Touchpoints), де «GameHub» може створити максимальну цінність.

CJM для типового користувача (персона «Олексій») складається з п'яти етапів:

Етап 1: Усвідомлення потреби (Awareness)

- *Сценарій:* Олексій хоче пограти в «Відьмака 3», але не пам'ятає, чи купував він гру в Steam, чи отримав безкоштовно в GOG.
- *Дії:* Перевіряє ярлики на робочому столі, запускає по черзі Steam, Epic, GOG.

- *Емоційний стан:* Роздратування, втрата часу.
- *Бар'єри:* Необхідність оновлення кожного лаунчера перед запуском.
- *Можливість для GameHub:* Позиціонування через контент-маркетинг (статті «Як знайти всі свої ігри») як рішення проблеми «цифрової амнезії».

Етап 2: Пошук рішення та встановлення (Consideration & Acquisition)

- *Сценарій:* Олексій шукає в Google «об'єднати бібліотеки ігор». Знаходить GameHub.
- *Дії:* Перегляд лендінгу, завантаження інсталлятора.
- *Ключові очікування:* Програма має бути «легкою» (не Electron, що гальмує систему) та безпечною (Open Source).
- *Момент істини (Moment of Truth):* Перший запуск. Якщо імпорт бібліотек не спрацює за 30 секунд, користувач видалить програму.
- *Agile-рішення:* Включення задачі «Оптимізація першого запуску (Onboarding)» у пріоритетний беклог першого спринту.

Етап 3: Активне використання (Service / Usage)

- *Сценарій:* Щоденний запуск ігор через GameHub.
- *Дії:* Натискання кнопки «Грати». Автоматичне згортання GameHub у трей (Ghost Mode).
- *Емоційний стан:* Задоволення від комфорту та швидкодії.
- *Бар'єри:* Збій інтеграції (наприклад, Steam змінив API, і бібліотека не оновлюється).

- *Можливість для GameHub:* Система автоматичних сповіщень про статус плагінів та можливість оновити їх в один клік силами спільноти.

Етап 4: Лояльність та Розширення (Loyalty & Expansion)

- *Сценарій:* Олексій звик до програми і хоче більшого.
- *Дії:* Налаштування темної теми, додавання емуляторів консолей, спілкування з друзями через агрегатор чатів.
- *Можливість для GameHub:* Пропозиція Premium-підписки для хмарної синхронізації налаштувань між ноутбуком та ПК.

Етап 5: Адвокація (Advocacy)

- *Сценарій:* Олексій рекомендує GameHub друзям.
- *Дії:* Написання відгуку на Reddit, поширення реферального посилання.
- *Agile-рішення:* Впровадження механік віральності (наприклад, бейджі «Early Adopter» у профілі).

Розробка CJM дозволила виявити критичні точки, які потребують першочергової уваги команди розробки. Зокрема, етап «Онбордингу» (автоматичний імпорт ігор) є вирішальним для утримання користувача (Retention), тому відповідні User Stories отримали найвищий пріоритет у Product Backlog¹.

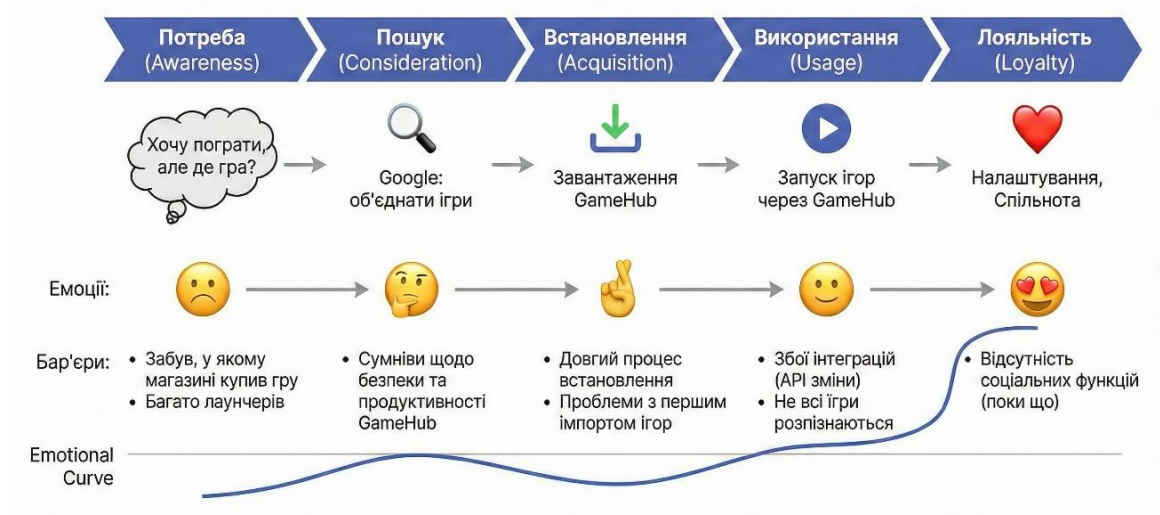


Рисунок 2.3 – Customer Journey Map

Джерело: розроблено автором.

2.5. Економічне обґрунтування та Unit-економіка проєкту

Навіть в умовах Agile-розробки, де фокус зміщено на цінність продукту, розуміння економічної моделі є необхідним для забезпечення життєздатності проєкту (Sustainability). Для оцінки фінансової ефективності «GameHub» розраховано базову Unit-економіку (економіку одного юніта/клієнта).

Оскільки GameHub використовує модель Freemium, розрахунок базується на конверсії безкоштовних користувачів у платних підписників.

Основні параметри моделі:

1. Юніт (Unit): Один активний користувач (MAU).
2. ARPU (Average Revenue Per User): Середній дохід з одного користувача.
3. CAC (Customer Acquisition Cost): Вартість залучення одного клієнта.
4. LTV (Lifetime Value): Пожиттєва цінність клієнта.

Розрахунок CAC (Вартість залучення):

Припустимо, що маркетинговий бюджет на тестовий місяць становить 1000 USD.

