

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

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Гнучке управління створенням програмного забезпечення для автоматизації бізнес-процесів у великих корпораціях»

Ступінь вищої освіти – магістр

Спеціальність – 073 «Менеджмент»

Освітня програма «Agile-технології розробки програмного забезпечення»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Керівник: старший викладач
Олег ЛУКУТИН

Керівник: викладач, к.ф-м.н.,
доцент
Віра ТКАЧЕНКО

Виконав: здобувач
групи МЕН/Agile-23м-в
Ігор ТЕСЛЕНКО

Київ, 2024 р.

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

ЗАТВЕРДЖУЮ:
завідувач кафедри інформаційного
менеджменту, математики та
статистики
_____ Денис БАЛДИК
«__» ____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
ТЕСЛЕНКО ІГОР ВАЛЕРІЙОВИЧ

Тема роботи	Гнучке управління створенням програмного забезпечення для автоматизації бізнес-процесів у великих корпораціях
Номер та дата наказу про затвердження теми	№ 56-3 від 27 червня 2024 р.
Коротка постановка завдання	Розробка та впровадження ефективного підходу до гнучкого управління створенням програмного забезпечення для автоматизації бізнес-процесів у великих корпораціях з використанням методології Scrum
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	What is scrum master? // The Home of Scrum! – URL: https://www.scrum.org/resources/what-is-a-scrum-master Debugging teams with the Lencioni Model // Addy Osmani Engineering leader and senior thinker – URL: https://addyosmani.com/blog/debugging-teams-lencioni/
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має містити теоретичне та/або практичне дослідження за темою роботи, яку слід розглядати як складне спеціалізоване завдання або практичну проблематику в галузі управління та адміністрування, яка характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів Agile технологій.

Дата видачі завдання «14» липня 2024 р.

Керівник

Олег ЛУКУТІН

Керівник

Віра ТКАЧЕНКО

Здобувач

Ігор ТЕСЛЕНКО

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напрямку дослідження та керівника	01.07.2024 р.	Виконано
2	Формування теми та призначення керівника	08.07.2024 р.	Виконано
3	Затвердження теми кваліфікаційної роботи	09.07.2024 р.	Виконано
4	Затвердження завдання на кваліфікаційну роботу	15.07.2024 р.	Виконано
Основний етап			
5	Розробка концепції кваліфікаційної роботи	22.07.2024 р.	Виконано
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів.	29.07.2024 р.	Виконано
7	Затвердження розширеної постановки завдання. Підготовка та подання керівнику розділу 1 кваліфікаційної роботи	18.09.2024 р.	Виконано
8	Проектування інформаційної системи. Підготовка та подання керівнику розділу 2 кваліфікаційної роботи	18.09.2024 р.	Виконано
9	Реалізація інформаційної системи. Підготовка та подання керівнику розділу 3 кваліфікаційної роботи	25.09.2024 р.	Виконано
10	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	01.10.2024 р.	Виконано
11	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівнику доопрацьованого варіанту кваліфікаційної роботи	04.10.2024 р.	Виконано
Завершальний етап			
12	Представлення рукопису для перевірки на плагіат	07.10.2024 р.	Виконано
13	Підготовка презентації та доповіді на передзахист	07.10.2024 р.	Виконано
14	Передзахист кваліфікаційної роботи	08-11.10.2024 р.	Виконано
15	Технічна самоекспертиза роботи на відповідність вимогам до оформлення та виправлення недоліків	08-11.10.2024 р.	Виконано
16	Експертиза роботи керівником та зовнішнім експертом	14.10.2024 р.	Виконано
17	Доопрацювання доповіді та презентації для захисту	18.10.2024 р.	Виконано
18	Захист кваліфікаційної роботи	21-25.10.2024 р.	Виконано

Керівник

Олег ЛУКУТІН

Керівник

Віра ТКАЧЕНКО

Здобувач

Ігор ТЕСЛЕНКО

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. ДИЗАЙН БІЗНЕСУ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Опис предметної області автоматизації бізнес-процесів у великих корпораціях	8
1.1.1 Аналіз факторів макросередовища	8
1.1.2 Аналіз факторів мікросередовища	9
1.2 Постановка цілей та завдань проєкту	10
1.2.1 Визначення проблеми та потреби в автоматизації	10
1.2.2 Формулювання цілей проєкту	12
1.2.3 Розробка завдань проєкту	14
1.3 Визначення вимог до програмного забезпечення.....	14
1.3.1 Функціональні вимоги.....	14
1.3.2 Нефункціональні вимоги.....	15
1.3.3 Технічні вимоги.....	15
Висновки до розділу 1	17
РОЗДІЛ 2. ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ.....	19
2.1 Обґрунтування вибору фреймворку Scrum для управління проєктом	19
2.1.1 Порівняння Scrum з іншими Agile-методологіями.....	19
2.1.2 Переваги Scrum для даного проєкту	20
2.2 Планування проєкту з розробки програмного забезпечення	21
2.2.1 Формування Scrum-команди.....	21
2.2.2 Розробка Product Backlog	23
2.2.3 Планування спринтів	25
2.2.4 Оцінка часу та ресурсів	26
2.3 Моніторинг виконання проєкту.....	28
2.3.1 Організація щоденних Scrum-зустрічей	28
2.3.2 Відстеження прогресу за допомогою Burndown Chart.....	29

2.3.3 Проведення Sprint Review та Sprint Retrospective.....	30
Висновки до розділу 2	31
РОЗДІЛ 3. РЕЗУЛЬТАТИ РОБОТИ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КОМАНДИ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	33
3.1 Огляд інкременту продукту	33
3.1.1 Опис реалізованої функціональності	33
3.1.2 Аналіз відповідності вимогам.....	33
3.2 Ретроспектива роботи команди	34
3.2.1 Аналіз ефективності Scrum-процесів.....	34
3.2.2 Виявлення проблем та шляхів їх вирішення.....	35
3.3 Сучасний інструментарій Scrum-майстра в agile-середовищі.....	35
3.3.1 Огляд програмних засобів для управління Agile-проєктами	35
3.3.2 Методи підвищення продуктивності Scrum-команди.....	36
3.4 Оцінка економічної ефективності впровадження програмного забезпечення	37
3.5 Аналіз ризиків та стратегії їх мінімізації.....	38
3.6 Оцінка якості та тестування програмного забезпечення	39
3.7 Перспективи розвитку та масштабування проєкту	41
Висновки до розділу 3	43
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	47
ДОДАТКИ.....	50

ВСТУП

Актуальність теми: В умовах стрімкого розвитку технологій та зростаючої конкуренції, великі корпорації стикаються з необхідністю автоматизації своїх бізнес-процесів для підвищення ефективності та зменшення витрат. Створення програмного забезпечення для такої автоматизації є складним завданням, яке вимагає гнучкого підходу до управління проектами.

Мета дослідження: Розробка та впровадження ефективного підходу до гнучкого управління створенням програмного забезпечення для автоматизації бізнес-процесів у великих корпораціях з використанням методології Scrum.

Завдання дослідження:

1. Проаналізувати предметну область автоматизації бізнес-процесів у великих корпораціях.
2. Визначити вимоги до програмного забезпечення для автоматизації бізнес-процесів.
3. Обґрунтувати вибір фреймворку Scrum для управління проектом.
4. Розробити план проекту з створення програмного забезпечення.
5. Впровадити систему моніторингу виконання проекту.
6. Оцінити результати роботи та розробити рекомендації щодо підвищення ефективності команди.

Об'єкт дослідження: Процес створення програмного забезпечення для автоматизації бізнес-процесів у великих корпораціях.

Предмет дослідження: Методи та інструменти гнучкого управління проектами з розробки програмного забезпечення з використанням методології Scrum.

Методи дослідження: Аналіз літератури, системний аналіз, моделювання бізнес-процесів, методи проектного менеджменту, емпіричні методи (спостереження, опитування).

Практичне значення: Результати дослідження можуть бути використані для підвищення ефективності управління проектами з розробки програмного забезпечення в великих корпораціях.

РОЗДІЛ 1. ДИЗАЙН БІЗНЕСУ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області автоматизації бізнес-процесів у великих корпораціях

1.1.1 Аналіз факторів макросередовища

У цьому підрозділі проведемо PEST-аналіз для виявлення ключових факторів макросередовища, що впливають на автоматизацію бізнес-процесів у великих корпораціях.

Політичні фактори:

- Законодавство у сфері захисту даних (напр., GDPR в ЄС, CCPA в Каліфорнії)
- Державні програми підтримки цифровізації бізнесу (напр., Індустрія 4.0 в Німеччині)
- Політика кібербезпеки на державному рівні
- Міжнародні торгові угоди та їх вплив на глобальні корпорації
- Політична стабільність у країнах, де розташовані офіси корпорацій

Економічні фактори:

- Глобальна економічна ситуація та її вплив на ІТ-бюджети корпорацій
- Курс валют (важливо для міжнародних проектів)
- Рівень інфляції та його вплив на вартість розробки ПЗ
- Доступність кваліфікованої робочої сили в ІТ-сфері
- Тенденції аутсорсингу та їх вплив на розробку ПЗ

Соціальні фактори:

- Зміна культури праці (віддалена робота, гнучкий графік)
- Підвищення цифрової грамотності населення
- Зростання попиту на персоналізовані послуги
- Демографічні зміни та їх вплив на робочу силу
- Зростаюча важливість корпоративної соціальної відповідальності

Технологічні фактори:

- Розвиток хмарних технологій
- Поширення технологій штучного інтелекту та машинного навчання
- Зростання важливості кібербезпеки
- Розвиток технологій блокчейн та їх потенційне застосування в бізнес-процесах
- Інтернет речей (IoT) та його вплив на автоматизацію процесів

1.1.2 Аналіз факторів мікросередовища

Проведемо аналіз п'яти сил Портера для оцінки конкурентного середовища в галузі розробки ПЗ для автоматизації бізнес-процесів.

