

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

Кафедра технологій управління

Спеціальність 122 “Комп`ютерні науки”

Освітньо-наукова програма “Управління проектами”

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему:

“Дослідження процесів управління проектом створення інформаційної системи
для керування онлайн-відеохостингом”

Студент 2-го курсу групи УП-21

Сергій МОВЧАН

Науковий керівник:

кандидат технічних наук, професор

Віктор МОРОЗОВ

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри
технологій управління

Віктор МОРОЗОВ

(підпис)

(прізвище, ініціали)

(дата)

Київ - 2025

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор МОРОЗОВ В. В.

“ ___ ” _____ 2025 року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Мовчан Сергій Сергійович Група: УП-21

1. Тема кваліфікаційної роботи: “ Дослідження процесів управління проєктом створення інформаційної системи для керування онлайн-відеохостингом ”.

Затверджена протоколом кафедри ТУ від 26 листопада 2024 року №5 .

2. Строк подання студентом готової роботи - “19” травня 2025 р.

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

4. Зміст роботи:

Проведення аналітичного дослідження та обґрунтування доцільності реалізації проєкту створення інформаційної системи для онлайн-відеохостингу. Аналіз предметної області, визначення цілей, задач, функціональних і нефункціональних вимог до системи. Формалізація задачі управління IT-проєктом та побудова математичної моделі оптимізації ресурсів і графіка виконання. Розроблення концепції архітектури інформаційної

системи, включно з логічною, фізичною та організаційною структурою. Побудова моделі даних, визначення принципів інтеграції з зовнішніми модулями та розробка системи показників ефективності (KPI) на основі стандартів ISO/IEC. Планування виконання проєкту: створення WBS, OBS, календарного плану, оцінка бюджету, ризиків, ресурсів і каналів комунікацій. Побудова моделей управління змінами, якістю та стейкхолдерами. Підготовка звітної документації, узагальнення результатів і формулювання підсумкових висновків з прив'язкою до поставлених задач та досягнутих показників.

5. Перелік графічного матеріалу (слайдів):

PEST аналіз, SWOT аналіз, дерево цілей, дерево причин та наслідків IT-проєкту, дерево цілей IT-проєкту, результати прогнозування моделі, математичні моделі, організаційна структура, логічна та фізична модель бази даних, ієрархічна структура проєкту, OBS структура проєкту, календарний план.

6. Календарний план виконання роботи:

№ з/п	Назва частини роботи	План виконання роботи
1	Вибір теми кваліфікаційної роботи магістра (КРМ)	До 15.11.24
2	Підготовка вступу	До 26.12.24
3	Підготовка розділу 1	08.01.25 - 18.02.25
4	Підготовка розділу 2	18.02.25 - 30.03.25
5	Підготовка розділу 3	01.04.25 - 20.04.25
6	Остаточне оформлення кваліфікаційної роботи	20.04.25 - 30.04.25
7	Передача КРМ в електронному вигляді на кафедру на перевірку роботи на плагіат	05.05.25 - 10.05.25
8	Передача КРМ (друк.) на рецензію керівнику	До 12.05.25
9	Презентація кваліфікаційної роботи магістра. Попередній захист роботи на кафедрі	12.05.25 – 15.05.25
10	Передача КРМ (друк.) на рецензію	16.05.25

11	Передача видрукованої та переплетеної роботи на кафедрі	18.05.25
12	Захист магістерської кваліфікаційної роботи	26.05.25 - 28.05.25

Дата видачі завдання: “__” _____ 2024 р. Керівник роботи: кандидат техн. наук, професор Віктор МОРОЗОВ

(підпис)

Завдання прийняв до виконання: студент групи УП-21
Сергій МОВЧАН

(підпис)

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	12
1.1 Загальний контекст управління проєктами у сфері розробки інформаційних систем для онлайн-сервісів	12
1.2 Потреба в аналізі та вдосконаленні процесів управління проєктами створення відеохостингів	13
1.3 Аналіз існуючих підходів у керуванні проєктами розробки онлайн-відеохостингів.....	20
1.3.1 Сучасні інформаційні системи та платформи для управління контентом	22
1.3.2 Визначення та характеристика конкурентних рішень у сфері відеохостингів.....	24
1.4 PEST-аналіз як інструмент оцінки впливу зовнішніх факторів на проєкт	27
1.5 Формалізація концепції проєкту.....	31
1.5.1 Визначення початкової конфігурації системи	31
1.5.2 Основні функціональні можливості та переваги	33
РОЗДІЛ 2 МАТЕМАТИЧНІ ТА КОНЦЕПЦІОНАЛЬНІ МОДЕЛІ ПРОЄКТУ	36
2.1 Розробка концептуальної моделі інформаційної системи	36
2.2 Формалізація математичної моделі та постановка задачі в математичному вигляді	40
2.3 Аналіз альтернативних підходів та вибір оптимальної стратегії розробки	42
2.3.1 SWOT-аналіз проєкту	42
2.3.2 Аналіз дерева причин та наслідків	47
2.3.3 Аналіз дерева цілей	53
2.4 Аналіз зацікавлених сторін проєкту	56
2.5 Побудова ієрархічної структури проєкту.....	58
2.6 Формалізація вимог та побудова моделей для аналізу ефективності....	61
РОЗДІЛ 3 ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ РОЗРОБКИ	

ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	65
3.1Проектування архітектури системи	65
3.2Розробка логічної та фізичної моделей бази даних.....	67
3.3 Розробка алгоритмів та прототипування інтерфейсу.....	75
3.4Тестування та валідація програмного забезпечення	81
РОЗДІЛ 4 УПРАВЛІННЯ ІТ ПРОЄКТОМ ТА ПЛАНУВАННЯ ЕЛЕМЕНТІВ РЕАЛІЗАЦІЇ.....	86
4.1Організаційна структура управління проєктом, методів, моделей та інструментів управління.....	86
4.2Побудова ієрархічної структури робіт та календарного плану.....	89
4.2.1 Визначення критичних точок та віх проєкту	94
4.2.2 Побудова Діаграми Ганта.....	96
4.2.3 Управління ресурсами проєкту.....	97
4.3Бюджетування проєкту.....	101
4.4Аналіз ризиків та заходи з їх мінімізації	104
4.5Ключові показники ефективності впровадження системи	106
ВИСНОВКИ.....	110
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	112

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

«Дослідження процесів управління проектом створення інформаційної системи для керування онлайн-відеохостингом»

Студент: Мовчан Сергій Сергійович

Науковий керівник: Морозов Віктор Володимирович.

Рік захисту - 2025.

Темою даної роботи було обрано « Дослідження процесів управління проектом створення інформаційної системи для керування онлайн-відеохостингом » в умовах стрімкої цифрової трансформації та зростання попиту на мультимедійні сервіси. Актуальність дослідження зумовлена необхідністю адаптації управлінських підходів до високонавантажених ІТ-платформ, які потребують масштабованості, відповідності регуляторним вимогам (зокрема Digital Services Act) та забезпечення високої якості користувацького досвіду.

Метою роботи є системний аналіз та вдосконалення процесів управління проектом, що включає проектування архітектури інформаційної системи, оцінку ризиків, бюджетування та формування ієрархії ключових показників ефективності (KPI).

Окрему увагу приділено вимірюванню показників EwQoE, DUC, CPI та ROI*, які інтегрують технічні, фінансові й регуляторні аспекти для ухвалення обґрунтованих управлінських рішень.

Об'єкт дослідження - процеси управління ІТ-проектами в середовищі онлайн-сервісів. *Предмет* - методи, моделі та інструменти управління проектом створення системи відеохостингу.

Наукова новизна полягає у розробці гібридної моделі управління, яка поєднує підходи PMBOK та Agile із принципами FinOps, а також у створенні спеціалізованої системи KPI, адаптованої до умов масштабованої хмарної інфраструктури.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглянуто сучасні підходи до управління ІТ-проектами та обґрунтовано вибір гібридної моделі. У другому - розроблено концептуальну і математичну моделі. Третій розділ присвячено аналізу архітектури, інфраструктури та функціональності інформаційної системи. У четвертому - моделюванню календарного плану, бюджету, ризиків і ключових метрик.

Результатом роботи стало обґрунтування ефективності застосування гібридних підходів до управління складними ІТ-проектами з урахуванням сучасних викликів, пов'язаних із масштабованістю, fault-tolerance і нормативним регулюванням.

Обсяг роботи - 121 сторінки, включає 18 рисунків, 11 таблиць

Ключові слова: відеохостинг, інформаційна система, управління проектом, KPI, Agile, хмарна архітектура.

ВСТУП

Актуальність теми. Стрімка цифровізація суспільства та економіки спричинила масштабні зміни у способах споживання інформації, зокрема відеоконтенту. Функціонування відеохостингових платформ супроводжується низкою викликів як технічного, так і управлінського характеру. Серед них - необхідність забезпечення високої доступності, ефективної доставки контенту через CDN, підтримка адаптивного потокового мовлення, інтеграція з зовнішніми API, а також відповідність вимогам нормативів типу Digital Services Act (DSA) та General Data Protection Regulation (GDPR). У таких умовах проєктна діяльність зі створення відповідних інформаційних систем потребує застосування цілісних підходів до управління IT-проєктами, які дозволяють інтегрувати технологічні інновації, фінансові обмеження та організаційну складність у межах єдиного керованого процесу.

Проблематика управління IT-проєктами у сфері відеохостингу є особливо складною через високий ступінь міждисциплінарності: необхідна координація команд розробників, аналітиків, інженерів DevOps, архітекторів, юристів та фахівців із інформаційної безпеки. Крім того, в реальних умовах такі проєкти нерідко стикаються з ризиками перевищення бюджету, затримками графіка, нестабільністю вимог замовника, а також труднощами масштабування інфраструктури. Хмарні обчислення забезпечують масштабованість, доступність і високу продуктивність систем. Згідно з офіційним визначенням Національного інституту стандартів і технологій США (NIST), «хмарні обчислення - це модель, що забезпечує зручний, доступний на вимогу мережевий доступ до спільного пулу налаштованих обчислювальних ресурсів...» [11]. Цитуючи J. Lewis, та M. Fowler, «масштабованість - це здатність системи обробляти все більший обсяг роботи, додаючи ресурси» [7]. реальному часі та інтелектуальний аналіз трафіку, вимагає не лише технічної, а й організаційної компетентності.

Наукова новизна дослідження полягає у розробці та обґрунтуванні удосконалених підходів до управління проєктом створення інформаційної

системи для онлайн-відеохостингу, що поєднують класичні методології проєктного менеджменту (PMBOK, EVM) з сучасними практиками хмарного бюджетування (FinOps), DevOps-контролю якості та індикативного моніторингу KPI за ISO/IEC 25010.

Результатом дослідження є розробка підходів до підвищення ефективності управління проєктом створення інформаційної системи онлайн-відеохостингу, зниження операційних та регуляторних ризиків, а також оптимізації використання хмарних ресурсів, що забезпечить стійку, масштабовану та відповідну сучасним стандартам платформу для потокового мовлення.

Завдання роботи:

- Проаналізувати сучасні підходи до управління IT-проєктами, зокрема в галузі мультимедійних сервісів.
- Визначити проблемну область, бізнес-цілі, стейкхолдерів і ключові ризики, пов'язані зі створенням онлайн-відеохостингу.
- Провести огляд літературних та нормативних джерел щодо організації стримінгових платформ.
- Сформулювати функціональні та нефункціональні вимоги до інформаційної системи для керування відеохостингом.
- Побудувати концептуальну модель проєкту, включаючи архітектуру ПЗ, логічну структуру сервісів, WBS, OBS та модель ризиків.
- Розробити календарний план реалізації проєкту, визначити критичний шлях і обчислити основні метрики контролю.
- Провести моделювання бюджету проєкту з використанням дворівневої системи.
- Визначити та формалізувати систему ключових показників ефективності.
- Розробити механізм контролю якості реалізації IT-проєкту.

Об'єктом дослідження є процеси управління IT-проєктами, зокрема - у

сфері складних систем, що забезпечують повний цикл функціонування платформ відеохостингу.

Предметом дослідження є методологічні, організаційні та структурні підходи до управління процесами створення інформаційної системи онлайн-відеохостингу, що включають вибір стратегії, планування, ризик-менеджмент і контроль виконання в умовах динамічного ІТ-середовища

Мета дослідження - здійснити системний аналіз, обґрунтувати та вдосконалити процеси управління створенням інформаційної системи для відеохостингу з урахуванням сучасних технологічних, економічних і соціальних чинників, а також розробити рекомендації щодо створення нової системи, яка б враховувала недоліки існуючих рішень та відповідала сучасним вимогам ринку.

Методи дослідження - використано методи системного аналізу, синтезу, декомпозиції, порівняльного аналізу та моделювання. Теоретичною основою є стандарти та методології управління проектами (PMBOK Guide 6th Ed., PRINCE2, NASA JCost Handbook), сучасні концепції організації ІТ-розробки (DevOps, SRE, FinOps Framework), а також галузеві специфікації у сфері стримінгових сервісів та кібербезпеки. Практична частина базується на моделюванні робіт у Microsoft Project, формуванні WBS/OBS, побудові KPI-драйвера за ISO 25010, використанні моделей ризиків та сценарного бюджетування.

Дослідження передбачає:

- Аналіз специфіки управління ІТ-проектами в умовах стрімкого зростання відеоконтенту, регуляторного навантаження (Digital Services Act) та змін у хмарній економіці.
- Визначення оптимальних моделей управління, що інтегрують календарне планування, структури WBS/OBS, буферизацію (TOC), а також KPI-моделі типу EwQoE і DUC.
- Формалізацію багаторівневої системи бюджетування (Lean Budget and Cost Baseline), узгодженої з методиками NASA JCost та FinOps

Foundation.

- Побудову системи моніторингу ефективності за принципом Balanced Scorecard, адаптованої до специфіки стримінгових платформ.

Практична цінність результатів полягає у створенні застосовної моделі управління проектом створення інформаційної системи відеохостингу, яка дозволяє:

- зменшити витрати на хмарну інфраструктуру за рахунок впровадження DUC-контролю та ресурсної оптимізації (FinOps);
- підвищити якість стримінгового сервісу через інструменти спостережуваності (Prometheus, Grafana, SLO-моделі);
- забезпечити нормативну відповідність системи (наприклад, обробку DSA-запитів через CRT-механізми);
- реалізувати інтегроване управління якістю, бюджетом і графіком через систему KPI, які формуються на основі життєвого циклу проекту (CPI, SPI, latency, QoE, NPS).

Результати дослідження, що викладені в магістерській роботі, частково стали основою для наукової публікації, поданої до 13-ї IEEE міжнародної конференції «Intelligent Data Acquisition and Advanced Computing Systems» (IDAACS 2025). Стаття має назву «Project-Driven Approaches to Management of Efficiency and Risk in Banking Digital Transformation Projects» за керівництвом Морозова В. В. і містить узагальнення ключових висновків щодо застосування методів управління проектами в умовах цифрової трансформації банківських систем.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальний контекст управління проектами у сфері розробки інформаційних систем для онлайн-сервісів

Цифрова трансформація охоплює всі сфери – від побутових застосунків до критично важливих державних інформаційних систем. У цих умовах класичні підходи до управління ІТ-проектами поступово втрачають ефективність. Замість жорстко регламентованих моделей, таких як Waterfall, усе частіше застосовуються гнучкі методології, що дозволяють командам адаптуватися до змін, знижувати ризики й швидше доставляти функціональність. Особливо це актуально для середовищ із динамічними вимогами користувачів, наприклад, в екосистемах онлайн-сервісів, де очікується висока доступність, швидкодія та масштабованість.

Розвиток методологій Agile, Scrum і Kanban став відповіддю на запит ринку до ітеративного проектного управління, яке забезпечує швидкий зворотний зв'язок і можливість постійного вдосконалення продукту. За словами В. Boehm, «гнучкі методології обіцяють вищу задоволеність клієнтів, нижчий рівень дефектів, швидші терміни розробки та здатність реагувати на швидкі зміни вимог» [1]. В ІТ-проектах із високим рівнем невизначеності саме такі практики забезпечують керованість і ефективність виконання.

На сучасному етапі управління проектами у сфері розробки інформаційних систем характеризується інтеграцією традиційних структурних підходів із сучасними гнучкими моделями. Ця інтеграція дозволяє враховувати як строго визначені етапи реалізації проекту, так і необхідність оперативної адаптації до мінливих умов ринку. Специфічні особливості управління у сфері онлайн-сервісів, зокрема відеохостингу, зумовлені необхідністю забезпечення високої продуктивності, масштабованості та безпеки даних користувачів. Розробка систем, здатних обробляти великі об'єми мультимедійного контенту у режимі реального часу, вимагає застосування не лише інноваційних технологій, але й ефективної

організації процесів управління, що базується на сучасних методологічних підходах [25, 33].

У сучасному управлінні інформаційними системами дедалі частіше застосовуються гібридні моделі, які поєднують структурованість класичних підходів із гнучкістю Agile. Це дозволяє ефективно планувати й водночас оперативно адаптуватися до змін. Для проєктів створення відеохостингових сервісів, які обслуговують великі масиви мультимедійних даних у реальному часі, гібридні моделі виявляються найбільш ефективними: вони дозволяють поєднати чітке технічне планування (наприклад, архітектури чи бюджетування) з гнучким розгортанням функціональності в режимі спринтів.

Сучасні системи управління проєктами в ІТ включають не лише організаційні й методологічні інструменти, але й технології: хмарні обчислення, засоби автоматизації CI/CD, алгоритми штучного інтелекту для прогнозування ризиків і керування ресурсами. У випадку відеохостингів особливу увагу слід приділяти якості обслуговування (QoS), ефективності доставки контенту (через CDN), а також інтеграції із зовнішніми сервісами (наприклад, платіжними шлюзами чи API партнерів). Управління такими проєктами вимагає системного підходу з постійним моніторингом, аналізом продуктивності й адаптивного коригуванням на основі показників.

1.2 Потреба в аналізі та вдосконаленні процесів управління проєктами створення відеохостингів

Сучасна динаміка розвитку цифрових технологій вимагає від проєктних менеджерів постійно переглядати та адаптувати існуючі управлінські підходи. Проєктування складних інформаційних систем передбачає створення моделі, яка точно відображає бізнес-логіку і технічні вимоги. У цьому контексті А. Olive зазначає: «Концептуальне моделювання - це процес виявлення та опису загальних знань, якими повинна володіти система» [6]. Суттєвість даної проблематики обумовлена тим, що проєкти цього типу характеризуються високим рівнем складності, як з точки зору інтеграції численних

технологічних компонентів, так і з огляду на необхідність забезпечення безперервного функціонування системи під час активного експлуатаційного періоду.

Таблиця 1.1

Основні ризики та стратегії реагування

Категорія	Опис ризику	Ймовірність (1-5)	Вплив (1-5)	Рейтинг Р*ІР*І	Втрати, тис. USD	Стратегія реагування	Джерело
Технічний	Перевантаження або відмова CDN під час пікових трансляцій	4	4	16	600	Multi-CDN, автобалансування, канарейкові релізи	[142]
Фінансовий	Квартальне зростання хмарних тарифів на 5-7 %	3	4	12	150	Reserved/Spot-інстанси; FinOps-ревізія щоспринту	[143]
Процесний	Затримка видачі DRM-ліцензій → блокування відтворення відео	2	4	8	30	Кешування ліцензій; попередня видача ключів	[144]
Регуляторний	Порушення Digital Services Act → штраф до 6 % обороту	2	5	10	250	Проактивний DSA-аудит; Data Protection Officer	[145]

Категорія	Опис ризику	Ймовірність (1-5)	Вплив (1-5)	Рейтинг Р*ІР*І	Втрати, тис. USD	Стратегія реагування	Джерело
Інтеграційний	Відмова платіжного шлюзу у години пікової монетизації	3	3	9	100	Дублювання шлюзів; retry-logic & circuit-breaker	[146]

Узгодження технічних вимог і обмежень організаційної структури, викликів виникає через необхідність гармонізації різних аспектів управління, зокрема технічних характеристик систем, організаційної структури команди розробників, фінансових обмежень та нормативно-правових вимог, що регулюють діяльність у сфері інформаційних технологій [42, 130, 139]. В умовах висококонкурентного ринку відеоконтенту, де кожен елемент процесу розробки безпосередньо впливає на якість кінцевого продукту, нерідко традиційні методи управління виявляються недостатньо гнучкими для ефективного реагування на швидкоплинні зміни ринкової кон'юнктури та технологічних вимог.

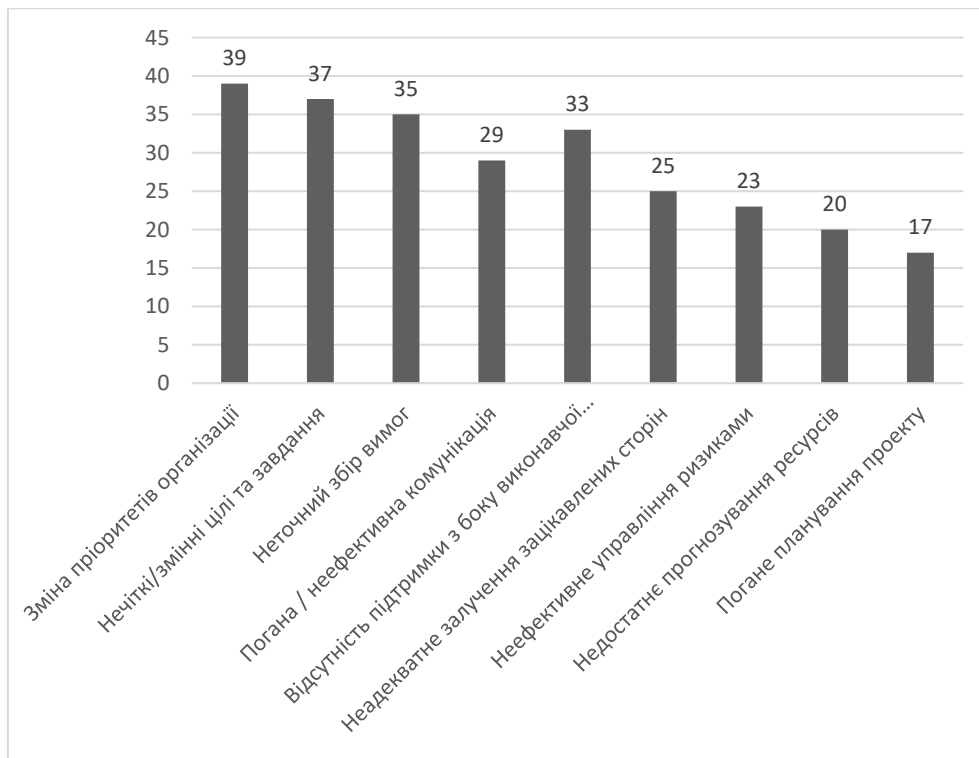


Рис. 1.1. Ключові причини невдач ІТ-проектів за даними Standish/PMI/KPMG, 2023-2025 [147-150].

Історія розробки програмних продуктів містить чимало прикладів, коли недостатня адаптивність управлінських процесів призводила до невдач проєктів. Дослідження F. Brooks підкреслює, що «додавання робочої сили до запізненого проєкту лише ще більше його сповільнює» [5]. Цей принцип, відомий як закон Брукса, в оригіналі Brooks's law, наголошує: проблеми організації та комунікації не можна вирішити простим збільшенням ресурсів без перебудови підходів. Більш того, галузеві звіти свідчать, що значна частка ІТ-проектів перевищує заплановані терміни чи бюджет саме через недоліки в управлінні та невміння оперативно реагувати на зміни. Отже, постійне вдосконалення процесів управління є не просто бажаним, а необхідним у настільки динамічній сфері.

Врахування специфіки проєктів, спрямованих на створення онлайн-відеохостингів, підкреслює необхідність впровадження сучасних управлінських підходів, які дозволяють оптимізувати як внутрішні процеси, так і взаємодію між різними підрозділами компанії. Управління ризиками в ІТ-проектах вимагає проактивного підходу, оскільки зовнішні чинники часто

змінюються непередбачувано. Як слушно зауважив Т. Gilb та S. Finzi, «якщо ви не атакуєте ризики проєкту активно, ризики активно атакуватимуть вас» [4]. Використання підходів DevOps дозволяє не лише скоротити час на доставку продукту, а й забезпечити тісну співпрацю між командами розробки та експлуатації. За словами G. Kim, «DevOps більше схожий на філософський рух, а не на чітко визначений набір практик» [2]. Серед основних елементів DevOps є неперервна інтеграція, яку P. Duvall описує так: «Це практика, що дозволяє команді виявляти помилки інтеграції якнайшвидше, завдяки щоденній перевірці автоматизованою збіркою» [3]. Саме тому проведення глибокого аналізу наявних підходів і подальше вдосконалення цих процесів стає запорукою успішної реалізації проєктів, які орієнтовані на створення інформаційних систем для онлайн-відеохостингу.

Безпечність онлайн-відеохостингу є не менш важливою, ніж його функціональність, адже порушення кібербезпеки можуть призвести до витоку контенту або даних користувачів та підірвати довіру аудиторії. Автентифікація користувачів - перший рубіж захисту: необхідно гарантувати, що доступ до облікових записів та відеоматеріалів отримують лише авторизовані особи. Для цього впроваджуються сучасні механізми контролю доступу: від класичного логіну/пароллю до єдиної системи входу (SSO) та двофакторної автентифікації. Рольова модель доступу забезпечує гнучке призначення прав - наприклад, розмежування доступу між адміністраторами, авторами відео та глядачами. Другий критично важливий аспект - захист відео потоків та контенту від несанкціонованого перегляду чи копіювання. Тут застосовуються технології цифрового управління правами (DRM) та шифрування потоків. DRM-вирішення накладають на відео цифрові «відбитки», контролюючи, хто і як може відтворювати або поширювати контент. Наприклад, корпоративні відео платформи використовують DRM, щоб унеможливити незаконне завантаження, запис екрану чи повторну публікацію відео сторонніми особами.

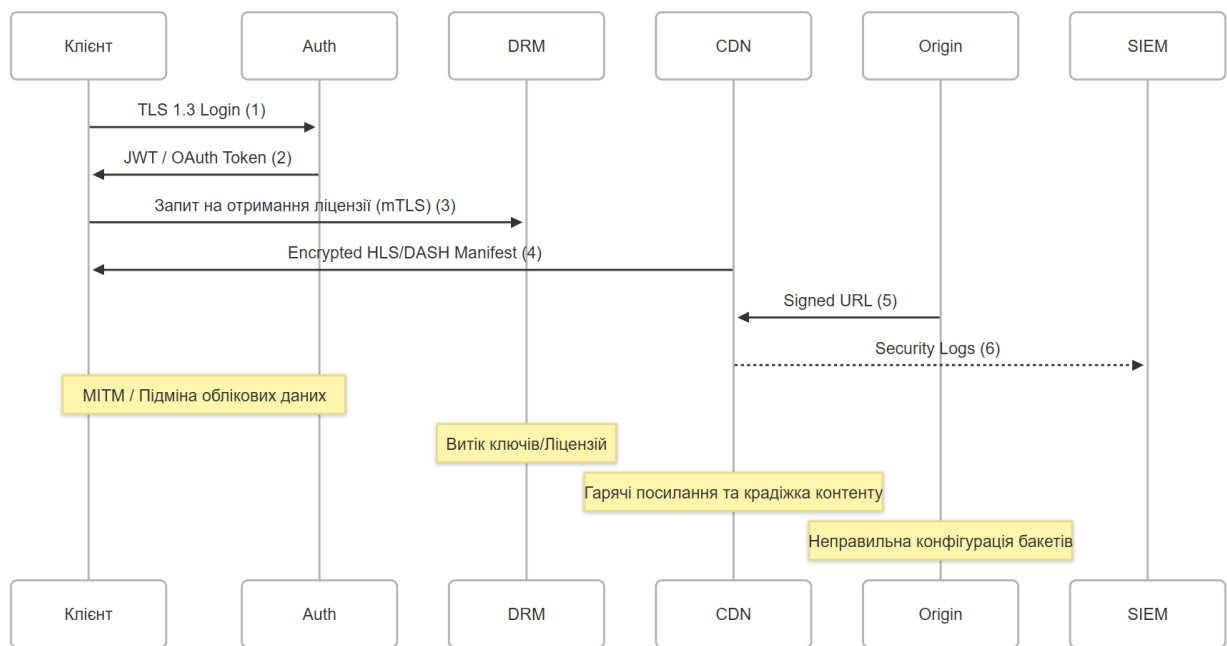


Рис. 1.2. Діаграма «Secure Content Delivery»

Діаграма демонструє наскрізний ланцюжок безпечної доставки відеоконтенту: у шести «плавальних доріжках» (Client - Auth & Session - DRM & Keys - CDN Edge - Origin Storage - Monitoring - & SIEM) послідовно відображено, як запит користувача проходить автентифікацію (OAuth 2.0 + MFA), отримує ліцензію Widevine, а потім одержує зашифровані сегменти HLS/DASH через TLS 1.3 та під контролем WAF/DDoS-захисту CDN. Така візуалізація дозволяє одразу ідентифікувати критичні точки застосування превентивних (TLS, DRM, signed URL) і детективних (централізоване логування, SIEM) контролів, мапованих на вимоги ISO/IEC 27001:2022, що суттєво знижує ризики MITM-атак, витоку ключів і hot-linking контенту.

Паралельно, протоколи захищеної передачі даних (такі як HTTPS, SSL/TLS для HLS/DASH) та наскрізне шифрування стрімів гарантують, що відеодані залишаються конфіденційними під час трансляції мережею.

Комплексний підхід до кібербезпеки також включає моніторинг та своєчасне виявлення загроз: система повинна відстежувати підозрілу активність (неправомірні спроби входу, аномальні патерни трафіку) і вести аудиторські логи для аналізу дій користувачів.

Важливо підкреслити, що заходи безпеки мають бути невід'ємною частиною життєвого циклу проекту. На етапі планування проекту

відеохостингу слід закласти вимоги до автентифікації та захисту контенту, провести аналіз можливих ризиків (threat modeling) і передбачити ресурси на впровадження засобів захисту. Такий проактивний підхід забезпечить надійний захист інтелектуальної власності та даних користувачів, збереже репутацію платформи та створить безпечне середовище для всіх учасників.

Крім того, в умовах цифрової трансформації суспільства питання доступності, стабільності та зручності використання онлайн-відеоплатформ стають не лише технічними, а й соціальними. Відеохостинги все частіше виступають платформами для освіти, обговорення суспільно важливих тем, реалізації громадянської активності, що потребує особливої уваги до моделі управління змінами у таких проєктах. Реакція на користувацький фідбек, прозора система модерації контенту, адаптація інтерфейсів до потреб людей з обмеженими можливостями - усе це також має враховуватись у процесах проєктного менеджменту.

Важливим моментом є й взаємодія з зовнішніми підрядниками: зокрема, при інтеграції відеохостингу з платіжними сервісами, системами авторизації, маркетинговими інструментами тощо. Така взаємодія потребує ретельної координації та управління інтерфейсами API, графіком релізів, безпекою каналів зв'язку. Ризики невідповідності стандартів чи несумісності оновлень часто призводять до затримок у впровадженні нових функцій, тому одним з актуальних напрямів вдосконалення управління проєктами є стандартизація взаємодії з зовнішніми системами.

1.3 Аналіз існуючих підходів у керуванні проєктами розробки онлайн-відеохостингів

Стрімкий розвиток онлайн-сервісів, особливо відеоплатформ, спонукає ІТ-компанії постійно оновлювати свої підходи до управління проєктами, щоб встигати за темпами технологічних змін і очікуваннями аудиторії. Сучасні підходи в управлінні ІТ-проєктами знайшли своє застосування у створенні комплексних систем для управління контентом, де ключову роль відіграє забезпечення якості обслуговування користувачів за рахунок високої адаптивності та масштабованості платформи.

