

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Управління ІТ-проектами»)

на тему:

**«Розробка проєкту веб-додатку для ефективного
управління часом на основі методу Pomodoro»**

Виконав студент групи УІТПм-23-1
МОСКВІН Дмитро Володимирович

Керівник роботи:
СВІНЦИЦЬКА Олександра Миколаївна

Рецензент:
АНТОНЮК Дмитро Сергійович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«01» жовтня 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **МОСКВІН Дмитро Володимирович**

Керівник роботи: **СВІНЦИЦЬКА Олександра Миколаївна**

Тема роботи: **«Розробка проєкту веб-додатку для ефективного управління часом на основі методу Pomodoro»,**

затверджена Наказом закладу вищої освіти від «01» жовтня 2024 р. № 522/с

Вихідні дані для роботи: **розроблений проєкт веб-додатку, що дозволяє користувачам та командам ефективно управляти своїм часом за допомогою методу Pomodoro, поліпшення комунікації та взаємодії між членами команди.**

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Свінцицька О.М.	01.10.2024	01.10.2024
2	Свінцицька О.М.	25.10.2024	25.10.2024
3	Свінцицька О.М.	12.11.2024	12.11.2024
4	Свінцицька О.М.	03.12.2024	03.12.2024

РЕФЕРАТ

Метою даного дослідження є систематичне вивчення методів управління часом та їх впливу на особисту і командну продуктивність, з особливим акцентом на інформаційні технології (ІТ) та розробку веб-додатку, що забезпечує ефективне управління часом на основі методу Pomodoro. Для досягнення зазначеної мети було використано комплексний підхід, що включає теоретичний аналіз наукових джерел, а також емпіричні дослідження, спрямовані на виявлення ключових факторів, які впливають на ефективність управління часом.

Результати дослідження свідчать про те, що впровадження методу Pomodoro суттєво підвищує продуктивність користувачів у порівнянні з традиційними методами управління часом. Зокрема, метод сприяє покращенню концентрації та зменшенню втоми, що, в свою чергу, позитивно позначається на якості виконання завдань.

В результаті проведеного дослідження було розроблено проєкт веб-додатку під назвою «Days», який інтегрує метод Pomodoro в командні процеси та надає інструментарій для підвищення організованості роботи. Цей додаток має на меті оптимізацію управлінських практик у командах, що працюють над ІТ-проєктами, та забезпечення більш ефективного використання часу.

Магістерська атестаційна робота на тему «Розробка проєкту веб-додатку для ефективного управління часом на основі методу Pomodoro» складається з вступу, 4-х розділів, висновків, списку використаної літератури із 37 найменувань (4 стор.), 0 додатків, 36 таблиць та 17 ілюстрацій по тексту, з них 0 оформлено на окремій сторінці. Повний обсяг роботи становить 90 сторінки, з них 79 сторінок основного тексту.

Ключові слова: УПРАВЛІННЯ ЧАСОМ, МЕТОД POMODORO, ВЕБ-ДОДАТОК, КОМАНДНА РОБОТА, ПРОДУКТИВНІСТЬ, РОЗРОБКА ПРОЄКТУ.

ABSTRACT

The purpose of this study is to systematically examine time management methods and their impact on personal and team productivity, with a particular focus on information technology (IT) and the development of a web application that provides effective time management based on the Pomodoro method. To achieve this goal, an integrated approach was used, including theoretical analysis of scientific sources, as well as empirical research aimed at identifying key factors that affect the effectiveness of time management.

The results of the study show that the implementation of the Pomodoro method significantly increases user productivity compared to traditional time management methods. In particular, the method helps to improve concentration and reduce fatigue, which, in turn, has a positive effect on the quality of task performance.

As a result of the study, a web application project called Days was developed that integrates the Pomodoro method into team processes and provides tools to improve work organization. This application aims to optimize management practices in teams working on IT projects and ensure more efficient use of time.

The master's thesis on "Development of a web application project for effective time management based on the Pomodoro method" consists of an introduction, 4 chapters, conclusions, a list of references of 37 titles (4 pages), 0 appendices, 36 tables and 17 illustrations in the text, of which 0 are on a separate page. The total volume of the work is 90 pages, including 79 pages of the main text.

Keywords: TIME MANAGEMENT, POMODORO METHOD, WEB APPLICATION, TEAMWORK, PRODUCTIVITY, PROJECT DEVELOPMENT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ПРОБЛЕМИ РОЗРОБКИ ІТ-ПРОЄКТІВ ТА УПРАВЛІННЯ ЧАСОМ.....	10
1.1. Основні підходи до процесу розробки ІТ-проєктів	10
1.2. Теоретичні основи та аналіз методів управління часом	17
1.3. Світовий досвід вирішення проблем ефективного управління часом	23
РОЗДІЛ 2. ОПИС ПРОДУКТУ ТА ВИМОГ	30
2.1. Вибір цільової аудиторії та розробка моделей користувачів	30
2.2. Формулювання вимог до основного функціоналу додатку, формування беклогу продукту.....	40
2.3. Проектування макету додатку	46
РОЗДІЛ 3. ПЛАН ПРОЄКТУ РОЗРОБКИ ІТ-ПРОДУКТУ	53
3.1. Стейкхолдери проєкту та їх класифікація	53
3.2. Декомпозиція робіт, етапи та компоненти проєкту.....	57
3.3. Реєстр ризиків.....	66
РОЗДІЛ 4. РОЗРОБКА БІЗНЕС-МОДЕЛІ ПРОДУКТУ	74
1.1. Канва бізнес-моделі продукту	74
1.2. Модель ціноутворення додатку	77
1.3. План фінансування/інвестування	81
ВИСНОВКИ	84
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface (прикладний програмний інтерфейс)

CI/CD – Continuous Integration/Continuous Delivery (безперервна інтеграція та доставка)

IT – Information Technology (інформаційні технології)

JTBD – Jobs-to-be-Done (методологія аналізу потреб користувачів)

KPI – Key Performance Indicators (ключові показники ефективності)

MVP – Minimum Viable Product (мінімально життєздатний продукт)

QA – Quality Assurance (забезпечення якості)

SEO – Search Engine Optimization (оптимізація для пошукових систем)

UI – User Interface (користувацький інтерфейс)

UX – User Experience (досвід користувача)

WBS – Work Breakdown Structure (декомпозиція робіт)

PMI – Project Management Institute

ВСТУП

У сучасному світі розробка ІТ-проектів стає все більш складним та багатогранним процесом, що вимагає системного підходу та врахування численних факторів успіху. Особливої актуальності набуває питання ефективного управління часом як окремих розробників, так і команд в цілому.

Актуальність теми дослідження зумовлена зростаючою потребою в інструментах для підвищення продуктивності ІТ-команд. Метод Pomodoro пропонує інноваційний підхід до вирішення цієї проблеми, але більшість існуючих рішень зосереджені на індивідуальному використанні, що обмежує їх застосування в командному середовищі. Розробка веб-додатку, який адаптує цей метод для командного використання, є актуальною задачею, оскільки вона дозволяє не лише покращити управління часом, але й сприяє поліпшенню комунікації та взаємодії між членами команди. Це, в свою чергу, може призвести до підвищення загальної продуктивності та ефективності роботи. Впровадження такого рішення може стати значним кроком у напрямку підвищення ефективності управління часом у сучасному бізнес-середовищі.

Метою дослідження є розробка проекту веб-додатку, що реалізує метод Pomodoro для покращення управління часом у командах, підвищення продуктивності та оптимізації робочих процесів. Особлива увага приділяється адаптації методу для командної роботи та інтеграції з існуючими процесами розробки програмного забезпечення.

Завданням дослідження для досягнення встановленої мети є провести комплексний аналіз підходів до процесу розробки ІТ-проектів та методів підвищення продуктивності, розробити проект веб-додатку для ефективного управління часом та інтеграції методу Pomodoro в командні процеси, розробити детальні моделі взаємодії користувачів, сформулювати функціональні та технічні вимоги до веб-додатку, створити макети основних сторінок додатку та карти навігації, розробити комплексний план проекту, включаючи аналіз стейкхолдерів, декомпозицію робіт, реєстр ризиків та бізнес-модель продукту.

Об'єктом дослідження є процеси управління часом в ІТ-командах та методологія Pomodoro як інструмент підвищення продуктивності.

Предметом дослідження є методи та інструменти розробки проєкту веб-додатку для ефективного управління часом на основі методу Pomodoro, а також особливості його впровадження в командну роботу.

Методи дослідження, які були використані в даній роботі включають теоретичний аналіз наукових джерел, порівняльний аналіз, емпіричні дослідження, використання візуальних інструментів для структурування бізнес-моделі продукту, а також ідентифікація та оцінка потенційних ризиків, пов'язаних з розробкою та впровадженням продукту.

Наукова новизна дослідження полягає у теоретичному обґрунтуванні розробки комплексного підходу до інтеграції методу Pomodoro в командні процеси розробки програмного забезпечення, що дозволило підвищити ефективність командних зустрічей та знизити рівень стресу серед працівників. Метод Pomodoro отримав подальший розвиток завдяки його адаптації до специфіки командної роботи.

Практичне значення роботи визначається створенням проєкту веб-додатку, який дозволяє ефективно організовувати командні зустрічі, покращити процес планування спринтів та оптимізувати розподіл робочого часу. Впровадження розробленого рішення забезпечує підвищення концентрації уваги, зниження рівня стресу та покращення якості виконаних завдань. Особлива практична цінність для бізнесу полягає у підвищенні ефективності командної роботи та покращенні процесів планування й оцінки задач, що набуває особливої актуальності в умовах зростаючої популярності віддаленої роботи та необхідності ефективної організації командних процесів у цифровому середовищі.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ПРОБЛЕМИ РОЗРОБКИ ІТ-ПРОЄКТІВ ТА УПРАВЛІННЯ ЧАСОМ

1.1. Основні підходи до процесу розробки ІТ-проектів

Розробка ІТ-проектів є складним та багатогранним процесом, що включає в себе комплекс взаємопов'язаних заходів, спрямованих на створення унікального програмного продукту в умовах обмежених ресурсів та часу[19]. У сучасному світі успішна реалізація ІТ-проектів стає критично важливим фактором розвитку бізнесу та організацій.

Основною особливістю ІТ-проектів є їх інноваційний характер, що вимагає постійного оновлення знань та навичок команди розробників. За даними досліджень, близько 71% ІТ-проектів стикаються з викликами через постійні зміни вимог та технологій[20]. Це підкреслює необхідність гнучкого підходу до управління такими проектами.

Ефективна розробка ІТ-проектів базується на чіткому визначенні цілей, плануванні ресурсів та встановленні реалістичних термінів виконання. Згідно з дослідженнями Zipdo, лише 16% ІТ-проектів завершуються вчасно та в межах бюджету[21]. Це свідчить про важливість правильного розуміння сутності процесу розробки та застосування відповідних методологій управління.

Ключовими складовими успішної розробки ІТ-проектів є[22]:

- чітке визначення вимог;
- ефективна комунікація між усіма учасниками проекту;
- контроль якості;
- управління ризиками та постійний моніторинг прогресу.

Особлива увага приділяється документуванню процесів та створенню детальної технічної документації.

Важливим аспектом розробки ІТ-проектів є забезпечення інформаційної безпеки та захисту даних. Дослідження Cybersecurity Ventures показують, що витрати на кібербезпеку в ІТ-проектах щорічно зростають на 15%[23]. Це вимагає інтеграції відповідних заходів безпеки на всіх етапах розробки.

Особливу роль у розробці ІТ-проектів відіграє якість коду та тестування. Згідно з дослідженнями IBM, виправлення помилок на етапі експлуатації коштує в 4-5 разів дорожче, ніж їх усунення на етапі розробки[24]. Тому важливо забезпечити належний рівень тестування та контролю якості протягом усього життєвого циклу проекту.

Основою сучасної технології розробки є гнучкі методології, які дозволяють ефективно управляти проектами в умовах постійних змін. Найбільш поширеними є методології Agile, зокрема Scrum та Kanban, які забезпечують ітеративний підхід до розробки та тісну співпрацю між членами команди та замовником[25]. Порівняємо дані методології між собою та з менш поширеними Waterfall і Extreme Programming, результат порівняння подамо у вигляді таблиці 1.1:

Таблиця 1.1

Порівняння найпоширеніших методологій управління проектами

Критерій	Scrum	Waterfall	Kanban	XP (Extreme Programming)
Гнучкість	Висока гнучкість з фіксованими спринтами	Низька гнучкість, складно вносити зміни	Висока гнучкість, безперервний потік	Висока гнучкість з фокусом на технічні практики
Тривалість ітерацій	Фіксовані спринти (2-4 тижні)	Немає ітерацій, послідовні фази	Безперервний потік	Дуже короткі ітерації (1-2 тижні)
Ролі	Чітко визначені (Scrum Master, Product Owner, Team)	Традиційна ієрархія	Немає чітко визначених ролей	Командна структура з акцентом на парне програмування
Планування	Детальне планування спринту	Детальне попереднє планування всього проекту	Планування за потребою	Планування ітерацій та релізів
Зміни вимог	Можливі між спринтами	Складно внести зміни	Можливі в будь-який час	Можливі в будь-який час

Продовження таблиці 1.1

Метрики	Burndown charts, Velocity	Gantt charts, milestone tracking	Lead time, Cycle time	Velocity, тестове покриття
Артефакти	Product/Sprint Backlog, Burndown Chart	Детальна документація	Kanban board	User stories, тести
Зустрічі	Daily Scrum, Sprint Planning, Review, Retrospective	Формальні зустрічі за планом	За потребою	Daily stand-ups, планування ітерацій
Продуктивність	+25% порівняно з традиційними методами	Базова	+15% порівняно з традиційними методами	+20% порівняно з традиційними методами
Масштабованість	Добре масштабується	Добре масштабується	Обмежена масштабованість	Складно масштабується

Порівняльний аналіз методологій розробки програмного забезпечення демонструє суттєві переваги Scrum над іншими підходами. На відміну від послідовного Waterfall, він пропонує гнучке коригування вимог та пріоритетів. Порівняно з Kanban, забезпечує кращу структуру та передбачуваність завдяки фіксованим спринтам та чітким ролям, що особливо важливо для складних проєктів. А на відміну від Extreme Programming (XP), який концентрується на технічних аспектах, Scrum надає збалансовану структуру управління, адаптовану під різні технічні підходи, зберігаючи ефективне керування проєктом та командою.

Таким чином, Scrum є оптимальним вибором для сучасних проєктів розробки програмного забезпечення, особливо в умовах високої невизначеності та динамічних змін. Він забезпечує необхідний баланс між структурованістю та гнучкістю, дозволяючи командам ефективно працювати над створенням якісного продукту, який відповідає потребам користувачів та бізнесу.

Формування команди є критично важливим першим кроком у розробці ІТ-проєкту, оскільки якісно підібрана команда не лише дозволяє точніше оцінити можливості реалізації та спланувати ресурси, але й забезпечує ефективний розподіл бюджету, узгодження технічного бачення (включаючи вибір технологій

та архітектурних рішень), формування командної культури та процесів комунікації, а також краще розуміння вимог замовника та взаємодію зі стейкхолдерами, що в результаті значно підвищує шанси на успішну реалізацію проєкту та відповідність кінцевого продукту очікуванням замовника. Дослідження показують, що проєкти з попередньо сформованими командами значно рідше виходять за рамки запланованого бюджету[9]. Також статистика показує, що проєкти з чітко сформованими командами на старті мають значно вищу ймовірність успішного завершення, нижчий ризик перевищення бюджету та менше затримок у термінах виконання[26]. Це підтверджує, що інвестиції часу та ресурсів у формування команди на початковому етапі є критично важливими для успіху всього проєкту.

Також дуже важливим складовим успіху проєкту є чітко визначений процес управління. У сучасному управлінні проєктами просте збільшення ресурсів, таких як час, фінанси та людські ресурси, не є гарантією успіху проєкту. Ключовими чинниками успіху є залучення кваліфікованих фахівців та чітке визначення основних елементів проєкту[10].

Технологія розробки ІТ-проєктів є комплексним процесом, який включає в себе набір методів, інструментів та практик, спрямованих на створення якісного програмного забезпечення. Цей процес охоплює всі етапи життєвого циклу проєкту – від початкового планування до впровадження та подальшої підтримки продукту.

Перший етап – аналіз вимог, який є фундаментальним для всього проєкту. На цьому етапі відбувається детальне вивчення потреб замовника, визначення цілей проєкту та формування чіткого розуміння кінцевого результату. Команда проєкту збирає та аналізує всі вимоги, створює необхідну документацію та оцінює необхідні ресурси і терміни виконання. Вимоги зазвичай поділяються на кілька категорій: функціональні (що система повинна робити), нефункціональні (як система повинна працювати) та технічні (технологічні обмеження та вимоги)[27].

Документування вимог відбувається у вигляді специфікацій, які можуть включати текстові описи, діаграми, прототипи інтерфейсу користувача та інші артефакти. Важливо забезпечити простежуваність вимог – можливість відстежити зв'язок між вимогами та їх реалізацією. Результатом аналізу вимог має бути чітке розуміння того, що потрібно розробити, узгоджене між всіма зацікавленими сторонами.

Після завершення аналізу починається етап планування. На цьому етапі розробляється технічне завдання, обираються технології та інструменти для реалізації проєкту. Також визначаються основні віхи проєкту та встановлюються терміни їх досягнення. Оскільки ми прийшли до висновку, що Scrum є оптимальним вибором для вирішення досліджуваної проблеми (див. табл. 1.1), розглянемо використання саме цієї методології під час розробки.

Першим кроком є формування загального бачення продукту та створення Product Backlog – впорядкованого списку всіх необхідних функцій та вимог. Product Owner разом із командою визначає пріоритети та оцінює трудомісткість завдань в рамках мітингів Backlog Refinement[7].

Перед кожним спринтом проводиться Sprint Planning, де команда обирає завдання з Product Backlog для реалізації в поточному спринті. Тривалість спринту зазвичай становить 2-4 тижні та оптимальна кількість завдань у Sprint Backlog повинна забезпечувати 85-90% завантаженості команди. Під час спринту команда щодня проводить Daily Scrum, 15-хвилинні зустрічі для синхронізації роботи. Розробники працюють над завданнями згідно з принципами самоорганізації. Scrum Master забезпечує дотримання процесів та усуває перешкоди[28].

Наприкінці спринту проводиться Sprint Review для демонстрації виконаної роботи стейкхолдерам. Після цього команда проводить Sprint Retrospective для аналізу процесів та визначення шляхів покращення. За даними AgileAlliance, регулярне проведення ретроспектив покращує продуктивність команди на 15-20%[7].

Після досягнення достатньої функціональності відбувається випуск продукту. Процес розробки може продовжуватись ітеративно для додавання нових функцій та вдосконалень. Згідно з дослідженням PMI, гнучкий підхід до релізів зменшує час виходу на ринок на 37%[3].

Етап проєктування передбачає створення детальної архітектури системи, розробку прототипів користувацького інтерфейсу та проєктування бази даних. На цьому етапі визначаються всі компоненти системи та їх взаємодія між собою. Важливо приділити особливу увагу проєктуванню, оскільки помилки на цьому етапі можуть призвести до серйозних проблем у майбутньому.

Особлива увага приділяється проєктуванню інтерфейсів, як користувацьких, так і програмних. Розробляються API для взаємодії між компонентами системи, проєктується користувацький інтерфейс, який повинен бути зручним та інтуїтивно зрозумілим.

Етап розробки програмного забезпечення – це ключовий етап життєвого циклу проєкту, під час якого відбувається безпосереднє створення програмного продукту на основі проєктної документації. На цьому етапі програмісти пишуть код, використовуючи обрані мови програмування та технології, створюють компоненти системи згідно з архітектурними рішеннями.

Важливою частиною етапу розробки є постійне тестування створюваного коду та виправлення знайдених помилок. Дослідження, проведене IBM Systems Sciences Institute, показує, що вартість виправлення помилки зростає експоненціально на кожному наступному етапі розробки[17]. Тому розробники проводять модульне тестування, перевіряють якість коду за допомогою статичного аналізу, дотримуються стандартів кодування та проводять код-рев'ю. Як зазначає Бек К., тест-кейси повинні охоплювати як позитивні, так і негативні сценарії використання програми[6]. За даними World Quality Report, автоматизоване тестування становить близько 45% всіх тестових активностей в сучасних IT-проєктах[4].

Безпека даних також є критичним аспектом тестування. Згідно з дослідженням Ponemon Institute, середня вартість витоку даних у 2020 році

склала \$4.27 мільйона[5]. Належне тестування безпеки може запобігти таким інцидентам та зберегти репутацію компанії.

Документування результатів тестування та звітність є завершальним етапом. За даними ISO/IEC 29119, правильне документування підвищує ефективність процесу тестування на 25%[8].

Етап впровадження передбачає розгортання системи в робочому середовищі, міграцію даних та навчання користувачів. Цей етап включає налаштування серверів, баз даних та мережевої інфраструктури, встановлення необхідного програмного забезпечення та конфігурацію системних параметрів.