- Канали: Таргетована реклама, робота з інфлюенсерами.
- Отриманий трафік: 5000 відвідувачів сайту.
- Конверсія в установку (Install Rate): 20% (1000 нових користувачів).
- $CAC = 1000 \text{ USD} / 1000 \text{ користувачів} = 1 \text{ USD}$.

Це є прийнятним показником для десктопного ПЗ у ніші Utility.

Розрахунок LTV (Пожиттєва цінність):

Дохід формується з двох потоків:

- *Підписка (Subscription)*: 3 USD/міс. Конверсія в підписку (Conversion Rate) – 5%.
- *Партнерські програми (Affiliate)*: Середній дохід з комісії за купівлю ігор через агрегатор – 0.5 USD/міс на активного користувача.

Середній дохід на місяць з одного користувача (ARPU):

$$ARPU = (\text{Ціна_підписки} \times \text{Конверсія}) + \text{Дохід_Affiliate}$$

$$ARPU = (3 \times 0.05) + 0.05 = 0.15 + 0.05 = 0.20$$

Середній час життя користувача в сервісі (Lifetime) оцінюється у 12 місяців (завдяки високому Retention Rate, забезпеченому якістю Agile-розробки).

$$LTV = ARPU \times Lifetime = 0.20 \times 12 = 2.40$$

Оцінка ефективності:

$$\text{Співвідношення } LTV/CAC = 2.40 / 1.00 = 2.4$$

Згідно з бенчмарками IT-індустрії, успішною вважається бізнес-модель, де $LTV/CAC > 3$. Отриманий результат (2.4) свідчить про те, що на етапі MVP проєкт є перспективним, але потребує роботи над підвищенням конверсії в підписку або зниженням вартості залучення.

Окремим, надзвичайно важливим аспектом управління якістю в довгострокових Agile-проєктах є стратегія управління технічним боргом (Technical Debt Management). У гонитві за швидким випуском функцій (Time-to-Market) команди часто вдаються до "швидких та брудних" рішень у кодї. Для проєкту «GameHub», який базується на інтеграціях зі сторонніми API, технічний борг може накопичуватися дуже швидко у вигляді "костилів" (workarounds) для обходу захисту платформ.

Для системного управління цим процесом у GameHub впроваджуються наступні практики:

1. Правило 20%: У кожному спринті 20% ємності команди (Capacity) резервується виключно на задачі з рефакторингу та оптимізації коду, які не несуть прямої нової функціональності для користувача, але забезпечують стабільність системи.

2. Tech Debt Backlog: Технічні завдання не губляться, а заносяться в окремий розділ беклогу в Jira з міткою `tech_debt`. Власник продукту (PO) спільно з техлідом пріоритезують їх на рівні з бізнес-фічами.

3. Архітектурні рев'ю: Раз на квартал проводиться сесія перегляду архітектури (Architecture Review Board), щоб переконатися, що поточні рішення не блокують майбутнє масштабування продукту (наприклад, перехід на мікросервісну архітектуру при збільшенні кількості користувачів до 100 000).

Також слід деталізувати процес проведення ретроспектив. Щоб уникнути формалізму, Scrum Master використовує різні техніки фасилітації. Наприклад, метод «Starfish» (Морська зірка), де команда аналізує процеси за п'ятьма категоріями: *Keep Doing* (продовжувати робити те, що добре), *Less of* (робити менше), *More of* (робити більше), *Stop Doing* (припинити, бо це шкодить), *Start Doing* (спробувати нове). Це дозволяє не просто "поговорити про проблеми", а

сформувати конкретний план дій (Action Items) для покращення процесів у наступному спринті.

Стратегія покращення економіки через Agile:

Гнучке управління дозволяє ітеративно покращувати ці показники. Наприклад, впровадження фічі «Персоналізовані знижки» (заплановано на Q3 Roadmap) підвищить Affiliate-дохід, а оптимізація онбордингу знизить відтік користувачів (Churn Rate), збільшуючи Lifetime.

2.6. Організаційно-правове забезпечення проєкту

Успіх проєкту залежить не лише від коду, а й від юридичної чистоти. Для «GameHub» розроблено стратегію захисту інтелектуальної власності та дотримання законодавства.

1. Ліцензування коду: Оскільки проєкт є Open Source, обрано ліцензію MIT License. Вона дозволяє спільноті вільно використовувати та модифікувати код, що стимулює розвиток екосистеми плагінів, але водночас захищає авторів від відповідальності.

2. Захист даних (Privacy Policy): Враховуючи вимоги GDPR, архітектура GameHub спроектована за принципом «Local First». Дані про бібліотеку ігор, час гри та друзів зберігаються виключно на локальному диску користувача у зашифрованому вигляді (SQLite). Синхронізація з хмарою відбувається лише за явним бажанням користувача (Opt-in) та з використанням наскрізного шифрування (End-to-End Encryption).

3. Взаємодія з платформами: Щоб уникнути блокування з боку Steam чи Epic Games, GameHub не використовує методи скрапінгу (Web Scraping), які навантажують сервери партнерів. Натомість використовуються офіційні API та локальне читання маніфестів встановлених ігор, що не порушує User Agreement сторонніх сервісів.



Рисунок 2.4 – Product Road Map

Джерело: розроблено автором.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи здійснено комплексне проектування бізнесу та середовища для платформи «GameHub».

1. Ринковий аналіз за методиками PESTEL та 5 сил Портера підтвердив наявність вільної ніші для незалежного агрегатора ігор, здатного до швидкої технічної адаптації.

2. Моделювання досвіду користувача (CJM) виявило, що ключовим фактором успіху є безшовна інтеграція бібліотек на етапі першого запуску. Це стало основою для пріоритизації технічних завдань.

3. Розрахунок Unit-економіки довів потенційну прибутковість проекту ($LTV/CAC = 2.4$) навіть за консервативного сценарію.

