

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

СЕНЬКІВ Мар'ян Миколайович

Управління проектом розробки веб-орієнтованої інформаційної системи / Project management of web-based information system development

спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Управління проектами

Кваліфікаційна робота

Виконав студент групи КНУПм-21
М. М. Сеньків

Науковий керівник:
к.т.н., доцент І.В. Турченко

Кваліфікаційну роботу
допущено до захисту:

«___» _____ 20__ р.

В.о. завідувача кафедри

_____ Н.В. Дзюбановська

ТЕРНОПІЛЬ - 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «магістр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Управління проектами

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
«__» _____ 20__ року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
СЕНЬКІВ Мар'ян Миколайович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

**Управління проектом розробки веб-орієнтованої інформаційної системи /
Project management of web-based information system development**

керівник роботи к.т.н., доцент І. В. Турченко

затверджені наказом по університету від 20 грудня 2024 року № 938

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

- провести аналіз предметної області та сучасних підходів до управління проектами веб-розробки;
- виконати порівняльний аналіз методологій та обґрунтувати вибір оптимального підходу;
- розробити модель планування робіт на основі Scrumban, адаптовану до специфіки веб-проектів;
- визначити методику оцінювання вартості та підхід до управління ризиками проекту;
- реалізувати процеси планування та виконання проекту у програмному середовищі управління проектами;
- ідентифікувати стейкхолдерів, сформулювати команду, розрахувати вартість проекту та провести оцінку якості продукту.

5. Перелік графічного матеріалу у роботі

- дворівнева модель планування робіт з використанням Scrumban;
- схема алгоритму оцінки вартості проекту розробки інформаційної системи.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 20.05.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 28.10. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 11.11.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 25.11.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту.	до 1.12.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 04.12.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 14.12. 2025 р.	

Студент _____ М.М. Сеньків
підпис

Керівник роботи _____ к.т.н., доцент І. В. Турченко
підпис

РЕЗЮМЕ

Кваліфікаційна робота на тему «Управління проектом розробки веб-орієнтованої інформаційної системи» на здобуття освітнього ступеня «Магістр» зі спеціальності 122 «Комп'ютерні науки» освітньо програми “Управління проектами” написана обсягом 74 сторінки і містить 8 ілюстрацій, 10 таблиць, 4 додатки та 51 використаних джерел.

Метою роботи є розробка підходу до управління проектом розробки веб-орієнтованої інформаційної системи.

Методами дослідження, що використовуються в роботі, є: системний аналіз (для вивчення предметної області та архітектурних підходів), порівняльний аналіз (для оцінки різних методологій управління), моделювання (для створення формалізованих моделей), а також методи теоретичного узагальнення та структурування.

Результати дослідження: формування комплексного підходу до управління веб-проектами, який поєднує концепції сучасних методологій. У роботі уточнено специфічні чинники ризику, характерні для веб-орієнтованих інформаційних систем та запропоновано модель їх класифікації. Крім того, розроблено структуру планування та контролю, що забезпечує адаптивне управління ресурсами й термінами в умовах високої мінливості вимог і технологій.

Результати роботи можуть бути впроваджені в діяльності ІТ-компаній для підвищення точності планування, покращення передбачуваності результатів, оптимізації вартості та мінімізації ризиків у виконанні проектів розробки веб-орієнтованих інформаційних систем.

Ключові слова: УПРАВЛІННЯ ПРОЄКТОМ, ВЕБ-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА, РИЗИКИ, КОМАНДА ПРОЄКТУ, ЯКІСТЬ ПРОЄКТУ.

ABSTRACT

Qualification work on the topic “Project management of web-based information system development” for Master's degree on speciality 122 “Computer Science” educational and professional program “Project management” is written on 74 pages and it contains 8 figures, 10 tables, 4 annexes, and 51 sources.

The purpose of this qualification work is to develop an approach to project management for the development of a web-based information system.

The research methods used in the work include: systems analysis (for studying the subject area and architectural approaches), comparative analysis (for evaluating different management methodologies), modelling (for creating formalized models), as well as methods of theoretical generalization and structuring.

The research results include the formation of a comprehensive approach to managing web projects that integrates concepts from modern methodologies. The study clarifies specific risk factors characteristic of web-based information systems and proposes a model for their classification. In addition, a planning and control framework has been developed that ensures adaptive management of resources and timelines under conditions of high variability of requirements and technologies.

The results of the work can be implemented in IT companies to improve planning accuracy, enhance the predictability of outcomes, optimize costs, and minimize risks in the execution of web-based information system development projects.

Keywords: PROJECT MANAGEMENT, WEB-ORIENTED INFORMATION SYSTEM, RISKS, PROJECT TEAM, PROJECT QUALITY.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної області та постановка задачі дослідження.....	10
1.1 Аналіз предметної області та виклики управління проектами розробки веб-орієнтованих інформаційних систем.....	10
1.2 Аналіз літературних джерел з досліджуваної тематики	12
1.3 Постановка задачі дослідження	16
Висновки до першого розділу.....	18
2 Методи та інструменти управління проектом розробки веб-орієнтованої інформаційної системи	19
2.1 Порівняльний аналіз методологій та обґрунтування вибору підходу до управління проектом розробки інформаційних систем	19
2.2 Дворівнева модель планування робіт з використанням Scrumban для проектів веб-розробки	22
2.3 Оцінка вартості проектів розробки веб-орієнтованих інформаційних систем.	25
2.4 Управління ризиками в проекті розробки веб-орієнтованої інформаційної системи	27
Висновки до другого розділу	34
3 Практична реалізація процесів управління проектом розробки веб-орієнтованої інформаційної системи	35
3.1 Визначення стейкхолдерів та команди проекту.....	35
3.2 Практична реалізація процесів управління проектом у Microsoft Project.....	38
3.3 Бюджетування вартості проекту розробки веб-орієнтованої інформаційної системи	43
3.4 Управління якістю продукту проекту	44
Висновки до третього розділу.....	46
Висновки	48
Список використаних джерел	49
Додаток А Структура розбиття робіт проекту	55

Додаток Б Графік виконання робіт проєкту розробки веб-орієнтованої інформаційної системи	57
Додаток В Довідка про використання результатів	59
Додаток Г Копії сертифікатів участі в конференціях.....	60
Додаток Д Копія опублікованих результатів	61

ВСТУП

Актуальність теми. Сучасний етап цифровізації суспільства та бізнесу супроводжується стрімким зростанням попиту на веб-орієнтовані інформаційні системи. Ці системи стали основними активами для організацій різних сфер діяльності, забезпечуючи ефективну комунікацію, автоматизацію бізнес-процесів та доступ до даних. Однак процес їх розробки характеризується високою складністю, динамічністю вимог та необхідністю інтеграції сучасних технологій, що породжує значні виклики в галузі управління проєктами.

Дослідження зумовлено суттєвою необхідністю вдосконалення інструментів та методів управління проєктами в контексті розробки веб-орієнтованих інформаційних систем. Специфіка таких проєктів полягає у високій динаміці змін вимог, необхідність швидкого виходу на ринок, технологічна складність інтеграцій та робота з розподіленими командами. Існуючі універсальні підходи часто виявляються недостатньо гнучкими, що обумовлює потребу в дослідженні та систематизації спеціалізованих практик управління.

Мета і завдання дослідження. Метою кваліфікаційної роботи є розробка підходу до управління проєктом розробки веб-орієнтованої інформаційної системи.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- здійснити аналіз предметної області, виявити ключові виклики та систематизувати сучасні підходи до управління проєктами розробки веб-орієнтованих інформаційних систем на основі огляду літературних джерел;
- провести порівняльний аналіз існуючих методологій та обґрунтувати вибір гібридного підходу (або конкретної методології) для управління проєктами веб-розробки;
- розробити модель планування робіт із застосуванням підходу Scrumban, адаптовану для специфіки веб-проєктів;
- визначити методику оцінки вартості та сформулювати концепцію управління ризиками для проєкту розробки веб-систем;

– здійснити практичну реалізацію процесів планування та виконання проекту з використанням інструментарію програмного середовища управління проектами, включаючи налаштування ресурсної бази та календарного планування.

– визначити стейкхолдерів і команду проекту, обчислити вартість проекту та провести аналіз якості продукту проекту.

Об’єктом дослідження є процеси управління проектами розробки інформаційних систем.

Предметом дослідження є методи та інструменти процесів управління проектом розробки веб-орієнтованої інформаційної системи.

Методами дослідження, що використовуються в роботі, є: системний аналіз (вивчення предметної області та архітектурних підходів), порівняльний аналіз (оцінки різних методологій управління), моделювання (створення формалізованих моделей), а також методи теоретичного узагальнення та структурування.

Результати дослідження: запропоновано підхід до управління веб-проектами, що інтегрує прогнозне планування, гібридну методологію Scrum та інструменти адаптивного контролю. Запропоновано дворівневу модель планування робіт і спеціалізовані підходи до бюджетування, управління ризиками та забезпечення якості, які враховують особливості веб-орієнтованих інформаційних систем і підвищують точність, керованість та передбачуваність проекту.

Практичне значення отриманих результатів роботи полягає у застосуванні розробленого підходу в діяльності ІТ-компаній для підвищення точності планування, покращення передбачуваності результатів, оптимізації вартості та мінімізації ризиків у виконанні проектів розробки веб-орієнтованих інформаційних систем.

Публікації та апробація кваліфікаційної роботи. Результати дослідження опубліковані та апробовані у матеріалах міжнародної науково-практичної конференції “Інноваційні дослідження в науці та економіці”, 3-5.12.2025 р., м. Брюссель, Бельгія і матеріалах міжнародної науково-практичної конференції “Наукові інновації: теоретичні засади та практичний вплив”, 8-10.12.2025 р., м. Неаполь, Італія (Додатки Г та Д).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області та виклики управління проектами розробки веб-орієнтованих інформаційних систем

Управління проектами розробки інформаційних систем є важливим видом діяльності для успішної реалізації складних завдань у сфері інформаційних технологій. Предметна область даного дослідження зужує фокус до управління проектами, спрямованими на створення веб-орієнтованих інформаційних систем. Під веб-орієнтованими інформаційними системами розуміються програмні комплекси, архітектура яких базується на моделі “клієнт-сервер”, де в якості клієнтської частини виступає веб-браузер, а основним механізмом комунікації та доступу до даних є стандартні інтернет-протоколи [1].

Специфіка розробки веб-орієнтованих інформаційних систем формує виклики для керівника і команди проекту. До ключових належать: висока динаміка змін вимог від бізнесу та кінцевих користувачів, необхідність інтеграції з численними зовнішніми сервісами та API, забезпечення кібербезпеки в умовах постійних загроз, управління розподіленими командами, а також необхідність швидкого виходу на ринок для отримання конкурентних переваг [2]. Перелічені виклики обумовлюють необхідність застосування спеціалізованих підходів до управління. Дослідження показують, що реалізація масштабованих Agile-фреймворків супроводжується значними складнощами, пов'язаними з координацією роботи між командами та узгодженням бізнес-цілей, що особливо актуально для великих веб-проектів [2].

Ключовими характеристиками веб-орієнтованих інформаційних систем як об'єкта управління є:

– ітеративність та еволюційність – веб-системи рідко розробляються як завершений продукт. Вони постійно еволюціонують шляхом додавання нового функціоналу, поліпшення існуючого та адаптації до нових вимог. Як зазначає [3], еволюційна модель життєвого циклу передбачає постійне вдосконалення системи,

що вимагає від методології управління гнучкості для вбудовування змін на будь-якому етапі розробки;

- орієнтація на користувача – успіх веб-орієнтовних інформаційних систем визначається її прийняттям кінцевими користувачами, що робить процеси UX/UI дизайну, юзабіліті-тестування та збирання зворотного зв'язку інтегральними частинами проєкту, які вимагають активної участі команди на всіх етапах [4];

- технологічна складність та мікросервісна архітектура – сучасні веб-орієнтовні інформаційні системи часто будуються на основі мікросервісних архітектур. Як описує [5], мікросервісний підхід передбачає розбиття системи на дрібні, незалежні сервіси, що контрастує з монолітними підходами та вносить корективи в управління командами, процеси розгортання та моніторингу.

- безпека – загрози безпеці є вродженими для веб-середовища, тому управління проєктом повинно включати практики раннього врахування безпеки, коли питання безпеки розглядаються не на етапі тестування, а на етапах проєктування та розробки [6].

Життєвий цикл розробки веб-орієнтованих інформаційних систем може суттєво відрізнятись залежно від обраної методології. Дослідження [7] показує, що для сучасних веб-проєктів характерні гнучкі моделі життєвого циклу, що включають наступні узагальнені фази:

- ініціація та планування – визначення бізнес-цілей та цінності продукту, ідентифікація ключових стейкхолдерів, формування бюджету, попередніх термінів та ризиків проєкту;

- аналіз вимог та проєктування – збір функціональних та нефункціональних вимог, проєктування архітектури системи, створення UX/UI прототипів. Дана фаза часто є ітеративною та повторюється в міру отримання зворотного зв'язку;

- ітеративна розробка – безпосереднє створення системи короткими циклами, що включає паралельну роботу над логікою, інтерфейсом, інтеграцію зовнішніх сервісів та впровадження заходів безпеки;

- тестування та забезпечення якості – комплексне тестування, інтегроване в процес розробки: функціональне тестування, навантажувальне тестування,

перевірка безпеки на наявність поширених вразливостей та тестування зручності використання;

- впровадження та експлуатація – автоматизоване розгортання системи на продуктивних серверах за допомогою конвеєрів CI/CD, навчання користувачів, моніторинг працездатності та продуктивності системи в реальному часі;

- супровід та постійне вдосконалення активна фаза розвитку, що включає усунення помилок, аналіз поведінки користувачів, A/B тестування нового функціоналу, пріоритизацію змін на основі даних та технічну підтримку.

Такий ітеративний підхід забезпечує необхідну гнучкість та адаптивність до змін протягом усього життєвого циклу проєкту [7].

Основним напрямом дослідження в роботі є обґрунтування підходів, методів та інструментів управління, здатних комплексно враховувати специфіку веб-орієнтованих інформаційних систем.

1.2 Аналіз літературних джерел з досліджуваної тематики

Сучасна наукова думка щодо управління проєктами розробки веб-орієнтованих інформаційних систем відображає перехід від класичних методологій до динамічних, гібридних підходів. Аналіз наукових праць, опублікованих за останні роки у провідних фахових виданнях, дозволяє ідентифікувати низку стійких тенденцій. До них належать: глибока інтеграція гнучких принципів з елементами прогностного управління, поширення масштабованих фреймворків, синтез DevOps-культури з практиками безпеки, а також зростання ролі дата-орієнтованого управління якістю на основі моніторингу продуктивності системи в реальному часі. Даний аналіз становить теоретичну основу для подальшого формування завдань дослідження.

Гнучкі методології продовжують залишатися основним інструментом для проєктів, що характеризуються високою динамікою змін. Сучасні дослідження зосереджуються не лише на підтвердженні ефективності Agile, але й на аналізі їх еволюції та адаптації до нових умов. Дослідження [8], проведене на прикладі

масштабної трансформації в компанії Ericsson, аналізує фактори успіху гнучкої розробки. Автори приходять до висновку, що для веб-орієнтованих інформаційних систем критично важливими є поетапне впровадження змін, підготовка лідерів та адаптація процесів під специфічні потреби організації.

Продовжуючи цю думку, дослідження [9] вказує на тенденцію до контекстуалізації гнучкого підходу, коли команди веб-розробки не дотримуються догматично канонічних правил, а адаптують та гібридизують практики під свої потреби. Сучасне управління проєктами веб-розробки найчастіше являє собою синтез різних підходів, що зображено на рисунку 1.1.

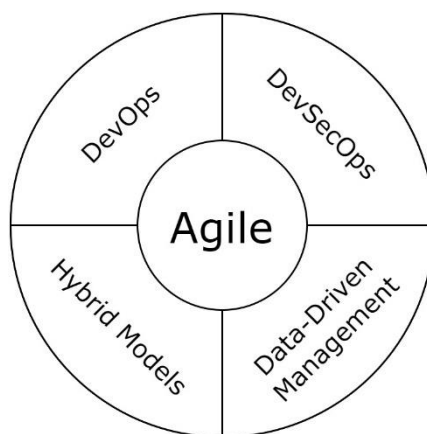


Рисунок 1.1 – Ключові компоненти сучасного управління проєктами веб-розробки

Складність сучасних корпоративних веб-систем вимагає поєднання гнучкості з елементами структурованого планування. У систематичному огляді [10] детально аналізується досвід використання масштабованих фреймворків, таких як Scaled Agile Framework (SAFe) та Large-Scale Scrum (LeSS). Автори стверджують, що ці фреймворки найефективніші для розробки складних веб-платформ, оскільки забезпечують необхідну узгодженість та синхронізацію випуску.