На рисунку 1.1 схематично зображено процеси безперервної інтеграції та доставки:

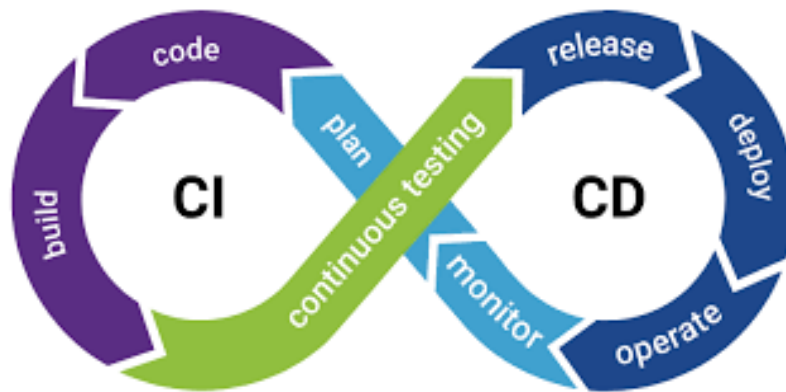


Рисунок 1.1. Схематичне зображення процесів CI/CD

Сучасні практики впровадження часто використовують принципи безперервної інтеграції та доставки (CI/CD), що дозволяє автоматизувати процес розгортання та зменшити ризики людських помилок. На цьому етапі також проводиться фінальне тестування в продуктивному середовищі, перевірка безпеки та продуктивності системи, а також розробляються плани відновлення у разі збоїв. Успішне впровадження вимагає ретельного планування, координації між різними командами (розробники, системні адміністратори, служба підтримки) та наявності чіткого плану відкату змін у разі виникнення критичних проблем[29].

Заключним, але не менш важливим етапом є підтримка та обслуговування. Після введення системи в експлуатацію необхідно забезпечити її стабільну

роботу, виправляти виявлені помилки, оновлювати функціонал та оптимізувати продуктивність. Також важливо забезпечити технічну підтримку користувачів системи.

Успішна реалізація всіх етапів розробки вимагає ефективної комунікації між членами команди, чіткого розуміння цілей проєкту та використання відповідних інструментів управління розробкою. Важливо також постійно контролювати якість роботи та дотримуватися встановлених термінів виконання завдань.

1.2. Теоретичні основи та аналіз методів управління часом

Управління часом – це набір методів і інструментів, що допомагають організувати роботу та особисте життя для максимально ефективного використання часу, виконуючи завдання швидше і віддаючи пріоритет важливим справам. Основна мета управління часом полягає в підвищенні продуктивності через правильне планування та пріоритети, що дозволяє виконувати більше завдань за менший час. Чітка структура роботи допомагає зосередитися на завданнях, зменшуючи відволікання, і сприяє впевненості у виконанні планів. Управління часом корисне в навчанні, роботі, фрілансі та особистому житті, дозволяючи краще організувати підготовку до іспитів, виконання завдань, баланс між проєктами та особистими захопленнями. Існує досить багато різних методів управління часом і вибір залежить від потреб та особливостей конкретної людини[11]. Деякі методи управління часом виходять за межі конкретних сфер та є популярними у широких колах користувачів.

До них відносяться:

- метод Pomodoro, який базується на принципі роботи в інтервалах (наприклад, 25 хвилин роботи та 5 хвилин відпочинку), що сприяє підвищенню продуктивності та управлінню концентрацією;

- метод GTD (Getting Things Done), що розвивається навколо ідеї систематичного оброблення завдань та їх впорядкування для ефективного виконання;
- метод Матриці Ейзенгауера (Eisenhower Matrix), базується на класифікації завдань залежно від їх важливості та терміновості.

Важливо зазначити, що не існує універсального методу управління часом, який підходив б усім. Найкращий метод залежить від індивідуальних особливостей людини, її цілей та сфери діяльності. Під час розробки веб-додатку для ефективного управління часом важливо врахувати різні методи управління часом та їх особливості. Це дозволить створити додаток, який буде максимально корисним для широкого кола користувачів.

Техніка Pomodoro – це метод управління часом, заснований на 25-хвилинних ділянках зосередженої роботи, розбитих на п'ятихвилинні перерви. Більш тривалі перерви, як правило, від 15 до 30 хвилин, робляться після чотирьох послідовних інтервалів роботи[12]. На рисунку 1.2 зображено візуалізацію принципу його роботи:

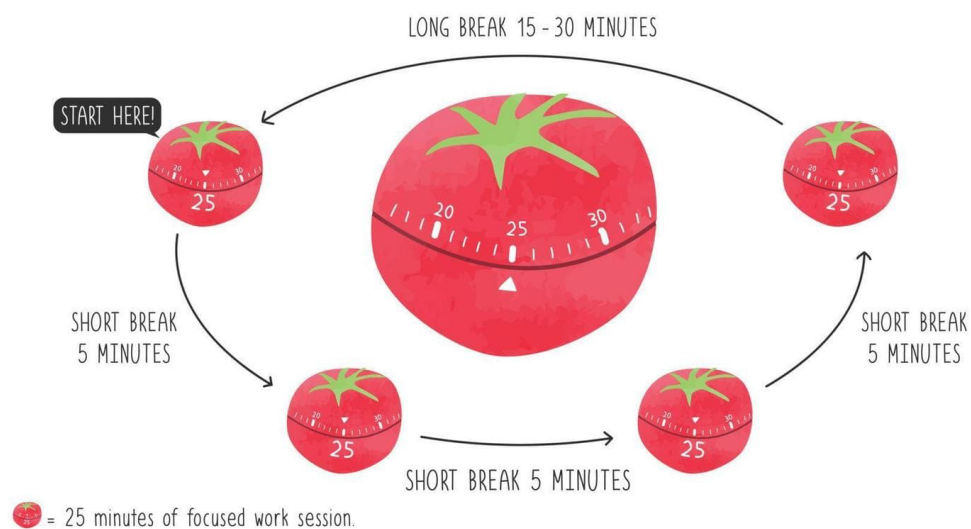


Рисунок 1.2. Візуалізація принципу роботи методу Pomodoro

Спочатку людина повинна обрати завдання, яке планує виконати, після чого встановлюється таймер на 25 хвилин (один інтервал або "помідор"). Протягом цього часу необхідно сконцентровано працювати над завданням до моменту спрацювання таймера. Після закінчення "помідора" слід зробити

п'ятихвилинну перерву для розслаблення мозку. Цей цикл фокусування та відпочинку повторюється після кожного завершеного "помідора", а після чотирьох таких інтервалів рекомендується взяти довшу перерву тривалістю 15-30 хвилин[13].

Pomodoro довів свою ефективність у багатьох галузях та отримав визнання від користувачів як інструмент для підвищення продуктивності та зменшення стресу[2].

Метод GTD, або Getting Things Done, базується на переконанні, що чим більше інформації відстежується подумки, тим менша продуктивність та зосередженість[14]. Замість того, щоб покладатися на мозок, методологія GTD заохочує зберігати всю свою робочу інформацію у зовнішньому організованому джерелі правди. Таким чином завжди відома відповідь на питання «Що робити далі?» не турбуючись про те, що робота впаде крізь щілини. Візуалізацію принципу роботи методу Getting Things Done зображено на рисунку 1.3:



Рисунок 1.3. Візуалізація принципу роботи методу GTD

Спершу потрібно зібрати всі завдання, думки та ідеї в одноманітний системний список, після чого регулярно переглядати та розподіляти їх за конкретними категоріями (виконати, відкласти, викинути, делегувати чи розпочати дію в майбутньому). Далі рекомендується створити ієрархію завдань, визначаючи їх важливість та терміновість, використовуючи контексти та

проекти для забезпечення структурованості. На етапі виконання мають бути призначені чіткі дії для кожного завдання та визначені пріоритети, що дозволяє зберігати високий рівень енергії шляхом фокусування на одному завданні одночасно. Завершальним кроком є періодичне рецензування завдань, їх актуалізація та внесення необхідних змін у систему[15].

Метод GTD виявляється досить ефективним для особистого та професійного управління завданнями. Це простий у застосуванні, гнучкий метод, який зменшує навантаження на мозок і допомагає виконувати роботу з більшою віддачою. Його комплексний підхід та фокус на системність роблять його популярним інструментом серед тих, хто шукає структурований метод управління своїм часом та завданнями.

Складання списку справ є першим кроком до виконання роботи. Але як визначити, за що братися в першу чергу якщо не вистачає часу, щоб зробити все за один день? Завдяки ефективному визначенню пріоритетів можна підвищити продуктивність і гарантувати, що найтерміновіші завдання привернуть увагу негайно.

Метод Eisenhower Matrix, створений 34-м президентом США Дуайтом Ейзенхауером, є інструментом для пріоритезації завдань та керування часом. Цей метод використовує матрицю для класифікації завдань за двома ключовими критеріями: важливість та терміновість[16]. Це інструмент, який допомагає розрізнити термінові та важливі завдання, щоб налагодити ефективний робочий процес.

Матриця складається з чотирьох квадрантів, кожен з яких представляє різний пріоритет для завдань[1]. До першого квадранту "Do" (Важливо і терміново) входять найважливіші та найтривожніші завдання, які мають явні наслідки та впливають на довгострокові цілі - їх слід виконати найперше. Другий квадрант "Plan" або "Schedule" (Важливо, але не терміново) містить завдання, які хоч і не мають термінового характеру, але є важливими, тому їх виконують одразу після завершення завдань першого квадранту. У третьому квадранті "Delegate" (Терміново, але не важливо) знаходяться термінові, але не важливі

особисто для людини завдання, які через відсутність особистої прихильності та потреби в специфічних навичках варто делегувати іншим членам команди. Четвертий квадрант "Eliminate" (Не терміново і не важливо) включає завдання, які не є ані важливими, ані терміновими - вони можуть заважати досягненню цілей, тому їх варто або відкласти на майбутнє, або взагалі відмовитися від їх виконання.

Схематичне зображення квадрантів представлено на рисунку 1.4:



Рисунок 1.4. Візуалізація квадрантів методу Eisenhower Matrix

Принцип роботи методу Eisenhower Matrix ґрунтується на чотирьох послідовних етапах[18]. Спочатку необхідно здійснити збір завдань, записавши всі свої завдання та обов'язки. Далі потрібно провести класифікацію за важливістю та терміновістю, розподіливши всі завдання відповідно до їх характеристик у чотири квадранти матриці. Наступним кроком є безпосереднє виконання пріоритетних завдань, починаючи з першого квадранта та поступово переходячи до наступних у порядку пріоритету. Завершальним етапом є періодична перевірка та корекція, що передбачає регулярний перегляд та оновлення матриці відповідно до нових завдань та змін у пріоритетах.

Метод Eisenhower Matrix виявляється потужним інструментом для пріоритезації завдань та ефективного управління часом. Його структура дозволяє швидко визначати, які завдання потрібно виконати першими, а які можна відкласти або делегувати.

Виконаємо порівняння розглянутих методів та представимо результат у вигляді таблиці 1.2:

Таблиця 1.2

Порівняльна таблиця методів управління часом

Метод	Переваги	Недоліки	Використання
Pomodoro	- збільшення продуктивності завдяки фокусу на коротких інтервалах; - концентрація сприяє зменшенню відволікань; - регулярні перерви допомагають зберегти енергію та підтримувати високий рівень концентрації; - легка реалізація без складних технологій або інструментів.	Може обмежувати деяких людей часовими рамками	Коли потрібно швидко покращити фокус та продуктивність завдяки коротким та інтенсивним інтервалам.
GTD	- комплексний підхід до управління завданнями; - зниження рівня стресу та відчуття перевантаження; - підвищення продуктивності завдяки чіткому визначенню завдань.	Складність впровадження, вимагає систематичного підходу та витрат часу на організацію	Коли потрібен комплексний підхід та можливість інвестувати час в організацію завдань.
Eisenhower Matrix	- ясність пріоритетів, чіткий механізм для визначення та розподілу пріоритетів; - ефективне використання часу, орієнтація на важливі та термінові завдання; - допомагає зменшити стрес та відчуття перевантаження.	Не враховує можливі зміни, матриця може стати застарілою	Коли важливо чітко розподілити пріоритети та термінові завдання.

Отже, метод Pomodoro є чудовим вибором для реалізації веб-додатку для управління часом завдяки своїй простоті, ефективності та універсальності.

Також даний метод є особливо корисним для командного використання, оскільки він сприяє підвищенню продуктивності та ефективності спільної роботи. Завдяки чіткій структурі, що включає 25-хвилинні інтервали зосередженої роботи та короткі перерви, команда може організувати свій робочий процес, зберігаючи при цьому високий рівень концентрації. Це дозволяє кожному учаснику зосередитися на виконанні конкретних завдань, що підвищує загальну продуктивність команди. Крім того, під час перерв команди можуть обговорювати прогрес, ділитися ідеями та вирішувати проблеми, що сприяє покращенню комунікації та співпраці. Веб-додаток, заснований на методі Pomodoro, може включати не тільки функціонал налаштування інтервалів, відстеження прогресу та нагадування, а і функції для спільного планування завдань, відстеження часу та обміну результатами, що робить його ідеальним інструментом для команд, які прагнуть оптимізувати свою роботу.

1.3. Світовий досвід вирішення проблем ефективного управління часом

Проблема ефективного управління часом стала особливо актуальною з початком цифрової епохи. Неправильне використання часу є помилкою найвищого ступеня, тому неправильне управління часом потребує справжньої відданості з боку окремих осіб та організацій[36].

Дослідження, проведене Microsoft Research, показало, що середній час концентрації уваги людини знизився з 12 секунд у 2000 році до 8 секунд у 2023 році[30]. Це створило потребу в розробці нових методів та інструментів для підвищення продуктивності, а метод Pomodoro, розроблений Франческо Чірілло, став одним із найбільш науково обґрунтованих підходів до організації робочого процесу.

Дослідження, проведені в різних країнах, демонструють високу ефективність методу Pomodoro в різноманітних сферах діяльності. Наприклад, у США метод активно використовується в академічному середовищі. Згідно з

дослідженням, проведеним у Стенфордському університеті[6], використання методу Pomodoro призводить до:

- підвищення концентрації уваги на 35%;
- зниження рівня стресу на 28%;
- покращення якості виконаних завдань на 23%.

Проведені науковцями університету Reed College дослідження також демонструють цікаві цифри. В рамках експерименту протягом тижня фіксувався час, який студенти витрачають на навчання та загальний час відволікання від завдань на різні зовнішні чинники. Результат показав, що з однаковим об'ємом виконаної роботи середній час навчання скоротився на 8.95%, час на відволікання зменшився аж на 46% і викладачі університету відмітили покращення у якості виконаних завдань[32].

Також у світовій спільноті деякі відомі люди діляться досвідом не тільки теоретичних досліджень, а і результатами практичного використання техніки Pomodoro. Розглянемо декілька таких згадок.

Shama Hyder, яка займає посаду CEO та є засновницею Zen Media у своєму блозі ділиться досвідом деяких членів її команди, що слідує техніці Pomodoro. Співробітники ставлять емодзі «помідор» у статусі Slack та інші колеги знають, що така людина бере участь у інтенсивному 25-хвилинному сеансі Pomodoro, який не слід переривати, окрім крайньої необхідності. Емодзі помідора також сигналізує, що за звичайних обставин не варто очікувати відповіді на повідомлення до зміни статусу, сигналізуючи про те, що сеанс Pomodoro закінчився[33].

Dr. Kat Cohen, засновниця та SEO провідної незалежною консалтинговою компанії в галузі освіти IvyWise, яка надає консультації щодо вступу до коледжу в Нью-Йорку та в усьому світі, у інтерв'ю Authority Magazine поділилася досвідом особистого використання та зазначила, що багато її співробітників практикують цю техніку та погоджуються, що це чудовий спосіб усунути відволікання та справді відточити проєкт, який є пріоритетним. Також вона ділиться позитивним досвідом використання Pomodoro в галузі освіти. Kat Cohen

зазначає, що управління часом може бути складним завданням, особливо для мілленіалів, які вирости з багатьма формами стимуляції, що змагаються за їхню увагу, але це критично важлива навичка, над розвитком якої вона заохочує професіоналів будь-якого віку. В таких випадках метод Pomodoro досить ефективно протидіє відволіканням й допомагає зосередитися на тому, що потрібно зробити[34].

Засновниця та CEO фірми Tribe Builder Media, яка займається медіа, зв'язками з громадськістю, контент-маркетингом і розвитком бізнесу, Danielle Sabrina в своїй статті також ділиться позитивним досвідом використання Pomodoro, вказує на особисте відчуття підвищення рівня енергії та спостереження, що запровадження даного методу зробило команду загалом більш позитивною та орієнтованою на рішення[35].

На міжнародному ринку представлено значну кількість програмних продуктів, що базуються на методі Pomodoro. Розглянемо найбільш успішні реалізації та їх особливості.

Focus@Will являє собою комплексне рішення, розроблене американськими спеціалістами, поєднує класичний метод Pomodoro з нейронауковими дослідженнями. На рисунку 1.5 зображено інтерфейс та особливості додатку:

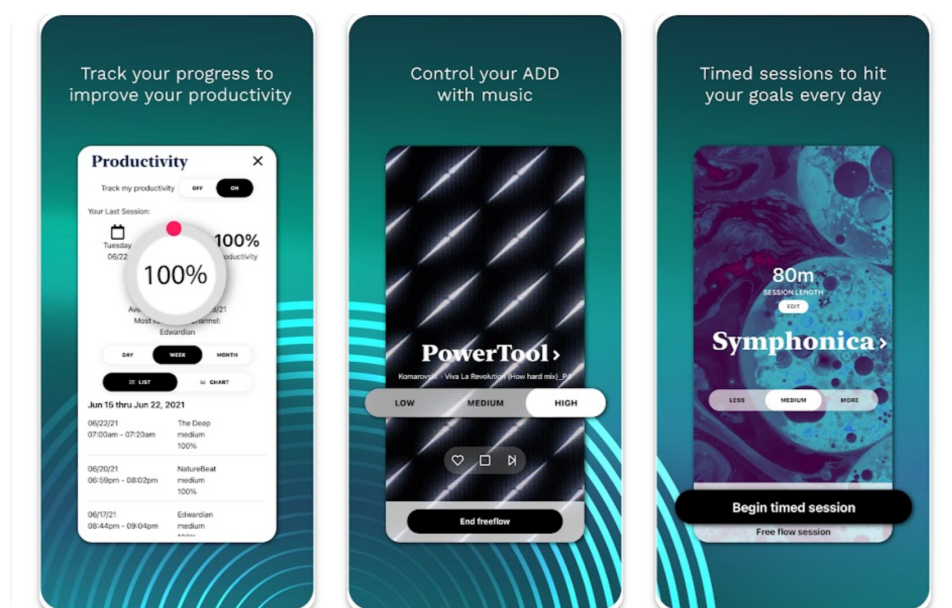


Рисунок 1.5. Інтерфейс та особливості додатку Focus@Will

Програма використовує спеціально підібрану музику, яка сприяє концентрації уваги. За даними внутрішніх досліджень компанії, користувачі відзначають підвищення продуктивності в середньому на 200-400%. Особливістю програми є адаптивний алгоритм, який враховує індивідуальні особливості користувача та оптимізує тривалість робочих сесій[31].

Інноваційний підхід до реалізації методу Pomodoro представлений тайванською компанією Seekrtech в додатку Forest App. Розглянемо інтерфейс додатку, який представлений на рисунку 1.6:



Рисунок 1.6. Інтерфейс додатку Forest App

Програма використовує гейміфікацію для підвищення мотивації користувачів. Під час роботи користувач "виросує" віртуальне дерево, яке гине при передчасному припиненні робочої сесії. Більше того, компанія співпрацює з організаціями з озеленення планети, і за певну кількість успішних сесій висаджуються реальні дерева. Такий підхід створює додаткову соціальну мотивацію для користувачів.

Один з найпопулярніших менеджерів завдань Todoist також інтегрував функціонал Pomodoro у свою систему. На рисунку 1.7 наведено інтерфейс додатку:

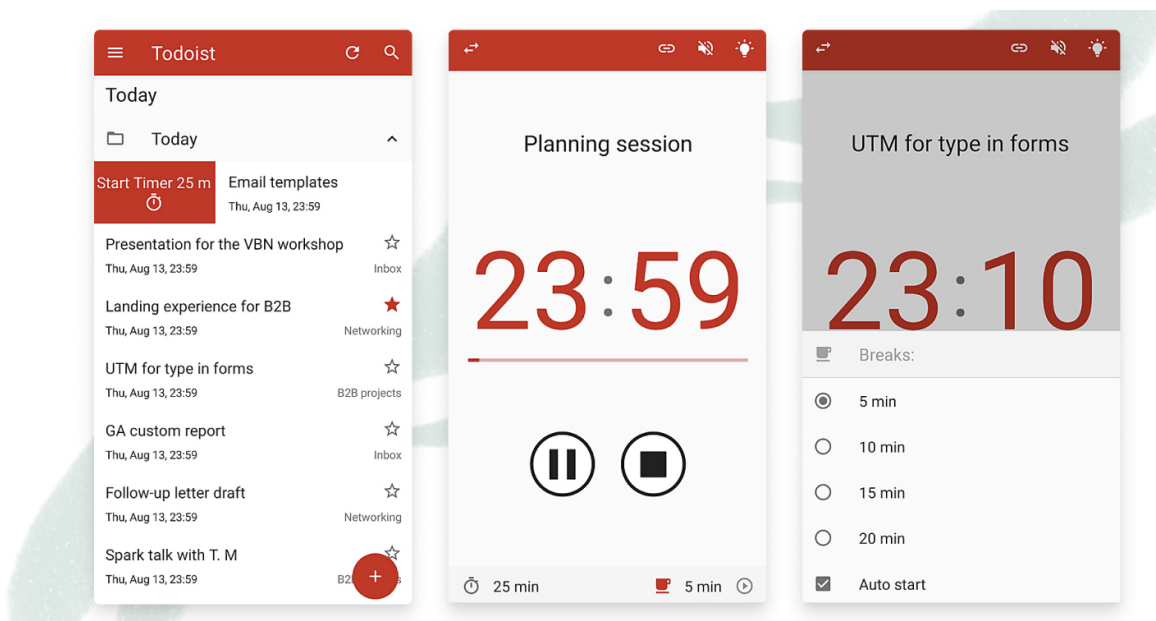


Рисунок 1.7. Інтерфейс додатку Todoist

Особливістю цього рішення є глибока інтеграція з іншими інструментами продуктивності та можливість командної роботи. Програма надає детальну аналітику використання часу та дозволяє оптимізувати робочий процес на основі зібраних даних.

