

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

**КВАЛІФІКАЦІЙНА РОБОТА**

Тема: «Гнучке управління розробкою програмного забезпечення  
автоматичного тестування двигунів літака Boeing 737 MAX»

Ступінь вищої освіти – магістр

Спеціальність: 073 – Менеджмент

Освітня програма: «Agile-технології розробки програмного забезпечення»

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

Керівник: д.е.н., доц., професор  
кафедри ІММС  
Ольга ОРЛОВА-КУРИЛОВА

Керівник: к.е.н., проф.,  
професор кафедри УТ  
Наталія ЛІТВІН

Виконав: здобувач групи  
МЕН/Agile-23м  
Микола ЧЕРВІНЕЦЬ

Київ, 2025 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  
ЧЕРВІНЕЦЬ МИКОЛА ВАСИЛЬОВИЧ**

Тема роботи	ГНУЧКЕ УПРАВЛІННЯ РОЗРОБКОЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЧНОГО ТЕСТУВАННЯ ДВИГУНІВ ЛІТАКА BOEING 737 MAX
Номер та дата наказу про затвердження теми	№ 106-1 від 11 листопада 2024 року
Коротка постановка завдання	Впровадження гнучкого підходу до управління створенням програмного забезпечення автоматичного тестування двигунів літака Boeing 737 MAX призначеної для вирішення проблем з якістю тестування не автоматичними методами.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	What is scrum master? // The Home of Scrum! – URL: <a href="https://www.scrum.org/resources/what-is-a-scrum-master">https://www.scrum.org/resources/what-is-a-scrum-master</a> Debugging teams with the Lencioni Model // Addy Osmani Engineering leader and senior thinker – URL: <a href="https://addyosmani.com/blog/debugging-teams-lencioni/">https://addyosmani.com/blog/debugging-teams-lencioni/</a>
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 «11» листопада 2024 р.

Керівник

Ольга ОРЛОВА-КУРИЛОВА

Керівник

Наталія ЛІТВІН

Здобувач

Микола ЧЕРВІНЕЦЬ

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
<b>Підготовчий етап</b>			
1	Вибір напрямку дослідження та керівника	01.11.2024 р.	<i>виконано</i>
2	Формування теми та призначення керівника	08.11.2024 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи	11.11.2024 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу	11.11.2024 р.	<i>виконано</i>
<b>Основний етап</b>			
5	Розробка концепції кваліфікаційної роботи	12.11.2024 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів.	12.11.2024 р.	<i>виконано</i>
7	Затвердження розширеної постановки завдання. Підготовка та подання керівнику розділу 1 кваліфікаційної роботи	15.11.2024 р.	<i>виконано</i>
8	Проектування інформаційної системи. Підготовка та подання керівнику розділу 2 кваліфікаційної роботи	22.11.2024 р.	<i>виконано</i>
9	Реалізація інформаційної системи. Підготовка та подання керівнику розділу 3 кваліфікаційної роботи	29.11.2024 р.	<i>виконано</i>
10	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	02.12.2024 р.	<i>виконано</i>
11	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівнику доопрацьованого варіанту кваліфікаційної роботи	09.12.2024 р.	<i>виконано</i>
<b>Завершальний етап</b>			
12	Представлення рукопису для перевірки на плагіат	09.12.2024 р.	<i>виконано</i>
13	Підготовка презентації та доповіді на передзахист	13.12.2024 р.	<i>виконано</i>
14	Передзахист кваліфікаційної роботи	23.12.2024 р.	<i>виконано</i>
15	Технічна самоекспертиза роботи на відповідність вимогам до оформлення та виправлення недоліків	29.12.2024 р.	<i>виконано</i>
16	Експертиза роботи керівником та зовнішнім експертом	06.01.2025 р.	<i>виконано</i>
17	Доопрацювання доповіді та презентації для захисту	13.01.2025 р.	<i>виконано</i>
18	Захист кваліфікаційної роботи	20-24.01.2025 р.	<i>виконано</i>

Керівник

Ольга ОРЛОВА-КУРИЛОВА

Керівник

Наталія ЛІТВІН

Здобувач

Микола ЧЕРВІНЕЦЬ

## АНОТАЦІЯ

*Червінець М.В. Гнучке управління розробкою програмного забезпечення автоматичного тестування двигунів літака Boeing 737 MAX.*

Пояснювальна записка кваліфікаційної роботи за спеціальністю 073 – Менеджмент (освітня програма – Agile-технології розробки програмного забезпечення), СО Магістр. – ВНЗ «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра інформаційного менеджменту, математики та статистики, Київ, 2025 р.

*Робота присвячена розробці програмного забезпечення автоматичного тестування двигунів літака Boeing 737 MAX для підвищення якості роботи. Основна мета – зменшення часу на тестування двигунів, завдяки впровадженню передової програми, яка замінить роботу механіків, та підвищення якості тестування шляхом виключення людського фактору. Проєкт реалізується з використанням Agile-методології, що дозволяє адаптувати розробку до вимог користувачів та швидко реагувати на зміни.*

Ключові слова: якість, авіабудування, гнучке управління, Agile, програмне забезпечення.

## ANNOTATION

*Chervinets M.V. Agile Management of the software development for automatic testing of Boeing 737 MAX aircraft engines.*

Project explanatory note by specialty 073 – Management (educational program – Agile software development technologies), Master's Degree. – "KROK" University, Educational and Scientific Institute of Information and Communication Technologies, Department of Information Management, Mathematics and Statistics, Kyiv, 2025.

*The work is devoted to the development of software for automatic testing of Boeing 737 MAX aircraft engines to improve the quality of work. The main goal is to reduce the time for engine testing, due to the introduction of an advanced program that will replace the work of mechanics, and to improve the quality of testing by eliminating the human factor. The project is implemented using Agile methodology, which allows you to adapt the development to user requirements and quickly respond to changes.*

Keywords: quality, aircraft manufacturing, flexible management, Agile, software.

## ЗМІСТ

ВСТУП.....	7
<b>РОЗДІЛ 1. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, АВТОМАТИЧНОГО ТЕСТУВАННЯ ДВИГУНІВ ЛІТАКА.....</b>	<b>11</b>
<b>BOEING 737 MAX .....</b>	<b>11</b>
1.1. Опис предметної області: авіабудівні підприємства та їх потреби....	11
1.2. Постановка цілей і завдань проєкту.....	14
1.3. Визначення вимог до продукту.....	17
Висновки до розділу 1.....	22
<b>РОЗДІЛ 2. ГНУЧКЕ УПРАВЛІННЯ РОЗРОБКОЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>23</b>
2.1. Обґрунтування вибору Agile-фреймворку для управління проєктом.....	23
2.2. Планування проєкту розробки додатку, тестування двигунів літака Boeing 737 MAX : етапи, бюджет та команда .....	27
2.3 Фактори макросередовища .....	35
Висновки до розділу 2.....	38
<b>РОЗДІЛ 3. РЕЗУЛЬТАТИ РОЗРОБКИ ТА ОПТИМІЗАЦІЯ продукту .....</b>	<b>40</b>
3.1. Аналіз результативності та ретроспектива роботи команди .....	40
3.2 Виявлення проблем та шляхів їх вирішення.....	46
3.3 Методи підвищення продуктивності Scrum-команди.....	47
3.4 Аналіз ризиків та стратегії їх мінімізації .....	49
Висновки до розділу 3.....	51
<b>ВИСНОВКИ .....</b>	<b>52</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>55</b>
<b>ДОДАТКИ.....</b>	<b>58</b>

## ВСТУП

**Актуальність теми.** Авіаційна галузь є одним із найдинамічніших і технологічно розвинених секторів світової економіки. Безперервний розвиток авіаційної техніки, зростання кількості пасажирів та посилення вимог до безпеки польотів створюють нові виклики для виробників літаків та авіакомпаній. Одним з ключових аспектів забезпечення безпеки польотів є надійність та ефективність роботи двигунів літаків.

Основні фактори, що підтверджують актуальність теми дослідження:

- високі вимоги до безпеки - авіакатастрофи мають катастрофічні наслідки, тому безпека польотів є пріоритетом для всіх учасників ринку;
- зростання складності авіаційної техніки - сучасні літаки оснащені складними системами управління, що потребують ретельного тестування;
- висока вартість ручного тестування – ручне тестування є трудомістким і дорогим процесом, який вимагає значних часових витрат;
- потреба в підвищенні ефективності - автоматизація процесу тестування дозволяє скоротити час проведення тестувань, зменшити кількість помилок, пов'язаних з людським фактором, та підвищити загальну ефективність виробництва;
- розвиток технологій - сучасні технології, такі як штучний інтелект, машинне навчання та великі дані, відкривають нові можливості для розробки інноваційних систем тестування.

Конкретно стосовно Boeing 737 MAX:

- попередні інциденти - відомі випадки аварій, пов'язаних з цим типом літаків, підкреслюють необхідність вдосконалення систем безпеки та контролю;
- публічний резонанс - великий суспільний інтерес до безпеки польотів літаків Boeing 737 MAX створює додатковий тиск на виробника та регуляторні органи.

Таким чином, розробка програмного забезпечення для автоматизованого тестування двигунів Boeing 737 MAX є актуальним завданням, яке відповідає

сучасним вимогам авіаційної галузі та сприяє підвищенню рівня безпеки польотів.

**Мета дослідження.** Метою дослідження є розробка інноваційного програмного забезпечення для автоматизованого тестування двигунів літаків Boeing 737 MAX з метою підвищення безпеки польотів шляхом забезпечення надійності та ефективності роботи двигунів. Автоматизація процесу тестування дозволить скоротити час проведення процедур, зменшити кількість помилок, пов'язаних з людським фактором, та підвищити загальну ефективність виробництва.

**Завдання дослідження.** Перед розробкою інноваційного програмного забезпечення для тестування двигунів, необхідно провести всебічне дослідження сучасного стану авіаційної галузі. Це включає в себе аналіз існуючих рішень, виявлення їхніх недоліків та визначення потреб ринку.

Для ефективного управління проектом буде застосовано гнучку методологію Agile. Такий підхід дозволить оперативно реагувати на зміни вимог та забезпечить високу якість кінцевого продукту. Планування проекту включатиме детальне розбиття задач, оцінку ресурсів та встановлення чітких термінів.

На кожному етапі розробки буде здійснюватися ретельне тестування та збір зворотного зв'язку від зацікавлених сторін. Це дозволить вчасно виявляти та усувати недоліки, а також вносити необхідні корективи в проект. Після завершення розробки буде розроблена стратегія виведення продукту на ринок, включаючи маркетингові заходи та підтримку користувачів.

**Об'єкт, предмет та методи дослідження.** Об'єктом дослідження є процес створення системи моніторингу та діагностики двигунів літаків Boeing 737 MAX. Предметом дослідження виступає розробка та оцінка ефективності алгоритмів предиктивного аналізу для виявлення аномалій у роботі двигуна та прогнозування потенційних відмов.

У дослідженні застосовуються різноманітні методи. Теоретичні підходи включають аналіз наукової літератури, вивчення існуючих рішень та

моделювання бізнес-процесів. Емпіричні методи передбачають спостереження за процесом розробки, проведення опитувань серед потенційних користувачів і аналіз даних із використанням інструментів управління проєктами. Практичні методи охоплюють розробку та тестування функціональних модулів програмного забезпечення для автоматичного тестування двигунів літака Boeing 737 MAX, впровадження інструментів для виконання розрахунків і проведення ретроспективних сесій.

**Очікувані результати дослідження.** Після впровадження запропонованого проєктного рішення, буде створено новий, більш точний алгоритм прогнозування відмов двигуна на основі машинного навчання. Цей алгоритм зможе виявляти ранні ознаки потенційних проблем, які можуть призвести до відмови, з більшою точністю та заздалегідь до того, як вони стануть критичними.