Одночасно отримали розвиток більш адаптивні гібридні моделі. Дослідження [11] присвячене порівняльному аналізу різних конфігурацій гібридних моделей розробки програмного забезпечення. Автори доводять, що для веб-проєктів у високо регламентованих галузях оптимальною є модель, де фаза архітектурного та

безпекового проектування проводиться за прогнозним підходом, що забезпечує стабільність фундаменту, тоді як безпосередня розробка інтерфейсів та бізнес-логіки ведеться гнучкими ітераціями.

Еволюція веб-розробки пов'язана з культурою DevOps, яка об'єднала розробку та експлуатацію в єдиний потік. Емпіричне дослідження [12] доводить прямий вплив практик DevOps на швидкість доставлення нових функцій веб-додатків. Автори виявляють, що автоматизація конвеєрів CI/CD дозволяє командам виконувати сотні розгортань на день, що є ключовим для підтримки конкурентоспроможності.

Безпека, як важливий аспект веб-середовища, закономірно інтегрувалася в практики DevOps, сформувавши напрям DevSecOps. У роботі [13] детально описується, як інтегрувати інструменти статичного та динамічного аналізу безпеки (SAST/DAST) у конвеєр CI/CD, що дозволяє виявляти вразливості ще на етапі створення коду.

Сучасні підходи до управління проектами все більше орієнтуються на прийняття рішень на основі даних. Дослідження [14] пропонує модель безперервного моніторингу якості веб-додатків у реальному часі, рекомендуючи відстежувати метрики продуктивності, що дозволяє кількісно оцінювати вплив нових функцій на користувацький досвід.

Систематичний огляд [15] вивчає виклики масштабних Agile-трансформацій. Автори виявляють, що найефективнішими є підходи, що інтегрують управління ризиками в рутину команди: регулярні обговорення загроз на ретроспективах, ведення “беклогу ризиків” та використання предикативної аналітики.

Архітектурні рішення безпосередньо впливають на організаційну структуру. Дослідження [16] аналізує зв'язок між мікросервісною архітектурою та організаційними структурами, стверджуючи, що оптимальною є модель, де кожна автономна команда повністю відповідає за життєвий цикл окремих мікросервісів. Це змінює роль менеджера проекту у бік координації взаємодії між такими командами.

Окремої уваги заслуговує вплив хмарних технологій. У роботі [17] розглядаються стратегії управління проектами розробки, орієнтовані на хмарні сервіси, зокрема безсерверні архітектури, що впливають на планування бюджету, управління інфраструктурою як кодом та безпеку.

Також, успіх веб-орієнтованих інформаційних систем безпосередньо залежить від їхнього сприйняття кінцевими користувачами, що обумовило виокремлення управління якістю та користувацьким досвідом в критично важливий напрям дослідження. Сучасні підходи до якості веб-продукту орієнтуються не лише на відсутність дефектів, але й на сукупність нефункціональних характеристик, таких як продуктивність, зручність використання, доступність та емоційний відгук [18].

Дослідження останніх років демонструють перехід від епізодичного контролю якості до її безперервного моніторингу. Так, дослідження [19] пропонують фреймворк для постійного відстеження якості веб-додатків, який наголошує на ключових метриках продуктивності, таких як час завантаження основних елементів сторінки, затримка при першій взаємодії та стабільність візуального макету. Ці показники дозволяють командам кількісно оцінювати вплив кожної нової функції на користувацький досвід і оперативно виправляти негативні тенденції.

Інтеграція UX-процесів в Agile-цикли розробки вимагає спеціальних організаційних рішень. Робота [20] аналізує моделі взаємодії UX-фахівців з командами розробників. Автори виділяють ефективні практики, серед яких регулярне тестування зручності використання з прототипами в межах спринту, безпосередня участь дизайнерів у плануванні завдань та створення стандартизованих бібліотек інтерфейсних компонентів для забезпечення візуальної узгодженості. Це вимагає від менеджера проекту активної координації міждисциплінарної взаємодії, що стає ключовим фактором успіху.

На основі проведеного аналізу можна зробити висновок, що управління проектами розробки веб-орієнтованих інформаційних систем перетворилося на високоадаптивну, багатоаспектну дисципліну. Сучасний підхід являє собою

динамічну комбінацію методологій, практик та інструментів, що включає гнучкі та гібридні моделі, масштабовані фреймворки, практики DevOps/DevSecOps та дата-орієнтоване управління якістю. Роль менеджера проєкту еволюціонувала від адміністратора плану до архітектора процесів, здатного будувати методологічний ландшафт, враховуючи унікальні характеристики проєкту.

Аналіз літератури виявив, що більшість досліджень зосереджено на окремих аспектах або на загальних принципах гібридизації. Відсутніми є комплексні методики або рамки, які б структуровано інтегрували ці різнобічні аспекти спеціально для контексту розробки веб-орієнтованих інформаційних систем, враховуючи їхню еволюційність, технологічну складність та орієнтацію на користувача. Ця прогалина обґрунтовує необхідність формування цілісного підходу до управління та визначає актуальність даного дослідження.

1.3 Постановка задачі дослідження

Проведений аналіз предметної області та наукової літератури виявив потребу у комплексному підході до управління проєктами розробки веб-орієнтованих інформаційних систем, який би системно інтегрував вибір методологій, сучасні архітектури, практики безпеки та управління якістю в умовах високої динаміки [21, 22].

У цьому контексті особливої уваги потребує обґрунтований вибір методології управління проєктом, оскільки саме вона визначає рівень адаптивності команди до змін, тривалість циклу постачання функціоналу та ефективність використання ресурсів [23]. Проєкти веб-орієнтованих інформаційних систем характеризуються швидкими змінами вимог, інтерактивним характером взаємодії з користувачами, залежністю від ринкових трендів і необхідністю оперативного вдосконалення функціональності. За таких умов традиційні лінійні моделі, зокрема Waterfall, демонструють обмежену придатність, оскільки їх жорстка послідовність етапів не дозволяє швидко реагувати на зміни середовища [24].

Гнучкі методології Agile, навпаки, спрямовані на ітераційний розвиток продукту, постійний перегляд пріоритетів та активну участь замовника у формуванні вимог. Найбільш поширеним фреймворком у цій групі є Scrum, який забезпечує чітку структуру ітерацій, прозорість процесу та регулярне отримання зворотного зв'язку, що сприяє підвищенню якості та своєчасності розробки [25, 26]. У проєктах, де спостерігається нерівномірний або неперервний потік задач, особливо ефективним є підхід Kanban, оснований на візуалізації робочого процесу та регулюванні навантаження команди шляхом обмеження обсягу робіт у виконанні [25].

Водночас особливості сучасної веб-розробки вимагають не лише гнучкості, але й оптимізації потоків створення цінності, що закладено у філософії Lean Software Development. Орієнтація на усунення втрат, прискорення потоку поставки функціоналу, автономність команд та безперервне вдосконалення забезпечують високу ефективність розробки у середовищах з високою частотою змін [27].

Порівняльний аналіз цих підходів підтверджує, що жодна з методологій не є універсальною. Саме тому в індустрії набувають поширення гібридні моделі, які дозволяють поєднати переваги різних практик. Одним із найбільш ефективних рішень для веб-проєктів є модель Scrumban, що інтегрує ітеративність і структурованість Scrum із гнучкістю та оптимізаційними механізмами Kanban [28]. На стратегічному рівні Scrumban забезпечує планування та контроль розвитку системи, тоді як на операційному – дозволяє швидко реагувати на зміни пріоритетів чи появу невідкладних завдань.

Застосування Scrumban сприяє скороченню часу виконання робіт, зниженню кількості дефектів та підвищенню задоволеності стейкхолдерів завдяки більш рівномірному розподілу навантаження та постійному вдосконаленню процесу. Для проєктів розробки веб-орієнтованих інформаційних систем, що поєднують планову розробку із реактивними змінами, такий підхід є оптимальним, оскільки забезпечує баланс структурованості та адаптивності, необхідний у сучасному динамічному середовищі.

Метою кваліфікаційної роботи є розробка підходу до управління проектом розробки веб-орієнтованої інформаційної системи.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- здійснити аналіз предметної області, виявити ключові виклики та систематизувати сучасні підходи до управління проектами розробки веб-орієнтованих інформаційних систем на основі огляду літературних джерел;
- провести порівняльний аналіз існуючих методологій та обґрунтувати вибір гібридного підходу (або конкретної методології) для управління проектами веб-розробки;
- розробити модель планування робіт із застосуванням підходу Scrumban, адаптовану для специфіки веб-проектів;
- визначити методику оцінки вартості та сформувати концепцію управління ризиками для проекту розробки веб-систем;
- здійснити практичну реалізацію процесів планування та виконання проекту з використанням інструментарію програмного середовища управління проектами, включаючи налаштування ресурсної бази та календарного планування.
- визначити стейкхолдерів і команду проекту, обчислити вартість проекту та провести аналіз якості продукту проекту.

Висновки до першого розділу

1. Виявлено характерні риси веб-орієнтованих інформаційних систем як об'єкта управління та визначено ключові виклики, зумовлені їх специфікою, що обґрунтовує необхідність адаптації традиційних підходів до управління проектами.

2. Проведений аналіз літератури систематизує сучасні тенденції, серед яких гібридизація методологій, інтеграція DevOps/DevSecOps та data-орієнтоване управління, водночас виявляючи відсутність єдиного комплексного підходу, спеціалізованого для управління проектами веб-орієнтованих інформаційних систем, що зумовлює необхідність його подальшого розроблення.

3. Здійснено постановку задачі дослідження.

2 МЕТОДИ ТА ІНСТРУМЕНТИ УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Порівняльний аналіз методологій та обґрунтування вибору підходу до управління проєктом розробки інформаційних систем

Управління проєктами розробки веб-орієнтованих інформаційних систем, є важливим процесом, що визначає успішність їх реалізації. Специфіка таких проєктів, як висока динаміка змін вимог, необхідність швидкого виходу на ринок та ітеративний характер розробки, обумовлює вимоги до застосовуваних методологій управління. Вибір методології безпосередньо впливає на гнучкість, прогнозованість та ефективність використання ресурсів протягом усього життєвого циклу проєкту.

Сучасна практика управління проєктами розробки веб-орієнтованих інформаційних систем пропонує спектр підходів, серед яких ключовими є каскадна модель (Waterfall), гнучкі методології (Agile), а також філософія Lean Software Development [29].

Кожен із цих підходів має чітко визначену сферу ефективного застосування, обумовлену характеристиками та контекстом конкретного проєкту.

Модель Waterfall є класичним лінійно-послідовним підходом, у якому етапи розробки виконуються строго послідовно. Основними перевагами є чіткість структури та планування, що робить її придатною для проєктів зі стабільними, заздалегідь повністю визначеними вимогами. Однак низька адаптивність до змін обмежує її ефективність для більшості сучасних проєктів розробки веб-орієнтованих інформаційних систем, які потребують ітеративного вдосконалення [29].

Гнучка методологія Agile базується на ітеративному та інкрементальному підході, орієнтованому на адаптацію до змін, безперервне вдосконалення продукту та тісну взаємодію із замовником [30].

Це робить Agile найбільш поширеним та прийнятним підходом для веб-розробки, де вимоги часто змінюються під впливом ринкових умов та зворотного зв'язку від користувачів.

Однією з найпопулярніших реалізацій Agile є фреймворк Scrum, який передбачає розподіл роботи над проектом на короткі ітерації, що тривають 2-4 тижні. У межах кожного спринту команда розробляє попередньо визначений набір функціональності з беклогу продукту. Завершення кожного спринту супроводжується демонстрацією робочого інкременту продукту замовнику та збором зворотного зв'язку, що дозволяє оперативно коригувати напрям розробки. Такий підхід забезпечує високу прозорість процесу, підвищує продуктивність команди, мінімізує ризики та скорочує час виходу продукту на ринок порівняно з традиційними методами [30].

Іншою поширеною практикою в рамках Agile є методологія Kanban, яка орієнтована на візуалізацію робочого процесу та обмеження обсягу робіт у потоці. Основним інструментом управління виступає Kanban-дошка, що відображає статус завдань у реальному часі. Перевагою Kanban є відсутність жорстких ітерацій, що робить його ефективним для проектів із постійним потоком вхідних завдань, зокрема при підтримці та поступовому вдосконаленні вже існуючих веб-орієнтованих інформаційних систем [31].

Філософія Lean Software Development концентрується на максимізації цінності для кінцевого користувача шляхом усунення всіх видів втрат, таких як надлишкова функціональність, затримки та неефективні процеси. Її ключові принципи включають усунення непотрібних дій, швидке доставлення мінімально життєздатного продукту, надання команді автономії та постійне вдосконалення [32]. Для веб-розробки цей підхід є особливо актуальним, оскільки дозволяє мінімізувати витрати на розробку функціоналу та забезпечити швидкий вихід на ринок.

Порівняльна характеристика розглянутих методологій управління ІТ-проектами, в тому числі розробки інформаційних систем наведена в таблиці 2.1.

Таблиця 2.1 – Порівняльна характеристика методологій управління ІТ-проектами

Критерій	Методології			
	Waterfall	Scrum	Kanban	Lean
Гнучкість	Низька	Висока	Дуже висока	Висока
Структурованість	Дуже висока	Висока	Низька	Середня
Придатність для проектів із невизначеними вимогами	Низька	Висока	Дуже висока	Висока
Швидкість виходу на ринок	Низька	Висока	Висока	Дуже висока
Ефективність використання ресурсів	Середня	Середня	Дуже висока	Дуже висока
Ідеальна сфера застосування	Проекти з жорсткими стандартами	Продуктова розробка, стартапи, проекти з динамічними вимогами	Підтримка, DevOps, оперативне вирішення інцидентів	Стартапи, R&D, оптимізація бізнес-процесів
Рівень документації	Дуже високий	Середній	Низький	Низький

На основі проведеного аналізу для управління проектом розробки даної веб-орієнтованої інформаційної системи було обґрунтовано вибір гібридного підходу, що інтегрує елементи Scrum та Kanban (так званий Scrumban). Таке рішення зумовлене специфікою проекту:

- наявність досить чіткого початкового технічного завдання дозволило сформулювати беклог продукту та працювати короткими ітераціями, що відповідає принципам Scrum. Це забезпечило структурованість процесу розробки та передбачуваність результатів на стратегічному рівні;

- протягом життєвого циклу розробки очікувалися оперативні зміни вимог та нові задачі, для ефективного управління якими запроваджувалися практики Kanban. Це дозволило забезпечити гнучкість на оперативному рівні та швидко реагувати на зміни пріоритетів;

- комбінація двох підходів дозволила уникнути недоліків, властивих кожному з них окремо. Зокрема, було подолано надмірну бюрократизацію Scrum та недостатню структурованість Kanban.

Емпіричні дані свідчать, що гібридний підхід Scrumban дозволяє досягти значно кращих результатів у веб-розробці: зменшення часу виконання завдань на 30-35%, підвищення задоволеності замовника на 40-45%, зниження кількості критичних помилок на 25-30% порівняно з використанням чистих методологій.

Отже, поєднання структурованості та передбачуваності Scrum з оперативною гнучкістю Kanban є оптимальним для досягнення цілей проєкту сучасної веб-розробки, що характеризується високою динамікою змін та жорсткими вимогами до якості та швидкості розробки.

2.2 Дворівнева модель планування робіт з використанням Scrumban для проєктів веб-розробки

Управління проєктами розробки сучасних веб-орієнтованих інформаційних систем супроводжується фундаментальною протилежністю між вимогою до передбачуваності та необхідністю гнучкості [33]. Як розв'язання цієї проблеми пропонується дворівнева модель планування робіт і процесу реалізації проєкту, де макрорівень формує стратегічний план, а мікрорівень – тактичний, щоденний. Ця дворівнева модель системно інтегрує прогностичні методології стратегічного рівня з адаптивними практиками оперативного виконання. Дана модель слугує формалізацією застосування гібридного підходу Scrumban, забезпечуючи детермінований перехід від високорівневих цілей до щоденних задач розробки.

Макрорівень відповідає за формування архітектурного та фінансового каркасу проєкту. На цьому рівні життєвий цикл концептуалізується як послідовність фаз з використанням класичних прогностичних інструментів управління проєктами:

– метод структурної декомпозиції робіт застосовується для ієрархічного поділу всієї роботи на менші, керовані компоненти. Це забезпечує повноту охоплення робіт, точність оцінок та чіткий розподіл відповідальності, що є особливо важливим на початкових етапах веб-розробки [34];

– методи аналогії та експертні оцінки використовуються для первинного визначення тривалості та ресурсомісткості робочих пакетів, формуючи базис для бюджетування [35].

Мікрорівень визначає процес безпосередньої реалізації робочих пакетів в умовах технічної та бізнес-невизначеності. На цьому рівні реалізується синтез фреймворків Scrum та Kanban, що забезпечує ритм, фокус на створенні цінності та циклічне вдосконалення процесів:

– scrum-компонент вносить ритм та структуру через фіксовані за тривалістю ітерації. Кожна ітерація має чітку мету, що формується на плануванні спринту на основі пріоритетного беклогу продукту. Регулярні щоденні стендапи, демонстрації результатів та ретроспективи забезпечують координацію, безперервний зворотний зв'язок від замовника та ітеративне вдосконалення робочих процесів команди [36];

– kanban-компонент оптимізує потік робіт всередині спринту за допомогою принципів візуалізації та обмеження. Kanban-дошка зі стовпцями станів задач надає повну прозорість процесу. Ключовим механізмом є обмеження роботи в процесі, що запобігає перевантаженню команди, ідентифікує вузькі місця в потоці та суттєво скорочує середній час виконання завдань [37].