4. Сформована бізнес-модель (Lean Canvas) та дорожня карта (Roadmap) створюють надійний фундамент для переходу до етапу практичної реалізації, який буде розглянуто в наступному розділі. Застосування інструментів стратегічного менеджменту в поєднанні з Agile-філософією дозволило мінімізувати ризики «непотрібного продукту» ще до початку активної фази розробки.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ «GAMEHUB» ЗА МЕТОДОЛОГІЄЮ AGILE

3.1. Формування Product Backlog та планування ітерацій

Практична реалізація проєкту починається з трансформації стратегічного бачення (Product Vision), розробленого у попередньому розділі, у тактичний план дій. Згідно з обраним Agile-підходом, ключовим артефактом, що описує вимоги до продукту, є Product Backlog (Беклог продукту)¹.

Процес наповнення беклогу здійснювався методом декомпозиції. Функціональність «GameHub» була розділена на ієрархічні рівні:

1. Features (Фічі). Великі частини продукту, що несуть значну бізнес-цінність (наприклад, «Інтеграція магазинів», «Інтерфейс користувача»)².
2. Epics (Епіки). Менші частини Features, які потребують більше одного спринту для реалізації³.
3. User Stories (Історії користувачів). Опис конкретних функцій з точки зору кінцевого користувача, які можуть бути реалізовані за один спринт⁴.

Для кожної User Story було визначено критерії прийняття (Acceptance Criteria), що є необхідною умовою для фіксації вимог якості. Оцінка трудомісткості (Estimation) проводилася командою в умовних одиницях – Story Points, з використанням послідовності Фібоначчі, що дозволяє врахувати невизначеність та складність завдань.

Нижче (табл 3.1) наведено фрагмент розробленого та пріоритезованого Product Backlog (рис 3.1) для етапу MVP (Minimum Viable Product).

Таблиця 3.1 – Фрагмент Product Backlog проєкту «GameHub»

ID	Тип	Елемент беклогу (User Story)	Критерії прийняття (Acceptance Criteria)	Пріоритет	Оцінка (SP)
GH-1	Епік	Система авторизації та профілів		High	-

GH-1.1	Story	Як користувач, я хочу створити локальний профіль без відправки даних у хмару, щоб зберегти конфіденційність.	1. Дані профілю зберігаються у файлі JSON локально. 2. Додаток працює без інтернет-з'єднання.	High	3
GH-1.2	Story	Як геймер, я хочу підключити акаунт Steam через OAuth, щоб не вводити пароль у самій програмі.	1. Відкривається офіційне вікно входу Steam. 2. Отримано токен доступу. 3. Пароль не зберігається в GameHub.	High	5
GH-2	Epic	Агрегація бібліотек (Core)		High	-
GH-2.1	Story	Як користувач, я хочу, щоб GameHub автоматично знайшов встановлені ігри Steam та Epic Games.	1. Сканування стандартних директорій бібліотек. 2. Відображення знайдених ігор у списку «Встановлені».	High	8
GH-2.2	Story	Як користувач, я хочу запускати гру кліком на її обкладинку в GameHub.	1. Запуск нативного лаунчера у «тихому» режимі. 2. Запуск виконуваного файлу гри (.exe).	High	5
GH-3	Epic	Оптимізація (Performance)		Medium	-
GH-3.1	Story	Як власник слабкого ПК, я хочу, щоб GameHub автоматично закривав сторонні лаунчери після виходу з гри.	1. Моніторинг процесу гри. 2. При закритті гри – примусове завершення процесу лаунчера.	Medium	8

На основі загальної оцінки беклогу MVP (сумарно 120 Story Points) та розрахункової швидкості команди (Velocity) у 20 SP за спринт, було складено план релізів. Розробка ведеться ітераціями по 2 тижні.