1. Конкуренція в галузі:

- Високий рівень конкуренції між великими ІТ-компаніями (напр., SAP, Oracle, Microsoft) та інноваційними стартапами
- Постійна необхідність інновацій для збереження конкурентоспроможності
- Ціновий тиск через наявність безкоштовних та open-source рішень
- Важливість репутації та досвіду в реалізації великих проєктів

2. Загроза появи нових конкурентів:

- Низькі бар'єри входу для стартапів завдяки доступності технологій та хмарних платформ
- Високі вимоги до експертизи та репутації для роботи з великими корпораціями
- Можливість швидкого масштабування завдяки глобальному ринку
- Складність створення комплексних рішень для автоматизації як бар'єр для нових гравців

3. Ринкова влада постачальників:

- Залежність від постачальників хмарних послуг (напр., AWS, Azure, Google Cloud)
- Вплив вартості ліцензій на загальну вартість розробки

- Важливість партнерських відносин з ключовими технологічними компаніями
- Залежність від спеціалізованих інструментів та платформ розробки

4. Ринкова влада покупців:

- Високі вимоги корпорацій до якості та функціональності ПЗ
- Тенденція до створення власних ІТ-відділів у великих корпораціях
- Можливість переходу на конкурентні рішення як фактор тиску на постачальників
- Важливість кастомізації та гнучкості рішень для задоволення специфічних потреб клієнтів

5. Загроза появи продуктів-замінників:

- Розвиток low-code та no-code платформ
- Поява готових галузевих рішень для автоматизації
- Розвиток технологій RPA (Robotic Process Automation) як альтернатива традиційній автоматизації
- Використання AI та машинного навчання для автоматизації завдань, які раніше вимагали складного ПЗ

На основі проведеного аналізу можна зробити висновок, що ринок розробки ПЗ для автоматизації бізнес-процесів є високо конкурентним та динамічним. Це підкреслює важливість використання гнучких методологій управління проектами, таких як Scrum, для швидкої адаптації до змін та ефективної розробки продукту.

1.2 Постановка цілей та завдань проєкту

1.2.1 Визначення проблеми та потреби в автоматизації

Великі корпорації стикаються з низкою проблем, які можуть бути вирішені за допомогою автоматизації бізнес-процесів:

1. Низька ефективність ручних операцій:

- Повторювані завдання займають багато часу співробітників

- Високий ризик помилок при ручному введенні даних
- Складність відстеження та аналізу ефективності процесів

2. Висока ймовірність людських помилок:

- Неточності в фінансових розрахунках та звітності
- Помилки при передачі інформації між відділами
- Невідповідності в документації та контрактах

3. Складність контролю та аналізу бізнес-процесів:

- Відсутність єдиної системи моніторингу процесів
- Складність отримання актуальної інформації в реальному часі
- Труднощі з виявленням "вузьких місць" у процесах

4. Повільна адаптація до змін на ринку:

- Довгий цикл впровадження нових продуктів або послуг
- Складність модифікації існуючих процесів
- Недостатня гнучкість у реагуванні на зміни попиту

5. Неefективне використання ресурсів компанії:

- Дублювання функцій між відділами
- Неоптимальне використання робочого часу співробітників
- Надмірні витрати на підтримку застарілих систем

Потреба в автоматизації виникає з необхідності вирішення цих проблем та досягнення таких цілей:

А. Підвищення продуктивності праці:

- Скорочення часу на виконання рутинних завдань
- Звільнення часу співробітників для більш важливих та творчих завдань
- Збільшення обсягу оброблюваної інформації без збільшення штату

В. Зменшення операційних витрат:

- Оптимізація використання ресурсів
- Скорочення витрат на обробку паперових документів
- Зменшення витрат на виправлення помилок

С. Покращення якості обслуговування клієнтів:

- Прискорення обробки запитів клієнтів
- Забезпечення точності та послідовності в комунікаціях
- Надання клієнтам доступу до самообслуговування через автоматизовані системи

D. Прискорення прийняття рішень на основі даних:

- Забезпечення доступу до актуальної інформації в реальному часі
- Автоматизація збору та аналізу даних для підтримки прийняття рішень
- Впровадження предиктивної аналітики для прогнозування тенденцій

E. Забезпечення масштабованості бізнесу:

- Створення гнучкої інфраструктури, здатної адаптуватися до зростання компанії
- Стандартизація процесів для легкого розширення на нові ринки або продукти
- Забезпечення інтеграції між різними підрозділами та географічними локаціями

1.2.2 Формулювання цілей проєкту

Основна мета проєкту: Розробка програмного забезпечення для автоматизації ключових бізнес-процесів великих корпорацій з використанням гнучкої методології Scrum.

Підділі проєкту:

1. Створення модульної архітектури ПЗ для легкої адаптації до різних бізнес-процесів:

- Розробка гнучкої системи, здатної інтегрувати різні модулі залежно від потреб конкретної корпорації
- Забезпечення можливості легкого додавання нових функцій без порушення існуючої функціональності
- Створення API для інтеграції з іншими системами та розробки додаткових модулів

2. Забезпечення інтеграції з існуючими корпоративними системами:

- Розробка коннекторів для популярних ERP, CRM та інших корпоративних систем
- Створення механізмів для безпечного обміну даними між різними системами
- Забезпечення сумісності з різними форматами даних та протоколами
 3. Впровадження аналітичних інструментів для моніторингу ефективності бізнес-процесів:
- Розробка дашбордів для візуалізації ключових показників ефективності (KPI)
- Створення системи сповіщень для своєчасного виявлення відхилень від нормальних показників
- Впровадження інструментів для аналізу тенденцій та прогнозування
 4. Розробка зручного користувацького інтерфейсу для різних рівнів користувачів:
- Створення інтуїтивно зрозумілого інтерфейсу для кінцевих користувачів
- Розробка розширених інструментів для адміністраторів системи
- Забезпечення адаптивності інтерфейсу для роботи на різних пристроях (десктоп, планшет, смартфон)
 5. Забезпечення високого рівня безпеки та захисту даних:
- Впровадження багаторівневої системи аутентифікації та авторизації
- Шифрування даних при зберіганні та передачі
- Розробка системи аудиту для відстеження всіх дій в системі
 6. Забезпечення масштабованості та високої продуктивності системи:
- Оптимізація архітектури для роботи з великими обсягами даних
- Впровадження механізмів балансування навантаження
- Забезпечення можливості горизонтального масштабування системи
 7. Розробка системи автоматизованого тестування та розгортання:
- Створення комплексного набору автоматизованих тестів
- Впровадження практик безперервної інтеграції та доставки (CI/CD)

- Розробка інструментів для автоматизованого розгортання в різних середовищах

1.2.3 Розробка завдань проєкту

Для досягнення поставлених цілей необхідно виконати наступні завдання:

1. Провести детальний аналіз бізнес-процесів цільових корпорацій
2. Розробити технічне завдання на створення ПЗ
3. Спроекувати архітектуру програмного забезпечення
4. Розробити прототип користувацького інтерфейсу
5. Реалізувати базову функціональність ПЗ
6. Провести інтеграційне тестування з існуючими корпоративними системами
7. Впровадити аналітичні модулі для моніторингу ефективності
8. Провести комплексне тестування безпеки
9. Підготувати документацію та навчальні матеріали для користувачів
10. Здійснити пілотне впровадження у вибраній корпорації

1.3 Визначення вимог до програмного забезпечення

1.3.1 Функціональні вимоги

1. Автоматизація ключових бізнес-процесів (напр., управління замовленнями, інвентаризація, фінансова звітність)
2. Можливість налаштування workflow для різних бізнес-процесів
3. Інтеграція з існуючими ERP, CRM та іншими корпоративними системами
4. Генерація аналітичних звітів та дашбордів
5. Управління доступом на основі ролей користувачів
6. Можливість масштабування для обробки великих обсягів даних

1.3.2 Нефункціональні вимоги

1. Висока продуктивність системи (час відгуку не більше 2 секунд для 95% запитів)
2. Доступність системи 99.9% часу
3. Підтримка одночасної роботи до 10 000 користувачів
4. Можливість горизонтального масштабування
5. Відповідність стандартам безпеки (напр., ISO 27001)
6. Локалізація інтерфейсу на кілька мов

1.3.3 Технічні вимоги

1. Розробка з використанням сучасного стеку технологій (напр., .NET Core, React, PostgreSQL)
2. Використання мікросервісної архітектури
3. Розгортання в хмарному середовищі (AWS або Azure)
4. Підтримка контейнеризації (Docker) для легкого розгортання
5. Використання CI/CD для автоматизації процесу розробки та розгортання
6. Підтримка API для інтеграції з іншими системами

1.4 Огляд аналогів

При розробці програмного забезпечення для автоматизації бізнес-процесів важливо враховувати існуючі рішення на ринку. Розглянемо основних гравців та їх продукти:

1.4.1 Корпоративні рішення

1. Microsoft:

- Dynamics 365: комплексне рішення для управління бізнес-процесами, що включає CRM та ERP функціональність.

- Power Platform: набір інструментів для створення власних бізнес-додатків та автоматизації процесів.

2. Oracle:

- Oracle E-Business Suite: повний набір інтегрованих глобальних бізнес-додатків.
- Oracle Cloud Applications: хмарне рішення для автоматизації фінансів, HR, ланцюжків поставок та інших бізнес-процесів.

3. SAP:

- SAP S/4HANA: інтелектуальна ERP-система для цифрової трансформації бізнесу.
- SAP Business ByDesign: хмарне рішення для середнього бізнесу.

1.4.2 Спеціалізовані додатки для комунікації та управління проектами

1. Slack:

- Платформа для корпоративного спілкування та співпраці.
- Інтеграція з багатьма бізнес-інструментами для автоматизації робочих процесів.

2. Discord:

- Спочатку орієнтований на геймерів, але все частіше використовується для бізнес-комунікацій.
- Підтримує відео-конференції та інтеграцію з іншими сервісами.

1.5 Аналіз стейкголдів

Для успішної реалізації проекту важливо враховувати інтереси та потреби всіх зацікавлених сторін:

1. Великі корпорації:

- Очікування: підвищення ефективності бізнес-процесів, скорочення витрат, покращення контролю.
- Вплив: основні замовники та користувачі системи.

2. Менеджери різних рівнів у корпораціях:

- Очікування: інструменти для ефективного управління та аналітики, автоматизація рутинних завдань.

- Вплив: ключові користувачі, які визначають вимоги та оцінюють ефективність системи.

3. Співробітники корпорацій:

- Очікування: зручні інструменти для щоденної роботи, зменшення рутинних завдань.

- Вплив: кінцеві користувачі системи, від яких залежить успішність впровадження.

4. IT-відділи корпорацій:

- Очікування: сумісність з існуючою інфраструктурою, безпека, масштабованість.

- Вплив: відповідальні за технічну інтеграцію та підтримку системи.

5. Партнери з інтеграції:

- Очікування: можливість розширення функціональності, документація для інтеграції.