Механізм інтеграції рівнів виступає критичним елементом запропонованої моделі, що забезпечує когерентний перехід від стратегічного плану до оперативних дій. Він реалізується через послідовну декомпозицію та поступове уточнення та деталізацію завдань, формуючи єдиний потік створення цінності:

– від макрорівня до беклогу продукту – елементи WBS деталізуються на конкретні, оцінені історії користувача, які накопичуються в беклозі продукту. Кожна історія описує функціональність з точки зору кінцевого користувача та її бізнес-цінність, що відповідає принципам гнучкої розробки [36];

– від беклогу продукту до спринту – на етапі планування спринту команда спільно з власником продукту відбирає з пріоритетного беклогу продукту історії для наступної ітерації, формуючи беклог спринту. На цьому етапі історії деталізуються, розбиваються на технічні завдання та оцінюються за складністю;

– від спринту до потоку робіт – під час виконання спринту деталізовані технічні завдання розміщуються на Kanban-дошці, де їх статус оновлюється в реальному часі. Команда фокусується на завершенні завдань, послідовно переміщуючи їх по дошці з дотриманням встановлених обмежень, що максимізує ефективність потоку [37].

Дворівнева модель планування робіт з використанням Scrumban у вигляді схеми наведено на рисунку 2.1.

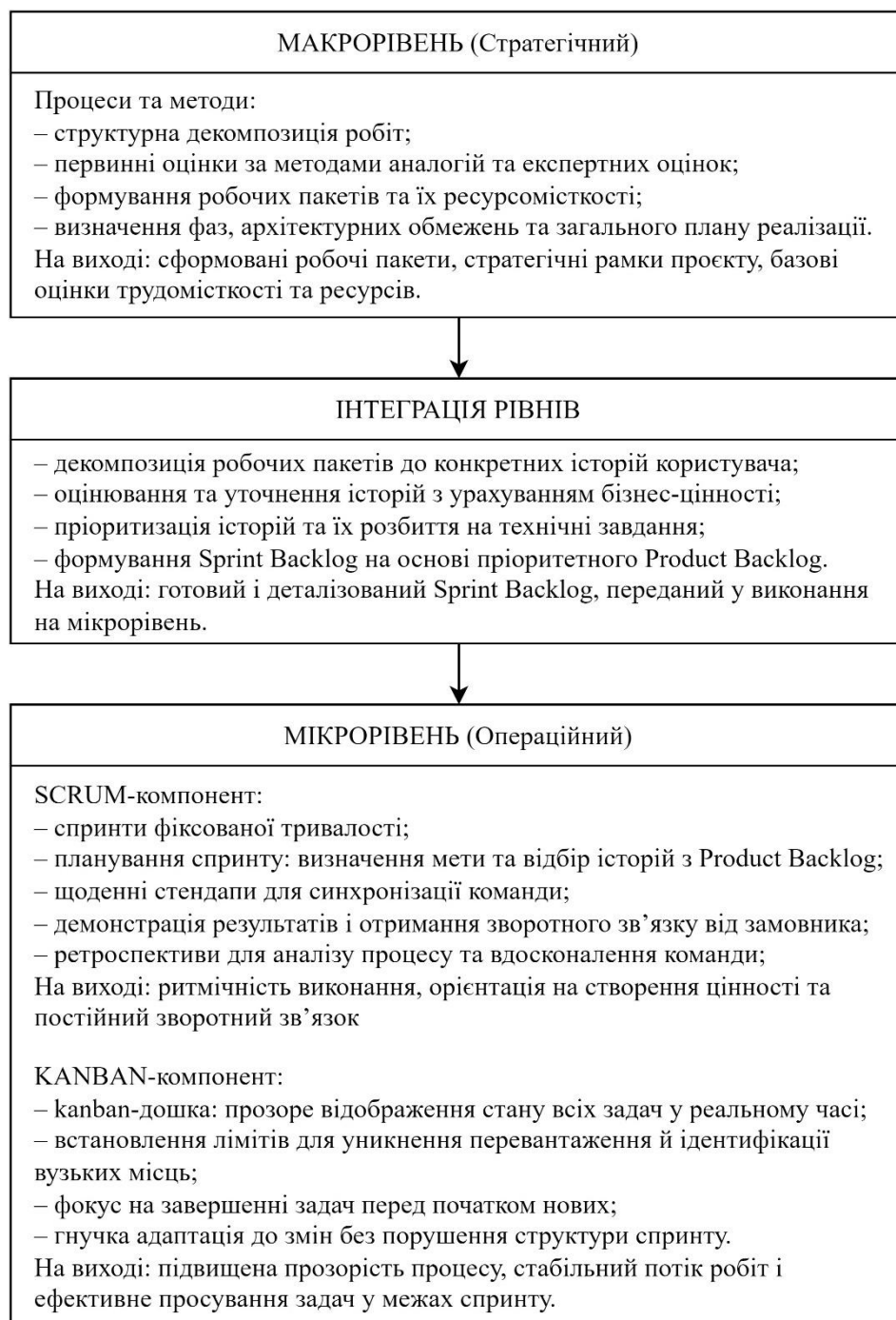


Рисунок 2.1 – Дворівнева модель планування робіт з використанням Scrumban

Запропонована дворівнева модель формалізує циклічний процес, у якому макрорівень задає напрямок та системні обмеження, а мікрорівень забезпечує максимально ефективно просування у заданому напрямку з можливістю оперативної корекції траєкторії на основі зворотного зв'язку та виявлених вузьких місць.

2.3 Оцінка вартості проєктів розробки веб-орієнтованих інформаційних систем

Для веб-орієнтованих інформаційних систем, які характеризуються динамічним середовищем та ітеративною природою розробки, традиційні параметричні моделі, такі як COCOMO (Constructive Cost Model) або методика функціональних точок (Function Point Analysis), часто демонструють обмежену точність на ранніх етапах [38]. Це пов'язано з високим рівнем невизначеності щодо детальних вимог, технологічного стеку та архітектурних рішень, що є типовими для початку веб-проєктів.

Для проєктів з достатньо деталізованою структурою розбиття робіт найвищу точність забезпечує підхід до оцінювання “знизу-вверх” [39]. При цьому рівень деталізації, на практиці, зазвичай визначається досвідом керівника проєкту. Суть цього методу полягає в послідовній оцінці вартості кожного компонента проєкту, починаючи з найменшого, з подальшим підсумовуванням цих оцінок для отримання загальної вартості. Його ключові переваги для веб-розробки включають:

- висока точність – оцінка базується на детальному розумінні конкретних задач, а не на узагальнених параметрах всього проєкту;
- врахування специфіки технологій – дозволяє застосовувати різні нормативи вартості для різних видів робіт;
- прозорість та обґрунтованість – кожна частина бюджету може бути чітко протрасована до конкретного робочого пакету та відповідального виконавця, що підвищує довіру замовника;

– сумісність з гнучкими підходами – метод добре інтегрується з практиками Scrum та Kanban, оскільки дозволяє оцінювати окремі користувачькі історії або задачі, які агрегуються на рівні спринту чи проєкту.

Запропонований підхід “знизу-вверх” можна представити у вигляді наступного алгоритму (рисунок 2.2).



Рисунок 2.2 – Схема алгоритму оцінки вартості проєкту розробки інформаційної системи

Підхід “знизу-вверх” включає такі етапи:

1) здійснюється декомпозиція на основі вимог та загального плану будови системи формується структура розподілу робіт;

2) виконується оцінка трудомісткості, шляхом оцінювання кожного окремого завдання у людино-днях або годинах. Для цього використовуються експертні оцінки, дані з аналогічних проєктів або методи групового оцінювання;

3) визначаються вартості одиниці ресурсу шляхом встановлення середньої вартості робочого дня для кожної типу фахівця, необхідної для виконання робіт з урахуванням ринкових ставок та рівня кваліфікації;

4) розраховується пряма вартість праці. Вартість кожного робочого завдання розраховується як добуток його трудомісткості на вартість робочого дня фахівця;

5) пряма вартість праці по всіх робочих пакетах підсумовується;

6) до отриманої суми додаються оцінені матеріальні (апаратне та програмне забезпечення, ліцензії, хостинг та інші) та адміністративні (навчання, оренду та інші) витрати;

7) визначається резерв на непередбачені витрати для керування ризиками. Розмір резерву обґрунтовується результатами аналізу ризиків проєкту.

8) формування фінальної вартості проєкту, шляхом сумування всіх розрахованих витрат.

Запропонований підхід безпосередньо пов’язує фінансове планування з технічним змістом проєкту та календарним планом, що є основою для формування бюджету, подальшого контролю витрат та ефективного управління ризиками.

2.4 Управління ризиками в проєкті розробки веб-орієнтованої інформаційної системи

Управління ризиками є систематичним процесом, спрямованим на ідентифікацію, аналіз, планування реагування та моніторинг невизначеностей, які можуть вплинути на досягнення цілей проєкту. У контексті розробки веб-орієнтованих інформаційних систем цей процес має велике значення для

забезпечення дотримання обмежень за термінами, бюджетом, якістю продукту та змістом робіт. Динамічність галузі, швидка зміна технологій та високі вимоги безпеки обумовлюють необхідність структурованого підходу до мінімізації потенційних загроз [40].

Ризики в ІТ-проектах прийнято класифікувати за джерелом виникнення та сферою впливу. Для веб-розробки виділяють такі основні категорії:

- технічні ризики – пов’язані з інфраструктурою, технологічним стеком, архітектурою системи, її продуктивністю, безпекою та надійністю.

- фінансові ризики – обумовлені порушенням бюджетних обмежень, нестабільністю фінансування або непередбачуваними витратами.

- операційні ризики – виникають у процесі планування та виконання проекту, включаючи помилки в оцінці робіт, управлінні вимогами та взаємодії із замовником.

- політичні та юридичні ризики – зумовлені невідповідністю законодавчим вимогам, ліцензійними обмеженнями або можливими юридичними суперечками.

- соціальні ризики – пов’язані з людським фактором: недостатньою кваліфікацією команди, плинністю кадрів та неефективною внутрішньою комунікацією [41].

Для проєктів веб-розробки, які характеризуються високою динамікою та обмеженими ресурсами для глибокого статистичного моделювання, оптимальною є підхід на основі матриці ймовірність-вплив [40]. Вона дозволяє швидко пріоритезувати ризики без трудомістких кількісних розрахунків, що часто неможливо через відсутність історичних даних або через швидкі зміни в умовах проєкту.

Підхід базується на двох ключових параметрах:

- ймовірність – оцінка шансів настання ризикової події. Використовується порядкова шкала, наприклад, від 1 (дуже мало ймовірно) до 5 (дуже ймовірно);

- вплив – оцінка наслідків реалізації ризику на цілі проєкту. Шкала також може бути порядковою, наприклад, від 1 (мінімальний вплив) до 5 (критичний вплив).

На основі параметрів визначається ранг ризику, який розраховується як добуток чисельних оцінок ймовірності та впливу. Ранг дозволяє відсортувати ризики за рівнем небезпеки. Для інтерпретації результатів використовується класифікація:

- прийнятний (1-5) – мінімальний вплив;
- низький (6-10) – можна прийняти;
- середній (11-15) – потребує заходів пом'якшення;
- високий (16-20) – необхідні активні дії;
- критичний (21-25) – вимагає негайного реагування.

В рамках проєкту розробки веб-орієнтованої інформаційної системи було ідентифіковано технічні, фінансові, операційні, політичні і юридичні та соціальні ризики.

Технічні ризики:

- ризик вибору технологій – використання застарілих або непридатних інструментів, що ускладнить підтримку, масштабування та інтеграцію системи;
- ризик інтеграції – виникнення проблем під час підключення до зовнішніх сервісів;
- ризик безпеки – недостатній рівнем захисту системи від типових атак;
- ризик хостингу – ненадійність інфраструктури провайдера або неспроможність серверів обробляти пікові навантаження;
- ризик втрати даних – втрата інформації через збої обладнання, помилки програмного забезпечення або зловмисні дії;
- ризик збереження даних – невідповідне або небезпечне зберігання персональних та конфіденційних даних.

Фінансові ризики:

- ризик перевищення бюджету – непередбачені витрати, затримки чи доопрацювання, що виводять витрати за межі затвердженого бюджету;
- ризик фінансової нестабільності – затримка або припинення фінансування з боку замовника, що загрожує зупинкою проєкту.

Операційні ризики:

- ризик недооцінки обсягу робіт – неправильна оцінка складності та тривалості завдань;
- ризик зміни вимог – постійне коригування вимог проєкту;
- ризик планування – неправильне планування виконання проєкту;
- ризик формулювання вимог – нечіткий або неповний опис очікувань замовника щодо функцій системи.
- ризик залучення замовника – відсутність своєчасного зворотного зв'язку або затримками у прийнятті рішень з боку замовника;
- ризик проміжного тестування – виявлення помилок у системі через пасивну участь замовника у валідації проміжних результатів.

Політичні та юридичні ризики:

- ризик ліцензійних умов – використання програмних компонентів з порушенням ліцензійних угод;
- ризик вимог законодавства – невідповідність системи вимогам законодавства щодо захисту даних, що може призвести до юридичних наслідків, штрафів або санкцій;
- ризик юридичних спорів – розбіжності з замовником щодо умов договору або якості результату.

Соціальні ризики:

- ризик кваліфікації фахівців – недостатній рівень знань та досвіду членів команди для вирішення проєктних завдань;
- ризик плинності кадрів – втрата ключових розробників або спеціалістів під час реалізації проєкту;
- ризик комунікації – неефективна взаємодія всередині команди, що призводить до неузгодженості дій.

Для проведення якісного аналізу було прийнято шкалу оцінки ймовірності реалізації ризику, що наведено в таблиці 2.2, де 1 – дуже мало ймовірно, а 5 – дуже ймовірно.

Таблиця 2.2 – Шкала оцінки ймовірності реалізації ризику

Ймовірність	Числова оцінка	Опис
Дуже мало ймовірно	1	Ризик майже неможливий для реалізації
Мало ймовірно	2	Ризик має низькі шанси на реалізацію
Можливо	3	Ризик може реалізуватися з певною ймовірністю
Ймовірно	4	Ризик, швидше за все, реалізується
Дуже ймовірно	5	Ризик має високі шанси на реалізацію

Також було прийнято шкалу оцінки впливу ризику на проєкт, що наведено в таблиці 2.3, де 1 – можна знехтувати, а 5 – суттєвий.

Таблиця 2.3 – Шкала оцінки впливу ризику на проєкт

Вплив	Числова оцінка	Опис
Можна знехтувати	1	Мінімальний вплив на цілі проєкту
Невеликий	2	Незначне відхилення від плану
Помірний	3	Помітне відхилення, що вимагає корекції плану
Значний	4	Серйозне порушення цілей проєкту
Суттєвий	5	Критичне відхилення, що загрожує успіху проєкту

Експертним шляхом було проведено оцінювання кожного ідентифікованого ризику. Результати оцінки наведені в таблиці 2.4, де також розраховано ранг кожного ризику.

Таблиця 2.4 – Матриця оцінки та пріоритезації ризиків

Код	Ідентифіковані ризики	Ймовірність	Вплив	Ранг ризику
1	2	3	4	5
101	Ризик вибору технологій	3	4	12
102	Ризик інтеграції	4	3	12
103	Ризик безпеки	4	5	20
104	Ризик хостингу	2	4	8
105	Ризик втрати даних	3	5	15
106	Ризик збереження даних	3	5	15
201	Ризик перевищення бюджету	3	4	12
202	Ризик фінансової нестабільності	3	5	15
301	Ризик недооцінки обсягу робіт	4	4	16
302	Ризик зміни вимог	5	4	20
303	Ризик планування	3	5	15
304	Ризик формулювання вимог	3	4	12
305	Ризик залучення замовника	4	3	12
306	Ризик проміжного тестування	3	4	12
401	Ризик ліцензійних умов	2	3	6

Продовження таблиці 2.4

1	2	3	4	5
402	Ризик вимог законодавства	2	5	10
403	Ризик юридичних спорів	2	4	8
501	Ризик кваліфікації фахівців	3	4	12
502	Ризик плинності кадрів	3	3	9
503	Ризик комунікації	4	3	12

Результати візуалізовано за допомогою карти ризиків, де осі відповідають ймовірності та впливу, а ризики розміщуються у відповідних квадрантах. На рисунку 2.3 зображено карту ризиків, де візуально виділено зони критичних, високих та середніх ризиків.