На основі аналізу представлених програмних продуктів можна зробити висновок, що сучасні реалізації методу Pomodoro значно еволюціонували від простого таймера до комплексних систем управління часом. Можемо виділити наступні ключові особливості успішних рішень:

- інтеграція наукових досліджень;
- використання інноваційних мотиваційних механізмів;
- розширені можливості для командної роботи.

Важливим аспектом є адаптивність систем, які враховують індивідуальні особливості користувачів та дозволяють налаштовувати параметри роботи під конкретні потреби. Такий комплексний підхід, що поєднує технологічні інновації з психологічними та соціальними аспектами мотивації, забезпечує високу ефективність цих інструментів та їх популярність серед користувачів.

Як ми можемо помітити, розглянуті програмні продукти більш орієнтовані на особисте використання методу Pomodoro, лише Todoist має деякі інструменти для командного використання методу, що свідчить про незайнятість даної ніші

на ринку додатків та наявність шансу на розробку дійсно інноваційного продукту. Це може стати ще одним фактором, який позитивно впливає на успіх розроблюваного проєкту.

Отже, ми надали комплексний огляд міжнародного досвіду у вирішенні проблем управління часом та розробці відповідних технологічних рішень. Проведене дослідження міжнародного досвіду дозволяє стверджувати, що розробка веб-додатку для ефективного управління часом на основі методу Pomodoro є актуальним та перспективним напрямком. Існуючі рішення мають значний потенціал для вдосконалення, а впровадження сучасних технологій та переміщення фокусу на командну роботу дозволить створити більш ефективний та користувацько-орієнтований продукт. Отримані результати дослідження створюють надійну теоретичну та практичну базу для подальшої розробки проєкту веб-додатку, який враховуватиме виявлені недоліки існуючих рішень та впроваджуватиме інноваційні підходи до управління часом.

Висновки до першого розділу

У першому розділі було досліджено основні підходи до процесу розробки ІТ-проєктів. Розглянуто особливості та ключові складові успішної розробки, проаналізовано найпоширеніші методології управління проєктами (Scrum, Waterfall, Kanban, XP) та проведено їх порівняльний аналіз. Визначено, що Scrum є оптимальним вибором для сучасних проєктів розробки програмного забезпечення завдяки балансу між структурованістю та гнучкістю.

Розглянуто теоретичні основи та проведено аналіз методів управління часом. Детально досліджено три основні методи: Pomodoro, GTD (Getting Things Done) та Eisenhower Matrix. Проведено їх порівняльний аналіз та визначено, що метод Pomodoro є найбільш оптимальним для реалізації веб-додатку завдяки своїй простоті, ефективності та універсальності, особливо для командного використання.

Проаналізовано світовий досвід вирішення проблем ефективного управління часом. Розглянуто результати наукових досліджень, які підтверджують ефективність методу Pomodoro, вивчено досвід успішних компаній та особистостей у використанні цього методу. Проаналізовано існуючі програмні рішення на ринку та виявлено потенційну нішу для розробки інноваційного продукту з фокусом на командну роботу.

Проведене дослідження створило надійне теоретичне підґрунтя для подальшої розробки веб-додатку, визначило основні вимоги та напрямки для його реалізації, а також підтвердило актуальність та перспективність обраного напрямку розробки.

РОЗДІЛ 2. ОПИС ПРОДУКТУ ТА ВИМОГ

2.1. Вибір цільової аудиторії та розробка моделей користувачів

Сегментування ринку та вибір цільової аудиторії є ключовими елементами маркетингової стратегії, де використовуються різні моделі сегментування, включаючи психографічну (фокус на способі життя, цінностях та інтересах споживачів) та поведінкову (базується на характері споживання продукту), а вибір цільової аудиторії здійснюється на основі оцінки привабливості кожного сегмента та можливості компанії для ефективної диференціації продукту в них.

Наш проєкт буде орієнтований на команду, яка розробляє програмне забезпечення організовуючи свої процеси за методологією Scrum, основою якої є ітерації, які називаються спринтами. Вона забезпечує гнучке та адаптивне середовище для розробки, де команда працює над чітко визначеними завданнями з постійним зворотним зв'язком і можливістю внесення коректив.

Цільова команда постійно вносить зміни у внутрішні процеси з метою їх покращення і нашою задачею буде створити веб-додаток, який допоможе з цим. Команди, що складаються з близько 9 людей, що є досить великим (але все ще допустимим) числом для методології Scrum, часто стикаються з різними проблемами, серед яких найбільш впливовими являються затяжне планування та втрата продуктивності під час довгих мітингів. Нашою задачею буде спроектувати веб-додаток, який зменшить вплив цих чинників.

Методології створення персон та Jobs-to-be-Done (JTBD) є ключовими інструментами для розуміння користувачів та їхніх потреб: персони як уявні представники допомагають командам зосередитися на реальних потребах клієнтів та покращити комунікацію, тоді як JTBD фокусується на завданнях користувачів та контексті використання продукту, що дозволяє створювати інноваційні рішення, які перевершують очікування та підвищують конкурентоспроможність на ринку.

Персона Senior Full-Stack розробника Олександра

Олександр - Senior Full-Stack розробник з 7-річним досвідом роботи в продуктових IT компаніях. Він працює в команді з 8 людей. Олександр відповідальний за технічні рішення, контроль якості коду та менторинг молодших розробників. Він одружений, має маленьку дитину, тому особливо цінує баланс між роботою та особистим життям. У вільний час займається спортом та читає технічну літературу для підтримки професійного розвитку.

В процесі виконання своїх обов'язків досить досвідченого спеціаліста Олександр стикається з проблемою ефективності управління часом та фокусуванням на роботі. Також він помічає, що молодші колеги часто губляться в потоці інформації під час довгих технічних обговорень, а відсутність чітких часових рамок призводить до того, що важливі деталі губляться або забуваються. Крім того, через часті незаплановані звернення по допомогу від команди, йому складно підтримувати власний фокус на складних технічних задачах. Внаслідок такої завантаженості та постійного прагнення розвиватися Олександр не помічає, що не дотримується гігієни праці, і починає помічати проблеми з очима.

Опис персони Олександра наведено на рисунку 2.1:



Олександр Сухотруб
Senior Full-Stack Developer

Вік: 32
Освіта: Київський столичний університет імені Бориса Грінченка
Рідне місто: Чоп
Сім'я: Одружений, має маленьку дитину

“Успіх — це не випадковість, а результат наполегливої праці та віри в себе.”

Цілі	Розчарування
○ Підвищити ефективність командних зустрічей ○ Зменшити фізичний дискомфорт під час роботи ○ Поліпшити комунікаційні навички та передачу знань молодшим колегам	○ Неструктуровані стендапи ○ Затягнуті дискусії під час планування спринтів ○ Втрата фокусу через часті переривання ○ Проблеми зі здоров'ям

Олександр — досвідчений Senior Full-Stack розробник з 7-річним стажем роботи в продуктових IT-компаніях. Він активно займається технічними рішеннями, кодує та менторингом молодших розробників. Олександр цінує баланс між роботою та особистим життям, оскільки є одруженим і має маленьку дитину. У вільний час він займається спортом та читає технічну літературу для підтримки свого професійного розвитку. Однак, він стикається з труднощами в організації командних зустрічей, що впливають на його продуктивність та емоційний стан.

Рисунок 2.1. Персона розробника Олександра

Таблиця 2.1

Journey Map розробника Олександра

Етап	Активність	Відчуття	Можливості для покращення
Ранковий стендап	Звітування про статус задач, планування дня	Напружений через складність згадати всі деталі за попередній день	Впровадження інструменту для швидких нотаток протягом дня
Планування спринту	Оцінка задач, технічні обговорення, розподіл роботи	Втомлений зтяжними дискусіями та фізичним дискомфортом та невпевнений через	Встановлення чітких часових рамок з перервами на фізичну активність
Робота над власними задачами	Розробка складних функціональностей, технічні дослідження, архітектурні рішення	Перевантажений, починає стресувати через постійні переривання на допомогу колегам та втрату фокусу	Виділення "тихих годин", використання техніки Pomodoro, встановлення статусу "не турбувати"
Кінець робочого дня	Підведення підсумків, планування наступного дня, перехід до особистого життя	Виснажений накопиченою втомою, але відчуває довгоочікуване полегшення	Чіткий ритуал завершення дня, що включає підведення підсумків по виконаним задачам та планування наступного дня

Мапа подорожей Олександра виявляє дві ключові проблеми в його робочому процесі: складнощі під час командних зустрічей, що потребують кращої структури комунікації, та неефективне управління часом через часті переривання, що вказує на необхідність впровадження "тихих годин" для підвищення продуктивності як особистої, так і командної роботи.

Таблиця 2.2

Емоційний та поведінковий профіль розробника Олександра

Аспект	Опис
Думає	Постійно аналізує свою роль та вплив на команду, відчуває тиск через необхідність підтримувати баланс між роботою та особистим життям, сумнівається у своїй здатності ефективно передавати знання молодшим колегам.

Бачить	Невизначеність у структурі командних зустрічей, що призводить до втрати фокусу, важливість чіткої комунікації під час планування спринтів, можливості для покращення процесів, але відсутність чітких рішень.
Відчуває	Фрустрацію через затягнуті дискусії та неструктуровані зустрічі, втомленість від тривалого сидіння та фізичного дискомфорту, стрес через незаплановані звернення колег, які заважають зосередитися на власних задачах.
Робить	Бере активну участь у командних зустрічах, але відчуває труднощі в структурованій подачі інформації, прагне впроваджувати нові методи для покращення комунікації та організації роботи, використовує техніки управління часом, такі як Pomodoro, для підвищення продуктивності.

Олександр постійно аналізує свою роботу та команду, прагнучи до вдосконалення, але його сумніви у власних силах можуть заважати впевненості в прийнятті рішень. Він усвідомлює проблеми в організації командних зустрічей, що негативно впливають на продуктивність, і відчуває фрустрацію та стрес через неефективну комунікацію та фізичний дискомфорт. Олександрові слід зосередитися на структуризації обговорень і впровадженні чітких часових рамок, що допоможе йому підвищити впевненість у своїх комунікаційних навичках, зменшити стрес і покращити загальне самопочуття, що, в свою чергу, позитивно вплине на його продуктивність.

Таблиця 2.3

Jobs-to-be-Done та пропозиції цінності розробника Олександра

Jobs-to-be-Done	Поточні альтернативи	Пропозиція цінності
Олександр хоче ефективно проводити командні зустрічі та донести інформацію про статус задач	Структуризація виконаної роботи та запланованої роботи з пам'яті на ходу під час мітингу	Інструменти для підведення підсумків по прогресу задач, що дозволяють ефективно донести інформацію та зберегти фокус команди під час щоденних мітингів
Олександрові потрібно підтримувати фокус на технічних задачах	Безперервно робити задачі та відпочивати по мірі виконання	Фокусування на роботі та регулярні короткострокові перерви на відпочинок
Олександрові необхідно зменшити навантаження на зір	Працювати за комп'ютером доки втома очей не стане відчутною	Регулярні перерви на переведення фокусу очей на дальні об'єкти

Болючі точки розробника Олександра

Компонент	Опис
Проблеми	Неструктуровані стендапи, затягнуті дискусії під час планування спринтів, погіршення стану здоров'я
Емоційні аспекти	Невпевненість внаслідок неструктурованості робочого процесу, стрес від втрати фокусу

Отже, даний аналіз Jobs-to-be-Done може допомогти виявити потреби Олександра та сформулювати пропозиції цінності, які відповідають цим потребам. Зосередження на структуризації обговорень і впровадженні чітких часових рамок може суттєво покращити його комунікаційні навички, зменшити стрес і підвищити загальну продуктивність.

Персона проєктного менеджера Ірини

Ірина працює на посаді Project Manager та має 5-річний досвід роботи в ІТ-компаніях, керує командою з 10 осіб, відповідальна за планування проєктів, управління ресурсами та комунікацію з клієнтами. Вона активно впроваджує нові методи управління проєктами, зокрема техніку Pomodoro, щоб підвищити продуктивність команди. Ірина заміжня, має двох дітей, тому намагається знайти баланс між роботою та сімейним життям. У вільний час займається йогою та читає книги з особистісного розвитку.

Ірина часто стикається з труднощами в плануванні та організації робочого часу команди. Вона помічає, що без чіткої структури робочих сесій команда втрачає фокус, а терміни виконання проєктів затягуються. Під час щотижневих мітингів Ірина відчуває, що обговорення стають занадто тривалими та неструктурованими. Це призводить до втрати часу та зниження продуктивності та відчуттю фізичного дискомфорту, що впливає на її концентрацію та загальну продуктивність.

На рисунку 2.2 надано опис персони Ірини:



Ірина Бобрик
Project Manager

Вік: 35
Освіта: Національний університет "Кієво-Могилянська академія"
Рідне місто: Київ
Сім'я: Заміжня, має двох дітей

"Ваша мрія не має терміну придатності; дайте собі час і простір для її реалізації."

Цілі

- Знайти баланс між роботою та сімейним життям
- Підвищити продуктивність команди
- Впроваджувати нові методи управління проектами

Розчарування

- Складнощі в плануванні та організації робочого часу команди
- Неструктуровані зустрічі
- Затягнуті обговорення

Ірина — досвідчений Project Manager з п'ятирічним стажем роботи в IT-сфері, яка керує командою з десяти осіб. Вона активно впроваджує нові методи управління проектами, зокрема техніку Pomodoro, щоб підвищити продуктивність команди. Ірина заміжня і виховує двох дітей, тому намагається знайти баланс між роботою та сімейним життям. У вільний час займається йогом та читає книги з особистісного розвитку. Проте, вона стикається з труднощами в організації робочого часу, що призводить до стресу та фрустрації.

Рисунок 2.2. Персона проєктного менеджера Ірини

Таблиця 2.5

Journey Map проєктного менеджера Ірини

Етап	Активність	Відчуття	Можливості для покращення
Ранковий стендап	Звітування про статус проєктів	Фрустрація через неструктуровану подачу інформації та втрату фокусу членів команди	Формування структурованого звіту по виконаних та запланованих задачах
Планування проєкту	Оцінка задач, розподіл роботи	Невпевнена в ефективності обговорення через великий об'єм матеріалу і недостатню залученість інших учасників обговорення внаслідок їх втоми	Організація колективних коротких перерв на фізичну активність, освіження думок
Кінець робочого дня	Підведення підсумків, планування наступного дня, перехід до особистого життя	Виснажена довгою безперервною роботою, відчуває фрустрацію внаслідок неможливості бачення виконаної роботи	Виведення списку виконаних та не виконаних завдань у вигляді списку для можливості оцінки продуктивності і планування наступного дня

Ірина починає свій робочий день з ранкового стендапу, де команда звітує про статус проєктів. Однак, вона часто стикається з проблемою неструктурованої подачі інформації, що призводить до втрати фокусу та фрустрації. Ірина відчуває напруженість, коли обговорення затягуються, і це негативно впливає на загальну продуктивність команди. Під час планування проєкту Ірина намагається оцінити задачі та розподілити роботу між членами команди. Проте, затягнуті дискусії та відсутність чіткої структури викликають у неї втому та невпевненість. Вона усвідомлює, що без встановлення часових рамок та перерв на відпочинок, команда не може ефективно працювати. Наприкінці робочого дня Ірина підводить підсумки і планує наступний день та не може скласти повноцінну картину актуального обсягу робіт.

Загалом, Journey Map Ірини демонструє її боротьбу з управлінням часом, командною комунікацією та фізичним дискомфортом. Впровадження структурованих підходів до організації зустрічей та навчальних сесій, а також використання техніки Pomodoro, може суттєво підвищити її ефективність та зменшити стрес, що в свою чергу позитивно вплине на загальну продуктивність команди.

Таблиця 2.6

Емоційний та поведінковий профіль Project Manager Ірини

Аспект	Опис
Думає	Постійно аналізує ефективність команди, відчуває тиск через необхідність підтримувати баланс між роботою та особистим життям
Бачить	Невизначеність у структурі командних зустрічей, що призводить до втрати фокусу
Відчуває	Фрустрацію через затягнуті дискусії та неструктуровані зустрічі
Робить	Активно шукає рішення для покращення процесів, впроваджує нові методи управління часом

Емоційний та поведінковий профіль Ірини характеризується високою чутливістю до стресу та прагненням до структурованості в роботі. Вона відчуває тривогу під час неорганізованих командних зустрічей, але отримує задоволення від визнання її ідей колегами. Її схильність до перфекціонізму та бажання впроваджувати чіткі процедури іноді призводить до надмірного контролю та

напруженості у стосунках з колегами. Хоча Ірина активно працює над покращенням комунікації, її страх невдачі може заважати відкритому спілкуванню. Її профіль відображає постійний пошук балансу між професійними амбіціями та особистими переживаннями, що вказує на необхідність розвитку навичок управління емоціями для досягнення більшої гармонії в роботі.

Таблиця 2.7

Jobs-to-be-Done та пропозиції цінності Project Manager Ірини

Jobs-to-be-Done	Поточні альтернативи	Пропозиція цінності
Ірина хоче організовувати ефективні командні зустрічі незважаючи на затяжні дискусії	Вести дискусії поки не будуть обговорені та з'ясовані всі проблеми	Таймер, який буде сигналізувати про необхідність зробити регулярну перерву для освіження думок
Ірині необхідно розуміти статус виконання роботи, мати представлення про об'єм невиконаної роботи	Структуризація інформації у голові, приблизне розуміння картини	Інструменти для фіксування прогресу та структуризації прогресу роботи

Таблиця 2.8

Болючі точки Project Manager Ірини

Компонент	Опис
Проблеми	Неструктуровані стендапи, затягнуті дискусії під час планування проєктів.
Емоційні аспекти	Відчуття фрустрації через неефективну комунікацію, стрес від недостатньої організації зустрічей.

Отже, Ірина, як Project Manager, стикається з численними викликами в управлінні часом та командною комунікацією. Впровадження структурованих підходів до організації зустрічей та навчальних сесій, а також використання техніки Pomodoro, може суттєво підвищити її ефективність та зменшити стрес.

На основі проведеного аналізу персон та їх Jobs-to-be-Done можемо зобразити їх потреби та проблеми у вигляді порівняльної таблиці, таким чином буде наочно видно в яку сторону нам варто зробити акцент під час проєктування веб-додатку.

Порівняльна таблиця потреб персон

Потреба	Розробник Олександр	Проектний менеджер Ірина
Управління часом	Фокусування під час роботи над задачами	Організація регулярних коротких перерв для покращення продуктивності затяжних мітингів
Планування роботи	Покращення процесів планування спринта, їх продуктивності	Отримання актуальної структурованої інформації від членів команди під час стендапів
Структурованість роботи	Нотування та структуризація виконаної та майбутньої роботи	Розуміння яку роботу було виконано та який об'єм залишається

Use case діаграма є важливим інструментом проектування, який допомагає візуалізувати та документувати взаємодію між користувачами та системою. Вона дозволяє чітко визначити функціональні вимоги до продукту, показуючи, які дії можуть виконувати різні типи користувачів та як система реагує на ці дії.

Зобразимо use case діаграму для персони Олександра на рисунку 2.3:

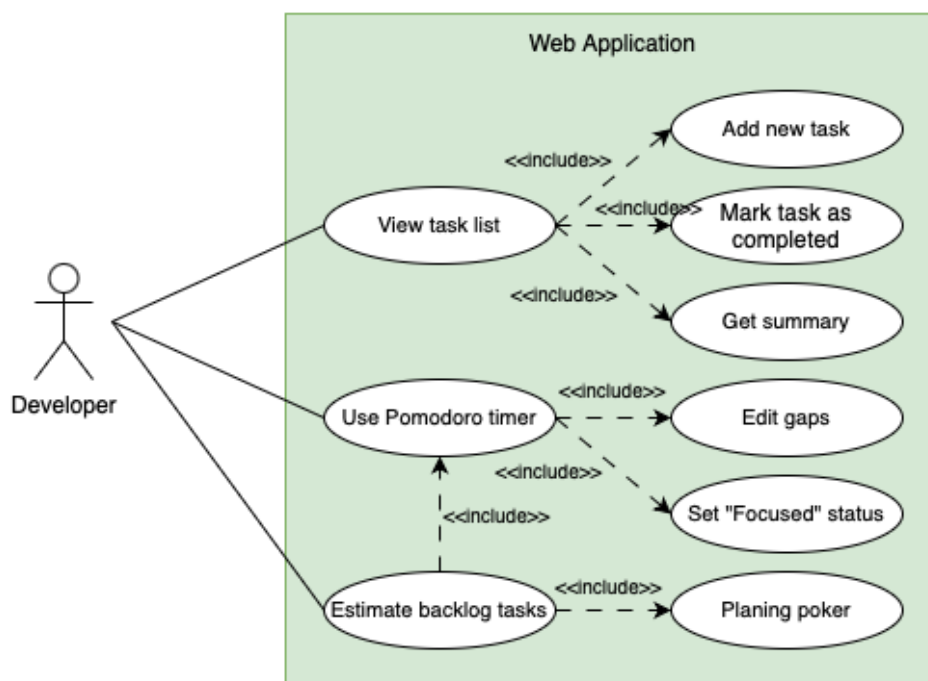


Рисунок 2.3. Use case діаграма розробника Олександра

Основні проблеми, з якими стикається Олександр у своїй роботі, пов'язані з затяжними командними зустрічами, фокусуванням і погіршенням стану здоров'я. На щоденних стендапах йому складно структуровано донести інформацію про статус задач, особливо коли потрібно згадати всі деталі за

попередній день. Щоб упорядкувати цей процес, він використовує список задач. Впродовж дня Олександр варто робити короткі перерви для запобігання проблем зі здоров'ям, у цьому йому допоможе Pomodoro таймер, який також автоматично встановить статус фокусування в робочий час. Під час планування спринтів розробник помічає, що команда часто втрачає фокус через затягнуті дискусії, а оцінка задач займає забагато часу через відсутність чіткої структури обговорення. Для пришвидшення та організованості процесу використовується Planning poker та встановлюються чіткі рамки за допомогою таймеру.