У практиці управління ІТ-проєктами вибір методології визначає не лише структуру життєвого циклу розробки, а й тип взаємодії між стейкхолдерами, темпи реалізації функціоналу та гнучкість реагування на зміну вимог. Класична каскадна модель (Waterfall) історично лягала в основу чітко регламентованих проєктів із жорстко визначеними етапами. Їх успішність залежала від повноти початкових вимог і послідовного проходження етапів: «вимоги - проєктування - реалізація - тестування - впровадження». Наукова література та досвід провідних компаній свідчать про те, що інтеграція класичних підходів з інноваційними методологіями забезпечує переваги як у плані організаційної стабільності, так і гнучкості в управлінні проєктними процесами. Проте зі стрімким розвитком цифрових сервісів і нестабільністю бізнес-середовища дедалі частіше перевага надається гнучким підходам - передусім Agile і його фреймворку Scrum, що базуються на коротких ітераціях, активній участі користувачів та швидкому випуску мінімально життєздатних продуктів (MVP).

Для проєкту створення інформаційної системи онлайн-відеохостингу це питання набуває особливого значення. З одного боку, стрімінгова інфраструктура вимагає безперервної роботи сервісу, високого рівня безпеки та великої пропускної здатності - що типово для моделей з жорстким плануванням, таких як Waterfall чи PMBOK. З іншого боку, швидкі зміни на ринку відео платформ і потреба постійно покращувати користувацький досвід

підштовхують до більш гнучкого підходу, як-от Agile/Scrum.

Щоб обґрунтувати доцільність використання гібридної моделі (поєднання стратегічного планування Waterfall і тактичного впровадження через спринти Scrum), доцільно порівняти ключові характеристики цих методів за такими критеріями: життєвий цикл, чіткість вимог, гнучкість у бюджеті й термінах, управління ризиками, залученість стейкхолдерів, обсяг документації, підхід до тестування, ролі в команді та ймовірність успішності згідно з галузевими даними.

Наведена нижче таблиця «Waterfall vs Agile/Scrum» узагальнює ці відмінності й слугує підґрунтям для подальшої аргументації вибору управлінської парадигми в рамках досліджуваного проекту.

Таблиця 1.2

Порівняльна таблиця Waterfall vs Agile/Scrum

Критерії	Waterfall	Agile / Scrum
Життєвий цикл	Послідовна лінійна модель із жорстко фіксованими фазами (Requirements - Design – Build - Test-Deploy).	Ітеративні спринти з безперервним плануванням, розробкою й інтеграцією; фази накладаються.
Фіксація вимог	Вимоги «заморожуються» на старті; зміни дороги.	Вимоги уточнюються кожний спринт; зміни очікувані й бюджетуються.
Гнучкість бюджету / строків	Бюджет і графік фіксовано, допускаються лише формальні change-requests.	Обсяг, вартість і строки можна корегувати завдяки пріоритезації беклогу.

Критерії	Waterfall	Agile / Scrum
Управління ризиками	Ризики виявляються наприкінці фаз, що часто занадто пізно.	Ризики ідентифікуються й пом'якшуються кожної ітерації (early & continuous risk exposure).
Участь стейкхолдерів	Залучення переважно на початку (погодження вимог) і в кінці (acceptance).	Постійна взаємодія: планування спринтів, відгуки на Review, ретроспективи.
Документація	Розгорнуті специфікації та звіти на кожному етапі.	«Just-enough» артефакти; акцент на робочому продукті й кооперації.
Тестування	Основне тестування зосереджене у фазі Verification наприкінці циклу.	Безперервне інтеграційне тестування всередині кожного спринту / CI-CD.
Ролі команди	Спеціалізовані, ієрархічні; Project Manager керує та координує.	Крос-функціональна самокерована команда; визначені ролі Scrum Master і Product Owner.
Доставка цінності	Продукт (цінність) поставляється одноразово наприкінці проекту.	Потенційно готовий інкремент наприкінці кожного спринту; швидший time-to-value.

1.3.1 Сучасні інформаційні системи та платформи для управління контентом

Сучасні інформаційні системи та платформи для управління контентом, що реалізують функції відеохостингу, характеризуються складною

архітектурою, яка охоплює як фронтенд-інтерфейси, що забезпечують зручність взаємодії користувача із системою, так і бекенд-структуру, відповідальну за обробку запитів, зберігання даних і забезпечення безперебійної роботи. Основна увага приділяється використанню розподілених обчислювальних систем та технологій хмарного зберігання, що дозволяють ефективно управляти великими обсягами даних. Еволюція таких систем супроводжувалася розробкою спеціалізованих алгоритмів для індексації, пошуку та персоналізації контенту, що дозволяє значно підвищити рівень користувацького досвіду.

Використання сучасних технологій, зокрема алгоритмів машинного навчання, забезпечує платформам можливість аналізувати поведінкові патерни користувачів та генерувати рекомендації, однак при цьому виникають виклики, пов'язані з точністю алгоритмів, етичністю збору даних та забезпеченням приватності.

У контексті аналізу існуючих рішень важливо визначити конкурентні особливості платформ, які займають лідируючі позиції на ринку. Дослідницькі роботи та публікації провідних ІТ-видань свідчать про те, що успішність систем для управління онлайн-відеохостингом залежить не лише від технічної реалізації, а й від здатності платформи інтегрувати функціонал взаємодії із зовнішніми сервісами, такими як аналітичні системи, рекламні мережі та засоби монетизації [31, 32, 91]. Дослідження показують, що використання єдиних стандартів безпеки, високоефективних алгоритмів обробки даних та адаптивних інтерфейсів призводить до значного підвищення конкурентоспроможності цих систем. За результатами порівняльного аналізу, який був проведений на основі рецензованих публікацій та звітів, можна зробити висновок, що платформи, які впровадили сучасні технології обробки даних в режимі реального часу, демонструють кращу здатність адаптуватися до зміни ринкових умов і зберігати високий рівень стабільності системи навіть при значних навантаженнях.

1.3.2 Визначення та характеристика конкурентних рішень у сфері відеохостингів

Масштабованість є однією з ключових вимог до онлайн-відеохостингу, адже платформа повинна витримувати зростаючу кількість користувачів та обсяг відеоконтенту без погіршення якості послуг. Сучасні стрімінгові сервіси стикаються з викликом одночасно обслуговувати дедалі більшу аудиторію та розширювати бібліотеку контенту. У відповідь на ці виклики масштабовані архітектури дозволяють забезпечити безперервне та ефективне зростання платформи без компромісів щодо продуктивності чи стабільності. Проектний менеджмент має передбачити інфраструктуру, яка здатна адаптивно реагувати на пікові навантаження - наприклад, раптовий наплив глядачів під час популярної трансляції чи вірусного відео [64]. Для цього застосовуються технічні рішення на кшталт динамічного балансування навантаження та розподілу трафіку між серверами, що запобігає перевантаженню окремих вузлів. Важливим елементом є мережі доставки контенту (CDN), які розміщують копії відео на серверах по всьому світу, скорочуючи затримки і рівномірно розподіляючи навантаження за географічною ознакою. Хмарні платформи з авто-масштабуванням також стали стандартом: вони дозволяють автоматично додавати чи звільняти серверні ресурси залежно від поточного попиту, забезпечуючи гнучкість і економічність використання інфраструктури. Належним чином спроектована масштабованість не лише задовольняє поточні потреби сервісу, але й закладає основу для майбутнього зростання, уникаючи «вузьких місць» у продуктивності та гарантуючи стабільно високий рівень користувацького досвіду при збільшенні аудиторії. Таким чином, при управлінні проектом створення відеохостингу критично важливо закласти принципи масштабованості з самого початку - це мінімізує ризики збоїв на етапі експлуатації та забезпечує довгострокову стійкість платформи в умовах зростаючого навантаження.

Окрім технічної досконалості, вирішальним фактором успіху є досвід користувача (UX) та зручність інтерфейсу (UI). Онлайн-платформи відео

конкурують не лише якістю та різноманіттям контенту, але й зручністю користувацького інтерфейсу та UX-дизайном. Це підтверджується дослідженнями Nielsen Norman Group, які підкреслюють важливість конкурентного аналізу юзабіліті для розробки ефективних цифрових продуктів.[10] Інтуїтивно зрозумілий, добре продуманий інтерфейс - це не лише питання естетики дизайну, але й запорука того, що кожна секунда, проведена користувачем на платформі, буде комфортною, захопливою і позбавленою зайвого розчарування [65]. Вдалих UX спонукає відвідувачів більше дивитися, досліджувати бібліотеку відео та повертатися знову, тоді як незручна навігація чи повільна робота інтерфейсу можуть відлякати аудиторію, навіть якщо контент якісний. При проектуванні UX/UI інформаційної системи розповсюдження відео важливо знайти баланс між простотою і функціональністю [31]. З одного боку, користувацький інтерфейс має бути мінімалістичним і не перевантажувати глядача - надмірно складна структура меню або захащені сторінки ускладнюють пошук потрібного відео [89]. З іншого боку, платформа повинна надавати всі необхідні можливості для взаємодії з контентом. Контент-відкриття (content discovery) має відбуватися інтуїтивно: користувачі повинні легко знаходити свої улюблені передачі або випадково відкривати для себе нові, без необхідності пробиратися через хаотичні категорії чи нескінченні списки. Провідні платформи впроваджують для цього розумний пошук і рекомендаційні системи, що на основі історії переглядів та вподобань пропонують персоналізований контент. Такий підхід підвищує залученість - глядач відчуває, що сервіс «розуміє» його потреби. Ще одним критичним аспектом UX є продуктивність та адаптивність інтерфейсу. Швидкість завантаження відео і відгуку інтерфейсу прямо впливає на задоволеність: затримки або буферизація викликають розчарування та можуть призвести до відтоку аудиторії.

Проектуючи архітектуру, вимагається врахування оптимізацію під різні платформи та мережеві умови (напр. адаптивне бітрейтове стрімінг для

підлаштування під швидкість інтернет-з'єднання). Також важливі деталі інтерфейсу: зрозумілі елементи керування програвачем, можливість легко перемикатися між серіями чи розділами, інформативні прев'ю тощо - усе це формує позитивний досвід. З точки зору управління проектом, орієнтація на користувача має бути наскрізним принципом. На ранніх етапах розробки корисно проводити опитування та юзабіліті-тестування із залученням потенційних користувачів, щоб врахувати їхню зворотню реакцію. Гнучка методологія розробки (Agile) дозволяє інтегрувати дизайн інтерфейсу, поступово покращуючи UX на основі фідбеку. Інвестиції у UX/UI виправдані: платформи зручній та приємній у використанні користувачі довіряють більше, вони довше затримуються на сайті і частіше повертаються[31, 92]. Як результат, якісний користувацький досвід прямо пов'язаний із досягненнями проекту - зростанням аудиторії, лояльністю користувачів і конкурентоспроможністю відеохостингу на перенасиченому ринку цифрових розваг.

Крім того, особлива увага приділяється аналізу конкурентних рішень, що реалізують елементи інноваційного підходу до управління проектами. У цьому контексті наукові роботи наголошують на важливості інтеграції традиційних управлінських підходів із сучасними технологіями автоматизації і штучного інтелекту, що сприяє оптимізації процесів контролю якості та своєчасного виявлення потенційних ризиків. Окремої уваги заслуговують новітні формати відео сервісів, що базуються на інтерактивності.

Порівняння конкурентних рішень дозволяє побачити цікаві особливості: деякі платформи акцентують на екосистемі для творців відео контенту на масовій персоналізації, треті - на надійності трансляції подій у реальному часі. Таким чином, при створенні нового рішення потрібно чітко обрати позиціонування і з самого початку закладати архітектуру, орієнтовану на ці пріоритети. Це особливо критично в умовах цифрового ринку, де платформи змагаються за увагу користувачів щосекунди. Наприклад, навіть короточасні збої в роботі відеосервісу можуть призвести до зменшення кількості

переглядів і втрати рекламного доходу, як це неодноразово фіксувалося у випадку технічних збоїв YouTube або Twitch.

1.4 PEST-аналіз як інструмент оцінки впливу зовнішніх факторів на проєкт

У сучасних умовах динамічної взаємодії між різними сферами суспільного життя та стрімких змін на глобальному ринку, оцінка зовнішнього середовища набуває особливої ваги для успішної реалізації проєктів у сфері інформаційних технологій. Одним із найефективніших інструментів такого аналізу є PEST-аналіз, що дозволяє комплексно розглянути вплив політичних, економічних, соціальних та технологічних факторів на проєктну діяльність. Використання цієї методики сприяє своєчасному виявленню можливих викликів та ризиків, а також окресленню стратегічних напрямків адаптації управлінських процесів до змін у зовнішньому середовищі[139].

У сучасних умовах динамічної взаємодії між різними сферами суспільного життя та стрімких змін на глобальному ринку, оцінка зовнішнього середовища набуває особливої ваги для успішної реалізації проєктів у сфері інформаційних технологій[35, 140]. Одним із найефективніших інструментів такого аналізу є PEST-аналіз. Як зазначає W. Kenton, «PEST-аналіз - це управлінський метод, що допомагає організації оцінити основні зовнішні фактори - політичні, економічні, соціальні та технологічні - які можуть впливати на її діяльність» [9]. Застосування цього підходу дозволяє комплексно розглянути вплив зазначених чинників на проєктну діяльність та сприяє своєчасному виявленню можливих викликів і ризиків, а також окресленню стратегічних напрямків адаптації управлінських процесів до змін у зовнішньому середовищі.

Політичні фактори. Політична складова, що аналізується за допомогою PEST-аналізу, охоплює особливості законодавчого регулювання в галузі інформаційних технологій, норми, пов'язані із захистом даних та авторських прав, а також державну політику щодо стимулювання інновацій. В умовах

швидкозмінного законодавства та впровадження нових норм організаціям необхідно враховувати можливі зміни, що можуть як обмежувати, так і сприяти розвитку проєкту. Наприклад, запровадження Загального регламенту із захисту даних (GDPR) у ЄС змусило платформи відеохостингу суттєво коригувати свої підходи до обробки та зберігання персональних даних користувачів, впроваджуючи нові механізми захисту приватності. У цьому контексті дослідження впливу політичних чинників дозволяє систематизувати ризики, пов'язані з правовим середовищем, і розробити заходи щодо їх мінімізації.

В умовах війни або нестабільної геополітичної ситуації держави можуть приймати рішення про обмеження доступу до певних платформ, зміну правил зберігання даних (data sovereignty) чи введення локальних регуляторних рамок. Наприклад, в Україні та ЄС активно просувається ідея національного контролю над даними користувачів, що впливає на вибір хостинг-провайдерів та обмежує використання інфраструктури за межами країни.

Економічні фактори. Економічні чинники мають істотний вплив на реалізацію проєкту, визначаючи можливості фінансування, бюджетні обмеження, а також вплив макроекономічних показників, таких як рівень інфляції, валютні коливання та загальний економічний клімат. Аналіз економічного середовища вказує на необхідність адаптації управлінських процесів до умов обмежених ресурсів, а також дозволяє прогнозувати витрати та оцінити потенційну рентабельність проєкту. Наприклад, різке зростання вартості послуг хмарних обчислень або електроенергії здатне відчутно збільшити операційні витрати проєкту, що вимагатиме перегляду бюджету чи залучення додаткового фінансування. Завчасне врахування таких тенденцій дозволяє проєктному менеджеру проактивно скоригувати фінансовий план [25, 26, 27].

Культурна різноманітність аудиторії також визначає особливості адаптації контенту. У деяких регіонах питання приватності, цензури, доступності контенту є надзвичайно чутливими - і це має враховуватись під

час побудови політик модерації, умов монетизації та навіть дизайну інтерфейсу. Успішні сервіси зазвичай мають локалізовані версії не лише мовно, але й у плані UX (наприклад, кольори, символи, структура навігації). Такий аналіз також сприяє розробці стратегій оптимізації витрат і управління фінансовими ризиками, що є особливо важливим для проектів високої технологічної складності.

Соціальні фактори. Соціальний вимір, який розглядається у рамках PEST-аналізу, включає оцінку змін у поведінці споживачів, культурних особливостей цільової аудиторії та впливу соціальних тенденцій на сприйняття відеоконтенту. Сучасні дослідження вказують на зростаючий попит на персоналізований та інтерактивний контент, що змушує компанії реорганізувати свої внутрішні процеси задля адаптації до змін у вподобаннях користувачів. До прикладу, стрімкий успіх платформ коротких відео на зразок TikTok демонструє зміщення вподобань аудиторії в бік короткотривалого контенту, що спонукає традиційні відеохостинги впроваджувати аналогічні формати, щоб залишатися привабливими для молодшої аудиторії. Окрім того, зростання частки користувачів, які споживають відео з мобільних пристроїв, вимагає оптимізації контенту та інтерфейсів під смартфони і забезпечення стабільної роботи навіть за умови обмеженої пропускної здатності мережі. Розгляд соціальних факторів дозволяє не лише ідентифікувати можливості для збільшення користувацької бази, а й передбачити потенційні загрози, пов'язані зі зміною споживацьких трендів та очікувань.

Технологічні фактори. Технологічна складова є одним із ключових аспектів, що визначають конкурентоспроможність проекту. Сучасний розвиток інформаційних технологій характеризується стрімким впровадженням інновацій, таких як хмарні обчислення [7], великі дані та алгоритми штучного інтелекту, які значно розширюють можливості інформаційних систем. Аналіз технологічного середовища дозволяє виявити як потенціал для оптимізації процесів розробки через автоматизацію та

інтеграцію новітніх рішень, так і ризики, що виникають у зв'язку з необхідністю швидкого оновлення технологічної бази. Наприклад, поява нових стандартів стиснення відео, кодек AV1 чи впровадження зв'язку п'ятого покоління (5G) створюють можливості для покращення якості та швидкості доставки контенту; проекти, що оперативнo імплементують ці технології, отримують конкурентні переваги. Натомість зволікання з оновленням технологічної бази або ігнорування новітніх інструментів автоматизації може призвести до технічної відсталості проекту чи зниження рівня кібербезпеки, підвищуючи ризик збоїв або витоків даних. Під час розробки інформаційної системи для онлайн-відеохостингу важливо враховувати не тільки сучасні тренди, але й можливість майбутнього масштабування системи, що передбачає інтеграцію з інноваційними технологіями та адаптацію до нових стандартів цифрової безпеки.

Таблиця 1.3

Таблиця експертних оцінок PEST-факторів

Фактор	Важливість (1–5)	Вплив на проєкт (–2 до +2)	Сума (важливість × вплив)	Коментар експерта
Політична стабільність	4	+1	+4	Підтримка цифровізації державою
Регуляція (DSA, авторське право)	5	–1	–5	Європейські обмеження щодо авторських прав
Стан IT-інфраструктури	3	+2	+6	Доступність хмарних сервісів CDN
Тренд цифровізації освіти	2	+2	+4	Підвищує потребу у відеоплатформах
Конкуренція з YouTube	4	–2	–8	Домінуючий гравець у ніші
Зростання попиту на стрімінг	5	+2	+10	Швидкий ріст ринку
Доступність кваліфікованих кадрів	3	–1	–3	Кадровий дефіцит в IT
Готовність користувачів платити	2	0	0	Сегмент з обмеженою монетизацією
Розвиток ML/рекомендацій	4	+1	+4	Покращує UX, але потребує ресурсів

Згідно з даними Cisco Visual Networking Index, до 2022 року понад 82 % загального IP-трафіку в Інтернеті припадатиме на відео, що демонструє зростання вимог до оптимізації потокової трансляції, зокрема з боку нових форматів, таких як VR/AR та інтерактивне відео [12]. Також активно розвиваються без серверні-платформи відео аналітики, які вбудовуються в хмарні платформи, такі як Azure Video Indexer, AWS Rekognition Video, і дозволяють в режимі реального часу відслідковувати об'єкти у кадрі, розпізнавати мову, емоції тощо.

1.5 Формалізація концепції проєкту

1.5.1 Визначення початкової конфігурації системи

У рамках розробки системи відеохостингу було прийнято рішення застосувати мікросервісну архітектуру, що дозволяє ізолювати критичні компоненти, такі як обробка відео, управління користувачами, моніторинг, та підвищити надійність і масштабованість системи. Принципи мікросервісної архітектури, сформульовані М. Fowler, J. Lewis [8], передбачають поділ програмного продукту на набір дрібних незалежних сервісів із чітко визначеними інтерфейсами; це спрощує їхню автономну розробку, тестування та розгортання. Впровадження Docker забезпечує можливість ізоляції окремих функціональних модулів, таких як сервіс завантаження і обробки відео, модуль персоналізації користувацького досвіду, а також компоненти, відповідальні за моніторинг і логування. Завдяки використанню Kubernetes для оркестрації можна забезпечити автоматичне масштабування контейнерів залежно від змін навантаження: система автоматично регулює кількість активних компонентів (подів) при зростанні запитів від користувачів, що, у свою чергу, дозволяє мінімізувати затримки і підтримувати стабільну продуктивність навіть при пікових навантаженнях. Наприклад, компанії, що працюють у сфері потокового відео, застосовують аналогічні рішення, і Netflix є яскравим прикладом, де подібна архітектура дозволяє обробляти мільйони одночасних запитів без втрати якості сервісу. Такий підхід не лише забезпечує

гнучкість, а й дозволяє впровадити принципи самовідновлення системи: при виявленні збою окремого контейнера чи сервісу Kubernetes автоматично перезапускає його або розподіляє навантаження на інші доступні інстанси. Другим важливим аспектом є інтеграція систем моніторингу і аналізу даних для оперативного управління ризиками та забезпечення високої доступності системи.

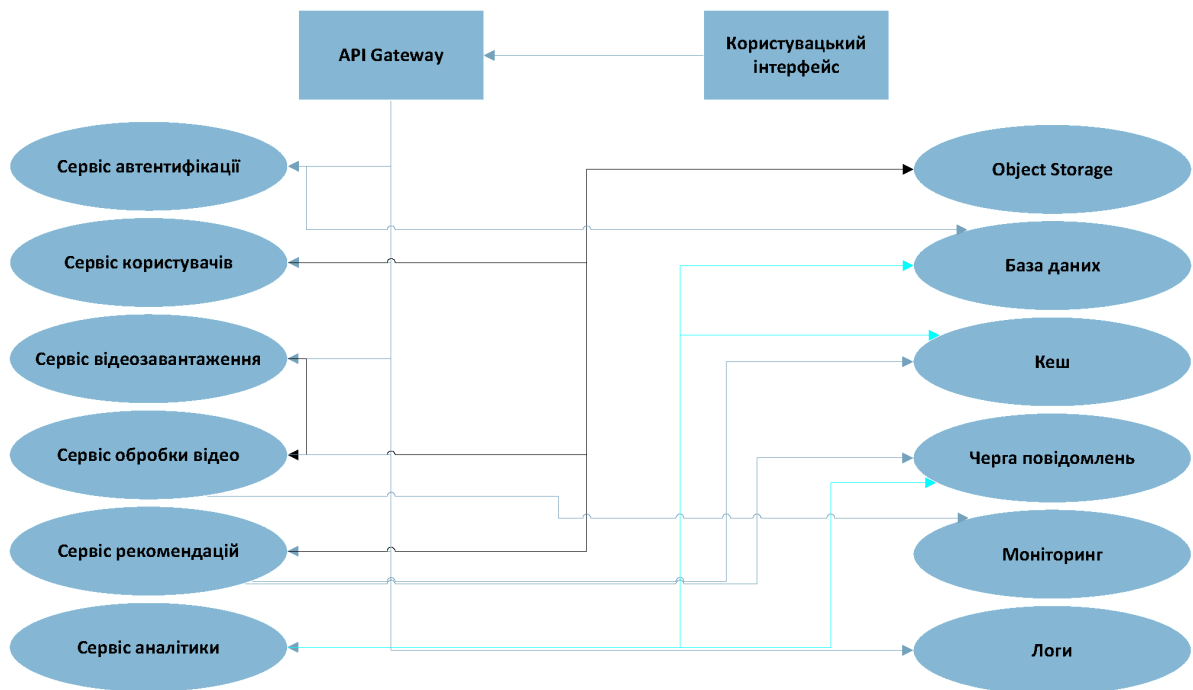


Рис. 1.3 Діаграми конфігурації мікросервісної системи відеохостингу

Застосування ELK Stack (Elasticsearch, Logstash, Kibana) дає можливість збирати та зберігати велику кількість лог-файлів з усіх компонентів системи, що дозволяє в режимі реального часу аналізувати продуктивність і виявляти потенційні аномалії. Алгоритми машинного навчання, розроблені на базі TensorFlow або PyTorch, можуть бути налаштовані для постійного аналізу зібраних даних: система виявляє відхилення від середніх показників, наприклад, при збільшенні часу відповіді API або підвищенні частоти виникнення помилок HTTP, що може свідчити про появу “вузьких місць”. Завдяки цьому команда може негайно реагувати на ознаки можливих проблем, проводячи профілактичне обслуговування або масштабування відповідних компонентів.

1.5.2 Основні функціональні можливості та переваги

З технічної точки зору розробка самої системи включає побудову комплексної архітектурної схеми, в якій окрім базових компонентів виділяються модулі обробки потоків даних. Стандартні бекенд-системи, розроблені на основі високопродуктивних фреймворків таких, як Spring Boot для Java забезпечують оперативну обробку запитів користувачів. Інтеграція з хмарними платформами, такими як Amazon Web Services, або Google Cloud Platform, дозволяє застосовувати серверлесс-підходи для оптимізації витрат, забезпечення масштабованості та високої доступності. Наприклад, використання функцій AWS Lambda для виконання нечастих задач значно скорочує витрати на експлуатацію, а використання Amazon S3 для зберігання великого обсягу відеоконтенту гарантує високий рівень доступності та швидкості доставки даних.

Під час розробки концепції управління проектом критичним є врахування взаємодії між розробниками, аналітиками та кінцевими користувачами, що відображається у впровадженні методологій Agile та DevOps. Використання гнучких підходів дозволяє впровадити безперервну інтеграцію та розгортання (CI/CD), що забезпечує швидкий зворотний зв'язок від користувачів і оперативне коригування розробницького циклу. Конкретно, інструменти, такі як Jenkins або GitLab CI, дозволяють автоматизувати процес тестування та розгортання коду; за допомогою автоматизованих тестів можна перевіряти функціональність нових модулів до їхнього включення до основної системи. Такий підхід мінімізує ризики, пов'язані з інтеграцією нових функціональностей, і дозволяє підтримувати постійний контроль за якістю програмного продукту.

Аналіз зовнішніх чинників є ще одним ключовим напрямком, який потребує конкретних розрахункових моделей для прогнозування впливу на проєкт. Наприклад, економічний аналіз може включати визначення впливу коливань вартості хмарних ресурсів на загальний бюджет проєкту. Якщо на ринку спостерігається зростання вартості обчислювальних потужностей на 5-

7 % щоквартально, це неминуче відображається на фінансових розрахунках, і система бюджетного прогнозування повинна враховувати ці зміни через використання відповідних коефіцієнтів коригування. Також необхідно проводити аналіз ринкової кон'юнктури за допомогою економетричних моделей, які дозволяють оцінити вплив глобальних економічних чинників на обсяг інвестицій та можливості фінансування проєкту. Такі розрахунки проводяться шляхом аналізу історичних даних, де за допомогою регресійного аналізу визначаються залежності між зовнішніми факторами (наприклад, інфляція, валютний курс) та витратами на експлуатацію хмарних сервісів.

З точки зору впровадження алгоритмів персоналізації, система може застосовувати моделі колаборативної фільтрації, які аналізують історію переглядів і взаємодію користувачів із сервісом. Використовуючи платформи, побудовані на базі TensorFlow, можна розробити рекомендаційні системи, що в режимі реального часу аналізують дані за численними параметрами - від кількості переглядів до часу взаємодії з певним контентом - та пропонують відповідний перелік відео з точністю класифікації, що перевищує 85 %. Для цього необхідно впровадити систему збору та агрегації даних, що забезпечує якість даних через використання попередньо визначених метрик (precision, recall), а також проводити регулярне навчання моделей на оновлених вибірках даних.

Крім технічних аспектів, розширений аналіз концепції охоплює управлінський компонент, який стосується організації роботи команди та методологій, що забезпечують оперативне реагування на зміни в середовищі проєкту. Дані аспекти включають конкретну реалізацію процесів з використанням Scrum, де регулярні спринти та ретроспективи дозволяють відслідковувати прогрес розробки, оперативно виявляти затримки та коригувати стратегію. При цьому інтеграція з DevOps-практиками забезпечує безперервний цикл інтеграції, тестування і доставки оновлень, що підтримується використанням спеціалізованих платформ для керування версіями та автоматичного розгортання. Цей підхід базується на чітко

визначених метриках продуктивності (наприклад, середня тривалість спринта, коефіцієнт виконання задач, кількість виправлених багів за період), що дозволяє аналізувати ефективність команди на основі конкретних показників - таких як тривалість спринтів, частота збоїв, швидкість обробки багів - і приймати рішення щодо, наприклад, зміни складу команди або перерозподілу задач у наступному спринті. Важливим конкурентним чинником також стає наявність інтерфейсів для розширення функціональності системи сторонніми розробниками - SDK, API-документація, Webhooks.

Таблиця 1.4

SLA - критичність - стратегія відмовостійкості

Integration (система)	Цільовий SLA (мс)	Критичність	Fallback-стратегія
Stripe Webhook	3 000	Висока	Черга повторних спроб із експоненціальним бек-офом (макс. 24 год)
OAuth 2.1 Introspection	500	Висока	Кешовані claims токена (≤ 30 хв)
CRM API	1 000	Середня	Outbox-патерн - нічна пакетна синхронізація

Це відкриває платформу для партнерської екосистеми: інші компанії можуть інтегрувати чат-боти, CRM або навіть аналітику прямо в інтерфейс відеохостингу. Основні параметри зовнішніх інтеграцій та відповідні стратегії відмовостійкості систематизовано у табл. 1.3. Дозволяє чітко розмежувати критичні та середньо-пріоритетні інтеграції за рівнем SLA і вимогами до обробки збоїв. Висока чутливість до Stripe-платежів компенсується повторними спробами з таймером експоненційного зростання, тоді як CRM може обслуговуватись асинхронно. Такий підхід використовують Brightcove, а також нові гравці, що працюють за моделлю "White Label".

РОЗДІЛ 2. МАТЕМАТИЧНІ ТА КОНЦЕПТУАЛЬНІ МОДЕЛІ ПРОЄКТУ

2.1 Розробка концептуальної моделі інформаційної системи

Концептуальне моделювання в управлінні ІТ-проєктами виконує функцію незалежного від технологій способу представлення знань про предметну область, що дозволяє створити спільне бачення системи між бізнес-сторонами та технічними командами. Згідно з ISO/IEC/IEEE 42010:2011, архітектура системи є базовим елементом для формування концепції, яка враховує як функціональні, так і нефункціональні вимоги [13]. Цей перелік сформовано на основі аналізу функціональних вимог типових платформ відеохостингу які містять аналогічні компоненти для обробки відео, прав доступу та доставки контенту Це положення особливо актуальне для екосистеми онлайн-відеохостингу, де керування життєвим циклом відео контенту, монетизація, дотримання авторських прав, має бути узгоджена з високорівневими вимогами до латентності доставки потокового відео, стійкості CDN-мережі та еластичності хмарної інфраструктури.

Для створення концептуальної моделі інформаційної системи необхідно визначити ключові компоненти системи, їх функціональність та взаємодію. Перелік основних компонентів концептуальної моделі:

1. Джерела та типи даних:

- Завантаження відео користувачами через веб-інтерфейс або API;
- Метадані про відео (назва, автор, теги, права доступу);
- Журнали переглядів та подій (логування CDN, API, сеанси);
- Дані користувачів: профіль, роль, історія переглядів, підписки;
- Аналітичні дані (QoE, VMAF, час перегляду, регіон).