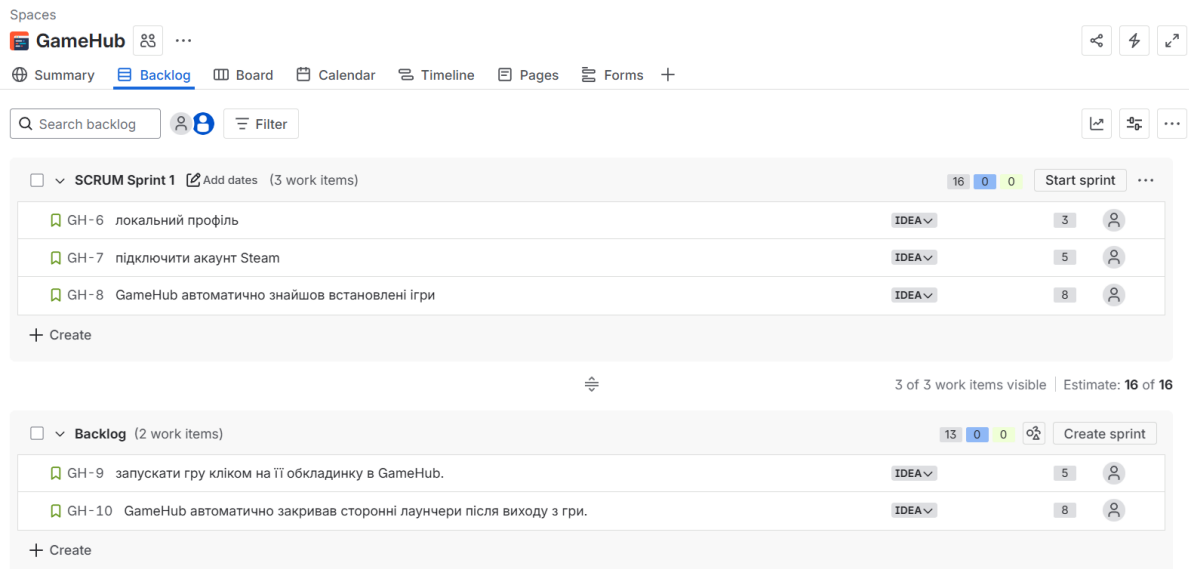


Рисунок 3.1 – Інтерфейс сформованого Product Backlog проекту «GameHub» в Jira

Джерело: розроблено автором на базі Jira

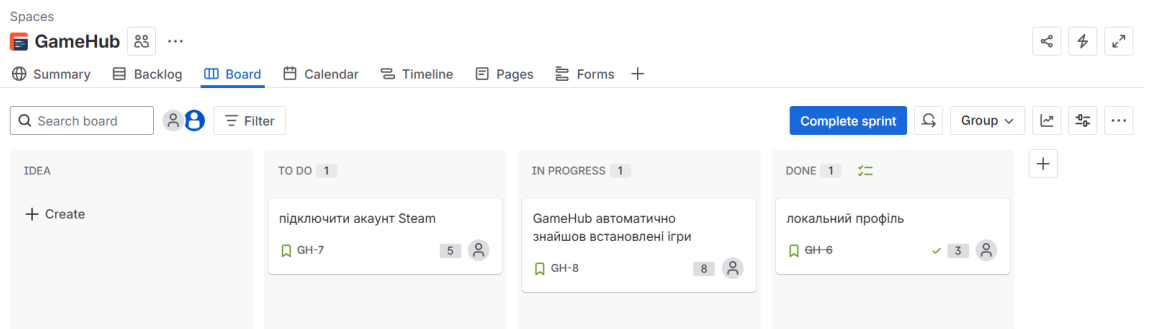


Рисунок 3.2 – Візуалізація потоку виконання завдань (Scrum Board) в активному спринті

Джерело: розроблено автором на базі Jira.

Спринт 0 (Підготовчий): Налаштування середовища розробки, репозиторіїв GitLab, CI/CD пайплайнів та базової архітектури на Rust.

Спринт 1 (Авторизація): Реалізація завдань GH-1.1 та GH-1.2.

Спринт 2 (Імпорт): Реалізація ядра агрегації (GH-2.1).

3.2. Організаційна структура команди та ресурсне забезпечення

Для реалізації проєкту сформовано команду згідно з рекомендаціями Scrum Guide та методичними вказівками щодо розподілу ролей [3,16]. Команда є крос-функціональною, тобто володіє всіма компетенціями, необхідними для створення готового інкременту продукту без залучення зовнішніх ресурсів.

Склад команди та розподіл відповідальності:

1. Product Owner (Власник продукту): Відповідає за формування візії, управління Product Backlog, пріоритезацію завдань та прийняття роботи. У контексті GameHub він також виконує роль комунікатора зі спільнотою⁸.

2. Scrum Master: Забезпечує дотримання процесів Scrum, фасилітує події (дейлі, ретроспективи) та усуває перешкоди, що заважають команді працювати⁹.

3. Developers (Команда розробки):

- *Senior Backend Engineer*: відповідає за архітектуру ядра, безпеку даних та роботу з файловими системами¹

- *Middle Frontend Engineer*: реалізує інтерфейс користувача, анімації та взаємодію¹

- *QA Automation Engineer*: забезпечує якість через написання автотестів для перевірки інтеграцій, що є критичним для стабільності агрегатора¹

- *UX/UI Designer*: проєктує зручні сценарії взаємодії та візуальний стиль¹

Враховуючи обрану стратегію Community-Driven Development (розробка, керована спільнотою), організаційна структура проєкту має специфічні особливості, які відрізняють її від класичної аутсорсинг-команди. Роль Product Owner (PO) у «GameHub» трансформується у роль Technical Evangelist.

Його обов'язки розширюються за рахунок наступних функцій:

1. Developer Relations (DevRel): Активна робота з авторами сторонніх плагінів на GitHub. Це включає рев'ю їхнього коду (Pull Requests), надання технічної підтримки та модерацію екосистеми.

2. Управління "Bounties": Розподіл мікро-грантів або бонусів для розробників, які найшвидше виправляють критичні помилки інтеграції (наприклад, при зміні API Epic Games).

Це змінює і вимоги до Senior Backend Engineer. Окрім знання Rust, цей спеціаліст повинен вміти писати якісну публічну документацію (SDK Documentation), оскільки без зрозумілого гайду для зовнішніх розробників ідея "відкритої архітектури плагінів" залишиться нереалізованою. Таким чином, управління людськими ресурсами в проєкті фокусується не лише на наймі штату, а й на вирощуванні лояльної спільноти зовнішніх контриб'юторів.

Технічне та програмне забезпечення:

Для управління проєктом використовується екосистема Atlassian:

- Jira Software. Для ведення беклогу, планування спринтів та трекінгу задач;
- Confluence. Для ведення технічної документації та бази знань проєкту;
- GitLab. Для контролю версій коду та налаштування CI/CD.

3.3. Розрахунок бюджету проєкту (Кошторис)

Фінансове планування проєкту здійснюється за моделлю «Time & Material» з фіксованою місячною вартістю команди. Розрахунок витрат базується на ринкових ставках фахівців та тривалості етапу розробки MVP (3 місяці).

Витрати на оплату праці розраховано відповідно до методичних рекомендацій [9].

Таблиця 3.2 – Кошторис витрат на розробку MVP (3 місяці)

Посада / Роль	Місячна ставка (USD)	Кількість	Витрати на місяць (USD)	Загальні витрати (3 міс.)
Senior Backend Engineer	6 000	1	6 000	18 000
Middle Frontend Engineer	3 500	1	3 500	10 500
QA Automation Engineer	3 000	1	3 000	9 000
UX/UI Designer (Part-time)	3 500 (0.5)	1	1 750	5 250
Scrum Master (Part-time)	4 500 (0.5)	1	2 250	6 750
Разом ФОП		5	16 500	49 500

Операційні витрати				
Ліцензії ПЗ та сервери	500	-	500	1 500
Маркетинг (запуск)	1 000	-	1 000	3 000
ВСЬОГО			18 000	54 000

З урахуванням резервного фонду на непередбачувані витрати у розмірі 10% (5 400 USD), загальний бюджет проєкту становить **59 400 USD**. Такий обсяг інвестицій є обґрунтованим для створення технологічного продукту з високим потенціалом масштабування та монетизації.

