

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

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Гнучке управління створенням системи керування
фермерським господарством SmartAgri»

Ступінь вищої освіти – магістр

Спеціальність – 073 «Менеджмент»

Освітня програма «Agile-технології розробки програмного
забезпечення»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Керівник:

завідувач кафедру, к.е.н., доцент,
Денис БАЛДИК

Керівник:

перший проректор, к.е.н, професор,
Наталія ЛІТВІН

Виконав : здобувач

групи МЕН/Agile-23м

Віктор БОГАЧОВ

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

ЗАТВЕРДЖУЮ:

завідувач кафедри інформаційного
менеджменту, математики та
статистики

_____ Денис БАЛДИК
«__» _____ 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
БОГАЧОВА ВІКТОРА ЄВГЕНІЙОВИЧА**

Тема роботи	Гнучке управління створенням системи керування фермерським господарством SmartAgri
Номер та дата наказу про затвердження теми	№ 56-5 від 27.06.2024 р.
Коротка постановка завдання	Обґрунтування доцільності створення системи керування фермерським господарством «SmartAgri». Опис особливостей гнучкого управління розробкою MVP системи «SmartAgri» з використанням фреймворка Scrum. Розкриття особливостей та підходів до управління Agile командою під час розробки MVP системи «SmartAgri».
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	1. Алькема В. Г., Кириченко О. С. Менеджмент організацій: навчальний посібник. Кн.1. Київ: Університет «КРОК», 2023. 276 с. Електронний ресурс 2. Горковець, Д. О. Балдик. Управління ресурсами в IT проектах. Держава, регіони, підприємництво: інформаційні, суспільно-правові, соціально-економічні аспекти розвитку: тези доповідей V Міжнародної конференції (Київ, 7 грудня 2023 р.). Київ: Університет "КРОК", 2023. 3. Данченко О.Б. Практичні аспекти реінжинірингу бізнес-процесів / О.Б. Данченко. Київ. Університет економіки та права «КРОК», 2017. – 238 с.
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має містити теоретичне, системотехнічне або експериментальне 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

Денис БАЛДИК

Керівник

Наталія ЛІТВІН

Здобувач

Віктор БОГАЧОВ

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

Денис БАЛДИК

Керівник

Наталія ЛІТВІН

Здобувач

Віктор БОГАЧОВ

АНОТАЦІЯ

**Богачов В.Є ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ
СИСТЕМИ КЕРУВАННЯ ФЕРМЕРСЬКИМ ГОСПОДАРСТВОМ
«SmartAgri»**

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, 2024р.

В кваліфікаційній роботі розглянуто теоретико-методичні основи гнучкого управління, зокрема особливості Agile підходів в управлінні створенням продукту. Представлено дизайн бізнесу та опис запропонованого продукту SmartAgri який має на меті створення конкурентної системи для зайняття долі на ринку. Розглянуті окремі аспекти гнучкого управління створенням продукту SmartAgri, зокрема: цілі проєкту, проблеми які можуть виникнути при реалізації проєкту. Детально розглянуто план виконання та терміни проєкту з розробки MVP системи «SmartAgri». Розраховано бюджет проєкту, описана команда проєкту, стейкхолдери проєкту, етичний кодекс (цінності), принципи (Agile-маніфест), правила роботи, основні етапи тестування та принципи вирішення проблем та покращення роботи команди, план комунікацій в проєкті та підходи до лідерства в команді.

Ключові слова: Agile, гнучкий менеджмент, проєкт, продукт, система, Agile-команда, Scrum.

Табл. 9. Рис. 13. Бібліограф.: 35 найм.

ANNOTATION

Bohachov V.Y. “FLEXIBLE MANAGEMENT OF THE DEVELOPMENT OF THE “SmartAgri” FARM MANAGEMENT SYSTEM

Project explanatory note by specialty 073 - Management (educational program - Agile software development technologies). – «KROK» University, Educational and Scientific Institute of information and communication technologies, Department of Information Management, Mathematics and Statistics, Kyiv, 2024.

The qualification work considers the theoretical and methodological foundations of agile management, in particular the features of Agile approaches in product development management. The business design and description of the proposed “SmartAgri” product is presented, which aims to create a competitive system for taking market share. Some aspects of flexible management of “SmartAgri” product development are considered, in particular: project goals, problems that may arise during project implementation. The implementation plan and timing of the project for the development of the MVP of the “SmartAgri” system are considered in detail. The project budget is calculated, the project team, project stakeholders, code of ethics (values), principles (Agile manifesto), work rules, main stages of testing and principles of problem solving and team improvement, project communication plan and approaches to team leadership are described.

Keywords: Agile, agile management, project, product, system, Agile team, Scrum.

Tabl. 9. Fig. 13. Bibliography: 35 Items

ЗМІСТ

ЗМІСТ	6
ВСТУП.....	7
РОЗДІЛ 1. ДИЗАЙН БІЗНЕСУ SMARTAGRI.....	10
1.1 Опис системи керування фермерським господарством SmartAgri.....	10
1.2 Оцінка конкурентів та стейкхолдерів	13
1.3 Постановка цілей та завдань проєкту.....	17
1.4 Бізнес вимоги до системи SmartAgri	23
Висновки до Розділу 1	28
РОЗДІЛ 2. ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ СИСТЕМИ КЕРУВАННЯ ФЕРМЕРСЬКИМ ГОСПОДАРСТВОМ SMARTAGRI.....	30
2.1 Обґрунтування вибору фреймворку гнучкого управління	30
2.2 Планування змісту і тривалості проєкту з розробки MVP.....	34
2.3 Планування команди проєкту.....	40
2.4 Планування вартості проєкту з розробки MVP	45
2.5 Планування процесів проєкту.....	47
2.6 Моніторинг виконання проєкту.....	52
Висновки по розділу 2.....	54
РОЗДІЛ 3. РЕЗУЛЬТАТИ РОЗРОБКИ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КОМАНДИ З РОЗРОБКИ SMARTAGRI.....	56
3.1 Огляд інкременту системи SmartAgri.....	56
3.2 Ретроспектива роботи команди	58
3.3 Сучасний інструментарій менеджера в agile-середовищі	64
Висновки по розділу 3.....	72
Висновок.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	75

ВСТУП

Актуальність теми. Сільськогосподарська галузь стикається з численними викликами, такими як зміна клімату, зменшення родючості ґрунтів, висока вартість ресурсів та потреба в підвищенні врожайності для забезпечення зростаючого населення планети. Традиційні методи управління господарством часто не забезпечують достатньої ефективності та точності для досягнення цих цілей. У цьому контексті впровадження сучасних цифрових технологій є необхідністю для підвищення продуктивності та стійкості сільського господарства.

Проєкт "SmartAgri" спрямований на вирішення цих викликів шляхом розробки інтегрованої системи управління сільськогосподарським господарством, яка забезпечить фермерам доступ до інструментів, що дозволяють.

- Підвищити ефективність виробництва;
- Покращити врожайність і якість продукції;
- Забезпечити стале управління ресурсами;
- Зменшити ризики та покращити стійкість до зовнішніх факторів;
- Підтримати інноваційний розвиток агросектору.

Таким чином, проєкт "SmartAgri" є важливим кроком у напрямку модернізації сільськогосподарської галузі, забезпечуючи фермерам ефективні інструменти для підвищення продуктивності, зниження витрат та підтримки екологічної стійкості їхньої діяльності. Цей проєкт сприятиме довгостроковому розвитку аграрного сектора, задовольняючи потреби як сьогодення, так і майбутніх поколінь.

Мета дослідження. Виявити особливості та реалізувати функції гнучкого управління створенням MVP системи SmartAgri, для покращення практичних навичок та розвитку вмінь притаманних ролі Scrum-майста в команді.

Завдання дослідження. Для досягнення мети необхідно виконати наступні завдання.

- Обґрунтувати видіння створюваного продукту;
- Детально описати особливості використання підходів гнучкого менеджменту створення MVP системи SmartAgri з використанням фреймворку Scrum;
- Розкрити необхідні навички та інструменти Scrum-майста необхідні для реалізації MVP системи SmartAgri.

Об'єкт дослідження. Процеси та методи реалізації функцій гнучкого менеджменту під час створення програмних засобів.

Предмет дослідження. Процеси управління розробкою програмного забезпечення з використанням методології Agile (на прикладі MVP системи SmartAgri).

Методи дослідження. Для реалізації процесів гнучкого управління створенням MVP системи SmartAgri застосовано підходи описані у фреймворку Scrum та інструментарій системи для планування та аналізу виконання роботи Jira.

Практичне значення результатів дослідження. Створення MVP для системи SmartAgri є важливим кроком, який дозволяє швидко вивести на ринок основні функції, що підлягають перевірці та вдосконаленню. Це особливо критично в аграрному секторі, де затримки можуть призвести до значних збитків. MVP забезпечує ранній зворотний зв'язок від користувачів, що допомагає адаптувати продукт до реальних потреб і зменшити ризики інвестицій у нерентабельні рішення.

Фокусування на ключових функціях MVP оптимізує ресурси, дозволяючи уникнути перевантаження продукту і зосередитися на найбільш цінних аспектах. Це також знижує витрати на ранніх етапах розробки, що важливо для стартапів з обмеженим бюджетом. Дані про використання системи в реальному часі сприяють вдосконаленню продукту та підвищенню його конкурентоспроможності.

Крім того, MVP закладає основу для майбутнього масштабування системи. Після тестування основних функцій команда може додавати нові

можливості відповідно до запитів користувачів, що стимулює зростання бізнесу в аграрному секторі. Таким чином, MVP не лише знижує ризики та витрати, але й створює базу для постійного розвитку продукту, що відповідає потребам аграріїв.

Структура та обсяг роботи. Робота складається зі вступу, трьох основних розділів, висновку та списку використаних джерел. Загальний обсяг роботи 79 сторінка, обсяг основного тексту 66 сторінок.

РОЗДІЛ 1. ДИЗАЙН БІЗНЕСУ SMARTAGRI

1.1 Опис системи керування фермерським господарством SmartAgri

SmartAgri - це система керування фермерським господарством.

Система SmartAgri призначена для вирішення низки задач, пов'язаних із управлінням фермерським господарством, забезпечуючи оптимізацію сільськогосподарських процесів та підвищення ефективності виробництва. Задачі які має вирішити система SmartAgri зображені на Рисунок 1.1.



Рисунок 1.1 - Задачі які вирішує система

Джерело: розроблено автором

Моніторинг стану посівів та ґрунтів – за допомогою датчиків та дронів система збирає інформацію про стан полів, вологість ґрунту, рівень добрив, погодні умови та наявність шкідників у режимі реального часу.

Планування та управління ресурсами – автоматизований розрахунок оптимальної кількості насіння, води, добрив та пестицидів для кожного поля на основі зібраних даних, що знижує витрати та покращує врожайність.

Аналіз та прогнозування врожайності – система використовує алгоритми аналізу великих даних (Big Data) для прогнозування врожайності на основі історичних даних і поточних умов на полях.

Оптимізація робочих процесів – планування та управління сільськогосподарськими операціями, такими як посів, полив, обробка культур і збір урожаю, для досягнення максимального результату з мінімальними затратами часу та ресурсів.

Управління технікою та обладнанням – інтеграція з сільськогосподарською технікою дозволяє автоматизувати виконання завдань, таких як посів чи внесення добрив, а також слідкувати за технічним станом обладнання.

Екологічний моніторинг – система допомагає контролювати та зменшувати негативний вплив господарської діяльності на навколишнє середовище, сприяючи стійкому розвитку фермерського господарства.

Економічний аналіз та управління фінансами – система надає інструменти для обліку витрат та планування бюджету, що допомагає фермерам приймати фінансово обґрунтовані рішення.

SmartAgri інтегрує всі ці процеси в єдину систему управління, дозволяючи фермерам більш ефективно контролювати всі аспекти свого господарства.

Метою проєкту "SmartAgri" є розробка та впровадження інноваційної системи керування сільськогосподарським господарством, яка забезпечить фермерам та агрономам інструменти для підвищення ефективності та рентабельності їхньої діяльності. Проєкт "SmartAgri" спрямований на створення комплексного рішення, яке поєднує технологічні нововведення з практичними потребами сільськогосподарських підприємств, забезпечуючи їх стале зростання та конкурентоспроможність.

Сільськогосподарська галузь стикається з численними викликами, такими як зміна клімату, зменшення родючості ґрунтів, висока вартість ресурсів та потреба в підвищенні врожайності для забезпечення зростаючого населення планети. Традиційні методи управління господарством часто не забезпечують достатньої ефективності та точності для досягнення цих цілей. У цьому контексті впровадження сучасних цифрових технологій є необхідністю для підвищення продуктивності та стійкості сільського господарства.

Проект "SmartAgri" спрямований на вирішення викликів з якими стикається агросектор шляхом розробки інтегрованої системи управління сільськогосподарським господарством, яка забезпечить фермерам доступ до інструментів, що дозволяють вирішити ряд проблем зображених на Рисунок 1.2.