Зобразимо use case діаграму для персони Ірини на рисунку 2.4:

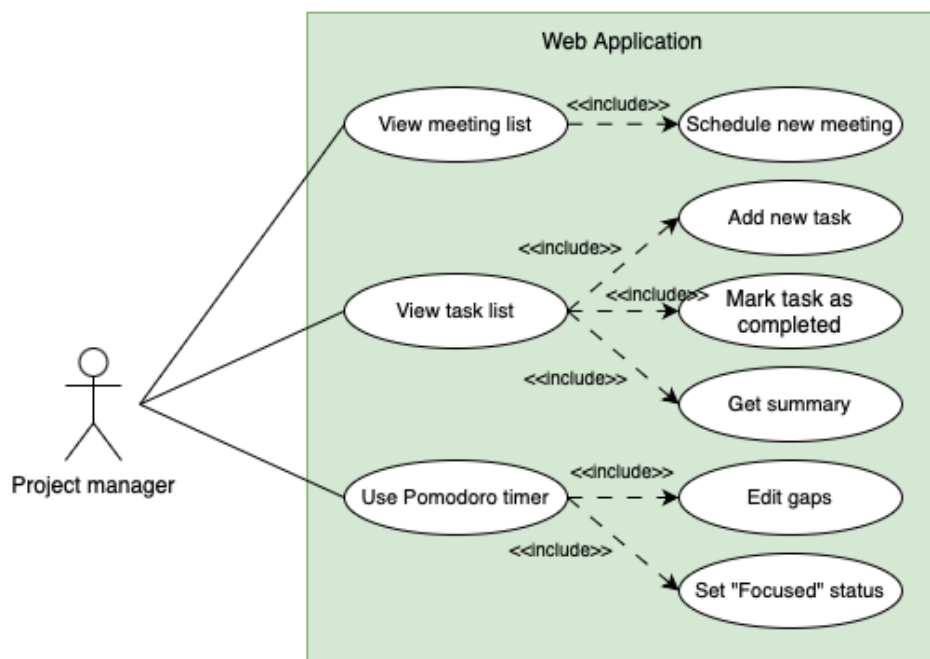


Рисунок 2.4. Use case діаграма проєктного менеджера Ірини

Діаграма ілюструє дещо інші сценарії взаємодії Ірини з додатком у порівнянні з сценаріями Олександра, оскільки проєктний менеджер не бере участі в оцінці задач беклогу, але в її обов'язки входить організація командних зустрічей, для чого їй потрібен функціонал планування мітингів. Щоб не упустити деталі та завдання минулого і сьогоднішнього дня, Ірині необхідно їх нотувати у відповідний список, де вона пізніше зможе підвести підсумки прогресу. Аби вести ефективні дискусії та підтримувати продуктивність впродовж дня, менеджеру стане у нагоді Pomodoro таймер, який розіб'є розумову активність на послідовні фази фокусування та відпочинку.

2.2. Формулювання вимог до основного функціоналу додатку, формування беклогу продукту

Формулювання вимог до веб-додатку з перспективи користувача та стейкхолдера є ключовим для створення успішного продукту, оскільки такий підхід забезпечує відповідність потребам цільової аудиторії, допомагає уникнути розробки непотрібних функцій, узгоджує розробку з бізнес-цілями, покращує комунікацію між розробниками та замовниками, сприяє створенню ефективних тестових сценаріїв та дозволяє створити гнучкий, масштабований продукт, що легко адаптується до змінних потреб ринку та забезпечує кращу окупність інвестицій.

Перш за все, додаток має вирішувати проблему затяжних мітингів та втрату фокусування на задачах. Для цього має бути взятий за основу метод Pomodoro, суть якого працювати інтервалами 25 хвилин та робити перерву на 5 хвилин. Оскільки ми хочемо покращити не тільки особисті процеси, а і командні, варто синхронізувати ці інтервали під час мітингів, щоб всі члени конференції могли одночасно відпочити та повернутися до обговорення з новими силами.

Також слід покращити продуктивність Daily Scrum мітингів, суть яких обговорити з іншими членами команди прогрес по своїм задачам та напрям руху на наступний день. В результаті дослідження процесів команди було виявлено проблему згадування минулого дня, що займає деякий час. Тому у нашому додатку має бути інструмент, який буде допомагати відстежувати свій прогрес за минулий день, відображати виконані та не виконані задачі.

На мітингах Product Backlog Refinement проводиться оцінка задач беклогу в сторі поінтах (зазвичай числа Фібоначчі). Для покращення даного процесу варто доречно додати такий інструмент, як planning poker, де кожен учасник команди виставляє оцінку, яка на його думку відповідає складності задачі, після цього виводиться середня оцінка, або проводиться повторне голосування у випадку розбіжностей думок. Таким чином планування та оцінки історій будуть

більш точними і враховувати думку усіх членів команди, а також такий процес буде набагато продуктивнішим.

Отже, ґрунтуючись на описаних вище проблемах та варіантах їх рішень, можемо сформулювати наступні вимоги до веб-додатку:

Таблиця 2.10

Функціональні вимоги веб-додатку

Вимога	Опис
Управління фокусуванням	Додаток має містити інструмент, який буде автоматизовано вираховувати інтервали роботи та відпочинку і сповіщати користувача
Синхронізація між членами команди	Фокусування має бути синхронізовано між усіма учасниками мітингів, щоб уникнути випадків коли лише частина команди бере участь у обговоренні, а інша відпочиває
Список командних мітингів	Додаток має містити дошку розкладом особистих та командних мітингів, командні мітинги мають бути синхронізовані між усіма членами команди
Трекінг завдань та прогресу	Інтерфейс для відображення списку та статусу виконання відповідних завдань на вчорашній та сьогоднішній день. Так користувач буде знати що він зробив або не встиг зробити вчора і планує зробити сьогодні
Естимування беклогу	Команда повинна мати інструмент (planning poker), за допомогою якого може автоматизовано та незалежно від думки інших поставити оцінку відповідній user story

Опис нефункціональних вимог проєкту є критично важливим, оскільки вони визначають якісні характеристики та обмеження системи, які безпосередньо впливають на її успішність та прийнятність користувачами. Нефункціональні вимоги охоплюють такі аспекти як продуктивність, безпека, масштабованість, зручність використання та надійність системи. Їх чітке визначення допомагає команді розробників приймати правильні архітектурні рішення, вибирати відповідні технології та встановлювати відповідні метрики якості. Без належного опису нефункціональних вимог існує ризик створення продукту, який хоч і виконує всі необхідні функції, але може бути повільним, ненадійним, небезпечним або складним у використанні, що призведе до незадоволення користувачів та можливих проблем у майбутньому.

Проаналізувавши вищевказаний необхідний функціонал, можемо сформулювати наступні технічні вимоги у таблиці 2.11:

Таблиця 2.11

Нефункціональні вимоги веб-додатку

Вимога	Опис
Продуктивність	Веб-додаток повинен швидко працювати з даними та підтримувати одночасну роботу для команди до 20 користувачів без втрати продуктивності
Масштабованість	Додаток має бути представлений у вигляді окремих модулів API та безпосередньо веб-додатку. Така структура надасть можливість подальшого масштабування проєкту шляхом розробки не лише веб-додатку, а і нативних мобільних/десктопних додатків, які будуть працювати з одним API
Локалізація	Інтерфейс додатку повинен підтримувати українську та англійську локалізацію на початковому етапі та мати можливість додавання інших мов в процесі подальшого розвитку додатку
Технологічний стек	Фронтенд додатку варто реалізувати за допомогою бібліотеки React, яка є досить швидкою та дозволяє створювати динамічні інтерфейси. Бекенд повинен бути розроблений на базі .NET, оскільки саме ця платформа забезпечить безпроблемний подальший розвиток проєкту
Зберігання даних	На початку життєвого циклу проєкту варто використовувати Azure Table Storage в якості бази даних. Це досить швидко та недороге рішення, яке чудово працює у зв'язці з .NET сервером і якого буде більш ніж достатньо на час розробки і перший час функціонування додатку
Архітектура	Має бути відокремлений рівень роботи з базою даних, щоб у майбутньому можна було безпроблемно перейти на Microsoft SQL Server, або інші подібні рішення
Безпека даних	Має бути забезпечена конфіденційність користувачів, особливу увагу варто приділити шифруванню записів нотаток та запланованих мітингів користувачів, адже дана інформація може нести засекречену та важливу для бізнесу інформацію
Зручність	Інтерфейс має бути зручним та інтуїтивно зрозумілим для різних груп користувачів
Надійність	Система повинна мати високу доступність (uptime $\geq 99.9\%$) і підтримувати безперебійну роботу

Опис обмежень програмного продукту визначає технічні, ресурсні та функціональні рамки проєкту, допомагає в управлінні ризиками, встановлює чіткі очікування для всіх зацікавлених сторін та забезпечує ефективне планування розробки, що є критично важливим для успішної реалізації проєкту

та досягнення поставлених цілей. Опишемо такі обмеження у вигляді таблиці 2.12:

Таблиця 2.12

Обмеження веб-додатку

Обмеження	Опис
Технічні	Веб-додаток має коректно працювати на десктопних пристроях, правильно відображатись в усіх популярних браузерах як Google Chrome, Safari, Mozilla Firefox
Ресурсні	Мають бути враховані доступність усіх фахівців, які будуть причасні до розробки продукту, передбачити форс-мажорні випадки
Функціональні	Має бути реалізована система доступів та сесій, користувачі повинні мати доступ лише до своїх та тільки публічних даних інших користувачів

Беклог продукту являє собою впорядкований список всіх необхідних функцій, покращень, виправлень та інших завдань, які потрібно виконати для успішної розробки продукту. Це динамічний документ, який постійно оновлюється та пріоритезується Product Owner'ом відповідно до потреб бізнесу та користувачів. Беклог можна представити як "живий" список побажань, де кожен елемент описує певну цінність, яку продукт повинен принести користувачам.

Користувацькі історії (User Stories) представляють собою короткі, прості описи функціональності продукту з точки зору кінцевого користувача. Вони зазвичай слідуєть формату: "Як [персона], я хочу [дія/функція], щоб [отримати певну цінність/результат]". Наприклад: "Як покупець, я хочу додати товар у кошик, щоб пізніше його придбати". Персоною може виступати будь-хто, наприклад, користувач, розробник або менеджер[37]. User Stories допомагають команді розробки краще зрозуміти потреби користувачів та забезпечують чітке бачення того, що потрібно розробити. Вони є основними елементами беклогу продукту і часто доповнюються критеріями прийняття, які визначають, коли історія вважається завершеною.

Сформуємо первинний беклог продукту. Одразу варто зробити акцент на слові «первинний», адже, як було сказано раніше, це динамічний документ та ми

не можемо сформувати чіткий список, який одразу опише всю розробку продукту, всі пункти ще можуть бути обговорені та змінені за згодою розробників та Product Owner'a в процесі розробки. Узагальнимо його у вигляді таблиці 2.13:

Таблиця 2.13

Первинний беклог продукту

Опис	Acceptance Criteria
Як архітектор веб додатку, я хочу ініціалізувати проєкт, щоб розробники могли почати паралельну роботу над серверною та клієнтською частинами	- серверна частина має представляти собою API у вигляді .NET Core рішення; - клієнтська частина являє собою SPA проєкт на базі React з підтримкою typescript.
Як front-end розробник, я хочу мати API для отримання локалізованих текстів, щоб відобразити їх у інтерфейсі користувача	- API має повертати список усіх доступних мов; - API має повертати локалізований текст у запиті, який містить параметри мови та ключа тексту.
Як користувач, я хочу бачити локалізовані тексти та мати можливість змінити мову, щоб розуміти контент сторінки	- додаток має містити перемикач мов, який містить список усіх доступних локалізацій; - за замовчуванням має бути встановлена мова інтерфейсу користувача. Якщо така мова не підтримується додатком, локалізація за замовчуванням – англійська; - після зміни мови всі тексти додатку мають автоматично підвантажити нову локалізацію.
Як користувач, я хочу мати панель навігації, яка буде відображати основні сторінки, щоб я міг зручно та ефективно між ними перемикатися	- панель навігації має бути закріпленою в лівій частині сторінки та представляти собою відповідні; - елементи панелі навігації: домашня сторінка (іконка «будинок»), сторінка списку справ (іконка «список»), сторінка планування (іконка «плитки»); - по кліку на елемент має відбуватися навігація на відповідну сторінку.
Як користувач, я хочу мати Pomodoro таймер, щоб ефективно управляти своїм часом	- в центрі домашньої сторінки має бути розміщений таймер; - керування таймером здійснюється кнопками «стоп» та «старт» і «пауза», які чередуються інтервалами 25 та 5 хвилин; - таймер має містити інформацію про залишок часу та час, у який він спрацює.

Продовження таблиці 2.13

Як користувач, я хочу мати можливість редагувати інтервалу таймеру, щоб більш гнучко налаштувати свій робочий процес	- таймер має містити кнопку «Редагувати»; - при натисканні на кнопку має випасти попап для введення нових значень інтервалів.
Як користувач, я хочу постійно бачити прогрес таймеру на усіх сторінках, щоб краще орієнтуватися у часі	- панель навігації повинна містити блок, в якому буде зображено залишок часу та кнопки управління таймером; - вищеописаний блок повинен бути відображений на усіх сторінках окрім домашньої, оскільки вона вже містить таймер.
Як користувач, я хочу мати список мітингів, які заплановані на сьогоднішній день, щоб краще планувати свій час	- головна сторінка має містити блок, на якому відображено заплановані на сьогодні особисті та командні мітинги; - має бути зображена шкала часу та показчик поточного часу. - компонент повинен містити кнопки навігації вперед та назад по дням і кнопку вибору конкретної дати - повинна бути можливість запланувати новий мітинг за допомогою кнопки «+», при її натисканні має з'явитися попап вікно, в якому користувач може заповнити всю необхідну інформацію про мітинг
Як користувач, я хочу мати список справ, щоб фіксувати що я зробив та не зробив за сьогодні	- сторінка списку справ має містити список з елементами; - кожен елемент повинен складатися з заголовка і контенту та підсвічуватися обраним кольором; - контент може бути як звичайним текстом, так і списком з чекбоксами; - користувач повинен мати можливість внести зміни в елементи; - кожен елемент може бути помічений як виконаний; - виконані елементи переміщуються в верхню частину списку.
Як користувач, я хочу мати можливість згорнути та розгорнути список виконаних елементів	- між виконаними та невиконаними елементами має бути роздільник з кнопкою для відображення або приховання виконаних елементів списку; - за замовчуванням список виконаних елементів має бути згорнутим.

Як користувач та член команди, я хочу мати planning poker, щоб проголосувати за оцінку, яку вважаю за потрібне поставити в якості story point відповідної user story	- на сторінці планування має бути planning poker, що являє собою список карток, які належать відповідному члену команди та зображують поставлену ним оцінку; - в нижній частині сторінки заходиться панель з оцінками та кнопками підтвердження та скасування голосу; - сторінка містить блок, який відображає середню оцінку, моду та медіану; - після вибору та підтвердження оцінки голос користувача відображається у інших членів команди.
Як член команди, я хочу бачити список та статус своїх колег, щоб розуміти коли краще їх не турбувати	- в правій стороні сторінки має знаходитись блок з списком членів команди і їх статусом доступності; - можливі статуси: доступний, оффлайн, відійшов, у мітингу, не турбувати; - блок має бути закріплений та відображатись на усіх сторінках.

Отже, описаний набір перших користувацьких історій являється відправною точкою для формування беклогу продукту. Даний список має бути розглянутий командою розробників в рамках мітингу Backlog Refinement, де фахівці обговорять об'єм робіт та оцінять їх в спеціальних одиницях вимірювання Story Points. Також нормальною практикою вважається прохання команди до Product Owner'а внести зміни, розбити історію на декілька менших, або додати нові історії. Важливо розуміти, що неможливо створити беклог одразу для всього продукту по причині його динамічності, але варто мати список історій на декілька спринтів вперед, щоб не блокувати процес розробки та не допустити ситуації, коли розробники не знають що робити далі.

2.3. Проектування макету додатку

Проектування макету додатку — це комплексний процес створення візуального представлення інтерфейсу користувача (UI) та структури додатку, що включає розробку ескізів, прототипів та графічних елементів, які можуть

бути як статичними, так і інтерактивними. Цей критично важливий етап розробки програмного забезпечення дозволяє командам візуалізувати концепцію продукту до початку розробки, тестувати ідеї, отримувати зворотний зв'язок від користувачів та виявляти потенційні проблеми на ранніх етапах, що сприяє економії ресурсів, покращенню комунікації між учасниками проєкту та створенню інтуїтивно зрозумілого інтерфейсу, який відповідає очікуванням замовника та потребам користувачів. в команді, знижуючи ризики та підвищуючи користувацький досвід.

Ґрунтуючись на сформованих раніше вимогах, за допомогою інструменту Figma спроектуємо макети основних трьох сторінок інтерфейсу нашого веб-додатку. На рисунку 2.5. зображено макет головної сторінки:

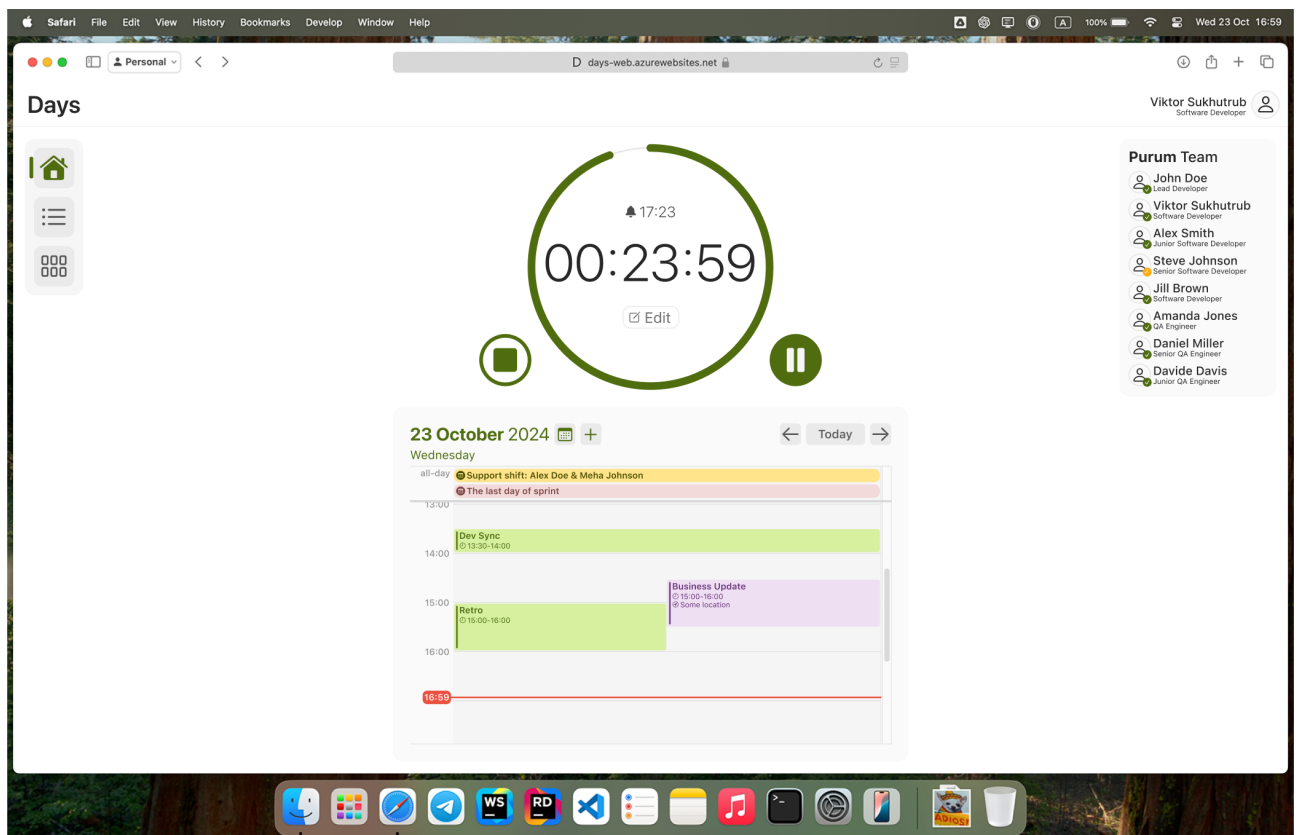


Рисунок 2.5. Макет головної сторінки веб-додатку

На головній сторінці буде доречно розмістити безпосередньо сам таймер, оскільки він є головним елементом нашого продукту. Нижче буде розміщуватися блок, який відображає інтервали та заплановані події на сьогоднішній день з прив'язкою до часу, який відображається на вертикальній шкалі.