3.4. Аналіз метрик ефективності та звітність у Jira

Впровадження Agile-підходів вимагає переходу від суб'єктивних оцінок до управління на основі даних (Data-Driven Management). Середовище Jira Software автоматично генерує звіти, які дозволяють Scrum-майстру та команді аналізувати "здоров'я" процесу розробки. Для проєкту «GameHub» визначено критичні метрики, моніторинг яких здійснюється наприкінці кожного спринту.

1. Деталізація вимог (User Stories) Основою для збору метрик є якісно оформлена картка задачі. Кожна User Story в GameHub містить чіткі критерії прийняття (Acceptance Criteria) та оцінку складності в Story Points (рис 3.4).

2. Velocity Chart (Діаграма швидкості) Ця метрика відображає продуктивність команди від спринту до спринту. Вона показує співвідношення між запланованим обсягом роботи (Commitment – сірий стовпчик) та фактично виконаним (Completed – зелений стовпчик). Аналіз Velocity дозволяє:

- прогнозувати дату релізу MVP. Якщо середня швидкість команди складає 20 SP, а беклог містить 100 SP, то для завершення знадобиться 5 спринтів;
- виявляти проблеми планування. Якщо стовпчик "Completed" постійно значно нижчий за "Commitment", це свідчить про те, що команда бере на себе забагато зобов'язань (Overcommitment) або задачі оцінені неправильно.

☑ Jira work item 🔗 ✕

📄 Add epic / 📌 GH-9 🔒 👁 1 🔗 ⋮

запускати гру кліком на її обкладинку в GameHub.

Idea ▾ ⚡ ✨ ⚙ +

Story point estimate 5

Description

Запускати гру кліком на її обкладинку в GameHub

Subtasks

Add subtask

Linked work items

Add linked work item

Рисунок 3.4 – Приклад оформлення User Story в системі Jira
Джерело: розроблено автором на базі Jira.

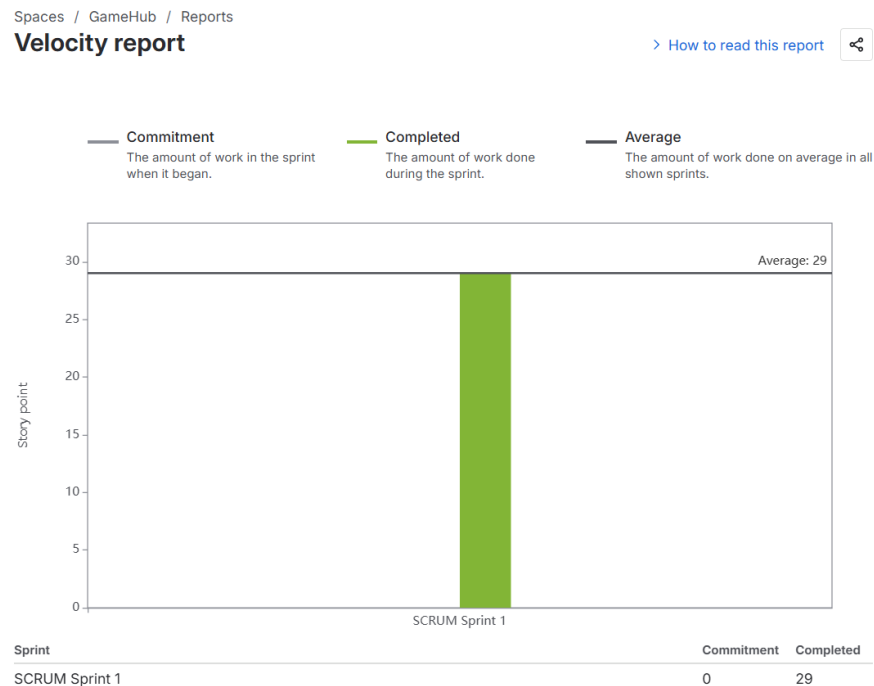


Рисунок 3.5 – Діаграма швидкості команди (Velocity Chart)
Джерело: розроблено автором на базі Jira.

3. Sprint Report (Звіт по спринту) Цей звіт аналізується на ретроспективі. Він показує список завершених і незавершених задач, а також "роздування обсягу" (Score Creep) – задачі, які були додані в спринт вже після його початку. Для GameHub, де інтеграції можуть "ламатися" раптово, контроль Score Creep є критичним, щоб не зірвати основну ціль спринту.



Рисунок 3.6 – Звіт про результати спринту (Sprint Report)

Джерело: розроблено автором на базі Jira.

4. Cumulative Flow Diagram (CFD - Накопичувальна діаграма потоку) CFD є основним інструментом для аналізу стабільності потоку в Kanban-частині проекту (підтримка інтеграцій). Діаграма показує розподіл задач за статусами (To Do, In Progress, Testing, Done) у часі. Ключові сигнали CFD для менеджера:

- якщо смуга "Testing" (Тестування) різко розширюється, це свідчить про "пляшкове горлечко" (bottleneck): QA-інженер не встигає перевіряти задачі, які постачають розробники;
- рівномірне зростання зони "Done" свідчить про стабільний процес поставки цінності.