Рисунок 1.2 – Проблеми які дозволить вирішити система "SmartAgri"

Джерело: розроблено автором

Таким чином, проєкт "SmartAgri" є важливим кроком у напрямку модернізації сільськогосподарської галузі, забезпечуючи фермерам ефективні

інструменти для підвищення продуктивності, зниження витрат та підтримки екологічної стійкості їхньої діяльності. Цей проєкт сприятиме довгостроковому розвитку аграрного сектора, задовольняючи потреби як сьогодення, так і майбутніх поколінь.

Створення системи SmartAgri вирішить одну з ключових проблем сучасного сільського господарства – неефективне управління ресурсами та процесами, що виникає через відсутність єдиної платформи для моніторингу, аналізу та автоматизації виробничих операцій. У традиційному фермерському господарстві використання таких ресурсів, як вода, добрива, електроенергія та робоча сила, часто базується на досвіді або інтуїції, що призводить до втрат та зниження продуктивності. Впровадження системи автоматизації та управління SmartAgri дозволить оптимізувати використання ресурсів за допомогою інноваційних рішень, таких як Інтернет речей (IoT), штучний інтелект та аналітика великих даних. Фермери отримають можливість у режимі реального часу моніторити стан ґрунту, кліматичні умови, здоров'я рослин і тварин, а також контролювати процеси поливу, внесення добрив та збору врожаю. Це сприятиме підвищенню врожайності, зниженню витрат і зменшенню негативного впливу на довкілля. SmartAgri надасть фермерам інструменти для оперативного реагування на зміни умов, планування виробництва та прогнозування ризиків, що підвищить загальну ефективність сільського господарства і дозволить адаптуватися до сучасних викликів аграрної галузі.

1.2 Оцінка конкурентів та стейкхолдерів

Основними конкурентами SmartAgri є John Deere, Trimble, DJI Agriculture, AG Leader Technology та Raven Industries. Всі п'ять компаній мають значний досвід роботи, позитивний імідж та стабільні тенденції розвитку.

Імідж. John Deere лідирує з оцінкою 9, тоді як DJI Agriculture і Raven Industries мають оцінку 7, а Trimble та AG Leader Technology мають оцінки 6

та 5 відповідно. Це свідчить про лідерство John Deere у сегменті програмного забезпечення для сільського господарства.

Досвід роботи. John Deere також має найбільший досвід, що підкреслює його високу репутацію серед користувачів. Trimble і Raven Industries трохи відстають з оцінкою 7, що все ж вказує на їхню високу компетенцію, DJI Agriculture і AG Leader Technology мають найменший досвід з оцінкою 6.

Тенденції розвитку. John Deere демонструє кращі тенденції розвитку з оцінкою 9, що підтверджує його лідерство в галузі. DJI Agriculture має оцінку 8, що свідчить про швидкий розвиток та впровадження інновацій. Trimble та Raven Industries мають оцінку 6 та демонструють стабільний розвиток, а Raven Industries фокусується на стабільності та вдосконаленні своїх рішень з оцінкою 5.

Потенціал розвитку. John Deere має великий потенціал для подальшого розвитку завдяки своєму сильному іміджу, інвестиціям в інновації та інтеграцію нових технологій. DJI Agriculture має значний потенціал завдяки швидкому впровадженню нових технологій. Trimble та Raven Industries трохи поступаються, а AG Leader Technology має обмежений потенціал розвитку через консервативний підхід.

Як глобальний лідер, John Deere займає значну частку на ринку смарт-агрокультур завдяки своїм технологіям точного землеробства. Компанія домінує в Північній Америці, яка займає приблизно 36% ринку розумного сільського господарства [6].

Trimble це важливий гравець у галузі програмного забезпечення для управління фермерськими господарствами та точного землеробства. Trimble має сильну позицію, особливо в апаратних рішеннях, таких як GPS та IoT-пристрої. Вона змагається з іншими провідними компаніями в Північній Америці та на глобальному рівні, стимулюючи технологічні інновації у смарт-фермерстві [6].

DJI Agriculture лідер у технологіях дронів, DJI Agriculture швидко збільшує свою частку на ринку, особливо в Азії, де точне землеробство

стрімко розвивається. DJI відіграє ключову роль в інтеграції дронів з системами управління фермами, що дозволяє проводити передовий аналіз даних [7].

AG Leader Technology провідний гравець на ринку програмного забезпечення для управління фермерськими господарствами, AG Leader Technology сприяє моніторингу та аналітиці в реальному часі. Компанія активно надає рішення для управління як рослинництвом, так і тваринництвом [7].

Raven Industries є важливим гравцем у точному землеробстві, пропонуючи рішення, такі як системи наведення та технології змінної норми внесення. Компанія є частиною конкурентного ландшафту, роблячи свій внесок у загальний розвиток точного землеробства [7].

Порівняння долі конкурентів на ринку зображена на Рисунок 1.3.

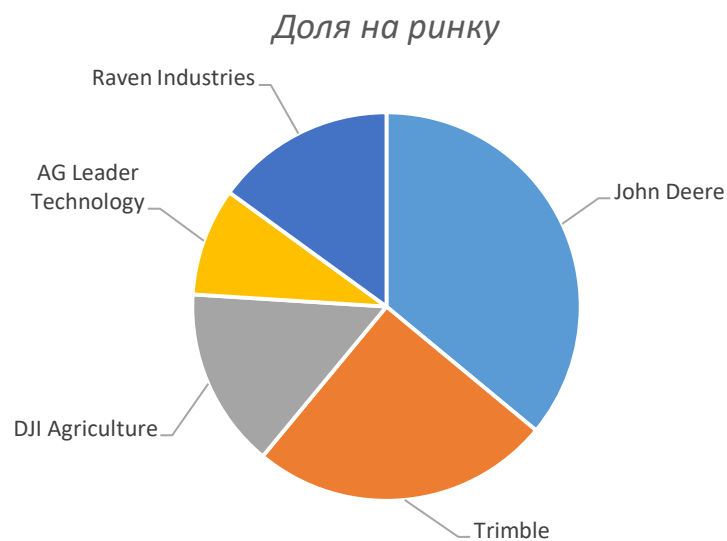


Рисунок 1.3 – Доля конкурентів на ринку

Джерело: розроблено автором

Високий рівень конкуренції у цій галузі означає, що для SmartAgri важливо відрізнитися інноваційними рішеннями та високою якістю послуг для ефективною конкуренції. Враховуючи досвід, імідж та потенціал розвитку

конкуренції, SmartAgri повинна зосередитися на унікальних пропозиціях та бездоганному обслуговуванні клієнтів, щоб зайняти свою нішу на ринку сільськогосподарського програмного забезпечення. Короткі висновки порівняння конкурентів зібрано в Таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз конкурентів

Критерії оцінки	Оцінка конкурентів				
	Конкурент 1 (John Deere)	Конкурент 2 (Trimble)	Конкурент 3 (DJI Agriculture)	Конкурент 4 (AG Leader Technology)	Конкурент 5 (Raven Industries)
Досвід роботи	9	7	6	6	7
Імідж	9	6	7	5	7
Тенденції розвитку	9	6	8	5	6
Потенціал розвитку	9	7	8	6	7
Доля на ринку	9	7	6	5	6

Висновки з аналізу конкурентів. "SmartAgri" повинна зосередитися на інноваціях і гнучкості, щоб конкурувати з лідерами, такими як John Deere, який має сильну ринкову позицію та великий потенціал розвитку. Зокрема, Trimble та Raven Industries мають помітні частки ринку, але потребують вдосконалення підтримки клієнтів і інновацій. DJI Agriculture швидко розвивається, але має високі ціни, а AG Leader Technology має обмежений потенціал розвитку. Використовуючи ці можливості, "SmartAgri" може зайняти конкурентну позицію на ринку, зокрема шляхом пропозиції інтегрованих рішень за конкурентоспроможною ціною.

Основними стейкхолдерами системи SmartAgri є фермери та агровиробники, які використовують систему для автоматизації та оптимізації сільськогосподарських процесів, інвестори та фінансові партнери, які забезпечують фінансування та очікують зростання ефективності агробізнесу,

державні органи, відповідальні за контроль екологічних стандартів та регуляторних норм, і місцеві громади, які отримують вигоду від сталого розвитку сільськогосподарських земель і покращення соціальних та екологічних умов у своїх регіонах. Оцінка стейкхолдерів показана в Таблиці 1.2.

Таблиця 1.2 – Ситуаційна оцінка стейкхолдерів

Критерії оцінки	Оцінка стейкхолдерів			
	Стейкхолдер 1 (Фермери та агровиробники)	Стейкхолдер 2 (Інвестори та фінансові партнери)	Стейкхолдер 3 (Державні органи)	Стейкхолдер 4 (Місцеві громади)
Досвід роботи	8	7	7	6
Імідж	7	8	6	7
Тенденції розвитку	7	8	6	7
Потенціал розвитку	7	8	6	7

Висновки з оцінки стейкхолдерів. Основними стейкхолдерами для SmartAgri є Фермери та агровиробники, Інвестори та фінансові партнери, Державні органи та Місцеві громади. Всі вони мають значний досвід роботи, позитивний імідж та потенціал для розвитку. Співпраця з цими організаціями може сприяти покращенню якості наданих послуг та залученню нових користувачів.

1.3 Постановка цілей та завдань проєкту

Постановка цілей у гнучкому менеджменті (Agile) базується на адаптивному, ітеративному підході, який дозволяє швидко реагувати на зміни та постійно вдосконалювати процес розробки. Основні принципи гнучкого підходу до постановки цілей включають.

Ітеративність та інкрементальність – цілі формуються та реалізуються короткими циклами (ітераціями), що дозволяє поступово створювати функціональні частини системи SmartAgri. Це дає можливість оцінювати результати кожного етапу, вносити корективи та вдосконалювати продукт на кожній стадії розробки.

Гнучкість і адаптація – цілі не є статичними, вони можуть змінюватися залежно від нових вимог користувачів або умов ринку. Це дозволяє швидко підлаштовуватися під зміни, що особливо важливо в аграрному секторі, де умови можуть змінюватися через погодні або економічні фактори.

Пріоритетність на основі цінності – у гнучкому менеджменті цілі пріоритезуються залежно від їхньої цінності для кінцевого користувача. Найбільш важливі та критичні для бізнесу задачі виконуються першочергово, наприклад, автоматизація моніторингу посівів або аналіз даних для оптимізації врожайності.

Тісна співпраця з користувачем – постановка цілей здійснюється з урахуванням постійного зворотного зв'язку від кінцевих користувачів. Це дозволяє коригувати цілі та вимоги, щоб продукт максимально відповідав потребам фермерів.

Самоорганізовані команди – у гнучкому підході команди самостійно визначають цілі на кожен ітерацію, базуючись на глобальних цілях проєкту. Це підвищує мотивацію та відповідальність кожного члена команди за результат, водночас дозволяючи швидше досягати запланованих цілей.

Фокус на швидкому результаті – цілі формуються таким чином, щоб на кожній ітерації отримувати функціональну частину системи SmartAgri, яку можна перевірити та продемонструвати користувачам. Це забезпечує раннє виявлення проблем і можливість швидкого їх вирішення.

Цілі створення системи базуються на принципах гнучкого менеджменту описаних вище та представлені нижче.

1. Орієнтація на потреби користувача: забезпечити фермерів та аграріїв інтуїтивно зрозумілим інтерфейсом, що дозволить легко управляти

господарством, збирати й аналізувати дані з різних сенсорів, і приймати обґрунтовані рішення., тобто створення зручного інтерфейсу для відстеження та управління процесами в господарстві.

2. Мінімальний життєздатний продукт (MVP). Запустити базову версію системи з ключовими функціями у найкоротші терміни, щоб швидко отримати зворотний зв'язок від користувачів і адаптувати подальші етапи розробки, плановий запуск MVP через 3-6 місяців після початку розробки.

3. Ітераційна розробка. Розбивати створення системи на короткі ітерації, кожна з яких додає новий функціонал або вдосконалює існуючий, дозволяючи гнучко реагувати на змінені вимоги або нові виклики. Мета: завершувати кожну ітерацію з додаванням конкретної функціональності, наприклад, інтеграція нових сенсорів або аналітичних інструментів.

4. Адаптація на основі зворотного зв'язку. Регулярно отримувати зворотний зв'язок від фермерів, агрономів і стейкхолдерів, щоб вносити корективи в систему. Це дозволить врахувати практичні потреби користувачів і зробити продукт більш ефективним. Мета: регулярний збір зворотного зв'язку після кожного релізу і внесення корективів у наступні ітерації.

5. Оптимізація управління ресурсами. Забезпечити ефективне управління ресурсами (водою, добривами, енергією) через автоматизацію та аналітику, що підвищить продуктивність господарства та знизить витрати. Мета: створити модулі автоматизованого управління ресурсами для зниження витрат на 10-20% за перший рік використання.

6. Швидке впровадження нових технологій. Сприяти інтеграції нових технологій у систему (дрони, IoT-сенсори, аналітика на базі ШІ) для постійного вдосконалення та розширення функціоналу. Мета: щоквартальна інтеграція нових технологій для підвищення ефективності системи.