Завдяки своєчасному виявленню потенційних проблем, можна буде здійснювати планове технічне обслуговування, що значно підвищить надійність двигуна та безпеку польотів. Раннє виявлення проблем дозволить уникнути непланових ремонтів та продовжити термін служби двигуна.

Будуть розроблені нові тестові сценарії, які дозволять більш ефективно оцінювати працездатність двигуна та виявляти потенційні проблеми. За рахунок оптимізації процесу тестування можна значно скоротити час, необхідний для проведення всіх необхідних перевірок. Оптимізація процесу дозволить зменшити витрати на проведення тестів. Буде розроблений новий програмний комплекс, який дозволить збирати, зберігати, обробляти та візуалізувати дані з датчиків двигуна. Програмний комплекс автоматизує рутинні операції з аналізу даних, що дозволить інженерам зосередитися на більш складних завданнях.

Загальні очікувані результати виходячи з вищезазначених підпунктів:

- підвищення безпеки польотів - завдяки своєчасному виявленню та усуненню потенційних проблем можна значно підвищити безпеку польотів;

- збільшення ресурсу двигуна - оптимізація процесів обслуговування та прогнозування відмов дозволить збільшити ресурс двигуна;
- зниження витрат на обслуговування - за рахування ефективнішого використання ресурсів та оптимізації процесів тестування можна значно знизити витрати на обслуговування двигунів;
- створення нових знань проведення досліджень дозволить отримати нові знання про роботу двигунів Boeing 737 MAX, що може бути використано для подальшого розвитку авіаційної галузі.

**Структура роботи.** Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, викладених на 66 сторінках тексту. Матеріали кваліфікаційної роботи містять 6 таблиць і 8 рисунків. Список використаних джерел складається з 20 найменувань, які уміщено на 3 сторінках, додатки розміщені на 2 сторінках.

# РОЗДІЛ 1

## РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, АВТОМАТИЧНОГО ТЕСТУВАННЯ ДВИГУНІВ ЛІТАКА BOEING 737 MAX

### 1.1. Опис предметної області: авіабудівні підприємства та їх потреби

Авіабудівні підприємства є складною галуззю, яка об'єднує інженерні, виробничі, науково-дослідні та логістичні процеси для створення авіаційної техніки. Вони займаються розробкою, виготовленням, випробуваннями, модернізацією та обслуговуванням літаків, вертольотів і безпілотних літальних апаратів. Ця галузь орієнтована на високотехнологічні рішення, де кожен етап виробництва вимагає високої точності, інноваційного підходу та значних фінансових інвестицій.

Основою діяльності таких підприємств є конструювання літальних апаратів із застосуванням передових матеріалів, таких як композити, титанові сплави, вуглецеві волокна. Крім того, важливою складовою є інтеграція авіоніки, програмного забезпечення та систем управління, які забезпечують безпеку, маневровість і ефективність експлуатації. Процеси передбачають взаємодію багатьох інженерних дисциплін: аеродинаміки, механіки, електроніки, програмування, а також системного аналізу [7].

Інновації є ключовим драйвером розвитку авіабудування, оскільки галузь потребує постійного вдосконалення технологій для створення більш ефективної та екологічної авіації. Сучасні дослідження спрямовані на розробку надлегких матеріалів, які зменшують масу літального апарата, підвищуючи його енергоефективність. Наприклад, широке застосування вуглецевих композитів не лише забезпечує високу міцність конструкцій, але й сприяє зниженню витрати палива. Одним із актуальних напрямів є створення електричних і гібридних літальних апаратів, що відповідає глобальним тенденціям до зниження викидів CO<sub>2</sub>.

Для забезпечення своєї діяльності авіабудівні підприємства мають низку потреб, серед яких ключове місце займають науково-технічні ресурси.

Постійний розвиток технологій вимагає тісної співпраці з дослідницькими центрами, університетами та лабораторіями, де розробляються нові матеріали, конструкції та методи виробництва. Це зумовлено необхідністю адаптуватися до змін у вимогах ринку, таких як підвищення паливної ефективності, зменшення впливу на довкілля та вдосконалення безпекових стандартів.

Також ці підприємства потребують сучасної інфраструктури, включаючи верстати з числовим програмним управлінням (ЧПУ), лінії автоматизованого складання, випробувальні стенди, аеродинамічні труби. Ефективна логістика і постачання мають вирішальне значення, адже авіаційне виробництво залежить від своєчасного постачання високоякісних компонентів. Тут важливі партнерські зв'язки з постачальниками, які здатні забезпечити продукцію відповідно до суворих стандартів [11].

Діджиталізація процесів відкриває нові можливості для оптимізації виробництва, проектування та експлуатації авіаційної техніки. Застосування цифрових двійників, віртуальної реальності та автоматизації дозволяє авіабудівним компаніям скоротити терміни виведення продукції на ринок і мінімізувати ризики, пов'язані з технічними помилками. Проте впровадження таких технологій потребує значних капіталовкладень і високого рівня підготовки персоналу.

Управління проектами та контроль якості є ще одними критичними аспектами. Будівництво літальних апаратів вимагає координації між багатьма підрозділами, а також забезпечення відповідності міжнародним стандартам, таким як AS9100 або ISO 9001. Особливе значення має сертифікація продукції, яка підтверджує її надійність і безпеку.

Окремою категорією потреб є підготовка кваліфікованих кадрів. Авіабудування залучає фахівців із різних галузей, які повинні володіти глибокими знаннями та практичними навичками. Підприємства часто інвестують у програми навчання, стажування та підвищення кваліфікації для інженерів, технологів, програмістів і управлінців.

На сучасному етапі глобалізації авіабудівні підприємства зосереджуються на налагодженні міжнародної кооперації. Спільна розробка проєктів у межах міжнародних консорціумів дозволяє знижувати витрати та обмінюватися передовим досвідом. Крім того, вихід на нові ринки вимагає адаптації технічних рішень до місцевих регулятивних стандартів і особливостей споживчого попиту.

Важливим аспектом функціонування авіабудівних підприємств є дотримання принципів соціальної відповідальності. Це включає не лише створення екологічно безпечної продукції, але й забезпечення відповідних умов праці для співробітників, інвестування в розвиток локальних громад та участь у ініціативах зі збереження довкілля. Водночас посилення нормативних вимог і суспільний тиск стимулюють галузь до переходу на сталі виробничі практики.

На рівні управління стратегічне значення має фінансування. Висока вартість розробки та виробництва авіаційної техніки зумовлює необхідність пошуку інвесторів, державної підтримки або участі в міжнародних консорціумах. Також важливо передбачити ризики, пов'язані зі змінами економічної кон'юнктури, геополітичними факторами та коливаннями попиту.

У майбутньому авіабудування зосередиться на створенні ще більш енергоефективних і автономних літальних апаратів. Значну роль відіграватиме розвиток штучного інтелекту, який забезпечить новий рівень автоматизації управління та моніторингу систем. Крім того, удосконалення логістичних і постачальницьких ланцюгів сприятиме підвищенню гнучкості та адаптивності виробничих процесів. Очікується, що такі зміни дозволять не лише задовольнити зростаючий попит на авіаційні послуги, а й мінімізувати витрати підприємств на етапах виробництва та експлуатації техніки.

Таким чином, авіабудівні підприємства функціонують у багатогранному середовищі, де інтегруються технології, наукові досягнення, виробничі процеси та бізнес-стратегії. Їхні потреби охоплюють широкий спектр аспектів, від наукових досліджень до фінансової стійкості, кожен із яких відіграє

важливу роль у забезпеченні успішної діяльності та конкурентоспроможності на глобальному ринку.



*Рисунок 1.1 – Виробнича лінія літаків Boeing 737 MAX*

*Джерело: [12]*

## **1.2. Постановка цілей і завдань проєкту**

**Ціль проєкту:** створення інноваційного програмного забезпечення, яке значно покращить процес тестування двигунів літаків Boeing 737 MAX.

*Таблиця 1.1 - Критерії досягнення цілей проєкту*

<b>№</b>	<b>Ціль</b>	<b>Критерій</b>
1	Підвищення ефективності	Автоматизація процесу тестування значно скоротить час, необхідний для проведення всіх необхідних перевірок, та зменшить потребу у ручній праці. Прогнозоване скорочення роботи працівників до 80%.
2	Збільшення точності	Програмне забезпечення забезпечить об'єктивний та детальний аналіз даних, мінімізуючи людський фактор та підвищуючи точність результатів. Прогнозована якість тестування, 99,8%

Головною метою проєкту є створення інструменту, який:

- *збільшить безпеку авіаперельотів* - завдяки детальному та автоматизованому аналізу роботи двигунів, програмне забезпечення виявить потенційні проблеми на ранніх етапах, запобігаючи аваріям;
- *зменшить час та витрати на тестування* - автоматизація процесу скоротить тривалість тестування та зменшить потребу у ручній праці;
- *підвищить точність результатів* - програмне забезпечення забезпечить об'єктивний та детальний аналіз даних, мінімізуючи людський фактор;
- *сприяє розвитку авіаційної галузі* - впровадження інноваційних технологій в процес тестування двигунів сприятиме розвитку авіаційної галузі в цілому.

Основні завдання проєкту можна описати таким чином:

- створення архітектури програмного забезпечення - визначення структури програми, яка дозволить інтегрувати різні модулі та підсистеми. Це включає в себе розробку інтерфейсів для підключення до апаратного забезпечення стендів тестування та двигунів, а також забезпечення гнучкості для оновлень і розширень;
- розробка алгоритмів автоматичного тестування - розробка й оптимізація алгоритмів для проведення різних типів тестів, таких як статичні випробування, тестування під навантаженням, перевірка пускових і зупинних характеристик, а також симуляція екстремальних умов експлуатації. Алгоритми мають враховувати специфікацію двигунів Boeing 737 MAX, включаючи їхню структуру та поведінкові особливості;
- інтеграція з сенсорами та контролерами - підключення програмного забезпечення до датчиків, які вимірюють ключові параметри роботи двигуна, такі як температура, тиск, швидкість обертання турбіни, рівень вібрації тощо [4]. Забезпечення точного збору, передачі та обробки цих даних у реальному часі;

- реалізація системи аналізу даних і звітності - створення модуля для обробки даних тестування, що включає аналіз результатів, виявлення відхилень від норми, генерацію звітів і рекомендацій. Система має підтримувати автоматичну діагностику несправностей на основі зібраних даних;
- забезпечення безпеки та надійності програмного забезпечення - розробка функцій захисту від збоїв, аварійного зупинення двигуна під час тестів, а також шифрування даних для забезпечення їхньої конфіденційності та цілісності;
- тестування й налагодження програмного забезпечення - проведення верифікації та валідації програми, включаючи модульні, інтеграційні та системні тести. Це допоможе забезпечити відповідність програмного забезпечення вимогам замовника та міжнародним стандартам;
- документування системи - створення повного комплексу документації, включаючи технічний опис програми, інструкції з використання, технічне обслуговування та можливе доопрацювання програмного забезпечення;
- навчання персоналу - організація тренінгів і навчальних матеріалів для співробітників, які будуть експлуатувати або підтримувати програму. Це включає ознайомлення з інтерфейсом, функціоналом і можливими сценаріями роботи;
- узгодження з авіаційними стандартами - забезпечення відповідності програмного забезпечення міжнародним стандартам авіаційної галузі, таким як DO-178C (сертифікація програмного забезпечення для авіаційних систем), що є критично важливим для використання на комерційних літаках.

Реалізація цих завдань дозволить створити систему, яка забезпечить ефективне, безпечне та точне тестування двигунів, мінімізує людський фактор у процесі перевірки та підвищить рівень контролю якості продукції.

### 1.3. Визначення вимог до продукту

Вимоги до продукту в проєкті розробки програмного забезпечення для автоматичного тестування двигунів літака Boeing 737 MAX охоплюють широкий спектр технічних, функціональних, користувацьких і нормативних аспектів.