Spaces / GameHub / Reports

Cumulative flow diagram[> How to read this report](#)

Date filter

All Time

From date

2/18/1993

To date

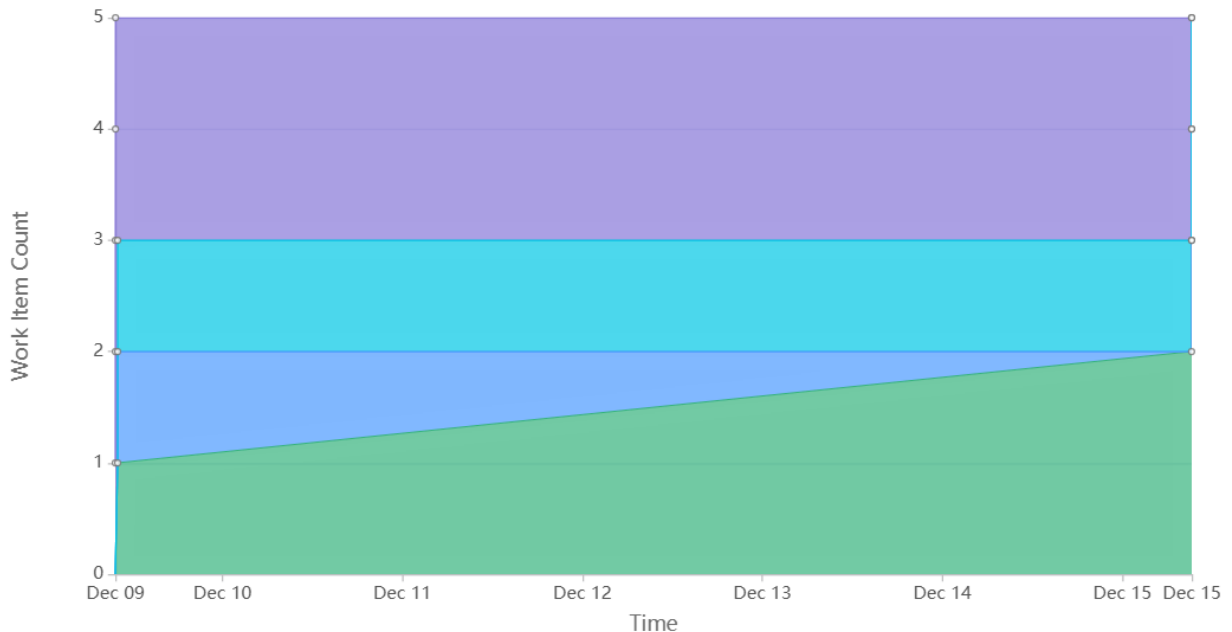
2/18/1993
 Idea
 To Do
 In Progress
 Done


Рисунок 3.7 – Накопичувальна діаграма потоку (CFD)

Джерело: розроблено автором на базі Jira.

Таким чином, комплексне використання обраних інструментів звітності в середовищі Jira забезпечує повну прозорість (Transparency) процесу розробки платформи «GameHub». Поєднання метрик прогнозування (Velocity Chart) з інструментами аналізу потоку (Cumulative Flow Diagram) дозволяє реалізувати підхід Data-Driven Management. Це дає змогу менеджеру проєкту відмовитися від

інтуїтивного планування на користь обґрунтованих прогнозів, своєчасно виявляти «вузькі місця» в процесах тестування та інтеграції, а також аргументовано комунікувати стан проекту стейкхолдерам, спираючись на об'єктивні історичні дані, а не на суб'єктивні очікування.

3.5. Моніторинг ефективності, контроль якості та CI/CD процеси

Впровадження гнучких методологій вимагає налаштування прозорої системи моніторингу прогресу. На відміну від традиційного проектного менеджменту, де контролюються відхилення від жорсткого плану, у проєкті «GameHub» фокус зміщено на емпіричні метрики продуктивності команди та якості продукту.

Основним середовищем для управління задачами обрано Atlassian Jira, що відповідає сучасним стандартам індустрії. Для візуалізації потоку робіт налаштовано Scrum Board з колонками, що відповідають життєвому циклу задачі (Workflow):

1. *To Do* (Задачі, відібрані у спринт).
2. *In Progress* (В роботі).
3. *Code Review* (Перевірка коду іншим розробником).
4. *QA / Testing* (Тестування).
5. *Done* (Готово до релізу).

Для оцінки ефективності роботи команди використовуються ключові Agile-метрики, візуалізація яких доступна в Jira¹:

- **Velocity Chart** (Діаграма швидкості): Дозволяє прогнозувати, скільки Story Points команда здатна виконати в наступних спринтах, базуючись на історичних даних². Це критично важливо для планування дати релізу MVP;
- **Burndown Chart** (Діаграма згоряння): Щоденний графік, що показує залишок роботи в межах поточного спринту [6]. Дозволяє Scrum-майстру вчасно виявити ризик невиконання цілі спринту³;
- **Cumulative Flow Diagram**: Допомогає виявити «вузькі місця» (bottlenecks), наприклад, якщо задачі накопичуються на етапі тестування⁴.