7. Гнучкість та масштабованість. Створити систему, яка легко масштабуватиметься для управління як малими, так і великими господарствами, адаптуючись під різні типи бізнесів. Мета: Забезпечити

можливість адаптації системи для господарств різного масштабу без значних змін в архітектурі.

8. Безперервна підтримка та вдосконалення. Постійно оновлювати систему на основі нових технологій та користувацького зворотного зв'язку для забезпечення довгострокової актуальності і стабільної роботи. Мета: щоквартальне оновлення системи з додаванням нових функцій та виправленням помилок.

Цілі створення системи SmartAgri, засновані на принципах гнучкого менеджменту, спрямовані на забезпечення ефективного управління сільськогосподарськими процесами шляхом впровадження сучасних технологій, автоматизації та аналітики. Система повинна швидко адаптуватися до потреб користувачів, забезпечуючи гнучкість, масштабованість та зручність у використанні, що дозволить фермерам оптимізувати ресурси, підвищити продуктивність і знизити витрати. Постійний зворотний зв'язок та ітераційний підхід до розробки забезпечать оперативне внесення змін, а також інтеграцію нових технологій, що підтримає ефективність і конкурентоспроможність системи в довгостроковій перспективі.

Гнучкий менеджмент (Agile) пропонує особливий підхід до постановки завдань, орієнтований на адаптивність, швидку реакцію на зміни та постійний зворотний зв'язок. Цей підхід ефективно застосовується для створення системи SmartAgri, щоб забезпечити оптимальне управління сільськогосподарським процесом. Основні принципи постановки завдань.

Декомпозиція завдань – кожне завдання розбивається на менші, більш конкретні підзадачі, які легко виконати протягом короткого часу (ітерації або спринту). Це дозволяє швидко бачити прогрес і вносити корективи.

Пріоритизація завдань – завдання пріоритезуються за принципом їхньої цінності для кінцевого користувача. Найбільш критичні та важливі завдання виконуються першочергово. Наприклад, автоматизація поливу або моніторинг стану ґрунту можуть мати пріоритет перед іншими функціями.

Чітка формулювання вимог – завдання формулюються так, щоб бути зрозумілими для всіх членів команди. Часто використовуються user stories (історії користувачів), які описують функціональність з точки зору користувача, наприклад: "Як фермер, я хочу отримувати повідомлення про рівень вологості ґрунту, щоб своєчасно здійснювати полив".

Ітераційність – завдання виконуються у межах коротких ітерацій, що дозволяє регулярно перевіряти їх виконання та коригувати план. Це дає можливість постійно адаптуватися до нових вимог або змін ринку.

Зворотний зв'язок – після виконання кожного завдання здійснюється оцінка результатів та отримується зворотний зв'язок від користувачів або замовників. Це дозволяє своєчасно вносити зміни в наступні завдання або покращувати вже створені функції.

Прозорість і візуалізація прогресу – завдання відображаються на спільних візуальних дошках (Kanban або Scrum-дошка), щоб кожен член команди міг бачити поточний стан справ. Це сприяє прозорості в процесі розробки та покращує взаємодію в команді.

Орієнтація на результат – завдання формулюються таким чином, щоб мати конкретні, вимірювані результати. Наприклад, завдання може звучати як "інтегрувати модуль для збору даних із сенсорів", а результатом буде функціональний модуль, що передає дані у систему.

Самоорганізація команди – члени команди самостійно визначають, які завдання брати у роботу, виходячи з власної спеціалізації та експертизи, що підвищує відповідальність за результат і стимулює інновації.

Розробка програмного забезпечення для управління господарством.

1. Створити інтуїтивно зрозумілий інтерфейс для управління даними, пов'язаними з сільськогосподарськими процесами, такими як планування посівів, моніторинг стану полів, управління запасами, фінансовий облік та аналіз врожайності.

2. Інтегрувати алгоритми аналізу даних та машинного навчання для надання рекомендацій щодо оптимізації агротехнічних практик.

3. Розробити модулі для підключення та управління різноманітними датчиками (вологоміри, температурні датчики, ґрунтові сенсори тощо), що забезпечать точний збір даних про стан полів.
4. Забезпечити сумісність з сучасними сільськогосподарськими дронами та обладнанням для аерофотозйомки та внесення добрив.
5. Впровадити систему точного землеробства.
6. Створити функціонал для аналізу супутникових зображень та аерофотознімків, що дозволить виявляти проблемні зони на полях та приймати рішення щодо диференційованого внесення добрив і пестицидів.
7. Реалізувати можливість побудови карт урожайності та інших агрономічних показників для точного планування польових робіт.
8. Створити мобільний додаток для зручного доступу до інформації та управління господарством з будь-якого місця, забезпечуючи можливість роботи офлайн.
9. Впровадити функції миттєвих сповіщень про важливі події, такі як зміни в погоді, необхідність поливу чи внесення добрив.
10. Тестувати та вдосконалювати систему.
11. Провести повномасштабне тестування системи на різних типах господарств, від малих сімейних ферм до великих агрокомпаній, для виявлення можливих проблем та їх усунення.
12. Зібрати відгуки від користувачів для постійного вдосконалення функціоналу та адаптації системи до реальних потреб фермерів.
13. Розробити навчальні матеріали, такі як відеоуроки, посібники та вебінари, для навчання фермерів і агрономів ефективному використанню системи "SmartAgri".
14. Забезпечити технічну підтримку користувачів через гарячу лінію, електронну пошту та онлайн-платформи.
15. Розробити маркетингову стратегію для просування продукту на ринок сільськогосподарських технологій, визначивши цільові аудиторії та канали комунікації.

16. Здійснити вихід на ринок, включаючи участь у галузевих виставках, презентаціях, партнерських програмах і рекламних кампаніях.

Виконання цих завдань дозволить реалізувати проєкт "SmartAgri", сприяючи впровадженню передових технологій у сільське господарство, підвищенню продуктивності та рентабельності фермерських господарств, а також забезпеченню сталого управління ресурсами.

1.4 Бізнес вимоги до системи SmartAgri

Бізнес-вимоги — це ключові цілі та умови, які система або продукт повинні виконувати для досягнення успіху проєкту і задоволення потреб бізнесу. Вони визначають, що саме потрібно зробити для досягнення комерційних і організаційних цілей компанії, і зазвичай описують загальні потреби або проблеми, які система повинна вирішувати.

Основні характеристики бізнес-вимог.

Цілі проєкту — це конкретні результати, які система або продукт повинні досягти для вирішення бізнес-завдань, підвищення ефективності бізнес-процесів або створення цінності для організації. Вони відображають стратегічне бачення компанії та те, як проєкт сприятиме його реалізації. Цілі формулюються на основі бізнес-вимог і є основою для прийняття рішень під час проєктування та розробки системи.

Користь для бізнесу — це конкретна цінність або переваги, які організація отримає від впровадження системи чи продукту. Вона відображає, як реалізація бізнес-вимог допоможе досягти ключових цілей компанії та підвищить її ефективність, конкурентоспроможність або прибутковість.

Стратегічне значення відображає, як реалізація проєкту або системи сприяє довгостроковій стратегії розвитку компанії та досягненню її ключових бізнес-цілей. Це означає, що система повинна підтримувати основні напрямки діяльності бізнесу, підсилювати його позиції на ринку та забезпечувати стійке зростання і конкурентоспроможність.

Бізнес-вимоги до системи визначають ключові аспекти, які повинні бути реалізовані для досягнення стратегічних цілей компанії та створення реальної цінності. Вони включають чітко сформульовані цілі, які демонструють, як система сприятиме підвищенню ефективності, зниженню витрат, зростанню доходів і покращенню клієнтського досвіду. Важливо, щоб бізнес-вимоги підтримували стратегічне значення проєкту, сприяли реалізації місії компанії та забезпечували її конкурентоспроможність і стійке зростання на ринку. Таким чином, правильно сформульовані бізнес-вимоги є основою для успішного впровадження системи, яка відповідає довгостроковим цілям та потребам організації.

Бізнес-вимоги до системи SmartAgri визначають ключові цілі та функціональні можливості, необхідні для досягнення успіху проєкту та забезпечення його цінності для фермерів і агробізнесу. Основні вимоги зображені на Рисунок 1.4.



Рисунок 1.4 - Бізнес-вимоги до системи SmartAgri

Джерело: розроблено автором

Оптимізація агропроцесів у системі SmartAgri передбачає автоматизацію та удосконалення ключових сільськогосподарських операцій, таких як полив, внесення добрив і моніторинг стану рослин. Завдяки інтеграції сучасних технологій, таких як сенсори, дрони та системи управління, SmartAgri дозволяє точно налаштувати процеси відповідно до реальних потреб рослин та умов навколишнього середовища. Це забезпечує зменшення ручної праці, підвищення ефективності використання ресурсів, зниження витрат і покращення загального врожаю. Автоматизація також допомагає зменшити людський фактор у прийнятті рішень, що сприяє підвищенню точності і консистентності агропроцесів.

Аналіз даних та прогнозування в системі SmartAgri включає збір та обробку інформації з різних джерел, таких як сенсори, дрони, метеостанції та інші технології. Система використовує ці дані для створення точних прогнозів щодо врожайності, потреби у воді, внесення добрив і боротьби зі шкідниками. Завдяки потужним аналітичним інструментам та алгоритмам машинного навчання, SmartAgri здатна виявляти закономірності та тенденції, які допомагають аграріям приймати обґрунтовані рішення і планувати свої дії. Це не лише знижує ризики, але й оптимізує використання ресурсів, покращує продуктивність і забезпечує максимальний результат від кожного етапу агропроцесу.

Інтеграція з існуючими системами в SmartAgri є критично важливою для забезпечення безперебійної роботи та максимальної ефективності агропроцесів. Система повинна бути сумісною з іншими технологіями, такими як GPS-навігація, дрони для моніторингу, автоматичні системи поливу та управління добривами, а також з уже існуючими програмами для планування і управління. Інтеграція забезпечує єдиний інформаційний простір, де дані з різних джерел об'єднуються і аналізуються в реальному часі. Це дозволяє фермерам отримувати комплексну інформацію, покращує точність прогнозів і

оптимізацію ресурсів, а також спрощує управлінські процеси, що в свою чергу підвищує загальну ефективність і продуктивність агробізнесу.

Покращення прийняття рішень за допомогою системи SmartAgri досягається завдяки інтеграції потужних аналітичних інструментів та алгоритмів обробки даних, які надають аграріям точну та актуальну інформацію про стан посівів, потреби в ресурсах і вплив навколишнього середовища. Система обробляє великі обсяги даних, щоб виявити тренди, аномалії та кореляції, що дозволяє формувати обґрунтовані прогнози та рекомендації. Це забезпечує можливість оперативного реагування на зміни умов, планування ресурсів і прийняття рішень, що базуються на фактичних даних, а не на інтуїції або припущеннях. Як результат, фермери можуть ефективніше управляти своїм господарством, знижуючи ризики і максимізуючи врожайність.

Мобільність та доступність системи SmartAgri забезпечують фермерам можливість управляти своїм господарством з будь-якого місця завдяки доступу до платформи через мобільні пристрої, такі як смартфони та планшети. Ця функція дозволяє отримувати актуальні дані, моніторити стан полів, здійснювати управлінські рішення і контролювати процеси в режимі реального часу, навіть перебуваючи далеко від об'єкта господарства. Мобільний доступ до системи сприяє гнучкому управлінню, оперативному реагуванню на зміни в умовах, швидкому доступу до аналітичних звітів та рекомендацій, що значно підвищує ефективність роботи і дозволяє фермерам зосередитися на стратегічних аспектах ведення бізнесу.

Економія витрат у системі SmartAgri досягається через ефективне управління ресурсами та зменшення витрат на операційні процеси. Система дозволяє оптимізувати використання води, добрив та пестицидів завдяки точному моніторингу і прогнозуванню потреб, що знижує їх надмірне використання і втрати. Автоматизація процесів, таких як полив і внесення добрив, також зменшує необхідність у ручній праці та знижує витрати на трудозатрати. Крім того, завдяки точним даним і прогнозам, фермери можуть

уникати непотрібних витрат і краще планувати свої ресурси, що в свою чергу веде до загального зниження витрат і підвищення рентабельності господарства.

Покращення клієнтського досвіду в системі SmartAgri забезпечується завдяки інтуїтивному інтерфейсу, швидкому доступу до інформації та персоналізованим рішенням. Система дозволяє користувачам легко отримувати аналітичні дані про стан господарства, прогнозувати врожайність і керувати агропроцесами в режимі реального часу. Завдяки інтеграції з мобільними платформами та доступу до звітів через різні пристрої, фермери можуть швидко реагувати на зміни та отримувати своєчасну підтримку. Такий підхід робить управління господарством простішим і зручнішим, підвищуючи задоволеність користувачів і сприяючи їхньому довгостроковому успіху.