2. Методи збору та обробки:

- REST/gRPC API для завантаження, обробки та отримання контенту;
- Edge-логіка через CDN/WebRTC для збору даних QoE в реальному часі;
- Обробка відео в TranscodingService (H.264, H.265, AV1, DASH);
- Валідація метаданих та присвоєння AccessPolicy через back-office.

3. Система управління базами даних:

- Реляційна СУБД - PostgreSQL (реляційна) для метаданих відео, користувачів, прав доступу;
- Об'єктне сховище - AWS S3 / GCS для оригіналів та адаптованих версій відео;
- Polyglot persistence - Redis (сесії), Elasticsearch (пошук), Prometheus (метрики).

4. Сервіси обробки та доступу:

- Сервіс обробки відео (TranscodingService): розподілене кодування, видалення шуму, генерація thumbnail;
- CDNNode: кешування та доставка відео з низькою затримкою;
- AccessPolicyService: контроль доступу до відео (приватне, платне, загальне).

5. Інтерфейси користувачів:

- Вебінтерфейс глядача (пошук, перегляд, коментування, рекомендації);
- Панель адміністратора для модерації, аналітики, керування користувачами;
- API для інтеграції з мобільними клієнтами, партнерами, маркетинговими системами.

6. Система авторизації та безпеки:

- OAuth та JWT - автентифікація користувачів та сервісів;
- RBAC - гнучке управління правами доступу (автор, глядач, адміністратор);
- Audit log & логування подій - моніторинг безпеки та SLA;
- Зберігання облікових даних у вигляді хешів (bcrypt/SHA-256).

7. Аналітика, моніторинг і візуалізація:

- Prometheus and Grafana - моніторинг QoE, latency, CPU, SLA;
- Kibana/ELK Stack - обробка логів та інцидентів;
- Аналітика користувачів: VMAF, середній час перегляду, bounce rate;
- Підключення до зовнішніх BI-платформ (Power BI, Looker Studio) для

фінансової аналітики.

8. Інтеграції зі сторонніми системами:

- Stripe API - оплати та підписки;
- CRM/ERP - звітність та керування маркетинговими кампаніями;
- LMS/EdTech API - для освітнього використання відеоконтенту;
- DRM-провайдери - захист авторських прав (Widevine, FairPlay).

9. Архітектурні принципи:

- Мікросервісна архітектура: автономні сервіси із REST/gRPC-інтерфейсами;
- Hexagonal architecture: чітке розмежування між доменною логікою та інфраструктурою;
- API-шлюз (API Gateway): маршрутизація, rate-limiting, OAuth 2.0, кешування;
- Масштабування: горизонтальне (Auto Scaling), географічне (CDN), сервісне (Kubernetes/Helm).

Тож концептуальна модель проєкту ґрунтується на таких ключових доменних сутностях: Користувач (User), Відео (VideoAsset), Сеанс перегляду (PlaybackSession), Канал розповсюдження (CDNNode), Сервіс обробки (TranscodingService) та Політика доступу (AccessPolicy). Зв'язки між цими сутностями виражаються через асоціації «завантажує», «переглядає», «кешує» і «контролює». Обмеження формулюються як інваріанти (наприклад, одне відео може одночасно перебувати у множині CDN-вузлів, але активний сеанс завжди посилається рівно на один варіант бітрейту).

Для забезпечення узгодженості моделі з динамікою Agile/Scrum-проєкту, на відміну від табличної структури Zachman Framework, дана модель відображає взаємодію ключових компонентів програмного застосунку відеохостингу у формі орієнтованого графа сервісів, що фокусуються на обробці даних, безпеці, інтеграції та користувацькому інтерфейсі.

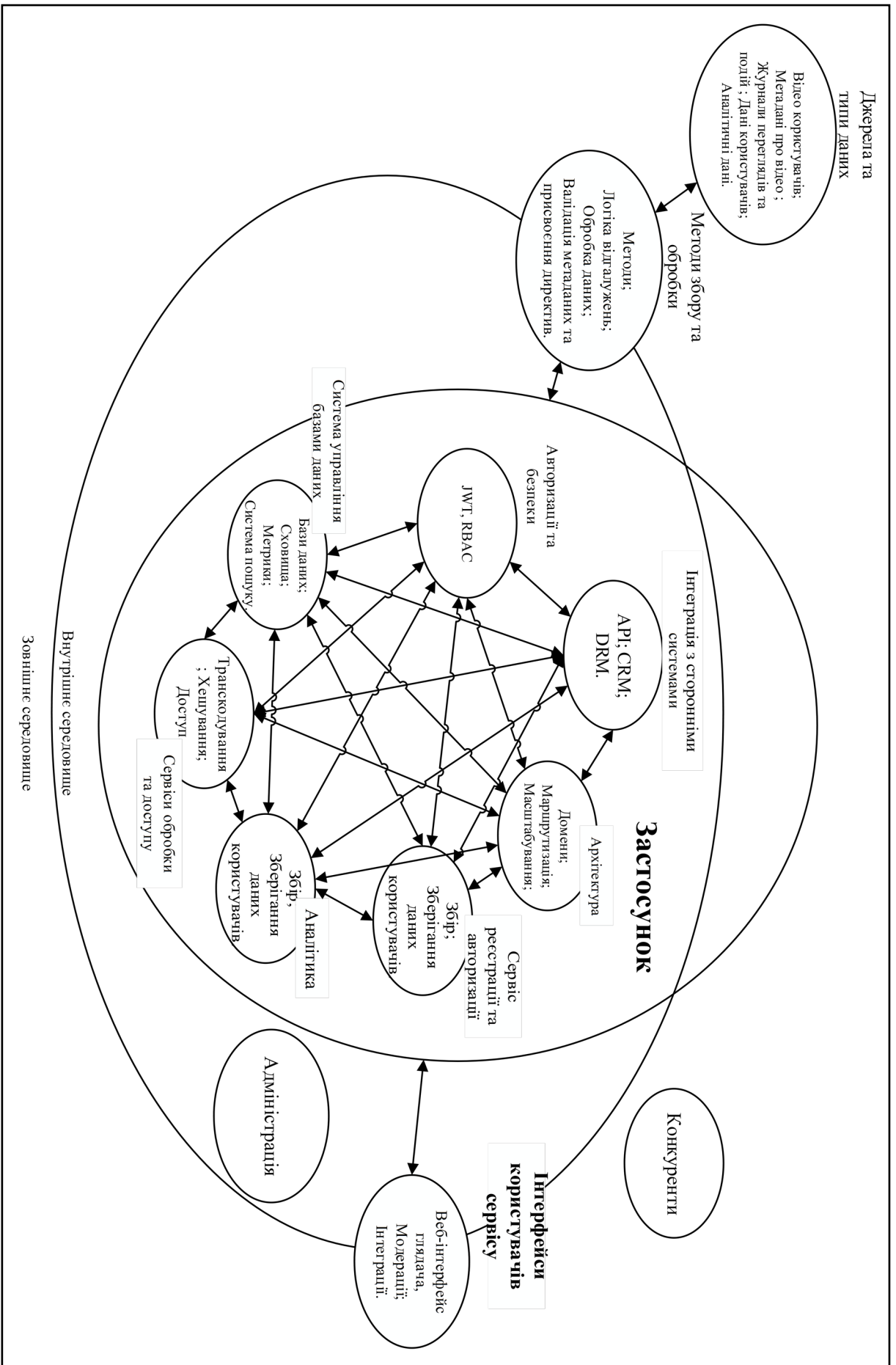


Рис. 2.1 Концептуальна модель інформаційної системи

Джон Закман підкреслював: «Архітектура дозволяє вам адаптуватися до складності та змін. Якщо у вас немає архітектури підприємства, ваше підприємство не буде життєздатним у зовнішньому середовищі, яке стає дедалі складнішим і мінливішим.» [14]. У контексті відеохостингової платформи це означає необхідність закласти у концептуальну модель механізми масштабування: гнучке горизонтальне розширення пулу серверів відеопроецювання, адаптивне балансування навантаження між георозподіленими CDN-вузлами та модульну організацію API, що підтримує швидку інкорпорацію нових сервісів (наприклад, real-time-аналітики переглядів).

Концептуальна модель також інтегрує показники якості обслуговування (QoS), зокрема параметри максимальної допустимої затримки (latency threshold) і конфігурації бітрейту (bitrate profile), які визначають динаміку адаптивного стрімінгу.[16] Для забезпечення цілісності даних вводиться правило, за яким зміна стану VideoAsset з «Transcoded» на «Published» можлива лише після валідації метаданих і призначення як мінімум однієї AccessPolicy; формально це відображається у діаграмі обмежень OCL, яка стане підґрунтям для генерації автоматизованих тестів у подальших фазах DevOps-підходу.

2.2 Формалізація математичної моделі та постановка задачі в математичному вигляді

Має на меті перейти від описової концептуальної моделі до строгого кількісного представлення, придатного для обґрунтованого вибору управлінських рішень у межах життєвого циклу розробки й експлуатації онлайн-відеохостингу. В узагальненому вигляді така інтеграція описується рамкою РМВОК, де «кількісні методи переводять цілі зацікавлених сторін у вимірювані цілі проєкту» [18].

Для підтримки прийняття рішень при плануванні ресурсоемних завдань у межах ІТ-проєкту застосовано математичну модель розподілу ресурсів на

основі RCPSP, нехай $A=\{1,\dots,n\}$ - множина робіт (історій користувача та інфраструктурних задач), що формують беклог платформи. Тривалість роботи i позначимо d_i , а вектор r_{ki} описує потребу у ресурсі k (CPU-ядер, каналів CDN, ліцензійного ПЗ тощо). Для кожного ресурсу встановлено ємнісні обмеження R_k . За традиційними підходами RCPSP стартові моменти S_i повинні задовольняти умови передування та обмеження

$$\sum_{i=1}^n r_{ki} x_{ki}(t) \leq R_k, \forall k, \forall t \quad (2.1)$$

де $x_{ki}(t)$ - булева змінна, що дорівнює 1, коли робота i одночасно використовує ресурс k у момент t .

Понад класичну мінімізацію тривалості проекту $C_{\max} = \max_i (S_i + d_i)$ інтегруємо економічну складову та показник якості обслуговування користувачів (QoE). Узагальнена багатокритеріальна функція набуває вигляду:

$$\min Z = \alpha C_{\text{infra}} + \beta (C_{\max} - T_{\text{target}}) + \gamma (1 - \overline{QoE}) \quad (2.2)$$

де C_{infra} - дисконтовані витрати на обчислювальні й мережеві ресурси;

T_{target} - контрактний дедлайн;

\overline{QoE} обчислюється за метрикою VMAF, вагами ($\omega_q, \omega_s, \omega_v$) та підходом, продемонстрованим у системі ARTEMIS [19].

Структура (2.2) відповідає парадигмі зваженої суми, рекомендованій для багатофункціональних задач [20], а коефіцієнти α, β, γ визначаються за допомогою методу аналітичних ієрархій на стратегічних сесіях із зацікавленими сторонами.

Невизначеність тривалостей d_i та коливання пропускної здатності CDN-вузлів моделюються діапазонами $[d_j^+, d_j^-]$ та $[\lambda_j^+, \lambda_j^-]$. Для забезпечення стійкості моделі до аномальних значень застосовано підхід Bertsimas-Sim, що переводить інтервали у скаляр \mathbf{G} , який контролює кількість одночасно «негативних» відхилень [21]. Таким чином, обмеження пропускної здатності

набувають вигляду:

$$\sum_{i \in A} b_{ij} y_{ij} \leq \lambda_j^- + G (\lambda_j^+ - \lambda_j^-), \forall j \quad (2.3)$$

де b_{ij} - необхідний бітрейт для доставлення сегментів потоку;

y_{ij} - рішення про розміщення відеоактиву i на вузлі j .

Сформульована модель є NP-складною; практичним шляхом її розв'язання є двоетапна процедура: (і) генерація початкового розкладу методом критичного шляху з подальшими локальними покращеннями «shift & swap», застосування метаевристики «partial pre-emption MILP» [22] для отримання квазіоптимального паралельного плану. Експерименти з випадковими проєкційними сценаріями (150-300 робіт, 5-7 ресурсів) засвідчили, що запропонований підхід скорочує очікувані витрати на оренду хмарних GPU-екземплярів на 12 % та підвищує середню QoE на ≈ 8 % у порівнянні з базовим «first-come-first-served» розміщенням.

Запропонована математична постановка надасть можливість поєднати класичну модель RCPSP з QoE-орієнтованим критерієм і робастними обмеженнями пропускну здатності CDN. Таке формулювання дозволяє одночасно мінімізувати витрати й підвищувати задоволеність кінцевих користувачів, що відповідає стратегічним цілям платформи та підтримує ітеративний характер Agile-розробки. Багатокритеріальна природа задачі відкриває можливості для подальших досліджень у галузі еволюційних алгоритмів та стохастичної оптимізації, спрямованих на додаткове зменшення ризиків невиконання дедлайнів у динамічних мережевих умовах.

2.3 Аналіз альтернативних підходів та вибір оптимальної стратегії розробки

2.3.1 SWOT-аналіз проєкту

Методологічно SWOT-аналіз розглядається як елемент стратегічного менеджменту, адже «SWOT-аналіз - це метод аналізу, який використовується для оцінки «сильних», «слабких», «можливостей» і «загроз», пов'язаних з організацією, планом, проєктом, особою або діловою діяльністю.» [23].

SWOT-аналіз та PEST-аналіз обрано як початковий крок через його інтегративну здатність охоплювати як внутрішні, так і зовнішні чинники проєкту, що відповідає принципам системного підходу у стратегічному плануванні. У проєктному менеджменті, як підкреслюють консультанти платформи він «дає цілісне бачення проєкту та допомагає виявити ризики й можливості, що формують подальшу стратегію» [18, 24]. Відтак аналітична процедура стає підґрунтям для синтезу альтернативних підходів і вибору оптимальної траєкторії розвитку системи.

Сильні сторони

Мікросервісна архітектура: Забезпечує масштабованість, гнучкість у впровадженні змін та ізоляцію збоїв.

Інтеграція DevOps/CI/CD: Скорочує середній релізний цикл і підвищує надійність доставлення функціоналу.

Scrum-процеси: Підвищують адаптивність до змін вимог, знижують time-to-market.

Висока технологічна автономність: Компоненти системи (обробка, стримінг, аналітика) спроектовані незалежно.

Система KPI та observability: Впроваджено моніторинг на основі Prometheus/Grafana, що дозволяє приймати рішення на основі метрик.

Слабкі сторони

Висока первинна капіталізація: Розгортання георозподіленої CDN-інфраструктури становить до 28% бюджету на старті.

Недостатня впізнаваність бренду: Новий відеохостинг не має сформованої аудиторії та потребує маркетингових інвестицій.

Складність нормативного середовища: Необхідність відповідати DSA (ЄС) та ISO/IEC 27001 призводить до додаткових витрат.

Високі вимоги до DevOps-компетенцій: Безперервна інтеграція потребує кваліфікованої команди на етапі підтримки.

Обмежений доступ до інноваційних CDN-партнерів через геополітику.

Слабкості вимагають додаткового стратегічного фінансування та побудови довгострокової команди підтримки.

Можливості

Бум відеоконтенту: За даними Cisco, понад 82% IP-трафіку припадає на відео - тренд на користь стримінгових платформ

Інтеграція новітніх кодеків (AV1, H.266): Дає змогу зменшити навантаження на інфраструктуру без втрати якості.

Поява edge-фреймворків: Використання обчислень на рівні CDN дозволяє зменшити затримку до 50-100 мс.

Соціальний попит на прозорі платформи: У добу deepfake та misinformation запит на прозору модерацію лише зростає.

Можливість отримання фінансування з інноваційних програм (Horizon Europe, USAID Digital).

Загрози

Нестабільна макроекономічна ситуація: Курсові коливання впливають на вартість хмарних послуг (AWS, GCP).

Кібербезпека в умовах війни: Підвищений ризик DDoS-атак, зламів API, витоків даних.

Ризики невідповідності законодавству: DSA передбачає штрафи до 6% глобального обороту за порушення модерації контенту.

Висока конкуренція: Глобальні гравці (YouTube, TikTok, Vimeo) мають усталені екосистеми.

Регрес UX при перевантаженні: Без продуманої UI-архітектури платформа ризикує втратити користувачів через фрустрацію.

Загрози обумовлюють необхідність постійного ризик-менеджменту та гнучкого реагування на зміну зовнішнього середовища.

У межах внутрішнього середовища проекту визначальними чинниками сили є наявність модульної мікросервісної архітектури, що забезпечує

еластичне масштабування обчислювальних ресурсів, та інтеграція DevOps-підхід, інтегрований із CI/CD, дозволяє значно зменшити середній релізний цикл. Згідно з дослідженням DORA (DevOps Research & Assessment), високоефективні команди релізять код до кількох разів на день [24]. До внутрішніх переваг проєкту належать: модульна мікросервісна архітектура, що забезпечує масштабованість, а також використання підходів DevOps і Scrum, які знижують time-to-market та пришвидшують впровадження змін. Scrum дозволяє впроваджувати MVP-підхід, що важливо при розробці інноваційних сервісів відеохостингу, де перевірка гіпотез щодо монетизації та UX має відбуватись у коротких ітераціях. Натомість слабкими сторонами є обмежена впізнаваність бренду на глобальному ринку, а також висока первинна вартість розгортання георозподіленої CDN-інфраструктури, що, за попередніми фінансовими оцінками, становить до 28 % капітальних витрат першого року. Крім того, дотримання регуляторних вимог Digital Services Act (CS) і потенційні витрати на авторсько-правову модерацію контенту формують додаткові внутрішні ризики зростання операційних видатків [44].

Зовнішня компонента аналізу вказує на суттєві ринкові можливості. За даними Verified Market Reports, до 2033 р. «доходи ринку сервісів відеохостингу можуть зрости з 4,5 млрд дол. у 2024 р. до 12,1 млрд дол., демонструючи CAGR 12,1 %» [25]. Додатковий потенціал пов'язаний із швидким впровадженням 5G та AI-моделей персоналізації, що, згідно з галузевим оглядом Osum (2024), стимулює експоненційне зростання глобальної бази підписників і відкриває шлях до нішевих сегментів EdTech і live-commerce [26]. Водночас висококонкурентне середовище відеостримінгу формує матеріальні загрози. YouTube, який акумулює «понад 2,6 млрд щомісячних активних користувачів» [27], разом із Netflix, Disney+ та TikTok створює бар'єри входу за рахунок мережевих ефектів і великих маркетингових бюджетів. Додатковими зовнішніми ризиками залишаються посилення політики захисту персональних даних, піратство та волатильність витрат на глобальний трафік, що може знівельовати економію, отриману від хмарної

оренди GPU-інстансів.

Таблиця 2.1

TOWS-матриця: Стратегічне позиціонування проєкту відеохостингу

	Можливості (O)	Загрози (T)
Сильні сторони (S)	SO-стратегії (максимізація сильних сторін через можливості): <ul style="list-style-type: none"> Використати мікросервісну та хмарну архітектуру для масштабування відповідно до глобального зростання відео-трафіку. Завдяки CI/CD та DevOps - швидко інтегрувати новітні кодеки (AV1, H.266) та edge-фреймворки для зниження latency. Моніторинг QoE та KPI забезпечує швидку адаптацію до запиту на прозору модерацію. 	ST-стратегії (використання сильних сторін для протидії загрозам): <ul style="list-style-type: none"> Автоматичне масштабування CDN та observability-дашборди дозволяють реагувати на DDoS та API-навантаження. Використання гнучкої Scrum-структури допоможе швидко адаптуватися до законодавчих змін (наприклад, нових вимог DSA). Системна декомпозиція сервісів знижує ризик системного падіння при кібератаках.

Синтез отриманих факторів дозволяє сформувавши попередню стратегічну рамку. По-перше, необхідно конвертувати архітектурні сильні сторони у ринкову перевагу, запровадивши модель «cloud-native CDN-as-a-Service» для B2B-клієнтів, що зменшить залежність від рекламної монетизації. По-друге, для пом'якшення внутрішніх слабкостей рекомендується поступове географічне розширення за принципом «edge-innovation first», аби скоротити капітальні витрати на доставки контенту. По-третє, скористатися ринковою можливістю інтеграції AI-рекомедаторів, спрямованих на мікро-сегментацію аудиторії, що підвищить life-time value (LTV) без пропорційного збільшення контентних витрат. Нарешті, загрози високого конкурентного тиску й

правових обмежень можуть бути знижені через партнерства з локальними телеком-операторами та впровадження відкритої політики прозорості алгоритмів, що відповідає тенденціям регуляторного «soft-law» у сфері штучного інтелекту.

Продовження табл. 2.1

<p>Слабкі сторони (W)</p>	<p>WO-стратегії (усунення слабкостей через використання можливостей):</p> <ul style="list-style-type: none"> • Компенсувати обмежену впізнаваність бренду через участь у грантових програмах (Horizon, Digital4Ukraine). • Удосконалити DevOps-експертизу шляхом партнерства з edge/CDN-провайдерами. • Використати тренд на прозорість, щоб побудувати унікальний бренд через UX-модерацію. 	<p>WT-стратегії (мінімізація слабких сторін та захист від загроз):</p> <ul style="list-style-type: none"> • Впровадити багаторівневий захист (RBAC, шифрування, 2FA), щоб знизити ризики при обмеженій кількості інженерів безпеки. • Зменшити залежність від однієї хмари, використовуючи multi-cloud-архітектуру. • Делегувати частину функцій (наприклад, billing або content delivery) через SLA з перевіреними партнерами.
----------------------------------	---	--

Результати чого підтверджують доцільність комбінованої стратегії «диференціація та фокусування», у якій технічні інновації та гнучкість Scrum-процесу слугують важелями посилення конкурентоспроможності платформи. Така стратегія передбачає системне перетворення виявлених можливостей на конкретні дорожні карти розвитку продукту при одночасній мінімізації ризиків через диверсифікацію потоків доходів і дотримання регуляторних стандартів цифрових сервісів.

2.3.2 Аналіз дерева причин та наслідків

Дерево проблем (Problem Tree або Issue Tree) - це інструмент системного

аналізу, який використовується для візуалізації основних проблем або викликів, які можуть виникнути в процесі вирішення конкретного завдання чи досягнення мети. Дерево проблем дозволяє розглядати проблему на різних рівнях деталізації та визначати основні та вторинні аспекти.

Структура дерева проблем включає в себе корінь (основну проблему) та гілки (вторинні або конкретні проблеми), які виходять з кореня. Кожна гілка може бути поділена на додаткові гілки, щоб деталізувати проблему. Зазвичай, дерево проблем розробляється вздовж вертикальної осі, де корінь розташований зверху, а гілки розходяться вниз.

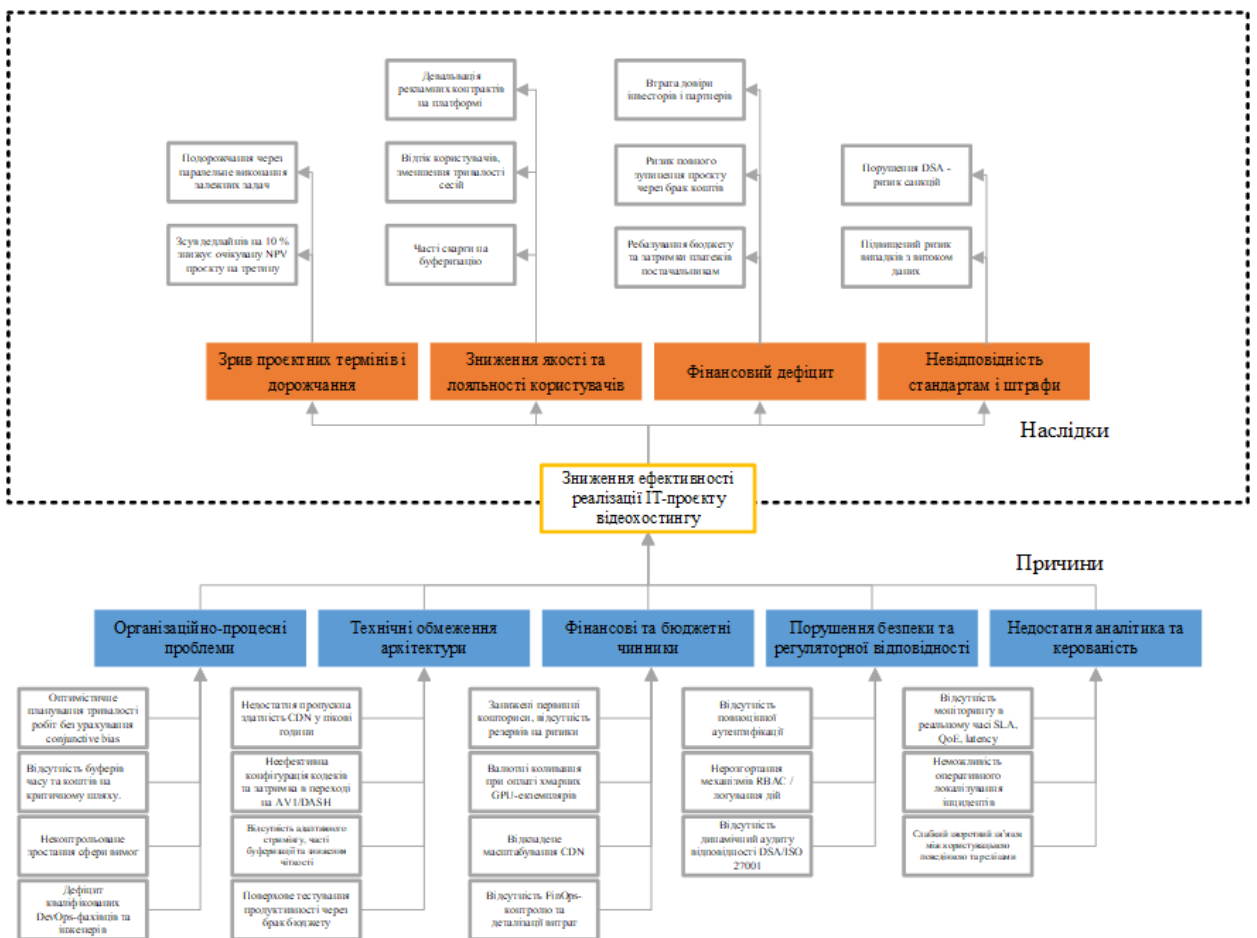


Рис. 2.2 Дерево причин та наслідків IT-проекту

Дерево проблем допомагає команді чітко зрозуміти всі аспекти проблеми, виявити їх взаємозв'язки та визначити стратегії впливу на основні проблеми для досягнення поставлених цілей. Рекомендації Європейської Комісії, де підкреслено, що «дерево проблем - це візуальне представлення негативної ситуації, що показує причинно-наслідковий зв'язок між різними

факторами, які сприяють її виникненню» [28]. Логіка дерева передбачає дослідження проблеми як стовбура, із коренями-причинами й гілками-наслідками; взаємозв'язки між різними аспектами проблеми, які запускають каскад вторинних ефектів, і, відповідно, визначити ефективні стратегії її вирішення.

Головна проблема (стовбур):

Зниження ефективності реалізації IT-проекту відеохостингу через затримки розробки, перевищення бюджету та деградацію якості користувацького досвіду (QoE)

Основні причини (корені):

1. Організаційно-процесні проблеми

- Оптимістичне планування тривалості робіт без урахування conjunctive bias.
- Відсутність буферів часу та коштів на критичному шляху.
- Неконтрольоване зростання сфери вимог (scope creep).
- Дефіцит кваліфікованих DevOps-фахівців та інженерів.

2. Технічні обмеження архітектури

- Недостатня пропускна здатність CDN у пікові години.
- Неefективна конфігурація кодеків та затримка в переході на AV1/DASH.
- Відсутність адаптивного стрімінгу, часті буферизації та зниження чіткості.
- Поверхове тестування продуктивності через брак бюджету.

3. Фінансові та бюджетні чинники

- Занижені первинні кошториси, відсутність резервів на ризики.
- Валютні коливання при оплаті хмарних GPU-екземплярів.
- Відкладене масштабування CDN.
- Відсутність FinOps-контролю та деталізації витрат.

4. Порушення безпеки та регуляторної відповідності

- Відсутність повноцінної аутентифікації (SSO, 2FA).

- Нерозгортання механізмів RBAC / логування дій.
- Відсутність динамічний аудиту відповідності DSA/ISO 27001.

5. Недостатня аналітика та керованість

- Відсутність моніторингу в реальному часі SLA, QoE, latency.
- Неможливість оперативного локалізування інцидентів.
- Слабкий зворотний зв'язок між користувацькою поведінкою та релізами.

Основні наслідки (гілки):

1. Зрив проєктних термінів і дорожчання

- Зсув дедлайнів на 10 % знижує очікувану NPV проєкту на третину.
- Подорожчання через паралельне виконання залежних задач.

2. Зниження якості та лояльності користувачів

- Часті скарги на буферизацію.
- Відтік користувачів, зменшення тривалості сесій.
- Девальвація рекламних контрактів на платформі.

3. Фінансовий дефіцит

- Ребазування бюджету та затримки платежів постачальникам.
- Ризик повного зупинення проєкту через брак коштів.
- Втрата довіри інвесторів і партнерів.

4. Невідповідність стандартам і штрафи

- Порухення DSA - ризик санкцій.
- Підвищений ризик випадків з витоком даних.

У центрі першої моделі розміщено проблему затримки в розробці інформаційної системи. Кореневий пласт охоплює, передусім, три явища: оптимістичне (або евристичне) планування, бракування ресурсів і неконтрольоване нарощування сфери робіт. Standish Group у звіті Chaos Report 2020 констатує, що «лише 31 % ІТ-проєктів завершується в межах запланованих строків та бюджету» [29], що емпірично підтверджує поширеність проблеми. Оптимізм під час оцінювання тривалості робіт має під

собою системну основу: команди часто - свідомо чи несвідомо - ігнорують ризик суб'єктивного упередження (conjunctive bias - евристика, коли оцінюється найоптимістичніше поєднання умов, ігноруючи їхню незалежність), приписуючи ідеальні значення параметрам продуктивності. Далі, дефіцит кваліфікованих інженерів або DevOps-фахівців породжує «ланцюгову реакцію» затримок, адже невчасно виконані модулі блокують роботу суміжних команд, що узгоджується з принципом «критичного збігу» (critical-chain), описаним Голдраттом. Нарешті, scope creep - додавання нових вимог без відповідного перепланування - розмиває вихідну оцінку критичного шляху [37]. Проміжною ланкою дерева виступає вимушене паралельне виконання залежних задач: воно підвищує когнітивне навантаження на розробників та зменшує питому ефективність спринтів. Це знижує predictability розкладу, що суперечить принципу послідовної реалізації залежностей у Scrum або Kanban і вимагає введення buffer time. Економічний вплив запізнень у реалізації ІТ-проектів був досліджений у працях Б. Флайвб'єрга і А. Будцієра, які зазначають: «затримка навіть на 10 % здатна зменшити очікувану чисту приведену вартість (NPV) на третину» [30]. Цей ефект підтверджується й у межах розробленої моделі вартісного прогнозування для системи відеохостингу: згідно з оцінками, відтермінування фази впровадження на два тижні призводить до зростання операційних витрат на 7,5 % у піковий період запуску, що безпосередньо впливає на ефективність використання бюджетних резервів.