*Функціональні вимоги наступні:*

- автоматизація тестування - програмне забезпечення повинно забезпечувати повністю автоматизоване проведення тестів двигуна без необхідності втручання оператора;
- реальний час роботи - система повинна працювати в режимі реального часу для збору, обробки та аналізу даних із сенсорів двигуна;
- підтримка сценаріїв тестування - програмне забезпечення має включати сценарії для різних видів тестування, таких як статичні, динамічні та спеціальні (наприклад, симуляція екстремальних умов експлуатації);
- діагностика несправностей - інтеграція алгоритмів виявлення та аналізу несправностей на основі зібраних даних про роботу двигуна;
- звітування - генерація детальних звітів із результатами тестування, включаючи графіки, таблиці та висновки.

*Технічні вимоги:*

- сумісність із апаратним забезпеченням - програмне забезпечення повинно підтримувати підключення до існуючих тестових стендів і сенсорів, які використовуються для перевірки двигунів Boeing 737 MAX [5];
- стабільність і надійність - система повинна бути стійкою до збоїв і передбачати можливість аварійного завершення тестів із збереженням даних;
- продуктивність - затримка між зчитуванням даних із сенсорів і їх обробкою не повинна перевищувати допустимих меж для роботи в реальному часі (наприклад, не більше 50 мс);

- масштабованість - система має бути легко адаптованою для роботи з іншими моделями двигунів або розширення функціональності в майбутньому;

- захист даних - реалізація протоколів шифрування для захисту даних від несанкціонованого доступу.

*Користувацькі вимоги наступні:*

- дружній інтерфейс - інтерфейс повинен бути інтуїтивно зрозумілим, із мінімальним набором необхідних дій для запуску тестів і перегляду результатів;

- можливість ручного втручання - передбачити функціонал для ручного запуску та управління тестами в екстрених ситуаціях;

- логування та відстеження - забезпечення прозорості роботи системи через детальне логування всіх виконаних операцій.

*Нормативні вимоги наступні:*

- відповідність стандартам авіаційного програмного забезпечення - система повинна відповідати стандарту DO-178C, який визначає процеси та вимоги до розробки програмного забезпечення для авіаційних систем;

- інтеграція сертифікаційних процедур - забезпечення можливості сертифікації продукту відповідно до норм FAA або EASA;

- безпека та екологічність - врахування вимог до мінімізації впливу на навколишнє середовище під час тестів, наприклад, зниження шумового впливу та контроль викидів.

*Експлуатаційні вимоги представлені наступним:*

- надійність у широкому діапазоні умов - програмне забезпечення повинно функціонувати без збоїв навіть за умов нестабільного електропостачання або високих навантажень на тестовий стенд;

- підтримка оновлень - реалізація механізму регулярного оновлення для вдосконалення функціональності та виправлення помилок.



*Рисунок 1.2 – Літак Boeing 737 MAX  
Джерело: [13]*

*Бізнес-вимоги наступні:*

- зменшення витрат - система повинна сприяти скороченню витрат на проведення тестів за рахунок зменшення ручної праці та прискорення процесів діагностики;
- підвищення якості - гарантувати точність тестів і діагностики, що зменшить ризики несправностей під час експлуатації двигунів.

Ці вимоги є основою для створення продукту, який не лише забезпечить ефективне тестування двигунів, а й відповідатиме високим стандартам авіаційної галузі.

*Аналіз аналогів*

Аналіз аналогів проєкту розробки програмного забезпечення для автоматичного тестування двигунів літака Boeing 737 MAX передбачає вивчення існуючих рішень у сфері тестування авіаційних двигунів. Це

допомагає зрозуміти тенденції, виявити сильні та слабкі сторони аналогів, а також визначити унікальні особливості майбутнього продукту.

*Аналіз існуючих систем автоматичного тестування двигунів*

На ринку представлені різні рішення від провідних авіабудівних компаній, виробників двигунів та постачальників тестового обладнання.

Програмне забезпечення від GE Aviation GE Aviation розробляє спеціалізовані платформи для тестування своїх двигунів, таких як CFM56 і LEAP-1B (Рис 1.3), що використовуються на Boeing 737 MAX [6]:

- сильні сторони - глибока інтеграція з фірмовими сенсорами та системами управління двигуном, реалізація алгоритмів прогнозного обслуговування;
- слабкі сторони - висока вартість адаптації для використання з іншими типами двигунів, залежність від власного обладнання.



*Рисунок 1.3 – Двигун LEAP-1B*

*Джерело:[13]*

Honeywell Aerospace пропонує автоматизовані системи тестування двигунів і авіоніки, що спеціалізуються на великих і малих літаках:

- сильні сторони - модульна архітектура, що дозволяє адаптувати систему для різних типів тестів, високий рівень автоматизації та інтеграція зі стандартами;
- слабкі сторони - можливі обмеження у підтримці нестандартних конфігурацій двигунів.

Rolls-Royce активно використовує цифрові платформи для тестування своїх двигунів Trent:

- сильні сторони - використання штучного інтелекту для аналізу результатів тестування, надійна підтримка сертифікаційних процесів;
- слабкі сторони - орієнтованість лише на власні двигуни, що обмежує гнучкість системи.

Автоматизовані стенди AVL. AVL є постачальником універсальних систем для тестування двигунів різних типів, включаючи авіаційні:

- сильні сторони - підтримка широкого спектру двигунів завдяки модульному дизайну, висока точність і деталізація тестів;
- слабкі сторони - складність інтеграції з унікальними вимогами Boeing 737 MAX.

*Основні виклики аналогів:*

- залежність від фірмового обладнання - багато рішень розроблено виключно для тестування власних двигунів, що робить їх непридатними для універсального використання;
- обмежена гнучкість - більшість систем недостатньо адаптивні до змін вимог або нових сценаріїв тестування;
- висока вартість - розробка та впровадження таких рішень часто потребують значних фінансових вкладень.

## Висновки до розділу 1

У першому розділі було детально проаналізовано авіаційну галузь, її підприємства та їхні потреби, що є фундаментом для створення програмного забезпечення, автоматичного тестування двигунів літака Boeing 737 MAX. Аналіз аналогів демонструє, що ринок має потужні рішення, проте багато з них спеціалізуються на конкретних типах двигунів або виробників.

Унікальні можливості майбутнього продукту можуть полягати у створенні:

- гнучкої системи, яка легко адаптується до різних моделей двигунів;
- інтеграції інноваційних технологій, таких як штучний інтелект і передові методи обробки даних;
- забезпечення універсальності для роботи з різними стендами та умовами тестування.

Це допоможе забезпечити конкурентоспроможність проєкту, задовольняючи потреби не лише Boeing, а й ширшого кола клієнтів.

## РОЗДІЛ 2

### ГНУЧКЕ УПРАВЛІННЯ РОЗРОБКОЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 2.1. Обґрунтування вибору Agile-фреймворку для управління проектом

У сучасних реаліях стрімкого прогресу інформаційних технологій та постійних змін на ринку традиційні підходи до управління проектами, такі як каскадна модель (Waterfall), виявляються все менш ефективними. Це обумовлено їхньою недостатньою гнучкістю перед змінними вимогами та повільністю реагування на нові виклики. Натомість гнучкі методології управління проектами, відомі як Agile, здобувають дедалі більшу популярність завдяки своїй здатності забезпечувати адаптивність, швидкість і ефективність у процесі розробки програмного забезпечення [1].

Agile є набором принципів і методик, які акцентують увагу на ітеративній та поступовій розробці продуктів, підтримуючи тісну співпрацю між командою та замовником, швидку реакцію на зміни та безперервне вдосконалення робочих процесів. Основоположні цінності Agile, викладені у Маніфесті Agile (2001 року) [21], відображають наступні пріоритети:

- людський фактор і взаємодія перевершують значення процесів та інструментів;
- робочий продукт є важливішим, ніж всеосяжна документація;
- співпраця з клієнтом стоїть вище за формальні контрактні домовленості;
- гнучкість у реагуванні на зміни важливіша за суворе дотримання планів.

Рішення застосувати Agile-фреймворк для управління розробкою програмного забезпечення, автоматичного тестування двигунів літака Boeing 737 MAX базується на кількох ключових аспектах. Перш за все, це динамічність і невизначеність вимог. У процесі створення інноваційного продукту, як-от програмне забезпечення, вимоги замовників і користувачів

можуть постійно змінюватися. Agile забезпечує можливість швидко адаптувати процес розробки до актуальних запитів [16].

Другим важливим чинником є необхідність у регулярному зворотному зв'язку. Ітеративний підхід Agile дозволяє створювати робочі версії продукту на кожному етапі розробки, що сприяє отриманню своєчасного фідбеку від усіх зацікавлених сторін. Це, у свою чергу, дозволяє коригувати напрям роботи відповідно до потреб користувачів.

Крім того, Agile орієнтований на створення найбільшої цінності для клієнта, що є критично важливим для успішної роботи застосунку на конкурентному ринку. Підхід спрямований на максимальну вигоду для кінцевого споживача, забезпечуючи оперативну реакцію на зміну ринкових умов і запитів. Особливо слід відзначити, що Agile покращує внутрішню комунікацію в команді, сприяючи відкритому обговоренню проблем і оперативному прийняттю рішень. Це не лише підвищує ефективність співпраці, а й створює більш продуктивне середовище для розробки.

Для успішного управління проектом важливо вибрати Agile-фреймворк, який найкраще відповідає потребам команди та специфіці самого проекту. Далі буде проаналізовано два популярні фреймворки – Scrum і Kanban, щоб визначити, який з них є найбільш доцільним для реалізації програмного забезпечення [10].

Scrum є підходом, що структуровано розбиває роботу на фіксовані часові періоди, відомі як спринти, які зазвичай тривають до чотирьох тижнів [19].

У Scrum передбачено три основні ролі:

- Власник продукту (Product Owner) - відповідає за управління беклогом і забезпечення максимальної цінності продукту;
- Scrum-майстер (Scrum Master) - гарантує дотримання принципів та процесів Scrum, сприяючи ефективності роботи команди;
- розробницька команда - займається створенням інкрементів продукту, які є завершеними етапами розробки.

Основними артефактами Scrum є:

- беклог продукту - перелік усіх вимог і функціональностей, які потрібно реалізувати;
- беклог спринту - завдання, що визначають обсяг роботи на конкретний спринт;
- інкремент продукту - завершена та готова до використання частина продукту, створена за результатами кожного спринту.

Процес Scrum передбачає кілька важливих подій:

- планування спринту - команда обирає завдання з беклогу продукту для реалізації протягом наступного спринту;
- щоденні стендапи - короткі зустрічі для обговорення прогресу, перешкод і подальших дій;
- огляд спринту - представлення результатів роботи спринту зацікавленим сторонам;
- ретроспектива - аналіз виконаної роботи та визначення шляхів для вдосконалення в майбутніх спринтах.

Вибір Scrum [8] або Kanban залежить від специфічних потреб команди та проєкту, але Scrum забезпечує чітку структуру ітераційної роботи, яка може бути надзвичайно корисною для розробки програмного забезпечення, автоматичного тестування двигунів літака Boeing 737 MAX.