- Вплив: розширюють можливості системи та адаптують її під специфічні потреби клієнтів.

Висновки до розділу 1

У першому розділі було проведено комплексний аналіз предметної області автоматизації бізнес-процесів у великих корпораціях та визначено ключові аспекти розробки програмного забезпечення для вирішення цього завдання.

1. Аналіз макро- та мікросередовища виявив ряд важливих факторів, які впливають на розробку ПЗ для автоматизації бізнес-процесів:

- Зростаючий вплив законодавства у сфері захисту даних та кібербезпеки

- Економічні виклики, що впливають на IT-бюджети корпорацій

- Зміни в культурі праці та зростання цифрової грамотності
- Розвиток технологій штучного інтелекту та хмарних обчислень
- Високий рівень конкуренції в галузі розробки ПЗ

2. Визначено ключові проблеми та потреби великих корпорацій в автоматизації, що включають підвищення ефективності операцій, зменшення людських помилок, покращення контролю та аналізу бізнес-процесів.

3. Сформульовано цілі та завдання проєкту, спрямовані на розробку гнучкого, масштабованого та безпечного програмного забезпечення для автоматизації бізнес-процесів з використанням методології Scrum.

4. Визначено комплекс функціональних, нефункціональних та технічних вимог до програмного забезпечення, що забезпечать його відповідність потребам великих корпорацій та сучасним технологічним стандартам.

Проведений аналіз та визначені вимоги створюють міцну основу для подальшої розробки проєкту з використанням гнучкої методології Scrum. Це дозволить ефективно управляти процесом створення програмного забезпечення, враховуючи складність та динамічність середовища великих корпорацій.

У наступному розділі буде детально розглянуто застосування методології Scrum для управління процесом розробки визначеного програмного забезпечення.

РОЗДІЛ 2. ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

2.1 Обґрунтування вибору фреймворку Scrum для управління проектом

2.1.1 Порівняння Scrum з іншими Agile-методологіями

Порівняння Scrum з іншими Agile-методологіями, показує основні характерні відмінності між методологіями.

Таблиця 2.1 – Критерії досягнення цілей проекту

Характеристика	Scrum	Kanban	Extreme Programming (XP)	Lean
Ітерації	Фіксовані спринти	Безперервний потік	Короткі ітерації	Безперервний потік
Ролі	Чітко визначені	Не визначені	Визначені, але менш формально	Не визначені
Зміни в ході ітерації	Не рекомендуються	Дозволені	Дозволені	Дозволені
Пріоритизація	Product Backlog	Kanban-дошка	План релізів	Усунення втрат
Метрики	Velocity, Burndown	Lead Time, Cycle Time	Недоліки, швидкість тестів	Ефективність, час циклу
Розмір команди	5-9 осіб	Не обмежено	5-9 осіб	Не обмежено
Найкраще підходить для	Складні проекти з невизначеними вимогами	Підтримка та операційні завдання	Технічно складні проекти	Оптимізація процесів

2.1.2 Переваги Scrum для даного проєкту

1. Гнучкість до змін вимог:

- Можливість адаптуватися до нових вимог корпоративних клієнтів
- Здатність швидко реагувати на зміни в технологічному ландшафті
- Легкість внесення змін до пріоритетів розробки

2. Швидка доставка цінності:

- Регулярні релізи працюючого програмного забезпечення
- Можливість отримати зворотній зв'язок від клієнтів на ранніх етапах
- Поступове нарощування функціональності продукту

3. Прозорість процесу:

- Чітке розуміння статусу проєкту всіма зацікавленими сторонами
- Регулярні демонстрації результатів роботи
- Відкрита комунікація в команді та з клієнтами

4. Постійне вдосконалення:

- Регулярні ретроспективи для аналізу та покращення процесів
- Можливість експериментувати з новими підходами
- Швидке виявлення та усунення проблем

5. Ефективна комунікація:

- Щоденні стендапи для синхронізації команди
- Пряма взаємодія між розробниками та представниками бізнесу
- Зменшення кількості документації на користь прямого спілкування

6. Зменшення ризиків:

- Раннє виявлення проблем завдяки частим демонстраціям
- Можливість швидко змінити напрямок розробки при необхідності
- Поступове нарощування складності, що зменшує ризик великих невдач

7. Підвищення якості продукту:

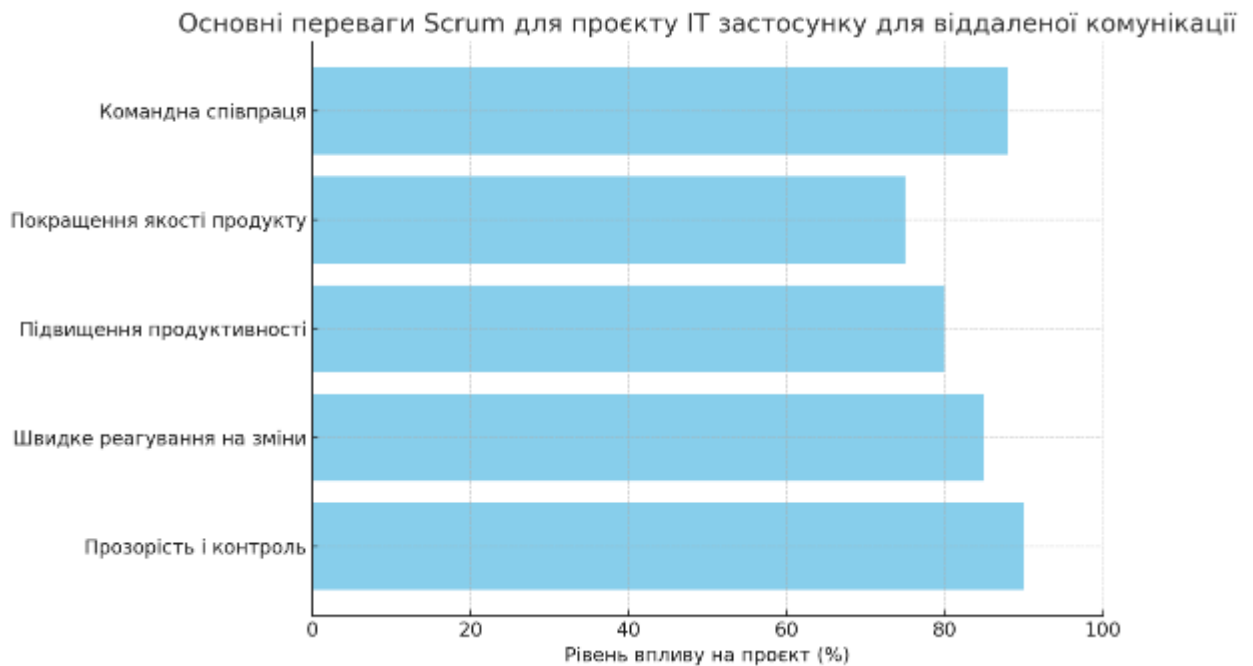
- Інтеграція тестування в процес розробки
- Постійний рефакторинг коду
- Фокус на створенні потенційно релізного продукту в кожному спринті

8. Підвищення мотивації команди:

- Самоорганізація команди, що підвищує відповідальність
- Чіткі цілі та видимий прогрес у кожному спринті
- Можливість для команди впливати на процес розробки

Для візуалізації переваг Scrum, зображена діаграма на рис. 2.1

Рисунок 2.1 – Переваги Scrum



2.2 Планування проекту з розробки програмного забезпечення

2.2.1 Формування Scrum-команди

1. Визначення ролей:

- Product Owner: відповідає за визначення вимог до продукту та пріоритизацію завдань
- Scrum Master: фасилітатор процесу, відповідає за дотримання принципів Scrum
- Development Team: команда розробників, що створює продукт

2. Критерії відбору членів команди:

- Технічні навички: досвід у розробці корпоративного ПЗ, знання відповідних технологій

- Soft skills: комунікабельність, здатність до самоорганізації, вміння працювати в команді
- Досвід роботи з Agile методологіями: перевага надається кандидатам з досвідом роботи в Scrum
- Розуміння предметної області: знання специфіки автоматизації бізнес-процесів у великих корпораціях

3. Структура команди:

- 1 Product Owner
- 1 Scrum Master
- 5-7 членів Development Team, включаючи:
 - Front-end розробники
 - Back-end розробники
 - QA інженери
 - DevOps спеціаліст

4. Процес формування команди:

- Визначення необхідних компетенцій
- Пошук кандидатів (внутрішній рекрутинг та зовнішній найм)
- Проведення інтерв'ю та тестових завдань
- Формування команди з урахуванням балансу навичок та особистостей
- Проведення командного тренінгу з принципів Scrum

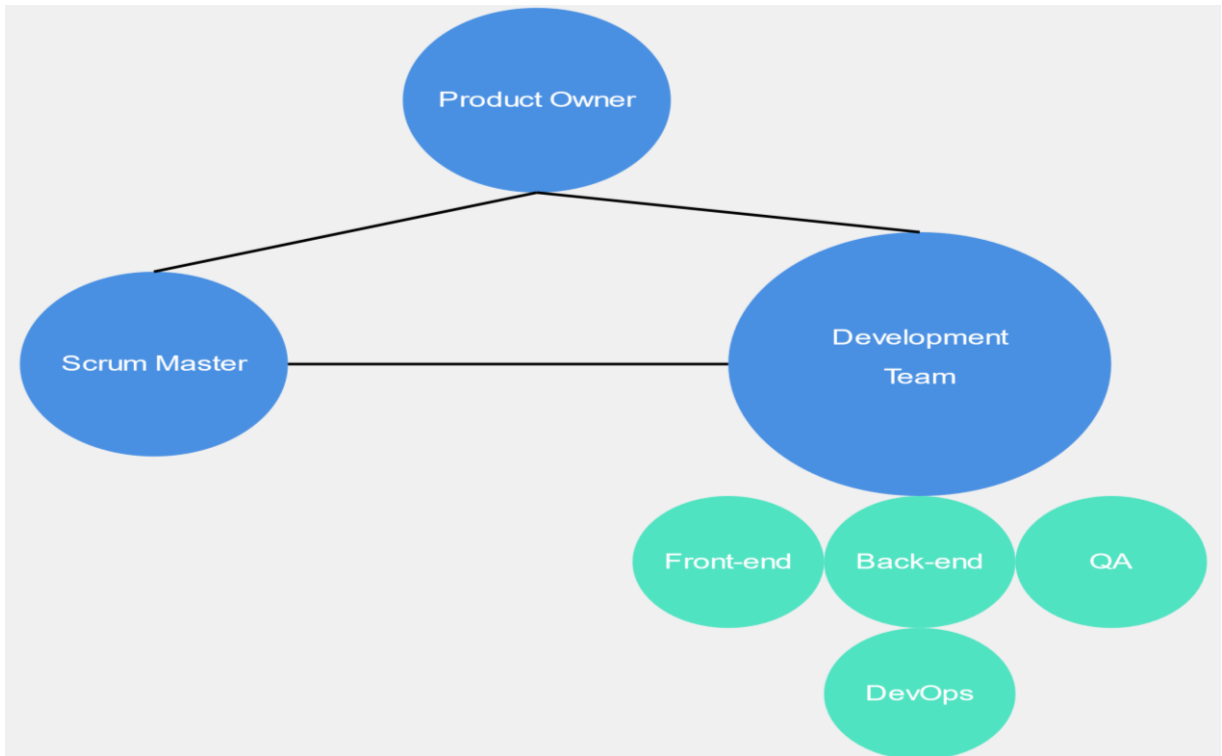
5. Розподіл відповідальності:

- Product Owner: взаємодія з клієнтами, формування та пріоритизація Product Backlog
- Scrum Master: організація Scrum-процесів, усунення перешкод, фасилітація зустрічей
- Development Team: самоорганізація для виконання завдань спринту, спільна відповідальність за результат

6. Навчання та адаптація:

- Проведення вступного тренінгу з Scrum для всієї команди
- Регулярні сесії з обміну знаннями всередині команди

- Поступове вдосконалення процесів через ретроспективи
- Структура Scrum-команди буде виглядати як зображено на рис. 2.2.
- Рисунок 2.2 – Структура Scrum-команди*



2.2.2 Розробка Product Backlog

1. Збір вимог:

- Проведення інтерв'ю з ключовими стейкхолдерами
- Аналіз існуючих бізнес-процесів клієнтів
- Вивчення конкурентних рішень та тенденцій ринку

2. Формування користувацьких історій:

- Створення користувацьких історій на основі зібраних вимог
- Визначення критеріїв прийняття для кожної історії
- Уточнення деталей з Product Owner та стейкхолдерами

3. Пріоритизація:

- Оцінка бізнес-цінності кожної історії
- Визначення залежностей між історіями
- Ранжування історій за пріоритетом

4. Оцінка трудомісткості:

- Проведення сесій Planning Poker для оцінки складності історій
 - Визначення відносної складності в Story Points
5. Структурування Product Backlog:
- Групування історій за функціональними модулями
 - Визначення епіків та тем
 - Створення ієрархічної структури беклогу
6. Уточнення (Refinement):
- Регулярні сесії з командою для уточнення деталей історій
 - Розбиття великих історій на менші
 - Оновлення оцінок та пріоритетів
7. Візуалізація:
- Створення візуального представлення Product Backlog
 - Використання інструментів для управління беклогом (напр., Jira, Trello)

Ілюстрація структури Product Backlog зведена у табл. 2.2.

Таблиця 2.2 – Структура Product Backlog

ID	Користувацька історія	Пріоритет	Оцінка (SP)
1	Як менеджер, я хочу мати можливість створювати нові бізнес-процеси, щоб автоматизувати роботу відділу	Високий	13
2	Як користувач, я хочу мати доступ до дашборду з ключовими метриками, щоб швидко оцінювати ефективність процесів	Високий	8
3	Як адміністратор, я хочу мати можливість налаштовувати права доступу для різних ролей, щоб забезпечити безпеку даних	Середній	5
4	Як аналітик, я хочу мати можливість експортувати дані в різних форматах, щоб проводити подальший аналіз	Низький	3
5	Як користувач, я хочу отримувати сповіщення про критичні події в процесах, щоб вчасно реагувати на проблеми	Середній	5

2.2.3 Планування спринтів

1. Підготовка до планування:

- Оновлення та пріоритизація Product Backlog
- Аналіз результатів попереднього спринту
- Визначення цілей спринту разом з Product Owner

2. Sprint Planning Meeting:

- Частина 1: Вибір елементів з Product Backlog
 - Обговорення пріоритетних історій з Product Owner
 - Уточнення деталей та критеріїв прийняття
 - Визначення мети спринту (Sprint Goal)
- Частина 2: Планування виконання
 - Розбиття історій на завдання
 - Оцінка трудомісткості завдань
 - Визначення можливостей команди (team capacity)

3. Формування Sprint Backlog:

- Створення списку завдань на спринт
- Визначення залежностей між завданнями
- Встановлення пріоритетів виконання

4. Визначення Definition of Done:

- Уточнення критеріїв завершеності для кожного завдання
- Узгодження загальних вимог до якості

5. Оцінка ризиків:

- Ідентифікація потенційних перешкод
- Розробка стратегій мінімізації ризиків

6. Візуалізація плану спринту:

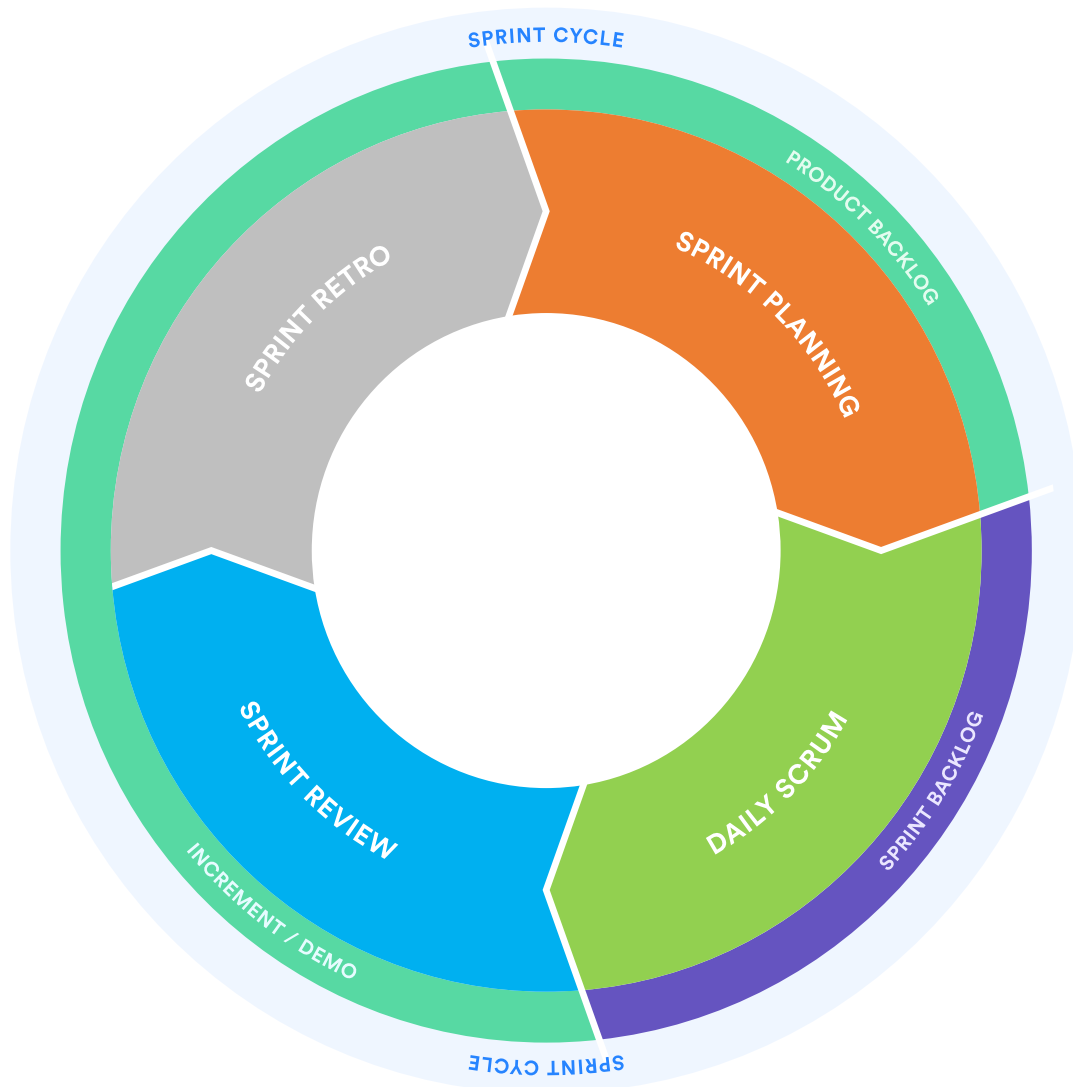
- Створення Scrum-дошки (фізичної або цифрової)
- Розміщення завдань на дошці

7. Завершення планування:

- Підтвердження зобов'язань команди
- Узгодження плану комунікацій на спринт

Планування спринтів зображено на рис 2.3.

Рисунок 2.3 – Планування спринтів



2.2.4 Оцінка часу та ресурсів

Для оцінки часу та ресурсів проєкту, була внесена інформація, яка відповідає критеріям подальшого плану.