Безпека даних у системі SmartAgri є критично важливою для захисту конфіденційної інформації та забезпечення довіри користувачів. Система впроваджує сучасні методи шифрування даних для захисту інформації від несанкціонованого доступу і зловмисних атак. Аутентифікація та авторизація користувачів гарантують, що лише уповноважені особи мають доступ до чутливої інформації та управлінських функцій. Регулярні оновлення безпеки та аудит системи допомагають виявляти та усувати потенційні вразливості, забезпечуючи безперебійний захист даних від будь-яких загроз. Це дозволяє аграріям бути впевненими в збереженні конфіденційності та цілісності їхньої інформації, що є важливим аспектом для довготривалого успіху системи.

Бізнес-вимоги для системи SmartAgri підкреслюють, що ключові вимоги орієнтовані на суттєве підвищення ефективності управління сільськогосподарськими процесами. Система повинна забезпечити автоматизацію агропроцесів, інтеграцію з існуючими технологіями, та аналітику даних для покращення прийняття рішень. Важливою є мобільність і доступність, що дозволяє фермерам управляти господарством з будь-якого місця. Економія витрат на ресурси та покращення клієнтського досвіду також є критичними аспектами, разом із забезпеченням високого рівня безпеки

даних. Загалом, ці вимоги гарантують, що система SmartAgri відповідає стратегічним цілям агробізнесу і приносить реальну бізнес-цінність.

Висновки до Розділу 1

Система SmartAgri представляє собою комплексне рішення для управління сільськогосподарськими процесами, яке включає в себе моніторинг стану посівів, планування ресурсів, аналіз врожайності, оптимізацію робочих процесів, управління технікою, екологічний моніторинг і економічний аналіз.

Впровадження системи SmartAgri базується на принципах гнучкого менеджменту (Agile), що забезпечує ітеративний підхід до розробки. Це дозволяє швидко адаптуватися до нових вимог, забезпечує тісну співпрацю з користувачами, пріоритетність на основі цінності, зворотний зв'язок і самоорганізацію команди. Такий підхід сприяє постійному вдосконаленню системи відповідно до потреб ринку та користувачів.

Цілі проєкту включають розробку інтуїтивно зрозумілого інтерфейсу, запуск мінімального життєздатного продукту (MVP), інтеграцію нових технологій, оптимізацію управління ресурсами та забезпечення гнучкості і масштабованості системи. Усі ці цілі спрямовані на підвищення продуктивності, зниження витрат та забезпечення конкурентоспроможності аграрного сектора.

Бізнес-вимоги для системи SmartAgri включають оптимізацію агропроцесів, аналітику даних та прогнозування, інтеграцію з існуючими системами, і покращення процесу прийняття рішень. Всі ці вимоги сприяють досягненню стратегічних цілей компанії, підвищенню ефективності, зниженню витрат, покращенню клієнтського досвіду та підтримці стійкого зростання на ринку.

Проєкт SmartAgri має стратегічне значення для модернізації сільськогосподарського сектору. Він вирішує ключові проблеми, такі як неефективне управління ресурсами, нестача єдиної платформи для

моніторингу та аналізу, та дозволяє адаптуватися до викликів сучасності, що постають перед аграрною галуззю, завдяки впровадженню сучасних технологій і інноваційних рішень.

Таким чином, реалізація системи SmartAgri забезпечить суттєве покращення управління агропроцесами, підвищить продуктивність і конкурентоспроможність фермерських господарств, а також сприятиме стійкому розвитку аграрного сектора.

РОЗДІЛ 2. ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ СИСТЕМИ КЕРУВАННЯ ФЕРМЕРСЬКИМ ГОСПОДАРСТВОМ SMARTAGRI

2.1 Обґрунтування вибору фреймворку гнучкого управління

Вибір фреймворка для розробки системи є критично важливим через те, що він визначає ефективність управління проектом, адаптацію до змін, управління ризиками, якість кінцевого продукту та організацію комунікації між учасниками. Правильний фреймворк забезпечує структурований підхід до планування та управління ресурсами, що допомагає виявити і вирішити проблеми на ранніх стадіях. Він також дозволяє гнучко реагувати на змінювані вимоги та обставини, що є важливим для успішної реалізації проекту. Вибір відповідного фреймворка підтримує ефективну командну співпрацю, сприяє зниженню ризиків і перевищення витрат, а також забезпечує якість і стабільність кінцевого продукту. Усе це разом дозволяє досягти цілей проекту в рамках визначених термінів і бюджету.

Розглянемо декілька популярних фреймворків гнучкого менеджменту.

Scrum – це фреймворк, що фокусується на ітеративній розробці продукту через короткі цикли (спринти), зазвичай тривалістю 2-4 тижні. Він включає ключові ролі: Scrum Master, Product Owner і команда розробників. Основні артефакти Scrum – це Product Backlog, Sprint Backlog і Increment. Переваги: забезпечує чітку структуру та регулярну оцінку прогресу. Розробка відбувається в коротких циклах, що дозволяє швидко реагувати на зміни і коригувати курс проекту. Недоліки: може бути складно впроваджувати в організаціях, які не звикли до гнучких підходів. Потребує активної участі всіх членів команди та чіткої координації.

Kanban – це метод управління проектами, який фокусується на візуалізації процесу роботи, обмеженні кількості завдань в роботі та постійному покращенні. Завдання відображаються на дошці Kanban, що дозволяє відстежувати їх стан (наприклад, "Заплановано", "В роботі", "Завершено"). Переваги: простий у впровадженні та зрозумілий у

використанні. Дозволяє легко адаптуватися до змін у пріоритетах і обсягах роботи. Недоліки: може бути менш структурованим порівняно зі Scrum. Може бути важко впровадити в проекти з високою складністю або великою кількістю залежностей.

Extreme Programming (XP) – це методологія розробки програмного забезпечення, що акцентує увагу на високій якості коду та частих релізах. Основні практики включають парне програмування, тестування на всіх етапах розробки, часті інтеграції та відстеження змін у вимогах. Переваги: сприяє високій якості коду і швидкому виявленню дефектів. Часті релізи дозволяють швидко отримувати зворотний зв'язок від користувачів. Недоліки: може бути інтенсивним і вимогливим, що не завжди підходить для всіх команд. Потребує високої дисципліни та відданості з боку розробників.

Lean Software Development заснований на принципах Lean Manufacturing і фокусується на максимальному усуненні витрат і оптимізації процесів. Основні принципи включають усунення непотрібної роботи, зменшення часу циклу і підвищення якості. Переваги: допомагає зменшити витрати та підвищити ефективність процесу розробки. Сприяє постійному вдосконаленню і адаптації до змін. Недоліки: може бути важко впроваджувати без попередньої підготовки або без належного розуміння принципів Lean. Потребує постійної уваги до вдосконалення процесів.

Feature-Driven Development (FDD) – це фреймворк, що фокусується на розробці функціональних можливостей продукту. Проект розбивається на функціональні елементи, які реалізуються поетапно. Основні етапи включають моделювання, планування, реалізацію та перевірку функціональностей. Переваги: фокусується на чітких функціональних вимогах, що може бути корисним для проектів з чітко визначеними цілями. Дає змогу створювати функціональні рішення швидше. Недоліки: може бути менш ефективним у середовищах, де вимоги часто змінюються або де є потреба в високій гнучкості.

Вибір фреймворка залежить від специфіки проекту, команди, організації та вимог до гнучкості і контролю.

Вибір фреймворку Scrum для реалізації системи SmartAgri обґрунтований кількома ключовими факторами, які роблять його особливо придатним для цього проекту.

1. Гнучкість і адаптивність: Scrum ідеально підходить для проектів, де вимоги можуть змінюватися протягом розробки. У контексті SmartAgri, де технології та потреби можуть еволюціонувати, Scrum забезпечує можливість швидко адаптуватися до нових вимог і змін у середовищі.

2. Інкрементальна розробка: Scrum організовує розробку у вигляді коротких ітерацій (спринтів), що дозволяє поступово досягати цілей проекту та забезпечувати зворотний зв'язок від стейкхолдерів на кожному етапі. Це дозволяє вчасно виявляти проблеми і коригувати курс розробки.

3. Прозорість і комунікація: Scrum акцентує увагу на регулярних зустрічах, таких як щоденні стендапи, огляди спринтів та ретроспективи, що покращує комунікацію між членами команди і стейкхолдерами. Це сприяє прозорості в управлінні проектом і дозволяє швидко вирішувати проблеми.

4. Визначені ролі і відповідальності: Scrum чітко визначає ролі, такі як Scrum Master, Product Owner і команда розробників, що сприяє ефективному управлінню проектом і забезпечує зрозумілість у розподілі обов'язків та відповідальностей.

5. Управління ризиками: Ітераційний підхід Scrum дозволяє виявляти і управляти ризиками на ранніх етапах, що допомагає уникнути значних проблем у пізніших стадіях розробки.

6. Фокус на цінності для кінцевого користувача: Scrum орієнтований на доставку найбільш цінного функціоналу в першу чергу, що відповідає потребам кінцевих користувачів і забезпечує, що основні функції системи SmartAgri реалізуються швидше.

7. Підтримка постійного вдосконалення: Через ретроспективи і постійний зворотний зв'язок Scrum сприяє безперервному вдосконаленню

процесів і підходів, що допомагає команді адаптуватися і покращувати свою роботу протягом всього проекту.

Вибір фреймворку Scrum для реалізації системи SmartAgri є обґрунтованим рішенням, яке відповідає потребам проекту на різних етапах розробки. Scrum пропонує гнучкість, інкрементальний підхід і чітко визначені ролі, що забезпечують ефективне управління, прозорість і адаптацію до змін. Завдяки регулярним зустрічам і зворотному зв'язку, Scrum дозволяє оперативно вирішувати проблеми, управляти ризиками і забезпечувати високий рівень якості кінцевого продукту. Його орієнтація на створення цінності для кінцевих користувачів та постійне вдосконалення процесів роблять його ідеальним для реалізації інноваційної системи SmartAgri. Вибір Scrum гарантує, що проект буде реалізований ефективно, з урахуванням усіх вимог і очікувань стейкхолдерів, що суттєво підвищує ймовірність успіху проекту.

Scrum це фреймворк для проектів agile менеджменту, що допомагає команді створити структуру команди та управляє їх роботою за допомогою набору цінностей принципів та практик [8].

Фундаментальною одиницею Scrum являється мала команда – Scrum команда. Команда включає в себе Scrum майстра, Product owner та розробників. В середині команди Scrum не існує під-команд та ієрархії [9].

Однією з особливостей фреймворку Scrum являються ролі що в ній присутні. Такими ролями являються:

- **Developers.** Розробники це люди що налаштовані до створення будь якого аспекту придатного для використання Increment кожного спринту [9].
- **Product owner.** Product owner несе відповідальність за максимізацію цінності продукту, отриманого в результаті роботи команди Scrum [9]. В даному проекті є необхідність в 1 product owner, в його задачі входить збір вимог від осіб що зацікавлені в проекті (власники фермерських

господарств), та відстеження increment спринту задовольняв усі задачі поставлені перед спринтом.

- **Scrum master.** Scrum майстер несе відповідальність за створення та налаштування Scrum в команді, як це визначено в Scrum Guide [9]. В даному проєкті scrum майстер відповідальний за навчання співробітників (членів Scrum команди) роботі в Scrum середовищі та за проведення усіх зустрічей необхідних для роботи команди.

Крім ролей, Scrum також додає події [9], такі як:

- **Sprint** це основа Scrum, де ідея перетворюється в цінність, наш проєкт буде розбитий на спринти за які команда має розробити платформу та виконати усі поставлені вимоги до застосунку.

- **Sprint planning** дана подія влаштовується всією командою для створення плану роботи на спринт. Перед кожним спринтом наша команда збирається та розглядає вимоги та створює план дій на час спринту.

- **Daily Scrum** це щоденна зустріч команди на котрій розглядаються питання щодо задач які є на спринті, що зараз триває.

- **Sprint review** на даній події команда збирається та показує зацікавленим особам проєкту, що було зроблено за спринт після чого команда вирішує що необхідно роботи далі.

- **Sprint retrospective** дана подія влаштовується для планування шляхів покращення роботи та якості команди за допомогою аналізу попереднього спринту.

Описані вище основні елементи фреймворку Scrum використовувались для побудови процесів роботи команди що ставить на меті успішне виконання проєкту.

2.2 Планування змісту і тривалості проєкту з розробки MVP

Розуміння і визначення можливих проблем, з якими може зіштовхнутися реалізація проєкту, є критично важливим для успішного планування та

виконання проєкту. Виявлення потенційних проблем дозволяє проактивно розробити стратегії для їх усунення або мінімізації їхнього впливу. Це допомагає уникнути непередбачених затримок, додаткових витрат і зменшити ризик невдачі проєкту. Завчасне виявлення проблем дозволяє забезпечити більш точний розрахунок ресурсів, часу та бюджету, а також налаштувати систему моніторингу і контролю для ефективного управління проєктом.

Реалізація системи SmartAgri, яка орієнтована на автоматизацію та оптимізацію аграрних процесів, може зіштовхнутися з рядом проблем, які потребують ретельного аналізу та планування:

1. Технічні складнощі: Інтеграція сучасних технологій, таких як сенсори, IoT-пристрої та аналітичні платформи, може бути ускладнена проблемами сумісності, налаштування та обслуговування. Технічні помилки або дефекти в обладнанні можуть призвести до збоїв у системі та вплинути на її функціональність.