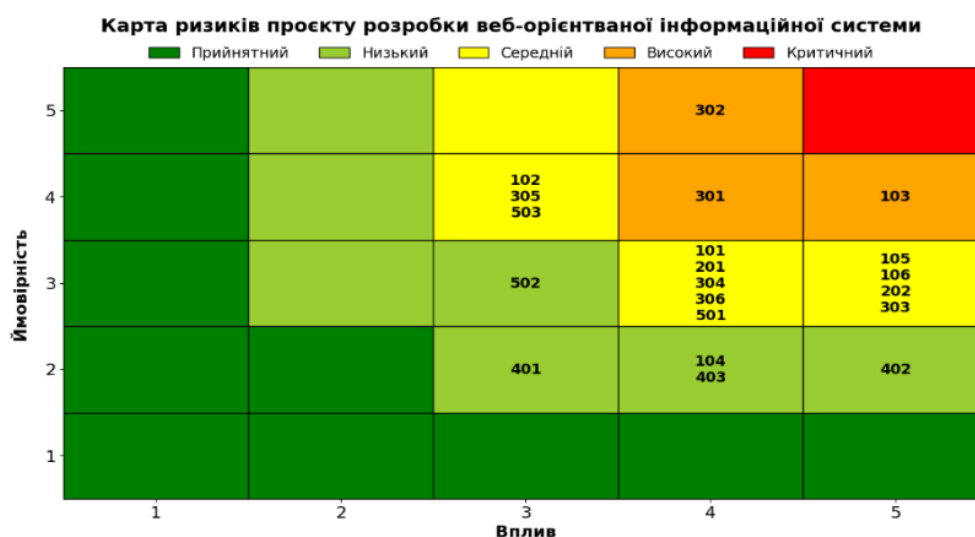


Рисунок 2.3 – Карта ризиків проєкту розробки веб-орієнтованої інформаційної системи

З карти ризиків проєкту видно, що до найбільш критичних та високих ризиків належать ризики безпеки (103) та зміни вимог (302), які мають високу ймовірність та суттєвий вплив. Ризик недооцінка обсягу робіт (301) класифікований як високий, що вказує на його значну загрозу для дотримання строків. Такі ризики, як втрати даних (105), збереження даних (106), фінансова нестабільність (202) та планування (303) вимагають розробки запобіжних заходів через їхній помірну ймовірність, але значний потенційний збиток для проєкту.

Наступним етапом процесу управління ризиками є розробка стратегій та конкретних дій для критичних та високоризикових позицій. Використовуються такі основні стратегії реагування:

- уникнення – зміна плану проєкту для повного усунення загрози;
- передача – перекладення відповідальності за ризик на третю сторону;
- пом'якшення – вжиття заходів для зниження ймовірності або наслідків ризику;
- прийняття – свідоме прийняття ризику без активних дій, якщо витрати на реагування перевищують потенційний збиток. Для таких ризиків може створюватися резервний план.

В таблиці 2.5 наведено план реагування для загрозливих ризиків проєкту, який містить стратегії реагування та заходи для ризиків.

Таблиця 2.5 – План реагування на ключові ризики проєкту

Код	Ризик	Стратегія реагування	Дії
103	Ризик безпеки	Запобігання / Пом'якшення	Проведення аудиту безпеки на етапі проєктування. Впровадження Secure SDLC, регулярне статичне тестування та пентест перед здачею.
302	Ризик зміни вимог	Запобігання / Пом'якшення	Детальне ТЗ, використання гнучкого підходу з фіксованими спринтами, формалізація процесу змін та регулярні демо для замовника.
301	Ризик недооцінки обсягу робіт	Запобігання / Пом'якшення	Детальне розбиття робіт із залученням команди, використання кількох методів оцінки, створення резерву часу у розкладі (10-15%).
105	Ризик втрати даних	Запобігання / Передача	Автоматичне щоденне резервне копіювання з шифруванням, перевірка цілісності копій, використання резервованих сховищ, угоди про рівень обслуговування з провайдером.
106	Ризик збереження даних	Запобігання	Шифрування конфіденційних даних у БД та на рівні файлової системи, інтегрування безпеки у процес розробки, визначення політики зберігання.
202	Ризик фінансової нестабільності	Пом'якшення / Запобігання	Регулярний фінансовий звіт, формування фінансового резерву (5-10%), прозоре планування витрат.
303	Ризик планування	Запобігання	Використання професійного ПЗ, розробка детального графіка, регулярний моніторинг статусу задач.

Для решти ідентифікованих ризиків з рангом нижче середнього застосовується стратегія “Прийняття”. Ці ризики будуть пасивно моніторитися в рамках регулярного огляду реєстру ризиків, але активні заходи з реагування на них не плануються, оскільки їхній потенційний вплив не виправдовує витрат ресурсів.

Запропонований підхід поєднує класифікацію, якісний аналіз на основі матриці ймовірність-вплив та стратегії реагування, що є оптимальним для управління ризиками в умовах динамічної веб-розробки, забезпечуючи баланс між ефективністю та витратами ресурсів.

Висновки до другого розділу

1. На основі порівняльного аналізу основних методологій обґрунтовано вибір гібридного підходу Scrumban, який оптимально поєднує структурованість Scrum та оперативну гнучкість Kanban для ефективного управління проектом розробки веб-орієнтованої інформаційної системи в умовах динамічно змінних вимог.

2. Запропонована дворівнева модель планування, що інтегрує стратегічне прогнозне управління на макrorівні з адаптивною практикою Scrumban на мікрорівні, формалізує ефективний механізм подолання протиріч між передбачуваністю та гнучкістю в управлінні проектами веб-розробки.

3. Для оцінки вартості проекту розробки веб-орієнтованої інформаційної системи запропоновано підхід "знизу-вверх", який забезпечує високу точність і прозорість через декомпозицію робіт, а його сумісність з гнучкими підходами дозволяє ефективно інтегрувати фінансове планування з технічним змістом та календарним планом.

4. Розроблено структурований підхід до управління ризиками на основі матриці ймовірність-вплив, що дозволило ідентифікувати та пріоритезувати критичні загрози проекту, та сформулювати для них чіткий план превентивних та пом'якшувальних заходів.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЦЕСІВ УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Визначення стейкхолдерів та команди проєкту

Управління стейкхолдерами та формування професійної команди є важливим фактором успішної реалізації проєкту розробки веб-орієнтованої інформаційної системи. Відповідно до стандартів управління проєктами, ідентифікація ключових стейкхолдерів, аналіз їхніх інтересів, очікувань та рівня впливу дозволяє проактивно керувати комунікацією, мінімізувати ризики конфлікту цілей та забезпечити широку підтримку проєкту на всіх етапах його життєвого циклу [42].

Для проєкту розробки веб-орієнтованої інформаційної системи визначено таких стейкхолдерів:

- замовник – основний ініціатор та фінансовий бенефіціар проєкту. Його інтереси сконцентровані на отриманні системи, що повністю відповідає бізнес-вимогам, встановленим бюджетним обмеженням та терміном реалізації;
- команда проєкту – безпосередні виконавці робіт, які зацікавлені в успішній реалізації проєкту в рамках виділених ресурсів, професійному розвитку та створенні якісного продукту;
- інвестори – фінансують проєкт та зацікавлені у віддачі від інвестицій. Вони беруть участь у контролі витрат та оцінці ефективності проєкту;
- кінцеві користувачі – особи, які будуть безпосередньо взаємодіяти з системою. Їхні інтереси полягають у зручності, інтуїтивно зрозумілому інтерфейсі, швидкості роботи та функціональності, що ефективно розв’язує повсякденні задачі. Вони є основним джерелом зворотного зв’язку під час тестування;
- постачальники технологій та обладнання – забезпечують проєкт необхідними програмними та апаратними рішеннями. Їхня роль полягає у гарантії стабільної роботи, технічній підтримці та дотриманні угоди про рівень сервісу;
- партнери – сторонні компанії чи організації, чії системи інтегруються з розроблюваним проєктом;

– громадськість – широка аудиторія, яка може бути опосередковано зацікавлена в проєкті. Її інтереси можуть включати доступність, прозорість роботи системи та її відповідність суспільним нормам.

Для виконання робіт було сформовано команду проєкту з урахуванням вимог сучасної веб-розробки. Структура команди включає наступні ключові ролі з чітко визначеними функціональними обов'язками:

– менеджер проєкту – відповідає за загальне керування проєктом, включаючи планування, контроль термінів, бюджету та ресурсів, комунікацію зі стейкхолдерами, управління ризиками та вирішення конфліктів;

– бізнес-аналітик – займається виявленням, аналізом та документуванням вимог. Формалізує їх у технічному завданні, специфікаціях, а також моделює бізнес-процеси за допомогою UML-діаграм;

– архітектор програмного забезпечення – визначає архітектурний вигляд системи, обирає технологічний стек, забезпечує відповідність вимогам до масштабованості, безпеки та продуктивності;

– дизайнер інтерфейсу – розробляє дизайн-концепції, інтерактивні прототипи та забезпечує зручність користувацького досвіду;

– розробник інтерфейсу – реалізує користувацький інтерфейс на основі макетів, забезпечує адаптивність верстки та взаємодіє з серверною частиною;

– розробник серверної частини – відповідає за реалізацію бізнес-логіки, розробку API, інтеграцію з базами даних та зовнішніми сервісами;

– розробник баз даних – проєктує, реалізує та оптимізує структуру бази даних, забезпечує цілісність, безпеку даних та планує процедури резервного копіювання;

– тестувальник – забезпечує якість продукту шляхом розробки тест-кейсів, проведенням різних видів тестування та документування дефектів;

– фахівець технічної підтримки – надає оперативну допомогу учасникам проєкту та кінцевим користувачам на етапі впровадження, збираючи інформацію для подальшого вдосконалення системи.

Для стандартизації очікувань щодо рівня підготовки кожного учасника використовується наступна шкала компетентності: 1 – початківець, 2 – практик, 3 – фахівець, 4 – експерт. Ця шкала дозволяє оцінити наявний досвід, глибину знань та самостійність у вирішенні задач. Вимоги до трудових ресурсів команди проєкту представлені в таблиці 3.1.

Таблиця 3.1 – Вимоги до трудових ресурсів команди проєкту

Ресурс	Рівень підготовки	Ключові навички	Основні компетенції
Менеджер проєкту	4	Управління проєктами, планування, оцінка ризиків, та координація команди	Досвід розробки веб-платформ, управління ресурсами, контроль термінів та взаємодія з зацікавленими сторонами
Бізнес-аналітик	3	Аналіз вимог, створення документації та дослідження ринку	Формування бізнес-вимог, аналіз конкурентів, оцінка рентабельності та відповідність продукту потребам
Архітектор ПЗ	4	Розробка архітектури, вибір технологій та інструментів	Проектування систем, консультування розробників, забезпечення продуктивності та безпеки
Розробник інтерфейсу	3	HTML, CSS, JavaScript, React/Vue.js/Angular, адаптивна верстка, користувацький дизайн	Створення зручного інтерфейсу, оптимізація швидкодії та кросбраузерна адаптація
Розробник серверної частини	3	Серверні технології (Node.js, Java, PHP), робота з базами даних, безпека веб-застосунків	Розробка серверної логіки, робота з БД та API, інтеграція з іншими сервісами
Розробник баз даних	2	SQL, NoSQL, нормалізація, оптимізація запитів	Проектування та адміністрування БД, оптимізація їх взаємодії
Дизайнер інтерфейсу	2	створення дизайн-концепцій та прототипів	Проектування користувацького досвіду, розробка візуального дизайну
Тестувальник	2	Автоматизоване та мануальне тестування, написання тест-кейсів, робота з інструментами тестування	Виявлення та документування помилок, перевірка функціональності, безпеки та продуктивності веб-системи
Фахівець підтримки	1	Технічна підтримка, розв'язування проблем, обслуговування програм	Допомога, комунікація, знання ПЗ та документація

Системне розуміння ролей, інтересів та впливу всіх стейкхолдерів, поєднане з формуванням збалансованої та кваліфікованої команди, створює міцну основу для ефективної комунікації, підвищення якості робіт та успішної реалізації проєкту веб-орієнтованої інформаційної системи.

3.2 Практична реалізація процесів управління проєктом у Microsoft Project

Для управління проєктом розробки веб-орієнтованої інформаційної системи можуть використовуватися різні інструменти управління (Jira, Trello, Asana, Wrike), які дозволяють інтегрувати планування, виконання, моніторинг та контроль усіх аспектів у єдиному середовищі. Для комплексного управління роботами, ресурсами, термінами та бюджетом було обрано професійне програмне забезпечення Microsoft Project.

Перевагами Microsoft Project є глибока інтеграція з екосистемою Microsoft, потужні засоби планування за методом критичного шляху (CPM), можливість побудови детальних календарних планів, моделювання завантаження ресурсів, а також розширені функції формування звітів та аналізу відхилень. Інструмент дозволяє у реальному часі відстежувати прогрес і порівнювати його з базовим планом, що є ключовим для контролю динаміки проєкту.

Етап планування розпочався зі створення детальної ієрархічної структури робіт, яка охопила всі фази: від ініціалізації до закриття проєкту. Це передбачає поділ проєкту на менші керовані частини, що дозволяє повністю охопити всі необхідні задачі, уточнити оцінки та розподілити відповідальність. Такий підхід забезпечує логічну послідовність робіт і відповідність процесу розробки вимогам стандартів управління проєктами [43].

Для сформованої ієрархічної структури робіт було визначено виконавців (див. таблицю А.1), що забезпечує прозорість і контрольованість процесу розробки. На її основі кожному елементу були призначені тривалість, ресурси та залежності, що дозволило автоматично сформувати діаграму Ганта, що зображено на рисунку 3.1 та визначити критичний шлях.

Базовий план було затверджено як орієнтир для подальшого контролю виконання проекту, що гарантує цілісність та узгодженість всього процесу управління.

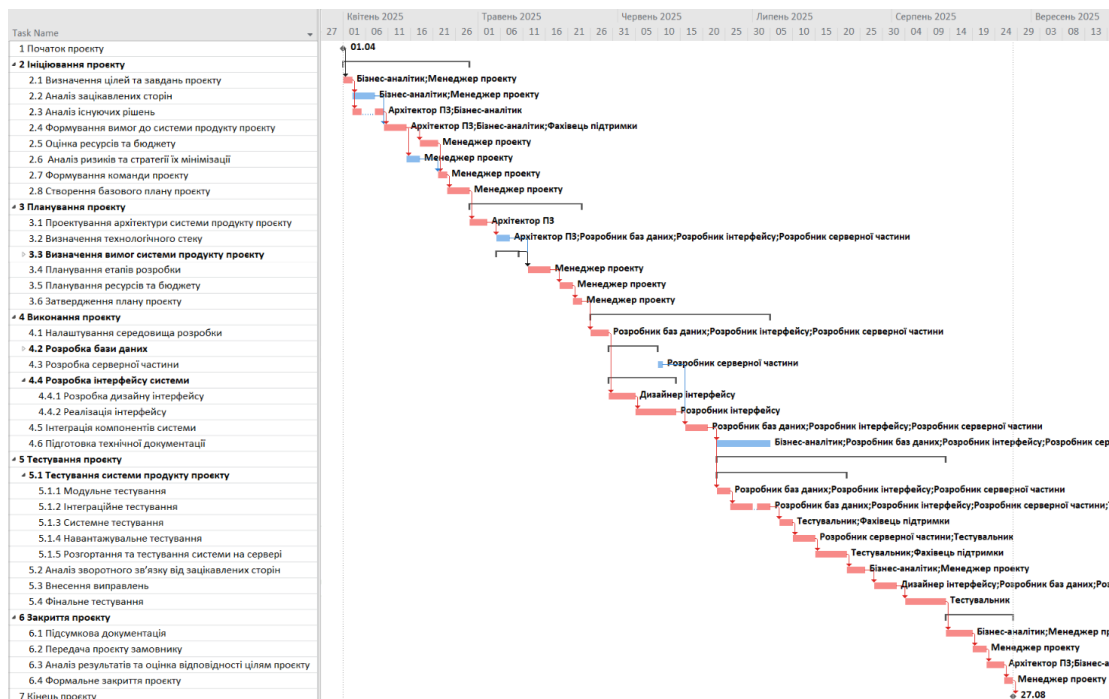


Рисунок 3.1 – Діаграма Ганта проекту розробки веб-системи з виділеним критичним шляхом

Процеси виконання робіт, що супроводжуються постійним моніторингом та контролем, мають ітеративний та динамічний характер. У ході проекту за допомогою Microsoft Project було системно впроваджено та підтримано такі ключові процеси контролю:

- відстеження прогресу – регулярно у план проекту вносяться фактичні дані про відсоток завершення завдань, витрачені години та реальні дати початку чи закінчення. Програма автоматично перераховує графік, показуючи можливі зсуви.
- контроль використання ресурсів – інструменти діаграм ресурсів дозволили виявити перенавантаження розробників та своєчасно провести вирівнювання навантаження, уникаючи вигорання команди. Наприклад, ресурс “Бізнес-аналітик”, що зображено на рисунку 3.2 є перевантажений.