Також на усіх сторінках з лівої сторони можна помітити навігаційну панель, яка містить три основні сторінки. Права панель відображує назву команди та список її членів з статусами доступності, таким чином можна наглядно побачити хто зараз доступний до комунікації, а кого не варто турбувати. Хедер на усіх сторінках однаковий, він відображує ім'я користувача, посаду та зображення профіля.

На рисунку 2.6 розглянемо наступну сторінку, на якій буде відображатися список задач, який користувач виконав або запланував:

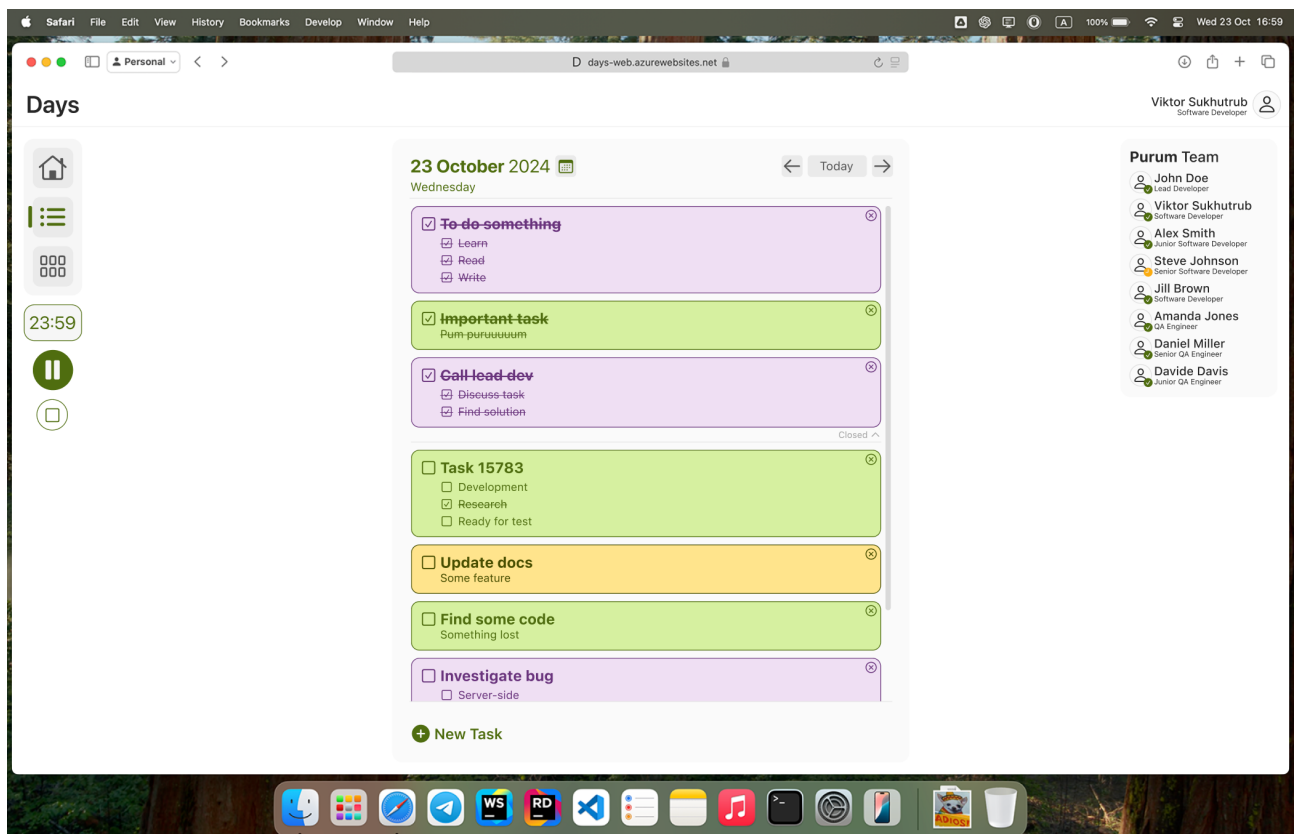


Рисунок 2.6. Макет сторінки списку справ

Всі задачі розбиті по дням, можна подивитися прогрес за вчорашній день, зафіксувати за сьогоднішній, запланувати задачі на завтра та взагалі переглянути список за будь-який день. Кожний елемент списку можна позначити як виконаний та відсортувати за статусом, щоб прогрес був більш наглядний. Це може бути корисно на щоденних мітингах, де учасник команди має поділитися з іншими колегами прогресом за минулий день та планами на наступний. Також

варто зауважити, но на сторінці разом з елементами управління таймером відображається і безпосередньо час.

На усіх (окрім головної) сторінках навігаційна панель також містить блок управління таймером, який є головним об'єктом нашого додатку, тому дані елементи керування та прогрес таймера будуть показані на цих сторінках для зручності користувача.

На третій основній сторінці буде розміщено Planning poker, її макет зображено на рисунку 2.7:

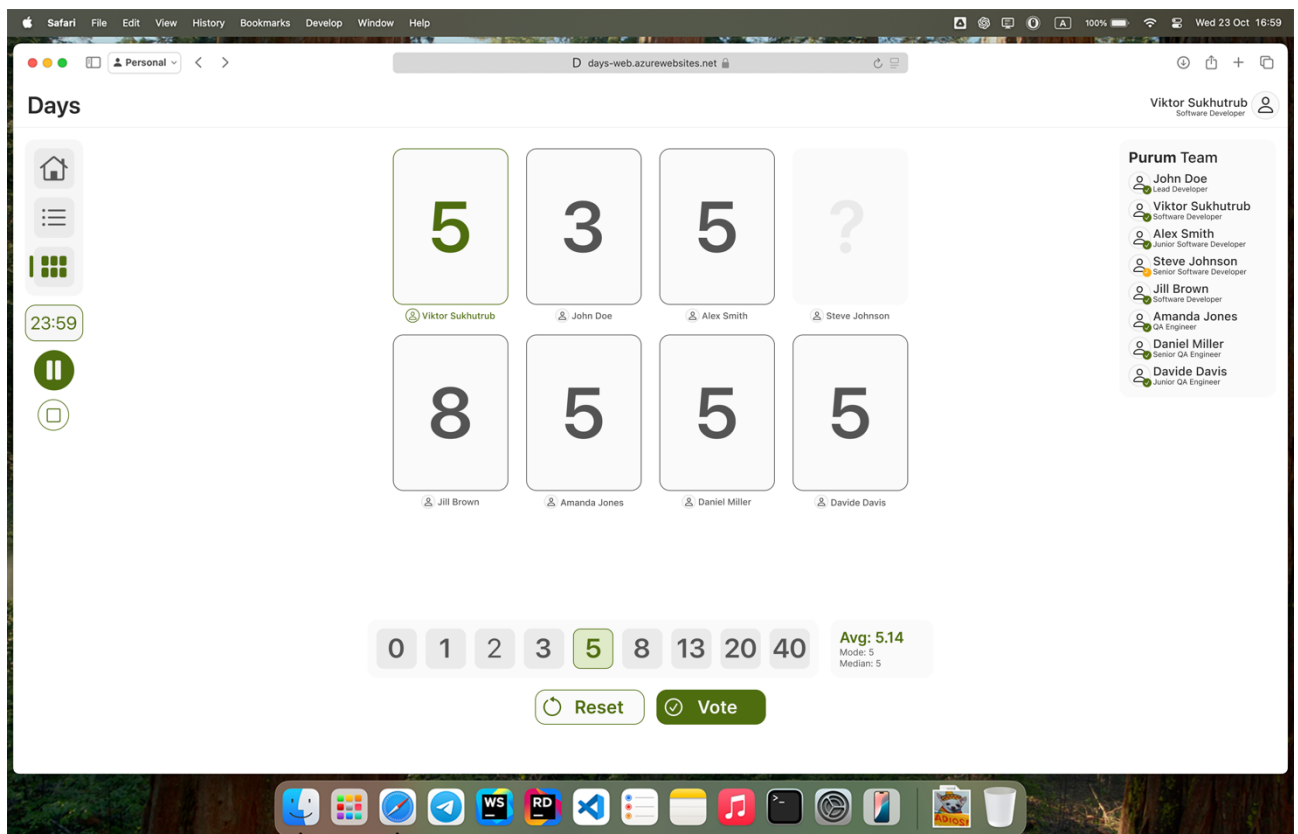


Рисунок 2.7. Макет сторінки планування

У верхній частині сторінки відображаються картки з оцінкою та ім'ям кожного члена команди, які беруть участь у сесії оцінювання. На нижній панелі можна буде обрати оцінку і проголосувати або відмінити свій голос. Хост сесії за допомогою відповідної кнопки буде мати змогу скинути прогрес голосування, щоб перейти до розгляду нової задачі.

Карти навігації є ключовим інструментом у проектуванні веб-додатків, що допомагають організувати контент і функціональність у логічну структуру,

дозволяючи користувачам швидко знаходити потрібну інформацію та знижуючи ймовірність фрустрації, а також надають можливість візуалізувати користувацький шлях, виявляти потенційні проблеми та служать спільною мовою для всіх учасників проєкту, що в результаті сприяє створенню більш інтуїтивних та функціональних продуктів, покращенню комунікації в команді та підвищенню загальної задоволеності користувачів.

Таблиця 2.14

Карти навігації користувача

Задача	Кроки
Запустити таймер на 25 хвилин	1. Відкриваємо додаток 2. Головний екран 3. Бачимо таймер 4. Натискаємо кнопку запуску таймеру 5. Починається відлік
Налаштувати час таймеру	1. Відкриваємо додаток 2. Головний екран 3. Бачимо таймер 4. Натискаємо кнопку «Edit» 5. Вводимо бажаний час у спливаючому вікні 6. Натискаємо кнопку «Save» 7. Бачимо збережені зміни
Додати заплановану подію в календарі	1. Відкриваємо додаток 2. Головний екран 3. Бачимо розклад дня і поточний час 4. Натискаємо кнопку «+» 5. Заповнюємо необхідні поля у спливаючому вікні 6. Підтверджуємо дію 7. Бачимо додану подію в нашому календарі
Додати пункт у список справ	1. Відкриваємо додаток 2. Головний екран 3. Переходимо на сторінку задач за допомогою бокового навігаційного меню 4. Бачимо список наявних справ 5. Натискаємо кнопку «New Task» 6. Заповнюємо необхідні поля 7. Підтверджуємо збереження

Відмітити пункт у списку справ як виконаний	1. Відкриваємо додаток 2. Головний екран 3. Переходимо на сторінку задач за допомогою бокового навігаційного меню 4. Бачимо список наявних справ 5. Натискаємо чекбокс напроти відповідного пункту 6. Бачимо, що задача помічена закресленням та перенесена в групу «Closed»
Переглянути список виконаних справ	1. Відкриваємо додаток 2. Головний екран 3. Переходимо на сторінку задач за допомогою бокового навігаційного меню 4. Бачимо список наявних справ 5. Натискаємо на пункт «Closed», щоб розгорнути список 6. Бачимо список виконаних справ
Видалити пункт з списку справ	1. Відкриваємо додаток 2. Головний екран 3. Переходимо на сторінку задач за допомогою бокового навігаційного меню 4. Бачимо список наявних справ 5. Натискаємо кнопку «х» відповідного пункту 6. Бачимо, що задача пропала з нашого списку
Проголосувати в Planning Poker	1. Відкриваємо додаток 2. Головний екран 3. Переходимо на сторінку Planning Poker за допомогою бокового навігаційного меню 4. Бачимо список приєднаних до сесії голосування учасників команди 5. В нижній панелі обираємо свою оцінку 6. Натискаємо кнопку «Vote» 7. Бачимо нашу оцінку та оцінки колег на панелі карток 8. Чекаємо поки всі проголосують 9. Бачимо середню оцінку, моду та медіану справа від панелі з оцінками

Виконаний аналіз карт навігації веб-додатку демонструє його інтуїтивно зрозумілу структуру з легким доступом до основних функцій (таймер, календар, список справ), чітко визначеними кроками для виконання різних дій та зручним боковим навігаційним меню, що забезпечує швидке перемикання між розділами, а така продумана організація інтерфейсу не лише підвищує ефективність роботи

та зменшує ймовірність помилок, але й позитивно впливає на загальне враження користувачів, їхню лояльність та успішність продукту в цілому.

Висновки до другого розділу

У другому розділі було проведено комплексний аналіз та проектування веб-додатку для ефективного управління часом.

По-перше, було детально описано цільову аудиторію продукту - команду розробників програмного забезпечення, що працює за методологією Scrum. Проаналізовано їх основні процеси, проблеми та потреби, що дозволило краще зрозуміти вимоги до функціоналу додатку.

По-друге, створено детальні моделі взаємодії користувачів через розробку персон Senior Full-Stack розробника та Project Manager. Для кожної персони було складено карти подорожей, емоційні профілі та проведено Jobs-to-be-Done аналіз, що допомогло виявити ключові сценарії використання продукту.

По-третє, сформульовано чіткі функціональні та технічні вимоги до додатку, включаючи управління фокусуванням, синхронізацію між членами команди, трекінг завдань, естимування беклогу та інші критичні функції. Визначено технологічний стек та системні вимоги.

По-четверте, розроблено детальні макети основних сторінок додатку - головної сторінки з таймером, сторінки списку справ та сторінки планування. Створено карти навігації, що описують покрокові сценарії взаємодії користувачів з системою.

Таким чином, проведена робота створила міцну основу для подальшої розробки продукту, що відповідає потребам цільової аудиторії та має потенціал для успішного впровадження в реальне робоче середовище команд розробників.

РОЗДІЛ 3. ПЛАН ПРОЄКТУ РОЗРОБКИ ІТ-ПРОДУКТУ

3.1. Стейкхолдери проєкту та їх класифікація

Ефективна взаємодія з усіма групами стейкхолдерів є критично важливою для успішної реалізації проєкту та досягнення поставлених цілей. Необхідно регулярно проводити аналіз їхніх потреб та очікувань шляхом проведення опитувань, інтерв'ю та збору зворотного зв'язку. Це дозволяє вчасно виявляти потенційні ризики та можливості для покращення продукту. Важливим аспектом є також коригування стратегії взаємодії відповідно до змін у проєкті, що забезпечує гнучкість та адаптивність процесу розробки. Такий підхід дозволяє підтримувати високий рівень залученості всіх зацікавлених сторін та забезпечує прозорість процесу розробки. Крім того, постійний моніторинг задоволеності стейкхолдерів допомагає вчасно вносити необхідні корективи в продукт та процеси його розробки, що значно підвищує шанси на успішну реалізацію проєкту.

У рамках проєкту розробки веб-додатку для ефективного управління часом на основі методу Pomodoro можна виділити наступні групи стейкхолдерів:

Таблиця 3.1

Групи стейкхолдери ІТ-продукту

Група	Члени групи	Інтереси та очікування
Команда розробки	- розробники (Frontend, Backend); - UI/UX дизайнери; - QA інженери; - Lead Developer; - Scrum Master.	- чіткі технічні вимоги; - зрозумілі терміни виконання; - можливість професійного розвитку; - якісний код та документація; - ефективна комунікація в команді.
Керівництво проєкту	- керівник проєкту; - технічний лідер; - product owner.	- успішна реалізація проєкту; - дотримання термінів та бюджету; - якісний кінцевий продукт; - задоволеність користувачів.

Продовження таблиці 3.1

Кінцеві користувачі	- розробники; - менеджери.	- зручний та інтуїтивний інтерфейс; - надійність роботи системи; - підвищення продуктивності роботи; - ефективна організація командної роботи.
Інвестори та спонсори	- венчурні фонди; - приватні інвестори; - компанії-партнери.	- повернення інвестицій; - ринковий успіх продукту; - зростання вартості проєкту; - прозора звітність.
Конкуренти	- розробники аналогічних рішень; - компанії з подібними продуктами.	- аналіз нового гравця на ринку; - оцінка конкурентних переваг; - можливості для співпраці.

Команда розробки відіграє ключову роль у створенні технічної складової проєкту. Розробники відповідають за написання якісного та підтримуваного коду, впровадження необхідного функціоналу та забезпечення стабільної роботи системи. UI/UX дизайнери створюють інтуїтивно зрозумілий інтерфейс, що відповідає потребам користувачів. QA інженери забезпечують якість продукту через всебічне тестування та виявлення потенційних проблем на ранніх етапах розробки. Lead Developer та Scrum Master координують роботу команди, забезпечують ефективну комунікацію та дотримання методології розробки.

Керівництво проєкту має стратегічну роль у визначенні напрямку розвитку продукту та прийнятті ключових рішень. Product Owner формує бачення продукту, визначає пріоритети розробки та забезпечує відповідність результатів очікуванням користувачів. Ведучий розробник приймає архітектурні рішення та забезпечує технічну життєздатність проєкту, відповідає за загальну успішність реалізації, включаючи дотримання термінів, бюджету та якості.

Кінцеві користувачі є центральними стейкхолдерами, оскільки саме для них створюється продукт. Їхній досвід використання, відгуки та побажання безпосередньо впливають на розвиток продукту. Важливо постійно збирати та аналізувати їхній відгук для вдосконалення функціоналу та користувацького

досвіду. Різні групи користувачів мають свої специфічні потреби, які необхідно враховувати при розробці.

Інвестори та спонсори забезпечують фінансову підтримку проєкту та очікують на його комерційний успіх. Їхня роль не обмежується лише фінансуванням, вони також можуть надавати цінні бізнес-поради, допомагати з нетворкінгом та сприяти просуванню продукту на ринку. Важливо підтримувати прозору комунікацію з інвесторами та регулярно звітувати про прогрес проєкту.

Конкуренти, хоча і є непрямими стейкхолдерами, відіграють важливу роль у формуванні ринкового середовища. Аналіз їхньої діяльності допомагає визначити конкурентні переваги власного продукту, виявити незайняті ніші та можливості для інновацій. Крім того, здорова конкуренція стимулює постійне вдосконалення продукту та пошук нових рішень.

Для успішної реалізації проєкту необхідно регулярно проводити аналіз впливу та інтересів стейкхолдерів, оновлювати стратегії взаємодії та забезпечувати ефективну комунікацію між усіма учасниками процесу. Це дозволяє вчасно виявляти потенційні ризики, вирішувати конфлікти інтересів та забезпечувати сталий розвиток проєкту.

Таблиця 3.2

Таблиця впливу стейкхолдерів

Стейкхолдер	Вплив	Інтерес	Стратегія взаємодії
Команда розробки	Високий	Високий	Активне залучення та мотивація
Керівництво проєкту	Високий	Високий	Регулярна комунікація та звітність
Кінцеві користувачі	Середній	Високий	Збір відгуку та адаптація продукту
Інвестори	Високий	Середній	Регулярна звітність про прогрес
Конкуренти	Низький	Середній	Моніторинг активності

Ефективна взаємодія між усіма групами стейкхолдерів забезпечується через регулярні зустрічі, звітність, збір та аналіз зворотного зв'язку. Важливо підтримувати баланс інтересів різних груп та вчасно реагувати на зміни їхніх потреб та очікувань. Це дозволяє створити продукт, який максимально відповідає потребам ринку та забезпечує задоволеність усіх зацікавлених сторін.

Для забезпечення ефективної взаємодії зі стейкхолдерами проєкту розроблено комплексний план їх залучення, який включає кілька ключових етапів. На початковому етапі проводиться детальний аналіз потреб та очікувань кожної групи стейкхолдерів через серію структурованих інтерв'ю та опитувань. Це дозволяє створити чітку картину вимог та побажань усіх зацікавлених сторін.

Для команди розробки передбачено регулярні щотижневі мітинги, де обговорюються поточні завдання, виклики та можливості для професійного розвитку. Важливим елементом є впровадження системи внутрішньої комунікації, яка забезпечує прозорість процесів та швидкий обмін інформацією. Також заплановано проведення щоквартальних workshops для підвищення технічних компетенцій команди.

Взаємодія з керівництвом проєкту структурована через щотижневі status-мітинги та щомісячні стратегічні сесії. На цих зустрічах обговорюються ключові показники ефективності, прогрес проєкту та необхідні корективи в стратегії розвитку. Важливим аспектом є створення системи швидкого прийняття рішень для оперативного реагування на зміни в проєкті.

Для роботи з кінцевими користувачами розроблено програму бета-тестування, яка включає регулярні опитування, збір відгуків та проведення користувацьких інтерв'ю. Передбачено створення спеціальної платформи для збору та аналізу користувацьких відгуків, а також регулярні оновлення продукту на основі отриманих даних.

Взаємодія з інвесторами та спонсорами базується на принципах прозорості та регулярної звітності. Щомісяця готуються детальні звіти про прогрес проєкту, фінансові показники та досягнуті milestone'и. Організуються квартальні презентації для демонстрації нових функціональних можливостей продукту та обговорення стратегічних планів розвитку.

Моніторинг конкурентного середовища здійснюється через регулярний аналіз ринку, відстеження інновацій та вивчення користувацького досвіду конкуруючих продуктів. Ця інформація використовується для вдосконалення власного продукту та пошуку нових можливостей для розвитку.

Таким чином, запропонований план залучення стейкхолдерів є комплексним інструментом, який забезпечує ефективну взаємодію між усіма учасниками проєкту. Систематичний підхід до комунікації, чітко визначені канали зворотного зв'язку та регулярний моніторинг взаємодії дозволяють своєчасно виявляти потенційні проблеми та вживати необхідних заходів для їх вирішення. Впровадження такого плану сприяє підвищенню прозорості процесів розробки, зміцненню довіри між стейкхолдерами та досягненню спільних цілей проєкту. Успішна реалізація описаних механізмів взаємодії створює міцну основу для сталого розвитку проєкту та максимізації його цінності для всіх зацікавлених сторін.