2. Обробка та зберігання даних: Величезні обсяги даних, які збираються від сенсорів та інших джерел, вимагають ефективних механізмів обробки, зберігання та аналізу. Недостатня потужність системи або проблеми з базами даних можуть уповільнити процеси та ускладнити прийняття рішень.

3. Безпека та конфіденційність: Захист даних від несанкціонованого доступу є критично важливим, особливо у контексті приватної інформації фермерів і комерційних даних. Кібератаки або порушення конфіденційності можуть серйозно зашкодити репутації та ефективності системи.

4. Інтерфейс та зручність використання: Система повинна бути зручною у використанні для кінцевих користувачів, включаючи фермерів з різним рівнем технічної грамотності. Складний інтерфейс або недостатня підтримка користувачів можуть зменшити ефективність і прийняття системи.

5. Фінансування та бюджетування: Розробка та впровадження такої системи потребують значних фінансових вкладень. Нестача коштів або невірно оцінені витрати можуть призвести до затримок або навіть зупинки проєкту.

6. Відповідність нормативним вимогам: Система повинна відповідати всім відповідним нормативним актам та стандартам, що можуть змінюватися. Невідповідність або ігнорування нормативних вимог може призвести до правових проблем і штрафів.

7. Адаптація до змінних умов: Аграрна галузь може бути піддана швидким змінам, таким як зміни в кліматі, технологіях або ринкових умовах. Система повинна бути достатньо гнучкою, щоб адаптуватися до таких змін без значних витрат або затримок.

8. Успішна реалізація системи SmartAgri вимагатиме всебічного підходу до цих проблем, що включає технічну підготовку, належне фінансування, увагу до безпеки та підтримку користувачів.

Планування проєкту реалізації системи SmartAgri повинно ґрунтуватися на усвідомленні та врахуванні зазначених вище проблем. Адекватне планування повинно включати розробку стратегії для вирішення технічних складнощів, забезпечення ефективної обробки та зберігання даних, а також забезпечення безпеки та конфіденційності інформації. Важливо передбачити забезпечення зручного інтерфейсу для користувачів, чітко визначити бюджетні витрати та врахувати нормативні вимоги. Крім того, проєкт повинен бути адаптивним до змінних умов аграрного середовища. Такий комплексний підхід дозволить зменшити ризики і забезпечити ефективну реалізацію та функціонування системи SmartAgri.

Загальний план створення системи SmartAgri включає 8 основних етапів:

1. Визначення мети проєкту та основних вимог
2. Планування проєкту
3. Створення системної архітектури
4. Розробка MVP (Minimum Viable Product)
5. Тестування та валідація
6. Впровадження та інтеграція
7. Підтримка та подальший розвиток
8. Моніторинг ефективності

Найбільш відповідальний та найдовший, не враховуючи підтримку готового продукту, етап це розробка MVP (Minimum Viable Product). Цей етап має на меті створення готового продукту який буде задовольняти головні вимоги висунуті до системи SmartAgri. Цей етап доцільно реалізовувати з використанням фреймворку Scrum.

Після планування усіх робіт з розробки MVP команда розробників отримала плановий roadmap за яким вона буде працювати протягом наступних 5 місяців. У разі виникнення проблем або різних ситуації, що можуть вплинути на план, roadmap буде змінюватись за необхідністю команди. Roadmap розробки MVP зображено на Рисунок 2.1.



Рисунок 2.1 – Roadmap розробки MVP

Джерело: розроблено автором

Рисунок 2.1 відображає план виконання розробки MVP. На даний момент розробки MVP розбито на 7 спринтів що тривають по 3 тижні, командою заплановано що усі цілі проекту буде досягнено впродовж 5 місяців при умові відсутності додаткових проблем та помилок.

Нижче наведено короткий опис цілей для кожного спринту в розробці MVP системи SmartAgri.

Sprint 1: Налаштування інфраструктури та інтеграція IoT. Ціль: Створення базової інфраструктури та налаштування інтеграції з IoT-пристроями для збору даних.

Sprint 2: Обробка та аналіз даних. Ціль: Розробка модулів для обробки та зберігання даних, а також початкова аналітика з використанням машинного навчання.

Sprint 3: Розробка інтерфейсу користувача (UI/UX). Ціль: Створення інтуїтивного інтерфейсу користувача для відображення даних у режимі реального часу.

Sprint 4: Інтеграція з системою та IoT. Ціль: Завершення інтеграції IoT-пристроїв, налагодження збору та обробки даних у реальному часі.

Sprint 5: Аналітика та тестування. Ціль: Впровадження алгоритмів прогнозування та функціональне тестування всіх основних компонентів системи.

Sprint 6: Покращення інтерфейсу та користувацьке тестування. Ціль: Оптимізація інтерфейсу на основі зворотного зв'язку від користувачів та завершення нефункціонального тестування.

Sprint 7: Пілотний запуск та інтеграція. Ціль: Пілотний запуск системи на реальних фермах, інтеграція з існуючими системами та навчання користувачів.

Ці спринти поступово розвивають проєкт від налаштування інфраструктури до повного впровадження і тестування системи.

Запланованою датою початку проєкту є 30 вересня 2024 року. При умові відсутності необхідності проведення додаткових спринтів та серйозних проблем при розробці, що можуть призвести до затримки розробки, планова дата закінчення розробки MVP – 24 Лютого 2025 року.

Під час планування спринтів, команда розподілила story points між завданнями. Після виконання двох завдань передбачених першим спринтом, стало зрозуміло що 1 story point це приблизно 2 робочих дні, один робочий день у команді це 8 годин.

На даний момент в проєкті заплановано 7 спринтів, терміни та закладені story point кожного спринта зображено на Рисунок 2.2.



Рисунок 2.2 – Рисунок запланованих спринтів

Джерело: розроблено автором

Рисунок вище показує заплановані спринти, дату початку та закінчення спринту та story points на спринт.

Таким чином маючи дану інформацію можна вирахувати кількість запланованих людино-годин в проєкті, без урахування додаткової роботи при виникненні непередбачених помилок, що призведуть збільшення завдань які необхідно виконати.

Загальна кількість story points:

$$SP_{sum} = 29 + 34 + 21 + 33 + 13 + 29 + 21 = 180 SP$$

Годин на 1 story point:

$$t_{SP} = 2 \text{ дні} * 8 \text{ робочих годин} = 16 \text{ годин}$$

Команда проєту складається з 12 спеціалістів, дивись розділ 2.3.

$$N_{empl} = 14$$

Загальна кількість людино-годин:

$$N_{люд.год.} = SP_{sum} * t_{SP} * N_{empl} = 180 * 16 * 14 = 40\,320 \text{ люд. год.}$$

Обсяг передбачуваних робіт становить 40 320 людино годин.

2.3 Планування команди проєкту

Створення команди для розробки MVP системи SmartAgri передбачає ретельний відбір спеціалістів з різними компетенціями, які відповідають специфічним вимогам проєкту. Для ефективної роботи команди необхідно забезпечити чітке визначення ролей і обов'язків, створити відкриту комунікацію та механізми для вирішення конфліктів.

Розрахуємо необхідну кількість робітників в команді. Зазвичай один розробник або інший технічний спеціаліст може виконати близько 8-10 сторі поінтів за один спринт, залежно від їхньої складності та додаткових факторів (відпочинок, інші проєкти тощо).

Якщо взяти середній показник продуктивності 9 SP на спринт, то для виконання 180 сторі поінтів потрібно:

$$N_{\text{людино-спринтів}} = \frac{180 \text{ SP}}{9 \text{ SP/виконавця}} = 20 \text{ людино – спринтів}$$

Для формування команди проєкту розглянемо кожне завдання в спринту, кількість story points для кожного завдання та спеціаліста який буде виконувати задачу, усі дані систематизовані в Таблиці 2.1.

Таблиця 2.1 – Список необхідних спеціалістів для кожного завдання

Спринт	Задача	Story points	Виконавець
1	Налаштування бекенд-сервера.	5 SP	Бекенд-розробник
	Конфігурація серверного середовища для обробки даних.	3 SP	Бекенд-розробник
	Встановлення системи моніторингу сервера.	5 SP	Системний адміністратор
	Налаштування системи резервного копіювання.	3 SP	Системний адміністратор
	Інтеграція датчиків вологості та температури.	8 SP	ІоТ-спеціаліст

Таблиця 2.1 – Список необхідних спеціалістів для кожного завдання

Спри нт	Задача	Story points	Виконавець
	Розробка базових API для збору даних.	8 SP	Бекенд-розробник
2	Створення бази даних для зберігання даних з IoT.	8 SP	Бекенд-розробник
	Розробка механізму резервного копіювання бази даних.	5 SP	Системний адміністратор
	Налаштування системи моніторингу бази даних.	3 SP	Системний адміністратор
	Створення модуля для аналітики початкових даних.	8 SP	Аналітик даних
	Впровадження алгоритмів для початкового аналізу.	5 SP	Фахівець з машинного навчання
3	Розробка дашборду для відображення зібраних даних.	8 SP	Фронтенд-розробник
	Створення інтуїтивно зрозумілого дизайну для мобільних пристроїв.	5 SP	UI/UX-дизайнер
4	Інтеграція з додатковими IoT-пристроями.	8 SP	IoT-спеціаліст
	Розробка документації по інтеграції для нових пристроїв.	5 SP	Аналітик даних
	Проведення тестування з'єднання всіх IoT-пристроїв.	3 SP	IoT-спеціаліст
	Налаштування та тестування API для зовнішніх систем.	5 SP	Бекенд-розробник
	Впровадження аутентифікації для API.	5 SP	Бекенд-розробник
	Оптимізація роботи з базою даних.	5 SP	Бекенд-розробник
5	Інтеграція зовнішніх джерел даних для порівняння.	8 SP	Аналітик даних
	Розробка візуалізації аналітичних даних.	5 SP	Фронтенд-розробник
	Розробка алгоритмів для прогнозування врожайності.	8 SP	Фахівець з машинного навчання
	Інтеграція алгоритмів машинного навчання.	5 SP	Фахівець з машинного навчання
	Впровадження механізму зворотного зв'язку для покращення моделей.	5 SP	Аналітик даних

Таблиця 2.1 – Список необхідних спеціалістів для кожного завдання

Спри нт	Задача	Story points	Виконавець
	Налаштування системи оцінки точності прогнозів.	3 SP	QA-інженер
6	Оптимізація UX на основі зворотного зв'язку.	5 SP	UI/UX-дизайнер
	Розробка механізму нотифікацій для користувачів.	3 SP	Фронтенд-розробник
	Проведення функціонального тестування всіх компонентів.	5 SP	QA-інженер
	Нефункціональне тестування (стабільність, безпека).	5 SP	QA-інженер
	Тестування на різних браузерах та пристроях.	3 SP	QA-інженер
	Проведення тестування навантаження для оцінки продуктивності.	5 SP	QA-інженер
7	Проведення пілотного запуску системи.	8 SP	DevOps-інженер
	Навчання користувачів роботі з системою.	5 SP	Аналітик даних
	Збір відгуків від користувачів під час пілотного запуску.	3 SP	Аналітик даних
	Виправлення виявлених проблем під час пілотного запуску.	5 SP	QA-інженер

Підрахуємо загальну кількість story points, яка відводиться на одного спеціаліста та необхідну кількість спеціалістів виходячи з того що для покращення продуктивності, при необхідності у більше ніж 20 SP/людину, запросимо у команду 2 спеціалістів, при необхідності у менше ніж 20 SP/людину – 1 спеціаліста. Необхідна кількість спеціалістів порахована в Таблиці 2.2.

Таблиця 2.2 – Розрахунок кількості виконавців

Виконавець	Story points	Кількість виконавців
DevOps-інженер	8	1
IoT-спеціаліст	19	1
QA-інженер	26	2
UI/UX-дизайнер	10	1
Аналітик даних	34	2
Бекенд-розробник	39	2
Системний адміністратор	16	1
Фахівець з машинного навчання	18	1
Фронтенд-розробник	16	1
Загальна кількість спеціалістів:		12

Таким чином команда складається з 12 спеціалістів також у склад команди входять product owner та Scrum майстер.

$$N_{empl} = 14$$

Обов'язки кожного спеціаліста команди представлені нижче.

Бекенд-розробник. Налаштовує серверну інфраструктуру, розробляє API для збору та обробки даних, інтегрує IoT-пристрої, працює з базою даних і оптимізує бекенд для обробки великих обсягів інформації.

IoT-спеціаліст. Відповідає за вибір, налаштування IoT-платформи та інтеграцію пристроїв (датчики вологості, температури, GPS), забезпечує стабільну передачу даних до серверів і тестує з'єднання.

Аналітик даних. Створює модулі для обробки та аналізу зібраних даних, впроваджує алгоритми статистики й оптимізації для великих масивів даних, інтегрує аналітичні алгоритми в систему.