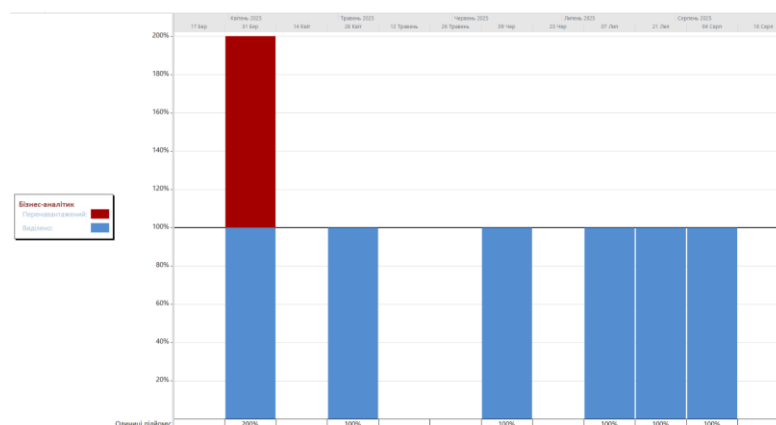


Рисунок 3.2 – Перевантажений ресурс “Бізнес-аналітик”

Вчасне проведення вирівнювання навантаження дозволяє уникати перевантаження трудових ресурсів. На рисунку 3.3 зображено вирівнюваний ресурс “Бізнес-аналітик”, який не містить перевантаження.

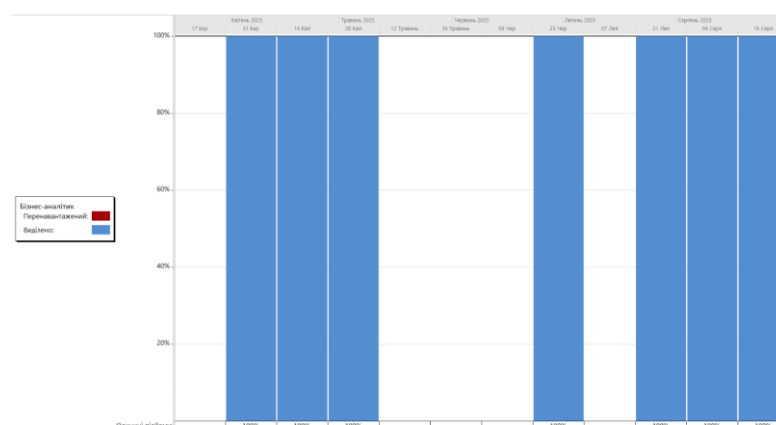


Рисунок 3.3 – Вирівнюваний ресурс “Бізнес-аналітик”

– аналіз відхилень – ключовою функцією є порівняння поточного стану з базовим планом за ключовими показниками: тривалістю, вартістю та обсягом робіт. Звіт про відхилення чітко показує завдання, які випереджають графік або відстають від нього;

– прогнозування – на основі поточного прогресу MS Project дозволяє прогнозувати дату завершення проєкту та очікувані кінцеві витрати, що є критично важливим для прийняття керівних рішень;

– формування звітів – програма надає широкий набір шаблонних та настроєваних звітів для різних зацікавлених сторін. Для команди розробників

генерувалися щоденні списки завдань, для керівництва – щотижневі звіти про стан, що містять візуальну діаграму Ганта з позначками прогресу, аналіз освоєного обсягу та список критичних проблем. Наприклад, на рисунку 3.4 зображено зведений звіт “Огляд ресурсів”.

ОГЛЯД РЕСУРСІВ

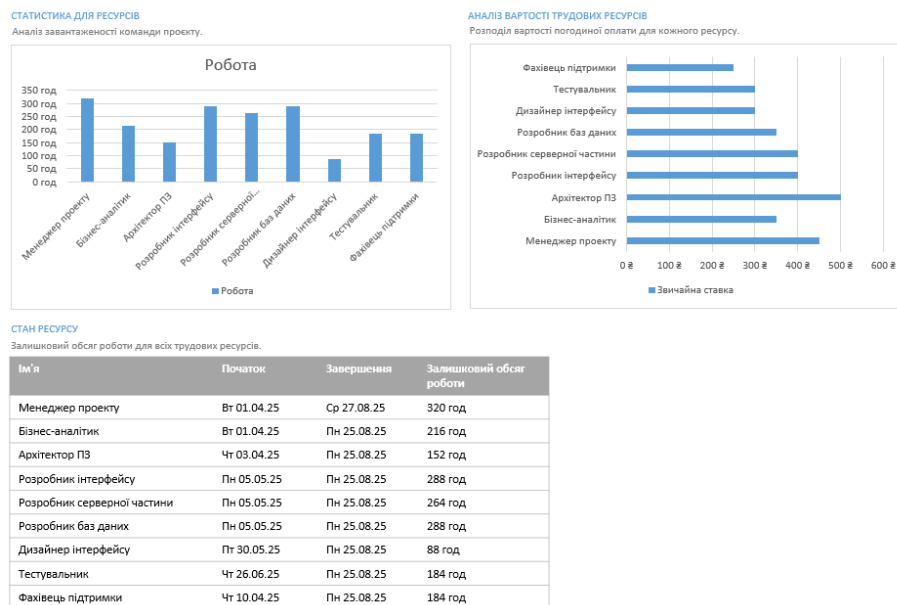


Рисунок 3.4 – Приклад звіту “Огляд ресурсів”, згенерованого в Microsoft Project

Використання Microsoft Project як інструменту управління забезпечило системний підхід до контролю проекту. Інструмент дав можливість не лише фіксувати відхилення, але й проводити їх оперативний аналіз, моделювати наслідки різних керівних рішень та підтримувати прозорість інформації для всієї команди та стейкхолдерів. Це дозволило мінімізувати ризики прострочень та перевищення бюджету, зробивши процес управління веб-розробкою прогнозованим та керованим.

Для визначення тривалості проекту та розробки графіка було застосовано комплексний підхід, що включає оцінку тривалості операцій, визначення залежностей між роботами та побудову календарно-мережевого графіка.

Оцінка тривалості операцій була проведена для кожного робочого завдання, визначеного структурою розбиття робіт. Трудомісткість оцінювалася в людиноднях з використанням експертного методу та методу оцінки за аналогією, з

урахуванням складності технічних завдань та кваліфікації членів команди. Для стандартизації оцінок було прийнято 8-годинний робочий день та 5-денний робочий тиждень.

Залежності між роботами визначені на основі технологічної послідовності виконання завдань. Основні залежності:

- завершення проєктування архітектури системи є передумовою для старту розробки;
- завершення розробки бази даних та серверної частини необхідне для початку інтеграційного тестування;
- отримання зворотного зв'язку від зацікавлених сторін є обов'язковим етапом перед внесенням фінальних виправлень.

На основі проведених оцінок та аналізу залежностей було розроблено розклад виконання робіт (див. таблицю Б.1) з використанням професійного програмного забезпечення для управління проєктами Microsoft Project.

Розроблений розклад виконання робіт дозволяє ідентифікувати низку ключових характеристик проєкту. Загальна тривалість проєкту становить 105 робочих днів з 1 квітня по 27 серпня 2025 року. Критичний шлях проєкту проходить через послідовність ключових етапів: ініціювання проєкту (20 днів), планування проєкту (19 днів), розробку бази даних (7 днів), розробку серверної частини (1 день), інтеграцію компонентів системи (5 днів), тестування системи (21 день), внесення виправлень (5 днів), фінальне тестування (7 днів) та закриття проєкту (11 днів). Ця послідовність завдань вимагає контролю, оскільки будь-яка затримка на цих етапах безпосередньо вплине на загальний термін завершення проєкту.

В рамках проєкту визначено ключові віхи, що відзначають завершення основних фаз: завершення ініціювання проєкту (28.04.2025), завершення планування проєкту (23.05.2025), завершення розробки (04.07.2025), завершення тестування (12.08.2025) та закриття проєкту (27.08.2025).

Важливим аспектом розкладу є наявність резерву часу на некритичних напрямках робіт. Цей резерв дозволяє компенсувати можливі затримки на критичному шляху без порушення загальних термінів реалізації проєкту.

Розроблений розклад виконання робіт забезпечує реалістичні терміни виконання проекту, враховує технологічні залежності між задачами та створює основу для ефективного управління ресурсами на всіх етапах життєвого циклу.

3.3 Бюджетування вартості проекту розробки веб-орієнтованої інформаційної системи

Для визначення фінансових параметрів реалізації проекту розробки веб-орієнтованої інформаційної системи було використано підхід деталізованої оцінки “знизу-вверх”, що базується на структурі розбиття робіт та календарно-мережевому графіку проекту. У таблиці 3.2 наведено бюджет проекту.

Таблиця 3.2 – Бюджет проекту

Стаття витрат	Вартість	Обґрунтування розрахунку
Менеджер проекту (40 днів × 3600 грн/день)	144 000	Керівництво проектом протягом усього життєвого циклу
Бізнес-аналітик (27 днів × 2800 грн/день)	75 600	Аналіз вимог на етапах ініціювання та планування
Архітектор ПЗ (19 днів × 4000 грн/день)	76 000	Проектування архітектури та технологічний супровід
Дизайнер інтерфейсу (11 днів × 2400 грн/день)	26 400	Розробка дизайну системи
Розробник інтерфейсу (36 днів × 3200 грн/день)	115 200	Розробка інтерфейсу
Розробник серверної частини (33 днів × 3200 грн/день)	105 600	Backend-розробка
Розробник баз даних (36 днів × 2800 грн/день)	100 800	Проектування та реалізація БД
Тестувальник (23 днів × 2400 грн/день)	55 200	Тестування на всіх етапах
Фахівець технічної підтримки (23 днів × 2000 грн/день)	46 000	Супровід на етапі впровадження
Хостинг та інфраструктура (5 місяців × 6000 грн)	30 000	Віртуальний сервер, мережеве обладнання
Ліцензійне програмне забезпечення	60 000	IDE, системи управління базами даних, інструменти тестування
Адміністративні витрати	20 000	Комунальні послуги, оренда приміщення, зв'язок
Навчальні заходи	12 000	Тренінги, сертифікація
Резерв на непередбачені витрати (15%)	130020	15% від суми основних витрат (1 144 000 грн)

Загальний бюджет проєкту розробки веб-орієнтованої інформаційної системи становить 996 820 грн. Основною статтею витрат є оплата праці (744 800 грн, 74.7% загального бюджету), що цілком закономірно для проєкту з тривалістю 105 робочих днів та необхідністю висококваліфікованих фахівців.

Витрат на оплату праці менеджера проєкту, розробників інтерфейсу та серверної частини, що відображають складність технічних завдань та відповідальність керівника за загальний успіх проєкту. Витрати на апаратне та програмне забезпечення (90 000 грн, 9% бюджету) включають необхідні технологічні ресурси для розробки та функціонування системи.

Розмір резерву на непередбачені витрати (130 020 грн, 11.3% від основних витрат) визначено з урахуванням ризиків проєкту. Резерв дозволить компенсувати можливі відхилення в термінах, зміну вимог замовника або непередбачені технічні складнощі без порушення загального бюджету проєкту.

Запропонований бюджет базується на детальній оцінці трудомісткості, визначеній в структурі розбиття робіт та враховує реальні ринкові умови оплати праці IT-фахівців.

3.4 Управління якістю продукту проєкту

Управління якістю в проєктах розробки веб-орієнтованих інформаційних систем реалізується через системний процес, що інтегрує три взаємозалежні компоненти: планування якості, забезпечення якості та контроль якості. Цей процес ґрунтується на концепції “якість за будівництвом”, згідно з якою заходи щодо забезпечення якості вбудовуються на кожному етапі життєвого циклу продукту, а не є окремою завершальною стадією [44].

Для визначення атрибутів якості програмного забезпечення слугує міжнародний стандарт ISO/IEC 25010:2023 що містить моделі якості продукту та якості використання [45]. Для веб-орієнтованих інформаційних систем ці моделі потребують спеціалізації та конкретизації у вимірюваних критеріях. В рамках проєкту розробки веб-орієнтованої інформаційної системи ця конкретизація була

виконана шляхом розробки вимог до якості та методів їх перевірки. Вимірювані критерії прийняття та конкретні інструменти їх верифікації наведені в Таблиці 3.3.

Таблиця 3.3 – Критерії прийняття та інструменти перевірки якості продукту

Аспект якості	Критерій прийняття	Інструменти та методики
Якість програмного коду	Дотримання стандартів кодування. Покриття модульними тестами >80%. Відсутність критичних помилок за результатами аналізу.	Статичний аналіз коду (спеціалізоване ПЗ). Автоматизоване тестування модулів. Перевірка коду іншими розробниками.
Функціональна відповідність	100% реалізація вимог з технічного завдання. Успішне проходження фінальних тестів замовником.	Модульне тестування. Тестування взаємодії компонентів. Повне системне тестування. Фінальне тестування приймання
Швидкодія та продуктивність	Час завантаження сторінки менше 2 секунд. Підтримка до 1000 одночасних користувачів. Працездатність системи понад 99.5% часу.	Тестування під навантаженням (спеціальні програми). Постійний моніторинг швидкодії.
Безпека системи	Відсутність основних типів веб-загроз. Шифрування даних при передачі. Успішна перевірка системи фахівцями з безпеки.	Сканування на вразливості. Перевірка безпеки програмного коду. Контрольоване тестування на проникнення.
Зручність використання	Високий бал за спеціальною шкалою оцінки зручності (>80). Успішне проходження тестів зручності. Відповідність стандартам доступності.	Тестування зручності з реальними користувачами. Порівняльне тестування варіантів інтерфейсу. Експертна оцінка інтерфейсу.
Якість документації	Повнота та актуальність. Відповідність стандартам оформлення. Відсутність критичних помилок.	Рецензування документації. Перевірка технічним редактором.

Процес забезпечення якості зосереджений на запобіганні недоліків шляхом впровадження стандартизованих процедур розробки. До ключових методів забезпечення якості належать:

– перевірка програмного коду – систематичний аналіз вихідного коду іншими розробниками для підвищення його якості, виявлення помилок та поширення знань всередині команди [46];

- безперервна інтеграція та безперервне розгортання – автоматизація процесів збирання, тестування та аналізу коду при кожній зміні, що дозволяє швидко виявляти інтеграційні проблеми [47];

- статичний аналіз безпеки застосунків – автоматизована перевірка вихідного коду на наявність вразливостей без його виконання.

Процес контролю якості спрямований на виявлення помилок у готовому програмному продукті шляхом верифікації його відповідності вимогам. Основними методами контролю якості є:

- стратегія рівневого тестування – послідовне проведення модульного, інтеграційного, системного та приймального тестування для комплексної перевірки продукту на різних рівнях деталізації [48];

- тестування продуктивності – оцінка поведінки системи під різним навантаженням для підтвердження відповідності нефункціональним вимогам;

- тестування на проникнення – контрольоване імітування атак зловмисника для виявлення вразливостей безпеки.

Запропонована методологічна база, що поєднує стандартизовані моделі якості, проактивні та реактивні процеси, а також формалізований план з вимірюваними критеріями та інструментами, утворює комплексний підхід до управління якістю. Це забезпечує об'єктивну оцінку якості на всіх етапах життєвого циклу проєкту, дозволяє вчасно вживати коригуючих дій і є гарантією створення веб-орієнтованої інформаційної системи, що відповідає очікуванням замовника та сучасним стандартам якості.

Висновки до третього розділу

1. Ідентифіковано ключових стейкхолдерів проєкту та сформовано збалансовану команду з чітким розподілом ролей і вимогами до компетентності, що становить організаційну основу для ефективної комунікації та успішної реалізації проєкту веб-орієнтованої інформаційної системи.

2. Практична реалізація процесів управління в середовищі Microsoft Project забезпечила системне планування, моніторинг і контроль проєкту, що дозволило ефективно керувати ресурсами, критичним шляхом та відхиленнями, забезпечуючи прогнозованість та прозорість усього життєвого циклу розробки веб-орієнтованої інформаційної системи.

3. На основі декомпозиції робіт та ринкових ставок було сформовано бюджет проєкту за підходом “знизу-вверх”, що визначило загальну вартість проєкту з обґрунтованим розподілом за статтями, де ключовою є оплата праці, та включенням резерву на непередбачені витрати для мінімізації фінансових ризиків.

4. На основі стандарту та запроваджених процесів безперервної інтеграції та рівневого тестування було реалізовано управління якістю проєкту, що забезпечує контроль відповідності продукту встановленим замовником вимогам.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи:

1. Виконано аналіз предметної області управління проєктами веб-орієнтованих інформаційних систем, досліджено сучасні підходи та тенденції, що визначають специфіку планування, організації та контролю веб-проєктів. На основі огляду наукових джерел систематизовано ключові виклики веб-розробки.

2. Обґрунтовано вибір та адаптовано гібридний методологічний підхід, що поєднує стратегічне прогнозне планування та гнучке оперативне управління. Розроблений підхід відповідає вимогам еволюційності, орієнтації на користувача, високої динамічності та технологічної складності веб-проєктів.