Таблиця 2.3 – Оцінка час та ресурсів

Фаза проекту	Тривалість (тижні)	Необхідні ресурси
Аналіз та планування	2	Product Owner, Scrum Master, 2 аналітики
Розробка архітектури	3	Scrum Master, 2 архітектори, 1 розробник
Розробка основного функціоналу	8	Scrum Master, 5 розробників, 2 тестувальники
Інтеграція та тестування	4	Scrum Master, 3 розробники, 3 тестувальники
Впровадження та навчання	3	Product Owner, Scrum Master, 2 розробники, 1 тренер
Загальна тривалість	20	

Додаткові аспекти оцінки:

1. Ресурсне планування:

- Розподіл завдань між членами команди
- Визначення потреби в додаткових спеціалістах
- Планування залучення зовнішніх експертів

2. Бюджетування:

- Оцінка витрат на оплату праці
- Розрахунок вартості необхідного обладнання та програмного забезпечення
- Визначення резерву на непередбачені витрати

3. Аналіз ризиків:

- Ідентифікація потенційних затримок
- Оцінка впливу технічних складнощів
- Розробка планів реагування на ризики

4. Моніторинг та контроль:

- Визначення ключових показників ефективності (KPI)
- Встановлення контрольних точок проекту

- Планування регулярних оглядів прогресу

2.3 Моніторинг виконання проєкту

2.3.1 Організація щоденних Scrum-зустрічей

1. Підготовка до зустрічі:

- Оновлення Scrum-дошки перед зустріччю
- Перевірка статусу завдань у системі управління проєктами
- Підготовка короткого звіту про прогрес для Scrum Master

2. Проведення зустрічі:

- Початок точно в запланований час (максимальна тривалість 15 хвилин)
- Кожен учасник відповідає на три ключові питання: а) Що я зробив вчора? б) Що я планую зробити сьогодні? с) Які перешкоди у мене є?
- Scrum Master фасилітує обговорення, не допускаючи відхилень від теми

3. Фіксація результатів:

- Оновлення статусу завдань на Scrum-дошці
- Запис виявлених перешкод для подальшого вирішення
- Короткий підсумок зустрічі в системі управління проєктами

4. Подальші дії:

- Scrum Master працює над усуненням виявлених перешкод
- За необхідності організуються додаткові зустрічі для вирішення специфічних проблем
- Оновлення Sprint Backlog відповідно до нової інформації

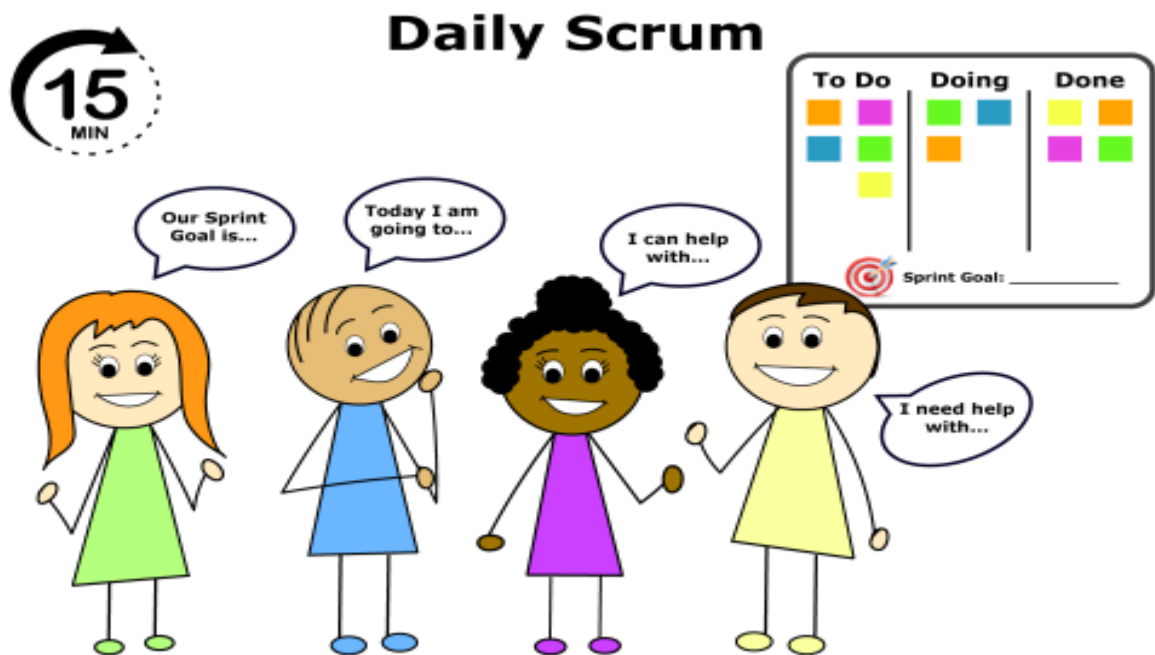
5. Аналіз ефективності:

- Регулярна оцінка корисності щоденних зустрічей
- Збір зворотнього зв'язку від команди щодо формату та змісту зустрічей
- Внесення змін у процес проведення зустрічей за необхідності

Візуалізації процесу щоденних Scrum-зустрічей, яка зображена на рис.

2.4, краще допоможе зрозуміти основний підхід

Рисунок 2.4 – Daily Scrum-meeting



2.3.2 Відстеження прогресу за допомогою Burndown Chart

1. Створення Burndown Chart:

- Визначення загального обсягу роботи на спринт (у story points або годинах)
- Розрахунок ідеальної лінії згорання
- Встановлення початкової точки на графіку

2. Щоденне оновлення:

- Фіксація фактично виконаної роботи
- Оновлення графіка після кожного щоденного Scrum
- Порівняння фактичного прогресу з ідеальною лінією

3. Аналіз відхилень:

- Виявлення відставань або випереджень графіка
- Обговорення причин відхилень на щоденних зустрічах
- Коригування плану спринту за необхідності

4. Прогнозування:

- Оцінка ймовірності завершення всіх завдань спринту
- Визначення потреби в додаткових ресурсах або перегляді обсягу роботи

5. Візуалізація для команди:

- Розміщення Burndown Chart на видному місці (фізично або цифрово)
- Забезпечення доступу всіх членів команди до актуальної версії графіка

6. Використання на Sprint Review:

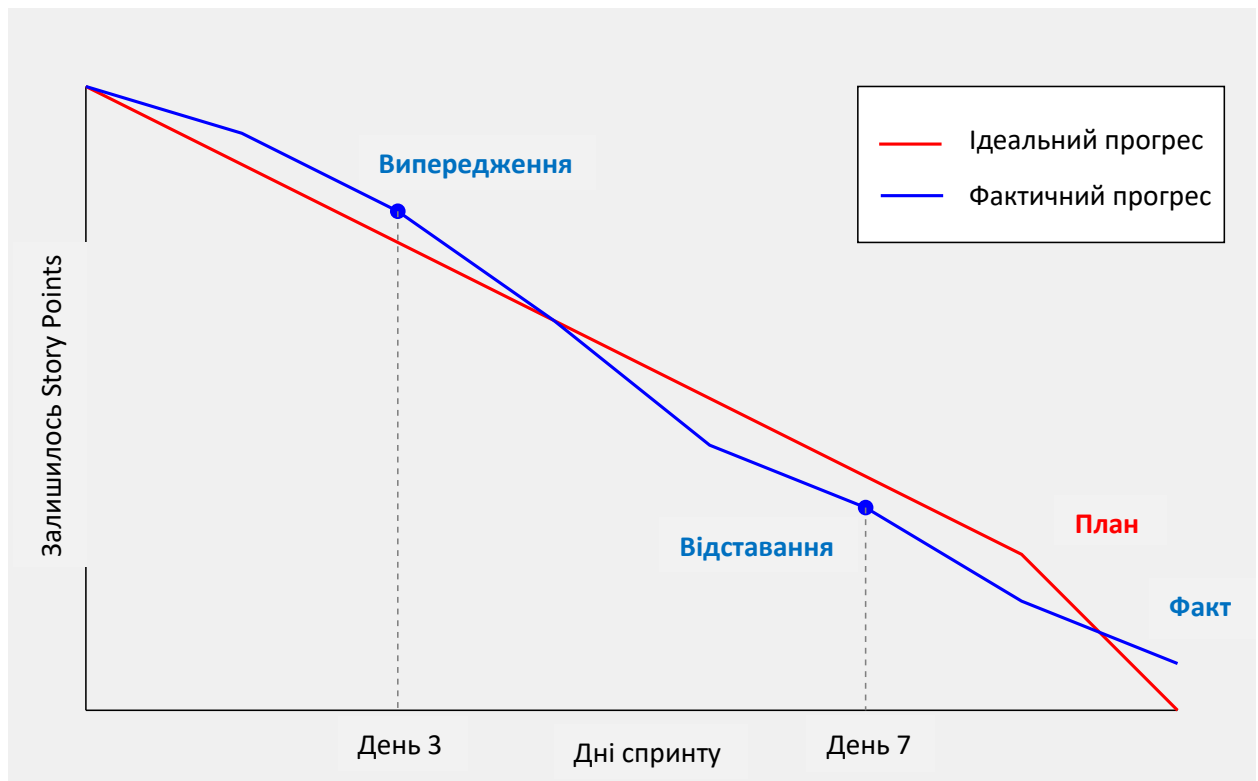
- Презентація Burndown Chart на огляді спринту
- Обговорення виконаної роботи та досягнутого прогресу

7. Ретроспективний аналіз:

- Використання даних з Burndown Chart для аналізу ефективності команди
- Виявлення патернів у роботі команди для подальшого вдосконалення процесів

Приклад Burndown Chart, який зображений на рис. 2.5, вказує на криві ідеального та фактичного прогресу.

Рисунок 2.5 – Приклад Burndown Chart



2.3.3 Проведення Sprint Review та Sprint Retrospective

Для візуалізації процесів Sprint Review та Sprint Retrospective, зібрана інформація у табл. 2.4.

Таблиця 2.4 – Проведення *Sprint Review* та *Sprint Retrospective*

<i>Характеристика</i>	<i>Sprint Review</i>	<i>Sprint Retrospective</i>
Мета	Оцінка результатів спринту	Аналіз процесу роботи команди
Учасники	Scrum Team + Stakeholders	Лише Scrum Team
Фокус	Продукт	Процес
Результат	Оновлений Product Backlog	План покращення процесу роботи
Тривалість	До 4 годин для місячного спринту	До 3 годин для місячного спринту

Висновки до розділу 2

У цьому розділі ми детально розглянули застосування гнучкої методології Scrum для управління створенням програмного забезпечення для автоматизації бізнес-процесів у великих корпораціях. Основні висновки можна узагальнити наступним чином:

1. Обґрунтування вибору Scrum:

- Порівняльний аналіз Agile-методологій показав, що Scrum найкраще підходить для даного проєкту завдяки своїй гнучкості, прозорості та орієнтації на швидку доставку цінності.

- Визначені переваги Scrum, такі як адаптивність до змін, ефективна комунікація та постійне вдосконалення, відповідають вимогам проєкту з автоматизації бізнес-процесів.

2. Планування проєкту:

- Сформовано Scrum-команду з чітко визначеними ролями Product Owner, Scrum Master та Development Team.

- Розроблено початковий Product Backlog, який відображає ключові вимоги до програмного забезпечення для автоматизації бізнес-процесів.

- Запропоновано підхід до планування спринтів, який забезпечує регулярну доставку інкрементів продукту.

- Проведено оцінку часу та ресурсів, необхідних для реалізації проєкту, що дозволяє ефективно розподілити зусилля команди.

3. Моніторинг виконання проєкту:

- Визначено процес проведення щоденних Scrum-зустрічей, які забезпечують постійну комунікацію та швидке виявлення проблем.