Фахівець з машинного навчання. Впроваджує моделі прогнозування на основі даних, оптимізує алгоритми для реального часу, тестує та вдосконалює точність прогнозів для підтримки прийняття рішень.

Фронтенд-розробник. Розробляє інтерфейс користувача для взаємодії з даними IoT у режимі реального часу, інтегрує фронтенд із бекендом, забезпечує зручність використання та впроваджує зміни на основі зворотного зв'язку.

UI/UX-дизайнер. Розробляє дизайн інтерфейсу з фокусом на зручність та інтуїтивність, покращує користувацький досвід, впроваджує нотифікації й поліпшення на основі відгуків.

QA-інженер. Проводить функціональне та нефункціональне тестування, тестує з'єднання IoT-серверів, перевіряє стабільність, продуктивність і безпеку системи, а також допомагає виправляти баги.

DevOps-інженер. Налаштовує серверну інфраструктуру, забезпечує безперебійну роботу системи через автоматизацію розгортання, моніторинг серверів і підтримку стабільності під час пілотних запусків.

Розглянемо канали пошуку фахівців:

а) професійні соціальні мережі (LinkedIn): тут можна знайти фахівців різних галузей та залучити їх до проєкту через прямі запити або публікацію вакансій;

б) кар'єрні портали (Indeed, Glassdoor): розміщення оголошень про вакансії допоможе знайти кандидатів, які активно шукають роботу;

в) університети та навчальні заклади: співпраця з університетами може допомогти залучити молодих фахівців та стажерів;

г) професійні конференції та заходи: відвідування галузевих заходів дозволяє знайти фахівців, які активно беруть участь у розвитку своєї професії;

д) рекомендації та мережеві контакти: використання особистих та професійних контактів для пошуку та рекомендацій фахівців.

Розглянемо умови залучення фахівців:

а) конкурентна заробітна плата та бонуси: забезпечення привабливих умов оплати праці для залучення висококваліфікованих фахівців;

б) гнучкий графік роботи: надання можливості працювати дистанційно та гнучкого робочого графіку для залучення талановитих фахівців;

в) професійний розвиток: можливості для навчання та професійного розвитку, участь у тренінгах та конференціях;

г) командна робота та підтримка: створення сприятливого робочого середовища, де фахівці можуть працювати ефективно та комфортно.

2.4 Планування вартості проєкту з розробки MVP

Розглянемо бюджет розробки MVP який складається з заробітної плати спеціалістів, витрат на придбання ліцензій необхідного програмного забезпечення, придбання обладнання та інші витрати.

Розрахуємо суму необхідну для виплати заробітної плати, яка розраховувалась за допомогою перемноження обсягів передбачуваних робіт на середню заробітну плату за годину роботи спеціалістів команди, що становить 22 долара на годину, пораховано в Таблиці 2.3

Таблиця 2.3 – Середня заробітна плата спеціаліста за годину

Виконавець	Середня ЗП за годину
DevOps-інженер	30 доларів
ІюТ-спеціаліст	30 доларів
QA-інженер	18 доларів
UI/UX-дизайнер	15 доларів
Аналітик даних	25 доларів
Бекенд-розробник	20 доларів
Системний адміністратор	10 доларів
Фахівець з машинного навчання	30 доларів
Фронтенд-розробник	20 доларів
Product owner	15 доларів
Scrum майстер	15 доларів

$$ЗП_{\text{сер.}} = \frac{30 + 30 + 18 + 15 + 25 + 20 + 10 + 30 + 20 + 15 + 15}{12} = 21 \text{ дол.}$$

Розрахуємо загальну суму витрат на заробітну плату:

$$N_{\text{люд.год.}} = 40320 \text{ люд. год} - \text{див. Розділ 2.2}$$

$$ЗП_{\text{загальна}} = ЗП_{\text{сер.год.}} * N_{\text{люд.год.}} = 21 \text{ дол.} * 40320 \text{ люд. год.} = 835724 \text{ дол.}$$

Отже частина бюджету закладена на заробітну плату це 835,8 тисяч доларів, Таблиця 2.4. показує розбивку бюджету проєкту.

Окрім заробітної плати, 5,5 тисяч доларів закладені на інші витрати команди. Такими витратами можуть бути, проведення корпоративних заходів (святкування свят, тимблідіг) та проведення необхідних тренінгів для збільшення компетенцій команди.

10 тисяч доларів закладено на придбання необхідних ліцензій для програмного забезпечення що буде використовуватись для розробки системи SmartAgri.

20 тисяч доларів закладено як витрати на необхідне обладнання. Дані кошти підуть на придбання серверного обладнання та його підтримку протягом періоду виконання проєкту.

Таблиця 2.4 – Загальний бюджет проєкту

Період проєкту	Загальна сума, доларів	Заробітна плата, доларів	Витрати на ліцензії, доларів	Витрати на обладнання, доларів	Інші витрати, доларів
На весь проєкт	870 800	835 800	10 000	20 000	5 500
Перший місяць роботи	186 160	167 160	6 000	12 000	1 500
Другий місяць роботи	171 160	167 160	1 000	2 000	1 000
Третій місяць роботи	171 160	167160	1 000	2 000	1 000
Четвертий місяць роботи	171 160	167 160	1 000	2 000	1 000
П'ятий місяць роботи	171 160	167 160	1 000	2 000	1 000

Окрім основних коштів закладаємо 150 тисяч доларів як резервних коштів, для використання лише в екстрених випадках. Екстрені кошти розподілені наступним чином:

- 135 тисяч виділяється на заробітну плату,
- 5 тисяч виділяється на ліцензії програмного забезпечення
- 10 тисяч виділяється на обладнання для команди.

Загальний бюджет на розробку MVP системи SmartAgri становить 1 020,8 тисяч доларів, з яких 870,8 тисяч доларів основних коштів та 150 тисяч доларів резервних коштів, які можуть бути не використані при відсутності необхідності.

2.5 Планування процесів проєкту

Управління вимогами — це систематичний і важливий процес у рамках управління проєктами та розробки програмного забезпечення, спрямований на досягнення кількох найважливіших цілей. Ці цілі покликані гарантувати, що вимоги проєкту фіксуються, документуються, аналізуються та контролюються таким чином, щоб забезпечити успішні результати проєкту та задоволені зацікавлені сторони [10].

Команда проєкту проводить requirements management за наступними етапами:

Збір вимог. На першому етапі здійснюється ідентифікація бізнес-вимог. Спільно з основними стейкхолдерами (фермерами, інвесторами, державними органами) визначаються ключові функціональні можливості системи, які мають бути реалізовані в MVP. Це можуть бути вимоги до збору та обробки даних з IoT-пристроїв, інтеграція з іншими фермерськими системами, необхідність алгоритмів прогнозування та інше.

Аналіз та уточнення вимог. Після збору, вимоги аналізуються на предмет технічної реалізованості, пріоритетності та відповідності цілям проєкту. В цьому процесі беруть участь як бізнес-аналітики, так і технічні експерти, що допомагають виявити будь-які ризики або технічні виклики, що можуть

виникнути при реалізації певних вимог. Задається питання про те, які з вимог є критичними для MVP, а які можна відкласти на майбутні етапи розвитку продукту.

Документування вимог. Вимоги фіксуються у вигляді детальної документації. Це можуть бути специфікації користувачьких історій (user stories), функціональні та нефункціональні вимоги, що описують очікувану поведінку системи. Для кращого управління можуть використовуватись системи управління вимогами, такі як Jira або Confluence, які дозволяють структурувати, відслідковувати і оновлювати вимоги на різних етапах проєкту.

Валідація та затвердження вимог. Після документування вимоги мають бути перевірені і затверджені стейкхолдерами. Це допомагає переконатися, що всі ключові учасники проєкту погоджуються з визначеними вимогами і немає непорозумінь щодо їхнього змісту або важливості.

Управління змінами вимог. Протягом усього життєвого циклу проєкту може виникнути необхідність змін у вимогах через зворотний зв'язок користувачів або виявлення нових технічних можливостей. Управління змінами включає оцінку впливу змін на терміни, бюджет та технічні аспекти проєкту. Важливо мати чітко визначений процес для фіксації та впровадження таких змін, щоб уникнути хаосу та затримок.

Відстеження виконання вимог. На кожному етапі розробки необхідно забезпечувати відстеження прогресу щодо реалізації вимог. Для цього використовуються системи моніторингу виконання завдань та звітності, що допомагають побачити, на якому етапі знаходиться розробка тієї чи іншої функціональності і чи відповідає вона початковим вимогам.

Тестування вимог. На етапі тестування здійснюється перевірка відповідності реалізованої функціональності початковим вимогам. QA-інженери розробляють тест-кейси на основі вимог і перевіряють, чи виконуються всі заявлені функції належним чином.

Управління вимогами є критичним елементом успішної розробки MVP для SmartAgri, оскільки це дозволяє чітко визначати цілі, зберігати фокус на ключових функціях продукту та забезпечувати їх своєчасне та якісне впровадження.

Планування тестування MVP.

План тестування для системи SmartAgri має включати кілька ключових етапів, щоб переконатися в тому, що всі компоненти системи працюють належним чином, і відповідають вимогам стейкхолдерів. У таблиці 2.5 наведено план тестування.

Таблиця 2.5 - План Тестування

Етап	Задача	Опис Задачі
Підготовка до тестування	Оцінка вимог та специфікацій	Перевірка вимог і функціональних специфікацій для забезпечення точного розуміння того, що необхідно тестувати.
	Планування тестування	Визначення типів тестів (функціональне, нефункціональне, інтеграційне тощо) та тестових підходів.
	Огляд середовища тестування	Підготовка необхідного обладнання, інструментів і ресурсів для тестування IoT-пристроїв, серверів та аналітичних модулів.
Функціональне тестування	Тестування збору даних з IoT-пристроїв	Перевірка, чи коректно передаються дані від IoT-пристроїв у систему та чи відповідають вони реальним показникам.
	Тестування обробки даних	Перевірка коректності обробки даних, включаючи їх збереження в базі даних, базову аналітику та передачу до UI.
	Тестування користувацьких інтерфейсів (UI/UX)	Перевірка взаємодії користувачів з системою через фронтенд-інтерфейс, зокрема дашборди, графіки та форми введення даних.
Інтеграційне тестування	Тестування взаємодії між компонентами	Перевірка коректної роботи інтеграції між IoT-пристроями, серверною інфраструктурою та базою даних.
	Інтеграція аналітичних модулів	Перевірка взаємодії між модулями для аналізу даних та їх відображенням у реальному часі в користувацькому інтерфейсі.
Нефункціональне тестування	Тестування продуктивності	Оцінка продуктивності системи під різним навантаженням, включаючи одночасну роботу з великою кількістю даних і IoT-пристроїв.

Таблиця 2.5 - План Тестування

Етап	Задача	Опис Задачі
	Тестування стабільності	Перевірка, як система працює протягом тривалого часу, зокрема під навантаженням, і її здатність до відновлення після помилок.
	Тестування безпеки	Перевірка системи на можливі вразливості, включаючи захист даних, аутентифікацію користувачів та захист API.
Тестування з користувачами	Користувацьке тестування	Запуск системи для обмеженої кількості користувачів (фермерів та інших стейкхолдерів) для отримання зворотного зв'язку та перевірки зручності використання.
	Збір відгуків	Визначення проблем або недоліків, з якими стикнулися користувачі під час роботи з системою, для подальших покращень
Тестування перед пілотним запуском	Фінальне функціональне тестування	Повторне тестування основних функцій системи для перевірки відсутності критичних багів.
	Тестування підготовки до запуску	Перевірка всіх компонентів на відповідність вимогам перед пілотним запуском на тестовій фермі
Звітність та підведення підсумків	Звіт про результати тестування	Створення звіту з докладним описом знайдених помилок, їхнього статусу та подальших кроків для їхнього виправлення.
	Розробка рекомендацій щодо поліпшення	Виявлення можливостей для покращення продуктивності, зручності використання та функціональності системи.

Цей план тестування забезпечує всебічну перевірку системи SmartAgri і гарантує, що всі компоненти працюють відповідно до вимог, що є важливим для успішного пілотного запуску системи.

Після завершення тестування, тестувальник задокументує усі знайдені помилки та дії які він виконував для отримання конкретної помилки, та деякі відомості про протестовану частину програмного системи. Такими даними є: швидкість виконання процесів, навантаження системи під час виконання, наявності візуальних багів або нестиківок інтерфейсу, що виникли під час виконання програми.

Командою було вирішено що на проєкті вони будуть описувати 3 категорії багів (помилки у роботі програми).

Critical – баги які критично впливають на роботу програми та можуть призвести до припинення її роботи.

Major – баги які серйозно впливають на роботу програми але не призводять до припинення її роботи, дані баги можуть впливати на роботу деяких функцій.

Minor – баги які не порушують функціонал програми але може бути виявлений візуально, наприклад помилка в тексті або дизайні.

Scrum церемонії проєкту.