3. Сформовано дворівневу модель планування робіт, яка поєднує ієрархічну структуру розбиття робіт на макрорівні та механізми Scrum-спринтів з Kanban-візуалізацією на мікрорівні. Така модель забезпечує баланс прогнозованості, керованості та гнучкої адаптації до змін.

4. Розроблено та адаптовано методики управління вартістю, якістю й ризиками для контексту веб-розробки. Запропоновано підходи “знизу вгору” до оцінки бюджету, систему метрик якості, а також класифікацію ризиків та матрицю “ймовірність-вплив”, орієнтовані на специфіку веб-проєктів.

5. Визначено ключових стейкхолдерів, сформовано ефективну команду та забезпечено організаційну структуру, що сприяє результативній комунікації й успішному управлінню проєктом веб-орієнтованої інформаційної системи.

6. Реалізовано планування, моніторинг і контроль проєкту в Microsoft Project, що дало змогу оптимально управляти ресурсами, визначити бюджет за підходом “знизу-вгору” та забезпечити прозорість і передбачуваність виконання робіт.

7. Управління якістю в проєкті реалізовано через планування якості, забезпечення якості та контроль якості, що гарантує відповідність продукту проєкту вимогам замовника та знижує ризики помилок у процесі розробки.

8. Результати кваліфікаційної роботи плануються до використання (Додаток В).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Waseem M., Liang P., Shahin M., Di Salle A., Márquez G. Design, monitoring, and testing of microservices systems: The practitioners' perspective. *Journal of Systems and Software*. 2021. Vol. 182. P. 111061. DOI: <https://doi.org/10.1016/j.jss.2021.111061>.
2. Maharao C. S. A COMPARATIVE ANALYSIS OF AGILE, WATERFALL, AND HYBRID METHODOLOGIES IN SOFTWARE PROJECT SUCCESS. *ShodhKosh: Journal of Visual and Performing Arts*. 2022. Vol. 3, no. 2. P. 917–926. DOI: <https://doi.org/10.29121/shodhkosh.v3.i2.2022.3396>.
3. Mirzaei M., Mabin V. J., Zwikael O. Customising Hybrid project management methodologies. *Production Planning & Control*. 2024. P. 1–18. DOI: <https://doi.org/10.5220/0010713800003058>.
4. Pilz J., Deutschlander J., Thomaschewski J., Schön E.-M. Integrating Agile Human-Centered Design with Lean UX and Scrum. *Proceedings of the 17th International Conference on Web Information Systems and Technologies (WEBIST 2021)*. 2021. DOI: <https://doi.org/10.5220/0010713800003058>.
5. Faustino D., Gonçalves N., Portela M., Silva A. R. Stepwise migration of a monolith to a microservice architecture: Performance and migration effort evaluation. *Performance Evaluation*. 2024. P. 102411. DOI: <https://doi.org/10.1016/j.peva.2024.102411>.
6. Rojas L. F., Quiñones D., Cubillos C. FRAMUX-EV: A Framework for Evaluating User Experience in Agile Software Development. *Applied Sciences*. 2024. Vol. 14, no. 19. P. 8991. DOI: <https://doi.org/10.3390/app14198991>.
7. Su, R., Li, X. Modular Monolith: Is this the trend in software architecture?. In *Proceedings of the 1st International Workshop on New Trends in Software Architecture*. 2024. P. 10–13. DOI: <https://doi.org/10.48550/arXiv.2401.11867>.
8. Uludağ Ö., Philipp P., Putta A., Paasivaara M., Lassenius C., Matthes F. Revealing the state of the art of large-scale agile development research: A systematic

mapping study. *Journal of Systems and Software*. 2022. P. 111473. DOI: <http://dx.doi.org/10.1016/j.jss.2022.111473>.

9. Leech B., Hanslo R. The Evolution of Agile and Hybrid Project Management Methodologies: A Systematic Literature Review. *arXiv preprint*. 2025. DOI: <https://doi.org/10.48550/arXiv.2511.02859>.

10. Verwijs C., Russo D. Do Agile scaling approaches make a difference? an empirical comparison of team effectiveness across popular scaling approaches. *Empirical Software Engineering*. 2024. Vol. 29, no. 4. DOI: <https://doi.org/10.1007/s10664-024-10481-5>.

11. Poth A., Jacobsen J., Riel A. Systematic Approach to Agile Development in Highly Regulated Environments. *Agile Processes in Software Engineering and Extreme Programming – Workshops*. 2020. P. 111–119. DOI: https://doi.org/10.1007/978-3-030-58858-8_12.

12. Moe N. B., Mikalsen M. Large-Scale Agile Transformation: A Case Study of Transforming Business, Development and Operations. *Lecture Notes in Business Information Processing*. 2020. Vol. 383. P. 115–131. DOI: https://doi.org/10.1007/978-3-030-49392-9_8.

13. Koneru N. M. K. Integrating Security into CI/CD Pipelines: A DevSecOps Approach with SAST, DAST, and SCA Tools. *International Journal of Science and Research Archive*. 2021. Vol. 3, no. 1. P. 250–265. DOI: <https://doi.org/10.30574/ijrsra.2021.3.1.0080>.

14. Jain V. WEB VITALS AND CORE METRICS FOR WEB PERFORMANCE OPTIMIZATION. *International Journal of Core Engineering & Management*. 2023. Vol. 7, no. 6. P. 198–205. DOI: <https://doi.org/10.5281/zenodo.14876757>.

15. Schön E-M., Radtke D., Jordan C. Improving Risk Management in a Scaled Agile Environment. *Agile Processes in Software Engineering and Extreme Programming*. 2020. P. 132-141. DOI: https://doi.org/10.1007/978-3-030-49392-9_9.

16. Li X., d’Aragona D. A., Cerny T. Lenarduzzi V. Exploring microservice ownership and organizational coupling in open-source projects: an empirical study. *Computing*. 2025. Vol. 107. P. 102. DOI: <https://doi.org/10.1007/s00607-025-01454-7>.

17. Shahrads M., Fonseca R., Goiri I., Chaudhry G., Batum P., Cooke J, Laureano E., Tresness C., Russinovich M., Bianchini R. Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider. USENIX ATC'20: Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference. 2020. P. 205–218.
18. Alim S. A Preliminary Analysis of Web Accessibility, Usability, and Security on the Front-Facing Webpages of Leading Universities: Comparing the UK, Israel, and Africa. *International Journal of Web Portals*. 2025. Vol. 15, No.1 2025. P. 1-30. DOI: <https://doi.org/10.4018/IJWP.379681>.
19. Ghattas M., Odeh S., Mora A.M. Predicting Website Performance: A Systematic Review of Metrics, Methods, and Research Gaps (2010–2024). *Computers*. 2025, Vol. 14, P. 446. DOI: <https://doi.org/10.3390/computers14100446>.
20. Єгорченкова Н., Єгорченков О. Інтеграція методологій Agile в проєктах міської трансформації. *Управління розвитком складних систем*. 2024. С. 6–11. DOI: <https://doi.org/10.32347/2412-9933.2024.59.6-11>.
21. Prenner N., Unger-Windeler C., Schneider K. Goals and challenges in hybrid software development approaches. *Journal of Software: Evolution and Process*. 2021. Vol. 33, no. 11. DOI: <https://doi.org/10.1002/smr.2382>.
22. Alsaqqa S., Sawalha S., Abdel-Nabi H. Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies*. 2020. Vol. 14, no. 11. P. 246-270. DOI: <https://doi.org/10.3991/ijim.v14i11.13269>.
23. Almeida F., Bálint B. Approaches for Hybrid Scaling of Agile in the IT Industry: A Systematic Literature Review and Research Agenda. *Information*. 2024. Vol. 15, No. 10. DOI: <https://doi.org/10.3390/info15100592>.
24. Ågren P., Knoph E., Berntsson Svensson R. Agile software development one year into the COVID-19 pandemic. *Empirical Software Engineering*. 2022. Vol. 27. Art. 121. DOI: <https://doi.org/10.1007/s10664-022-10176-9>.
25. Zayat W., Senvar O. Framework Study for Agile Software Development Via Scrum and Kanban. *International Journal of Innovation and Technology Management*. 2020. Vol. 17, No. 4. DOI: <https://doi.org/10.1142/S0219877020300025>.

26. Pashchenko D. S. Refining the Scrum Paradigm: A Comprehensive Research of Software Development Practices (2020–2023). *Computing & AI Connect*. 2024. Vol. 1. DOI: <https://doi.org/10.69709/CAIC.2024.103102>.

27. Donthi S. N. A Scrumban Integrated Approach to Improve Software Development Process and Product Delivery. *The American Journal of Interdisciplinary Innovations and Research*. 2025. Vol. 7, No. 09. P. 70-82. DOI: <https://doi.org/10.37547/tajiir/volume07issue09-07>.

28. Равлюк М., Ситник Й. ОПТИМІЗАЦІЯ AGILE: СИНЕРГІЯ SCRUM І КАНБАН У РОЗРОБЦІ ТА ПІДТРИМЦІ ПРОГРАМНОГО ПРОДУКТУ. *Економіка та суспільство*. 2024. Вип. 63. DOI: <https://doi.org/10.32782/2524-0072/2024-63-120>.

29. Winters T., Manshreck T., Wright H. *Software Engineering at Google: Lessons Learned from Programming Over Time*. Sebastopol: O'Reilly Media, 2020. 602 p.

30. Schwaber K., Sutherland J. *The Scrum Guide*. 2023. URL: <https://scrumguides.org/>.

31. Coleman J., Vacanti D., Johnson C., Singh P., Wester J., Neverdal C., Firlit M., Gilb T., Tendon S. *The Kanban Guide (May 2025)*. 2025. URL: <https://kanbanguides.org/the-kanban-guide/>.

32. Yadav R., Mittal M., Jain R. Adoption of lean principles in software development projects. *International Journal of Lean Six Sigma*. 2020. Vol. 11 No. 2 P. 285–308. DOI: <https://doi.org/10.1108/IJLSS-03-2018-0031>.

33. Gemino A., Horner Reich B., Serrador P. M. Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice?. *Project Management Journal*. 2020. Vol. 52, No. 2. P. 161–175. DOI: <https://doi.org/10.1177/8756972820973082>.

34. Cooper R. G. The 5th-Generation Stage-Gate Idea-to-Launch Process. *IEEE Engineering Management Review*. 2022. Vol. 50, no. 4. P. 43–55. URL: <https://doi.org/10.1109/EMR.2022.3222937>.

35. Johnivan J. R., Webster W. Work Breakdown Structure (WBS) for IT Projects with Examples. 2024. URL: <https://project-management.com/work-breakdown-structure-for-it-projects-wbs/>.

36. Schwaber K., Sutherland J. The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. 2020. URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>.

37. Dahule P. The Strategic Impact of Project Management and Kanban in Enhancing Data Analysis Efficiency. The Eastasouth Journal of Information System and Computer Science. 2023. Vol. 1, No. 2. P. 118–125. DOI: <https://doi.org/10.58812/esiscs.v1i02.494>.

38. Kaur, I., Narula, G.S., Wason, R. Jain V., Baliyan A., Neuro fuzzy – COCOMO II model for software cost estimation. International Journal of Information Technology. 2018. Vol. 10. P. 181–187. DOI: <https://doi.org/10.1007/s41870-018-0083-6>.

39. Practice Standard for Project Estimating / Project Management Institute. 2nd ed. Newtown Square: Project Management Institute, 2019. 150 p.

40. Masso J., Pino F. J., Pardo C., García F., Piattini M. Risk management in the software life cycle: A systematic literature review. Computer Standards & Interfaces. 2020. Vol. 71. DOI: <https://doi.org/10.1016/j.csi.2020.103431>.

41. Грабіна К. В., Шендрік В. В. Метод управління ризиками ІТ-проектів з урахуванням загроз та можливостей. Управління розвитком складних систем. 2023. № 55. С. 18-28. DOI: <https://doi.org/10.32347/2412-9933.2023.55.18-28>.

42. Prebanić K. R., Vukomanović M. Exploring Stakeholder Engagement Process as the Success Factor for Infrastructure Projects. Buildings. 2023. Vol. 13, No. 7. DOI: <https://doi.org/10.3390/buildings13071785>.

43. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 13th ed. Hoboken: John Wiley & Sons, 2022. P. 720.

44. Juran J. M., De Feo J. A. Juran's Quality Handbook: The Complete Guide to Performance Excellence, 7th Edition. 7th ed. New York: McGraw-Hill Education, 2017. P. 968.

45. ISO/IEC 25010:2023. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model. Geneva : International Organization for Standardization, 2023. 50 p.

46. Badampudi S., Unterkalmsteiner M., Britto R. Modern Code Reviews – Survey of Literature and Practice. ACM Transactions on Software Engineering and Methodology. 2023. Vol. 32. No. 4. P. 1-61. DOI: <https://doi.org/10.48550/arXiv.2403.00088>.

47. Shahin M., Ali Babar M., Zhu L. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. IEEE Access. 2017. Vol. 5. P. 3909-3943. DOI: <https://doi.org/10.1109/ACCESS.2017.2685629>.

48. Ammann P., Offutt J. Introduction to Software Testing, 2nd edition. Cambridge University Press, 2017. P. 345 p.

49. Сеньків М., Турчинко І. Управління ризиками проєкту розробки веб-орієнтованої інформаційної системи. Інноваційні дослідження у сфері науки та економіки: збірник наукових праць за матеріалами II Міжнародної науково-практичної конференції. International Scientific Unity. 3-5 грудня 2025 року. Брюссель, Бельгія. С. 761-764. URL: <https://isu-conference.com/en/archive/innovative-research-in-science-and-economy-03-12-25/>.

50. Сеньків М., Турчинко І. Управління проєктами розробки веб-орієнтованих інформаційних систем в контексті гібридизації методологій. Наукові інновації: теоретичні засади та практичний вплив: збірник наукових праць за матеріалами IV Міжнародної науково-практичної конференції. 8-10 грудня 2025 року. Неаполь, Італія. С. 138-141. URL: <https://www.eoss-conf.com/arkhiv/scientific-innovation-theoretical-insights-and-practical-impacts-08-12-25/>.

51. Комар М. П., Саченко А. О., Васильків Н. М., Гладій Г. М., Турченко І. В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Управління проєктами» спеціальності 122 «Комп'ютерні науки». Тернопіль: ЗУНУ, 2024. 32 с.

Додаток А
Структура розбиття робіт проекту

Таблиця А.1 – Структура розбиття робіт проекту розробки веб-орієнтованої інформаційної системи

Код	Назва роботи	Відповідальний виконавець
1.0	Фаза 1. Ініціювання проекту	
1.1	Визначення цілей та завдань проекту	Менеджер проекту, Бізнес-аналітик
1.2	Аналіз зацікавлених сторін	Бізнес-аналітик, Менеджер проекту
1.3	Аналіз існуючих рішень	Бізнес-аналітик, Архітектор ПЗ
1.4	Формування вимог до системи продукту проекту	Бізнес-аналітик, Архітектор ПЗ, Фахівець підтримки
1.5	Оцінка ресурсів та бюджету	Менеджер проекту
1.6	Аналіз ризиків та стратегії їх мінімізації	Менеджер проекту
1.7	Формування команди проекту	Менеджер проекту
1.8	Створення базового плану проекту	Менеджер проекту
2.0	Фаза 2. Планування проекту	
2.1	Проектування архітектури системи продукту проекту	Архітектор ПЗ
2.2	Визначення технологічного стеку	Архітектор ПЗ, Розробник серверної частини, Розробник інтерфейсу, Розробник БД
2.3	Визначення вимог системи продукту проекту	
2.3.1	Деталізація функціональних вимог	Бізнес-аналітик
2.3.2	Деталізація нефункціональних вимог	Архітектор ПЗ, Фахівець підтримки
2.4	Планування етапів розробки	Менеджер проекту
2.5	Уточнення ресурсного плану та бюджету	Менеджер проекту
2.6	Затвердження комплексного плану проекту	Менеджер проекту
3.0	Фаза 3. Виконання проекту	
3.1	Налаштування середовища розробки	Розробник серверної частини, Розробник інтерфейсу, Розробник БД
3.2	Розробка бази даних	
3.2.1	Проектування бази даних	Розробник БД, Архітектор ПЗ
3.2.2	Реалізація бази даних	Розробник БД
3.3	Розробка серверної частини (Backend)	Розробник серверної частини
3.4	Розробка інтерфейсу системи	
3.4.1	Розробка дизайну інтерфейсу	Дизайнер інтерфейсу
3.4.2	Реалізація користувацького інтерфейсу	Розробник інтерфейсу
3.5	Інтеграція компонентів системи	Розробник серверної частини, Розробник інтерфейсу, Розробник БД
3.6	Підготовка технічної документації	Бізнес-аналітик, Розробник серверної частини, Розробник інтерфейсу, Розробник БД, Фахівець підтримки
4.0	Фаза 4. Тестування проекту	
4.1	Тестування системи продукту проекту	
4.1.1	Модульне тестування	Розробник серверної частини, Розробник інтерфейсу, Розробник БД