- Запропоновано використання Burndown Chart для відстеження прогресу проєкту, що дозволяє візуалізувати швидкість роботи команди та прогнозувати терміни завершення спринту.

- Описано процеси Sprint Review та Sprint Retrospective, які забезпечують регулярний аналіз результатів роботи та постійне вдосконалення процесів команди.

Застосування Scrum до проєкту з автоматизації бізнес-процесів дозволяє забезпечити:

- Гнучкість у реагуванні на зміни вимог замовника
- Регулярну доставку працюючих інкрементів продукту
- Прозорість процесу розробки для всіх зацікавлених сторін
- Постійне вдосконалення продукту та процесів роботи команди

Використання візуальних інструментів, таких як діаграми та графіки, допомагає ефективно комунікувати стан проєкту та полегшує прийняття рішень на всіх рівнях управління.

У наступному розділі ми розглянемо практичні аспекти впровадження розробленого підходу та проаналізуємо результати його застосування для створення програмного забезпечення з автоматизації бізнес-процесів.

РОЗДІЛ 3. РЕЗУЛЬТАТИ РОБОТИ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КОМАНДИ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Огляд інкременту продукту

У цьому підрозділі ми розглянемо результати роботи команди над створенням програмного забезпечення для автоматизації бізнес-процесів.

3.1.1 Опис реалізованої функціональності

Реалізована функціональність зведена у табл. 3.1.

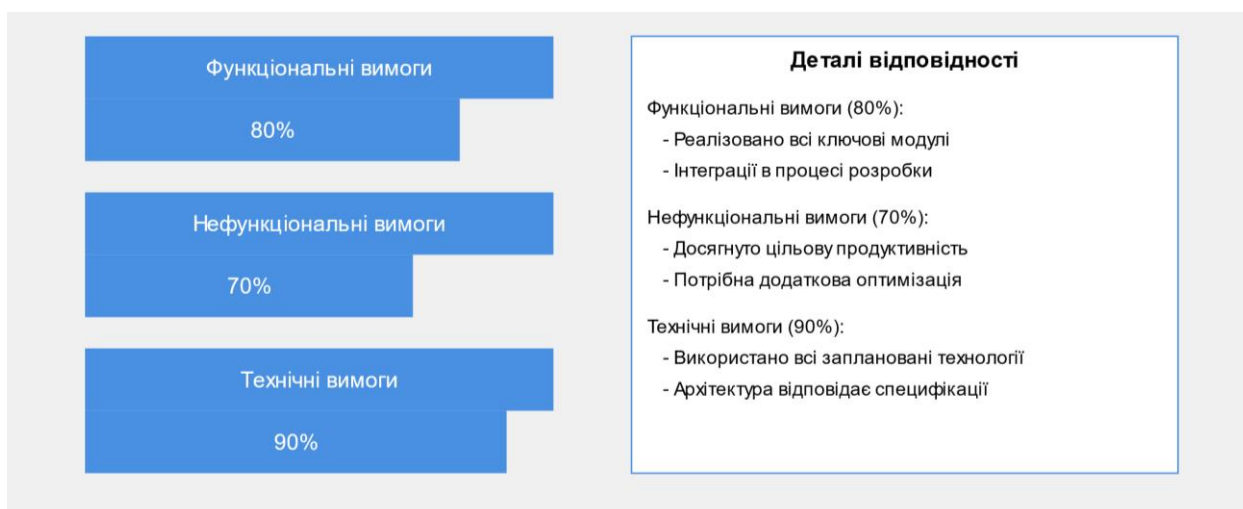
Таблиця 3.1 – Реалізована функціональність

Модуль	Опис функціональності	Статус
Управління бізнес-процесами	Створення, редагування та видалення бізнес-процесів	Завершено
Автоматизація завдань	Налаштування автоматичного виконання рутинних завдань	Завершено
Інтеграція з існуючими системами	API для обміну даними з ERP, CRM та іншими корпоративними системами	В процесі
Аналітичний дашборд	Візуалізація ключових метрик та KPI	Завершено
Управління користувачами	Створення ролей та налаштування прав доступу	Завершено
Сповіщення	Система сповіщень про критичні події та відхилення	В процесі
Звітність	Генерація та експорт звітів	Заплановано

3.1.2 Аналіз відповідності вимогам

Для візуалізації відповідності реалізованої функціональності початковим вимогам, створимо діаграму, яка зображена на рис. 3.1.

Рисунок 3.1 – Відповідність реалізації вимогам



3.2 Ретроспектива роботи команди

3.2.1 Аналіз ефективності Scrum-процесів

Для оцінки ефективності застосування Scrum у проекті, проведемо аналіз ключових метрик та результат отримуємо у табл. 3.2.

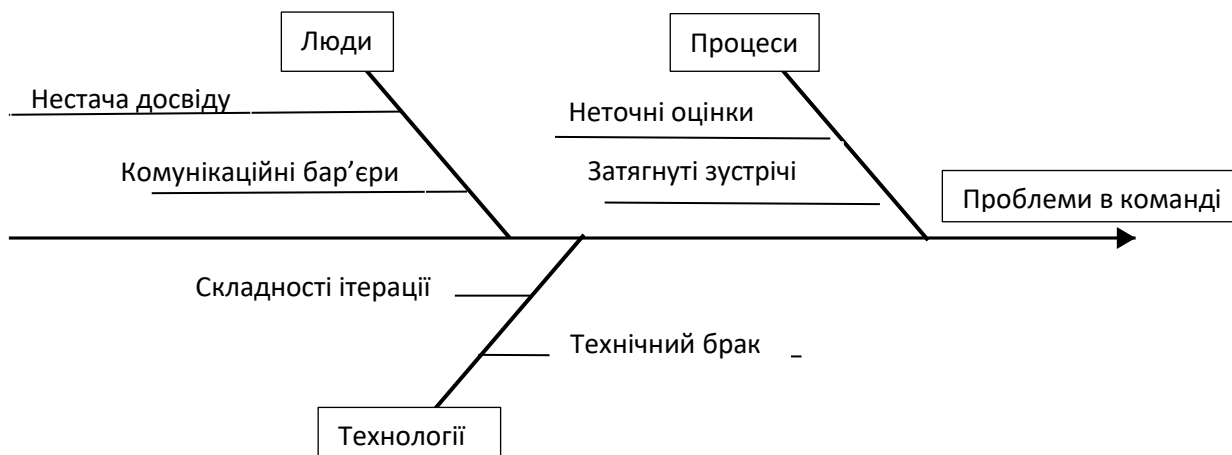
Таблиця 3.2 – Оцінка ефективності застосування Scrum

Метрика	Значення	Ціль	Коментар
Швидкість команди (velocity)	35 SP/спринт	40 SP/спринт	Близько до цілі, але є потенціал для покращення
Точність оцінки (estimation accuracy)	85%	90%	Хороший показник, потрібно продовжувати вдосконалювати
Відсоток завершених завдань (sprint completion rate)	92%	95%	Високий показник, свідчить про гарне планування
Задоволеність клієнта (customer satisfaction)	4.2/5	4.5/5	Позитивний результат, але є простір для покращення
Час виводу на ринок (time-to-market)	3 місяці	2.5 місяці	Потрібно оптимізувати процеси для швидшої доставки

3.2.2 Виявлення проблем та шляхів їх вирішення

На основі проведеної ретроспективи, команда виявила ряд проблем та запропонувала шляхи їх вирішення за допомогою діаграми Ішікави на рис. 3.2.

Рисунок 3.2 – Діаграма Ішікави



3.3 Сучасний інструментарій Scrum-майстра в agile-середовищі

3.3.1 Огляд програмних засобів для управління Agile-проектами

Достатньо багато засобів існує для управління Agile-проектами. У табл. 3.3 описані основні.

Таблиця 3.3 – Сучасний інструментарій Scrum-майстра

Інструмент	Ключові особливості	Переваги	Недоліки
Jira	Гнучкі дошки, розширена звітність, інтеграції	Потужний, масштабований	Складний для початківців
Trello	Прості канбан-дошки, легкість використання	Інтуїтивний, безкоштовний план	Обмежена функціональність для великих проєктів
Asana	Управління завданнями, співпраця	Зручний інтерфейс, багато інтеграцій	Може бути надмірним для малих команд

Продовження таблиці 3.3

Microsoft Azure DevOps	Повний набір інструментів для розробки	Глибока інтеграція з екосистемою Microsoft	Крута крива навчання
GitLab	Управління кодом, CI/CD, управління проєктами	Все в одному рішенні	Може бути надлишковим для нетехнічних команд

3.3.2 Методи підвищення продуктивності Scrum-команди

Для підвищення продуктивності Scrum-команди, було впроваджено ряд методів:

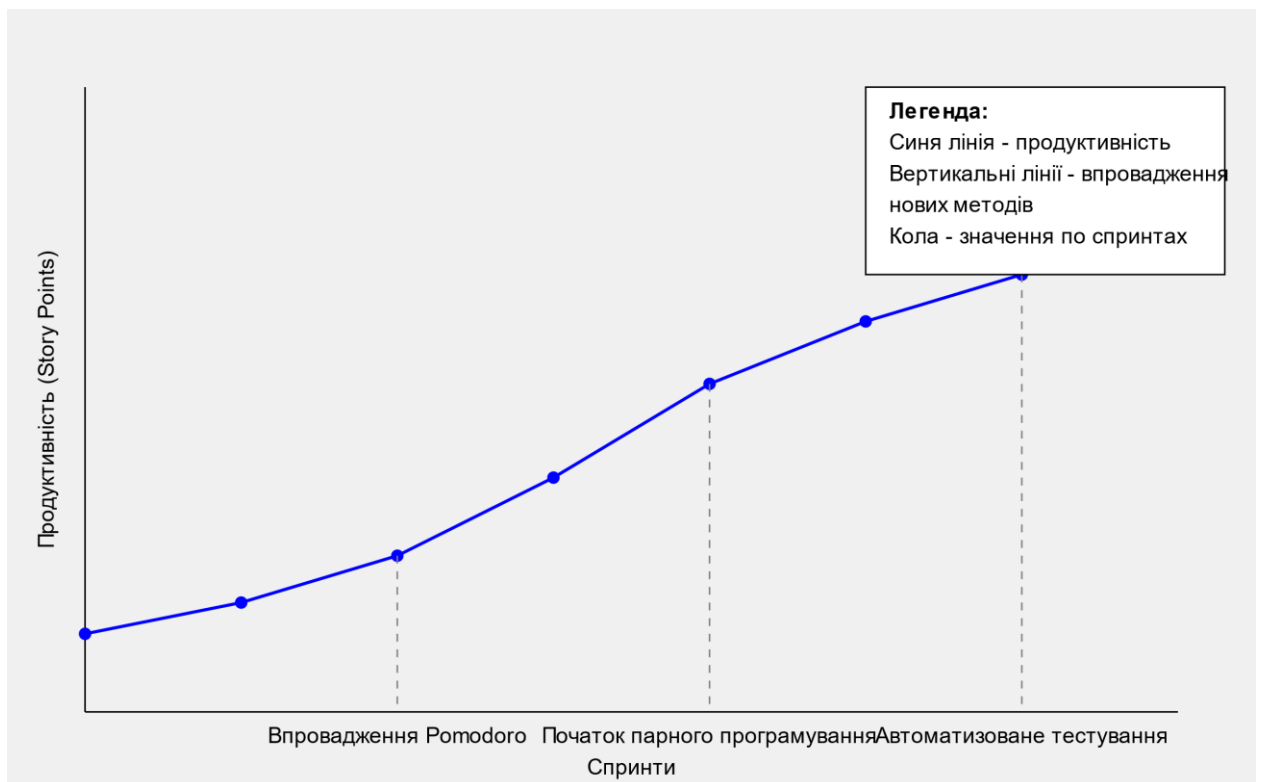
1. Техніка "Pomodoro" для покращення фокусу та управління часом:
 - Розбиття роботи на 25-хвилинні інтервали з короткими перервами
 - Використання таймерів для відстеження часу
 - Аналіз продуктивності на основі кількості завершених "помідорів"
2. Парне програмування для підвищення якості коду та обміну знаннями:
 - Ротація пар програмістів для обміну знаннями
 - Використання інструментів для віддаленого парного програмування
 - Регулярний аналіз ефективності парного програмування
3. Регулярні навчальні сесії для підвищення кваліфікації команди:
 - Організація внутрішніх воркшопів та презентацій
 - Забезпечення доступу до онлайн-курсів та навчальних ресурсів
 - Створення культури постійного навчання та розвитку
4. Впровадження практики Code Review для забезпечення якості коду:
 - Встановлення чітких критеріїв для проведення Code Review
 - Використання інструментів для автоматизованого Code Review
 - Регулярний аналіз результатів Code Review для виявлення областей для покращення

5. Використання автоматизованого тестування для швидкого виявлення помилок:

- Впровадження практик TDD (Test-Driven Development)
- Створення комплексного набору автотестів
- Інтеграція автоматизованого тестування в CI/CD процеси

Візуалізація впливу цих методів на продуктивність команди, зображено на рис. 3.3.

Рисунок 3.3 – Вплив методів на продуктивність команди



3.4 Оцінка економічної ефективності впровадження програмного забезпечення

Для оцінки економічної ефективності впровадження розробленого програмного забезпечення, проведемо аналіз ключових фінансових показників і отриманий результат відображений у табл. 3.4.

Таблиця 3.4 – Оцінка економічної ефективності

Показник	Значення (UAH)	Коментар
Загальний бюджет проекту	2,784,472 грн	Включає оплату скрам-команди, додаткові витрати та резерви
Оплата скрам-команди	1,171,272 грн	Сумарна оплата праці команди по всіх спринтах
Додаткові витрати	1,250,600 грн	Включає витрати на обладнання, інфраструктуру, ліцензії та оплату архітектора проекту
Резерви	362,600 грн	15% від суми оплати скрам-команди та додаткових витрат
Очікуване щорічне скорочення витрат	1,500,000	За рахунок автоматизації процесів
Очікуване збільшення доходу	2,000,000 / рік	Завдяки підвищенню ефективності та швидкості роботи
ROI (Return on Investment)	125%	Очікуваний ROI за перший рік після впровадження
Період окупності	0.8 року	Час, необхідний для повернення інвестицій

3.5 Аналіз ризиків та стратегії їх мінімізації

При реалізації проекту з розробки програмного забезпечення для автоматизації бізнес-процесів важливо враховувати потенційні ризики та розробити стратегії їх мінімізації. Результат винесено у табл. 3.5.

Таблиця 3.5 – Аналіз ризиків та стратегії їх мінімізації

Ризик	Ймовірність	Вплив	Стратегія мінімізації
Зміна вимог замовника	Висока	Середній	Регулярні зустрічі з замовником, гнучке планування спринтів

Продовження таблиці 3.5

Технічні складнощі при інтеграції	Середня	Високий	Детальне планування архітектури, залучення експертів
Відставання від графіку	Середня	Високий	Постійний моніторинг прогресу, коригування обсягу робіт
Проблеми з якістю коду	Низька	Середній	Код-ревію, автоматизоване тестування, дотримання стандартів
Відтік ключових спеціалістів	Низька	Високий	Мотивація команди, створення комфортних умов праці

3.6 Оцінка якості та тестування програмного забезпечення

Для забезпечення високої якості розробленого програмного забезпечення, було впроваджено комплексний підхід до тестування та зображений у табл. 3.6.

Таблиця 3.6 – Оцінка якості та тестування

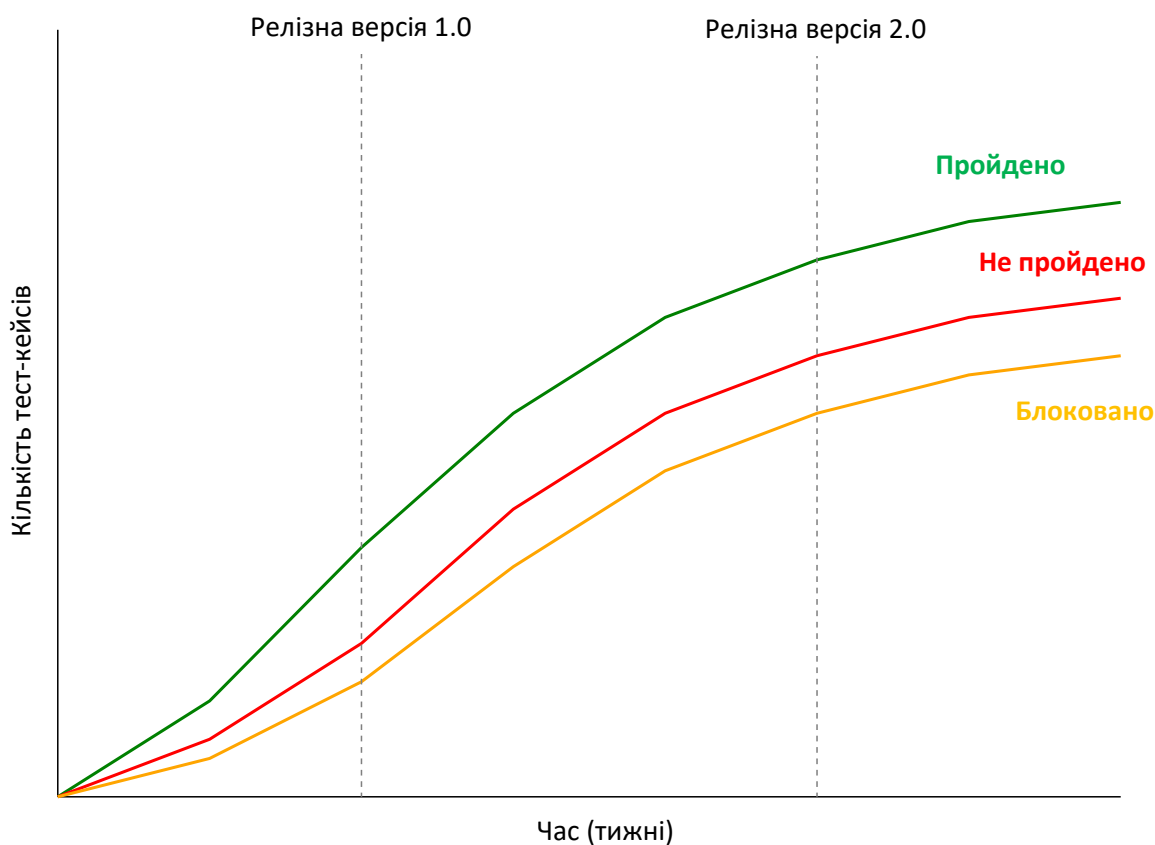
<i>Вид тестування</i>	<i>Опис</i>	<i>Результати</i>
Модульне тестування	Перевірка окремих компонентів системи	95% покриття коду, виявлено та виправлено 23 дефекти
Інтеграційне тестування	Перевірка взаємодії між компонентами	Успішно пройдено 87% тест-кейсів, виявлено 5 критичних проблем
Функціональне тестування	Перевірка відповідності функціональним вимогам	Реалізовано 92% запланованих функцій, 3 функції потребують доопрацювання

Продовження таблиці 3.6

Навантажувальне тестування	Перевірка продуктивності системи під навантаженням	Система витримує до 1000 одночасних користувачів, при більшому навантаженні потрібна оптимізація
Тестування безпеки	Перевірка захищеності системи	Виявлено та усунуто 3 потенційні вразливості, система відповідає стандартам безпеки

Для візуалізації прогресу тестування, створений графік, який зображено на рис. 3.4.

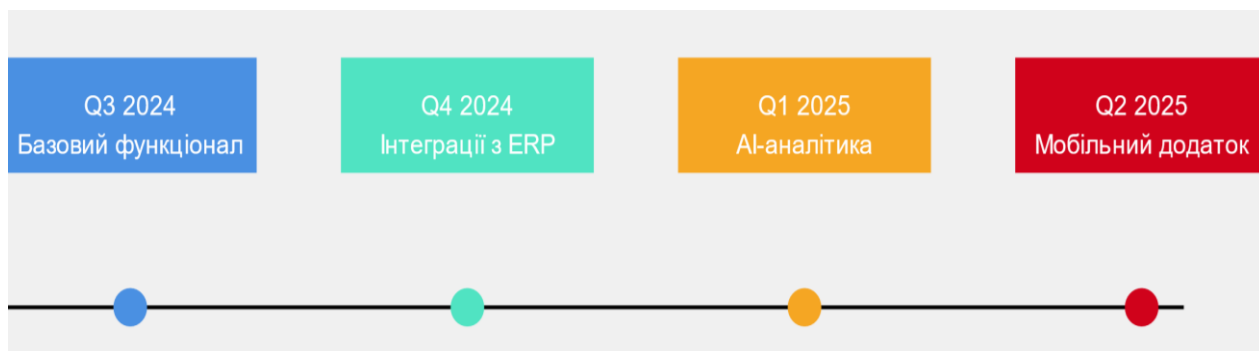
Рисунок 3.4 – Прогрес тестування



3.7 Перспективи розвитку та масштабування проєкту

У цьому підрозділі розглянемо можливості подальшого розвитку та масштабування розробленого програмного забезпечення для автоматизації бізнес-процесів. Подальший розвиток та масштабування зображено на рис. 3.5.