Після планування усіх спринтів команда розробила наступний графік церемоній Scrum при врахуванні того що команда працює за 8-годинним робочим днем (з 9:00 до 18:00 з 1 годиною виділеною на обідню перерву). Scrum-церемонії організовуються таким чином, щоб забезпечити ефективну комунікацію, планування та ретроспективний аналіз. Нижче описано ключові Scrum-церемонії

Daily Standup будуть проводитись кожен день о 9:15 по 15 хвилин. Під час якого кожен член команди коротко звітує про виконані задачі, перешкоди і план на день. Мітинг проводиться Онлайн (Microsoft Teams), офлайн (Мітинг рум в офісі) або змішано. Учасники відповідають на три питання: що було зроблено вчора, що планується на сьогодні, і чи є якісь перешкоди.

Sprint Planning буде проводитись кожний перший день спринту (кожен третій понеділок) та тривати 2-3 години. Під час зустрічі визначаються задачі для наступного спринту на основі Product Backlog і розподіляються серед команди. Обговорюються цілі спринту і деталі задач. Формат зустрічі аналогічний Daily Standup, хоча офлайн зустріч пріоритетніша.

Sprint Review буде проводитись ввечері останнього дня спринту та триватиме до 1 години. Мета зустрічі аналіз процесу роботи команди під час спринту та обговорення успіхів, проблем та шляхів покращення процесу на майбутні спринти.

Backlog Refinement буде проводитись кожен четвер та тривати 1-2 години. Головне завдання команди це Очищення та пріоритизація Product Backlog. Задачі деталізуються, оцінюються та готуються для наступного спринту.

Також при необхідності плануються зустрічі для вирішення специфічних технічних або організаційних питань можуть бути організовані за потребою. Вони плануються у зручний для команди час.

Загалом, скрам-церемонії інтегровані в робочий графік команди так, щоб кожен учасник був залучений до процесу розробки, з регулярними можливостями для аналізу та покращення.

2.6 Моніторинг виконання проєкту

Показники стану виконання.

Burndown Chart — це графічне зображення роботи, яку залишилося виконати, у порівнянні з часом. Невиконана робота (або відставання) часто відкладається по вертикальній осі, а час – по горизонталі. Burndown Chart — це графік роботи, що залишилася. Це корисно для прогнозування, коли вся робота буде завершена. Він часто використовується в гнучких методологіях розробки програмного забезпечення, таких як Scrum. Однак Burndown Chart можна застосувати до будь-якого проєкту, який містить вимірний прогрес з часом [11]. Приклад Burndown Chart зображено на Рисунок 2.3.

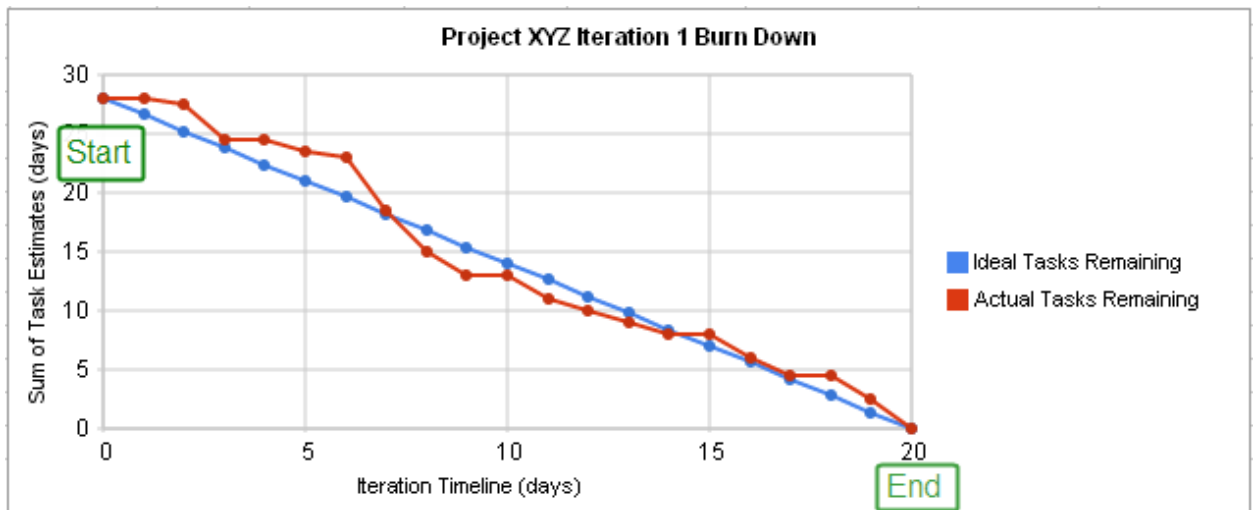


Рисунок 2.3 – Приклад Burndown Chart

Джерело: [11].

Поняття швидкості можна легко пояснити. Основна мета Velocity chart полягає в тому, щоб побачити, скільки роботи було виконано для кожного спринту. Це допоможе вам мати чітке уявлення про майбутні перспективи та навантаження вашої команди [12].

Щоб оцінити цей показник, обчислити його, вам потрібно визначити одиниці роботи для кожного завдання та тривалість кожного інтервалу (час). Під час спринтів кожен член команди повинен повідомляти про виконані або завершені завдання. І в кінці інтервалу, який ми хочемо проаналізувати, підраховується виконана кількість одиниць роботи [12].

Якщо ми застосуємо цю формулу, ми матимемо одну з найважливіших метрик для методології Scrum — швидкість робочого потоку [12]. Приклад Velocity Chart зображено на Рисунок 2.4.

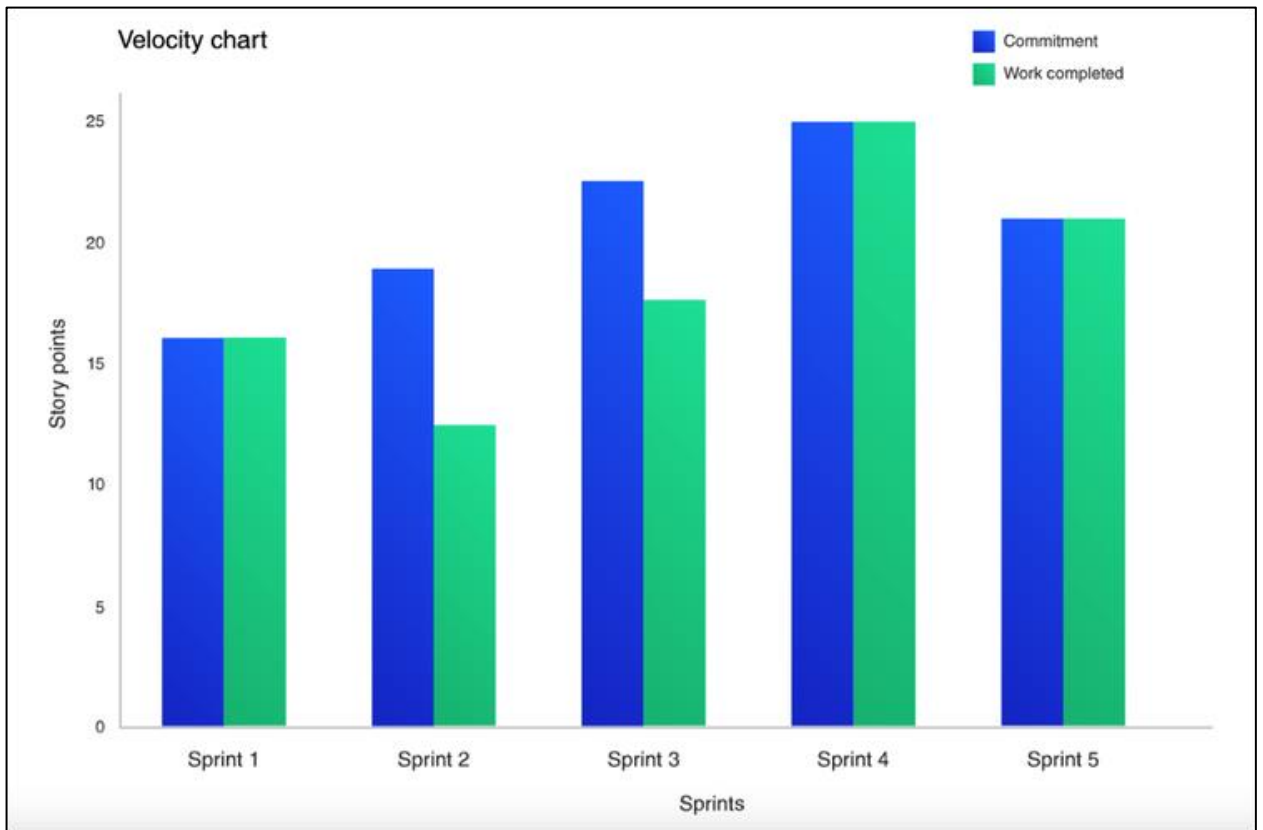


Рисунок 2.4 – Приклад Velocity Chart

Джерело: [12].

Висновки по розділу 2

Успішна реалізація проєкту SmartAgri вимагає ретельного планування з урахуванням можливих проблем, таких як технічні складнощі, питання безпеки, обробка та зберігання даних, а також відповідність нормативним вимогам. Про активне виявлення та аналіз цих проблем дозволяє розробити стратегії для їх усунення, що, в свою чергу, зменшує ризики затримок та додаткових витрат. Проєкт складається з восьми основних етапів, серед яких ключовим є розробка MVP (Minimum Viable Product), що реалізується через фреймворк Scrum. Чітко сплановані спринти, в якій розподілено завдання, допоможуть досягти цілей проєкту в терміни, попри можливі виклики. Загальна кількість запланованих людино-годин для реалізації проєкту становить 40 320, що підкреслює обсяги та ресурси, необхідні для успішної

роботи над системою. Комплексний підхід до управління проектом забезпечить його ефективну реалізацію та стабільну роботу в аграрному середовищі.

Також у даному розділі було детально розглянуто планування вартості та процесів розробки MVP системи SmartAgri. Бюджет проекту складається з витрат на заробітну плату команди, ліцензії на програмне забезпечення, обладнання та інші витрати. Загальна сума бюджету становить 1 020,8 тисяч доларів, з яких 870,8 тисяч доларів – основні кошти, а 150 тисяч доларів – резервні, призначені для екстрених випадків.

Важливим аспектом успішного виконання проекту є управління вимогами. Процес включає збір, аналіз, документування та валідацію вимог, що дозволяє забезпечити їх відповідність цілям проекту. Поряд з цим, планування тестування MVP передбачає кілька етапів, спрямованих на перевірку функціональності та якості системи.

Систематизація Scrum-церемоній, таких як Daily Standup, Sprint Planning, Sprint Review та Backlog Refinement, забезпечує ефективну комунікацію та моніторинг прогресу команди. Застосування показників, таких як Burndown Chart і Velocity Chart, дозволяє відстежувати стан виконання проекту та оцінювати продуктивність команди.

Узагальнюючи, проведене планування та організація роботи команди сприяють досягненню цілей проекту SmartAgri, що в свою чергу забезпечить успішний запуск системи та задоволення потреб зацікавлених сторін.

РОЗДІЛ 3. РЕЗУЛЬТАТИ РОЗРОБКИ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КОМАНДИ З РОЗРОБКИ SMARTAGRI

3.1 Огляд інкременту системи SmartAgri

Інкремент – це усі пункти беклогу зі статусом «виконано» (Done), які може переглянути і сам замовник під час перегляду спринту. Звісно, інкремент може бути надісланий клієнту ще до кінця спринту [13].

Інкремент системи SmartAgri – це ключовий результат, який створюється наприкінці кожного спринту в рамках Agile-підходу до розробки. Кожен інкремент представляє собою функціонально готову частину системи, яка може бути протестована та впроваджена. Основна мета кожного інкременту – надати користувачам додану вартість, поступово розширюючи можливості системи.

Приклад інкременту SmartAgri зображений на Рисунок 3.1.

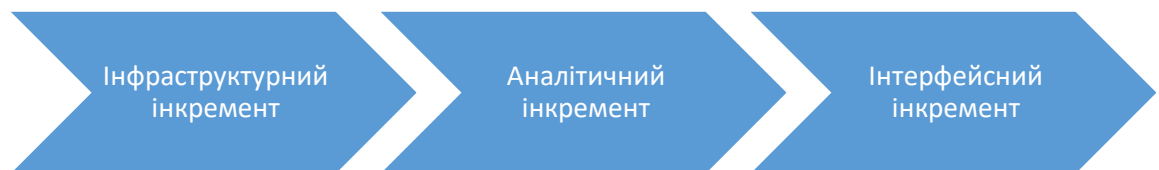


Рисунок 3.1 – Приклад Інкременту Системи SmartAgri

Джерело: розроблено автором

Розглянемо кожний інкремент окремо.

Інфраструктурний інкремент — це покращення або зміни в технічній інфраструктурі системи, які необхідні для підтримки функціональних можливостей продукту. Це стосується будь-яких технічних аспектів, що покращують надійність, масштабованість, безпеку або продуктивність системи, але не обов'язково впливають на кінцевого користувача безпосередньо.