4.1.2	Інтеграційне тестування	Тестувальник, Розробник серверної частини, Розробник інтерфейсу, Розробник БД
4.1.3	Системне тестування	Тестувальник, Фахівець підтримки
4.1.4	Навантажувальне тестування	Тестувальник, Розробник серверної частини
4.1.5	Розгортання та тестування системи на тестовому сервері	Тестувальник, Фахівець підтримки
4.2	Аналіз зворотного зв'язку від зацікавлених сторін	Менеджер проекту, Бізнес-аналітик
4.3	Внесення виправлень на основі зворотного зв'язку	Розробник серверної частини, Розробник інтерфейсу, Розробник БД, Дизайнер інтерфейсу
4.4	Фінальне приймальне тестування	Тестувальник
5.0	Фаза 5. Завершення проекту	
5.1	Підготовка підсумкової документації	Менеджер проекту, Бізнес-аналітик, Фахівець підтримки
5.2	Передача продукту проекту замовнику	Менеджер проекту
5.3	Аналіз результатів та оцінка відповідності початковим цілям	Команда проекту
5.4	Формальне закриття проекту	Менеджер проекту

Додаток Б

Графік виконання робіт проєкту розробки веб-орієнтованої інформаційної системи

Таблиця Б.1 – Розклад виконання робіт

ID	Назва роботи	Тривалість	Початок	Завершення	Попередні роботи
1	Початок проєкту	0 днів	01.04.25	01.04.25	
2	Ініціювання проєкту	20 днів	01.04.25	28.04.25	
3	Визначення цілей та завдань проєкту	2 днів	01.04.25	02.04.25	1
4	Аналіз зацікавлених сторін	3 днів	03.04.25	07.04.25	3
5	Аналіз існуючих рішень	4 днів	03.04.25	09.04.25	3
6	Формування вимог до системи продукту проєкту	3 днів	10.04.25	14.04.25	4;5
7	Оцінка ресурсів та бюджету	2 днів	18.04.25	21.04.25	6
8	Аналіз ризиків та стратегії їх мінімізації	3 днів	15.04.25	17.04.25	6
9	Формування команди проєкту	2 днів	22.04.25	23.04.25	7;8
10	Створення базового плану проєкту	3 днів	24.04.25	28.04.25	9
11	Планування проєкту	19 днів	29.04.25	23.05.25	
12	Проектування архітектури системи продукту проєкту	4 днів	29.04.25	02.05.25	10
13	Визначення технологічного стеку	3 днів	05.05.25	07.05.25	12
14	Визначення вимог системи продукту проєкту	5 днів	05.05.25	09.05.25	
15	Функціональні вимоги	3 днів	05.05.25	07.05.25	12
16	Нефункціональні вимоги	4 днів	05.05.25	09.05.25	12
17	Планування етапів розробки	5 днів	12.05.25	16.05.25	13;14
18	Планування ресурсів та бюджету	3 днів	19.05.25	21.05.25	17
19	Затвердження плану проєкту	2 днів	22.05.25	23.05.25	18
20	Виконання проєкту	30 днів	26.05.25	04.07.25	
21	Налаштування середовища розробки	4 днів	26.05.25	29.05.25	19
22	Розробка бази даних	7 днів	30.05.25	09.06.25	
23	Проектування бази даних	3 днів	30.05.25	03.06.25	21
24	Реалізація бази даних	4 днів	04.06.25	09.06.25	23
25	Розробка серверної частини	1 день	10.06.25	10.06.25	24
26	Розробка інтерфейсу системи	11 днів	30.05.25	13.06.25	
27	Розробка дизайну інтерфейсу	4 днів	30.05.25	04.06.25	21

28	Реалізація інтерфейсу	7 днів	05.06.25	13.06.25	27
29	Інтеграція компонентів системи	5 днів	16.06.25	20.06.25	28;25
30	Підготовка технічної документації	10 днів	23.06.25	04.07.25	29
31	Тестування проєкту	37 днів	23.06.25	12.08.25	
32	Тестування системи продукту проєкту	21 днів	23.06.25	21.07.25	
33	Модульне тестування	3 днів	23.06.25	25.06.25	29
34	Інтеграційне тестування	6 днів	26.06.25	04.07.25	33
35	Системне тестування	3 днів	07.07.25	09.07.25	34
36	Навантажувальне тестування	3 днів	10.07.25	14.07.25	35
37	Розгортання та тестування системи на сервері	5 днів	15.07.25	21.07.25	36
38	Аналіз зворотного зв'язку від зацікавлених сторін	4 днів	22.07.25	25.07.25	37
39	Внесення виправлень	5 днів	28.07.25	01.08.25	38
40	Фінальне тестування	7 днів	04.08.25	12.08.25	39
41	Закриття проєкту	11 днів	13.08.25	27.08.25	
42	Підсумкова документація	4 днів	13.08.25	18.08.25	40
43	Аналіз результатів та оцінка відповідності цілям проєкту	3 днів	19.08.25	21.08.25	42
44	Передача проєкту замовнику	2 днів	22.08.25	25.08.25	43
45	Формальне закриття проєкту	2 днів	26.08.25	27.08.25	44
46	Кінець проєкту	0 днів	27.08.25	27.08.25	45

Додаток В

Довідка про використання результатів



**Науково-дослідний інститут
інтелектуальних комп'ютерних систем**

**Research Institute for
Intelligent Computer Systems**



Західноукраїнський національний університет
Міністерство освіти і науки України
Інститут кібернетики ім. В.М. Глушкова
Національна академія наук України

West Ukrainian National University
Ministry of Education and Science of
Ukraine
Glushkov Institute of Cybernetics
National Academy of Sciences of Ukraine

Кімн. 11011, 11012
вул. Львівська 11а, Тернопіль, 46009, Україна
Тел. +380 (352) 517-524 internal: 10-012
<https://ics.wunu.edu.ua>
ics@wunu.edu.ua

Office 11011, 11012
Lvivska street, 11, Ternopil, 46009, Ukraine
Tel/Fax: +380 (352) 517-524 internal: 10-012
<https://ics.wunu.edu.ua>
ics@wunu.edu.ua

ДОВІДКА ПРО ВИКОРИСТАННЯ

Кваліфікаційна робота студента групи КНУПм-21 факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету Сеньківа Мар'яна Миколайовича на тему «Управління проектом розробки веб-орієнтованої інформаційної системи» має практичну значимість, а результати її виконання плануються до використання.

Директор НДІ ІКС, к.т.н., проф.

А.О. Саченко

Додаток Г

Копії сертифікатів участі в конференціях




CERTIFICATE

of conference participant

it is hereby certified, that

МАР'ЯН СЕНЬКІВ

took part in the 2nd International Scientific and Practical Conference
«**INNOVATIVE RESEARCH IN SCIENCE AND ECONOMY**»

December 3-5, 2025, Brussels, Belgium
24 Hours of Participation
(0,8 ECTS credits)



Head of the organizing committee



Viktoriiia Tsiundyk



ISU-25/1203-397

isu-conference.com





CERTIFICATE

of participation

Marian Senkiv

took part in the 4th International Scientific and Practical Conference
«**SCIENTIFIC INNOVATION: THEORETICAL INSIGHTS
AND PRACTICAL IMPACTS**»

24 Hours of Participation
(0,8 ECTS credits)



Head of the organizing committee

Helen Volokitina



December 8-10, 2025, Naples, Italy



eoss-conf.com

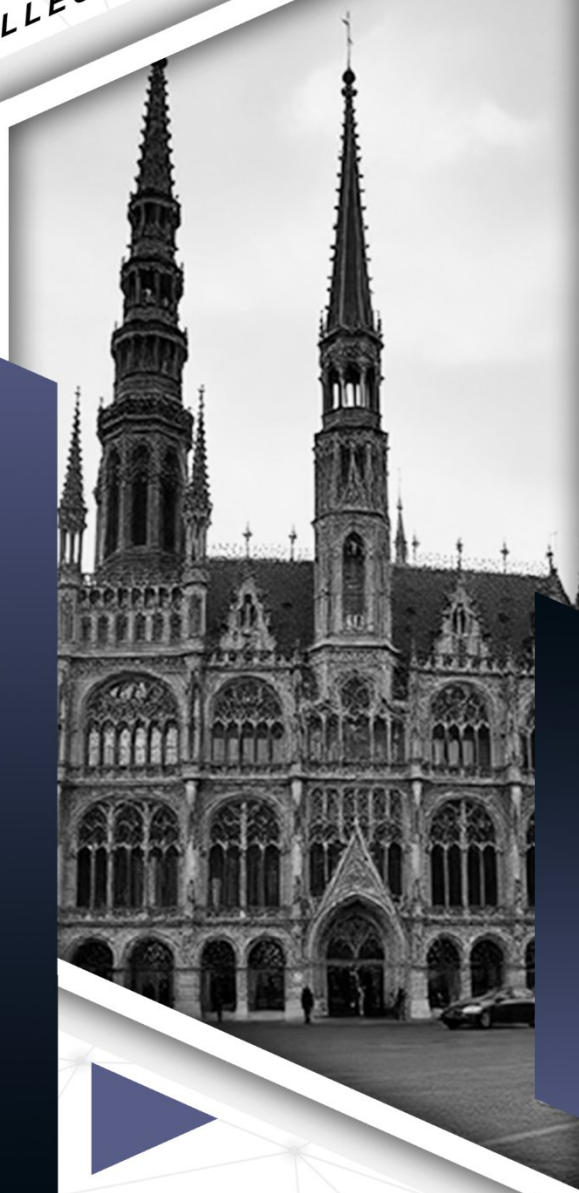


EOSS-25/1208-010

Додаток Д
Копія опублікованих результатів



COLLECTION OF SCIENTIFIC PAPERS



**ISSUE
№48**

**2ND INTERNATIONAL SCIENTIFIC
AND PRACTICAL CONFERENCE**

**INNOVATIVE
RESEARCH
IN SCIENCE
AND ECONOMY**

**DECEMBER 3-5, 2025
BRUSSELS, BELGIUM**





INTERNATIONAL SCIENTIFIC UNITY

2nd International Scientific and Practical Conference
**«Innovative Research in Science and
Economy»**

Collection of Scientific Papers

December 3-5, 2025
Brussels, Belgium

СІМЕЙ ВПО ТА СТАЛИХ РОДИН: ОБҐРУНТУВАННЯ ПРОГРАМИ
КОРЕКЦІЇ.....

SECTION: TECHNICAL SCIENCES

Денисенко К.В., Бутенко О.С.

СТРАТЕГІЇ ВІДНОВЛЕННЯ СІЛЬСЬКОГО ГОСПОДАРСТВА ПІСЛЯ
БОЙОВИХ ДІЙ: МІЖНАРОДНИЙ ДОСВІД ТА АДАПТАЦІЯ ДО
УМОВ УКРАЇНИ..... 746

Єлісєєв Д.А., Алексєєв В.Д., Єлісєєва О.В., Гуркаленко Ю.О

ГАДОЛІНІЙВМІСНА ДОБАВКА ДЛЯ НЕЙТРОНОЧУТЛИВИХ
ОРГАНІЧНИХ СЦИНТИЛЯТОРІВ..... 748

Молодцов Д.Е., Шевченко К.Л.

ОБҐРУНТУВАННЯ НЕЧІТКО-ЛОГІЧНОГО ПІДХОДУ ДО
МОДЕЛЮВАННЯ ВТРАТИ ТИСКУ ПІСЛЯ НАКЛАДАННЯ
МЕХАНІЧНИХ ТУРНИКЕТІВ..... 751

Chychura I., Derenivskiy B., Lakatosh I., Pylyp V.

THRESHOLD PARAMETERS OF THE SENSITIVE ELEMENT OF A
FIBER-OPTIC TEMPERATURE SENSOR (FOTS)..... 755

Долгушин В., Задворний А., Пантась С.

ПОРІВНЯННЯ ПРИКЛАДНИХ АЛГОРИТМІВ КЛАСТЕРНОГО
АНАЛІЗУ ГЕОПРОСТОРОВИХ ДАНИХ ДЛЯ ПІДВИЩЕННЯ
ТОЧНОСТІ ЦІЛЕВКАЗАННЯ В СИСТЕМАХ ПІДТРИМКИ
ПРИЙНЯТТЯ РІШЕНЬ..... 756

Штелиха О.В., Петрушко І.А.

АРХІТЕКТУРА ТА АНАЛІЗ ШІ ЗАСТОСУНКУ ДЛЯ
АВТОМАТИЧНОГО ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН У
TELEGRAM/WEBEX..... 758

Сеньків М., Турченко І.

УПРАВЛІННЯ РИЗИКАМИ ПРОЄКТУ РОЗРОБКИ
ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ..... 761

Курінько Д. Д.

ПРОБЛЕМИ ВИЯВЛЕННЯ ПОТРЕБИ У РЕФАКТОРИНГУ В
ОБ'ЄКТНО-ОРІЄНТОВАНОМУ КОДІ..... 764

2. Buzea M. C., Trausan-Matu S., Rebedea T. Automatic fake news detection for romanian online news // Information. 2022. Vol. 13, No. 3. P. 151. URL: <https://www.mdpi.com/2078-2489/13/3/151>.
3. Taher Y., Moussaoui A., Moussaoui F. Automatic fake news detection based on deep learning, FasTtext and news title // International Journal of Advanced Computer Science and Applications. 2022. Vol. 13, No. 1. URL: <https://www.proquest.com/openview/d33c6e732dbefeca80f1a02956418441/1?pq-origsite=gscholar&cbl=5444811>.

УПРАВЛІННЯ РИЗИКАМИ ПРОЄКТУ РОЗРОБКИ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Мар'ян Сеньків

здобувач вищої освіти магістерського рівня

Ірина Турченко

к.т.н., доцент

Кафедра інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, м. Тернопіль, Україна

Розробка веб-орієнтованих інформаційних систем супроводжується високим рівнем невизначеності, зумовленим динамічними вимогами замовників, технологічною складністю та кіберзагрозами [1-3]. Метою цієї роботи є ідентифікація ключових ризиків, характерних для створення веб-орієнтованих систем, та розробка практичних підходів до їхнього пом'якшення. Ефективне управління ризиками на всіх етапах реалізації проєкту дає змогу запобігти критичним помилкам і забезпечити своєчасне завершення робіт [4, 5]. Результати дослідження орієнтовані на застосування керівниками ІТ-проєктів, розробниками та аналітиками у галузі веб-технологій.

У ході виконання проєкту розробки веб-орієнтованої інформаційної системи, було виділено такі ризики:

- ризик вибору технологій (101) – пов'язаний з використанням застарілих або непридатних інструментів;
- ризик інтеграції (102) – пов'язаний з виникненням проблем під час підключення зовнішніх сервісів;
- ризик безпеки (103) – пов'язаний з недостатнім захистом системи від різноманітних атак;
- ризик хостингу (104) – пов'язаний з використанням ненадійного провайдера або недостатня потужність серверів обробляти навантаження;
- ризик втрати даних (105) – пов'язаний з втратою важливої інформації через збої, помилки чи кібератаки;
- ризик збереження даних (106) – пов'язаний з неналежним або небезпечним зберіганням персональних та конфіденційних даних;
- ризик перевищення бюджету (201) – пов'язаний з непередбачуваними

витратами, затримками чи доопрацюванням, що призводить до виходу за межі запланованого фінансування;

– ризик фінансової нестабільності (202) – пов’язаний із затримкою чи припиненням фінансування, що загрожує зупинкою проекту;

– ризик недооцінки обсягу робіт (301) – пов’язаний з неправильною оцінкою складності завдань;

– ризик зміни вимог (302) – пов’язаний з надмірним коригуванням вимог проекту з боку замовника;

– ризик планування (303) – пов’язаний з неправильним (виявленням помилок) плануванням виконання проекту;

– ризик визначення вимог (304) – пов’язаний з нечітким або неповним описом очікувань замовника щодо функцій системи;

– ризик залучення замовника (305) – пов’язаний з відсутністю своєчасного зворотного зв’язку або затримка у прийнятті рішень з боку замовника;

– ризик проміжного тестування (306) – пов’язаний з виявленням помилок у системі через пасивну участь замовника у валідації проміжних результатів;

– ризик ліцензійних умов (401) – пов’язаний з використанням несанкціонованих сторонніх матеріалів;

– ризик вимог законодавства (402) – пов’язаний з невідповідністю системи вимогам законодавства щодо обробки персональних даних;

– ризик юридичних спорів (403) – пов’язаний з невідповідністю умовам договору або їх невиконанням;

– ризик кваліфікації фахівців (501) – пов’язаний з недостатньою кваліфікацією команди, яка буде виконувати проєкт;

– ризик плинності кадрів (502) – пов’язаний з виходом з проєкту ключових розробників або інших спеціалістів;

– ризик комунікації (503) – пов’язаний з конфліктом розподілу обов’язків і взаємодії всередині команди та стейкхолдерів.