Рисунок 3.5 – Розвиток та масштабування проєкту



Основні напрямки розвитку продукту включають:

- Розширення функціональності для охоплення більшої кількості бізнес-процесів.
- Покращення інтеграційних можливостей з існуючими корпоративними системами.
- Впровадження елементів штучного інтелекту для прогнозної аналітики.
- Розробка мобільного додатку для доступу до системи з будь-якого пристрою.

Для оцінки потенціалу масштабування проєкту, проведемо SWOT-аналіз та результат відображений у табл 3.7.

Таблиця 3.7 – SWOT-аналіз

Сильні сторони	Слабкі сторони
Гнучка архітектура системи	Обмежені ресурси для швидкого масштабування
Досвідчена Scrum-команда	Залежність від ключових розробників
Високий рівень задоволеності клієнтів	Недостатньо розвинена партнерська мережа

Продовження таблиці 3.7

Можливості	Загрози
Зростаючий попит на автоматизацію бізнес-процесів	Посилення конкуренції на ринку
Потенціал виходу на міжнародні ринки	Швидкі технологічні зміни
Можливості інтеграції з новими технологіями (AI, IoT)	Зміни в законодавстві щодо обробки даних

На основі проведеного SWOT-аналізу можемо визначити ключові стратегічні напрямки для розвитку та масштабування проєкту:

1. Розвиток продукту:

- Продовжувати вдосконалення гнучкості архітектури для легкого додавання нових функцій.
- Інвестувати в розробку нових модулів, особливо з використанням AI та машинного навчання.
- Покращити локалізацію продукту для виходу на міжнародні ринки.

2. Розвиток команди:

- Впровадити програму передачі знань для зменшення залежності від ключових розробників.
- Розширити команду, залучаючи нових спеціалістів для підтримки масштабування.
- Інвестувати в навчання та розвиток існуючих членів команди.

3. Розширення ринку:

- Розробити стратегію виходу на міжнародні ринки, починаючи з найбільш перспективних регіонів.
- Створити партнерську програму для розширення мережі дистрибуції та підтримки.
- Розвивати галузеві рішення для специфічних секторів економіки.

4. Технологічний розвиток:

- Досліджувати можливості інтеграції з IoT та Blockchain для розширення функціональності.
- Вдосконалювати інфраструктуру для забезпечення високої масштабованості системи.
- Постійно оновлювати систему безпеки для відповідності новим стандартам та законодавству.

5. Клієнтський досвід:

- Розробити комплексну програму навчання та онбордингу для нових клієнтів.
- Створити спільноту користувачів для обміну досвідом та ідеями.
- Впровадити систему постійного збору та аналізу зворотного зв'язку від клієнтів.

Висновки до розділу 3

У третьому розділі були розглянуті результати розробки програмного забезпечення для автоматизації бізнес-процесів та проаналізовані шляхи підвищення ефективності команди. Основні висновки:

- Реалізовано ключові модулі системи, що відповідають більшості функціональних та нефункціональних вимог.
- Проведено ретроспективу роботи команди, яка виявила сильні сторони та області для вдосконалення Scrum-процесів.
- Проаналізовано економічну ефективність проєкту, загальний бюджет якого склав 2,784,472 грн, що включає оплату скрам-команди, додаткові витрати та резерви.
- Визначено та оцінено ключові ризики проєкту, розроблено стратегії їх мінімізації.
- Проведено комплексне тестування програмного забезпечення, яке показало високий рівень якості продукту.

- Розроблено дорожню карту подальшого розвитку продукту, яка включає розширення функціональності, покращення інтеграційних можливостей та впровадження нових технологій.
- SWOT-аналіз виявив значний потенціал для масштабування проєкту, але також вказав на необхідність подолання певних викликів для успішного зростання.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досліджено та реалізовано гнучке управління створенням програмного забезпечення для автоматизації бізнес-процесів у великих корпораціях з використанням методології Scrum. Основні результати роботи:

- Проведено аналіз предметної області автоматизації бізнес-процесів у великих корпораціях, визначено ключові проблеми та потреби. На основі цього аналізу сформульовано цілі та завдання проєкту, а також визначено функціональні, нефункціональні та технічні вимоги до програмного забезпечення.
- Обґрунтовано вибір фреймворку Scrum для управління проєктом. Розроблено детальний план проєкту, включаючи формування Scrum-команди, розробку Product Backlog, планування спринтів та оцінку часу і ресурсів.
- Реалізовано ключові модулі програмного забезпечення для автоматизації бізнес-процесів, включаючи управління бізнес-процесами, автоматизацію завдань, інтеграцію з існуючими системами та аналітичний дашборд.
- Проведено комплексне тестування розробленого програмного забезпечення, яке показало високий рівень відповідності вимогам та якості продукту.
- Здійснено економічну оцінку проєкту. Загальний бюджет склав 2,784,472 грн, що включає оплату скрам-команди, додаткові витрати та резерви.
- Проаналізовано ризики проєкту та розроблено стратегії їх мінімізації. Створено теплову карту ризиків для візуалізації їх впливу та ймовірності.
- Розроблено дорожню карту подальшого розвитку продукту, яка включає розширення функціональності, покращення інтеграційних можливостей та впровадження нових технологій.

- Проведено SWOT-аналіз, який виявив значний потенціал для масштабування проєкту, а також визначив ключові виклики для майбутнього зростання.

Застосування методології Scrum дозволило ефективно управляти процесом розробки, забезпечуючи гнучкість у реагуванні на зміни вимог та швидку доставку цінності клієнту. Регулярні спринт-огляди та ретроспективи сприяли постійному вдосконаленню процесів та підвищенню ефективності команди.

Результати роботи демонструють успішне застосування гнучких методологій для розробки складного програмного забезпечення в контексті автоматизації бізнес-процесів великих корпорацій. Розроблений продукт має значний потенціал для подальшого розвитку та масштабування, що відкриває перспективи для майбутніх досліджень та розробок у цій галузі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Положення про організацію освітнього процесу в університеті «КРОК». 2022 рік. Сайт університету «КРОК». Електронний ресурс. URL: <https://library.krok.edu.ua/media/library/category/publiczna-informatsiya/polozhennya-orhanizaaciya-op.pdf>
2. Положення про практичну підготовку здобувачів вищої освіти Університету «КРОК». 2020 рік. Сайт університету «КРОК». Електронний ресурс. URL: https://library.krok.edu.ua/media/library/category/publiczna-informatsiya/polozhennya-pro-praktichnu-pidgotovku_2020-04-05.pdf
3. Положення про внутрішню систему забезпечення якості вищої освіти та освітньої діяльності Університету «КРОК». 2018 рік. Сайт університету «КРОК». Електронний ресурс. URL: https://library.krok.edu.ua/media/library/category/publiczna-informatsiya/3-8_polozhennia-pro-vnutrishniu-systemu-yakosti_2018-04-26_stamp.pdf
4. Кодекс академічної доброчесності Університету «КРОК». 2018 рік. Електронний ресурс університету «КРОК». Електронний ресурс. URL: https://livekrokedu.sharepoint.com/sites/msteams_c1b3fe/Shared%20Documents/General/2018-10-18%20%D0%9A%D0%BE%D0%B4%D0%B5%D0%BA%D1%81%20%D0%90%D0%BA%D0%B0%D0%B4%D0%B5%D0%BC%D1%96%D1%87%D0%BD%D0%BE%D1%97%20%D0%B4%D0%BE%D0%B1%D1%80%D0%BE%D1%87%D0%B5%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%96.pdf
5. Положення про перевірку академічних та наукових текстів на плагіат в Університеті «КРОК». 2020 рік. Сайт університету «КРОК». Електронний ресурс. URL: https://library.krok.edu.ua/media/library/category/publiczna-informatsiya/2020-04-10_polozhennya-pro-perevirku-tekstiv-na-plagiat.pdf

6. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. Електронний ресурс. URL: http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF.
7. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні вимоги та правила складання. Видання офіційне. Електронний ресурс. URL: <http://pdf.lib.vntu.edu.ua/books/2018/%D0%94%D0%A1%D0%A2%D0%A3%208302%20%D0%BF%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9.pdf>.
8. Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org. <https://scrumguides.org/scrum-guide.html>
9. Project Management Institute. (2017). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) (6th ed.). Project Management Institute.
10. Cohn, M. (2009). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley Professional.
11. Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professional.
12. Stellman, A., & Greene, J. (2014). Learning Agile: Understanding Scrum, XP, Lean, and Kanban. O'Reilly Media.
13. Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional.
14. Kim, G., Debois, P., Willis, J., & Humble, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press.
15. Fowler, M. (2018). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley Professional.

16. Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
17. Beck, K., & Andres, C. (2004). Extreme Programming Explained: Embrace Change (2nd ed.). Addison-Wesley Professional.
18. Leffingwell, D. (2018). SAFe 4.5 Reference Guide: Scaled Agile Framework for Lean Enterprises (2nd ed.). Addison-Wesley Professional.
19. Atlassian. (n.d.). Agile Coach. <https://www.atlassian.com/agile>
20. Scrum Alliance. (n.d.). Learn About Scrum. <https://www.scrumalliance.org/learn-about-scrum>
21. Agile Alliance. (n.d.). Agile 101. <https://www.agilealliance.org/agile101/>
22. International Organization for Standardization. (2012). ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements. <https://www.iso.org/standard/54534.html>
23. OWASP Foundation. (2021). OWASP Top Ten. <https://owasp.org/www-project-top-ten/>
24. Highsmith, J. (2009). Agile Project Management: Creating Innovative Products (2nd ed.). Addison-Wesley Professional.
25. Poppendieck, M., & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit. Addison-Wesley Professional.
26. Kniberg, H. (2015). Scrum and XP from the Trenches (2nd ed.). InfoQ.
27. Derby, E., & Larsen, D. (2006). Agile Retrospectives: Making Good Teams Great. Pragmatic Bookshelf.

ДОДАТКИ