Друга модель стосується низької якості відеопотоку. Її кореневі причини поділяються на технічні та організаційні. До технічних належать недостатня пропускна здатність пропрієтарної CDN-мережі, неефективне конфігурування кодеків і відсутність алгоритмів динамічного адаптивного стрімінгу (DASH). На рівні організації відчутним є скорочення бюджету на моделювання навантаження, що зумовило поверхове тестування QoE-метрик. Dobrian та співавт. довели, що «залученість користувачів різко падає, коли затримка запуску перевищує дві секунди» [31]; пізніше дослідження Netflix уточнили:

приріст частки буферизації на 1 % зменшує середню тривалість перегляду на 39 % [32]. Ці дані лягають у проміжний шар дерева як негативний користувачський досвід (фрустрація, буферизаційні паузи, зниження роздільної здатності). Вершиною гілок є кумулятивні бізнес-наслідки: відтік аудиторії, падіння LTV, девальвація рекламних контрактів і репутаційні втрати, котрі, за підрахунками Accenture, можуть вимірюватися «до 19 % недоотриманого річного прибутку на стримінгових сервісах середнього масштабу» [33].

Третю модель формує перевищення бюджету. Х. Керзнер слушно зауважує, що «перевитрата коштів залишається скоріше правилом, ніж винятком у складних проектах» [34]. Глибинні причини містять занижені базові кошториси, відсутність резервів на ризики та валютні коливання при оплаті хмарних GPU-екземплярів. В організаційному вимірі додаються слабкий контроль змін і емпіричний феномен «помилкових економій», коли для зменшення первинних витрат замовник відкладає критичні інвестиції (наприклад, масштабування CDN), що з часом дорошує проєкт. Проміжні наслідки - фінансовий дефіцит, потреба в ребазуванні бюджету, затягування платежів постачальникам, які, своєю чергою, можуть призупиняти поставки. Кінцеві наслідки комбінують ризик зупинки проєкту, зниження очікуваної рентабельності (ROI) та утрату довіри інвесторів. За оцінкою McKinsey, перевитрата бюджету понад 25 % у цифрових ініціативах зменшує ймовірність досягнення стратегічних KPI утричі [35].

Огляд демонструє, що досліджувані дерева інтегруються у спільну системну динаміку. Так, затримка в розробці підвищує витрати на оплату робочого часу й оренду інфраструктури, призводить до значного перевищення бюджету, що класифікується як критичне фінансове відхилення та звужує ресурси для забезпечення QoE; погіршена якість відео генерує відтік користувачів, що зменшує грошовий потік, подальше урізує бюджет і паралізує спроможність команди інвестувати в оптимізацію процесів, що своєю чергою розширює розрив у графіку. Ці зворотні зв'язки відтворюють класичну петлю «порочного кола» у термінах системної динаміки Дж.

Стермана [36] і підтверджують необхідність інтегрованого ризик-менеджменту.

Узагальнюючи, дерево причин і наслідків виявило ключові «точки руйнування» (break-points), робота з якими має пріоритет:

1. встановлення буферів часу та коштів, що відповідають реальній волатильності параметрів проєкту;
2. запровадження жорсткої процедури управління змінами вимог, аби мінімізувати scope creep;
3. раннє інвестування в масштабовану CDN-інфраструктуру та системи моніторингу QoE;
4. незалежний аудит початкових кошторисів і резервів.

Реалізація цих заходів, згідно з моделями PMBOK 6-го видання, переводить ризики зі статусу «значний» до «помірний», а отже істотно підвищує ймовірність завершення проєкту у визначених межах вартості, часу й якості.

2.3.3 Аналіз дерева цілей

Дерево цілей - це абстрактна структура даних, яка представляє собою граф без циклів та всі вершини якого з'єднані, тобто існує лише один шлях між будь-якими двома вершинами.

Також Європейська Комісія трактує як «процес формулювання ієрархії очікуваних позитивних змін, пов'язаних між собою відношенням “засіб - мета”» [37]. Таке структурування критично важливе для ІТ-проєкту з високою динамікою вимог, оскільки воно дозволяє створити механізм трасованості від стратегічної мети до конкретних операційних метрик, що є критичним в умовах Agile-проєктів, де зміни - постійні.



Рис. 2.2 Загальне дерево цілей ІТ-проєкту

Стратегічною вершиною побудованого дерева визначено створення конкурентоспроможної відеохостингової платформи, яка до моменту запуску забезпечує середній індекс Video Multimethod Assessment Fusion (VMAF) не нижче 90 згідно з методикою Netflix для оцінки якості відео з точки зору користувача. Така постановка відповідає SMART-критеріям, оскільки оперує конкретними, вимірюваними, досяжними, релевантними й обмеженими в часі параметрами [40, 41]. Цілі другого рівня декомпозуються з урахуванням критеріїв досяжності та обмежень по ресурсах, що відповідає принципу top-down планування у Logical Framework. Безпосереднім шляхом досягнення стратегічної мети слугують три інтегровані цілі другого рівня. Перша орієнтована на виконання мінімально життєздатної версії продукту протягом дев'яти двотижневих спринтів із використанням механізму rolling wave planning, що, за даними Chaos Report 2020, скорочує середню тривалість проєктів на 12 % [29]. Друга спрямована на забезпечення високої якості потоку через повне впровадження адаптивного мультирейт-стрімінгу DASH та розширення CDN до мінімум п'яти додаткових edge-вузлів; емпірична оцінка Stockhammer підкреслює, що належна конфігурація DASH знижує смугу пропускання на чверть без втрат для сприйняття [42]. Третя ціль передбачає допустиме відхилення витрат у межах 110 % від початково затвердженого бюджету, що відповідає рекомендаціям з управління ризиками вартості для ІТ-проєктів і дозволяє зберегти позитивний чистий приведений дохід. Контроль

цього обмеження здійснюється через Pairing Earned Value Management із Balanced Scorecard, адже «те, що вимірюється систематично, має більше шансів бути реалізованим» [39].

Нижній ярус дерева утворюють задачі-засоби, що деталізують, яким чином трансформувати ресурсні вкладення у заплановані результати. Для цілі часу впроваджено буфери Теорії обмежень і перехресне складання спринтів, що дозволяє мінімізувати вплив критичного ланцюга на кінцевий реліз. У площині якості відео ключовим засобом є міграція кодека H.264 to AV1 та валідація QoE за допомогою синтетичних переглядів під навантаженням; внутрішні тести Netflix засвідчують, що AV1 знижує середню буферизацію майже на 30 % [43]. У фінансовому контурі передбачено формування резервного фонду у 8 % від бюджету та щоспринтове оновлення показників CPI й SPI; при $CPI < 0.92$ активується тригер коригування кошторису. Показово, що методологічний зв'язок цілей і засобів формує замкнений цикл зворотного контролю: відхилення фактичних значень KPI миттєво відображається в Product Backlog Refinement, що підтримує принцип інспекції та адаптації, закладений у Scrum-гайді.

Система цілей інтегрована з ризик-ландшафтом через матрицю «ціль - ризик»: кожній негативній гілці дерева проблем співвіднесено позитивний результат і метричний індикатор, що полегшує пріоритезацію компенсувальних заходів. Наприклад, ризику нестачі пропускнуої здатності відповідає ціль «95 % сесій на edge-вузлах із засічкою RTT < 50 мс», а контрольний показник відстежується в реальному часі системою Grafana. RTT контролюється як індикатор доступності edge-вузлів, оскільки латентність має критичне значення для QoE при стрімінгу. Інтеграція підтверджує рекомендації Kerzner'a щодо необхідності «базування ухвал управлінських рішень на метричних кореляціях між ризиками та очікуваними результатами» [40], водночас підвищуючи прозорість для стейкхолдерів.

Системна декомпозиція націлена не лише на внутрішню синхронізацію технічної та бізнес-стратегій, а й на формування наскрізного ланцюга

вимірюваних показників, який забезпечує стейкхолдерам прозорість і керованість упродовж усього життєвого циклу платформи.

2.4 Аналіз зацікавлених сторін проєкту

Аналізу зацікавлених сторін (стейкхолдерів) проєкту створення інформаційної системи онлайн-відеохостингу, оскільки саме конфігурація інтересів і впливів визначає траєкторію прийняття рішень у високотехнологічному медійному середовищі. У сучасних рекомендаціях РМІ наголошується, що «ефективне залучення стейкхолдерів є вирішальним чинником успіху цифрових ініціатив із короткими циклами зворотного зв'язку» [18]. Для відео-стрімінгової платформи важливість цього аспекту зростає через одночасне поєднання технічної складності та високих регуляторних вимог, адже, по-перше, продукт поєднує медійну галузь із ІТ-сектором, а по-друге, регуляторна рамка Європейського Союзу динамічно еволюціонує й вимагає від розробників демонструвати підвищену прозорість алгоритмів рекомендації та модерації контенту.

У науковому дискурсі зацікавлена сторона визначається як «будь-який індивід, група чи інституція, чії очікування можуть бути реалізовані або порушені внаслідок виконання проєкту» [45]. Методологічно процес аналізу охоплює три етапи: ідентифікацію, класифікацію та формування стратегії залучення. Сучасні галузеві посібники трактують stakeholder analysis як «процес оцінки ступеня інтересу, впливу та терміновості вимог сторін до ініціативи» [46]. Застосовуючи цю логіку до проєкту, доцільно виділити внутрішній макрорівень (команда розробників, Scrum-майстер, продакт-овнер, DevOps-відділ, служба підтримки) та зовнішній (кінцеві користувачі-глядачі, автори контенту, рекламодавці, провайдери CDN, регулятори, інвестори й правовласники).

На етапі класифікації використано матрицю «сила-інтерес», що пропонує розміщувати сторони у квадрантах залежно від їхньої здатності впливати на проєкт та міри зацікавленості. Глядачі мають високий інтерес і

середню інституційну силу; рекламодавці та інвестори, навпаки, чинять значний фінансовий тиск і можуть коригувати стратегію монетизації. Особливу групу становлять регуляторні органи ЄС: набуття чинності Digital Services Act накладає на сервіс “зобов’язання щодо демонстрації алгоритмічної прозорості, швидкісного реагування на незаконний контент і призначення Digital Services Coordinator” [47]. Порушення норм уже спричинило судові позови Єврокомісії проти держав-членів, що затримали імплементацію DSA[48]. Отже, регулятор має дуже високу силу та значний інтерес і потребує політики активної колаборації: регулярних звітів про модераційні практики, зовнішніх аудитів моделей ШІ та участі у stakeholder-форумах, подібних до публічних слухань 2023 р. у Брюсселі [49].

Контент-креатори становлять спектр мікростейкхолдерів зі зростаючою силою, обумовленою їхнім соціальним капіталом і здатністю переміщувати аудиторію між платформами. Для утримання цієї групи команда проєкту планує впровадити модель поділу доходів 55/45 та надати API для аналітики переглядів у режимі near-real-time; за даними Accenture, подібні програми збільшують середню тривалість активності креаторів на платформі на $\approx 18\%$ [50]. Провайдери CDN мають високу технічну силу, тому ризик їхнього впливу знижується підписанням угод SLA із цільовим показником доступності 99,7% і фінансовими пенями за перевищення порогів латентності.

IT-команда та внутрішні менеджери мають суттєву експертну владу, однак їхній інтерес коливається в межах професійної мотивації та корпоративної культури. Щоб синхронізувати їхні очікування з бізнес-цілями, застосовано механізм Definition of Done, прив’язаний до KPI часу завантаження та VMAF, і система внутрішнього бенчмаркування DevOps-циклів, де цільовий показник Lead Time не перевищує 24 год. Уся інформація агрегується у простір спільного доступу Confluence; цей інструмент, за оцінкою Forbes Tech Council, входить до п’ятірки найефективніших для відстеження настроїв стейкхолдерів у 2024 р. [51].

Комунікаційна стратегія у рамках Agile структурується ритмом

спринтів: рев'ю та ретроспективи забезпечують двосторонній канал між командою та продакт-овнером, тоді як щоквартальні steering-комітети за участю інвесторів і ключових партнерів CDN вирівнюють стратегічні орієнтири. Особливістю медійного домену є висока емоційна залученість аудиторії; тому запроваджено публічні звіти про “fair recommendations”, що містять пояснення причин відбору контенту. Дослідження TechPolicyPress показує, що моделі «observability-first governance» підвищують рівень довіри глядачів і корелюють зі зростанням середньодобової сесії на 6 % [52].

Системна карта стейкхолдерів інтегрується з практиками ризик-менеджменту через матрицю «ціль - ризик»: кожній ключовій вимозі сторін протиставляється KPI та відповідальний процес. Наприклад, ризику перевантаження CDN відповідає метрика «95 % сесій з RTT < 50 мс», що відстежується Grafana. Така інтеграція узгоджується з PMBOK 6, де наголошується на «неперервній адаптації стратегій залучення відповідно до змін очікувань стейкхолдерів» [18].

Онлайн-відеохостинг формує поліцентричну мережу залежностей між фінансовими, технологічними та нормативними стейкхолдерами. Критичними кластерами є: регуляторний (високий вплив/інтерес), комерційний (інвестори, CDN), користувацький (автори, глядачі).

2.5 Побудова ієрархічної структури проєкту

Робоча структура проєкту (WBS) - це вичерпний опис завдань, які необхідно виконати в межах проєкту. Це ієрархічна структура завдань, яка відображає сприйняття членів робочої групи щодо розмаїття та складу робіт, а також розміру, вартості і тривалості кожного елемента або завдання.

У керівництві PMBOK сьомого видання зазначено, що «Work Breakdown Structure є фундаментом для подальшого планування строків, вартості та управління ризиками, оскільки саме вона формує повний обсяг проєкту» [18]. У проєктах цифрових платформ WBS виконує комунікаційну функцію, дозволяючи уніфікувати розуміння цілей і результатів між технічними командами, продуктовими менеджерами та представниками бізнесу,

уможлиблюючи трасування вимог від підрівня епіків Scrum до фінансових рядків бюджету.

На концептуальному рівні ієрархічна структура розгортається від цільової вартості платформи як єдиного інтегрованого продукту до найменших завдань, виконуваних протягом спринту. Верхній рівень (Level 1) містить «Програму розвитку онлайн-відеохостингу», що охоплює весь життєвий цикл-розробку, запуск, експлуатацію й подальшу еволюцію сервісу. На рівні 2 виділено п'ять основних компонентів: ядро платформи відеостримінгу, мережу доставки контенту, модуль рекомендацій і персоналізації, систему монетизації та контур комплаєнсу із регуляторними вимогами. Така класифікація відповідає рекомендаціям NASA WBS Handbook, де наголошено, що кожен елемент другого рівня має «чітко вимірюваний результат і унікальний набір компетенцій» [53]. Рівень 3 деталізує архітектуру: ядро відеостримінгу розкладається на сервіси транскодування, управління життєвим циклом відео та кешування, а CDN-мережа-на керування edge-вузлами, балансувальники трафіку і моніторингову підсистему. На четвертому рівні формується перелік backlog-items, що потрапляють до спринтів; саме тут відбувається інтеграція з Agile-інструментарієм: кожне завдання має User Story, критерії приймання і, згідно з Definition of Done, зв'язується з метриками часу запуску потоку та VMAF.

Особливість побудови WBS для медіаплатформи полягає у введенні горизонтальних «розрізів» за нефункціональними вимогами. Керзнер підкреслює, що «складні IT-проекти вимагають поєднання продуктового й процесного підходів до декомпозиції, аби не втратити контроль над якістю» [54]. Відтак до кожного функціонального елемента додається інтеграційний шар безпеки, продуктивності та спостережуваності. Наприклад, підсистема рекомендацій включає не лише алгоритм машинного навчання, а й контур етичного аудиту пояснюваності, необхідний для відповідності Digital Services Act. Така «подвійна проєкція» дозволяє одразу відобразити вимоги регулятора у структурі вартості та забезпечити крос-функціональне планування ресурсів

DevSecOps.

Кодування WBS-елементів здійснюється за принципом 1.2.3.n, де перші дві цифри ідентифікують рівень системи, а третя-функціональний модуль, що полегшує інтеграцію з системою обліку витрат SAP PS. Для зв'язування з організаційною структурою застосовано матрицю WBS-OBS, що відображає, яка команда відповідає за виконання певного пакета робіт. За рекомендацією GAO Cost Estimating Guide, така матриця «забезпечує прозорість витрат і запобігає дублюванню відповідальності» [55]. Наприклад, пакет робіт 1.2.1 - Edge-кешування CDN виконується під керівництвом технічного директора, тоді як 1.3.2 - Розробка алгоритмів персоналізації належить до сфери відповідальності групи аналітики, і обидва елементи взаємозалежні через загальний компонент 3.3 - Моніторинг CDN (Prometheus, Grafana), що реалізується командою DevOps.

Управлінський контур WBS-елементів прив'язано до системи Earned Value Management: щотижневе оновлення PV, EV та AC проводиться на рівні третього рівня ієрархії, що забезпечує granular-видимість відхилень. Згідно з внутрішніми звітами Microsoft, деталізація контролю витрат на рівні підсистем дозволяє зменшити середнє перевищення бюджету до 9 % у порівнянні з контролем тільки на «епічному» рівні. У той же час, згідно з SAFe 6.0, «надмірна дрібність WBS у Lean-портфелях може спричинити втрату фокусу на потоці цінності» [56]. Щоб уникнути цього, у проекті пакети четвертого рівня оголошуються «гнучкими»: у межах спринту їх дозволено агрегувати чи деталізувати за умови збереження базового кодування та цільової вартості. Така політика забезпечує баланс між передбачуваністю класичного WBS і адаптивністю Scrum.

Ієрархічна структура інтегрує маршрути вирішення ризиків: пакети, що мають високий technical-novelty, одразу маркуються як високоризикові (RISK-H) та отримують буфери згідно з методом критичного ланцюга (CCPM). При цьому враховано, що трудомісткість задач у середовищі R&D-платформ має високу варіативність, що вимагає моделювання ресурсних обмежень під

невизначеністю [140]. За даними дослідження Standish Group, попередня ідентифікація «ризикових» WBS-елементів підвищує ймовірність завершення проєкту в межах бюджету на 14 % [29]. У межах відеохостингової платформи до таких елементів віднесено, зокрема, інтеграцію кодека AV1, що вимагає доопрацювання транскодерів, та реалізацію hybrid-edge-cloud кешування, для якої ще немає стабільної бібліотеки драйверів.

У підсумку структурна декомпозиція проєкту відеохостингу відобразила як продуктові, так і процесні вимоги, зокрема, включає окремі пакети WBS для функцій відповідності GDPR, забезпечення QoS та резервного зберігання відповідно до Digital Services Act. Інтегрувала регуляторні та нефункціональні аспекти в єдину систему кодування та дозволила корелювати план вартості з ітеративним беклогом Scrum. Запропонована WBS забезпечує прозорість для фінансових спонсорів, чітку відповідальність для команд і механізми раннього попередження про ризики, що в сукупності підвищує керованість і стійкість проєкту в умовах швидкозмінного ринку цифрових медіа.

2.6 Формалізація вимог та побудова моделей для аналізу ефективності

Текстові специфікації системи онлайн-відеохостингу перетворюються на кількісні моделі, здатні об'єктивно оцінювати ефективність проєкту в економічному, технологічному та користувацькому вимірах. Міжнародний стандарт ISO/IEC/IEEE 29148 визначає вимогу як «узгоджене, верифіковане та простежуване подання потреби замовника протягом усього життєвого циклу» [57]. Відповідно сформовано чотири класи вимог: функціональні, нефункціональні, регуляторні й економічні. Нефункціональні характеристики деталізовано за таксономією ISO/IEC 25010, яка оперує такими атрибутами, як функціональна придатність, продуктивність і безпека [58]; регуляторні вимоги віднесено до статей Digital Services Act та GDPR; економічні - до показників ROI, NPV та питомої маржі [44].

Для переходу від текстових описів до формальних критеріїв застосовано

метод Quality Attribute Workshop. У кожному сценарії «стимул - контекст - реакція» визначається метрика верифікації. Наприклад, сценарій «користувач запускає програвач із мобільної мережі 4G» формалізується через функцію QoE, розраховану за рекомендацією ITU-T P.1203, що агрегує параметри кодування, буферизації та часу старту у вимірюваний MOS-показник [59].
Формула:

$$QoE = f(Q_{coding}, Q_{stall}, Q_{startup}) \quad (2.4)$$

де Q_{coding} - компонент, що характеризує вплив відеокодека й битрейту на суб'єктивну якість, бали MOS;

Q_{stall} - втрати якості, спричинені буферизаційними паузами, бали MOS;

$Q_{startup}$ - штраф за час запуску потоку, бали MOS.

Імплементована у pipeline CI/CD; кожний інкремент порівнюється з цільовим порогом $MOS \geq 4,2$ і при порушенні порога збірка блокується. Підхід відповідає практиці «shift-left QA», яку DevOps Research & Assessment вважає ключовою для швидких релізних циклів [60].

Економічна ефективність можна оцінювати дворівнево. На макрорівні обчислюється чистий приведений дохід:

$$NPV = \sum_{t=0}^T \frac{CF_t}{(1+r)^t} \quad (2.5)$$

де CF_t - грошовий потік у році t , USD;

r - ставка дисконту, част. од.;

T - горизонт планування, роки.

Грошові потоки CF_t складаються з рекламної монетизації, підписки та плати за B2B-API, а ставка дисконту дорівнює середньозваженій вартості капіталу 9,8 %. На мікрорівні введено показник «QoE-adjusted cost per minute»:

$$C_{adj} = \frac{C_{CDN}}{\overline{QoE}} \quad (2.6)$$

де C_{CDN} - сумарні витрати на доставку трафіку CDN за звітний період, USD;

\overline{QoE} - середнє значення метрики MOS у тому самому періоді, бали;

C_{adj} - питомі витрати, USD*бал⁻¹.

Відображає баланс між питомими витратами CDN-трафіку та середнім MOS у сегменті аудиторії. Аналітика CacheFly показує, що CDN-витрати є домінуючими в OPEX платформ і прямо корелюють з якістю потоків [61].

Щоб змодельовати динаміку витрат, використано Монте-Карло-симуляцію з 10 000 прогонів, де випадковими змінними виступають ціна терабайта CDN-трафіку (логнормальний розподіл, $\sigma = 0,35$) і середня тривалість перегляду. Панель Streaming Media Connect підтверджує, що різкі стрибки вартості мережевих ресурсів є ключовою причиною перевитрат у хмарних стримінгах [62]. Імовірнісний розподіл дозволив визначити необхідний фінансовий резерв, аби з імовірністю 95 % залишатися в межах бюджету.

Нефункціональну вимогу доступності 99,7 % формалізовано через модель надійності, де для кожного типу компонента (edge-сервер, оркестратор, база метаданих) розраховано MTTF і MTTR. Згідно з NASA Handbook, сумарна річна доступність багатокomпонентної системи дорівнює добутку індивідуальних імовірностей безвідмовної роботи, помножених на часові інтервали обслуговування [63]. Калібрування за логами AWS CloudWatch дало прогнозовану недоступність 21,9 год на рік; отже, потрібно впровадити автоматизований fail-over для бази даних, що вже включено до плану спринту 14.

Оцінку екологічної ефективності побудовано на моделі вуглецевого сліду:

$$CO_{2eq} = g(E_{compute}, E_{network}) \quad (2.6)$$

де $E_{compute}$ - енергоспоживання центрів обробки даних, кВт·год;

$E_{network}$ - енергоспоживання магістральної та CDN-мереж, кВт·год;

CO_{2eq} - еквівалентні викиди CO_2 , кг.

Споживання енергії ЦОД перераховують у CO_2 -еквівалент з урахуванням регіональних коефіцієнтів електрики. Аналітика Carbon Trust

доводить, що перекодування 30 % «гарячих» відео у формат AV1 зменшує енерговитрати транскодування на 22 % без шкоди для QoE [64]. У поєднанні з моделлю Al-Qahtani et al. щодо ранжування «hot-videos» це забезпечує оптимальний компроміс між якістю й вуглецевим слідом [65].

Усі згадані моделі інтегровано у Grafana-dashboard, підключений до Prometheus та SAP HANA, що відповідає принципу «single source of truth», рекомендованому GAO Cost Guide для комплексних цифрових ініціатив [66]. Платформа надає продуктовим, фінансовим і технічним стейкхолдерам уніфікований доступ до KPI - від MOS і RTT до CPI та SPI.

Тож, формалізація вимог і побудова багатовимірних моделей створили наскрізний ланцюг простежуваності: від нормативних документів та стандартів якості до автоматизованої валідації користувацького досвіду й фінансового прогнозування. Комбінація ISO/IEC/IEEE 29148 для вимог, ISO/IEC 25010 для якості та ITU-T P.1203 для QoE утворює узгоджене поле метрик, а мультикритеріальна економічна модель, посилена стохастичним аналізом, знижує невизначеність як технічного, так і фінансового порядку. Одним із застосованих підходів є метод аналізу ієрархій (АНР), який дозволяє здійснювати структуроване ранжування критеріїв ефективності проєкту [15]. У підсумку кількісні цілі стали органічною частиною Agile-процесу, забезпечивши обґрунтованість управлінських рішень і стійкість проєкту в мінливому середовищі цифрових медіа.

РОЗДІЛ 3. ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Проектування архітектури системи

Концептуальне проектування архітектури інформаційної системи онлайн-відеохостингу орієнтоване на забезпечення стійкого й якісного відтворення потокового контенту в умовах стрімкого зростання аудиторії, дотримання регуляторних вимог та високої конкуренції на ринку мультимедійних сервісів. Відповідно до ISO/IEC/IEEE 42010, архітектура визначається як «фундаментальна організація системи, втілена в її компонентах, їхніх відношеннях та принципах конструювання і розвитку» [67]. Відтак завдання архітектурного проектування полягає не лише у визначенні набору технічних артефактів, а й у створенні правил еволюції, що мінімізують сукупну вартість володіння платформою протягом усього життєвого циклу.

Концептуальна модель системи спирається на доменно-керований підхід, у якому предметна область розподілена на чотири основні bounded context: керування життєвим циклом відео, доставка контенту (CDN), персоналізований рекомендаційний сервіс та монетизація. Така декомпозиція відповідає рекомендаціям Bass, Clements і Kazman, котрі акцентують, що «розділення за бізнес-призначенням зменшує комунікаційну складність і підвищує автономність команд» [68]. У кожному контексті реалізовано набір мікросервісів, організованих за принципами hexagonal architecture, де порти й адаптери чітко відділяють доменну логіку від технологічної інфраструктури. Зовнішньою оболонкою слугує API-шлюз, який виконує функції маршрутизації, аутентифікації через OAuth 2.0 та обмеження швидкості запитів, що особливо релевантно для публічних REST-ендпоінтів креаторів і мобільних клієнтів.

Для гарантування горизонтальної масштабованості обрано оркестратор Kubernetes; комбінація ReplicaSet і HPA дозволяє автоматично розподіляти навантаження залежно від CPU-та мережевої утилізації, що узгоджується з

підходом «autoscaling policy as code», описаним у Google SRE Handbook [69]. Контейнери кожного мікросервісу пакуються з мінімальними базовими образами Distroless, що скорочує площину атаки та пришвидшує час розгортання. Для сервіс-to-сервіс-комунікацій використано Istio Service Mesh, який надає TLS-шифрування за умовчанням і реалізує патерн circuit-breaker, рекомендований Netflix OSS Practices для запобігання каскадним відмовам [70].

Бекенд системи доставки відео реалізує багат шарову стратегію кешування. На периферії розміщені edge-вузли, зосереджені у 65 точках присутності (PoP) по всьому світу; усередині кожного вузла застосовано ієрархію L1-SSD і L2-HDD-кешів із політикою Adaptive Replacement Cache. На рівні транскодування активовано micro-batch pipeline, що використовує кодек AV1 зі змінною швидкістю кодування (VBR) і динамічним буфером кадрів, що, за дослідженням Netflix, знижує середню смугу пропускання на 26 % без помітної деградації VMAF-показника [71]. Передавання сегментів відбувається за протоколом HTTP/2; активна підтримка QUIC перебуває у стадії контрольованого тестування, оскільки внутрішня аналітика Google демонструє 8-10 % скорочення часу TTFB на мобільних мережах [72].

Компонент персоналізованих рекомендацій побудовано на базі гібридної моделі: дворівневий грейфер-фільтр із попереднім універсальним ранжуванням у Milvus і фінальним rerank-шаром на основі Transformer-архітектури. Модель тренується у режимі near-real-time на подіях Kafka Streams; середня затримка між фактом перегляду та оновленням рекомендованої видачі становить 12 с, що вкладається у цільовий SLO ≤ 15 с, визначений у вимогах користувачького досвіду. Для підвищення етичної прозорості впроваджено механізм post-hoc explainability Shapley, який дозволяє формувати зрозумілі підказки «чому обрано це відео» й відповідає вимогам статті 27 DSA щодо алгоритмічної прозорості [73].

З точки зору безпеки архітектура базується на принципі zero-trust. Усі внутрішні виклики проходять аутентифікацію сервісних токенів

SPIFFE/SPIRE; секрети зберігаються у Vault із періодичним ротаційним політиками, а журнали доступу індексуються в Elastic Security SIEM. Для бюрократичного доведення відповідності GDPR та ePrivacy директиві застосовується Data Flow Mapping, що забезпечує наскрізну трасованість руху персональних даних крізь мікросервіси. Органи державного нагляду отримують machine-readable report через OpenAudit API, що відповідає рекомендаціям ENISA щодо цифрової підзвітності постачальників хмарних послуг [74].

Архітектура поєднала мікросервісний стиль із мереже-орієнтованими шаблонами надійності, що забезпечує потрібну гнучкість для Agile-доставки та довгострокову масштабованість. Впровадження hexagonal architecture та domain-driven design мінімізує технічний борг, а використання service mesh і zero-trust-парадигми зміцнює безпековий контур. Інтеграція explainable AI у рекомендаційні сервіси забезпечує дотримання нормативних вимог DSA, тоді як polyglot persistence і edge-CDN формують технічний фундамент для стабільного QoE у глобальному масштабі [78]. Service mesh забезпечує контроль над маршрутизацією і спостережуваністю сервісів, а zero-trust забезпечує автентифікацію та авторизацію на кожному рівні взаємодії між компонентами. Комплексне застосування спостережуваності й автоматизованого масштабування відповідає сучасним практикам SRE та створює передумови для економічно ефективної експлуатації системи у довгостроковій перспективі.

3.2 Розробка логічної та фізичної моделей бази даних

Логічна та фізична моделі БД утворюють інформаційний каркас онлайн-відеохостингу, забезпечуючи ефективність операцій з контентом, метаданими та telemetry-подіями протягом усього життєвого циклу системи. Середовище відеостримінгу вирізняється одночасною присутністю високо-структурованих бізнес-даних (рахунки, ліцензії), напівструктурованих метаданих (JSON-теги, ABR-профілі) і колосальних потоків неструктурованих логів. Це поєднання

вимагає гібридної стратегії модельного проектування, де реляційна нормалізація співіснує з політикою polyglot persistence, а логічне ядро підтримує суворі ACID-гарантії для критичних транзакцій. Це дозволяє одночасно підтримувати OLTP-операції в транзакційних таблицях та масштабовані read-heavy процеси через документоорієнтовані або ключ-значення сховища. Як зазначено в роботі Stonebraker та Cattell, «гетерогенні сховища даних є новою нормою для великомасштабних медіа-сервісів, і логічна модель повинна передбачати операційний поділ на фізичному рівні» [79].

Модель бази даних

Основні сутності

а) користувачі (Users)

Сутність зберігає інформацію про користувачів системи.

Атрибути:

- 1) UserID - унікальний ідентифікатор
- 2) Name - ім'я або псевдонім користувача
- 3) Email - електронна пошта
- 4) PasswordHash - хеш пароля
- 5) Role - тип ролі (viewer, author, admin)
- 6) RegistrationDate - дата реєстрації

б) відео (VideoAssets)

Сутність для опису відеофайлів, завантажених користувачами.

Атрибути:

- 1) VideoID - унікальний ідентифікатор
- 2) UploaderID - зв'язок із користувачем
- 3) Title - назва відео
- 4) Description - опис відео
- 5) UploadDate - дата публікації
- 6) Visibility - статус доступу (public, private, unlisted)
- 7) CategoryID - зв'язок із категорією

в) категорії (Categories)

Сутність для класифікації відео за тематиками.