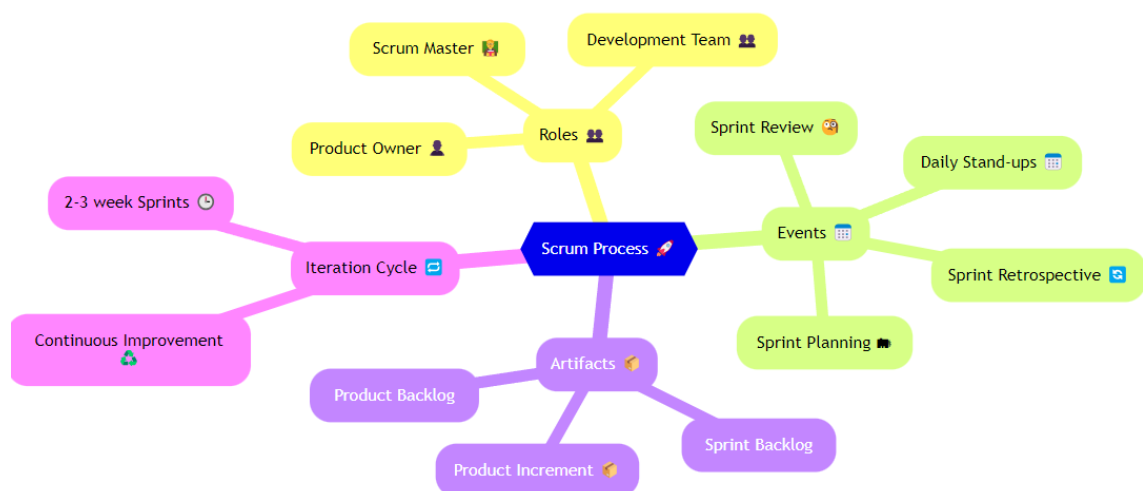


Рисунок 2.1 – Схема процесу Scrum

Джерело: розроблено автором на основі даних з веб-порталу [1]

**Kanban** – це методологія, орієнтована на візуалізацію робочих процесів і контроль кількості незавершених завдань. [2] Ключові особливості Kanban включають:

- відсутність чітко визначених ітерацій;
- використання візуальних інструментів для управління процесами;
- встановлення лімітів на кількість одночасно виконуваних завдань (WIP).

Чому для проєкту розробки програмного забезпечення обрано Scrum? Scrum пропонує структурований підхід до управління проєктами, який включає чітко визначені ролі, події та артефакти. Це особливо корисно для команд, які вперше впроваджують Agile-методологію, оскільки забезпечує зрозумілу організацію робочих процесів.

Розподіл роботи на короткі цикли – спринти – дозволяє регулярно отримувати функціональні версії продукту. Це є критично важливим для своєчасного тестування та збору зворотного зв'язку. Ретроспективи, що проводяться після завершення кожного спринту, дають змогу визначити проблеми та вдосконалити процеси розробки.

Крім того, активна участь власника продукту (Product Owner) у визначенні пріоритетів і тісній взаємодії з командою забезпечує створення продукту з максимальною цінністю.

Переваги використання Scrum для нашого випадку:

1. Ітеративна розробка продукту - Scrum дає змогу швидко створити мінімально життєздатний продукт (MVP) і представити його користувачам. Це сприяє пришвидшенню виходу на ринок і збору реального зворотного зв'язку.
2. Гнучкість у адаптації до змін - завдяки можливості оновлювати беклог між спринтами команда може оперативного реагувати на нові вимоги, зміну пріоритетів чи зовнішні обставини.
3. Підвищення якості продукту - регулярне тестування кожного інкременту дозволяє виявляти та виправляти помилки на ранніх етапах, що мінімізує ризики появи серйозних дефектів на фінальній стадії.

4. Прозорість і довіра - постійний моніторинг прогресу та відкриті комунікації між командою й зацікавленими сторонами забезпечують прозорість процесу розробки, що сприяє зміцненню взаємної довіри.

5. Зростання мотивації команди - участь команди в прийнятті рішень і розподілі відповідальності за кінцевий результат позитивно впливає на залученість і мотивацію співробітників.

Таким чином, застосування Scrum у контексті розробки додатку для тестування дозволяє ефективно організувати процес розробки, забезпечити високу якість продукту та швидко реагувати на потреби ринку й клієнтів.

Важливим компонентом проєкту є залучення галузевих експертів. Product Owner може тісно співпрацювати з фахівцями у сфері авіабудування для уточнення вимог до продукту та визначення ключових напрямів розвитку функціональності. Це дозволяє створити продукт, який повною мірою відповідає актуальним запитам ринку.

#### *Можливі виклики та способи їх подолання*

Одним з ключових аспектів успішного впровадження Scrum є ретельне планування та організація роботи. Неправильна оцінка обсягу роботи в рамках спринту може призвести до перевантаження команди та зниження якості результатів. Для уникнення цього необхідно використовувати ефективні методи оцінки складності задач, а також регулярно переглядати та коригувати плани. Важливо також забезпечити, щоб кожен член команди чітко розумів свої завдання та відповідальність.

## **2.2. Планування проєкту розробки додатку, тестування двигунів літака Boeing 737 MAX : етапи, бюджет та команда**

### *Етапи проєкту та переліку робіт*

#### **1. Аналіз вимог та збір даних:**

- визначення параметрів тестування: які саме аспекти роботи двигуна потрібно перевіряти? Це можуть бути показники потужності, витрати палива, рівень вібрації, температура, тиск тощо;

- створення детального технічного завдання - чітко сформулювати всі функціональні та нефункціональні вимоги до програмного забезпечення;
- інтеграція з існуючими системами - як програмне забезпечення буде взаємодіяти з іншими системами управління літаком та збору даних?
- розробка тест-кейсів - створення набору тестів, які охоплюють всі можливі сценарії роботи двигуна.

## **2. *Проектування архітектури системи:***

- вибір технологічного стеку - які мови програмування, фреймворки та інструменти будуть використані?
- розробка модульної структури - розбиття системи на окремі модулі для спрощення розробки, тестування та підтримки;
- визначення інтерфейсів - визначення того, як різні компоненти системи будуть взаємодіяти між собою.

## **3. *Розробка програмного забезпечення:***

- реалізація модулів - написання коду для кожного модуля відповідно до проєктної документації;
- інтеграція модулів - з'єднання всіх модулів в єдину систему;
- розробка інтерфейсу користувача - створення зручного інтерфейсу для взаємодії з системою.

## **4. *Тестування:***

- модульне тестування - перевірка кожного модуля окремо;
- інтеграційне тестування: перевірка взаємодії модулів між собою;
- системне тестування: перевірка всієї системи в цілому;
- тестування на відповідність вимогам - перевірка того, що система виконує всі поставлені завдання.

## **5. *Впровадження та підтримка:***

- інсталяція системи - встановлення програмного забезпечення на тестові стенди та літаки;
- навчання користувачів - навчання інженерів та техніків працювати з системою;

○ супровід та оновлення - регулярне оновлення програмного забезпечення та надання технічної підтримки.

*Зміст робіт:*

- модуль збору даних - збір даних з датчиків двигуна (температура, тиск, вібрація тощо);
- модуль обробки даних - обробка зібраних даних, фільтрація шумів, перетворення в необхідний формат;
- модуль порівняння з еталонними даними - порівняння отриманих даних з еталонними значеннями для виявлення відхилень;
- модуль візуалізації даних - відображення результатів тестування у зручному для сприйняття вигляді (графіки, діаграми);
- модуль генерації звітів - автоматичне створення звітів про результати тестування;
- інтерфейс користувача - інтуїтивно зрозумілий інтерфейс для управління системою та перегляду результатів;
- база даних - зберігання результатів тестування для подальшого аналізу.

*Ключові технології та інструменти:*

- мови програмування: - Python, C++, MATLAB;
- фреймворки - TensorFlow, PyTorch (для машинного навчання), Django, Flask (для веб-розробки);
- бази даних - PostgreSQL, MongoDB;
- інструменти для тестування - pytest, unittest;
- інструменти для версіонування - Git;
- інструменти для CI/CD - Jenkins, GitLab CI/CD.

*Виклики та ризики:*

- складність системи - велика кількість взаємопов'язаних компонентів;
- високі вимоги до надійності - будь-яка помилка в програмному забезпеченні може призвести до серйозних наслідків;

- безпека - захист системи від несанкціонованого доступу;
- сумісність з існуючими системами - інтеграція з різноманітним обладнанням.

### *Планування термінів проєкту*

Планування термінів проєкту є важливим етапом, що забезпечує ефективну організацію робіт, контроль за виконанням завдань та досягнення цілей у встановлені строки. Для проєкту варто враховувати ключові фази, залежності між завданнями, доступність ресурсів і ризику.

*Таблиця 2.1 – Календарне планування проєкту*

№ 1-го рівня	№ 2-го рівня	№ 3-го рівня	Назва задачі	Оцінка тривалості	Ресурси	Оцінка вартості
<b>1</b>			<b>Розробка програмного забезпечення</b>	<b>434 дні</b>	T	555 тис. дол.
	<b>1.1</b>		<b>Фаза ініціації</b>	<b>104 днів</b>	T1, T3	62 тис. дол.
		1.1.1	Аналіз готових рішень від інших компаній	24 дні	T1	12 тис. дол
		1.1.3	Формування ТЗ	42 дні	T1	26 тис. дол
		1.1.4	Формування команди проєкту	23 дні	T1, T3	9 тис. дол.
		1.1.5	Найм підрядників в разі потреби	15 днів	T1, T3	15 тис. дол.
	<b>1.2</b>		<b>Фаза підготовки</b>	<b>63 днів</b>	T1, P1	116 тис. дол
		1.2.1	Планування бюджету	21 день	T1, P1	15 тис. дол
		1.2.2	Планування ресурсів	21 день	T1	15 тис. дол
		1.2.3	Планування якості	22 дні	T1	16 тис. дол
		1.2.4	Планування комунікацій	12 дні	T1	6 тис. дол

Таблиця 2.1 Календарне планування проєкту

№ 1-го рівня	№ 2-го рівня	№ 3-го рівня	Назва задачі	Оцінка тривалості	Ресурси	Оцінка вартості
		1.2.5	Планування людських ресурсів	12 днів	T1	15 тис. дол.
		1.2.6	Розробка проєктної документації	63 дні	T1	49 тис. дол.
	<b>1.3</b>		<b>Фаза реалізації</b>	<b>220 днів</b>	T1, T2, T3	325 тис. дол.
		1.3.7	Найм персоналу, трансфер з інших команд	20 днів	T1, T3	5 тис. дол.
		1.3.8	Створення продукту, та тестування	200 днів	T1, T2, T3	320 тис. дол.
	<b>1.4</b>		<b>Фаза завершення</b>	<b>47 днів</b>	T1, T2, T3	52 тис. дол.
		1.4.1	Аналіз якості	12 днів	T1, T2	23 тис. дол.
		1.4.3	Архівування проєкту	21 день	T1	13 тис. дол.
		1.4.4	Запровадження проєкту в роботу	12 днів	T1, T2, T3	14 тис. дол.
		1.4.5	Розпуск команди	1 день	T1	1 тис. дол.
		1.4.6	Закриття проєкту	1 день	T1	1 тис. дол.

Під час фази підготовки деякі етапи виконуються одночасно, тому кількість днів не сумується.

Таблиця 2.2. Фази проєкту

№ фази	Назва	Критерії переходу на іншу фазу	Оцінка тривалості	Оцінка вартості*
1	Фаза ініціації	Команда проєкту сформована. T3 сформовано	104 дні	62 тис. дол.

Таблиця 2.2. Фази проєкту

№ фази	Назва	Критерії переходу на іншу фазу	Оцінка тривалості	Оцінка вартості*
2	Фаза підготовки	Готова проєктна документація	63 днів	116 тис. дол
3	Фаза реалізації	Продукт створений та протестований.	220 дні	325 тис. дол.
4	Фаза завершення	Робота оплачена. Проєкт закрито	47 днів	52 тис. дол
Всього проєкт			434 днів	555 тис. дол.