3.2. Декомпозиція робіт, етапи та компоненти проєкту

Розробка веб-додатку для управління часом на основі методу Pomodoro включає п'ять основних етапів, кожен з яких має критичне значення для успіху проєкту.

Перший етап - ініціація та планування, що закладає фундамент всього проєкту. На цьому етапі проводиться глибокий аналіз вимог користувачів, визначаються ключові функціональні та нефункціональні характеристики майбутньої системи. Важливою частиною є розробка технічного завдання, яке детально описує всі аспекти проєкту. Формується команда з необхідними компетенціями, складається детальний план робіт з урахуванням доступних ресурсів та часових обмежень. Особлива увага приділяється оцінці потенційних ризиків та розробці стратегій їх мінімізації.

Таблиця 3.3

Завдання та опис першого етапу

Завдання	Опис	Дії	Результати
Ініціація	Закладення фундаменту проєкту, аналіз вимог користувачів	Проведення аналізу, розробка технічного завдання	Технічне завдання, план робіт, оцінка ризиків

Продовження таблиці 3.3

Формування команди	Вибір учасників проєкту та їх ролей	Проведення співбесід, визначення ролей	Сформована команда, розподіл обов'язків
Аналіз ринку	Дослідження конкурентів та цільової аудиторії	Проведення опитувань, аналіз конкурентів, визначення потреб користувачів	Звіт про ринок, визначення цільової аудиторії

Другий етап – проєктування, який фокусується на технічних аспектах системи. Розробляється архітектура, яка повинна забезпечити надійність, масштабованість та ефективність додатку. Проєктується структура бази даних для оптимального зберігання та обробки даних. Створюються прототипи користувацького інтерфейсу, які дозволяють візуалізувати майбутній продукт та отримати ранній зворотний зв'язок. Визначається технічний стек з урахуванням вимог проєкту та компетенцій команди. Розробляється специфікація API для забезпечення ефективної взаємодії між frontend та backend частинами додатку.

Таблиця 3.4

Завдання та опис другого етапу

Завдання	Опис	Дії	Результати
Проєктування	Розробка архітектури системи, створення прототипів	Розробка архітектури, створення прототипів	Архітектура системи, прототипи інтерфейсу
Визначення технологій	Вибір технологій для реалізації проєкту	Аналіз технологій, вибір стеку, обґрунтування вибору	Документація з вибору технологій
Створення документації	Розробка документації для команди	Написання технічної документації	Технічна документація

Третій етап – розробка, він є найбільш трудомістким та поділяється на дві паралельні гілки: frontend та backend розробку. Frontend команда займається створенням інтуїтивно зрозумілого користувацького інтерфейсу, реалізацією таймера Pomodoro, системи сповіщень та забезпеченням адаптивності дизайну. Backend команда фокусується на розробці серверної частини, реалізації бізнес-логіки, створенні надійного API, налаштуванні бази даних та впровадженні системи безпеки.

Таблиця 3.5

Завдання та опис третього етапу

Завдання	Опис	Дії	Результати
Розробка frontend	Створення інтерфейсу користувача	Розробка макетів, реалізація інтерфейсу, інтеграція з API	Готовий інтерфейс користувача
Розробка backend	Реалізація серверної частини системи	Розробка бізнес-логіки, налаштування бази даних, реалізація API	Готовий сервер, API, база даних
Інтеграція компонентів	Об'єднання frontend та backend частин системи	Тестування інтеграції, виправлення помилок, оптимізація	Інтегрована система, готова до тестування

Четвертий етап – тестування, який забезпечує якість та надійність розробленого продукту. Проводиться комплексне тестування на різних рівнях: модульне тестування окремих компонентів, інтеграційне тестування для перевірки взаємодії між різними частинами системи, системне тестування для оцінки роботи додатку в цілому. Особлива увага приділяється тестуванню продуктивності та безпеки. Залучаються реальні користувачі для проведення користувацького тестування, що дозволяє виявити проблеми з точки зору кінцевого споживача.

Таблиця 3.6

Завдання та опис четвертого етапу

Завдання	Опис	Дії	Результати
Модульне тестування	Перевірка окремих компонентів системи	Написання тестів, виконання тестування, виправлення помилок	Звіт про модульне тестування, виявлені помилки
Інтеграційне тестування	Перевірка взаємодії між компонентами	Проведення тестів інтеграції, аналіз результатів	Звіт про інтеграційне тестування
Користувацьке тестування	Тестування системи реальними користувачами	Організація тестування, збір зворотного зв'язку, аналіз результатів	Звіт про користувацьке тестування, рекомендації

П'ятий етап - розгортання та підтримка, тобто введення системи в експлуатацію та її подальше функціонування. Підготовлюється серверна інфраструктура, налаштовується процес автоматичного розгортання (CI/CD),

система моніторингу та збору метрик. Організовується технічна підтримка користувачів, проводиться аналіз зворотного зв'язку для подальшого вдосконалення продукту. Важливим аспектом є забезпечення безперебійної роботи системи та оперативне реагування на виникаючі проблеми.

Таблиця 3.7

Завдання та опис п'ятого етапу

Завдання	Опис	Дії	Результати
Розгортання	Введення системи в експлуатацію.	Налаштування серверної інфраструктури, розгортання додатку.	Запущена система, доступна для користувачів.
Технічна підтримка	Забезпечення підтримки користувачів та системи.	Визначення каналів підтримки, навчання команди підтримки.	Налагоджена технічна підтримка, документація.
Збір зворотного зв'язку	Аналіз відгуків користувачів для покращення системи.	Проведення опитувань, аналіз даних, формування рекомендацій.	Звіт про зворотний зв'язок, план покращень.

На рисунку 3.1 зображено дані етапи у вигляді WBS діаграми:

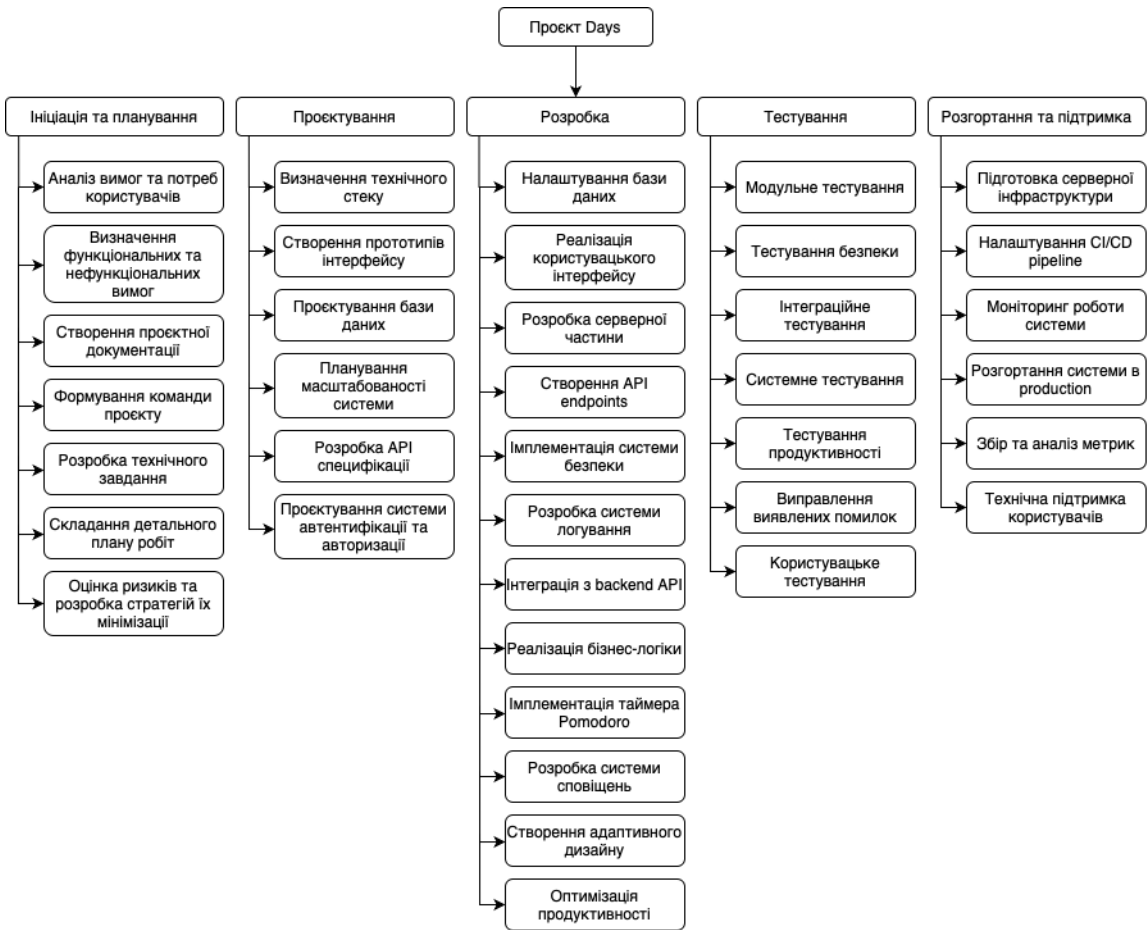


Рисунок 3.1. WBS діаграма робіт

Кожен з цих етапів вимагає ретельного планування, координації дій команди та постійного контролю якості. Успішна реалізація всіх етапів забезпечує створення якісного продукту, який відповідає потребам користувачів та технічним вимогам. Важливо підтримувати гнучкість у процесі розробки, щоб мати можливість адаптуватися до змін вимог та нових викликів, які можуть виникнути під час реалізації проєкту.

Зобразимо на рисунку 3.2 визначені етапи та розклад їх виконання за допомогою діаграми Ганта:

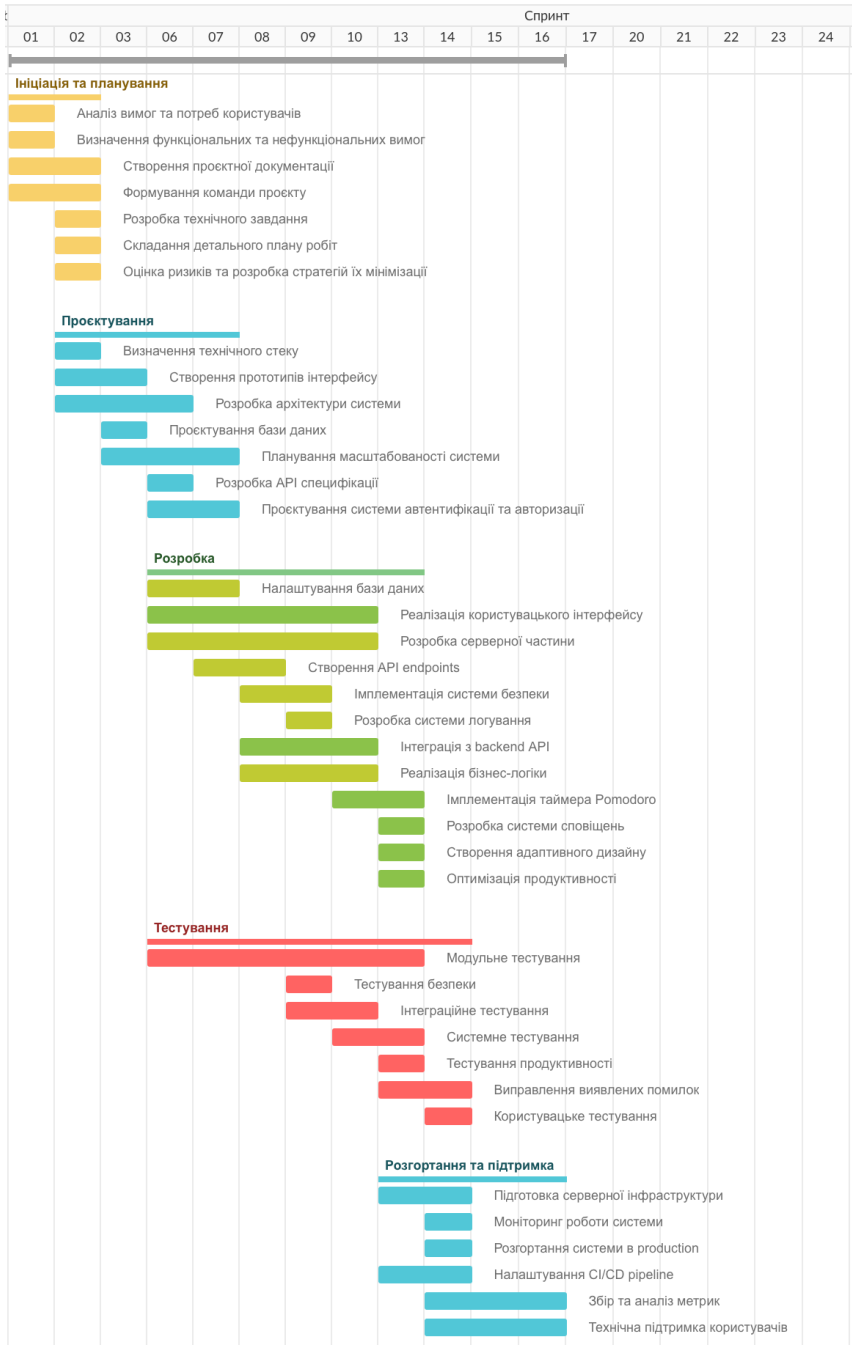


Рисунок 3.2. Діаграма Ганта

Варто зазначити, що діаграма розрахована на те, що спринт буде тривати 2 тижні, тобто це значення однієї горизонтальні поділки діаграми. З цього ми можемо зробити висновки, що на реалізацію основної частини проєкту знадобиться 32 тижні. Перший етап ініціалізації та планування займе чотири тижні. На другому тижні паралельно почнеться етап проєктування, який буде тривати вісім тижнів, і на його половині вже можна буде починати роботи над третім етапом розробки. Даний етап в свою чергу буде потребувати паралельного тестування, адже вважається гарною практикою покривати функціонал тестами одразу під час розробки. Ці етапи будуть найдовшими та тривати дванадцять та чотирнадцять тижнів відповідно, оскільки деякі складові тестування вимагають повної готовності розробки. Етап розгортання та підтримка почнеться вже на двадцять шостому тижні, щоб вчасно підготуватися і почати розгортання розробленої системи. Зображений порядок робіт є найбільш оптимальним варіантом, адже деякі складові етапів будуть виконуватися паралельно, де не потрібно чекати завершення попереднього етапу, що надасть можливість зекономити час та ресурси.

Додаткові компоненти проєкту

Таблиця 3.8

Додаткові етапи проєкту

Етап	Складові
Документація	- Технічна документація - Користувацька документація - API документація - Інструкції з розгортання - Документація з обслуговування
Навчання та підтримка	- Створення навчальних матеріалів - Проведення тренінгів для користувачів - Організація технічної підтримки - Збір та обробка зворотного зв'язку
Маркетинг та просування	- Розробка маркетингової стратегії - Створення промо-матеріалів - SEO-оптимізація - Налаштування аналітики - Проведення маркетингових кампаній

Документація є критично важливим компонентом проєкту, що забезпечує його довгострокову підтримку та розвиток. Технічна документація детально описує архітектуру системи, використані технології та принципи роботи кожного компонента. Користувацька документація надає вичерпні інструкції щодо використання додатку, включаючи покрокові керівництва та відповіді на поширені запитання. API документація містить детальний опис всіх доступних endpoints, методів їх використання та форматів даних. Інструкції з розгортання описують процес встановлення та налаштування системи в різних середовищах. Документація з обслуговування містить інформацію про моніторинг, діагностику та вирішення можливих проблем.

Навчання та підтримка користувачів є ключовим фактором успішного впровадження системи. Створюються різноманітні навчальні матеріали, включаючи відеоуроки, інтерактивні посібники та довідкові матеріали. Проводяться онлайн та офлайн тренінги для користувачів різного рівня підготовки. Організовується система технічної підтримки з можливістю звернення через різні канали комунікації. Налагоджується процес збору та аналізу зворотного зв'язку від користувачів для постійного вдосконалення системи та покращення користувацького досвіду.

Маркетинг та просування продукту вимагають комплексного підходу. Розробляється детальна маркетингова стратегія з урахуванням цільової аудиторії та конкурентного середовища. Створюються якісні промо-матеріали, включаючи презентації, відеоролики та інфографіку. Проводиться SEO-оптимізація для покращення видимості продукту в пошукових системах. Налаштовується система аналітики для відстеження ефективності маркетингових заходів та поведінки користувачів. Організуються цільові маркетингові кампанії з використанням різних каналів комунікації, включаючи соціальні мережі, email-розсилки та контекстну рекламу.

Особлива увага приділяється інтеграції всіх додаткових компонентів з основним процесом розробки. Документація постійно оновлюється відповідно до змін у системі, навчальні матеріали адаптуються під нові функції та

можливості, а маркетингові заходи координуються з релізами нових версій продукту. Важливим аспектом є забезпечення якості всіх додаткових матеріалів та їх відповідність загальній концепції продукту.

Для ефективного управління додатковими компонентами призначаються відповідальні особи та встановлюються чіткі процеси їх розробки та оновлення. Регулярно проводиться аудит всіх матеріалів для забезпечення їх актуальності та відповідності потребам користувачів. Створюється система метрик для оцінки ефективності кожного компонента та його впливу на загальний успіх проєкту.

Контроль та управління

Таблиця 3.9

Етапи контролю та управління

Етап	Складові
Проектний менеджмент	- Координація команди - Контроль виконання завдань - Управління ризиками - Комунікація зі стейкхолдерами - Звітність про прогрес
Якість та оптимізація	- Контроль якості коду - Оптимізація продуктивності - Аудит безпеки - Аналіз метрик - Впровадження покращень

Контроль та управління проєктом вимагає комплексного підходу до організації всіх процесів розробки та впровадження програмного забезпечення. Проектний менеджмент починається з детального планування та розподілу ресурсів, встановлення чітких цілей та термінів виконання. Координація команди здійснюється через регулярні зустрічі, використання систем управління завданнями та забезпечення ефективної комунікації між усіма учасниками проєкту.

Контроль виконання завдань базується на використанні гнучких методологій розробки, що дозволяють оперативно відслідковувати прогрес та вносити необхідні корективи. Впроваджується система щоденних stand-up зустрічей, планування спринтів та регулярні ретроспективи для аналізу

ефективності роботи команди. Особлива увага приділяється дотриманню встановлених дедлайнів та якості виконання завдань.

Управління ризиками передбачає систематичний аналіз потенційних загроз та розробку стратегій їх мінімізації. Створюється реєстр ризиків з оцінкою їх ймовірності та потенційного впливу на проєкт. Розробляються плани реагування на різні сценарії розвитку подій та впроваджуються превентивні заходи для зниження ризиків.

Комунікація зі стейкхолдерами здійснюється на регулярній основі через різні канали зв'язку. Проводяться регулярні презентації прогресу, надаються детальні звіти про стан проєкту та плануються особисті зустрічі для обговорення важливих питань. Забезпечується прозорість всіх процесів та оперативне інформування про будь-які зміни чи проблеми.

Контроль якості коду реалізується через впровадження чітких стандартів кодування, проведення code review та автоматизованого тестування. Використовуються інструменти статичного аналізу коду та метрики якості. Регулярно проводяться аудити кодової бази для виявлення потенційних проблем та технічного боргу.

Оптимізація продуктивності системи здійснюється на основі регулярного моніторингу та аналізу показників роботи. Проводяться навантажувальні тестування, профілювання коду та оптимізація запитів до бази даних. Впроваджуються кращі практики та патерни для покращення швидкодії та масштабованості системи.

Аудит безпеки включає регулярні перевірки на вразливості, тестування на проникнення та аналіз захищеності даних. Впроваджуються сучасні практики безпечного програмування та проводиться навчання команди з питань інформаційної безпеки. Створюється система моніторингу та реагування на інциденти безпеки.

Аналіз метрик базується на зборі та обробці різноманітних показників роботи системи та команди. Відслідковуються ключові показники ефективності (KPI), метрики якості коду, продуктивності системи та задоволеності

користувачів. На основі аналізу метрик приймаються рішення щодо необхідних покращень та оптимізацій.

Впровадження покращень здійснюється на основі постійного аналізу зворотного зв'язку від користувачів, результатів тестування та метрик системи. Розробляються та реалізуються плани оптимізації, впроваджуються нові технології та методології для підвищення ефективності розробки та якості продукту.

Така детальна декомпозиція робіт є ключовим інструментом успішного управління проєктом, оскільки вона забезпечує чітке розуміння повного обсягу необхідних робіт та дозволяє ефективно розподілити відповідальність між членами команди. Завдяки структурованому підходу стає можливим оптимальне планування ресурсів та часу, що в свою чергу забезпечує безперервний контроль за прогресом виконання завдань. Це також створює можливість для своєчасного виявлення потенційних проблем та їх оперативного вирішення, що в кінцевому результаті гарантує якісну реалізацію всіх компонентів системи відповідно до встановлених вимог та стандартів.