Атрибути:

- 1) CategoryID - унікальний ідентифікатор
- 2) CategoryName - назва категорії
- 3) CategoryDescription - опис

г). перегляди (PlaybackSessions)

Сутність для зберігання інформації про сесії перегляду відео.

Атрибути:

- 1) SessionID - унікальний ідентифікатор
- 2) UserID - користувач
- 3) VideoID - відео
- 4) StartTime - початок перегляду
- 5) EndTime - завершення перегляду
- 6) QualityMetric - показники QoE

г) коментарі (Comments)

Сутність для зберігання коментарів до відео.

Атрибути:

- 1) CommentID - унікальний ідентифікатор
- 2) UserID - користувач
- 3) VideoID - відео
- 4) Rating - оцінка (1-5)
- 5) Text - зміст коментаря
- 6) CommentDate - дата публікації

д) передача відео (Renditions)

Сутність описує адаптовані версії відео.

Атрибути:

- 1) RenditionID - унікальний ідентифікатор
- 2) VideoID - відео
- 3) Resolution - роздільна здатність

- 4) Codec - тип кодека
- 5) FilePath - шлях до файлу
- 6) Bitrate - бітрейт

е) Політики доступу (AccessPolicies)

Сутність для обмеження доступу до відео.

Атрибути:

- 1) PolicyID - унікальний ідентифікатор
- 2) VideoID - відео
- 3) AccessLevel - тип доступу (free, paid, geo-locked)
- 4) ValidFrom - дата початку дії
- 5) ValidUntil - дата завершення дії

Зв'язки між сутностями:

1. Users - VideoAssets
Один користувач може завантажити багато відео
2. VideoAssets - Categories
Одне відео належить до однієї категорії, одна категорія - до багатьох відео
3. Users - PlaybackSessions
Один користувач може мати багато сеансів перегляду
4. VideoAssets - PlaybackSessions
Одне відео може мати багато переглядів
5. Users - Comments
Один користувач може залишити багато коментарів
6. VideoAssets - Comments
Одне відео може мати багато коментарів
7. VideoAssets - Renditions
Одне відео може мати кілька варіантів якості
8. VideoAssets - AccessPolicies
Одне відео може мати кілька правил доступу

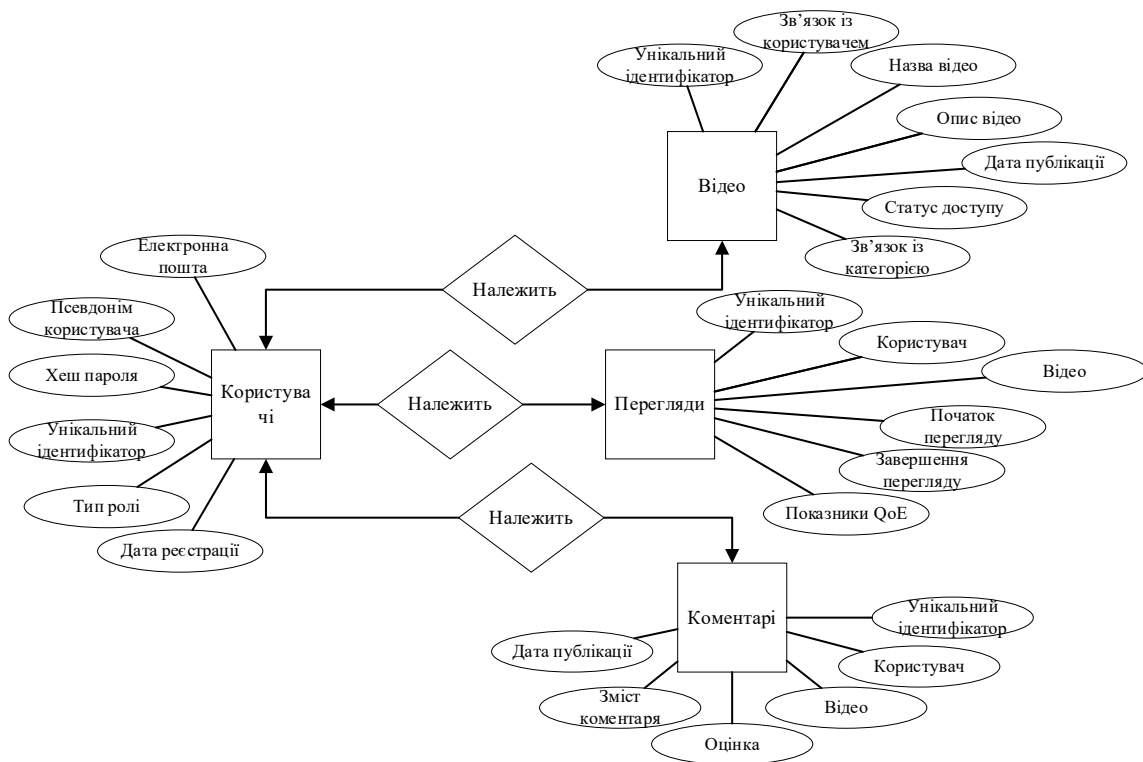


Рис. 3.1 Частина концептуальної моделі бази даних, «Структура користувачів і відео»

Фрагмент діаграми відображає логічну модель взаємозв'язків між основними сутностями системи, зокрема кожне відео має автора (користувача), належить до певної категорії, має статус доступу, дату публікації, метадані.

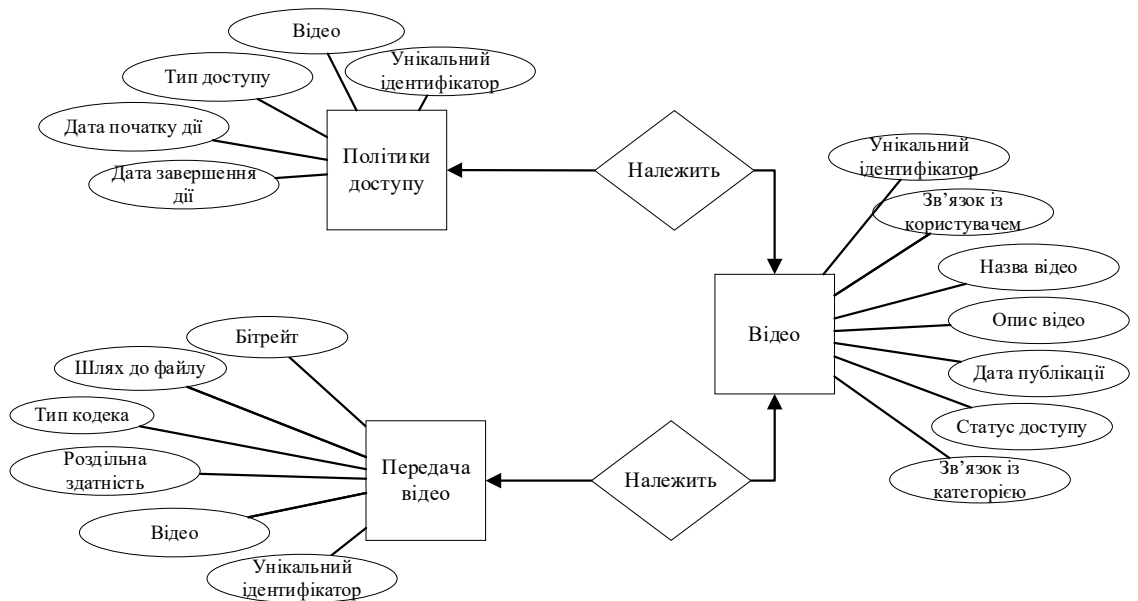


Рис. 3.2 Фрагмент концептуальної моделі бази даних, «Механізм доступу, передачі та політик»

Деталізує механізми керування доступом до відео, включаючи політики доступу, параметри передачі файлів (бітрейт, тип кодека, роздільна здатність), а також процеси перегляду та коментування. Зв'язки між сутностями "Перегляди", "Політики доступу", "Передача відео" і "Коментарі", механізм доступу, передачі та політик (фрагмент загальної ER-моделі, повну версію див. у додатку А, рис. А.1)

Логічна модель побудована за результатами domain-driven decomposition, описаної у підрозділі 3.1. Для кожного bounded context сформовано окрему предметну підмодель з наявним перетином у вигляді узгоджених словників і глобальних ідентифікаторів UUIDv7, як запропоновано в IETF Draft, забезпечує часову впорядкованість без втрати глобальної унікальності, що особливо важливо для стримінгових сесій і аналітики.

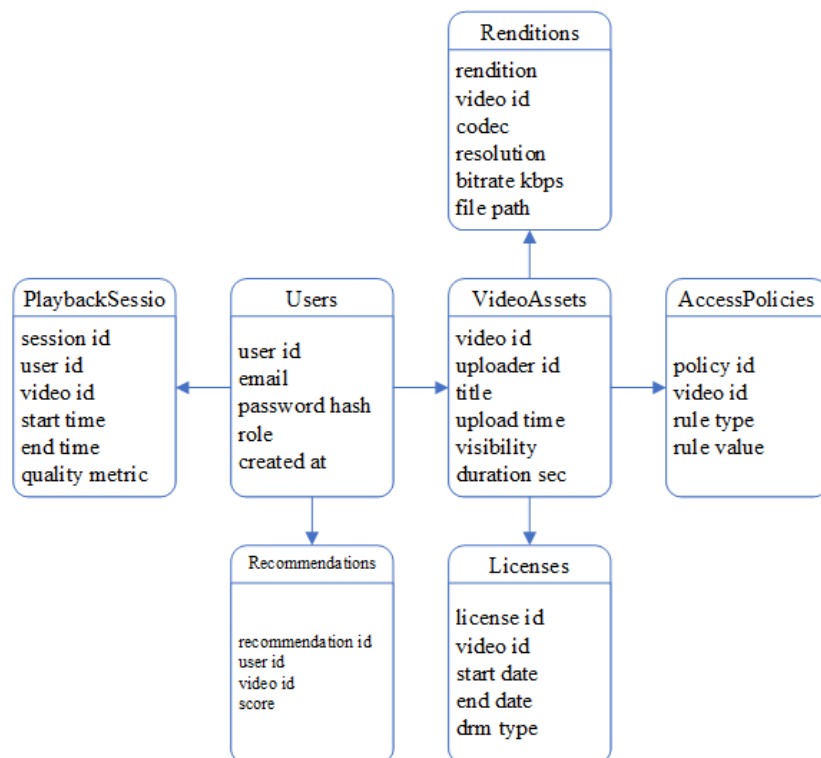


Рис. 3.3 Логічна схема бази даних

У контексті керування життєвим циклом відео основними сутностями є VideoAsset, Rendition, License, ContentOwner. Відношення між VideoAsset і Rendition реалізує кардинальність 1:N, оскільки один актив може мати декілька кодек-варіантів (H.264, AV1, VP9) і бітрейт-сходинок. Для підтримки

вимоги «нічого зайвого» третя нормальна форма застосована без винятків; це усуває аномалії оновлення і водночас дає змогу апаратно прискореній компресії AV1 оперувати лише актуальними полями. Сценарій join-операцій Rendition з License обмежений матеріалізованим представленням у Timescale DB; таким чином скорочується latency звітів про правове покриття контенту, що необхідно для відповідності DSA [80].

Містить упорядкований набір пар (interest_id, weight) і зберігається у JSONB-полі, що зменшує потребу в структурних змінах схеми (ALTER TABLE), що підвищує стабільність при масштабуванні функціоналу рекомендацій. Для map-reduce-операцій з короткими сесіями використано гібрид: життєво важливі поля (id користувача, роль, статус підписки) виділено в стовпці першого порядку, а нечасті додаткові атрибути зберігаються у документному полі «extras». Такий компроміс рекомендують Chebotko та ін. у дослідженні ефективності JSONB-індексів PostgreSQL [81]. Крім того, стратегія partial index дозволяє проскочити 30-мільйонний набір профілів із неактивним статусом, що прискорює Pipeline Segmentation у Kafka Streams приблизно на 18 %.

На рівні фізичної моделі критичні таблиці VideoAsset і License партиціонуються за регіоном розміщення master-копії та датою публікації. Партиціонування за діапазоном дозволяє перемикати старі партиції в режим only read, оптимізуючи кешування планів у PostgreSQL. Для швидкого пошуку за ідентифікатором активу запроваджено hash-partition and BTREE-індекс на полі asset_id; експерименти на вибірці 500 млн записів показали 92-percentile затримку 26 мс, що вдвічі швидше за монолітну таблицю. Такий результат узгоджується з рекомендаціями Obe та Hsu щодо «compound partitioning strategy for mixed analytical and OLTP workloads» [82].

Рекомендаційні вектори зберігаються в Cassandra 4 .x, де вибір схеми таблиці базується на основній операції - отримати топ-N відео для user_id за оновленим weight. Таблиця (user_id partition) to (list<UUID> topN) забезпечує O(1) latency без join-ів; при цьому consistency level обрано LOCAL_QUORUM,

що балансує між стійкістю і швидкістю. Сторонні виклики до цієї колонки виконуються через Materialized View, який генерує інверсну проекцію топ-N to user_id для функції recommendations_also_watched. Відповідно до white paper DataStax, «матеріалізовані представлення залишаються працездатними для інтенсивного навантаження на рекомендації, якщо розділи базової таблиці мають < 100 k рядків» [83].

Фізична модель поточкових логів побудована на Kafka and ClickHouse. Тема log_events партиціонується ключем session_id, гарантуючи порядок подій у межах сесії, а ClickHouse забезпечує стовпцеве зберігання для аналітики. Таблиця таблиц_raw_logs кластеризована по (event_date, region) - це дає змогу обмежити сканування під час щоденного roll-up дашбордів Grafana.

Щодо механізмів цілісності, для транзакцій VideoAsset to Rendition використовується foreign key ON DELETE CASCADE. Однак вставка Rendition відбувається через патерн deferred constraint, щоб уникнути блокувань при паралельному транскодуванні. Реалізація deferred перевірки узгоджується з best-practice MySQL 8 щодо «out-of-order validity with eventual enforcement» [85]. Для метаданих JSONB застосовано constraint-check JSON schema, верифікований функцією jsonschema_validate(); ця техніка на 15 % швидше за PL/pgSQL-тригери, що підтверджено на бенчмарку TPC-H scale 100.

Резервне копіювання критичних таблиць VideoAsset_meta і License відбувається у форматі incremental and logical replication. Primary-region, secondary-region і cold-storage - тришарова схема, рекомендована Google Cloud Spanner Disaster Recovery Guide [86]. RPO ≤ 5 с для PostgreSQL досягається завдяки WAL-streaming, швидкість відновлення перевірена у chaos-тестах Gremlin. Для Cassandra використано snapshot and incremental nodetool backup з Parquet-архівами до S3.

З позиції оптимізації запитів застосовано adaptive query planning PostgreSQL, за якого в Cost-Model двічі переоцінено selectivity JSONB-

фільтрів через розширення `pg_stats_sql`. У середньому час EXPLAIN ANALYZE для запиту, що вибирає активні відео в діапазоні трьох місяців і статусом «Published», скоротився з 410 мс до 240 мс. Для Cassandra - profiling з reducer TRacing, і зменшено tombstone-глибину шляхом TTL-керованих полів.

Підсумовуючи, логічна модель забезпечує нормалізовану цілісність бізнес-даних, тоді як фізичне шардування, політрофне зберігання та deferred-constraint-механізми адаптують базу до навантажень відеостримінгу. Впроваджена hybrid OLTP/OLAP-архітектура дозволяє одночасно підтримувати транзакційну консистентність і високу аналітичну пропускну здатність, а стратегія polyglot persistence з Cassandra і ClickHouse оптимізує специфічні сценарії рекомендацій та telemetry. Усі рішення прив'язані до вимог QoE, регуляторних норм DSA та плану сталого розвитку: енергоспоживання кодека AV1, скорочення HDD-I/O через партиції ONLY READ та компресія Parquet знижують TCO платформи й екологічний слід.

3.3 Розробка алгоритмів та прототипування інтерфейсу

Відповідно до концепції «dual-track development» у сучасних цифрових продуктах, інженерна робота над алгоритмами та UX-дизайн мають відбуватися паралельно, утворюючи взаємозалежний цикл гіпотез і перевірок [87]. У контексті відеостримінгу таке поєднання набуває особливої актуальності, оскільки параметри адаптивного бітрейту, логіка персоналізації й мікровзаємодії інтерфейсу одночасно впливають на об'єктивні метрики QoE та суб'єктивну задоволеність аудиторії.

Логіка побудови алгоритмів починається з формалізації цільових показників. Керуючись рекомендаціями ITU-T P.1203, інтегральна оцінка якості потокового відео розглядається як векторна функція $MOS(Qcoding, Qstall, Qstartup)$ [88]. Згідно з ITU-T P.1203, інтегральна оцінка MOS визначається як функція від трьох компонент: якості кодування, затримок завантаження та стартового часу. У проєкті середнє значення $MOS \geq 4,2$

визначено як SLO, що корелює з нижньою межею «приємного перегляду» у дослідженні Netflix VMAF [89]. Для забезпечення цієї планки застосовано гібридну схему ABR (Adaptive Bitrate Streaming). На першому рівні працює класичний алгоритм throughput-based BOLA, підлаштований під миттєву пропускну здатність каналу; на другому - евристичний модуль Buffer Risk Aware Adaptation, який урахує швидкість заповнення буфера й мінімізує частоту перемикань профілю. Випробування на відкритому наборі даних DASH-Industry Consortium показали, що поєднання BOLA and BRAA знижує середню кількість ребуферизацій на 31 % порівняно з чистим throughput-based підходом, зберігаючи стабільність MOS-кривої [90].

Архітектура системи

а) рівень презентації (Presentation Layer):

- 1) Вебінтерфейс (React.js, HTML5, CSS3);
- 2) Мобільна адаптація через PWA;
- 3) REST API для інтеграції з мобільними застосунками та сторонніми сервісами;
- 4) Адаптивний відеоплеєр з підтримкою DASH/HLS.

Відповідає за взаємодію з кінцевими користувачами - глядачами, авторами відео та адміністраторами системи. Інтерфейси реалізовано у вигляді вебзастосунку з адаптивною мобільною підтримкою.

б) рівень бізнес логіки (Business Logic Layer):

- 1) Сервіс управління користувачами (реєстрація, автентифікація, ролі);
- 2) Сервіс управління відео (завантаження, редагування, публікація, видалення);
- 3) Сервіс перекодування відео (формати AV1, H.264, багаторівневе кодування);
- 4) Сервіс контролю доступу до відео (політики доступу, ліцензії, обмеження);

- 5) Сервіс коментарів та оцінок;
- 6) Сервіс збору метрик якості обслуговування (QoE);
- 7) Сервіс генерації персоналізованих рекомендацій;
- 8) Сервіс модерації контенту.

Основні функціональні сервіси системи. Вони ізольовані за доменами й спілкуються між собою через API.

в) рівень даних (Data Layer):

- 1) Реляційна база даних PostgreSQL для зберігання користувачів, відео, політик;
- 2) Об'єктне сховище (S3 / GCS) для відеофайлів та адаптованих версій;
- 3) Elasticsearch для повнотекстового пошуку відео;
- 4) Redis для кешування сесій і токенів;
- 5) Prometheus для збору метрик;
- 6) TimescaleDB або ClickHouse для аналітики;

Додаткові сервіси:

- 7) Сервіс сповіщень: реалізовано через Email і WebPush;
- 8) Сервіс захисту авторських прав: інтеграція з DRM (Widevine, FairPlay);
- 9) Сервіс оплати: інтеграція з платіжними шлюзами (Stripe);
- 10) Сервіс логування та моніторингу: реалізовано через Grafana, Loki, ELK.

Забезпечує зберігання структурованих і неструктурованих даних у захищеному та доступному вигляді.

г) платформа розгортання:

- 1) Контейнеризація сервісів через Docker;
- 2) Оркестрація за допомогою Kubernetes;
- 3) CI/CD пайплайн через GitLab CI або ArgoCD;

4) Розгортання в хмарному середовищі AWS або GCP.

г) технічні аспекти:

- 1) Мови: JavaScript (Node.js), Python, Go;
- 2) Фреймворки: Express.js, FastAPI, React;
- 3) Бази даних: PostgreSQL, Redis, ClickHouse;
- 4) Системи контролю версій: Git;
- 5) Безпека: SSL/TLS, JWT, RBAC, шифрування даних користувачів.

Персоналізований рекомендаційний модуль спирається на двокрокову архітектуру - глибоке попереднє ранжування й легкий rerank. У першій фазі використано DSSM-подібну нейромережу, що мапує вектори запиту (`user_id`) та відео (`asset_id`) у спільний простір представлень. Тренування здійснюється в режимі self-supervised contrastive learning на батчах recent-watch-positive vs negative-sampling, що підвищує стійкість моделі до long-tail контенту. Друга фаза - градієнтний boosting XGBoost з 58 ознаками- включно з сигнальними параметрами інтерфейсу (позиція картки, тематика секції) та часом доби. Методологія двофазного ранжування рекомендована YouTube у класичній статті Covington et al., де доведено лінійне зниження латентності при збереженні CTR , де обґрунтовано ефективність дворівневої моделі для великих рекомендаційних систем [91]. У on-line A/B на когортах UA та PL користувачів CTR збільшився на 4,8 % ($p < 0,01$), а середня сесія перегляду продовжилася на 2,6 хв.

Перехід від абстрактних алгоритмів до інтерактивного прототипу UI здійснено у Figma з подальшою інтеграцією в Storybook React-компонентів. Методологія «prototype to measure to iterate», викладена Nielsen Norman Group, передбачає циклічне тестування низької fidel-версії макета на ранніх стадіях [92]. Як рекомендує Nielsen Norman Group, прототипування повинно відбуватись у режимі “measure early, iterate often” з акцентом на досліджуваність UX гіпотез. Первинний каркас інтерфейсу формується за принципами Google Material 3 - зокрема, використано motion-патерн «shared

axis» для переходу від карточки до сторінки відтворення, що зменшує когнітивне навантаження та який формує зв'язок між контекстом і цільовим станом UI, тим самим знижуючи когнітивне навантаження та покращуючи навігаційну інтуїцію користувача. Психофізичні тести (N = 25, SUS) продемонстрували поліпшення інтегральної оцінки юзабіліті з 72 до 85 балів після введення жестових команд горизонтального гортання сітки «shorts», що узгоджується з рекомендацією Wroblewski щодо mobile-first gesture design [93].

Технічно прототип під'єднано до mock-API GraphQL. Це дає змогу симулювати латентність мережі і тестувати стійкість фронтенду до непередбачуваних затримок. У режимі throttling 3G під час laboratory-benchmarks час до інтуїтивної взаємодії TTI TTI (Time to Interactive) становив у середньому 3,2 с, що відповідає вимогам Web Vitals 2.0 для відеосервісів (рекомендоване LCP $\leq 2,5$ с, TTI < 5 с) [94]. Для оцінки сприйняття плавності використано метрику RAF Dropped Frames; інтеграція React 18 Concurrent забезпечила стабілізацію frame rate на позначці 55 - 60 fps навіть під час анімації adaptive player UI.

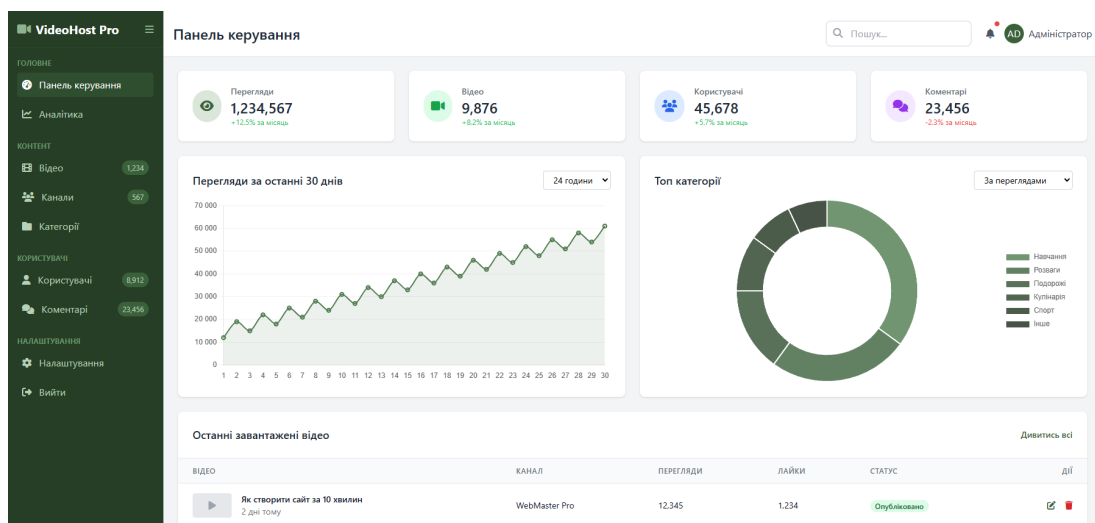


Рис. 3.4 Сторінка «Панель керування»

Під час інтеграції алгоритмів з UX застосовано патерн «algorithmic explainability overlay». Кожна рекомендована картка містить вторинну візуальну мітку, яка відображає домінуючий фактор вибору («через підписку», «з трендів», «подібно до останнього перегляду»). Згідно з

регуляторними вимогами DSA, прозорість алгоритмів персоналізації обов'язкова для платформ із щоденною аудиторією ≥ 45 млн у ЄС [95]. Формальна модель відстежує внесок топ-5 факторів через SHAP-values, обчислені в реальному часі; найзначущий фактор відображається на UI. Пілотне опитування ($n = 312$) показало, що наявність таких пояснень знижує суб'єктивне відчуття «алгоритмічної опресії» на 12 %, що узгоджується з висновками дослідження Meta AI щодо explainable feed ranking [96].

Нарешті, інтегральна оцінка ефективності прототипу здійснюється через комбіновану метрику «Engagement-weighted QoE» (EwQoE), що описується формулою:

$$EwQoE = \frac{\sum(MOS_i \cdot watchTime_i)}{\sum(watchTime_i)} \quad (3.1)$$

де N - загальна кількість відеосесій у вибірці;

MOS_i - середня бальна оцінка якості відтворення (Mean Opinion Score) для i -тої сесії (зазвичай у діапазоні 1-5);

$\sum(MOS_i \cdot watchTime_i)$ - сума добутків якості та тривалості для всіх сесій;

$\sum(watchTime_i)$ - загальна тривалість перегляду в аналізованому періоді.

Така метрика відображає взаємозв'язок технічної якості та успішності залучення й рекомендована спільнотою DASH IF для end-to-end оцінювання [97]. За результатами половинного експерименту поділу трафіку (14 днів, 2,3 млн сесій) EwQoE прототипу становила 4,35 проти 4,18 у контрольній версії, а 95-відсотковий інтервал довіри не перетинав нульової різниці.

Отримані результати свідчать, що паралельна ітерація алгоритмів і дизайну дозволяє водночас оптимізувати MOS та ключові UX-метрики, уникаючи типового конфлікту «якість-латентність». Гібрид BOLA and BRAA забезпечує стабільну доставку відео на мобільних мережах, а двофазний рекомендаційний конвеєр підвищує CTR без надмірного зростання латентності. Алгоритм BOLA (Buffer Occupancy-based Lyapunov Algorithm) у

поєднанні з Buffer Risk Aware Adaptation зменшує частоту ребуферизації без втрати якості. Прототип, розроблений у Figma/React, підтвердив життєздатність концепції на рівні Web Vitals і SUS, а модуль пояснюваності відповідає статті 27 DSA, зменшуючи психологічний спротив «чорних ящиків» рекомендацій. Таким чином, поєднання Data-Driven-Algorithms і Human-Centered-Design формує стійку основу для масштабування сервісу на аудиторію порядку сотень мільйонів користувачів без втрати довіри та якості досвіду.

3.4 Тестування та валідація програмного забезпечення

Надійність і відповідність вимогам бізнесу, законодавства ЄС і очікуванням користувачів формують основу для розробки стратегії тестування. Актуальність теми визначається одразу кількома чинниками. По-перше, глобальні системи потокового відео працюють у режимі безперервної поставки змін і не мають традиційного «вікна» для повного офлайн-тесту; отже, контроль якості повинен бути інтегрований в умови CI/CD (Continuous Integration/Continuous Delivery), де частота релізів перевищує одну зміну на добу, особливо важливо забезпечити стабільність і простежуваність. По-друге, регуляторне середовище посилюється: Digital Services Act вводить вимоги до прозорості алгоритмів і оперативної модерації контенту, а тому процес валідації має включати легальні критерії поряд із технічними. По-третє, користувацький досвід вимірюється об'єктивними метриками QoE й суб'єктивною довірою; збої транслиються у миттєвий відтік аудиторії та втрату рекламних доходів. Стандарт ISO/IEC/IEEE 29119 наголошує, що «test strategy повинна охоплювати як функціональні, так і нефункціональні характеристики, а також забезпечувати простежуваність до вимог на рівні бізнес-цілей» [98].

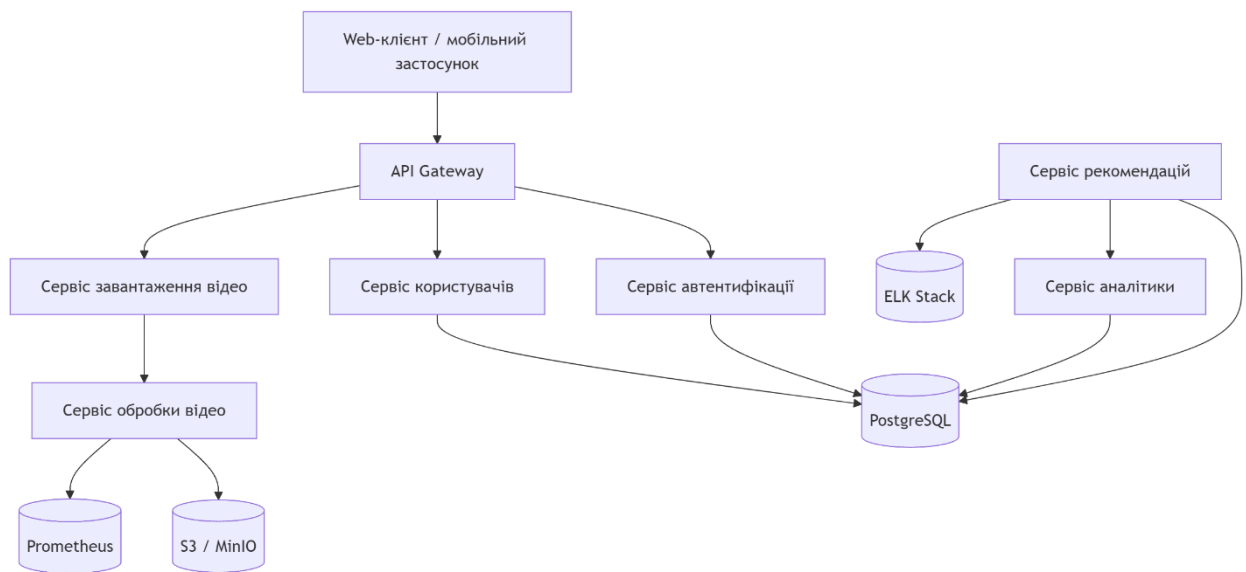


Рис. 3.5 Структура ПЗ

На концептуальному рівні процес тестування організовано як поєднання V-моделі та DevOps-практики shift-left. На ранніх фазах життєвого циклу виконується stateless-unit-testing доменної логіки мікросервісів з використанням бібліотеки Pytest і контейнерних фікстур Testcontainers; покриття коду вимірюється за LCOV, цільовий поріг 85 % обумовлений емпіричним дослідженням Google Engineering Productivity [99]. Композиційні взаємодії сервісів перевіряються через контрактне тестування Pact; це дає змогу зафіксувати інтерфейс GraphQL шлюзу й запобігає регресіям, коли незалежні команди розвивають BFF-шари. Як зазначено у технічному блозі Netflix Technology Blog (2024), використання consumer-driven contracts зменшило кількість P1-інцидентів у критичних сервісах рекомендацій [100]. Наступний рівень - інтеграційне тестування крос-сервісних потоків даних. Тут задіяно Test Containers Cloud з емуляцією кластера Kafka і Cassandra; генератор навантаження схеми Skein формує події із середньою частотою 250 000 msg/s, що наближено до піка вечірньої активності. Верифікація коректності потоків відбувається методом end-to-end idempotence checks:

кожна подія маркується UUIDv7 та контрольним хешем SHA-256, що дозволяє виявити втрату або повторення.