*Джерело: розроблено автором*

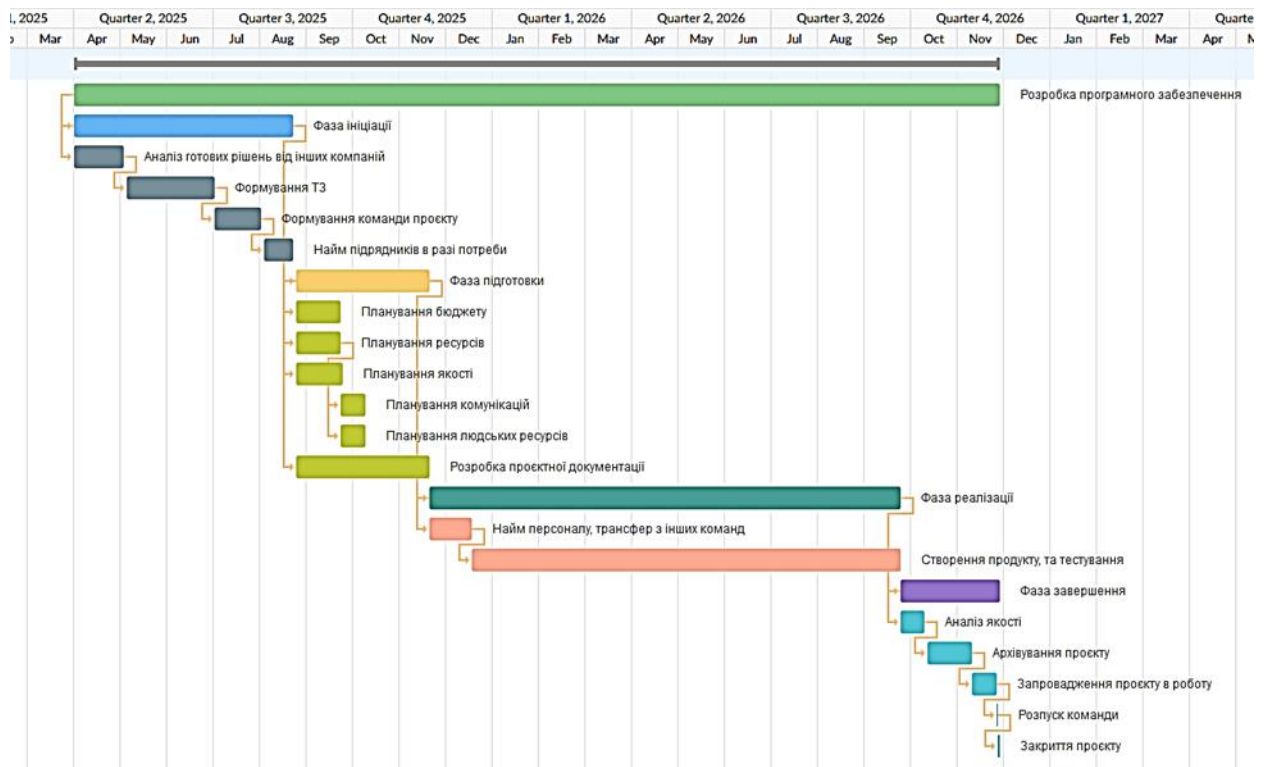


Рисунок 2.2 - Діаграма Ганта

*Джерело: розроблено автором*

Планування робіт у проєкті є важливим етапом, який дозволяє систематизувати всі завдання та процеси, забезпечуючи їх виконання у

визначені строки та з урахуванням наявних ресурсів. Для цього було застосовано діаграму Ганта (рис. 2.2), яка допомагає не лише візуалізувати кожен етап проєкту, а й зрозуміти взаємозалежність між окремими завданнями.

Ця діаграма чітко демонструє послідовність робіт: від аналізу проблем і визначення цілей до створення продукту й проведення ітераційного вдосконалення. Завдяки встановленню часових меж для кожного етапу вдається ефективно відстежувати прогрес виконання й уникати ризику затримок.

Ключові етапи, як-от аналіз досвіду інших організацій, оцінка зовнішніх умов, пошук ресурсів і формування організаційної структури, відіграють визначальну роль у розробці стратегії проєкту. Вони створюють основу для розподілу завдань і визначення необхідних ресурсів для кожного етапу роботи.

#### *Організаційна структура команди проєкту*

Ключові ролі та відповідальність:

- Менеджер проєкту:
  - загальне керівництво проєктом;
  - планування, координація та контроль виконання робіт;
  - управління бюджетом та ресурсами;
  - комунікація зі замовником та іншими стейкхолдерами.
- Технічний директор:
  - визначення технічної стратегії проєкту;
  - розробка архітектури системи;
  - контроль якості розробки.
- Scrum Master: фасилітатор процесу, відповідає за дотримання

принципів Scrum

- Програмні інженери:
  - розробка програмного коду;
  - тестування окремих модулів.
- Інженери з авіаційної техніки:

- розробка тест-кейсів;
  - аналіз результатів тестування;
  - інтеграція програмного забезпечення з фізичними системами.
  - Тестувальники:
    - розробка тест-планів та тест-кейсів.;
    - проведення функціонального, навантажувального та регресійного тестування.
  - Фахівець з баз даних:
    - проектування та розробка бази даних для зберігання результатів тестування.
  - Фахівець з безпеки:
    - забезпечення безпеки програмного забезпечення та захисту даних.
- Людські ресурси наведені у табл 2.3.

*Таблиця 2.3. – Людські ресурси*

№	Назва ресурсу	Оцінка вартості
T1	Команда менеджерів, Scrum Master	~16 тис. дол/міс
T2	Відділ тестування та ІТ фахівці, інженери	~53 тис. дол./міс
T3	Відділ кадрів	~3 тис. дол/міс.
P1	Бухгалтерський відділ	~4 тис. дол/міс

Все обладнання для роботи та тестування вже наявне в матеріальній базі компанії, додаткові витрати не передбачені

*Принципи побудови структури:*

- *функціональність* - кожен член команди має чітко визначені обов'язки;
- *гнучкість* - структура повинна дозволяти адаптуватися до змін у проєкті;
- *комунікація* - ефективна комунікація між усіма членами команди;

- *відповідальність* - кожен член команди несе відповідальність за свою частину роботи.

*Фактори, що впливають на структуру:*

- *розмір команди* - для великих проєктів може знадобитися додаткова ієрархія;
- *географічне розташування* - якщо команда розподілена географічно, можуть бути необхідні додаткові засоби комунікації та координації;
- *фази проєкту* - структура команди може змінюватися в залежності від фази проєкту.

*Важливі аспекти управління командою:*

- *регулярні зустрічі* - для обговорення прогресу, вирішення проблем та координації дій;
- *системи відстеження завдань* - для контролю виконання завдань і термінів;
- *інструменти для співпраці* - для забезпечення ефективної роботи в команді;
- *мотивація* - створення сприятливого середовища для роботи та розвитку співробітників.

Організаційна структура команди проєкту є одним з ключових факторів успіху. Вона повинна бути гнучкою, адаптивною і забезпечувати ефективну співпрацю всіх учасників.

### **2.3 Фактори макросередовища**

Проведемо аналіз чинників маркетингового зовнішнього середовища організації. Зовнішнє середовище (макро- та мікро- маркетингове середовище) організації формується чинниками, що є об'єктивними та впливають на досягнення поставленої в проєкті мети.

Критерії впливу факторів визначимо наступним чином:

1-3 бали – не впливають фактори

4-6 бали – мінімальний вплив факторів

7-8 бали – впливають фактори

9-10 бали – максимальний вплив факторів

Оцінку впливу факторів макросередовища на розробку програмного забезпечення, автоматичного тестування двигунів літака Boeing 737 MAX проведемо у табл. 2.4

*Таблиця 2.4. – Оцінка впливу факторів макросередовища*

<b>Фактор макросередовища</b>	<b>Опис</b>	<b>Оцінка впливу</b>	<b>Обґрунтування</b>
<b>Економічні фактори</b>			
Стан економіки	Загальний стан економіки, рівень ВВП, інфляція, безробіття	8	Економічна стабільність впливає на інвестиційну привабливість проекту та доступність фінансування.

*Таблиця 2.4. – Оцінка впливу факторів макросередовища*

<b>Фактор макросередовища</b>	<b>Опис</b>	<b>Оцінка впливу</b>	<b>Обґрунтування</b>
Валютні курси	Динаміка валютних курсів може вплинути на вартість імпортованих компонентів та послуг.	6	При наявності імпортованих компонентів, зміни курсів можуть вплинути на загальну вартість проекту.
Податкова політика	Податкові ставки, пільги для інноваційних компаній	7	Податкова політика може стимулювати або стримувати інвестиції в розробку нових технологій.
<b>Соціальні фактори</b>			
Демографічні зміни	Зміни у віковій структурі населення, рівень урбанізації	5	Може вплинути на потреби ринку та доступність кваліфікованої робочої сили.
Рівень освіти	Рівень освіти населення, доступність до вищої освіти	7	Від рівня освіти залежить доступність кваліфікованих фахівців для проекту.
Культурні цінності	Культурні цінності суспільства, ставлення до технологій	6	Культурні цінності можуть вплинути на сприйняття інноваційних рішень.
<b>Політичні фактори</b>			
Політична стабільність	Рівень політичної стабільності в країні	8	Політична нестабільність може створити додаткові ризики для інвестицій.
Правове регулювання	Законодавство, що регулює авіаційну галузь,	9	Правове регулювання безпосередньо впливає на

	захист інтелектуальної власності		умови реалізації проєкту та його подальшу експлуатацію.
Міжнародні відносини	Міжнародні договори, санкції	7	Міжнародні відносини можуть вплинути на доступність технологій та ринків збуту.

Таблиця 2.4 – Оцінка впливу факторів макросередовища

Фактор макросередовища	Опис	Оцінка впливу	Обґрунтування
<b>Технологічні фактори</b>			
Темпи розвитку технологій	Швидкість розвитку комп'ютерних технологій, штучного інтелекту	9	Швидкий розвиток технологій створює як нові можливості, так і нові виклики для проєкту.
Доступність технологій	Доступність необхідного обладнання, програмного забезпечення та інструментів розробки	8	Від доступності технологій залежить вартість та швидкість реалізації проєкту.
<b>Екологічні фактори</b>			
Екологічне законодавство	Вимоги до екологічної безпеки, викидів	6	Екологічні вимоги можуть вплинути на вартість проєкту та його реалізацію.
Громадська думка щодо екологічних проблем	Свідомість населення щодо екологічних проблем	5	Може вплинути на репутацію компанії та прийняття проєкту суспільством.

*Джерело: розроблено автором*

Економічні та політичні чинники значно впливають на розробку програмного забезпечення, автоматичного тестування двигунів літака Boeing 737 MAX, створюючи сприятливі умови для розвитку бізнесу і визначають специфіку попиту та адаптацію бізнес-стратегії.

Таблиця 2.5 – Ситуаційна оцінка стейкхолдерів

Критерії оцінки	Оцінка стейкхолдерів	
	Стейкхолдер 1 (Boeing company)	Стейкхолдер 2 (US Government)
Досвід роботи	Високий	Високий
Імідж	Середній	Високий
Тенденції розвитку	Стабільність	Зріст
Потенціал розвитку	Високий	Високий

На підставі експертних оцінок інформації по представленим стейкхолдерам зведена аналітична інформація представлена в таблиці 2.5. Дані свідчать про високий потенціал розвитку всіх розглянутих зацікавлених сторін.

## **Висновки до розділу 2**

У другому розділі було глибоко проаналізовано підходи до гнучкого управління процесом розробки програмного забезпечення для автоматизованого тестування двигунів літака Boeing 737 MAX. Здійснено обґрунтування вибору Agile-фреймворку, зокрема методології Scrum, яка демонструє переваги ітеративного та інкрементального підходу в умовах швидкозмінного ринку та постійно оновлюваних вимог. Такий підхід забезпечив створення адаптивного плану розробки, що враховує потреби користувачів.