3.3. Реєстр ризиків

Реєстр ризиків є важливим інструментом управління проєктом, який допомагає ідентифікувати, оцінити та розробити стратегії реагування на потенційні загрози успішній реалізації проєкту. Для проєкту веб-додатку для управління часом на основі методу Pomodoro було виявлено наступні категорії ризиків: технічні, управлінські, зовнішні та операційні. Розглянемо кожні з них більш детально у вигляді таблиць:

Таблиця 3.10

Технічні ризики

Ризик	Ймовірність	Вплив	Стратегія реагування
Проблеми з масштабованістю системи	Середня	Високий	Проведення навантажувального тестування, використання хмарних сервісів з автоматичним масштабуванням

Збої в синхронізації даних між користувачами	Висока	Високий	Впровадження механізмів відновлення даних, резервне копіювання
Проблеми з безпекою даних	Середня	Критичний	Регулярний аудит безпеки, впровадження шифрування даних
Несумісність з різними браузерами	Низька	Середній	Крос-браузерне тестування, використання стандартних веб-технологій

У процесі аналізу технічних ризиків проєкту було виявлено кілька ключових загроз.

Проблеми з масштабованістю системи представляють собою ризик середньої ймовірності з високим рівнем впливу. Для мінімізації цього ризику планується проведення комплексного навантажувального тестування та впровадження хмарних сервісів, що підтримують автоматичне масштабування.

Особливу увагу слід приділити збоям у синхронізації даних між користувачами, оскільки цей ризик має високу ймовірність виникнення та високий рівень впливу. Для управління цим ризиком передбачається впровадження надійних механізмів відновлення даних та системи резервного копіювання.

Проблеми з безпекою даних характеризуються середньою ймовірністю, але критичним рівнем впливу. Для захисту від цього ризику заплановано проведення регулярних аудитів безпеки та впровадження сучасних методів шифрування даних.

Несумісність з різними браузерами представляє найменшу загрозу з низькою ймовірністю та середнім впливом. Для запобігання цьому ризику буде проводитися систематичне крос-браузерне тестування та використовуватимуться стандартні веб-технології, що забезпечують максимальну сумісність.

Управлінські ризики

Ризик	Ймовірність	Вплив	Стратегія реагування
Недотримання термінів розробки	Висока	Високий	Детальне планування, регулярний моніторинг прогресу
Зміна вимог під час розробки	Висока	Середній	Гнучка методологія розробки, резервування часу на зміни
Конфлікти в команді	Низька	Середній	Чітке розподілення ролей, регулярні командні зустрічі
Недостатня кваліфікація персоналу	Середня	Високий	Навчання персоналу, залучення експертів

При аналізі управлінських ризиків проєкту було виявлено чотири основні категорії загроз.

Найбільш серйозним викликом є недотримання термінів розробки, що характеризується високою ймовірністю та високим рівнем впливу. Для управління цим ризиком передбачається впровадження детального планування та забезпечення регулярного моніторингу прогресу виконання робіт.

Другим значним ризиком є можливість зміни вимог під час розробки, що має високу ймовірність виникнення, але середній рівень впливу. Для мінімізації наслідків планується використання гнучкої методології розробки та завчасне резервування часу на внесення потенційних змін.

Щодо конфліктів у команді, цей ризик має низьку ймовірність виникнення та середній рівень впливу. Для запобігання конфліктним ситуаціям буде впроваджено чітке розподілення ролей та обов'язків, а також проведення регулярних командних зустрічей.

Недостатня кваліфікація персоналу представляє ризик середньої ймовірності з високим рівнем впливу. Для подолання цього ризику планується організація навчання персоналу та залучення профільних експертів для підвищення загального рівня компетенції команди.

Зовнішні ризики

Ризик	Ймовірність	Вплив	Стратегія реагування
Поява сильних конкурентів	Середня	Високий	Моніторинг ринку, фокус на унікальних перевагах продукту
Зміни в технологічних трендах	Середня	Середній	Гнучка архітектура, можливість адаптації до нових технологій
Економічна нестабільність	Середня	Високий	Диверсифікація джерел фінансування, створення фінансового резерву
Зміни в законодавстві	Низька	Високий	Моніторинг законодавства, консультації з юристами

Аналіз зовнішніх ризиків проєкту виявив чотири ключові загрози, що потребують особливої уваги та розробки відповідних стратегій реагування.

Першим значущим ризиком є поява сильних конкурентів, що характеризується середньою ймовірністю виникнення, але високим рівнем впливу на проєкт. Для мінімізації цього ризику запропоновано стратегію постійного моніторингу ринку та фокусування на розвитку унікальних конкурентних переваг продукту. Це дозволить своєчасно реагувати на зміни конкурентного середовища та зберігати сильні позиції на ринку.

Другим ризиком визначено можливі зміни в технологічних трендах, що мають середню ймовірність виникнення та середній рівень впливу. Для управління цим ризиком передбачається розробка гнучкої архітектури продукту, яка забезпечить можливість швидкої адаптації до нових технологічних рішень та тенденцій ринку.

Третім суттєвим ризиком є економічна нестабільність, що характеризується середньою ймовірністю виникнення та високим рівнем впливу на проєкт. Стратегія реагування на цей ризик включає диверсифікацію джерел фінансування та створення достатнього фінансового резерву, що забезпечить стабільність проєкту навіть у складних економічних умовах.

Четвертим ризиком визначено можливі зміни в законодавстві, які мають низьку ймовірність виникнення, але потенційно високий вплив на проєкт. Для управління цим ризиком запропоновано стратегію регулярного моніторингу

законодавчих змін та проведення консультацій з юристами, що дозволить своєчасно адаптувати діяльність проєкту до нових правових вимог.

Всі виявлені ризики, незважаючи на різну ймовірність виникнення, мають значний потенційний вплив на успішність проєкту, тому вимагають систематичного контролю та своєчасного впровадження розроблених стратегій реагування. Запропоновані стратегії спрямовані на мінімізацію негативного впливу ризиків та забезпечення стабільного розвитку проєкту.

Таблиця 3.13

Операційні ризики

Ризик	Ймовірність	Вплив	Стратегія реагування
Проблеми з інтеграцією компонентів	Середня	Високий	Модульне тестування, чітка документація API
Втрата критичних даних	Низька	Критичний	Регулярне резервне копіювання, розподілене зберігання даних
Перевищення бюджету	Висока	Високий	Детальне планування бюджету, регулярний фінансовий моніторинг
Проблеми з продуктивністю системи	Середня	Високий	Оптимізація коду, моніторинг продуктивності

У таблиці 3.13 представлено аналіз операційних ризиків проєкту та відповідні стратегії реагування на них.

Першим ризиком є проблеми з інтеграцією компонентів, що має середню ймовірність виникнення та високий рівень впливу на проєкт. Для мінімізації цього ризику пропонується впровадити модульне тестування та забезпечити чітку документацію API.

Другим ризиком визначено втрату критичних даних. Хоча ймовірність такої події низька, її вплив оцінюється як критичний. Для запобігання цьому ризику планується впровадити систему регулярного резервного копіювання та забезпечити розподілене зберігання даних.

Перевищення бюджету є третім ризиком з високою ймовірністю виникнення та високим рівнем впливу. Для управління цим ризиком передбачається детальне планування бюджету та впровадження системи регулярного фінансового моніторингу.

Останнім ризиком є проблеми з продуктивністю системи, що має середню ймовірність виникнення та високий вплив. Для подолання цього ризику пропонується здійснювати оптимізацію коду та забезпечити постійний моніторинг продуктивності системи.

Для забезпечення ефективного управління ризиками в проєкті розробки веб-додатку для управління часом на основі методу Pomodoro впроваджується комплексна система заходів, яка складається з трьох основних напрямків: превентивні заходи, реактивні заходи, а також моніторинг та контроль.

Превентивні заходи спрямовані на попередження виникнення ризикових ситуацій та мінімізацію їх потенційного впливу. Регулярний моніторинг та оцінка ризиків здійснюються щотижня командою проєкту, що дозволяє своєчасно виявляти нові загрози та оцінювати ефективність існуючих заходів безпеки. Створення резервних копій даних відбувається автоматично кожні 6 годин з використанням розподілених систем зберігання даних, що забезпечує надійний захист від втрати інформації. Система контролю якості включає автоматизоване тестування коду, код-рев'ю та регулярні аудити якості продукту. Проведення регулярних аудитів безпеки здійснюється як внутрішніми фахівцями, так і залученими експертами з кібербезпеки.

Реактивні заходи забезпечують швидке та ефективне реагування на виникнення ризикових ситуацій. План аварійного відновлення детально описує процедури відновлення роботи системи у разі збоїв різного рівня складності, включаючи покрокові інструкції та розподіл відповідальності між членами команди. Процедури швидкого реагування на інциденти передбачають чітку послідовність дій при виникненні проблем з безпекою, продуктивністю або доступністю системи. Механізми масштабування ресурсів базуються на використанні хмарних технологій з автоматичним масштабуванням залежно від навантаження. Альтернативні сценарії розвитку проєкту розроблені для різних ситуацій, включаючи зміни ринкових умов або технологічних вимог.

Система моніторингу та контролю забезпечує постійний нагляд за ефективністю управління ризиками. Щотижневий огляд статусу ризиків

проводиться на спеціальних нарадах команди проєкту, де обговорюються поточні загрози та ефективність впроваджених заходів. Оновлення реєстру ризиків відбувається на основі нової інформації та змін у проєкті, що дозволяє підтримувати актуальність оцінки ризиків. Аналіз ефективності впроваджених заходів проводиться щомісяця з використанням кількісних та якісних показників. Коригування стратегій реагування здійснюється на основі результатів аналізу та змін у проєктному середовищі.

Особлива увага приділяється навчанню персоналу та підвищенню обізнаності щодо ризиків. Регулярно проводяться тренінги та воркшопи, де команда проєкту вивчає нові методи управління ризиками та практикує реагування на різні сценарії надзвичайних ситуацій. Це дозволяє підтримувати високий рівень готовності до потенційних загроз та забезпечувати ефективне виконання всіх запланованих заходів з управління ризиками.

Важливим аспектом є також документування всіх процесів управління ризиками та створення бази знань, яка містить інформацію про попередні інциденти, вжиті заходи та їх результативність. Це дозволяє використовувати накопичений досвід для покращення системи управління ризиками та запобігання повторенню проблемних ситуацій у майбутньому.

Висновки до третього розділу

У третьому розділі було детально розглянуто та проаналізовано ключові аспекти планування проєкту розробки веб-додатку для управління часом на основі методу Pomodoro. Було визначено та класифіковано основні групи стейкхолдерів проєкту, включаючи внутрішніх (команда розробки, керівництво проєкту) та зовнішніх (кінцеві користувачі, інвестори, конкуренти) зацікавлених сторін, а також розроблено стратегії взаємодії з кожною групою.

Проведено детальну декомпозицію робіт (WBS), яка включає п'ять основних етапів проєкту: ініціація та планування, проєктування, розробка, тестування, розгортання та підтримка. Для кожного етапу визначено конкретні

завдання та очікувані результати. Додатково виділено важливі компоненти проєкту, такі як документація, навчання та підтримка, маркетинг та просування.

Особливу увагу приділено системі контролю та управління проєктом, яка включає проєктний менеджмент, забезпечення якості та оптимізацію. Розроблено комплексний реєстр ризиків, що охоплює технічні, управлінські, зовнішні та операційні ризики. Для кожного типу ризиків визначено ймовірність виникнення, потенційний вплив та розроблено відповідні стратегії реагування.

Запропоновано комплексну систему управління ризиками, яка базується на трьох основних напрямках: превентивні заходи, реактивні заходи та система моніторингу і контролю. Особливий акцент зроблено на важливості навчання персоналу та документування процесів управління ризиками для створення ефективної бази знань проєкту.

Розроблений план проєкту забезпечує чітке розуміння обсягу необхідних робіт, дозволяє ефективно розподілити ресурси та встановити контроль над процесом розробки, що є критично важливим для успішної реалізації проєкту та досягнення поставлених цілей.

РОЗДІЛ 4. РОЗРОБКА БІЗНЕС-МОДЕЛІ ПРОДУКТУ

1.1. Канва бізнес-моделі продукту

Створення канви бізнес-моделі є критично важливим етапом у розробці успішного продукту, оскільки вона дозволяє систематично проаналізувати та структурувати всі ключові аспекти бізнесу. Для нашого веб-додатку з управління часом на основі методу Pomodoro, канва допомагає чітко визначити та узгодити всі компоненти бізнес-моделі, забезпечуючи цілісне бачення проєкту.

Представимо канву бізнес-моделі проєкту на рисунку 4.1:

Business Model Canvas		Designed for: Days	Designed by: Dmytro Moskvyn	Date: 07/12/2024	Version: 0.1
Key Partners - IT-компанії та стартапи. - Навчальні заклади. - Постачальники хмарних послуг. - Консалтингові компанії.	Key Activities - Розробка та підтримка веб-додатку. - Покращення командних процесів. - Запровадження організованості. Key Resources - Команда розробників. - Серверна інфраструктура. - Інтелектуальна власність. - База знань. - Бренд та репутація.	Value Propositions - Комплексне рішення для підвищення продуктивності. - Інтегрований підхід до управління часом. - Фокус на командній роботі.	Customer Relationships - Зворотній зв'язок та регулярні опитування в додатку. - Технічна підтримка користувачів у вигляді чат-боту або гарячої лінії. Channels - Власний веб-сайт. - Професійні соціальні мережі. - IT-конференції. - Партнерські програми.	Customer Segments - Команди, які прагнуть покращити організованість та ефективність взаємодії між членами команди. - Працівники, яким необхідно структурувати свій робочий день. - Користувачі, які потребують інструменти для ефективного управління часом.	
Cost Structure - Розробка та підтримка продукту. - Маркетингові активності. - Серверна інфраструктура. - Витрати на персонал.			Revenue Streams - Підписка на преміум-функції. - Корпоративні ліцензії. - Консалтинг та навчання.		

Рисунок 4.1. Канва бізнес-моделі проєкту

Ключовими партнерами проєкту виступають IT-компанії та стартапи, які можуть інтегрувати наш продукт у свої процеси, навчальні заклади для популяризації методології, та постачальники хмарних послуг для забезпечення надійної інфраструктури. Особливу роль відіграють партнерства з консалтинговими компаніями, які можуть рекомендувати наш продукт своїм клієнтам.

Пропозиція цінності нашого продукту полягає у комплексному рішенні для підвищення продуктивності команд через ефективне управління часом. Унікальність пропозиції забезпечується поєднанням методу Pomodoro з інструментами для командної роботи, що особливо актуально для розподілених команд та віддаленої роботи.

Для доставки цінності клієнтам будуть використовуватися різноманітні канали, включаючи власний веб-сайт, професійні соціальні мережі, IT-конференції та партнерські програми. Важливим аспектом буде створення сильної присутності в професійних спільнотах та активне залучення лідерів думок у сфері управління проєктами.

Ключові ресурси проєкту включають команду розробників, серверну інфраструктуру, інтелектуальну власність та базу знань методології. Особливу увагу приділено формуванню сильного бренду та репутації, що є критичним для довгострокового успіху продукту.

Структура витрат складається з витрат на розробку та підтримку продукту, маркетингові активності, серверну інфраструктуру та персонал. Важливим аспектом є оптимізація витрат через використання сучасних технологій та ефективне управління ресурсами.

Джерела прибутку формуються через модель підписки на преміум-функції, корпоративні ліцензії та додаткові послуги з консалтингу та навчання. Диверсифікація потоків доходу забезпечує стабільність та масштабованість бізнес-моделі.

Ключовими метриками успіху визначено кількість активних користувачів, рівень утримання клієнтів, середній дохід на користувача та рівень задоволеності продуктом. Ці метрики дозволяють оцінювати ефективність бізнес-моделі та приймати обґрунтовані рішення щодо розвитку продукту.

Значимість переваг нашого продукту підтверджується зростаючою потребою в ефективних інструментах для управління часом та продуктивністю в умовах сучасного бізнес-середовища. Особлива увага приділяється створенню

сильних взаємних зв'язків між різними компонентами продукту, що забезпечує синергетичний ефект для користувачів.

Пропозиція стосовно цінності для клієнта базується на глибокому розумінні потреб цільової аудиторії та постійному вдосконаленні продукту відповідно до отриманого зворотного зв'язку. Це дозволяє створювати продукт, який не лише вирішує поточні проблеми користувачів, але й передбачає їхні майбутні потреби.

Життєвий цикл продукту включає в себе кілька ключових етапів, починаючи від розробки та впровадження до зрілості та можливого оновлення. На початковому етапі фокус буде зроблений на створенні мінімально життєздатного продукту (MVP) з базовим функціоналом для управління часом за методом Pomodoro та простими інструментами для командної роботи.

Після запуску MVP планується активний збір зворотного зв'язку від перших користувачів та швидке впровадження покращень на основі отриманих даних. Це дозволить адаптувати продукт під реальні потреби цільової аудиторії та підвищити його цінність для користувачів.

На етапі зростання основна увага буде приділена розширенню функціоналу, покращенню користувацького досвіду та масштабуванню інфраструктури. Планується додавання розширених можливостей для командної роботи, інтеграцій з популярними інструментами для управління проектами та розробка мобільних додатків.

Етап зрілості продукту характеризуватиметься стабілізацією користувацької бази та фокусом на утриманні існуючих клієнтів через постійні покращення та оновлення. Важливим аспектом буде підтримка високого рівня надійності системи та якості обслуговування користувачів.

Для підтримки конкурентоспроможності продукту планується регулярний аналіз ринку та трендів, що дозволить вчасно впроваджувати інновації та адаптувати продукт до змінних потреб користувачів. Це може включати інтеграцію нових технологій, розширення функціоналу та вихід на нові сегменти ринку.

1.2. Модель ціноутворення додатку

Створення моделі ціноутворення додатку є критично важливим для успішної бізнес-стратегії, оскільки забезпечує фінансову стабільність і передбачуваний потік доходів. Це дозволяє планувати розвиток продукту, інвестувати в його вдосконалення та ефективно сегментувати користувачів, пропонуючи різні варіанти використання для різних цільових груп. Чітка модель також оптимізує витрати на розробку та підтримку, допомагаючи краще планувати ресурси та пріоритети.

Крім того, ціноутворення визначає ринкове позиціонування продукту, формуючи сприйняття його цінності користувачами. Вона є ключовим елементом маркетингової стратегії, впливаючи на просування продукту та залучення нових клієнтів. Правильна модель ціноутворення також сприяє довгостроковому розвитку бізнесу, залучаючи інвестиції та формуючи лояльну користувацьку базу. Таким чином, ефективна модель ціноутворення є стратегічним рішенням, що впливає на всі аспекти розвитку додатку.

Наш веб-додаток для ефективного управління часом позиціонує себе як інструмент, що допомагає користувачам оптимізувати їхній час та підвищити продуктивність. Він орієнтований на широкий спектр користувачів, від індивідуальних користувачів, які прагнуть покращити свої навички тайм-менеджменту, до малих і великих команд, зокрема працюючих за методологією Scrum, які потребують розширених функцій для спільної роботи. Завдяки різноманітним тарифним планам, включаючи безкоштовний доступ до базових функцій, додаток забезпечує можливість ознайомлення з продуктом без початкових витрат, що робить його доступним для всіх.

Базові тарифні плани є важливим елементом моделі ціноутворення, оскільки вони забезпечують доступ до основних функцій продукту безкоштовно або за невелику плату. Це дозволяє залучити широку аудиторію користувачів, які можуть бути зацікавлені в оптимізації свого часу, але не готові інвестувати значні кошти на початковому етапі. Базові плани створюють можливість для

користувачів ознайомитися з продуктом, оцінити його цінність і, в разі задоволення, перейти на більш просунуті тарифи.

Таблиця 4.1

Базові тарифні плани

Тарифний план	Плата за місяць	Функціонал
Free	Безкоштовний	- Базовий функціонал Pomodoro-таймера - Особистий трекінг часу - Базова статистика
Pro	\$9.99	- Усі функції Free-версії - Розширена можливості нотаток - Календар мітингів - Командні функції до 5 учасників - Пріоритетна підтримка
Business	\$19.99	- Усі функції Pro-версії - Необмежена кількість учасників команди - API доступ - Персональний менеджер - Можливість брендингу

Тарифний план «Free» є безкоштовним варіантом, який надає користувачам можливість ознайомитися з основними функціями веб-додатку. У рамках цього плану доступні базовий функціонал Pomodoro-таймера, особистий трекінг часу, та можливість додавати прості нотатки. Це дозволяє користувачам тестувати основні можливості програми без жодних витрат.

Тарифний план «Pro» коштує \$9.99 на місяць і розширює можливості Free-версії, пропонуючи додаткові функції для більш серйозних користувачів. Розширена можливості нотаток дозволяють структурувати записи та отримувати більш наглядну картину по виконаній та запланованій роботі, а календар мітингів допомагає організовувати графік робочого дня. Крім того, план включає командні функції до 5 учасників, що буде задовольняти потреби маленьких команд та спонукати більш серйозні організації переходити на дорожчий план, та пріоритетну підтримку, що є важливим для користувачів.

Тарифний план «Business» призначений для великих команд і організацій, які потребують розширених функцій, і коштує \$19.99 на місяць. Цей план включає всі функції Pro-версії, а також дозволяє залучати необмежену кількість

учасників команди. Доступ до API дозволяє інтегрувати додаток з іншими системами, а персональний менеджер допомагає з налаштуваннями та оптимізацією використання продукту. Можливість брендингу дозволяє компаніям налаштовувати інтерфейс додатку під свій бренд, що підвищує впізнаваність та лояльність клієнтів.

Ці тарифні плани забезпечують гнучкість для користувачів з різними потребами, від індивідуальних до корпоративних, що робить ваш веб-додаток привабливим для широкої аудиторії.