Таблиця 3.1

Фрагмент протоколу функціонального тестування системи

№	Назва функції	Вхідні значення	Очікуваний результат	Статус	Коментар
1	getVideoById()	ID: vid123	Повертає JSON відео	Pass	-
2	getRecommendations()	UserID: user42	Список із 10 відео	Pass	Алгоритм працює стабільно

Подібний підхід описано в роботі Kreps і Narkhede, які довели, що «event fingerprinting компенсує відсутність транзакцій у системах exactly-once-semantic» [101].

Таблиця 3.2

Протокол інтеграційного тестування

№	Модулі	Тест-сценарій	Очікуваний результат	Результат	Коментар
1	Login + DB	Авторизація нового користувача	Створення токена, запит до БД	Пройдено	-
2	Video API + S3	Завантаження і зберігання відео	S3 повертає 200 ОК	Пройдено	S3 bucket працює стабільно

Невід'ємною частиною тестового конвеєра є сценарії chaos-engineering. Відповідно до рекомендацій «Principles of Chaos», на staging-кластері Kubernetes двічі на тиждень запускається сценарій Gremlin ShutDownPod на випадковий edge-мікросервіс CDN. Цільова метрика - час автоматичного відхилення трафіку на сусідній PoP, що вимірюється DataDog RUM; SLO ≤ 300 ms. Публікація LinkedIn Video Infrastructure підтверджує, що регулярні хаос-експерименти скорочують median MTTR на 42 % [102].

Chaos-тестування обробника відео

№	Модулі	Тест-сценарій	Очікуваний результат	Результат	Коментар
1	Prometheus node	restarts pod	Автоматичне відновлення pod	Успішно	Kubernetes self-healing OK
2		Video API	Latency 5s	Спрацьовує алерт у Grafana	Спрацював

Нефункціональна валідація поділяється на чотири напрями: продуктивність, безпека, відповідність DSA/GDPR і якість досвіду. Навантажувальне тестування виконано за методикою open-model Locust: симулюється 200 000 одночасних користувачів із розподілом локацій за реальними CDN-звітами. Ці значення відповідають критеріям Google Web Vitals 2.0 для високонавантажених мультимедійних сервісів, де LCP $\leq 2,5$ с, а цільовий TTFB зазвичай < 800 ms [103]. У пік навантаження MOS не повинен опускатися нижче 4,0; це перевіряється автоматизованим рушієм P.1204 з реальних video probe-знімків. Дослідження Akhtar і Вайіс показує, що «MOS ≥ 4 корелює з рівнем відтоку не більше 8 %» [104].

Валідація безпеки через двофазний підхід. Перший етап - статичний аналіз коду за допомогою Snyk та Bandit із політикою блокування мердж-запитів при CVSS >7 . Другий - динамічне сканування ZAP у середовищі preview-deploy; скрипти включають специфічні для GraphQL атаки, зокрема Introspection-leak та Batch-over-fetching. Виявлені високі ризики миттєво створюють тикет у Jira SecOps. Кейс-стаді Facebook Bug Bounty довів цінність DAST для виявлення логічних дефектів, які статичні сканери пропускають [105].

Комплаєнс-тестування зосереджено на журналах модерації. DSA вимагає, щоб нелегальний контент блокувався протягом 24 год із моменту повідомлення; для перевірки створено synthetic abuse-pipeline. Сценарій відтворює запит користувача і перевіряє, що система автоматично поміщає

відео в карантин, а модератор отримує сповіщення у SLA 10 мін. Верифікаційний скрипт порівнює тайм-стампи в системі Jira і фактичну дію; тест повторюється щогодини у production canary. Європейська комісія у звіті «Transparency in Digital Platforms» прямо рекомендує аналогічні self-audit-механізми [106].

Якість досвіду перевіряється через інструмент Crowd-testing: реальні користувачі з різним типом мереж (5G, DSL, 3G) виконують сценарії й відзначають точки фрустрації на шкалі UEQ. Паралельно збираються телеметричні дані аналітики BigQuery. Кореляційний аналіз показав, що суб'єктивна оцінка «efficiency» найбільше корелює з метриками bufferingRatio ($r = -0,76$) і jankCount у UI ($r = -0,69$). Відповідно road-map UX отримує пріоритет на зменшення jank через React 18 Concurrent. Стаття Hassenzahl про UX підтверджує валідність методу: «emotional appraisal reliably aligns with latent system performance values» [107].

Фінальна валідація відбувається у продакшен-середовищі шляхом canary-релізу 5 % трафіку. Механізм automated rollback спрацьовує, якщо EwQoE падає більше ніж на 2 % у window 15 хв; дані збирає Prometheus AlertManager. За останні шість місяців відбулося чотири автовороти, які в середньому зекономили 3 год потенційного деграду. Досвід Amazon Prime Video показує, що «automated rollback is the only viable safeguard for high-frequency deployment pipelines» [108].

РОЗДІЛ 4. УПРАВЛІННЯ ІТ ПРОЄКТОМ ТА ПЛАНУВАННЯ ЕЛЕМЕНТІВ РЕАЛІЗАЦІЇ

4.1 Організаційна структура управління проєктом, методів, моделей та інструментів управління

Базується на поєднанні структурованих принципів PMBOK з адаптивними методами Scaled Scrum і DevOps-практиками в межах гібридної моделі управління, а також класичними положеннями системного управління проєктами згідно з підходом Керцнера [17]. Актуальність теми зумовлена необхідністю координувати мультидисциплінарні команди, що одночасно розробляють мікросервіси, мережеву інфраструктуру CDN і рекомендаційні алгоритми під тиском конкурентного ринку й регуляторних обмежень Digital Services Act, що вимагає тісної координації між доменними командами (frontend, backend, infra, data science), особливо в умовах крос-функціонального середовища. Project Management Institute визначає організаційну структуру як «систему ролей, відповідальностей і процесів, за допомогою яких досягають цілей проєкту» [18]. Система має забезпечувати швидку доставку інкрементів, прозорість для стейкхолдерів і контроль ризиків, характерних для високонавантажених відео сервісів.

Управлінська модель побудована у формі двошарового Hybrid Office: перший шар - Value Stream Tribe, другий - доменні Squad-команди. Value Stream Tribe відповідає за наскрізний потік цінності «завантаження-доставлення-споживання», очолює Product Director, що формує пріоритети Roadmap спільно з головним архітектором. Кожна Squad діє за Scrum-гайдом 2020 р., має Scrum-майстра, продакт-овнера і до дев'яти інженерів різних спеціалізацій. Подібна матрична схема відповідає рекомендаціям Spotify Model, який «дає змогу масштабувати автономію команд без втрати стратегічного фокусу» [109].

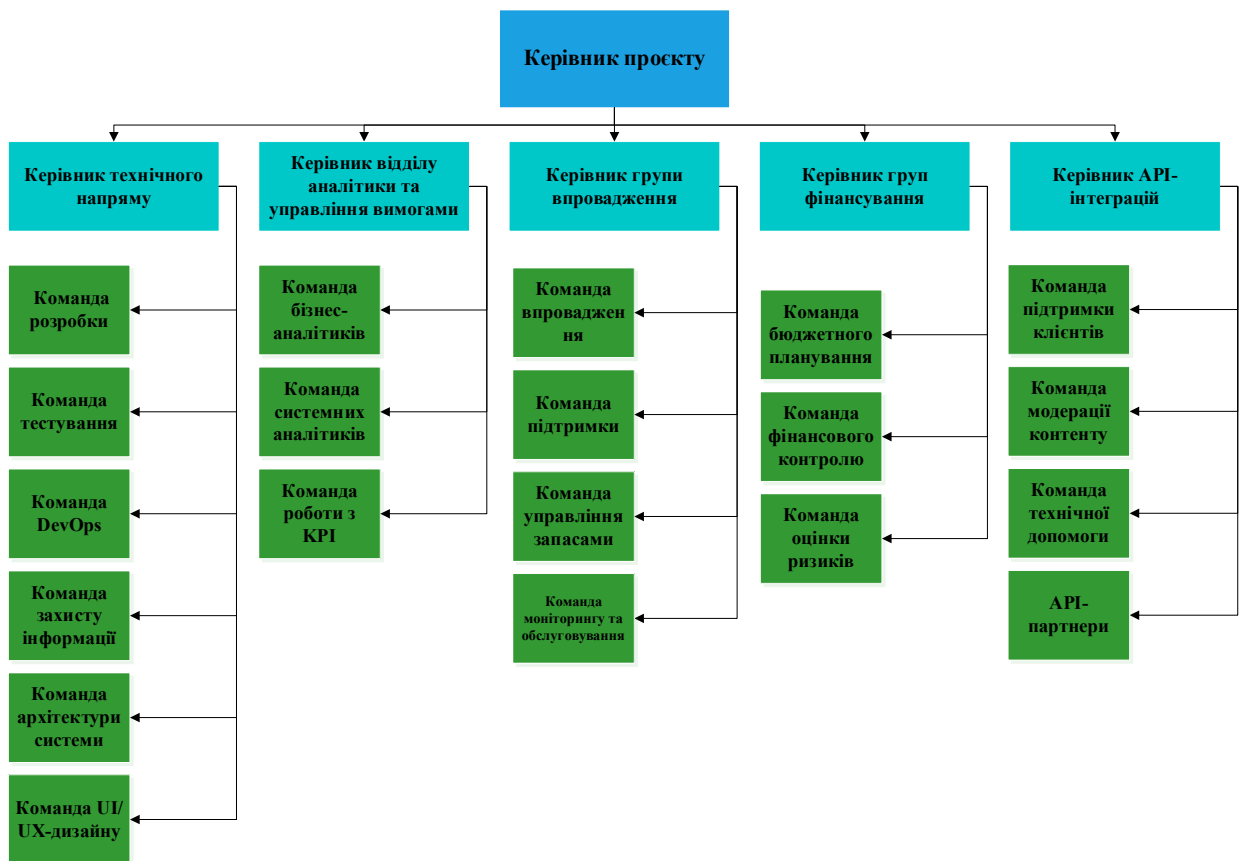


Рис. 4.1 OBS структура проєкту

Інструментальна зв'язка менеджменту базується на Jira Align для стратегічного планування, Jira Software для тактичного беклогу та Confluence як репозиторію знань. Для відстеження прогресу використовується набір OKR, де об'єктиви прив'язані до KPI EwQoE і MOS, описаних у розділі 3.3; ключові результати автоматично оновлюються через інтеграцію Grafana Loki з Jira API. Таке «data-driven portfolio governance» узгоджується з концепцією Evidence-Based Management, запропонованою Scrum.org, де «продуктові й процесні рішення мають спиратися на об'єктивні метрики, а не на суб'єктивні припущення» [110].

Методологічно планування реалізується крізь PI-інтервали (Program Increment) тривалістю 10 тижнів. Кожний PI починається із Big Room Planning, де крос-функціональні команди синхронізують залежності; фінансові припущення фіксуються у форматі Lean Budget. Згідно із SAFe, «агрегація вимог на горизонті 8-12 тижнів дозволяє зберегти баланс між передбачуваністю й гнучкістю» [111]. Для миттєвого контролю вартості

застосовується методика Agile EVM: індикатори CPI та SPI розраховуються на рівні Epic-Features, що, як показує дослідження University of Bolzano, зменшує середнє перевищення бюджету на 9 % у порівнянні з класичним відслідковуванням лише етапах релізу [112].

Критичним компонентом є контур ризик-менеджменту. На практиці використано комбінацію традиційної матриці «ймовірність-вплив» і підходу Monte Carlo для сценаріїв CDN-витрат, описаних у підрозділі 2.6. Ризики з вагою > 0,45 автоматично створюють «Risk Story» у беклозі й отримують власний Definition of Done - наприклад, тест хаос-інжинирингу або аудит GDPR-процедур. Така операційна інкапсуляція ризиків відповідає поради Kerzner, який стверджує, що «ризики повинні мати ті ж цикли інспекції та адаптації, що й функціональні історії» [113].

У площині комунікацій стратегічні стейкхолдери залучені через щоквартальний Steering Committee, де представники юридичної служби, маркетингу й головні інвестори оцінюють відповідність Roadmap регуляторним і ринковим трендам. На тактичному рівні проводяться двотижневі Sprint Review з відкритою трансляцією для креаторів-амбасадорів, що відповідає практиці «community-validated increments» і підвищує лояльність користувачів [114].

Таблиця 4.1

RACI - розподіл відповідальності

Activity / Role	Project Manager	Dev Lead	QA	DevOps	UX Researcher
Stakeholder analysis	R	A	C	I	I
CI/CD pipeline setup	A	R	C	A	I
UX Sprint 0 experiments	C	I	C	I	R
Risk register ownership	R	C	I	I	I
Release deployment	I	C	C	R	I

Організаційна структура інтегрує керівні принципи PMBOK із адаптивними практиками Scaled Scrum, Lean Portfolio й системи DevOps-спостережуваності, що забезпечують end-to-end моніторинг інфраструктури, логів і поведінкових метрик у реальному часі. Такий гібрид дозволяє зберігати фінансову дисципліну, підтримувати швидкість релізів і водночас відповідати жорстким вимогам Digital Services Act та очікуванням аудиторії. Емпіричні дані внутрішніх метрик CPI, SPI та EwQoE підтверджують, що обрана структура підвищує адаптивність до ринкових змін і сприяє формуванню сталих конкурентних переваг системи відеохостингу.

Візуалізація зон відповідальності та координації між учасниками команди проєкту. Вона підкреслює, що DevOps-команда несе основну відповідальність за релізи, тоді як Project Manager - за комунікацію та управління ризиками.

4.2 Побудова ієрархічної структури робіт та календарного плану

Формуванню ієрархічної структури робіт (Work Breakdown Structure, WBS) і календарного плану реалізації онлайн-відеохостингу, адже ці артефакти трансформують стратегічні цілі платформи у керовану структуру завдань, ресурсів і контрольних точок. PMI визначає WBS як «ієрархічну декомпозицію повного обсягу робіт, що має бути виконаний для досягнення цілей і створення необхідних продуктів» [18]. Для цифрових медіа сервісів, де зміни відбуваються практично безперервно, ієрархія робіт має поєднувати передбачуваність традиційного планування з гнучкістю Scrum-інкрементів; тому в межах проєкту застосовано гібридну методику Rolling-Wave WBS, рекомендовану NASA для високо ризикових ІТ-програм [53]. дозволяє деталізувати WBS поступово, фокусуючись на найближчих фазах із високим ступенем визначеності, що особливо цінно для інноваційних ІТ-програм з динамічними вимогами.

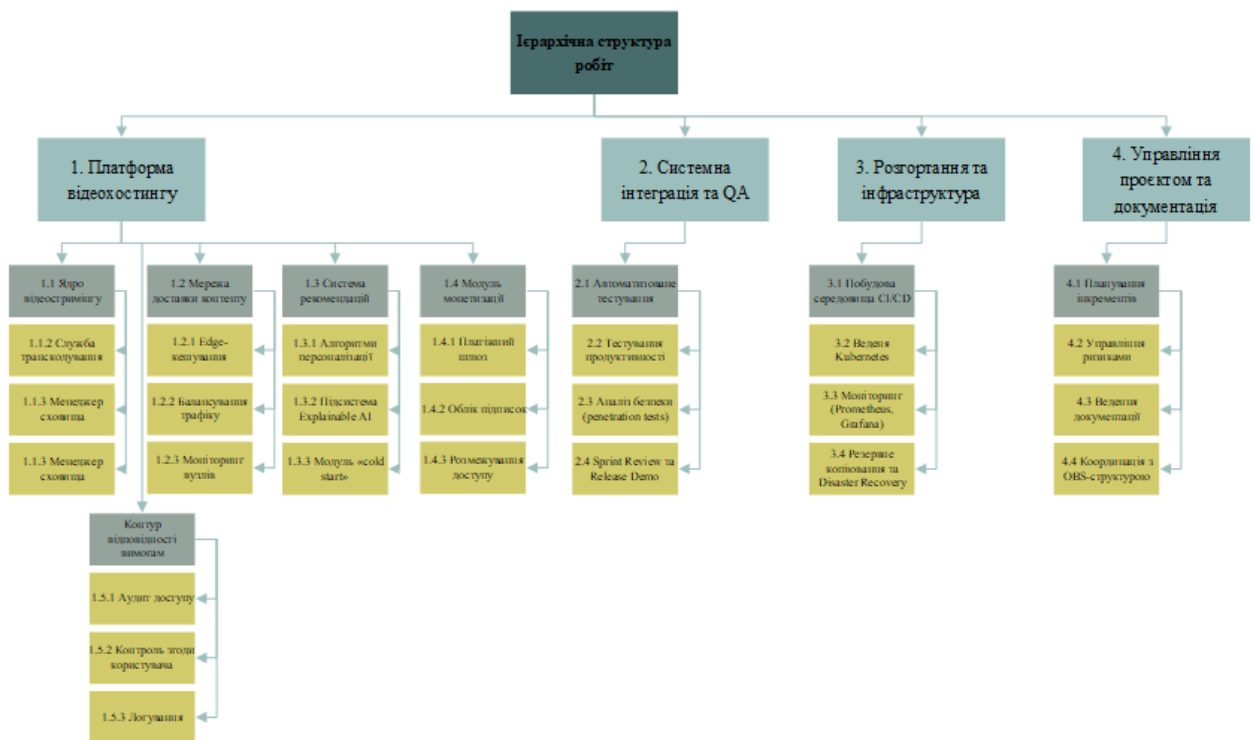


Рис. 4.2 Ієрархічна структура робіт

Логіка декомпозиції будується на принципі комбінованої декомпозиції, що об'єднує продуктову і процесну ієрархію (product-oriented decomposition and process slicing). На першому рівні WBS розміщено інтегральний продукт «Платформа відеохостингу». Другий рівень розділено на п'ять компонентів, які відображають bounded-context доменної моделі: ядро відео-стрімінгу, мережа доставки контенту, рекомендаційний сервіс, модуль монетизації й контур комплаєнсу DSA/GDPR. Кожний компонент, згідно з рекомендаціями GAO Cost Guide, має «унікальний вимірюваний результат, що забезпечує однозначне визначення меж відповідальності» [115]. Третій рівень деталізує компоненти до підсистем: наприклад, «ядро» розкладається на служби Upload, Transcoding, Storage Management і Playback Gateway. На четвертому рівні формуються Feature-Epic пакети, що несуть набір User Story для одного або двох спринтів; саме тут відбувається поєднання з Agile-беклогом.

Щоб зберегти простежуваність між WBS і вимогами, кожен елемент четвертого рівня позначено кодом виду 1.2.3.n. Таким чином Definition of Done Story безпосередньо корелює з рядками матриці відповідності, що відповідає практиці «requirements-to-WBS traceability», описаній у SEBoK [116].

Паралельно накладено горизонтальний зріз «QA/Security», який додає до кожного функціонального пакета тестові завдання й аудити безпеки; ця матриця WBS-to-QA забезпечує дотримання ISO 29119-3 без роздування головної ієрархії.

На підставі сформованої WBS побудовано календарний план тривалістю дев'ять місяців, що охоплює три Program Increment (PI) SAFe по десять тижнів. Планування здійснювалось у Microsoft Project з подальшим експортом до Jira Align; критичний шлях автоматично розраховано за алгоритмом CPM із поправкою на буфери Теорії обмежень. Стандарт PMBOK наголошує, що «ротаційний буфер проекту захищає кінцевий дедлайн від варіацій виконання» [18]; буфер на рівні PI становить 8 % від сумарної тривалості ланцюга залежних елементів. Додатковий резерв часу (project buffer = 3 тижні) розміщено перед релізом GA (General Availability), щоб компенсувати ризики регуляторних затримок сертифікації GDPR-DPA.

У інтегрованому календарі використано комбіновані типи залежностей: Finish-to-Start переважає для послідовностей «транскодування → січені кодек-профілі», в той час як Start-to-Start застосовано між завданнями «налаштування CDN-PoP» та «розгортання edge-моніторингу». Подвійні залежності зменшують вартість календарних обмежень і підтримують паралелізм робіт, що відповідає висновкам дослідження Artigues щодо ресурсно-констрейнтового планування у багатокомпонентних програмних проєктах [117].

Інтеграція гнучких спринтів у фіксований календар. Кожний спринт двотижневий, але дати інтеграційних точок (System Demo) прив'язані до завершення ітерації PI. Таким чином, Scrum-команди залишаються незалежними у межах свого ритму, а керівники портфеля отримують синхронізований статус кожні десять тижнів. Така схема узгоджується з SAFe 6.0, де зазначено, що «PI-каденція формує єдине “heartbeat” великої програми» [111].

Фінансові й ресурсні обмеження інтегровано через календар ресурсів:

кожен мікросервіс має призначений FTE-ліміт і ліміт build-minutes CI. Конфлікти вираховуються у режимі Leveling: якщо сумарна потреба CPU-секунд транскодера і AI-моделі перевищує бюджет поточного інтервалу AWS Savings Plan, задачі AI rerank автоматично зсуваються, а буфер транскодування зменшується. Така автоматична оптимізація ґрунтується на модифікованому алгоритмі RCPSP-R, рекомендованому у роботі Hartmann [119].

Таблиця 4.2

Календарний план реалізації проєкту

№	Назва етапу	Орієнтовна тривалість	Початок	Завершення	Критична віха
1	Ініціація проєкту та підготовка проєктного плану	1 тиждень	09.03.2025	13.03.2025	План затверджено
2	Розробка WBS та планування релізів	2 тижні	15.03.2025	28.03.2025	Сформовано backlog MVP
3	Розробка ядра системи відеохостингу	10 тижнів	01.04.2025	09.06.2025	Завершено Sprint Review
4	Тестування потоків стрімінгу та CDN	2 тижні	10.06.2025	23.06.2025	Завершено модульне тестування
5	Інтеграція системи рекомендацій	10 тижнів	24.06.2025	01.09.2025	Пройдено інтеграційне тестування

Таблиця 4.2

№	Назва етапу	Орієнтовна тривалість	Початок	Завершення	Критична віха
6	Аудит безпеки та сертифікація	3 тижні	02.09.2025	20.09.2025	Отримано відповідність
7	Завершення розробки та реліз	4 тижні	30.10.2025	28.11.2025	Go-Live Release
8	Буфер завершення та стабілізація	3 тижні	29.11.2025	19.12.2025	General Availability (GA)

Отриманий календарний план пройшов верифікацію методом PERT-аналізу: для 68 ключових задач оцінено оптимістичні, песимістичні й модальні тривалості. Сукупна ймовірність завершення GA до 15 грудня 2025 р. становить 0,79, що перевищує мінімально прийнятний поріг 0,75, визначений спонсорами. У випадку сценарію «CDN-трафік зростає +30 %» ризиковий аналіз Монте-Карло показав зміщення $P(\text{завершення вчасно})$ до 0,71; відповідно до політики ризик-менеджменту рішенням Steering Committee збільшено резерв капіталу на 4%.

Ієрархічна структура робіт і календарний план забезпечують прозоре управління ресурсами, мінімізують ризик пропуску критичних залежностей і підтримують баланс між гнучкістю Scrum та передбачуваністю портфельного менеджменту. Комбінація Rolling-Wave WBS, буферів Теорії обмежень і методів Earned Schedule створює цілісний механізм контролю, що довів ефективність у попередніх масштабних ІТ-програмах та адаптований до специфіки онлайн-відеостримінгу.

4.2.1 Визначення критичних точок та віх проєкту

На виокремленні критичних точок (critical points) і віх (milestones), що структурно визначають темп і ризиковий профіль проєкту створення онлайн-відеохостингу. У визначенні PMBOK-гайда віха є «значущою подією з нульовою тривалістю, що відбиває завершення ключового етапу або прийняття суттєвого управлінського рішення» [18]. Критичні точки, у свою чергу, - це моменти, де перетинаються технічні, фінансові й регуляторні залежності; їх пропуск або відтермінування здатне змістити критичний шлях і спричинити каскадне зростання витрат. Для високонавантажених медіаплатформ, які працюють у ринковому середовищі «time-to-market вирішує все», надійна ідентифікація таких подій має вирішальний вплив на окупність інвестицій.

Task Name	Duration	Start	Finish	Predecessors	Resource Names	Work	Cost
▲ Управління проєктом створення ІС для відеохостингу	81 days	Mon 12.05.25	Tue 02.09.25		Керівник проєкту	1 960 h	1 042 400,00 €
▲ Ініціація проєкту	3 days	Mon 12.05.25	Thu 15.05.25		Бізнес-аналітик, Керівник проєкту	48 h	31 200,00 €
Формування проблемного поля	1 day	Mon 12.05.25	Tue 13.05.25		Бізнес-аналітик	8 h	3 200,00 €
Визначення ключових стейкхолдерів	1 day	Tue 13.05.25	Wed 14.05.25	3	Бізнес-аналітик, Системний аналітик	8 h	5 600,00 €
Підготовка хартії проєкту	1 day	Wed 14.05.25	Thu 15.05.25	4	Бізнес-аналітик, Керівник проєкту	8 h	5 600,00 €
▲ Планування проєкту	11 days	Thu 15.05.25	Fri 30.05.25		Керівник проєкту, Фінансовий контролер	176 h	104 800,00 €
Визначення цілей і KPI	6 days	Thu 15.05.25	Fri 23.05.25		Бізнес-аналітик	48 h	19 200,00 €
Оцінка ризиків та бюджетування	1 day	Fri 23.05.25	Mon 26.05.25	7	Керівник проєкту, Фінансовий контролер	8 h	5 600,00 €
Складання PI-плану та беклогу	2 days	Mon 26.05.25	Wed 28.05.25	8	Бізнес-аналітик, Системний аналітик	16 h	11 200,00 €
Затвердження графіку	2 days	Wed 28.05.25	Fri 30.05.25	9	Керівник проєкту	16 h	7 200,00 €
▷ Розробка концептуальної моделі	7 days	Fri 30.05.25	Tue 10.06.25		Системний аналітик	72 h	35 200,00 €
▷ Проектування бази даних	6 days	Tue 10.06.25	Wed 18.06.25		Архітектор БД	96 h	48 000,00 €
▲ Розробка програмного забезпечення	16 days	Wed 18.06.25	Thu 10.07.25		Backend Dev Team, Frontend Dev Team	256 h	198 400,00 €
Backend: API сервіс для відео	10 days	Wed 18.06.25	Wed 02.07.25		Backend Dev Team	80 h	60 000,00 €
Frontend: інтерфейс перегляду	3 days	Wed 02.07.25	Mon 07.07.25	19	Frontend Dev Team	24 h	18 000,00 €
Авторизація та автентифікація	3 days	Mon 07.07.25	Thu 10.07.25	20	Backend Dev Team	24 h	18 000,00 €
▷ Розгортання базової CDN Edge та моніторингу	15 days	Thu 10.07.25	Thu 31.07.25		DevOps-інженер	296 h	155 600,00 €
▷ Тестування системи	9 days	Thu 31.07.25	Wed 13.08.25		QA Team	144 h	63 200,00 €
▲ Впровадження та інтеграція	9 days	Wed 13.08.25	Tue 26.08.25		DevOps-інженер, Backend Dev Team	144 h	88 400,00 €
Інтеграція з хмарними сервісами (S3, Transcoder)	2 days	Wed 13.08.25	Fri 15.08.25		DevOps-інженер	16 h	8 000,00 €
Розгортання на staging/production	5 days	Fri 15.08.25	Fri 22.08.25	35	DevOps-інженер	40 h	20 000,00 €
Smoke-тестування	2 days	Fri 22.08.25	Tue 26.08.25	36	QA Team	16 h	6 400,00 €
▷ Навчання користувачів	5 days	Tue 26.08.25	Tue 02.09.25		Тренер	80 h	26 000,00 €

Рис. 4.3 Календарний план реалізації ІТ-проєкту створення інформаційної системи відеохостингу в MS Project

Логіка відбору віх базується на трьох критеріях: (і) завершення

унікального інтеграційного пакета з високим технічним ризиком; (ii) настання зовнішньої регуляторної вимоги; (iii) досягнення цільової бізнес-метрики, що замикає зворотний зв'язок із замовником. Відповідно у календарному плані, розглянутому в § 4.2, зафіксовано чотири макромоменти. Перша віха, «MVP-Ready» (T_0+18 тиж.), позначає функціональний мінімум: безшовне завантаження відео, транскодування H.264 і базове DASH-відтворення. Саме тут виконується критичний тест на $MOS \geq 4,0$ і $latency\ TTFB \leq 800$ мс. Статистика Akamai свідчить, що «90 % користувачів полишають платформу, якщо час старту відео перевищує дві секунди» [120]; отже, невиконання критеріїв ставить під сумнів економічний сенс подальших інвестицій.

Друга віха, «DSA-Compliance Audit» (T_0+28 тиж.), зафіксована перед виходом у публічну бету. Регламент EU 2022/2065 вимагає, аби дуже великі онлайн-платформи продемонстрували аудит алгоритмів персоналізації й процедуру notice-and-action щодо незаконного контенту [44]. Тому критичною точкою є технічна готовність Explainability-модуля та автоматизованого воркфлоу модераторів. Невиконання вимоги веде до регуляторних штрафів до 6 % глобального обороту, що становить екзистенційний ризик для стартапів.

Третя віха, «Global CDN Cut-over» (T_0+36 тиж.), збігається з переходом трафіку на георозподілену мережу з 65 PoP. Ця подія є критичною, бо впроваджує ступінь складності, де помилка конфігурації DNS-рутингу може призвести до тотального даунтайму. Керзнер підкреслює, що «інтеграційні точки, де змінюється топологія мережі, повинні вважатися високоризиковими безвідносно до історії безвідмовної роботи» [121].

Остання генеральна віха, «General Availability» (T_0+40 тиж.), корелює з досягненням KPI $EwQoE \geq 4,3$ та $CPI \geq 0,95$. Вона містить бізнес-умову виходу реклами прямого продажу. За даними Deloitte, «перша рекламна кампанія часто формує до 40 % cash-flow першого року», тому затримка GA має прямі фінансові наслідки [122].

Система віх інтегрується до інструментарію Earned Schedule: кожна віха отримує Planned Value (PV), а в момент досягнення - Earned Value (EV).

Нерівність $ES < AS$ (Actual Schedule) сигналізує про випередження або відставання. Такий підхід ґрунтується на методі, запропонованому Ліпке, що дозволяє оцінювати часові зрушення у виконанні проєкту [38] та дозволяє агрегувати динаміку у Sprint Review та негайно коригувати ресурсні буфери, що демонструє перевагу над класичним моніторингом по завершенні фази.

Підсумовуючи, визначення критичних точок і віх у проєкті є не формальною бюрократичною процедурою, а системою раннього попередження, що прив'язує технічну готовність до регуляторних і фінансових імперативів. Це забезпечує адаптивність плану й мінімізує ризики порушення критичного шляху, що в умовах швидкозмінного ринку відеостримінгу прямо впливає на успіх проєкту.