Процес планування охоплював визначення масштабів робіт, формування реалістичних строків виконання та оцінку бюджету. Уважне опрацювання цих параметрів дозволило забезпечити контрольованість усіх етапів розробки та зберегти фінансову стабільність проєкту. Розробка чіткої рольової структури команди із визначеним розподілом завдань підвищила ефективність співпраці та продуктивність. Залучення крос-функціональних команд дало змогу оптимізувати робочі процеси та гарантувати високий рівень реалізації запланованих функцій.

Особливий акцент було зроблено на управлінні ризиками та забезпеченні високих стандартів якості. Ідентифікація можливих загроз і розробка заходів для їх мінімізації допомогли зменшити негативний вплив на перебіг проєкту. Захист даних і контроль доступу стали ключовими аспектами, зважаючи на необхідність роботи з конфіденційною інформацією клієнтів і підприємств. Технічні та організаційні заходи гарантували відповідність нормативним вимогам і зміцнили довіру користувачів до платформи.

Комунікаційна стратегія, яка включала регулярні зустрічі, використання сучасних інструментів зв'язку та організацію чітких каналів взаємодії, сприяла

ефективній співпраці як всередині команди, так і з зовнішніми партнерами. Такий підхід забезпечив оперативність у прийнятті рішень і гнучкість у реагуванні на зміни, що є надзвичайно важливим для успіху Agile-проектів.

Програма навчання та розвитку команди була спрямована на постійне підвищення кваліфікації учасників. Інвестиції у професійний ріст, участь у галузевих заходах і створення внутрішніх менторських програм розширили компетенції команди. Це сприяло підвищенню якості розробки та впровадженню інноваційних рішень, що позитивно вплинуло на кінцевий продукт.

## РОЗДІЛ 3

### РЕЗУЛЬТАТИ РОЗРОБКИ ТА ОПТИМІЗАЦІЯ ПРОДУКТУ

#### 3.1. Аналіз результативності та ретроспектива роботи команди

Ефективна організація та планування роботи команди є визначальними для успішного виконання проєкту з розробки програмного забезпечення, автоматичного тестування двигунів літака Boeing 737 MAX. Завдяки комплексному підходу до формування команди, впровадженню сучасних гнучких методологій та використанню інноваційних інструментів, стало можливим створення якісного продукту, який відповідає вимогам ринку.

Проєктна команда включає фахівців різних напрямків, що забезпечує комплексне покриття всіх етапів розробки. Розробка організована за принципами Agile, з використанням Scrum. Цей підхід дозволяє працювати короткими двотижневими спринтами, що дає змогу швидко реагувати на зміни та отримувати оперативний зворотний зв'язок. Щоденні зустрічі підтримують команду в курсі прогресу, а ретроспективи після кожного спринту дозволяють оцінити досягнення та визначити напрями для покращення. Регулярне планування спринтів допомагає оптимально розподіляти ресурси й встановлювати пріоритети [3].

Ефективна комунікація є ключовою для злагодженої роботи команди. У роботі використовуються сучасні інструменти:

- Slack і Microsoft Teams для швидкої взаємодії в реальному часі.
- Jira для управління беклогом, відстеження виконання завдань і планування спринтів.
- Confluence для спільної роботи над документацією, що дозволяє зберігати та оновлювати важливі матеріали.

Ці засоби сприяють оперативному обміну інформацією, вирішенню нагальних питань і забезпечують прозорість процесу розробки. Спільна робота фахівців різних профілів дозволяє оперативно вирішувати технічні завдання та покращувати якість продукту.

Орієнтація на потреби кінцевих користувачів є ще одним важливим компонентом роботи. Команда активно залучає клієнтів до процесу розробки через інтерв'ю, тестування прототипів і аналіз зворотного зв'язку. Це забезпечує чітке розуміння ринкових потреб та оперативне впровадження необхідних змін.

*Основні етапи взаємодії з користувачами:*

1. Проведення інтерв'ю та опитувань для збору відгуків і вивчення потреб.
2. Тестування прототипів для оцінки відповідності очікуванням користувачів.
3. Аналіз отриманого зворотного зв'язку після впровадження нових функцій.

У процесі роботи можливі виклики, наприклад, недостатньо деталізовані завдання в беклозі, що може призвести до неправильного трактування вимог і затримок. Для вирішення цієї проблеми бізнес-аналітики та продакт-менеджер забезпечують чіткий аналіз вимог і формують зрозумілі критерії прийняття задач.

Ще одним потенційним викликом є накопичення технічного боргу, що ускладнює подальший розвиток продукту. Для його мінімізації передбачено виділення часу на рефакторинг і регулярне проведення код-рев'ю. Проблеми з освоєнням нових технологій або методологій вирішуються через організацію тренінгів і менторство, що допомагає швидко підвищувати кваліфікацію команди.

Захист даних є критичним аспектом роботи, оскільки недотримання вимог безпеки може спричинити юридичні проблеми. Для цього впроваджуються політики інформаційної безпеки, проводяться регулярні аудити та навчання команди з кібербезпеки.

Щоб підтримувати ефективність команди, важливо приділяти увагу управлінню ресурсами та контролю за завантаженістю працівників. Перевантаження команди може призвести до вигорання, зниження

продуктивності та помилок у виконанні завдань. Для уникнення цих ризиків проводиться регулярний моніторинг завантаженості кожного члена команди та оптимізація робочих процесів через використання таких інструментів, як Kanban або Scrum.

Окремою складовою успішного виконання проєкту є управління ризиками. Ідентифікація можливих загроз на ранніх етапах дозволяє уникати значних проблем у майбутньому. Наприклад, проєкти можуть стикатися з ризиками затримки через зовнішні чинники, як-от залежність від сторонніх постачальників або зміни ринкових умов. Для ефективного управління ризиками формуються резервні плани та створюються протоколи для швидкого реагування на непередбачені обставини.

Чітка організація процесів, орієнтація на користувача та ефективне вирішення викликів забезпечують успішність проєкту. Взаємодія команди, інноваційні підходи та ретельне планування сприяють створенню конкурентоспроможного продукту, який відповідає потребам ринку.

*Планування спринту* – це ключовий етап у Scrum, який визначає роботу команди на наступний ітераційний період. Давайте розглянемо цей процес детальніше.

#### 1. Підготовка до планування:

- оновлення беклогу - регулярне оновлення та пріоритизація елементів беклогу (Product Backlog) забезпечує актуальність планування;
- аналіз минулого спринту - вивчення результатів попереднього спринту допомагає виявити сильні сторони, слабкі місця та внести необхідні корективи;
- визначення цілей - спільне з Product Owner формулювання конкретних і вимірюваних цілей спринту (Sprint Goal) забезпечує фокус команди.

#### 2. Зустріч планування спринту (Sprint Planning):

- вибір завдань - команда спільно з Product Owner обирає завдання з беклогу, які будуть виконані в цьому спринті, враховуючи пріоритети та доступні ресурси;
- уточнення деталей - для кожного завдання уточнюються деталі, критерії прийняття (Definition of Done) та розробляється план виконання;
- оцінка трудомісткості - команда оцінює трудомісткість кожного завдання, використовуючи обрану шкалу (наприклад, Story Points);
- визначення можливостей - визначається загальний обсяг роботи, який команда може виконати за спринт (team capacity).

### 3. Формування беклогу спринту (Sprint Backlog):

- створення списку - створюється деталізований список завдань, які команда зобов'язується виконати протягом спринту;
- визначення залежностей - визначаються взаємозв'язки між завданнями, щоб уникнути блокувань;
- пріоритизація - завдання ранжуються за важливістю та залежностями;

### 4. Визначення критеріїв завершеності (Definition of Done):

- уточнення для кожного завдання - для кожного завдання чітко визначаються критерії, за якими воно вважається завершеним;
- узгодження загальних вимог - визначаються загальні вимоги до якості для всіх завдань спринту.

### 5. Оцінка ризиків:

- ідентифікація - виявляються потенційні загрози, які можуть вплинути на виконання спринту;
- розробка стратегій - розробляються плани дій для мінімізації ризиків.

### 6. Візуалізація плану:

- створення дошки - створюється Scrum-дошка (фізична або цифрова) для візуалізації завдань та їхнього стану;
- розміщення завдань - завдання розміщуються на дошці відповідно до їхнього стану (To Do, In Progress, Done).

#### 7. Завершення планування:

- підтвердження зобов'язань - команда підтверджує, що готова взяти на себе відповідальність за виконання завдань спринту;
- планування комунікацій - визначаються канали комунікації та частота зустрічей для координації роботи протягом спринту;

#### 8. Адаптація до змін у процесі спринту:

- оперативне реагування. Команда повинна бути готовою до коригування планів у разі змін пріоритетів або виникнення непередбачених обставин;
- гнучкість. Додатково узгоджуються межі можливих змін, щоб вони не порушували основну мету спринту.

#### 9. Відображення прогресу:

- щоденні Scrum-зустрічі. Короткі щоденні зустрічі (Daily Scrum) допомагають відстежувати прогрес команди та визначати перешкоди;
- оновлення Scrum-дошки. Регулярне оновлення стану завдань на дошці забезпечує прозорість процесу.

#### 10. Контроль ефективності:

- моніторинг виконання. У процесі спринту регулярно проводиться аналіз витраченого часу та ресурсів для виявлення можливих затримок;
- використання метрик. Метрики, такі як Velocity, допомагають оцінити продуктивність команди та співвідношення обсягу виконаних завдань до спланованих.

#### 11. Ретроспектива та підбиття підсумків:

- аналіз досягнень. По завершенні спринту проводиться ретроспективна зустріч, де команда обговорює, які процеси працювали добре, а які потребують вдосконалення.

- застосування отриманого досвіду. Результати ретроспективи використовуються для покращення процесів у наступних спринтах.

#### 12. Залучення зацікавлених сторін (Stakeholders):

- прозорість. Перед початком спринту доцільно коротко проінформувати зацікавлені сторони про ключові цілі;

- відкритість до відгуків. Під час спринту важливо створити умови для зворотного зв'язку від стейкхолдерів, щоб врахувати їхні рекомендації.

#### 13. Інструменти планування:

- вибір відповідного інструмента. Для організації планування можуть використовуватися сучасні цифрові платформи, як-от Jira, Trello, Asana або Monday, які дозволяють візуалізувати беклог і спринт;

- автоматизація процесів. Інтеграція з іншими системами, такими як репозиторії коду або системи моніторингу, сприяє автоматизації відстеження стану виконання завдань.

#### 14. Командна співпраця та мотивація:

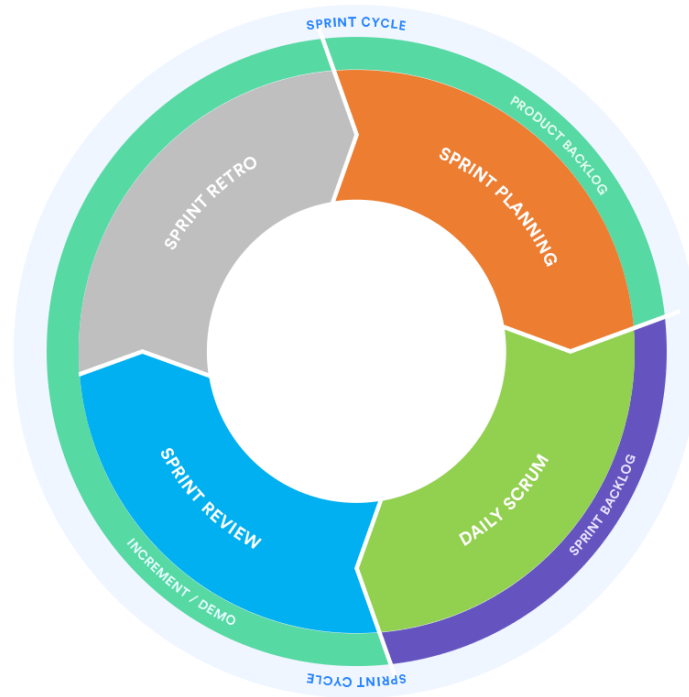
- створення умов для ефективної взаємодії. Під час планування важливо враховувати навантаження кожного члена команди, щоб уникнути перевтоми;

- формування командного духу. Обговорення цілей спринту та взаємної підтримки сприяє підвищенню мотивації команди.