У контексті розробки веб-додатку для ефективного управління часом, додаткові опції монетизації можуть суттєво підвищити його цінність та привабливість для користувачів. Наприклад, впровадження підписки на преміум-функції, такі як розширені аналітичні звіти, інтеграція з іншими сервісами або персоналізація, дозволить користувачам отримувати більше користі від додатку. Крім того, можливість пропонувати партнерські програми може не лише збільшити доходи, але й створити спільноту користувачів, які прагнуть покращити свої навички управління часом. Це, в свою чергу, сприятиме зростанню лояльності та залученню нових клієнтів.

Таблиця 4.2

Додаткові опції монетизації

Опція	Переваги
Спеціальні пропозиції	- Річна передплата зі знижкою 20% - Знижки для освітніх закладів (30%) - Спеціальні тарифи для некомерційних організацій
Преміум функції	- Додаткові шаблони звітів (+\$2.99/місяць) - Розширені інтеграції (+\$4.99/місяць) - Кастомні віджети (+\$1.99/місяць)

Розглянуті додаткові опції монетизації веб-додатку можуть суттєво підвищити його привабливість для різних категорій користувачів. Серед спеціальних пропозицій виділяється річна передплата зі знижкою 20%, а також знижки для освітніх закладів у розмірі 30% і спеціальні тарифи для некомерційних організацій, що робить продукт доступнішим для цих груп. Крім того, пропонуються преміум функції, які включають додаткові шаблони звітів за

\$2.99 на місяць, розширені інтеграції за \$4.99 на місяць та кастомні віджети з додатковою аналітикою та інформацією за \$1.99 на місяць. Ці опції дозволяють користувачам налаштовувати додаток відповідно до своїх потреб, що може сприяти підвищенню їхньої задоволеності та лояльності.

Також для визначення цін на свої продукти або послуги з метою максимізації прибутку, залучення клієнтів або утримання конкурентоспроможності варто сформувати стратегію ціноутворення.

Таблиця 4.3

Стратегія ціноутворення

Етап	Опис
Проникнення на ринок	- Запуск з конкурентними цінами - Акційні пропозиції для ранніх користувачів
Гнучкість	- Можливість щомісячної або річної оплати - Модульний підхід до додаткових функцій
Масштабування	- Поступове підвищення цін з розширенням функціоналу - Введення нових тарифних планів за потреби
Утримання клієнтів	- Програма лояльності - Спеціальні умови для довгострокових клієнтів

Етап проникнення на ринок, передбачає запуск продукту з конкурентними цінами, що дозволяє залучити увагу споживачів. Додатково, акційні пропозиції для ранніх користувачів стимулюють попит і сприяють швидкому формуванню бази клієнтів. Другий етап, «Гнучкість», акцентує увагу на можливості вибору між щомісячною або річною оплатою, що підвищує зручність для споживачів. Модульний підхід до додаткових функцій дозволяє клієнтам адаптувати продукт під свої потреби. На етапі «Масштабування» ми можемо поступово підвищувати ціни в міру розширення функціоналу та вводити нові тарифні плани відповідно до запитів ринку. Завершує стратегію етап «Утримання клієнтів», який включає програми лояльності та спеціальні умови для довгострокових клієнтів, що сприяє збереженню клієнтської бази та підвищенню задоволеності споживачів.

Розроблена модель ціноутворення додатку є критично важливою для забезпечення фінансової стабільності та залучення користувачів. Чітка структура тарифних планів дозволяє ефективно сегментувати аудиторію, пропонуючи безкоштовний доступ до базових функцій, що підвищує ймовірність переходу на платні тарифи. Додаткові опції монетизації та стратегії утримання клієнтів сприяють довгостроковому розвитку бізнесу, формуючи лояльну базу користувачів і оптимізуючи витрати на розробку. Таким чином, ефективна модель ціноутворення є стратегічним активом для успіху додатку.

1.3. План фінансування/інвестування

План фінансування та інвестування проєкту веб-додатку для ефективного управління часом базується на поетапному підході з чітким розподілом фінансових ресурсів на різних стадіях розробки та впровадження продукту.

Планується, що штат буде складатися з 17 працівників, з яких 12 розробників, 3 тестувальників, 1 Product Owner та 1 HR. Представимо розрахунки у вигляді таблиці:

Таблиця 4.4

Витрати на заробітню платню

Посада	Кількість	Ставка,\$	Заробітна плата (місяць), \$	Нарахування на оплату праці (місяць), \$	Усього на місяць, \$	Усього на рік, \$
Senior Developer	2	3 500	7 000	1 540	8 540	102 480
Middle Developer	6	2 000	12 000	2 640	14 640	175 680
Junior Developer	4	800	3 200	704	3 904	46 848
Senior QA	1	2 700	2 700	594	3 294	39 528
Middle QA	1	1 400	1 400	308	1 708	20 496
Junior QA	1	500	500	110	610	7 320
Product Owner	1	4 000	4 000	838	4 838	58 056
HR	1	2 300	2 300	506	2 806	33 672
Всього	17	17200	33 100	7 240	40 340	484 080

Розрахунок приведено з урахуванням курсу орієнтовно 42 гривні за долар США.

Отже, загальні витрати на оплату праці та нарахування на заробітну плату за рік становитимуть 484 080 \$. При обрахуванні витрат на оплату праці по ставці Product Owner приблизно враховано максимально допустимі розміри єдиного соціального внеску. Крім витрат на оплату праці та нарахування на оплату праці. Додамо ще витрати на оплату послуг маркетингу, сума яких за рік буде становити приблизно 30 000\$, оренду офісу приблизно 20 000\$ на рік, ліцензійне забезпечення та дрібні витрати, які будуть становити приблизно 5000\$, тобто ми маємо врахувати ще видатки у розмірі 55 000\$ на існування та розвиток компанії та її продукту. Загальні витрати компанії будуть становити приблизно 539 100\$ на рік.

Прибутки від додатку будуть надходити з реклами у безкоштовній ліцензії та платних підписок. Очікується, що реклама буде приносити 15 000\$ на рік, та продажі підписок будуть складати 1000 Business вартістю 240\$ за пакет та 5000 Pro пакетів вартістю 120\$ за один, що становитиме прибуток у розмірі 240 000\$ та 600 000\$ відповідно. Тобто прибуток має складати 855 000\$ на рік, а чистий прибуток очікується у розмірі 315 900\$ на рік.

Варто зазначити, що перший час продукт буде збитковий, адже потрібен деякий час, щоб сформувати аудиторію користувачів і вийти на бажані числа продажів ліцензій, тому на початковому етапі планується залучення seed-інвестицій у розмірі \$250 000 для створення мінімально життєздатного продукту (MVP). Ці кошти будуть спрямовані на формування core-команди розробників, дизайнерів та менеджерів проєкту, а також на базову технічну інфраструктуру.

Другий етап фінансування передбачає залучення венчурних інвестицій серії А у розмірі \$200 000-300 000 для масштабування продукту. Ці кошти будуть використані для розширення функціоналу додатку, маркетингових активностей та збільшення команди розробки. Особлива увага приділятиметься розвитку продажів та формуванню стабільної користувацької бази. Планується досягнення точки беззбитковості протягом 18-24 місяців після запуску продукту.

Довгострокова стратегія фінансування передбачає можливість залучення додаткових інвестицій серії В для міжнародної експансії та розробки нових

продуктових ліній. Прогнозований обсяг такого раунду може становити \$500 000-1 000 000 залежно від досягнутих показників росту та ринкової ситуації.

Висновки до четвертого розділу

У четвертому розділі було проведено комплексний аналіз та розробку бізнес-моделі веб-додатку для ефективного управління часом. Було створено детальну канву бізнес-моделі, яка охоплює всі ключові аспекти проєкту, включаючи ціннісну пропозицію, ключових партнерів, ресурси, канали збуту та структуру витрат і доходів. Було розроблено стратегію життєвого циклу продукту, починаючи від створення MVP до етапів зростання та зрілості. Особлива увага приділена постійному вдосконаленню продукту на основі зворотного зв'язку від користувачів та адаптації до змінних потреб ринку.

Створено гнучку модель ціноутворення з трьома базовими тарифними планами (Free, Pro, Business), що дозволяє ефективно сегментувати користувачів та забезпечити стабільний потік доходів. Розроблено додаткові опції монетизації та стратегію ціноутворення, що враховує різні етапи розвитку продукту.

Сформовано детальний план фінансування та інвестування проєкту, який передбачає поетапне залучення коштів від seed-інвестицій до можливого раунду серії В. Визначено структуру операційних витрат та прогнозовані терміни досягнення точки беззбитковості.

Таким чином, розроблена бізнес-модель створює міцну основу для успішної реалізації проєкту, забезпечуючи його фінансову стійкість та потенціал для масштабування. Запропоновані рішення враховують специфіку ринку, потреби цільової аудиторії та сучасні тенденції в сфері розробки програмного забезпечення.

ВИСНОВКИ

У результаті проведеного дослідження було досягнуто основну мету - розроблено проєкт веб-додатку для ефективного управління часом на основі методу Pomodoro, який дозволяє покращити продуктивність як індивідуальної, так і командної роботи.

У першому розділі було проведено комплексне дослідження теоретичних та практичних аспектів управління часом, зокрема детально проаналізовано методологію Pomodoro, її основні принципи, переваги та обмеження при застосуванні в командній роботі. Здійснено порівняльний аналіз існуючих програмних рішень для тайм-менеджменту, визначено їх функціональні можливості, сильні та слабкі сторони, а також виявлено незаповнені ніші на ринку подібних рішень. Досліджено специфіку командної роботи в ІТ-проєктах, включаючи особливості розподілу та синхронізації робочого часу, типові проблеми з фокусуванням та продуктивністю. На основі проведеного аналізу сформульовано вимоги до сучасних систем управління часом та зроблено висновок про необхідність розробки спеціалізованого рішення, яке б поєднувало принципи методу Pomodoro з можливостями командної взаємодії. Визначено, що ключовими вимогами до такого рішення є зручність і простота використання та синхронізація між членами команди. Встановлено, що впровадження такого рішення може значно підвищити ефективність роботи ІТ-команд через покращення координації, оптимізацію розподілу робочого часу та зменшення кількості відволікань.

У другому розділі було проведено комплексний аналіз та проєктування веб-додатку для ефективного управління часом. Спочатку було детально описано цільову аудиторію продукту – команду розробників програмного забезпечення, що працює за методологією Scrum. Проаналізовано їх основні процеси, проблеми та потреби для кращого розуміння вимог до функціоналу додатку. Було створено детальні моделі взаємодії користувачів через розробку персон Senior Full-Stack розробника та Project Manager. Для кожної персони було

складено карти подорожей, емоційні профілі та проведено Jobs-to-be-Done аналіз, що допомогло виявити ключові сценарії використання продукту. Сформульовано чіткі функціональні та технічні вимоги до додатку, включаючи управління фокусуванням, синхронізацію мітингів та статусів доступності між членами команди, нотатки з можливістю відмітити прогрес, planning poker та інші критичні функції. Визначено технологічний стек та системні вимоги. Розроблено детальні макети основних сторінок додатку – головної сторінки з таймером, сторінки списку справ та сторінки планування. Створено карти навігації, що описують покрокові сценарії взаємодії користувачів з системою.

У третьому розділі було проведено комплексну роботу з планування проєкту ІТ-продукту, яка включала детальний аналіз та класифікацію стейкхолдерів проєкту, включаючи команду розробників, менеджмент проєкту, кінцевих користувачів, інвесторів та конкурентів. Було розроблено структуровану декомпозицію робіт (WBS) з п'ятьма основними етапами: ініціація та планування, проєктування, розробка, тестування, впровадження та підтримка. Також було створено реєстр ризиків, який охоплював технічні, управлінські, зовнішні та операційні ризики проєкту. На основі проведеної роботи було зроблено важливі висновки щодо необхідності ефективної комунікації зі стейкхолдерами, важливості детального планування робіт, критичної ролі управління ризиками та цінності чітких проєктних контролів. Цей розділ заклав міцний фундамент для успішної реалізації проєкту через комплексний підхід до планування та управління ризиками.

У четвертому розділі було проведено комплексний аналіз та розробку бізнес-моделі веб-додатку для ефективного управління часом. В рамках роботи створено детальну канву бізнес-моделі, яка охоплює всі ключові аспекти проєкту, включаючи ціннісну пропозицію, ключових партнерів, ресурси, канали збуту, структуру витрат і доходів. Розроблено стратегію життєвого циклу продукту від створення MVP до етапів зростання та зрілості, з особливим акцентом на постійне вдосконалення на основі зворотного зв'язку користувачів. Створено гнучку модель ціноутворення з трьома базовими тарифними планами

(Free, Pro, Business), що дозволяє ефективно сегментувати користувачів та забезпечити стабільний потік доходів. Також сформовано детальний план фінансування та інвестування проєкту з визначенням структури операційних витрат та прогнозованих термінів досягнення точки беззбитковості. Розроблена бізнес-модель створює міцну основу для успішної реалізації проєкту, забезпечуючи його фінансову стійкість та потенціал для масштабування, враховуючи специфіку ринку, потреби цільової аудиторії та сучасні тенденції в сфері розробки програмного забезпечення.

Практична цінність отриманих результатів полягає в тому, що розроблений веб-додаток дозволить командам ефективніше управляти часом, підвищити продуктивність та покращити командну взаємодію. Особлива увага приділена адаптації методу Pomodoro для командної роботи, що робить продукт унікальним на ринку.

Подальші напрямки дослідження та вдосконалення програмного комплексу включають розширення функціоналу для підтримки різних методологій управління часом, інтеграцію з популярними інструментами для управління проєктами, розробку мобільної та нативної десктопної версії додатку, впровадження елементів штучного інтелекту для персоналізації рекомендацій та розширення аналітичних можливостей системи.

Таким чином, розроблений проєкт створює міцну основу для реалізації ефективного інструменту управління часом, що відповідає сучасним потребам команд розробки програмного забезпечення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Eisenhower Matrix: Prioritize Your Time on What Matters Most [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://slab.com/blog/eisenhower-matrix/>.
2. Stanford U. Improving Productivity [Електронний ресурс] / University Stanford – Режим доступу до ресурсу: <https://studentlearning.stanford.edu/improving-productivity>.
3. Agile Practice Guide [Електронний ресурс] // Project Management Institute, Inc. – 2017. – Режим доступу до ресурсу: <https://yourdigitalaid.com/wp-content/uploads/2021/02/Agile-Study-Guide.pdf>.
4. World Quality Report 2021-22 [Електронний ресурс] // Capgemini, Sogeti and Micro Focus. – 2021. – Режим доступу до ресурсу: <https://www.sogeti.com/newsroom/world-quality-report-2021-22/>.
5. Cyber-Security Threats and Challenges with Third Party Banking Partners. // International Journal of Engineering and Management Research. – 2021. – №5. – С. 223.
6. Test Driven Development: By Example / Kent Beck. – Addison-Wesley Professional, 2002. – 240 с.
7. Backlog refinement guide [Електронний ресурс] // Max Rehkopf – Режим доступу до ресурсу: <https://www.atlassian.com/agile/scrum/backlog-refinement>.
8. ISO/IEC 29119. Software Testing Standards. [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:tr:29119:-13:ed-1:v1:en>.
9. Модель життєвого циклу програмного проєкту [Електронний ресурс] // Кафедра ММСА | - КПП ім. Ігоря Сікорського – Режим доступу до ресурсу: http://mmsa.kpi.ua/sites/default/files/disciplines/Розробка%20і%20тестування%20програм/didkovska_m_v_testing_lecture_1.pdf.

10. Project Planning, Scheduling, and Control, Sixth Edition: The Ultimate Hands-On Guide to Bringing Projects in On Time and On Budget / Джеймс Льюїс. - McGraw-Hill, 2023. – 528 с.
11. 9 Proven Time Management Techniques and Tools [Електронний ресурс] // University of St. Augustine for Health Sciences. – 2022. – Режим доступу до ресурсу: <https://www.usa.edu/blog/time-management-techniques/>.
12. The Pomodoro Technique — Why It Works & How To Do It [Електронний ресурс] – Режим доступу до ресурсу: <https://todoist.com/productivity-methods/pomodoro-technique>.
13. A guide to the Pomodoro technique and how to use it [Електронний ресурс] // Indeed Editorial Team. – 2024. – Режим доступу до ресурсу: <https://uk.indeed.com/career-advice/career-development/pomodoro-technique>.
14. The Rise and Fall of Getting Things Done [Електронний ресурс] // Cal Newport. – 2020. – Режим доступу до ресурсу: <https://ref.feval.ca/202011280456-gtd--rise-and-fall-of-gtd.pdf>.
15. Getting Things Done: - an examination of David Allen's work-life system. / Daniel Karlsson. – Göteborg, Sweden, 2006. – 42 с.
16. The Eisenhower Matrix: How to prioritize your to-do list [Електронний ресурс] // Team Asana. – 2024. – Режим доступу до ресурсу: <https://asana.com/resources/eisenhower-matrix>.
17. The Cost of Poor Software Quality in the US: A 2020 Report [Електронний ресурс] // CISQ. – 2021. – Режим доступу до ресурсу: <https://www.it-cisq.org/cisq-files/pdf/CPSQ-2020-report.pdf>.
18. The Eisenhower Matrix: Avoid the "Urgency Trap" with Dwight D. Eisenhower's famous prioritization framework [Електронний ресурс] – Режим доступу до ресурсу: <https://todoist.com/productivity-methods/eisenhower-matrix>.
19. Основи управління ІТ проєктами [Електронний ресурс] // КПІ ім. Ігоря Сікорського. – 2019. – Режим доступу до ресурсу: <https://ela.kpi.ua/server/api/core/bitstreams/7c313e5c-5477-4be2-9806-d32e9eace0c3/content>.

20. Fixing the Software Requirements Mess [Електронний ресурс] // Christopher Lindquist. – 2005. – Режим доступу до ресурсу: <https://www.cio.com/article/255253/developer-fixing-the-software-requirements-mess.html>.
21. Software Survival in 2024: 2023 Statistics and Quality Assurance [Електронний ресурс] // Beta Breakers. – 2024. – Режим доступу до ресурсу: <https://www.betabreakers.com/software-survival-in-2024-understanding-2023-project-failure-statistics-and-the-role-of-quality-assurance/>.
22. Information Technology Project Management / Кеті Швальбе. – Cengage Learning, 2018. – 672 с.
23. Cybercrime costs to hit \$10.5 trn by 2025: How insurance may save your biz [Електронний ресурс] // Surbhi Gloria Singh. – 2024. – Режим доступу до ресурсу: https://www.business-standard.com/finance/personal-finance/cybercrime-costs-to-hit-10-5-trn-by-2025-how-insurance-may-save-your-biz-124072400476_1.html#goog_rewarded.
24. How much do bugs cost to fix during each phase of the SDLC? [Електронний ресурс] // Arvinder Saini. – 2017. – Режим доступу до ресурсу: <https://www.blackduck.com/blog/cost-to-fix-bugs-during-each-sdlc-phase.html>.
25. Learning Agile: Understanding Scrum, XP, Lean, and Kanban / Andrew Stellman, Jennifer Greene. – O'Reilly Media, Inc., 2014. – 420 с.
26. The Empirical Reality of IT Project Cost Overruns: Discovering A Power-Law Distribution [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tandfonline.com/doi/full/10.1080/07421222.2022.2096544#d1e262>.
27. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 13th Edition/ Harold Kerzner. – Wiley. John Wiley & Sons, LTD, 2022. – 880 с.
28. Agile Product and Project Management: A Step-by-Step Guide to Building the Right Products Right / Марія Брейтер. – APress, 2022. – 349 с.

29. CI/CD Pipeline with Docker and Jenkins: Learn How to Build and Manage Your CI/CD Pipelines Effectively / Sandeep Rawat. – BPB Publications, 2023. – 237 с.
30. Average Human Attention Span Statistics & Facts [2024] [Электронный ресурс] – 2024. – Режим доступа до ресурсу: <https://www.sambarecovery.com/rehab-blog/average-human-attention-span-statistics>.
31. How We Improve Concentration with Scientifically Designed Music [Электронный ресурс] – Режим доступа до ресурсу: <https://www.focusatwill.com/app/pages/how-it-works>.
32. Focus sessions and periodic breaks: a personal exploration of the pomodoro technique [Электронный ресурс] – 2020. – Режим доступа до ресурсу: <https://writineering.humspace.ucla.edu/student-work/the-effect-of-focus-sessions-and-periodic-breaks-on-studying-a-personal-exploration-of-the-pomodoro-technique/>.
33. How to Make Remote Work [Электронный ресурс] – 2019. – Режим доступа до ресурсу: <https://www.inc.com/shama-hyder/how-to-make-remote-work-work.html>.
34. Dr. Kat Cohen Shares Leadership Strategies To Improve Your Company's Culture [Электронный ресурс] / Krish Chopra – 2018. – Режим доступа до ресурсу: <https://medium.com/authority-magazine/dr-kat-cohen-shares-leadership-strategies-to-improve-your-companys-culture-842614e79778>.
35. How To Reclaim Your Attention And Creativity At The Office Amid Digital Distractions [Электронный ресурс] / Danielle Sabrina – 2019. – Режим доступа до ресурсу: <https://www.forbes.com/councils/forbesagencycouncil/2019/05/24/how-to-reclaim-your-attention-and-creativity-at-the-office-amid-distractions/#9cc2fbc42b86>.
36. Lawrence M. Effective Time and Self Management, Environment and Productivity in an Organization / M. Lawrence, M. Rombe. // Science Journal of Business and Management. – 2016. – №4.
37. Agile продукт-менеджмент за допомогою Scrum / Роман Піхлер. – Фабула, 2019. – 128 с.