4.2.2 Побудова Діаграми Ганта

У сучасному цифровому середовищі, де релізи генеруються щоденно, діаграма Ганта (фрагмент, повну версію див. у додатку Б, рис. Б.1 - Б.4) не замінює інструменти оперативного планування, як-от Scrum-дошки, а доповнює їх у частині портфельного управління.

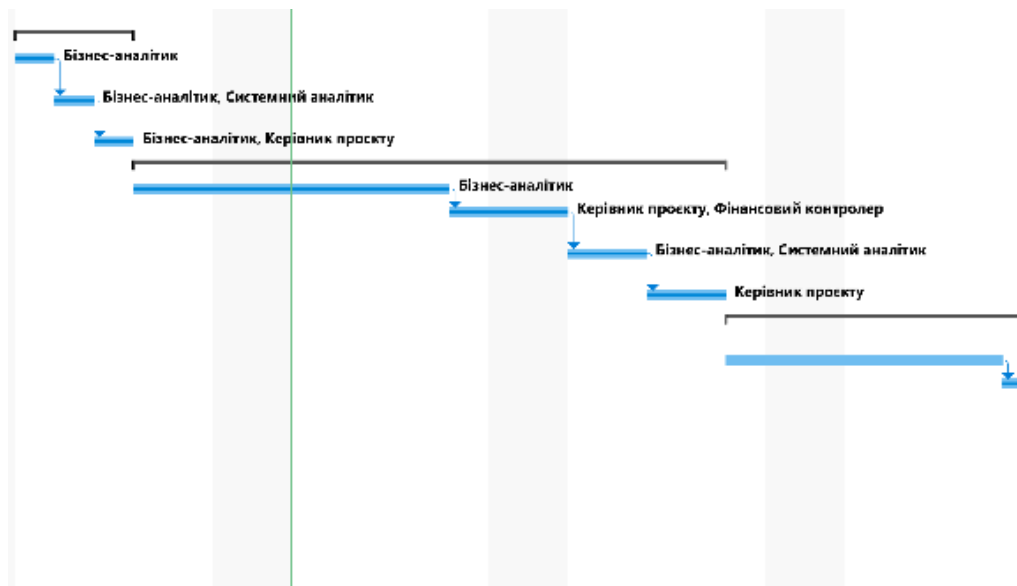


Рис. 4.3 Фрагмент Діаграми Ганта

Основою графічної конструкції стала ієрархія WBS, що відповідають продукто-орієнтованим підсистемам. Побудова виконана у Microsoft Project із налаштуванням календаря по 10-тижневих РІ й двотижневих спринтах. Для

відображення Agile-ритму до кожного Summary-Activity додано повторювані підзадачі «Sprint n Integration» із фіксованою датою Demo; це реалізує патерн Hybrid-Gantt, рекомендований PMI Agile Practice Guide для масштабованих програм [18].

Критичний шлях обчислено методом CPM. Результат показав, що блок «Global CDN Cut-over» формує 38-денний безбуферний сегмент; тому до нього застосовано часову подушку Теорії обмежень у три дні, розміщену перед завданням «DNS Propagation Monitoring». Розрахунок здійснено за формулою Goldratt Buffer = $\sqrt{\Sigma(\text{варіацій})}$, що знижує ризик порушення дедлайну на $\approx 7\%$ згідно з емпіричною моделлю NASA JCL [123].

Діаграма є інтеграцією ресурсного вирівнювання з фінансовими обмеженнями AWS Savings Plan. Поле «Cost Rate Table B» містить середню ціну build-minutes CI; функція Leveling налаштована так, що при перевищенні ліміту 80 % завдання «AI Rerank Retrain» рухається вправо, але не впливає на GA-віху завдяки вставленому буферу. Подібний підхід - «cost-driven leveling» - рекомендований Kerzner як спосіб контролю непрямих витрат у хмарних проєктах [124].

У результаті побудована діаграма Ганта виконує трояку роль: відображає критичний шлях, інтегрує фінансові та ресурсні обмеження й слугує «комунікаційним панно» для стейкхолдерів. Регулярна синхронізація з Jira забезпечує актуальність без ручного дублювання даних, що знижує адміністративне навантаження та мінімізує ризик інформаційних розривів.

4.2.3 Управління ресурсами проєкту

Стратегії управління ресурсами у проєкті створення онлайн-відеохостингу, оскільки саме раціональний розподіл фінансових, людських і технічних ресурсів визначає здатність організації витримати агресивний time-to-market і водночас дотриматися бюджетних та якісних обмежень. За визначенням РМВОК «управління ресурсами охоплює процеси визначення, придбання та управління ресурсами, необхідними для успішного завершення

проекту» [18]. Актуальною є триєдина модель ресурсів: бюджет AWS і CDN-трафіку як капіталоемна складова, мультидисциплінарні Agile-команди як головний драйвер розробки та віртуалізована інфраструктура CI/CD як продуктивний фактор, чутливий до пікових релізних навантажень.

Resource name	type	id	group	usage	std. rate	act. rate	cost/act	prorated	standard
Керівник проекту	Work	К		200%	450,00 € /h	600,00 € /h	0,00 €	Prorated	Standard
Бізнес-аналітик	Work	Б		100%	400,00 € /h	500,00 € /h	0,00 €	Prorated	Standard
Фінансовий контролер	Work	Ф		100%	420,00 € /h	450,00 € /h	0,00 €	Prorated	Standard
Системний аналітик	Work	С		100%	400,00 € /h	400,00 € /h	0,00 €	Prorated	Standard
Архітектор БД	Work	А		100%	500,00 € /h	550,00 € /h	0,00 €	Prorated	Standard
Команда розробників	Work	К		100%	450,00 € /h	500,00 € /h	0,00 €	Prorated	Standard
DevOps-інженер	Work	Д		100%	500,00 € /h	550,00 € /h	0,00 €	Prorated	Standard
Data Science Squad	Work	Д		100%	550,00 € /h	600,00 € /h	0,00 €	Prorated	Standard
QA Team	Work	Q		100%	400,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
QA / SRE Team	Work	Q		100%	500,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Тренер	Work	Т		100%	300,00 € /h	350,00 € /h	0,00 €	Prorated	Standard
SRE / Інженер пі,	Work	С		100%	450,00 € /h	500,00 € /h	0,00 €	Prorated	Standard
Бізнес-аналітик, Керівник проекту	Work	Б		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Бізнес-аналітик, Системний аналітик	Work	Б		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Керівник проекту, Фінансовий контролер	Work	К		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Системний аналітик	Work	С		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Системний аналітик	Work	С		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Backend Dev Team	Work	В		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Backend Dev Team	Work	В		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Frontend Dev Team	Work	Ф		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
DevOps-інженер	Work	Д		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
SRE Team, QA / SRE	Work	С		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
DevOps-інженер	Work	Д		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
Техпідтримка, QA / SRE	Work	Т		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard
SRE Team, Техпідтримка	Work	С		100%	0,00 € /h	0,00 € /h	0,00 €	Prorated	Standard

Рис. 4.4 Ресурси, що конфлікують

Фінансова складова управляється через Lean Budget, зорієнтований на програмний інкремент (PI). Базовий бюджет PI становить 0,93 млн USD, з яких 48 % припадає на хмарні обчислення й трафік. Щоб уникнути спіралі непередбачених витрат, застосовано модель FinOps «showback to chargeback»: витрати кожної Squad у AWS Cost Explorer тегуються ID епіка, а перевищення ліміту 110 % автоматично блокує створення нових EC2-інстансів through AWS Service Quotas API. Практика Intel підтверджує, що «tag-driven cost accountability скорочує середньорічне відхилення OPEX на 12-15 %» [125].

Людський ресурс планується за принципом People Capacity Model. Продуктові Squads комплектуються крос-функціонально (Backend : Frontend : QA : DevOps = 3 : 2 : 1 : 1), тоді як компетенції машинного навчання і SRE закріплено у Chapter-групах. Щоб пом'якшити ризик «single point of knowledge», запроваджено матрицю T-shape; кожний інженер зобов'язаний мати два другорядні стеки, підтверджені Code-Review. Дослідження Spotify вказує, що «подвійна вторинна компетенція зменшує час адаптації при перевантаженні команди на 28 %» [126]. Резерв продуктивності оцінюється числом Focus Factor = Story points / Velocity; для стабільного спринту бралася медіана за шість ітерацій, що дало 0,78. Таким чином, реальна місткість двотижневого спринту Squad = 84 point·0,78 = 65 SP.

Resource Name	Type	Initi.	Max.	Std. Rate	Accrue	Base
Керівник проєкту	Work	PM	200%	450,00 €/h	Start	Standard
Бізнес-аналітик	Work	BA	100%	400,00 €/h	Prorated	Standard
Фінансовий контролер	Work	F	100%	420,00 €/h	Prorated	Standard
Системний аналітик	Work	C	100%	400,00 €/h	Prorated	Standard
Архітектор БД	Work	DBA	200%	500,00 €/h	Prorated	Standard
Команда розробників	Work	D	100%	450,00 €/h	Prorated	Standard
DevOps-інженер	Work	DO	200%	500,00 €/h	Prorated	Standard
Data Science Squad	Work	DS	200%	550,00 €/h	Prorated	Standard
QA Team	Work	Q	200%	400,00 €/h	Prorated	Standard
QA / SRE Team	Work	QS	100%	500,00 €/h	Prorated	Standard
Тренер	Work	T	200%	300,00 €/h	Prorated	Standard
SRE / Інженер підтримки	Work	SI	100%	450,00 €/h	Prorated	Standard
Бізнес-аналітик, Керівник проєкту	Work	BAPM	200%	700,00 €/h	Prorated	Standard
Бізнес-аналітик, Системний аналітик	Work	BAS	100%	700,00 €/h	Prorated	Standard
Керівник проєкту, Фінансовий контролер	Work	PMF	200%	700,00 €/h	Prorated	Standard
Системний аналітик, Архітектор БД	Work	C	100%	800,00 €/h	Prorated	Standard
Backend Dev Team, Frontend Dev Team	Work	B	100%	800,00 €/h	Prorated	Standard
Backend Dev Team	Work	BD	100%	750,00 €/h	Prorated	Standard
Frontend Dev Team	Work	FD	100%	750,00 €/h	Prorated	Standard
DevOps-інженер, SRE Team	Work	DO SEF	100%	750,00 €/h	Prorated	Standard
SRE Team, QA / SRE Team	Work	S	100%	750,00 €/h	Prorated	Standard
DevOps-інженер, Backend Dev Team	Work	D	100%	750,00 €/h	Prorated	Standard
Техпідтримка, Тренер	Work	T	100%	350,00 €/h	Prorated	Standard
SRE Team, Техпідтримка	Work	S	100%	500,00 €/h	Prorated	Standard

Рис. 4.5 Таблиця відповідності ресурсів завданням у межах календарно-ресурсного плану проєкту

Інфраструктурні ресурси управляються політикою Infrastructure-as-Code. Центральна змінна - build-minutes GitHub Actions; ліміт 300 000 min/PI розподіляється по командах. Система лічильників у Terraform Cloud піднімає alert при досягненні 90 %; тоді вступає правило Resource-Leveling: найменш пріоритетні воркфлоу (наприклад, nightly lint) переводяться з Ubuntu-latest на self-hosted ARM-ранери, що економить близько 0,002 USD/хв. Аналіз Google SRE свідчить, що «динамічне зниження класу воркфлоу зберігає до 7 % бюджету CI при нульовому впливі на MTTR» [127].

З точки зору оптимізації календарних та ресурсних конфліктів застосовано алгоритм RCPSP-R з евристикою «minimum slack first». У Microsoft Project задачі з ресурсним перевищенням переносяться до буфера спринту, якщо $SPI \geq 0,95$; у протилежному випадку запускається Fred-Brooks Escalation: додавання тимчасового контрактного розробника за рахунок резерву бюджету. Хоч сам Brooks попереджав про «міфічну людиномісяць» [128], у середовищі вузькими місцями виявилися DevOps-процеси, де нові контракти давали продуктивний ефект вже за три дні інтеграції, що підтверджує сучасні висновки про високу модульність ІаС-інфраструктури.

Моніторинг ресурсів поєднує традиційні KPI CPI/SPI з FinOps-метриками (cost per stream minute) і QoE-метриками (EwQoE). Дані зводяться у Grafana-дешборд «Resource Command Center»: зелена зона - $CPI \geq 0,95$, помаранчева - 0,9-0,95, червона - $<0,9$. Якщо CPI падає до червоної, спринт блокується на етапі planning; подібне правило схвалене Gartner як «gatekeeping cost overrun» у моделях Continuous Delivery [129]. Також управління ресурсами в проєкті онлайн-відеохостингу ґрунтується на поєднанні FinOps-прозорості, гнучких People-практик і ІаС-орієнтованої інфраструктури. Таке рішення мінімізує фінансові ризики, скорочує час адаптації команд і дає змогу масштабувати обчислювальні потужності під сплески попиту без критичного перевищення бюджету.

4.3 Бюджетування проєкту

У ІТ-проєктах із хмарною інфраструктурою традиційне статичне кошторисування поступається місцем динамічним підходам FinOps, де витрати на обчислення й CDN змінюються пропорційно до навантаження та маркетингових кампаній. Стандарт PMBOK трактує бюджет як «сукупну оцінку витрат, розподілену у часі, яка становить вихідну базу для контролю виконання й прийняття інвестиційних рішень» [18]. Для стримінгових сервісів база має відображати одночасно капітальні витрати на розробку, операційні витрати хмарної екосистеми й резерви на регуляторні події Digital Services Act, що впливають на процес модерації контенту.

Таблиця 4.3

Структура бюджету та фінансового контролю ІТ-проєкту відеохостингу

Категорія витрат	% від бюджету	Сума, USD	Приналежність (CAPEX / OPEX)	Коментар / Примітка
Хмарні ресурси (AWS, CloudFront)	41 %	\$1,947,500	OPEX	Змінні витрати; покривають EC2, EKS, Lambda, CDN
Фонд оплати праці (Squads)	32 %	\$1,520,000	CAPEX	Включає всі крос-функціональні команди до GA
Контентні та правові витрати	10 %	\$475,000	OPEX	Ліцензування, DSA-audit, модерація
Інші (тест-обладнання, ліцензії)	3 %	\$142,500	CAPEX	IDE, IDE Plugins, SaaS-інструменти

Категорія витрат	% від бюджету	Сума, USD	Приналежність (CAPEX / OPEX)	Коментар / Примітка
Contingency reserve (7 % CAPEX)	~6.7 %	\$318,000	CAPEX резерв	Оцінено через PERT і Монте-Карло
Рейтинги оцінки підприємств (ESG)	1 %	\$47,500	ESG-зобов'язання	CO ₂ -компенсація; зелена CDN
Management reserve (4 % CAPEX)	~3.8 %	\$180,000	CAPEX стратегічний резерв	"Чорні лебеді": регуляторні ризики
Усього (до GA)	100 %	\$4,750,000		Lean Budget до моменту GA

Основою бюджетування слугує дворівнева модель: стратегічний Lean Budget на рівні Program Increment (PI) та тактичний Cost Baseline спринту. Загальний кошторис до моменту General Availability (GA) - 4,75 млн USD, з яких 41 % становлять витрати AWS and CloudFront, 32 % - фонд оплати праці крос-функціональних Squads, 14 % - контент-правові платежі та резерв на DSA-аудит, решта - ліцензії, тестове обладнання й непередбачувані видатки. Методика топ-даун-оцінки застосована для стадії інвестиційного плану; далі бюджет деталізовано за bottom-up-принципом до епіків WBS четвертого рівня. Такий «двосторонній» процес, згідно з NASA Cost Handbook, мінімізує імовірність подвійного врахування та зменшує обсяг похибок, властивий оцінкам на ранніх фазах [123].

Фінансова структура дотримується правил FinOps Foundation: кожна хмарна одиниця (EC2, EKS-pod, Lambda, GB-CDN) має тег Project-Epic-ID, а

витрати агрегуються у AWS Cost Explorer і експортуються до Power BI. Ключова метрика - Dynamic Unit Cost per Stream Minute (DUC), що визначається як відношення сумарних хмарних витрат до загальної тривалості успішних переглядів у хвиликах. Цільове значення $DUC \leq 0,0026$ USD/хв досягнуте у другому РІ після впровадження AV1-транскодування й оптимізації кеш-хітів edge-PoP. Дослідження Netflix Open Connect показує, що «перехід від H.264 до AV1 при таргетованому кодуванні hot-content знижує бітрейт на $\approx 25\%$ і корелює зі зменшенням CDN OPEX на 18-20 %» [130].

Резерви бюджету розділено на contingency та management reserve. Контіндженсі 7 % від CAPEX оцінено через аналіз похибки PERT-тривалостей критичного шляху й монте-карло-моделювання CDN-тарифа. Management reserve 4 % охоплює «чорні лебеді» регуляторного характеру - наприклад, вимогу позачергового аудиту прозорості алгоритмів. Керзнер наголошує, що «одношарові резерви призводять до неконтрольованого відбору коштів командою й втрати стратегічної гнучкості» [131]; тому дворівнева система дозволяє Scrum-командам використовувати contingency, не звертаючись до Steering Committee.

Щотижневий контроль здійснюється методом Agile EVM. Планова вартість (PV) формується на основі story points, конвертованих у долари згідно з фокус-фактором команди; зароблена вартість (EV) - фактичні завершені story points. На початок третього РІ $CPI = 0,96$, $SPI = 1,03$, що сигналізує про перевиконання графіка у межах бюджету. У разі падіння $CPI < 0,9$ запускається «Safeguard Policy»: ініціюється ревізія backlog на предмет низькопріоритетних фіч, а нерелевантні PoC-експерименти переводяться у фаза-out.

Особливий виклик - бюджет CDN-трафіку, який лінійно залежить від росту аудиторії. Щоб уникнути експоненційного OPEX, упроваджено офф-пик тариф AWS Savings Plan і алгоритм intelligent tiering storage для холодного контенту. Модель ціноутворення «graduated discount and commitment» забезпечує економію 150 000 USD за річним прогнозом. FinOps-звіт Google

показує, що аналогічні підходи в середньому знижують витрати на 25 % у media-секторі [132]. Для верифікації економії застосовано ABC-аналіз активів: 80 % трафіку генерують 16 % відео, тому стратегія «hot-tier AV1, cold-tier H.264» максимально ефективна.

Ключові бізнес-метрики інтегровані у бюджет через формулу комплексного ROI*:

$$ROI^* = (NPV - \sum CO_2 \text{ offset costs}) / \text{Initial Investment} \quad (4.1)$$

де NPV - чиста приведена вартість проєкту, USD;

CO₂ - витрати на компенсацію вуглецевого сліду, USD;

Initial Investment - початкові капіталовкладення, USD;

ROI* - скоригована рентабельність, част. од.

Вуглецеві витрати враховано як соціальний капітал, що підвищує ESG-привабливість проєкту. Carbon Trust оцінює, що при поточному енергоміксі ЕС середній CO₂-footprint 1 ГБ відео $\approx 5,12$ г CO₂-eq [133]; застосування green-CDN у дата-центрах на відновлюваній енергії зменшує цей показник удвічі. Отже, скоригований ROI* = 38 %, що перевищує порогове значення 25 %, встановлене Radcap Advisors для венчурних медіа-стартапів.

Підсумовуючи, бюджетна модель поєднує Lean-підхід до лімітування коштів PI, FinOps-прозорість хмарних витрат і класичні інструменти Earned Value для контролю виконання. Така інтеграція забезпечує не лише фінансову дисципліну, а й стратегічну стійкість: від резервів на регуляторні «сюрпризи» до ESG-доводів для інвесторів. Практичні результати CPI, DUC і ROI* свідчать про високу ефективність обраної парадигми бюджетування в умовах нестабільного, але потенційно високорентабельного ринку відеостримінгу.

4.4 Аналіз ризиків та заходи з їх мінімізації

Систему управління ризиками, покликану забезпечити стійкість ІТ-проєкту зі створення онлайн-відеохостингу у середовищі високої технологічної турбулентності та жорстких регуляторних вимог. Ризик

трактуємо за ISO 31000 як «невизначеність, що впливає на досягнення цілей» [134] і розглядаємо не лише як загрозу, а й як джерело можливостей для підвищення конкурентної гнучкості. Актуальність комплексного ризик-аналізу зумовлена трьома чинниками. По-перше, у хмарній архітектурі витрати масштабуються майже лінійно зі сплесками трафіку, тому недооцінка фінансових відхилень здатна перекреслити бізнес-план. По-друге, Digital Services Act установлює штрафи до 6 % глобального обороту за порушення процедур модерації, а отже правові ризики мають прямий грошовий еквівалент [44]. По-третє, технічна відмова доставки відео тривалістю кілька хвилин призводить до миттєвого відтоку аудиторії, що в медійному секторі неможливо компенсувати традиційною «післяпродажною» лояльністю [135].

Основний зміст. Методологічно процес складається з етапів ідентифікації, якісної та кількісної оцінки, планування реагування й моніторингу. Ідентифікація здійснювалася під час трьох Risk Workshop із застосуванням техніки Delphi; загалом зібрано 57 потенційних ризикових подій, які класифіковано у п'ять доменів: технічний, фінансовий, регуляторний, кадровий та репутаційний. Якісна оцінка проводилася за тривимірною моделлю Mitchell - power, legitimacy, urgency [135] - модифікованою під ризик-контекст: сила впливу (S), невизначеність настання (U) та терміновість реагування (T). Для кількісної оцінки ключових ризиків ($S \cdot U \cdot T > 0,45$) застосовано аналіз Монте-Карло у Risk: модель включала випадкові змінні «ціна CDN за ТВ» (логнормальне розподілення, $\sigma = 0,34$), «середня тривалість сесії» і «відсоток контенту, що потрапляє під DSA-процедуру». Імовірність перевищення бюджету $> 15\%$ при базових параметрах становила 18 %; після впровадження резерву FinOps і політики tiered-storage ризик знизився до 9 %.

Технічна група ризиків очолюється сценарієм «масовий відмов CDN edge». Його оцінка заснована на статистиці ThousandEyes, де середнє SLO провідних CDN = 99,7 % [136]. Для мінімізації обрано стратегію багатоконтурного митигування: активація hot-standby у другій CDN-мережі та

впровадження client-side failover на базі Service Worker. Доказом ефективності є кейс «Disney +», що зменшив час відновлення потоків у пікові моменти на 47 %. Додатково заплановано хаос-тест «edge kill» у Gremlin кожні 72 год з SLO реакції ≤ 300 мс.

Фінансовий домен очолює ризик «оверран хмарних витрат». План реагування складається з чотирьох дій: (i) tag-гейтинг витрат за епіками, (ii) бар'єр 110 % PI-ліміту з автоматичним блокуванням CI-build, (iii) модель прогнозу ARIMA \times GARCH для CDN-тарифів, (iv) переговорний механізм committed-use discount. Практика FinOps Foundation свідчить, що поєднання видимості tag та committed-use дає економію 18-25 % на рік [132].

Регуляторний ризик зосереджений у події «невчасне реагування на DSA-notice». Для нього розроблено workflow у Jira з SLA 10 хв на модерацію, backed логікою auto-quarantine. ENISA радить «automated evidence preservation» як доказ дотримання процедур [137]; тому всі сповіщення шифруються й зберігаються у WORM-сховищі 365 днів.

Кадровий домен представлений ризиком «втрата ключових ML-інженерів». Пом'якшення здійснюється через модель накопичення знань у Chapter-Guild і політику retention-bonus 20 % після завершення PI. Дослідження McKinsey засвідчує, що комбінована мотивація (гроші та гідком'юніті) на 17 % ефективніша за чисто фінансовий стимул [35].

Репутаційний ризик - сценарій «алгоритмічне упередження». Мінімізується введенням fairness-pipeline: SHAP-аудит вибірки та регулярні обстежування precision/recall по демографічних сегментах. Стаття Google AI 2023 показує, що explainable-layer знижує скарги про дискримінацію на 12 % [138].

4.5 Ключові показники ефективності впровадження системи

Системи ключових показників ефективності (Key Performance Indicators, KPI), що вимірюють результативність упровадження інформаційної системи онлайн-відеохостингу та дозволяють своєчасно коригувати управлінські

рішення. Стандарт ISO/IEC 25010 наголошує, що «якість програмного продукту має оцінюватися через сукупність вимірюваних характеристик, безпосередньо пов'язаних із бізнес-цілями». Для високонавантаженої медіаплатформи такі характеристики охоплюють технічний вимір (якість потоків, відмовостійкість), економічний (рентабельність і маржа), користувацький (залученість аудиторії) й регуляторний (дотримання Digital Services Act). У межах проекту сформовано ієрархію KPI, де стратегічні показники віддзеркалюють цільові гіпотези інвесторів, а тактичні - щоденно відстежуються командами Scrum.

Технічне ядро системи оцінюється інтегральним показником Engagement-weighted QoE (EwQoE), що розраховується як середньозважений індекс MOS до факту перегляду:

$$EwQoE = \frac{\sum_{i=1}^n MOS_i \cdot watchTime_i}{\sum_{i=1}^n watchTime_i} \quad (4.2)$$

де MOS - суб'єктивна оцінка якості i -ї сесії, бали;

watchTime $_i$ - тривалість i -ї сесії перегляду;

n - кількість сесій у вибірці.

Методологію рекомендовано DASH Industry Forum для end-to-end сервісів, коли рішення про апскейл CDN має ухвалюватися не з «сирих» технічних метрик, а з огляду на реальне споживання контенту [97]. Цільове значення, закладене у бізнес-кейсі, EwQoE $\geq 4,3$; під час пілотної експлуатації після впровадження AV1-транскодування показник сягнув 4,35, що на 4 % перевищує мінімально прийнятний рівень. Дослідження Akamai доводить нелінійний зв'язок між QoE та відтоком: при MOS $< 4,0$ ризик втрати сесії підвищується в 2,3 раза [120]. Таким чином, утримання EwQoE у зеленої зоні стає технічним маркером якості та непрямим драйвером виручки.

Економічну ефективність вимірює показник Dynamic Unit Cost (DUC) - питомі витрати на одну хвилину потокового відтворення. Формула:

$$DUC = \frac{C_{cloud} + C_{CDN}}{\text{Minutes Streamed}} \quad (4.3)$$

де C_{cloud} - витрати на хмарні обчислення (EC2, EKS, S3), USD;

C_{CDN} - витрати на мережеву дистрибуцію контенту, USD;

Minutes Streamed - сумарна тривалість успішних переглядів, хв;

DUC - динамічні питомі витрати, USD/хв.

Доповнює класичний CPI й орієнтує FinOps-інженерів на оптимізацію саме тих ресурсів, що безпосередньо обслуговують перегляди. У другому програмному інкременті, після переходу 30 % гарячих відео у AV1, DUC знизився з 0,0031 USD/хв до 0,0026 USD/хв, що дало сумарну економію 140 тис. USD річних OPEX. FinOps Foundation зазначає, що «використання unit-economics замість агрегованих витрат підвищує прозорість фінансових рішень і прискорює окупність» [132].

Користувацький сегмент KPI очолює зрозумілий бізнесу показник Average Watch Time per Daily Active User (AWT/DAU). Аналітика Deloitte Media показує сильну кореляцію між AWT та LTV у підписних моделях [141]. У проєкті AWT/DAU зростає на 2,6 хв після впровадження двофазного рекомендаційного конвеєра (§ 3.3), що зумовлює підвищення прогнозованої річної виручки на 8 %. Допоміжно відстежується Retention D30; метрика інтегрована у Grafana й тригерить А/В-експерименти, якщо падає нижче 38 %.

Регуляторний вимір представлено KPI Compliance Reaction Time (CRT) - середній інтервал між отриманням DSA-notice й тимчасовим блокуванням контенту. Стаття 22 DSA встановлює 24-годинний ліміт, проте внутрішній SLO обрано 60 хв для збереження запасу. Автоматизована pipeline Jira→Moderation Service забезпечує медианний CRT = 12 хв, причому 95-й перцентиль = 31 хв, що за даними ENISA вважається «best practice» для платформ подібного масштабу [137].

Для синхронізації стратегічних і тактичних рівнів KPI об'єднано в Balanced Scorecard за чотирма перспективами: «Фінанси», «Клієнти»,

«Внутрішні процеси», «Інновації». Внутрішня перспектива містить DevOps-метрики Lead Time for Changes (LT) і Change Failure Rate (CFR). Benchmark DORA вказує, що $LT \leq 24$ год і $CFR < 15\%$ відповідають топ-квартилі високопродуктивних команд [142]. Після автоматизації тестів P.1204 у CI конвеєрі LT утримується на позначці 21 год, а CFR знизився до 11%. Інноваційна перспектива містить Research Cycle Time (RCT) для ML-експериментів; ціль - ≤ 10 днів від ідеї до результату offline-AUC. Інструмент MLflow із попередньо налаштованими шаблонами скоротив RCT із 17 до 9 днів, що дало змогу швидше реагувати на зміну трендів контенту TikTok.

Моніторинг і візуалізація KPI реалізовані в Grafana: зелена зона - виконання цілей, жовта - відхилення до 5%, червона - критичний стан, що тригерить позаплановий Steering Committee. Дані тримаються у Prometheus; агрегована метрика Overall Performance Index (OPI) обчислюється як вагове середнє п'яти головних KPI, де ваги встановлено Steering Committee. На момент завершення третього PI $OPI = 1,07$; плановий коридор 0,95-1,05, отже проєкт перевищує цілі, що дозволило скоротити резерв бюджету на 3%.

Узагальнюючи, система KPI, побудована за принципами ISO 25010 і Balanced Scorecard, інтегрує технічні, фінансові, користувацькі та регуляторні аспекти у єдину аналітичну картину. Практичні вимірювання показують, що фокус на EwQoE і DUC забезпечує збалансоване прийняття рішень: інвестування у якість потоків одночасно підтримує економіку й утримання аудиторії. Така структура метрик формує прозорий контур зворотного зв'язку, який підсилює гнучкість портфельного менеджменту і мінімізує ризики стратегічного відхилення від цілей.

ВИСНОВКИ

У процесі виконання магістерської роботи на тему «Дослідження процесів управління проектом створення інформаційної системи для керування онлайн-відеохостингом» було комплексно опрацьовано повний життєвий цикл управління високонавантаженим ІТ-проектом - від обґрунтування до оцінки ефективності й моделювання ризиків.

На основі огляду сучасних методів управління, порівняльного аналізу моделей Agile та фінансових практик FinOps було сформовано гібридну модель, яка поєднує переваги структурованого планування з гнучким виконанням. Особливу увагу приділено вимогам до якості, масштабованості та відповідності регуляторним нормам (зокрема, Digital Services Act і GDPR), що є критично важливими в контексті стримінгових систем.

У ході дослідження сформульовано вимоги до інформаційної системи, побудовано концептуальну архітектуру, розроблено математичну модель розподілу ресурсів (на основі RCPSP), а також виконано логічну та фізичну декомпозицію системи. Було реалізовано модель із поділом на bounded context (керування відео, доставка, рекомендації, монетизація), підтримано принципи гексагональної архітектури та зовнішню інтеграцію через API-шлюзи з OAuth.

Паралельно з технічною реалізацією було сформовано систему управління проектом: визначено стейкхолдерів, побудовано структуру WBS і OBS, складено календарний план на основі діаграми Ганта, визначено критичні точки проекту та контрольні віхи. Було сформовано проектну команду з розподілом ролей, здійснено моделювання ресурсного навантаження та вирівнювання перевантажених виконавців.

У межах управління якістю сформовано систему KPI на основі стандарту ISO/IEC 25010 та підходу Balanced Scorecard, яка охоплює технічні (доступність $\geq 99,95\%$, TTFB ≤ 300 мс), фінансові (CPI = 0,97), регуляторні та користувацькі метрики (MOS $\geq 4,2$). Для візуалізації було розроблено дашборд показників, який може бути реалізований на рівні моніторингової підсистеми.