#### 15. Оцінка готовності до спринту:

- перевірка інфраструктури. Перед початком спринту переконуються, що всі необхідні інструменти, середовища розробки та доступи налаштовані для безперебійної роботи;

- готовність завдань. Завдання, які не пройшли уточнення чи мають невизначені вимоги, можуть бути перенесені в майбутні спринти.



*Рисунок 3.1 - Схема планування спринту*

*Джерело: [14]*

### **3.2 Виявлення проблем та шляхів їх вирішення**

Під час ретроспективи команда не лише виявила проблеми, але й запропонувала комплексні рішення, які візуалізувала за допомогою діаграми Ішікави. Цей метод допоміг детально структурувати першопричини проблем та визначити ефективні способи їх усунення.

Ключові етапи аналізу включали:

- групування причин за категоріями - команда розподілила основні фактори впливу на проблеми за напрямками, такими як "процеси", "ресурси", "технології" та "комунікація";
- поглиблений аналіз першопричин: обговорення виявлених факторів дозволило не лише зрозуміти, що саме спричинило проблему, але й оцінити її вплив на інші аспекти проєкту;
- пріоритизація рішень: для кожної проблеми було запропоновано декілька шляхів вирішення, які ранжувалися за критеріями ефективності, термінів реалізації та витрат.

Використання діаграми Ішікави (рис. 3.2) сприяло більш глибокому розумінню причинно-наслідкових зв'язків у роботі команди. Це забезпечило впровадження обґрунтованих рішень, які не лише допомагають усунути поточні недоліки, а й запобігти виникненню аналогічних проблем у майбутньому.

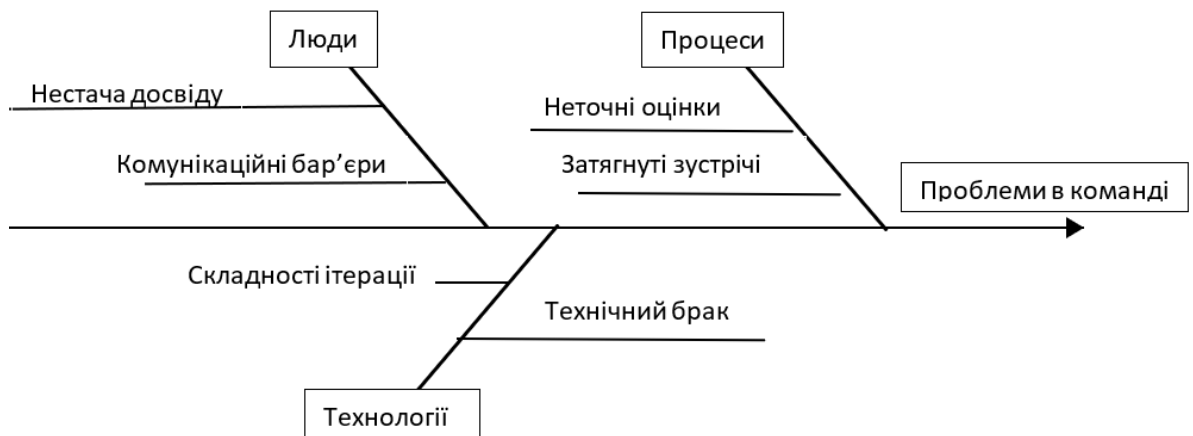


Рисунок 3.2 – Діаграма Ішікави

Джерело: розроблено автором

### 3.3 Методи підвищення продуктивності Scrum-команди

Щоб створити більш ефективну та злагоджену Scrum-команду, ми впровадили ряд інноваційних практик. Зокрема:

- адаптивний підхід до роботи – методика Pomodoro дозволила кожному розробнику знайти оптимальний ритм роботи, що позитивно вплинуло на загальну продуктивність;
- командна співпраця – парне програмування та регулярні code review сприяли обміну знаннями, підвищенню якості коду та формуванню культури взаємодопомоги;
- безперервне навчання – завдяки навчальним ініціативам, команда стала більш адаптивною до змін та відкритою до нових технологій;
- автоматизація процесів – впровадження автоматизованого тестування дозволило звільнити розробників від рутинної роботи та зосередитися на більш творчих завданнях;

- ефективне планування задач – запровадження чіткої структури під час планування спринтів із розподілом задач за пріоритетами. Це допомогло уникати перевантаження команди й забезпечити зосередження на критичних завданнях;
- ретроспективи з фокусом на вдосконалення – удосконалення ретроспектив шляхом впровадження методу «Start, Stop, Continue», що сприяло виявленню ефективних практик та усуненню слабких місць;
- гейміфікація процесів – запровадження елементів гейміфікації, таких як командні досягнення за швидке виконання задач або високу якість роботи. Це мотивувало учасників команди та сприяло здоровій конкуренції;
- розвиток soft skills – проведення тренінгів з комунікації, вирішення конфліктів і розвитку емоційного інтелекту дозволило команді ефективніше співпрацювати та оперативно вирішувати робочі суперечності;
- оптимізація зустрічей – скорочення тривалості непотрібних зустрічей і запровадження чітких регламентів для зборів, щоб уникати відволікань і зосереджуватися на обговоренні суті;
- прозорість процесів – використання інтерактивної Scrum-дошки та загальнодоступних звітів про стан виконання завдань сприяли підвищенню прозорості та дозволили уникати непорозумінь;
- менторство в команді – призначення менторів для нових учасників команди допомогло швидко адаптуватися до робочих процесів та уникати помилок;
- підтримка work-life balance – впровадження політики щодо уникнення надмірного навантаження, гнучкого графіку роботи та можливості віддаленої співпраці підвищили мотивацію команди й продуктивність на довгостроковій основі;
- тестування гіпотез – запровадження MVP-підходу для перевірки ідей перед повним впровадженням допомогло зменшити кількість непотрібної роботи та сконцентрувати ресурси на найважливіших напрямках. [9]

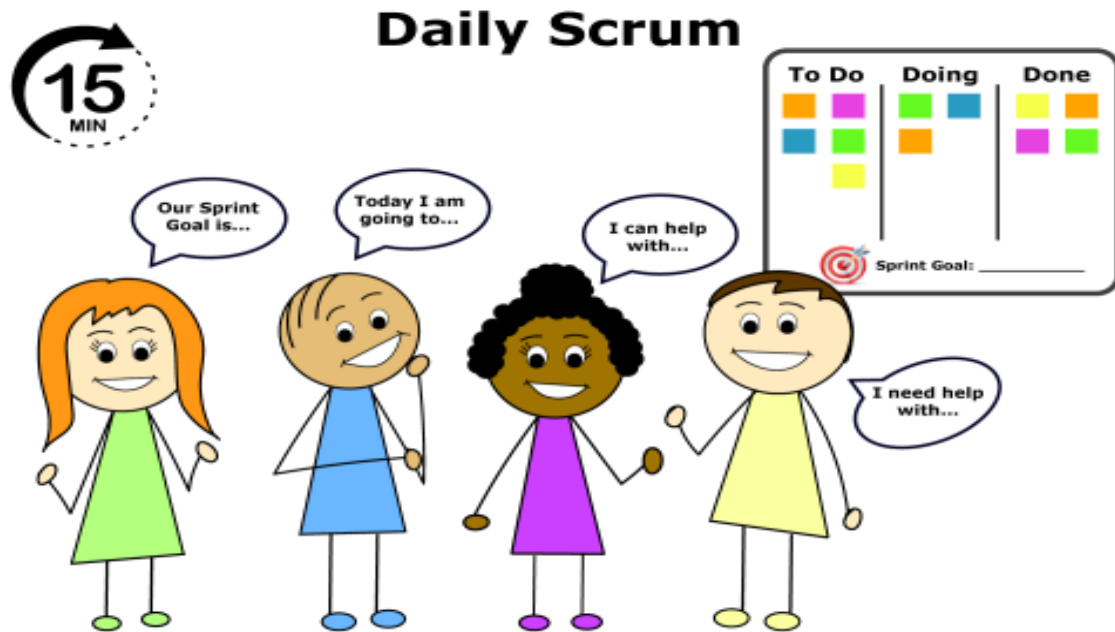


Рисунок 3.3 – Daily Scrum-meeting  
Джерело: [15]

### 3.4 Аналіз ризиків та стратегії їх мінімізації

#### *Внутрішні проблеми проекту*

1. Складність технічних завдань:
  - розробка алгоритмів для аналізу великих обсягів даних з датчиків двигуна;
  - інтеграція з різноманітними системами, що використовуються в авіаційній галузі;
  - забезпечення високої точності та надійності результатів тестування.
2. Недостатність ресурсів:
  - обмеженість фінансування може призвести до затримки проекту або зменшення його масштабів;
  - нестача кваліфікованих фахівців у певних областях (наприклад, в області аналізу даних, розробки вбудованих систем).
3. Ризики, пов'язані з розробкою програмного забезпечення:

- можливі помилки в коді, які можуть призвести до неправильних результатів тестування;
- затримки в розробці окремих модулів системи;
- труднощі з інтеграцією різних компонентів системи.

#### **4. Зміни вимог клієнта:**

- зміни в технічних вимогах або в нормативних документах можуть призвести до необхідності переробки частини вже виконаної роботи.

#### *Зовнішні проблеми проєкту*

##### **1. Нормативно-правова база:**

- зміни в нормативно-правових актах, що регулюють авіаційну галузь, можуть вплинути на вимоги до програмного забезпечення;
- труднощі з отриманням необхідних дозволів та сертифікатів.

##### **2. Конкуренція:**

- наявність інших розробок в цій галузі може створити додаткові труднощі при виведенні продукту на ринок;
- необхідність постійно відстежувати нові технології та тенденції в галузі.

##### **3. Залежність від постачальників обладнання та програмного забезпечення:**

- зміни в політиці постачальників можуть призвести до проблем з доступністю обладнання та програмного забезпечення;
- залежність від сторонніх розробок може обмежити можливості для кастомізації системи.

##### **4. Фактори ризику, пов'язані з експлуатацією літаків:**

- зміни в конструкції двигунів або літаків можуть вимагати відповідних змін у програмному забезпеченні;
- поява нових типів несправностей, які не були передбачені при розробці системи.

### Висновки до розділу 3

Третій розділ детально описує процес розробки програмного забезпечення для автоматичного тестування двигунів Boeing 737 MAX, зокрема, впровадження ітеративного підходу з використанням методологій Agile та Scrum. Цей підхід дозволяє постійно вдосконалювати продукт, швидко реагувати на зміни вимог та забезпечувати високу якість. Команда активно використовувала сучасні інструменти управління проектами, такі як Jira, Confluence та Slack, для ефективної організації роботи та комунікації між членами команди. Основні висновки:

- розроблено основні модулі системи, які покривають більшість функціональних та нефункціональних вимог;
- проведено аналіз роботи команди під час ретроспективи, що дозволило виявити її сильні сторони та визначити напрями для вдосконалення Scrum-процесів;
- ідентифіковано основні ризики проєкту, оцінено їхній вплив та розроблено стратегії зменшення потенційних загроз.

## ВИСНОВКИ

У процесі дослідження було проведено комплексний аналіз можливостей створення програмного забезпечення для автоматичного тестування двигунів літака Boeing 737 MAX із застосуванням гнучких методологій управління проєктами. На основі ретельного вивчення специфіки галузі, потреб і викликів, пов'язаних із тестуванням авіаційних двигунів, були визначені цілі та завдання проєкту, а також сформульовані вимоги до продукту, зважаючи на інтереси користувачів і технічні стандарти.