Наведені ризики були проаналізовані щодо ймовірності їх виникнення та потенційного впливу на проєкт. Результати оцінки наведені в таблиці 1, де також розраховано ранг кожного ризику (добуток ймовірності на вплив).

Таблиця 1 – Перелік та оцінка ідентифікованих ризиків

Код ризику	Ідентифікований ризик	Ймовірність	Вплив	Ранг ризику
101	Ризик вибору технологій	3	4	12
102	Ризик інтеграції	4	3	12
103	Ризик безпеки	4	5	20
104	Ризик хостингу	2	4	8
105	Ризик втрати даних	3	5	15
106	Ризик збереження даних	3	5	15
201	Ризик перевищення бюджету	3	4	12
202	Ризик фінансової нестабільності	3	5	15
301	Ризик недооцінки обсягу робіт	4	4	16

Продовження табл. 1

Код ризику	Ідентифікований ризик	Ймовірність	Вплив	Ранг ризику
302	Ризик зміни вимог	5	4	20
303	Ризик планування	3	5	15
304	Ризик визначення вимог	3	4	12
305	Ризик залучення замовник	4	3	12
306	Ризик проміжного тестування	3	4	12
401	Ризик ліцензійних умов	2	3	6
402	Ризик вимог законодавства	2	5	10
403	Ризик юридичних спорів	2	4	8
501	Ризик кваліфікації фахівців	3	4	12
502	Ризик плінності кадрів	3	3	9
503	Ризик комунікації	4	3	12

Для оцінки ризиків було використано п'ятирівневу шкалу [6]:

- прийнятний (1-5) – мінімальний вплив, не потребує дій;
- низький (6-10) – можна прийняти без пом'якшення впливу;
- середній (11-15) – потребує заходів для пом'якшення впливу;
- високий (16-20) – необхідні активні дії для управління ризиком;
- критичний (21-25) – вимагає негайного реагування.

За результатами оцінки ймовірності та впливу побудовано карту ризиків проекту розробки веб-орієнтовної інформаційної системи, що зображено на рисунку 1.

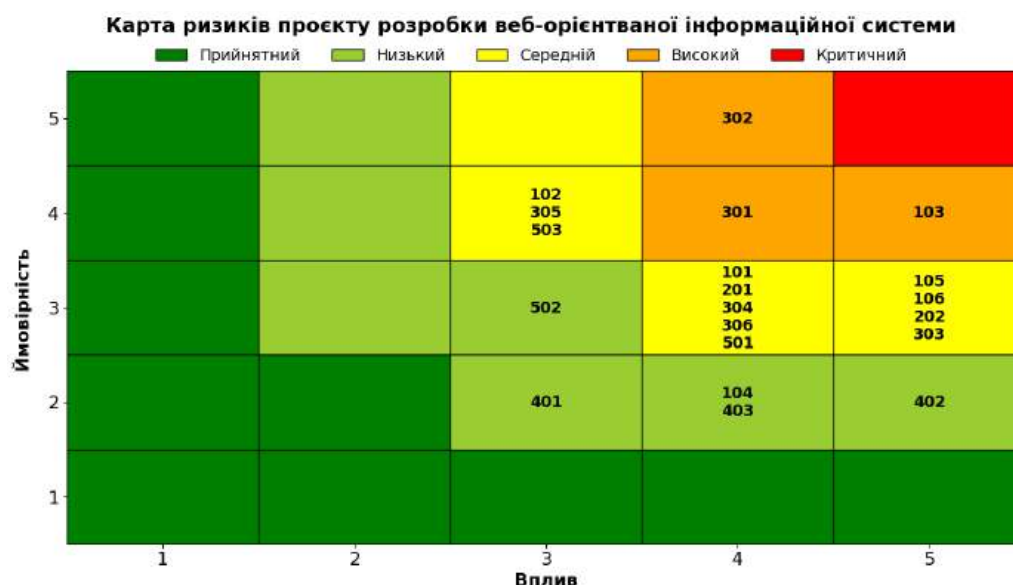


Рисунок 1 – Карта ризиків проекту розробки веб-орієнтовної інформаційної системи (розроблено автором)

Управління ризиками є важливим для успішного виконання проекту розробки веб-орієнтовної інформаційної системи. В результаті було ідентифіковано та проаналізовано ключові ризики, серед яких найбільш високий рівень загрози мають ризик безпеки (на 57% більше за середнє значення рангу),

ризик зміни вимог (на 57% більше) та у меншій мірі ризик недооцінки обсягу робіт (на 26% більше). Результати аналізу можуть бути використані для побудови стратегії пом'якшення ризиків у проєкті розробки веб-орієнтованої інформаційної системи.

Список використаних джерел

1. Данченко, О.Б., Альба, В.О., Березенський, Р.В. and Савіна, О.Ю., 2021. Ідентифікація та аналіз ризиків проєктів ІТ-аудиту. Вісник Національного технічного університету «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проєктами, (1 (3)), pp.24-31.
2. Леонов, С., Семко, О., 2023. Інформаційна технологія управління інформаційними ризиками для проєктів цифрової трансформації бізнесу. Управління розвитком складних систем, (56), pp.64-69.
3. ISO 31000:2018 - Risk management – Guidelines. International Organization for Standardization, 2018. url: <https://www.iso.org/standard/65694.html>
4. Fedushko, S., Trach, O., Syerov, Y., Kryvinska, N. and Calhoun, J.R., 2022. Web project development: emergency management. Journal of Web Engineering, 21(8), pp.2257-2286.
5. Cagliano, A. C., Grimaldi, S. and Rafele, C. (2014) Choosing project risk management techniques. A theoretical framework, Journal of Risk Research, 18(2), pp. 232–248. doi: 10.1080/13669877.2014.896398.
6. Li, J., Bao, C. and Wu, D., 2018. How to design rating schemes of risk matrices: a sequential updating approach. Risk Analysis, 38(1), pp.99-117.

ПРОБЛЕМИ ВИЯВЛЕННЯ ПОТРЕБИ У РЕФАКТОРИНГУ В ОБ'ЄКТНО-ОРІЄНТОВАНОМУ КОДІ

Курінько Дмитро Дмитрович

аспірант

Національний університет «Одеська Політехніка»

Вступ. Сучасні об'єктно-орієнтовані програмні системи характеризуються високою структурною складністю, динамічною еволюцією та значним технічним боргом, що накопичується у процесі розробки. Рефакторинг як практика підтримки якості коду відіграє ключову роль у забезпеченні довгострокової життєздатності систем, проте визначення оптимальних моментів і ділянок для його застосування залишається складним і багатоаспектним завданням. У великих та різномірних кодових базах ручне виявлення проблемних фрагментів стає не лише трудомістким, а й часто неможливим без спеціалізованих інструментів.



ISSUE
N°65



EUROPEAN OPEN
SCIENCE SPACE

COLLECTION OF SCIENTIFIC PAPERS



4TH INTERNATIONAL
SCIENTIFIC
AND PRACTICAL
CONFERENCE

SCIENTIFIC INNOVATION:
THEORETICAL INSIGHTS
AND PRACTICAL IMPACTS

DECEMBER 8-10, 2025, NAPLES, ITALY





**EUROPEAN OPEN
SCIENCE SPACE**

Proceedings of the 4th International Scientific
and Practical Conference

**"Scientific Innovation: Theoretical Insights and
Practical Impacts"**

December 8-10, 2025

Naples, Italy

Collection of Scientific Papers

Naples, 2025

Яремчук Р.Р.

ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНА ХАРАКТЕРИСТИКА АТ КБ
"ПРИВАТБАНК" ЯК ОБ'ЄКТА ДОСЛІДЖЕННЯ..... 113

Дорошенко А.П., Дорошенко О.О., Ларін О.М., Чупир О.С.

ОБОРОТНІ ТА НЕОБОРОТНІ АКТИВИ АГРАРНИХ
ПІДПРИЄМСТВ: МЕТОДИЧНІ ПІДХОДИ ДО АНАЛІЗУ
ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ..... 116

Hryshchuk N., Zhmurko V., Pogorilets A.

THEORETICAL RESEARCH INTO THE ESSENCE OF COMPETITIVE
ADVANTAGES OF ENTERPRISES IN WAR CONDITIONS..... 120

Section: Geography, Geology and Geodesy

Semehen O., Taranova N., Nadvodskiy A., Budzinskyi P.

PREREQUISITES FOR THE INTRODUCTION OF RENEWABLE
ENERGY SOURCES IN THE TERRITORY OF THE DOLYNA URBAN
HROMADA..... 124

Section: History and Cultural Studies

Григорук Н.

ДВОПАРТІЙНА СИСТЕМА У США НАПРИКІНЦІ ХІХ ст. 127

Савченко А.

ОРНАМЕНТИ В УКРАЇНСЬКІЙ КУЛЬТУРІ ТА ЇХ ЗНАЧЕННЯ..... 129

Гуцуляк С., Боян-Гладка С.

НАРОДНІ ЗАСОБИ ЛІКУВАННЯ ТВАРИННОГО ПОХОДЖЕННЯ
У НАСЕЛЕННЯ УКРАЇНСЬКИХ КАРПАТ..... 132

Section: Information Technology, Cyber Security and Computer Engineering

Сеньків М., Турченко І.

УПРАВЛІННЯ ПРОЄКТАМИ РОЗРОБКИ ВЕБ-ОРІЄНТОВАНИХ
ІНФОРМАЦІЙНИХ СИСТЕМ В КОНТЕКСТІ ГІБРИДИЗАЦІЇ
МЕТОДОЛОГІЙ..... 138

Давидович І.В., Настенко Є.А.

КЛАС-БАЛАНСНИЙ ПІДХІД ДО ФІЛЬТРАЦІЇ ІНТЕНСИВНОСТЕЙ
У КТ-ЗОБРАЖЕННЯХ ЛЕГЕНЬ..... 141

Section: Information Technology, Cyber Security and Computer Engineering

УПРАВЛІННЯ ПРОЄКТАМИ РОЗРОБКИ ВЕБ- ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ В КОНТЕКСТІ ГІБРИДИЗАЦІЇ МЕТОДОЛОГІЙ

Сеньків Мар'ян

здобувач вищої освіти магістерського рівня

Науковий керівник:

Турченко Ірина

к.т.н., доцент

Кафедра інформаційно-обчислювальних систем і управління

Західноукраїнський національний університет

м. Тернопіль, Україна

Анотація. У статті представлено результати дослідження сучасних підходів до управління проєктами зі створення веб-орієнтованих інформаційних систем. Проаналізовано особливості цієї сфери, зокрема поступовий розвиток продукту, технологічну складність та орієнтацію на користувача. Розглянуто актуальні методи планування та координації робіт, включаючи гнучкі підходи, масштабовані методики, інтеграцію DevOps-практик та використання даних для прийняття управлінських рішень. Обґрунтовано, що потрібен практичний, цілісний підхід до управління проєктом, який поєднує швидку реакцію на зміни і контроль основних архітектурних рішень.

Ключові слова: веб-орієнтовані інформаційні системи; управління проєктами; Agile; DevOps; гібридні моделі; життєвий цикл ПЗ; UX.

Вступ. Розробка веб-орієнтованих інформаційних систем є однією з найбільш динамічних сфер сучасних технологій. Висока конкуренція, швидкі зміни вимог, різноманітність технологій та високі очікування користувачів формують нові виклики для управління проєктами. Класичні лінійні підходи вже не забезпечують належної швидкості та адаптивності, а виключно гнучкі методики не завжди здатні підтримати структурованість великих або довготривалих проєктів.

У сучасних компаніях усе частіше застосовуються комбіновані методи управління, які поєднують елементи Agile, планування наперед, DevOps-підходів та постійного моніторингу результатів. Це вимагає від менеджера проєкту не тільки координації команди, а й побудови процесу, який забезпечить узгодженість усіх етапів створення і розвитку веб-системи.

Мета дослідження. Метою дослідження є аналіз сучасних методів управління проєктами з розробки веб-орієнтованих інформаційних систем та

формування основи для створення цілісного підходу, який забезпечує ефективну організацію робіт у складних та динамічних умовах.

Аналіз літератури та відомих рішень. Сучасні дослідження демонструють стійкий перехід від жорстко регламентованих моделей управління проектами до гнучких і комбінованих підходів. Література підкреслює, що хоча Agile залишається домінуючою методологією, на практиці команди адаптують його під специфіку веб-проектів, поєднуючи елементи різних моделей. У роботі [1] зазначається, що гібридні підходи часто демонструють кращі результати в умовах складних або багатокомпонентних проектів.

Паралельно дослідження масштабованих фреймворків, таких як SAFe та LeSS, підтверджують їхню ефективність для великих організацій, де необхідна синхронізація роботи численних команд [2]. У літературі наголошується, що поєднання прогностичного планування для архітектурної частини з ітеративною розробкою інтерфейсів та бізнес-логіки дозволяє зберігати баланс між передбачуваністю та гнучкістю.

Окремий блок досліджень присвячено DevOps-практикам, які значно пришвидшують випуск нових версій продукту завдяки автоматизації, безперервній інтеграції та контролю якості [3]. Приділяється увага також ролі UX у сучасних командах: роботами [4] доведено, що регулярна взаємодія дизайнерів із розробниками на всіх етапах підвищує якість користувацького досвіду та зменшує кількість переробок.

Важливим напрямом є й дослідження мікросервісної архітектури, яка трансформує підходи до організації робіт. Команди отримують більше автономії й відповідають за повний життєвий цикл сервісів від розробки до супроводу. Хмарні технології доповнюють цей підхід, забезпечуючи масштабованість, гнучке використання ресурсів та стійкість систем [5].

Сучасна наукова література охоплює окремі аспекти управління веб-проектами: методології, архітектури, UX, DevOps і масштабованість. Однак комплексні моделі, які об'єднують ці напрями в єдину систему управління, залишаються недостатньо представленими, що підкреслює актуальність даного дослідження.

Результати дослідження. На основі аналізу було сформовано узагальнений підхід до управління проектами веб-орієнтованих систем. Він враховує потребу в адаптації процесів, контролі якості в реальному часі та чіткому розподілі обов'язків у складних технічних умовах.

Веб-системи розглядаються як такі, що постійно змінюються та розвиваються. Тому на початкових етапах важливо зменшувати надмірне документування, робити акцент на поступовому уточненні вимог і підтримувати можливість масштабування архітектури. Важливо також мати механізми швидкого внесення змін без втрати стабільності роботи системи.

Технологічна складність, пов'язана з використанням мікросервісів, API та хмарних сервісів, потребує чіткого поділу відповідальності між командами,

впровадження інструментів для спостереження за роботою системи та планування частих оновлень. Це дозволяє швидко реагувати на зміну вимог і усувати проблеми.

Важливою частиною керування веб-проектами є орієнтація на користувача. Регулярні тестування інтерфейсу, збір відгуків і аналіз поведінки користувачів у реальному часі допомагають покращувати продукт і приймати обґрунтовані рішення.

Поєднання гнучких практик та елементів планування наперед забезпечує баланс між швидкістю та передбачуваністю. Наприклад, архітектуру доцільно планувати детальніше, тоді як функціональні модулі можуть створюватися ітеративно. Ризики варто інтегрувати у повсякденну роботу через їх регулярний перегляд та оцінку.

DevOps-практики підтримують швидкі цикли розробки та розгортання, забезпечують узгодженість між командами розробки й експлуатації та допомагають швидше отримувати результати. Постійний моніторинг продуктивності та якості доповнює цей підхід, створюючи більш прозорий процес і підвищуючи передбачуваність результатів.

Узагальнений підхід забезпечує цілісне управління веб-проектом і сприяє більшій стабільності, гнучкості та якості кінцевого продукту.

Висновки. Дослідження показало, що управління веб-проектами це складний комплекс робіт, який поєднує методологічні, організаційні та технічні підходи. Особливості сучасних веб-систем такі, як швидкі зміни вимог, багато компонентів, орієнтація на користувача потребують застосування комбінованих методів управління. У контексті управління проектами це означає, що команда не використовує одну методологію в оригінальному вигляді, а інтегрує елементи кількох підходів, здійснюючи гібридизацію методологій для поєднання переваг і компенсації недоліків. Хоча література детально описує окремі напрями, комплексний підхід, який об'єднує методи управління, UX-практики та DevOps, поки що мало розроблений. Це відкриває простір для подальших досліджень і створення практичних рекомендацій.

Список використаних джерел

1. Gemino A., Horner Reich B., Serrador P. M. Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice? *Project Management Journal*. 2021. Vol. 52, no. 2. P. 161–175.
2. Verwijs C., Russo D. Do Agile scaling approaches make a difference? An empirical comparison of team effectiveness across popular scaling approaches. *Empirical Software Engineering*. 2024. Vol. 29. Art. 75.
3. Humble J., Kim G. *Accelerate: The Science of Lean Software and DevOps*. Portland : IT Revolution, 2019.
4. Exploring principles of user-centered agile software development: A literature review / M. Brhel et al. *Information and Software Technology*. 2015. Vol. 61. P. 163–181. DOI: <https://doi.org/10.1016/j.infsof.2015.01.004>.