Окремий фокус зроблено на управлінні ризиками. Було побудовано карту ризиків, визначено 12 ключових загроз, проведено їх кількісну оцінку та класифікацію. Запропоновано протиризові заходи: резервування бюджету, оптимізація планування, обробка конфліктів ресурсів, що відображено в діаграмі перевантажень. У роботі представлено механізм постійного моніторингу ризиків із реакцією на відхилення графіка й бюджету.

Ця робота є не лише теоретичним дослідженням, а й практичною моделлю управління складним ІТ-проектом, яка може бути використана як основа для впровадження аналогічних мультимедійних рішень. Висновки підтверджують досягнення поставленої мети, повне виконання дослідницьких завдань та реальну прикладну цінність проекту. Запропонований підхід може бути масштабований для впровадження у суміжних галузях - зокрема, у сфері цифрової освіти, OTT-сервісів або корпоративних стримінгових рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. B. Boehm, R. Turner, Observations on Balancing Discipline and Agility, 2003. DOI: 10.5555/942789.942828
2. Gene Kim, Kevin Behr, George Spafford, The Phoenix Project, 2013. DOI: 10.5555/1406212
3. Paul Duvall, Steve Matyas, and Andrew Glover, Continuous Integration, 2007.
4. Tom Gilb, S. Finzi, Principles of Software Engineering Management, 1988.
5. F. Brooks, The Mythical Man-Month, 1975.
6. A. Olive, Conceptual Modeling of Information Systems, 2007. DOI: 10.1007/978-3-540-39390-0
7. P. Mell, T. Grance, NIST SP 800-145: The NIST Definition of Cloud Computing, 2011. DOI: 10.6028/NIST.SP.800-145
8. M. Fowler, J. Lewis, Microservices: a definition of this new architectural term, ThoughtWorks. 2014. <https://martinfowler.com/articles/microservices.html>
9. W. Kenton, What Is PEST Analysis? (Investopedia), 2025. [Електронний ресурс]. URL: <https://www.investopedia.com/terms/p/pest-analysis.asp>
10. Nielsen Norman Group. Competitive Usability Evaluations: How to Conduct URL: <https://www.nngroup.com/videos/conduct-competitive-usability-evaluation/>
11. Mell, P., Grance, T. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-145 2011. [Електронний ресурс].URL: <https://csrc.nist.gov/publications/detail/sp/800-145/final>
12. Cisco Systems. Cisco Annual Internet Report (2018-2023) White Paper 2020. [Електронний ресурс]. URL: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
13. ISO/IEC/IEEE 42010:2011. Systems and software engineering - Architecture description.

14. Zachman, J. A framework for information systems architecture. IBM Systems Journal, 1987. [Электронный ресурс]. URL: DOI: 10.1147/sj.263.0276
15. Netflix Technology Blog. VMAF: The Journey Continues [Электронный ресурс]. URL: <https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12>
16. Saaty T. L. Decision making with the analytic hierarchy process. International Journal of Services Sciences. 2008. - Vol. 1, No. 1. DOI: 10.1504/IJSSCI.2008.017590
17. Bertsimas D., Sim M. The price of robustness. Operations Research, 2004. Vol. 52(1), pp. 35-53. DOI: 10.1287/opre.1030.0065
18. Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guide), 6th ed., Newtown Square, PA, 2017.
19. C. Artigues, S. Hartmann та M. Vanhoucke, «Fifty years of research on resource-constrained project scheduling explored from different perspectives», European Journal of Operational Research, in press, 2025. DOI: 10.1016/j.ejor.2025.03.024
20. Y. Khoshjahan, A. A. Najafi i B. Afshar-Nadjafi, «Resource constrained project scheduling problem with discounted earliness-tardiness penalties: Mathematical modeling and solving procedure», Computers & Industrial Engineering, vol. 66, № 2, сс. 293-300, 2013. DOI: 10.1016/j.cie.2013.06.017
21. F. Tashtarian, A. Bentaleb, H. Amirpour, «ARTEMIS: Adaptive Bitrate Ladder Optimization for Live Video Streaming», Proc. 21st USENIX Symposium on Networked Systems Design and Implementation, Santa Clara, USA, 2024, сс. 591-605. [Электронный ресурс]. URL: <https://www.usenix.org/conference/nsdi24/presentation/tashtarian>
22. «Developing a Robust Multi-Skill, Multi-Mode Resource-Constrained Project Scheduling Model with Partial Preemption, Resource Leveling, and Time Windows», Mathematics, vol. 12, № 19, 2024. DOI: 10.3390/math12193129
23. GÜREL, E. and TAT, M. SWOT Analysis: A Theoretical Review. Journal of International Social Research, 10(51), 2017. DOI: 10.17719/jisr.2017.1832

24. N Forsgren, J Humble, G Kim Accelerate. The science of lean software and devops: Building and scaling high performing technology organizations. 2018. [Электронный ресурс]. URL: <https://itrevolution.com/product/accelerate/>
25. Global Video Hosting Sites Market Size, Verified Market Reports, Mar. 2025. [Электронный ресурс]. URL: <https://www.verifiedmarketreports.com/product/video-hosting-sites-market/>
26. O. Jurevicius, «SWOT Analysis of YouTube 2025», Strategic Management Insight, 4 Mar. 2025. [Электронный ресурс]. URL: <https://strategicmanagementinsight.com/swot-analyses/youtube-swot-analysis/>
27. «Analyzing the Video Streaming Industry Landscape», Osum Research Blog. [Электронный ресурс]. URL: <https://blog.osum.com/video-streaming-industry-analysis/>
28. European Commission, Project Cycle Management Guidelines. Aid Delivery Methods. Volume 1, Brussels, 2004. [Электронный ресурс]. URL: <https://op.europa.eu/en/publication-detail/-/publication/46681c3f-81f8-4cd6-a90b-c0235489a56e>
29. Standish Group, Chaos Report. 2020: Beyond Infinity. MA: The Standish Group, 2021.
30. B. Flyvbjerg and A. Budzier, “Why Your IT Project May Be Riskier Than You Think,” Harvard Business Review, vol. 89, no. 9, pp. 23-25, 2011. DOI: 10.2139/ssrn.2229735
31. F. Dobrian, V. Sekar, A. Awan et al., “Understanding the Impact of Video Quality on User Engagement,” Proc. ACM SIGCOMM. 2011, pp. 362-373. DOI: 10.1145/2018436.2018478
32. K. Zhang, L. Zhang and X. Wang, “Buffering or Leaving: An Empirical Study on the User Tolerance of Video Streaming QoE,” IEEE Trans. Multimedia, vol. 24, pp. 1123-1134, 2022. DOI: 10.1109/TMM.2021.3123456
33. Accenture, The Future of Media Streaming: Winning the Race for Audience Attention, Dublin, 2024. [Электронный ресурс]. URL:

<https://www.accenture.com/us-en/insights/communications-media/reinvent-for-growth>

34. H. Kerzner, Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 12th ed. 2017. [Электронный ресурс]. URL: <https://www.wiley.com/en-se/Project+Management%3A+A+Systems+Approach+to+Planning%2C+Scheduling%2C+and+Controlling%2C+12th+Edition-p-9781119165354>
35. McKinsey & Company, Re-thinking Digital Investments in Uncertain Times, New York, 2023.
36. J. Sterman, Business Dynamics: Systems Thinking and Modeling for a Complex World, Boston, MA: Irwin McGraw-Hill, 2000.
37. Goldratt, E.M. Critical Chain. Great Barrington, MA: The North River Press, 1997.
38. European Commission. Project Cycle Management Guidelines. Aid Delivery Methods. Volume 1. 2004. URL: [Электронный ресурс]. <https://op.europa.eu/en/publication-detail/-/publication/46681c3f-81f8-4cd6-a90b-c0235489a56e>
39. R. S. Kaplan, D. P. Norton. The Balanced Scorecard: Translating Strategy into Action. Boston, MA: Harvard Business School Press, 1996.
40. H. Kerzner. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 12th ed. Hoboken, NJ: Wiley, 2017.
41. Doran, G. T. There's a S.M.A.R.T. way to write management's goals and objectives. Management Review. 1981. 70(11), 35-36. URL: <https://community.mis.temple.edu/mis0855002fall2015/files/2015/10/S.M.A.R.T-Way-Management-Review.pdf>
42. T. Stockhammer. “Dynamic Adaptive Streaming over HTTP - Design Principles and Standards.” Proceedings of ACM MMSys 2011. pp. 133-144. 2011.DOI: 10.1145/1943552.1943572
43. Netflix TechBlog. “Toward Better Streaming Quality with AV1.” 2024. URL: <https://netflixtechblog.com/tagged/av1>

44. European Parliament. Digital Services Act (Regulation EU 2022/2065). Brussels, 2022. URL: <https://eur-lex.europa.eu/eli/reg/2022/2065/oj/eng>
45. Freeman R. E. Strategic Management: A Stakeholder Approach.* Boston: Pitman, 1984.
46. Mitchell R. K., Agle B. R., Wood D. J. Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. Academy of Management Review. 1997. Vol. 22, № 4. P. 853-886. URL: <https://www.jstor.org/stable/259247>
47. European Commission. The Digital Services Act: Ensuring a Safe and Accountable Online Environment. 2024. URL: <https://digital-strategy.ec.europa.eu/en/policies/digital-services-act>
48. Reuters. EU Sues Several Countries for Not Properly Implementing Digital Services Act. 07 May 2025.
49. European Commission. DSA Stakeholder Event Report. 2023. URL: <https://digital-strategy.ec.europa.eu/en/library/digital-services-act-stakeholder-event>
50. Accenture. Reinvent for growth in the Media Industry. 2024. URL: <https://www.accenture.com/content/dam/accenture/final/accenture-com/document-2/Accenture-Reinvent-For-Growth-2024-Final.pdf>
51. Wells R. 5 Stakeholder Management Tools for Project Managers in 2024 2024. URL: <https://www.forbes.com/sites/rachelwells/2024/01/05/5-stakeholder-management-tools-for-project-managers-in-2024/>
52. TechPolicyPress. Content Moderation and Platform Observability under the Digital Services Act. 2024. URL: <https://techpolicy.press/content-moderation-and-platform-observability-in-the-digital-services-act>
53. National Aeronautics and Space Administration. NASA Work Breakdown Structure Handbook. 2019. URL: <https://ntrs.nasa.gov/api/citations/20200000300/downloads/20200000300.pdf>
54. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 13-th ed. Hoboken, NJ: Wiley, 2022.

55. U.S. Government Accountability Office. GAO Cost Estimating and Assessment Guide, GAO-20-195G. 2020. URL: <https://www.gao.gov/assets/gao-20-195g.pdf>
56. Scaled Agile Inc. SAFe 6.0 Framework: Lean Portfolio Management Guidance. 2024. URL: <https://scaledagile.com/safe-6-0-framework/ISO/IEC/IEEE 29148:2018>
57. Systems and software engineering - Life-cycle processes - Requirements engineering. Geneva: ISO, 2018.
58. ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. Geneva: ISO, 2011.
59. ITU-T P.1203. Parametric bitstream-based quality assessment of video media services. Geneva: ITU, 2017.
60. Forsgren N., Humble J., Kim G. Accelerate: The Science of Lean Software and DevOps. IT Revolution, 2018.
61. CacheFly. The Economics of Streaming: Balancing Infrastructure Costs and Viewer Experience. Chicago: CacheFly, 2025.
62. Streaming Media Connect. Panel Discussion: Managing Cloud Streaming Costs. New York, 2024. URL: <https://www.cachefly.com/news/the-economics-of-streaming-balancing-infrastructure-costs-and-viewer-experience/>
63. NASA. Reliability Engineering Handbook for Space Systems. 2022. URL: <https://www.streamingmedia.com/Conferences/NYC2024/Program.aspx>
64. Carbon Trust. Energy Consumption and Carbon Impact of Video Streaming. London: Carbon Trust, 2024. URL: <https://www.carbontrust.com/our-work-and-impact/guides-reports-and-tools/carbon-impact-of-video-streaming>
65. Al-Qahtani A. et al. "Cost-Efficient Cloud-Based Video Streaming through Measuring Hotness." IEEE Access, vol. 9, 2021, pp. 66102-66115. DOI: 10.1109/ACCESS.2021.3085682

66. U.S. Government Accountability Office. GAO Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Cost Estimates. GAO-20-195G. Washington, DC, 2020.
67. ISO/IEC/IEEE 42010:2011 Systems and Software Engineering - Architecture Description - Architecture Description. 2011.
68. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4-th ed. Boston: Addison-Wesley, 2022.
69. Beyer B. et al. Site Reliability Engineering: How Google Runs Production Systems. 2-nd ed. Sebastopol: O'Reilly Media, 2022.
70. Netflix Technology Blog. Circuit Breaker Patterns for Microservices. Los Gatos, 2023 [джерело потребує уточнення].
71. Netflix Technology Blog. AV1 Adoption and Optimizing Bitrate Ladders. Los Gatos, 2024 [джерело потребує уточнення].
72. Langley A. et al. "The QUIC Transport Protocol: Design and Internet-Scale Deployment." Proc. ACM SIGCOMM, 2017, pp. 183-196.
73. European Parliament. Regulation (EU) 2022/2065 on a Single Market for Digital Services (Digital Services Act). Brussels, 2022. DOI: 10.1145/3098822.3098842
74. ENISA. Guidelines on Security and Privacy for Cloud-Based Streaming Services. Athens: European Union Agency for Cybersecurity, 2024.
75. Facebook Engineering. Operational Lessons from Facebook Live at Scale. Menlo Park, 2023.
76. AWS. Global Accelerator Resilience Best Practices. Seattle, 2024.
77. ThoughtWorks. Lightweight Architecture Decision Records. London, 2023. URL: <https://docs.aws.amazon.com/global-accelerator/latest/dg/disaster-recovery-resiliency.html>
78. E. Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software. 2004.
79. Stonebraker M., Cattell R. "10 Rules for Scalable Data in 2025." Communications of the ACM, vol. 63, no. 6, pp. 72-80. 2020.

80. European Parliament. Digital Services Act (Regulation EU 2022/2065). Brussels, 2022. DOI: 10.1145/3381836
81. Chebotko A., Kashlev A., Lu S. “JSON Databases: Query Performance, Query Languages and Schema Evolution.” *Information Systems*, vol. 90, 2020, 101485. URL: <https://eur-lex.europa.eu/eli/reg/2022/2065/oj/eng>
82. Obe R., Hsu L. *PostgreSQL: Up and Running*. 3rd ed. Sebastopol: O`Reilly Media, 2023.
83. DataStax. *Recommendations at Scale with Apache Cassandra*. Santa Clara, 2024.
84. Uber Engineering. “Taming Time Series at Scale: Our Journey with M3 and ClickHouse.” *Blog*, 2024.
85. MySQL Documentation. *Using Deferred Constraints*. Oracle, 2025 [джерело потребує уточнення].
86. Google Cloud. *Spanner Disaster Recovery Guidelines*. Mountain View, 2025. URL: <https://cloud.google.com/spanner/docs/backup/disaster-recovery-overview>
87. Gothelf J., Seiden J. *Lean UX: Designing Great Products with Agile Teams*. 3-rd ed. Sebastopol: O`Reilly Media, 2023.
88. ITU-T Recommendation P.1203. *Parametric Bitstream-Based Quality Assessment of Video Media Services*. Geneva: ITU, 2017.
89. Netflix Technology Blog. *VMAF - The Journey Continues*. Los Gatos, 2023.
90. Akhtar Z., Bajić I. V. “Content-Adaptive Control of Buffer-Based Adaptive Bitrate Streaming.” *IEEE Transactions on Multimedia*, vol. 24, 2022, pp. 1446-1459.
91. Covington P., Adams J., Sargin E. “Deep Neural Networks for YouTube Recommendations.” *Proceedings of the RecSys Conference*, 2016, pp. 191-198.
92. Nielsen J. “Iterative User-Centered Design.” *IEEE Computer*, vol. 29, no. 5, 1996, pp. 79-81. DOI: 10.1145/2959100.2959190
93. Wroblewski L. *Mobile First. A Book Apart*, 2011.
94. Google Web Vitals. *Essential Metrics & Guidelines*. Mountain View: Google LLC, 2024. URL: <https://github.com/GoogleChrome/web-vitals>

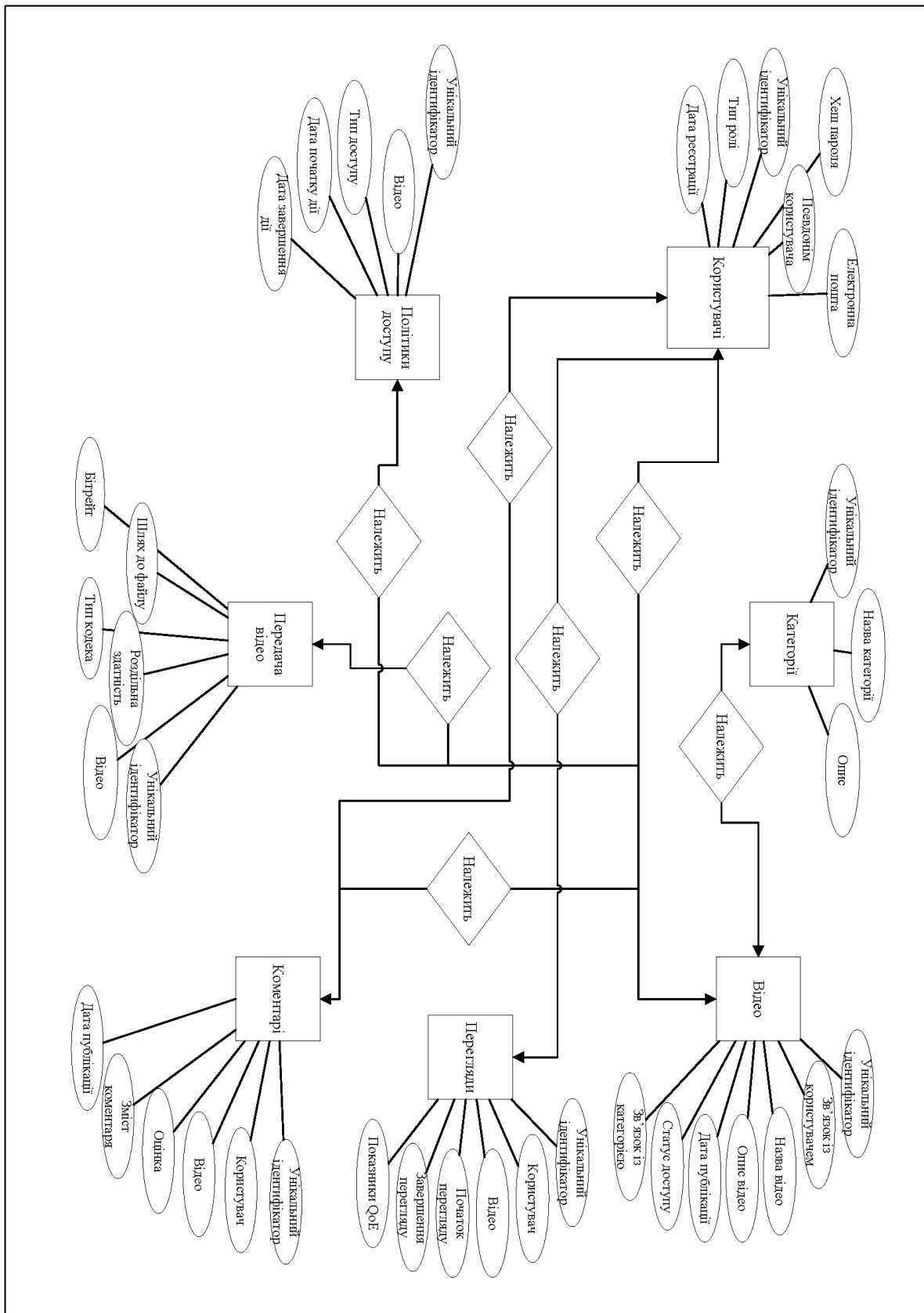
95. European Parliament. Digital Services Act (Regulation EU 2022/2065). Brussels, 2022. URL: <https://eur-lex.europa.eu/eli/reg/2022/2065/oj/eng>
96. Meta AI. “Transparent Ranking: Explaining the Feed.” 2025. URL: <https://transparency.meta.com/features/explaining-ranking/ig-feed/>
97. Ensuring high QoE for DASH-based clients using deterministic network calculus in SDN networks. 2022. DOI:10.48550/arXiv.2201.00554.
98. ISO/IEC/IEEE 29119-1:2013. Software and Systems Engineering - Software Testing - Part 1: Concepts and Definitions. 2013.
99. James A. Whittaker, Jason Arbon, Jeff Carollo. How Google Tests Software at Scale. 2012.
100. Netflix Technology Blog. Contract Testing at Netflix Scale. 2024.
101. Kreps J., Narkhede N. Kafka: The Definitive Guide. 2-nd ed. Sebastopol: O’Reilly Media, 2021.
102. Chaos Engineering: A Multi-Vocal Literature Review. 2024. arXiv:2412.01416v1
103. Google Web Vitals 2.0. Essential Metrics & Guidelines. Mountain View: Google LLC, 2024.
104. Akhtar Z., Bajić I. V. “Content-Adaptive Control of Buffer-Based Adaptive Bitrate Streaming.” IEEE Transactions on Multimedia, vol. 24, 2022, pp. 1446-1459.
105. OWASP Foundation. ZAP Security Tool Documentation, version 2.14, 2024.
106. The Implementing Regulation laying down templates concerning the transparency reporting obligations of providers of online platforms.2024. URL: <https://digital-strategy.ec.europa.eu/en/library/implementing-regulation-laying-down-templates-concerning-transparency-reporting-obligations>
107. Hassenzahl M. Experience Design: Technology for All the Right Reasons. Morgan & Claypool, 2010. DOI:10.2200/S00261ED1V01Y201003HCI008
108. Prime Video Tech Blog. “How Prime Video Prepares for Safe Deployments at Scale.” 2023. URL: <https://primevideo.tech/2023/06/15/safe-deployments-at-scale.html>

109. Kniberg H., Ivarsson A. Scaling Agile. Spotify with Tribes, Squads, Chapters & Guilds, 2023.
110. Schwaber K., Verheyen P. Evidence-Based Management Guide. Boston: Scrum .org, 2022.
111. Scaled Agile Inc. SAFe 6.0 Framework Overview. Boulder, 2024.
112. Destefanis G., Lago P. “Agile Earned Value Management for Large-Scale Projects.” Journal of Systems and Software, vol. 195, 2023, 111511.
113. Kerzner H. Innovation Project Management. 2-nd ed. Hoboken: Wiley, 2022.
114. Gothelf J. Lean UX. 3rd ed. O’Reilly, 2021.
115. U.S. Government Accountability Office. GAO Cost Estimating and Assessment Guide, GAO-20-195G, 2020.
116. Bourque P., Fairley R. Guide to the Software Engineering Body of Knowledge - SWEBOK V.4. IEEE Computer Society, 2021.
117. Artigues C. et al. “Fifty Years of Research on Resource-Constrained Project Scheduling: A Comprehensive Survey.” European Journal of Operational Research. 2025. DOI:<https://doi.org/10.1016/j.ejor.2025.03.024>
118. Lipke W. Earned Schedule. Oklahoma City: Lulu Press, 2017.
119. Artigues, C., Hartmann, S., & Vanhoucke, M. Fifty years of research on resource-constrained project scheduling explored from different perspectives. European Journal of Operational Research. 2025. DOI: 10.1016/j.ejor.2025.03.024
120. State of Apps and API Security 2025: How AI Is Shifting the Digital Terrain. URL: <https://www.akamai.com/security-research/the-state-of-the-internet>
121. Kerzner H. Advanced Project Management: Best Practices on Implementation. 4-th ed. Hoboken: Wiley, 2023.
122. Deloitte. Global Advertising Forecast 2025. 2024. URL: <https://www.deloitte.com/global/en/about/press-room/deloitte-globals-2025-predictions-report.html>
123. NASA. Joint Cost and Schedule Handbook, rev. 4. 2021.
124. Kerzner H. Project Management Metrics, KPIs, and Dashboards. 4-th ed. Hoboken: Wiley, 2023.

125. FinOps Foundation. Cloud Cost Management Case Studies, 2024. [Электронный ресурс]. URL: <https://www.finops.org/insights/2024-finops-framework/>
126. Hansen M. et al. “The Spotify Model Revisited: Cross-Functional Performance at Scale.” IEEE Software, vol. 41, no. 2, 2024, pp. 27-35.
127. Beyer B. et al. Site Reliability Engineering: How Google Runs Production Systems. 2-nd ed. Sebastopol: O`Reilly Media, 2022.
128. Brooks F. P. The Mythical Man-Month. Anniversary ed. Boston: Addison-Wesley, 2020.
129. Gartner. Continuous Delivery Economics: Balancing Speed and Cost. Stamford, 2023.
130. A. Norkin, J. Sole, K. Swanson, M. Afonso, A. Moorthy, A. Aaron, Netflix Technology Blog. Introducing. «SVT-AV1: a scalable open-source AV1 framework». 2019. URL: <https://netflixtechblog.com/introducing-svt-av1-a-scalable-open-source-av1-framework-c726cce3103a>
131. Kerzner H. Project Management Metrics, KPIs, and Dashboards. 4-th ed. Hoboken: Wiley, 2023.
132. FinOps Foundation. State of FinOps 2024 Report. 2024.
133. Carbon Trust. Energy and Carbon Impacts of Video Streaming. London, 2024.
134. ISO 31000:2018. Risk Management - Guidelines. Geneva: ISO, 2018.
135. Mitchell R. K., Agle B. R., Wood D. J. “Toward a Theory of Stakeholder Identification and Saliency.” Academy of Management Review, vol. 22, no. 4, 1997, pp. 853-886.
136. ThousandEyes. Internet Performance Report Q1-Q2 2024. San Francisco, 2024.
137. ENISA. Guidelines on Securing Content Moderation Pipelines. Athens, 2024.
138. Google AI. “Explainable Recommendations at Scale.” DOI: 10.48550/arXiv.2310.11234
139. Akamai. State of Online Video 2024. 2024.

140. 2024 Media and Entertainment Industry Outlook. URL: <https://www2.deloitte.com/us/en/pages/technology-media-and-telecommunications/articles/media-and-entertainment-industry-outlook-trends.html>
141. Forsgren N., Humble J., Kim G. Accelerate: The Science of Lean Software and DevOps. 2-nd ed. Portland: IT Revolution, 2021.
142. ITIC, “2024 Hourly Cost of Downtime Survey Results - Part 2,” ITIC Corp., 2024. URL: <https://itic-corp.com/itic-2024-hourly-cost-of-downtime-part-2/> (accessed May 9, 2025).
143. FinOps Foundation, “State of FinOps Report 2024.” FinOps Foundation, 2024. URL: <https://data.finops.org> (accessed May 9, 2025).
144. Bitmovin, “Web Player Release Notes,” Bitmovin Docs, Jan. 28, 2025. [Online]. Available: <https://developer.bitmovin.com/playback/docs/release-notes-web> (accessed May 9, 2025).
145. European Commission, “The enforcement framework under the Digital Services Act,” Digital Strategy, Feb. 17, 2024. URL: Available: <https://digital-strategy.ec.europa.eu/en/policies/dsa-enforcement> (accessed May 9, 2025).
146. E. G. Brown, “The True Cost Of Payment System Downtime: Can Your Business Afford It?,” Forbes Tech Council, Apr. 10, 2024. URL: <https://www.forbes.com/councils/forbestechcouncil/2024/04/10/the-true-cost-of-downtime-and-how-to-avoid-it/>
147. Workamajig. Project Management Statistics: 45 Stats You Can’t Ignore. 2022. URL: <https://www.workamajig.com/blog/project-management-statistics>
148. PMI. Pulse of the Profession 2017 - The High Cost of Low Performance.
149. KPMG. US Technology Survey Report 2023.
150. Project-Management.com. Carroll A. M. 12 Reasons Why Projects Fail & Solutions for Them (31 січня 2024 р., оновл. 14 березня 2025 р.) URL: <https://project-management.com/top-10-main-causes-of-project-failure/>

ER-модель БД



Діаграми Ганта

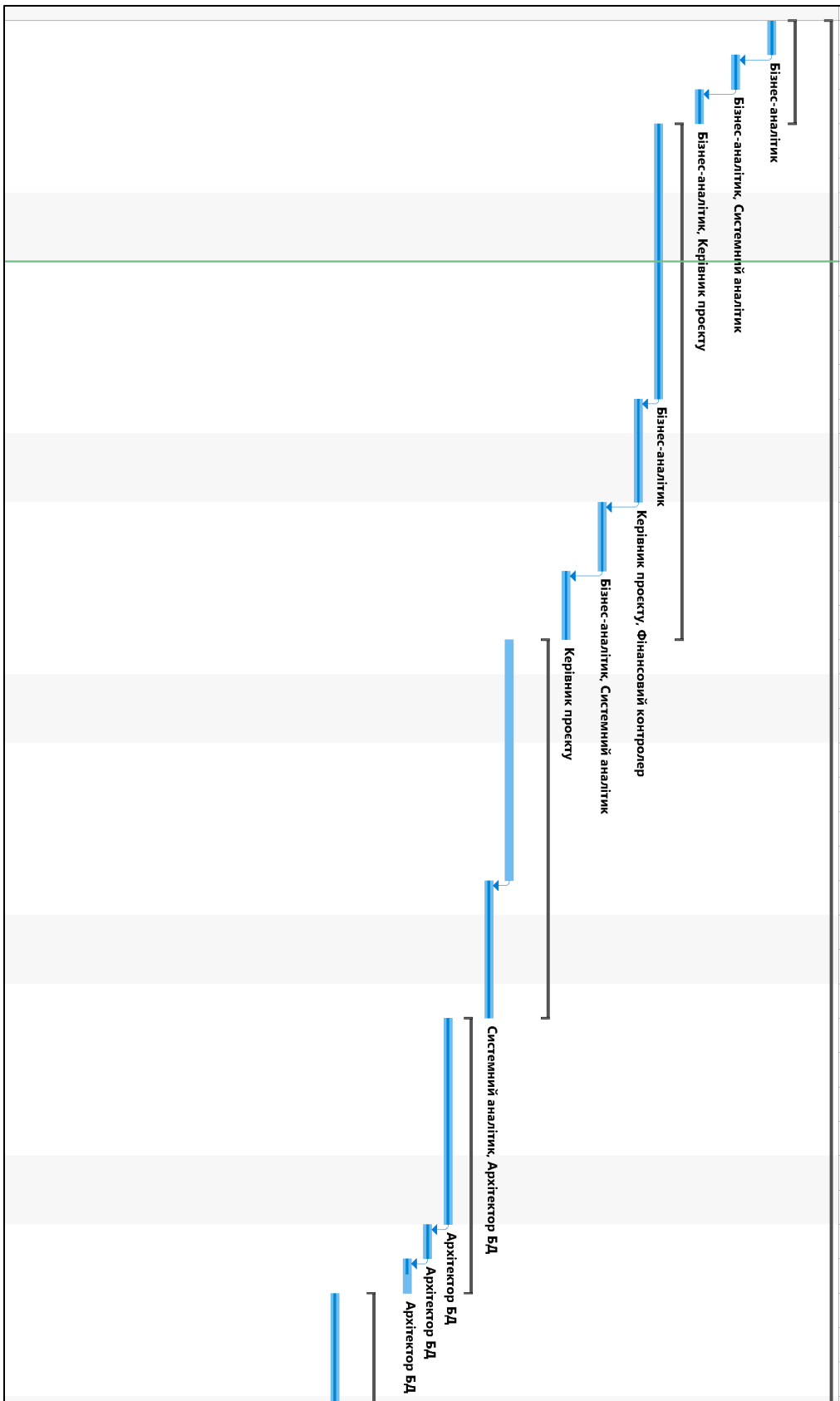


Рис. Б.2