Основні досягнення та результати роботи включають:

1. *Розробка концепції програмного забезпечення.* Створено концептуальну модель, яка враховує специфіку тестування авіаційних двигунів і відповідає актуальним вимогам галузі. Програма забезпечує автоматизацію ключових процесів тестування, інтеграцію з існуючими системами й інструментами, а також точність і надійність результатів.

2. *Обґрунтування вибору методології Scrum.* Застосування Scrum дозволило забезпечити гнучкість процесу розробки, оперативну адаптацію до змінних технічних вимог і покращену комунікацію в команді. Планування спринтів, регулярні ретроспективи та використання сучасних інструментів менеджменту сприяли зростанню продуктивності й покращенню кінцевого продукту.

3. *Детальне планування проєкту.* Розроблено структуру проєкту з урахуванням обсягів робіт, термінів, бюджету та рольової організації команди. Проведено фінансове планування, яке підтвердило економічну доцільність реалізації програмного забезпечення.

4. *Впровадження сучасних інструментів менеджменту.* Для підвищення ефективності розробки використано такі інструменти, як Jira, Confluence і GitLab CI/CD. Це забезпечило автоматизацію процесів, прозорість управління, злагоджену командну роботу та підвищення якості продукту.

5. *Аналіз та оптимізація робочих процесів.* Проведено оцінку ефективності роботи команди, ідентифіковано потенційні ризики та

розроблено стратегії їх подолання, зокрема в контексті інформаційної безпеки й відповідності нормативним вимогам. Вжиті заходи сприяли зниженню ризиків і підвищенню довіри до розроблюваного програмного забезпечення.

Програмне забезпечення для автоматичного тестування двигунів літака Boeing 737 MAX створює умови для підвищення ефективності тестування, відповідності стандартам безпеки й удосконалення процесів в авіаційній галузі. Наукова новизна дослідження полягає у використанні гнучких методологій управління проектами в контексті розробки програмного забезпечення для автоматичного тестування двигунів літака Boeing 737 MAX, що раніше не отримало достатнього висвітлення у науковій літературі. Це сприяє оптимізації процесів тестування та впровадженню інновацій у галузі авіабудування.

Практична значущість роботи полягає у створенні реального інструменту – програмного забезпечення для автоматизованого тестування, яке може бути інтегроване в авіаційній індустрії, забезпечуючи підвищення точності, ефективності та надійності тестування. Система дозволить зменшити ризики, стандартизувати процеси та створити основу для впровадження нових технологій у виробництві.

*Рекомендації та перспективи впровадження:*

- подальший розвиток функціональності програмного забезпечення, включно з розширенням спектра інтегрованих інструментів і сервісів для автоматизованого тестування, що збільшить його ефективність і гнучкість;
- активне впровадження розробки на ринку через співпрацю з авіаційними виробниками, участь у галузевих виставках та демонстрацію технологічних переваг системи;
- постійний моніторинг змін у вимогах індустрії, збір зворотного зв'язку від користувачів і впровадження покращень на основі аналізу отриманих даних;

- розгляд перспективи масштабування системи для використання на інших моделях літаків і в суміжних галузях, що відкриє нові ринки та сприятиме інтеграції сучасних технологій у виробничі процеси.

Загалом, проведені дослідження та реалізація проєкту демонструють ефективність поєднання сучасних технологій, гнучких методологій управління та глибокого розуміння потреб галузі. Це створює надійну основу для успішного впровадження програмного забезпечення і його подальшого розвитку.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Atlassian: Agile Workflows: Steps and Best Practices [Електронний ресурс] URL: <https://www.atlassian.com/agile/project-management/workflow> (дата звернення: 10.12.2024).
2. Brechner, E. Agile Project Management with Kanban. Microsoft Press. США: 2021, 15-19 с. (Впровадження Kanban).
3. Deemer, P., Benefield, G., Larman, C., & Vodde, B. The Scrum Primer. Scrum Training Institute. США: 2012, 30-32 с. (Планування спринтів у Scrum).
4. Farokhi S. Aircraft Propulsion, 2nd Edition. США: 2014. 1048 с.
5. Graham M. Simons Boeing 737 The World's Most Controversial Commercial Jetliner. 2021. 288 с.
6. Jonathan Falconer MODERN CIVIL AIRLINERS. 2021. 176 с.
7. Kroes M., Rardon J. Aircraft Basic Science, Eighth Edition, 2013. 480 с.
8. Rubin, K. S. . Essential Scrum: A Practical Guide to the Most Popular Agile Process. США: 2021, с. 5-27 (Основні принципи Scrum).
9. Schwaber, K., & Sutherland, J. The Scrum Guide. США: 2012, 18-21 с. (Організація роботи в команді та планування).
10. Апелло Ю. Менеджмент 3.0. Agile-менеджмент. Лідерство та управління командами : навч. посіб, / за ред. Якубовська Г. 2019. 432 с.
11. Електронний ресурс. URL: <https://aviationweek.com/> (дата звернення: 10.12.2024).
12. Електронний ресурс. URL: <https://www.aviationbusinessme.com/boeing/boeing-737max-production-slumps> (дата звернення: 10.12.2024).
13. Електронний ресурс. URL: <https://www.sps-aviation.com/story/?id=2164&h=LEAP-1B-Powering-the-Boeing-737-MAX> (дата звернення: 10.12.2024).
14. Електронний ресурс. URL: <https://www.delasign.com/blog/what-are-the-key-ceremonies-of-scrum/> (дата звернення: 10.12.2024).

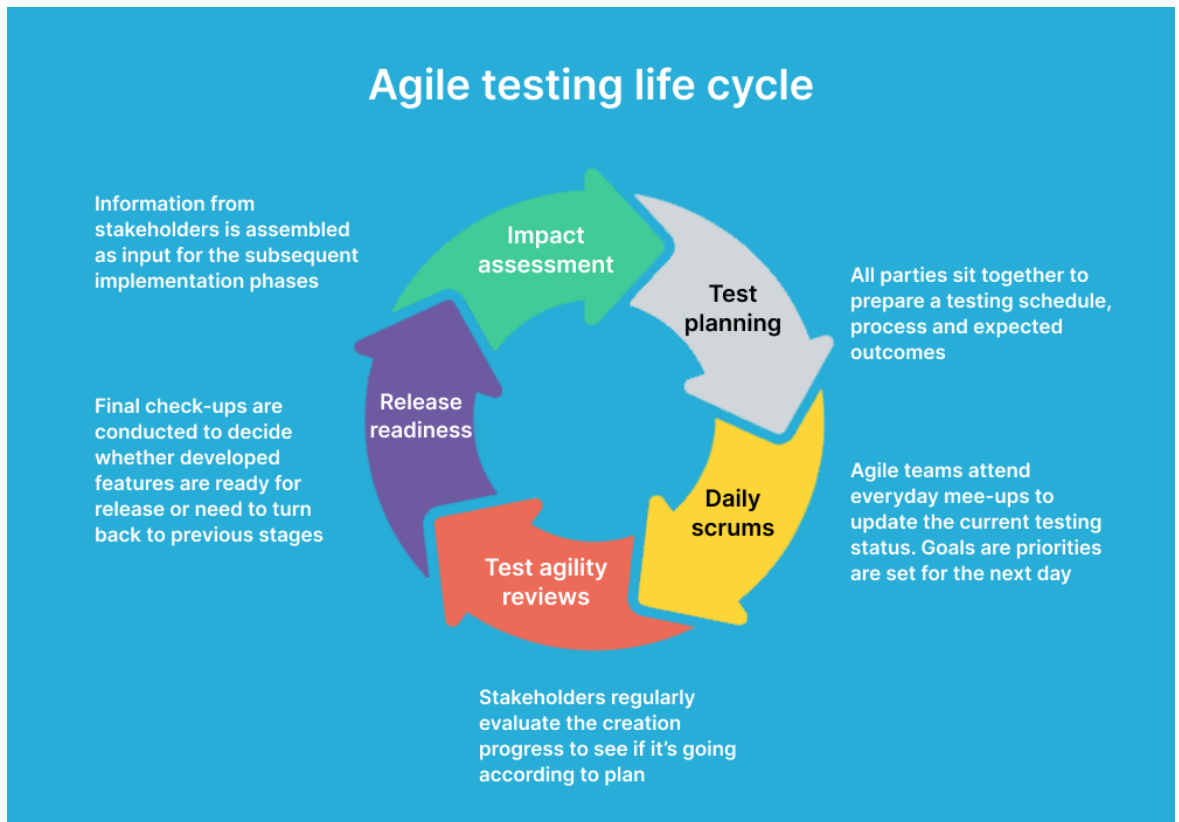
15. Електронний ресурс. URL: <https://www.scrum.org/resources/introduction-daily-scrum> (дата звернення: 10.12.2024).
16. Лофлер М. Ретроспектива в Agile. Перевірені методи та інноваційні підходи : навч. посіб., 2020. 336 с.
17. Майк Кон. Scrum: гнучка розробка ПЗ : навч. посіб. / за ред. Вільямс, 2015. 576 с.
18. Орлова-Курилова О. В., Скопненко Р.І. Роль ІТ в управлінні підприємством та автоматизація підприємств. Сучасний менеджмент організації: витоки, реалії та перспективи розвитку: тези доповідей IV Наукової конференції (18 квітня 2024 р.). - Київ: Університет "КРОК", 2024. [Електронний ресурс] URL: <https://conf.krok.edu.ua/MMO/MMO-2024/paper/view/2152> (дата звернення: 10.12.2024).
19. Орлова-Курилова О. В., Борщов В.О. AGILE-менеджмент у складних умовах: виклики та стратегії при дистанційній роботі під час війни в Україні. Сучасний менеджмент організації: витоки, реалії та перспективи розвитку: тези доповідей IV Наукової конференції (18 квітня 2024 р.). - Київ: Університет "КРОК", 2024. [Електронний ресурс] URL: <https://conf.krok.edu.ua/MMO/MMO-2024/paper/view/2126> (дата звернення: 10.12.2024).
20. Орлова-Курилова О. В., Басанець О.В. Адаптація AGILE-методологій для цифровізації процесів управління проєктами. Сучасний менеджмент організації: витоки, реалії та перспективи розвитку: тези доповідей IV Наукової конференції (18 квітня 2024 р.). - Київ: Університет "КРОК", 2024. [Електронний ресурс] URL: <https://conf.krok.edu.ua/MMO/MMO-2024/paper/view/2128> (дата звернення: 10.12.2024).
21. Орлова-Курилова О. В., Черниш М.В. AGILE MANIFESTO як статут інноваційної методології розробки програмного забезпечення. Сучасний менеджмент організації: витоки, реалії та перспективи розвитку:

тези доповідей IV Наукової конференції (18 квітня 2024 р.). - Київ: Університет "КРОК", 2024. [Електронний ресурс] URL: <https://conf.krok.edu.ua/MMO/MMO-2024/paper/view/2056> (дата звернення: 10.12.2024).

22. Пінє І., Остервальдер А. Побудова бізнес-моделей. Настільна книга стратега та новатора : навч. посіб. / за ред. Паблішер А. 2018. 288 с.

## ДОДАТКИ

## ДОДАТОК А



*Рисунок А.1 - Цикл Agile тестування*

*Джерело: Принципи тестування Agile, електронний ресурс URL:*

*[<https://www.globalapptesting.com/the-ultimate-guide-to-agile-testing>]*

## ДОДАТОК Б



*Рисунок Б.1 - Порівняння циклу Scrum та Kanban*

*Джерело: Brightwork: Comparing Agile Project Management*

*Methodologies: Scrum vs Kanban URL: [<https://www.brightwork.com/wp-content/uploads/Scrum-vs-Kanban-1-scaled.webp>]*