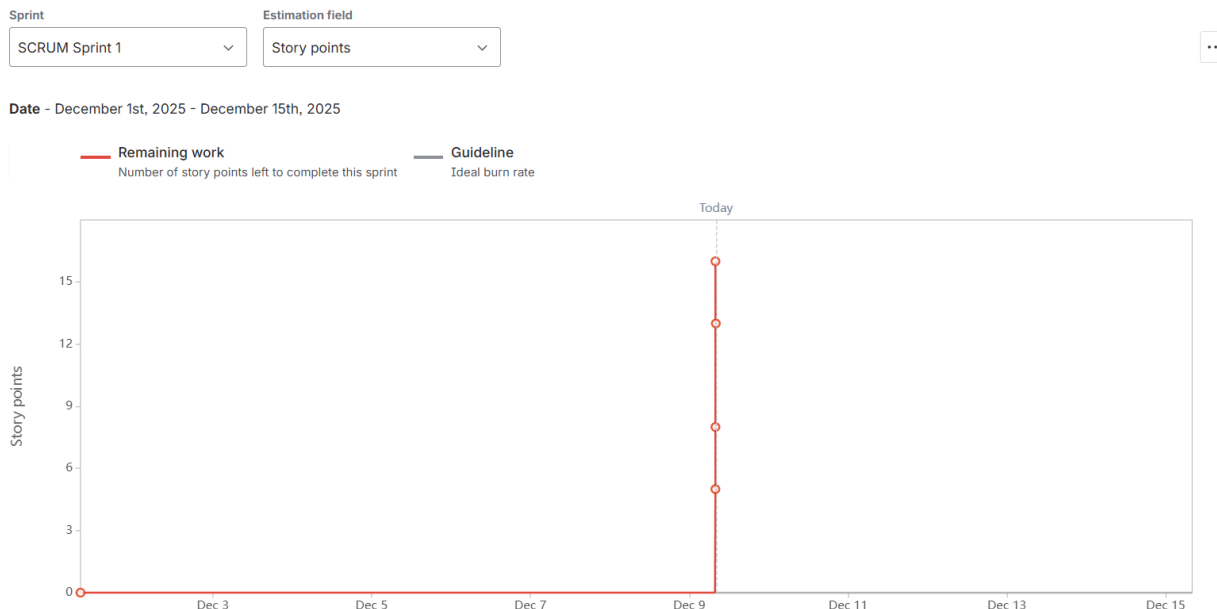


Рисунок 3.3 – Діаграма згоряння робіт (Burndown Chart) для контролю швидкості команди

Джерело: розроблено автором на базі Jira.

Специфіка агрегатора ігор вимагає бездоганної стабільності, оскільки будь-який збій при запуску гри викликає негативну реакцію користувача. Стратегія забезпечення якості базується на дотриманні критеріїв готовності (Definition of Done - DoD). Згідно з розробленими правилами, задача не може отримати статус «Done», якщо:

- код пройшов перехресну перевірку (Code Review);
- написані Unit-тести (покриття коду > 80%);
- функціонал перевірено на тестовому стенді (Windows 10/11);
- відсутні критичні баги.

Для перевірки інтеграцій зі сторонніми сервісами (Steam, Epic) впроваджено автоматизоване тестування. Це дозволяє нівелювати ризики регресії, коли нове оновлення ламає старий функціонал.

CI/CD (Continuous Integration / Continuous Delivery).

Для забезпечення швидкої доставки оновлень налаштовано пайплайн CI/CD. Це дозволяє автоматизувати процеси збірки (Build), тестування (Test) та розгортання (Deploy). Завдяки цьому команда може випускати хотфікси

(термінові виправлення) для плагінів інтеграції за лічені години, що є ключовою конкурентною перевагою над корпоративними аналогами.

Особливу увагу в системі забезпечення якості слід приділити методології "Canary Releases" (Канаркові релізи), яка є критично важливою для десктопних додатків з великою інсталяційною базою. Оскільки GameHub працює безпосередньо з файловою системою користувача та процесами ОС, помилка в оновленні може призвести до критичних наслідків (наприклад, помилкового видалення файлів гри).

Процес релізу в GameHub відбувається поетапно:

1. Alpha Ring (Внутрішнє коло): Автоматична збірка для команди розробників та QA.

2. Beta Ring (Спільнота): Доступ для 5-10% користувачів, які добровільно погодилися тестувати нові функції (Early Adopters). На цьому етапі збирається телеметрія про збої та помилки інтеграції на реальних конфігураціях "заліза".

3. Stable Ring (Масовий реліз): Якщо протягом 48 годин у Beta-каналі не виявлено критичних блокерів (Showstoppers), оновлення стає доступним для 100% аудиторії.

Такий підхід дозволяє мінімізувати репутаційні ризики та вартість виправлення помилок, що повністю відповідає принципам Agile щодо раннього виявлення дефектів.

3.6. Управління ризиками проєкту

Розробка програмного забезпечення, що залежить від сторонніх екосистем, пов'язана з високими ризиками. Управління ризиками в проєкті «GameHub» здійснюється ітеративно: реєстр ризиків переглядається на кожному плануванні спринту.

Таблиця 3.3 – Реєстр ризиків проєкту «GameHub»

№	Опис ризику	Ймовірність (1-5)	Вплив (1-5)	Ранг	Стратегія реагування (Mitigation Strategy)
1	Зміна API сторонніх платформ. Valve або Epic	5	5	25 (Крит.)	адаптація: модульна архітектура. Інтеграції

№	Опис ризику	Ймовірність (1-5)	Вплив (1-5)	Ранг	Стратегія реагування (Mitigation Strategy)
	Games змінюють протокол авторизації, що блокує функцію імпорту.				винесені в окремі плагіни, ЯКІ оновлюються незалежно від ядра.
2	Блокування акаунтів (Anti-cheat). Античіт-системи ігор можуть помилково сприйняти gamehub як заборонене ПЗ.	2	5	10 (Вис.)	уникнення: відмова від втручання в процеси гри (ін'єкцій). Gamehub працює лише як лаунчер. Отримання цифрового підпису коду (Code Signing).
3	Низька продуктивність. Додаток споживає забагато RAM, що відлякує геймерів.	3	4	12 (Вис.)	мінімізація: вибір технологічного стеку Rust замість «важких» фреймворків. Регулярне навантажувальне тестування.
4	Відсутність інтересу спільноти. Після релізу розробники не захочуть писати плагіни.	3	4	12 (Вис.)	стимулювання: створення якісної документації SDK. Проведення конкурсів для розробників плагінів.
5	Перевищення бюджету. Розробка затягнеться через технічні складнощі.	3	3	9 (Сер.)	контроль: робота за моделлю Fixed Team. Пріоритизація MVP (відмова від другорядних фіч) для дотримання термінів.

Висновки до розділу 3

У третьому розділі розкрито практичні аспекти реалізації проєкту. Сформовано та пріоритезовано беклог продукту, що дозволяє команді фокусуватися на найважливіших функціях. Розраховано бюджет MVP, який становить 59 400 USD за 3 місяці роботи крос-функціональної команди. Впроваджена система моніторингу на базі Jira та процеси CI/CD забезпечують прозорість та високу швидкість адаптації, що є критичним для виживання продукту в умовах «війни лаунчерів».

ВИСНОВКИ

У кваліфікаційній роботі наведено теоретичне узагальнення і нове вирішення науково-прикладного завдання, що полягає у розробці механізму гнучкого управління створенням ігрової платформи «GameHub».

1. Аналіз теоретико-методологічних засад управління ІТ-проєктами показав, що для продуктів з високим рівнем невизначеності (як агрегатори контенту) традиційні каскадні моделі є неефективними. Обґрунтовано доцільність застосування Agile-підходу, який дозволяє мінімізувати ризики та забезпечити швидку реакцію на зміни ринку⁵.

2. Дослідження ринкового середовища виявило проблему фрагментації цифрового простору геймерів. На основі інструментів стратегічного аналізу (Lean Canvas, Value Proposition Canvas) розроблено бізнес-модель «GameHub», що базується на модульній архітектурі та залученні спільноти до технічної підтримки.

3. Стратегічне планування реалізовано через формування візії продукту (Product Vision) та дорожньої карти (Roadmap). Це дозволило структурувати розвиток платформи на етапи: від MVP до повноцінної екосистеми.

4. Практична модель управління базується на гібридному поєднанні фреймворків: Scrum (для розвитку ядра) та Kanban (для підтримки інтеграцій). Сформовано Product Backlog, проведено оцінку завдань та розроблено план спринтів.

5. Економічне обґрунтування підтвердило реалістичність проєкту. Розрахунок бюджету показав доцільність інвестицій у розмірі 59 400 USD для створення MVP, що є конкурентним показником для технологічного стартапу.

6. Система контролю якості та ризиків гарантує стабільність продукту. Визначено ключові ризики (зміна API, блокування) та розроблено стратегії їх мінімізації.

Таким чином, розроблений проєкт є готовим до впровадження планом створення конкурентоспроможного продукту «GameHub».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алькема В. Г., Кириченко О. С. Менеджмент організацій: навчальний посібник. Кн.1. Київ: Університет «КРОК», 2023. 276 с. URL: <https://library.krok.edu.ua/ua/kategoriji/navchalni-posibniki/1440-menedzhment-orhanizatsii>¹¹.
2. Андерсон Д. Kanban: Практичне керівництво для організацій, що постійно змінюються / Д. Андерсон. Пер. з англ. Київ: Діалектика, 2012. 320 с.¹².
3. Горковець, Д. О. Балдик. Управління ресурсами в ІТ проєктах. Держава, регіони, підприємництво: інформаційні, суспільно-правові, соціально-економічні аспекти розвитку: тези доповідей V Міжнародної конференції. Київ: Університет "КРОК", 2023.¹³.
4. Данченко О.Б. Практичні аспекти реінжинірингу бізнес-процесів / О.Б. Данченко. Київ. Університет економіки та права «КРОК», 2017. – 238 с.¹⁴.
5. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. URL: http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF¹⁵.
6. Кон М. Scrum: Довідник / М. Кон. Пер. з англ. Харків: Віват, 2016. – 352 с.¹⁶.
7. Конінг П. Інструментарій agile-лідера. Учимося успішно розвиватися за допомогою самокерованих команд / пер.з англ. В. Луненко. Харків: Фабула, 2023. - 224 с.¹⁷.
8. Окерман С., Рейндл С. Опанування професійного Scrum: практичний посібник із подолання труднощів і максимізації переваг Agile-підходу. Харків: Фабула, 2023. 224 с.¹⁸.
9. Орлова-Курилова О. В., Балдик Д. О., Мічківський С. М. Методичні рекомендації до виконання кваліфікаційної роботи здобувачами ОП «Agile технології розробки програмного забезпечення». Київ: Університет «КРОК», 2024. 72 с.¹⁹.
10. Agile Manifesto. Manifesto for Agile Software Development. 2001. URL: <https://agilemanifesto.org/>²⁰.

11. Cohn M. Succeeding with Agile: Software Development Using Scrum // Agile Conference (AGILE), 2010. P. 238-242.²¹.
12. Highsmith, Jim. Agile Project Management: Creating Innovative Products. Boston: Addison-Wesley, 2009. 432 p.²².
13. Kniberg, Henrik. Scrum and XP from the Trenches. Boston: C4Media, 2015. 162 p.²³.
14. PMI.org. What is Agile? URL: <https://www.pmi.org/agile>²⁴.
15. Schwaber, Ken. Agile Software Development with Scrum // Agile Journal. 2004. Vol. 3, No. 5. P. 12-18.²⁵.
16. Scrum Alliance. About Agile Project Management. URL: <https://www.scrumalliance.org/agile>²⁶.
17. Lukutin, O. An investigation into leadership and trust as antecedents of innovative behavior during the organizational transformation in a complex environment // VI Міжнародна науково-практична конференція. – Київ: Київський нац. ун-т ім. Т. Шевченка, 2022.

Технологічний стек реалізації проєкту*Таблиця А.1 – Перелік використаних технологій та інструментів*

Категорія	Назва технології / Інструменту	Призначення в проєкті
Управління проєктом	Jira Software	Планування спринтів, ведення беклогу
	Confluence	Зберігання технічної документації
Backend (Серверна частина)	Rust	Мова програмування ядра (забезпечує швидкодію)
	SQLite	Локальна база даних для зберігання бібліотеки
Frontend (Клієнтська частина)	React.js	Створення інтерфейсу користувача

Порівняльний аналіз конкурентних рішень

Таблиця Б.1 – Порівняння GameHub з основними конкурентами

Критерій	GameHub (Розробка)	Steam	GOG Galaxy 2.0	Playnite
Тип	Агрегатор (Мета-лаунчер)	Магазин + Лаунчер	Агрегатор + Магазин	Агрегатор
Підтримка платформ	Steam, Epic, GOG, Xbox	Тільки Steam	Всі основні	Всі основні
Споживання ресурсів (RAM)	Низьке (<100 МБ)	Високе (>400 МБ)	Високе (>300 МБ)	Середнє
Ліцензія	Open Source (Безкоштовно)	Пропрієтарна	Пропрієтарна	Open Source
Автоматичне закриття зайвих програм	Так	Ні	Ні	Так (через скрипти)