Для проєкту SmartAgri інфраструктурний інкремент може включати налаштування серверів для обробки даних, впровадження систем моніторингу, конфігурацію мережевих підключень, інтеграцію сенсорів та датчиків IoT, налаштування резервного копіювання та розширення базової інфраструктури для обробки великих обсягів даних. Інфраструктурний інкремент є результатом виконання завдань запланованих у Першому та Другому Спринтах.

Основна мета інфраструктурного інкременту — забезпечити стабільну, гнучку та безпечну основу для функціонування основних сервісів і подальших розробок системи.

Аналітичний інкремент — це етап у розробці системи, що стосується вдосконалення функцій, пов'язаних зі збором, обробкою та аналізом даних для прийняття рішень.

Для проєкту SmartAgri цей інкремент включає інтеграцію аналітичних інструментів для збору даних з різних джерел (наприклад, IoT-сенсорів), їх обробку, візуалізацію та створення алгоритмів прогнозування, що допоможуть аграріям оптимізувати процеси на основі отриманих даних. Аналітичний інкремент забезпечує можливість аналізу показників, таких як рівень вологості, стан ґрунту або погодні умови, що дає змогу покращити управління фермерським господарством. Аналітичний інкремент стане результатом виконання завдань запланованих у Четвертому та П'ятому Спринтах.

Зібрані та проаналізовані дані лягли в основу методології прогнозування врожайності та створення системи прогнозування. Ця система використовує методи машинного навчання та алгоритми аналітики для передбачення критичних показників, таких як погодні умови, оптимальні строки поливу або можливі ризики для врожаю. На основі даних із сенсорів, аналітична платформа генерує прогнози, що дозволяють фермерам приймати обґрунтовані рішення, покращуючи управління ресурсами та підвищуючи ефективність виробництва.

Інтерфейсний інкремент в системі SmartAgri фокусується на розробці та вдосконаленні користувацького інтерфейсу (UI) для ефективної взаємодії користувачів із системою. Цей етап передбачає створення зручної, інтуїтивно зрозумілої панелі керування для фермерів, де вони можуть переглядати дані з різних сенсорів, графіки аналізу та прогнози. Інтерфейсний інкремент стане результатом виконання завдань запланованих у Третьому та Шостому Спринтах.

Також важливим аспектом є забезпечення адаптивного дизайну, що дозволить користуватися системою на різних пристроях (ПК, планшети, смартфони). Інтерфейс надає можливість налаштовувати сповіщення про критичні ситуації, такі як зміни у вологості ґрунту чи прогнозовані ризики для врожаю, допомагаючи фермерам приймати оперативні рішення в реальному часі.

З описаного вище можна зробити висновок що важливо поступово розширювати функціональність системи через впровадження інфраструктурних, аналітичних та інтерфейсних рішень. Кожен інкремент спрямований на вирішення окремих аспектів — від забезпечення стабільної технічної основи, збору та обробки даних для прогнозування, до покращення користувацького досвіду. Такий підхід дозволяє поетапно створювати продукт, що додає цінність користувачам на кожному етапі розробки, сприяючи покращенню управління аграрними процесами та підвищенню ефективності фермерського господарства.

3.2 Ретроспектива роботи команди

Ретроспектива — це формальна можливість оцінки й аналізу проведеної роботи. Ретро покриває дуже широкий спектр специфічних цілей, які слід узгодити із ціллю верхнього рівня: виробити кроки для поліпшення процесу й підвищення ефективності роботи команди [14].

Ретроспектива проводиться після кожного спринту, щоб проаналізувати співпрацю. У процесі збираються підходи до того, як ще успішніше працювати команді у наступному спринті [14].

З одного боку, кожен конкретний випадок може потребувати свого особливого ретроформату. З іншого боку, у всіх хороших ретроспектив є певний набір атрибутів, що дадуть їм змогу проходити максимально продуктивно [14].

Майже всі методи безперервного вдосконалення базуються на так званому циклі DMAIC (Define – Measure – Analyse – Improve – Control). Приклад простого циклу зображено на Рисунок 3.2. Ці методи мають різну кількість кроків. В основі кожного циклу лежать вимірювання, аналіз та імплементація [14].



Рисунок 4.2 – Приклад простого циклу DMAIC

Джерело: розроблено автором

Після пов'язування цих фаз з командою, стає зрозуміло, чому ретроспектива є важливим етапом роботи команди в гнучкому менеджменті. Ретроспектива є формою співпраці під час якої команда проводить аналіз своєї роботи та визначає заходи для постійного покращення процесів [14].

Для проєкту розробки MVP системи процес ретроспективи включає чотири основні кроки зображені на Рисунок 3.3.

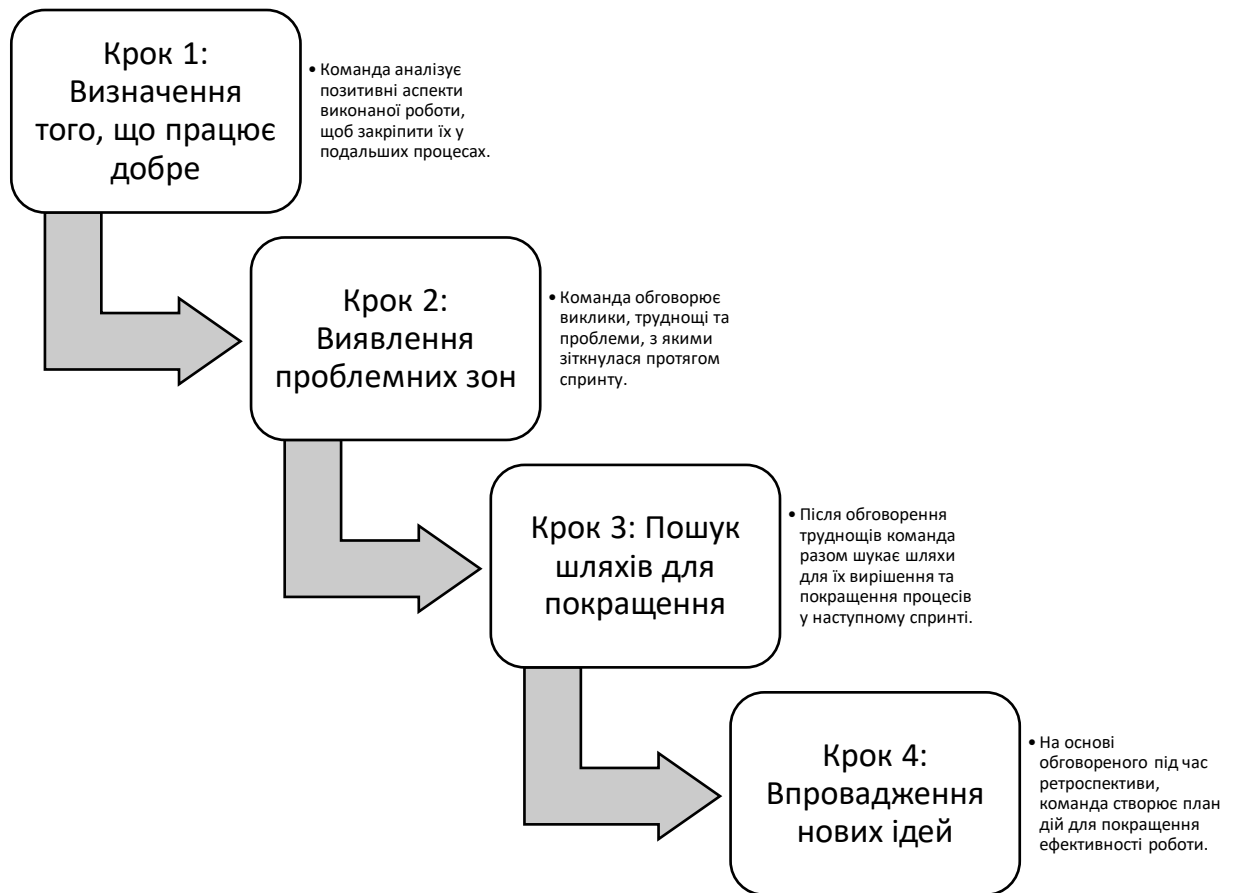


Рисунок 5.3 – Візуалізація процесу ретроспективи

Джерело: розроблено автором

Далі розглянемо кроки детальніше.

Крок 1: Визначення того, що працює добре під час ретроспективи — це етап, коли команда аналізує та виділяє аспекти роботи, які сприяли успіху спринту або загальному покращенню продуктивності. Це дозволяє команді зосередитися на позитивних моментах і закріпити їх у майбутніх робочих процесах.

Основні питання для аналізу цього етапу можуть включати:

- Що допомогло нам досягти цілей спринту? Це може стосуватися успішних інструментів, методик або підходів, які команда використовувала для вирішення задач.

- Які комунікаційні практики працювали ефективно? Це можуть бути регулярні зустрічі, швидкі рішення проблем або ефективний розподіл ролей між членами команди.

- Що сприяло продуктивності команди? Наприклад, злагоджена робота, взаємна підтримка, якісний код або вдала взаємодія з замовником.

Завдяки визначенню того, що працює добре, команда може закріпити позитивні моменти у своїй роботі та зробити їх основою для майбутніх спринтів, зберігаючи ефективність та покращуючи процеси в майбутньому.

Крок 2: Виявлення проблемних зон під час ретроспективи — це етап, коли команда аналізує аспекти роботи, що потребують покращення. Мета цього процесу — ідентифікувати та зрозуміти ті моменти, які сповільнювали роботу команди, заважали досягненню цілей спринту або призводили до втрати ефективності. Виявлення проблемних зон дозволяє команді визначити основні перешкоди і розробити стратегії для їх усунення в наступних спринтах.

Основні питання для аналізу проблемних зон можуть включати:

- Що затримало або ускладнило виконання завдань? Це можуть бути технічні проблеми, недоліки планування або нерівномірне розподілення навантаження.

- Які комунікаційні бар'єри виникли? Команда аналізує, чи були проблеми у взаємодії між членами команди або зі стейкхолдерами.

- Які інструменти або методики не виправдали очікувань? Це можуть бути процеси або технології, які виявилися менш ефективними, ніж очікувалося.

Це критичний крок для постійного вдосконалення, оскільки виявлені проблеми дозволяють впроваджувати конкретні дії для їх вирішення, таким

чином підвищуючи загальну продуктивність та якість роботи команди в майбутніх спринтах.

Крок 3: Пошук шляхів для покращення — це ключовий крок у ретроспективі, під час якого команда активно розробляє конкретні пропозиції щодо вдосконалення процесів або підходів. Після того, як ідентифіковані проблемні зони, команда обговорює, як ці труднощі можна вирішити та уникнути в майбутньому. Основною метою цього процесу є знайти оптимальні шляхи для підвищення продуктивності, покращення комунікації та загальної ефективності роботи.

Цей крок можна розділити на наступні етапи:

- **Генерація ідей:** Команда мозковим штурмом генерує ідеї щодо покращення робочих процесів. Це можуть бути нові підходи до організації роботи, використання нових інструментів або перегляд ролей і відповідальностей.
- **Аналіз пропозицій:** Всі запропоновані ідеї оцінюються з точки зору їх реалізації, ефективності та ресурсоємності. Команда визначає, які рішення будуть найбільш доцільними і принесуть найвищу цінність.
- **Визначення конкретних дій:** Формулюються чіткі кроки для впровадження змін у наступних спринтах. Це можуть бути зміни в процесах планування, комунікації або використання технологічних інструментів.
- **Призначення відповідальних:** Для того щоб зміни стали реальністю, призначаються конкретні особи, відповідальні за реалізацію кожної дії. Це допомагає впевнитися, що рішення не залишаться тільки на рівні обговорення, а будуть впроваджені на практиці.
- **Відстеження прогресу:** Команда вирішує, як буде вимірювати успіх впроваджених покращень і на якому етапі буде проведено повторний аналіз ефективності нових підходів.

Пошук шляхів покращення сприяє безперервному вдосконаленню командної роботи та якості продукту, допомагаючи уникати повторних помилок і підвищувати ефективність у майбутніх ітераціях.

Крок 4: Впровадження нових ідей є заключним етапом процесу ретроспективи, під час якого команда переходить від обговорення до практичної реалізації рішень. Це важливий етап, який дозволяє командам безпосередньо покращувати свою роботу на основі виявлених проблем та пропозицій. Процес впровадження нових ідей включає наступні необхідні дії:

- Створення плану дій: Після обговорення пропозицій команда визначає, які з нових ідей будуть реалізовані. Це зазвичай включає чіткі кроки, терміни, відповідальних осіб та ресурси, необхідні для втілення.
- Пріоритизація завдань: Команда оцінює ідеї та визначає, які зміни матимуть найбільший вплив або можуть бути реалізовані найбільш швидко. Це допомагає сфокусуватися на тих нововведеннях, що принесуть максимальну користь за короткий час.
- Експериментальне впровадження: На цьому етапі нові підходи та інструменти вводяться у практику. Часто це починається з тестування на невеликому масштабі або в конкретній частині проєкту, щоб оцінити ефективність і зібрати відгуки.
- Моніторинг і адаптація: Команда уважно стежить за результатами впроваджених змін. Важливо регулярно збирати відгуки та аналізувати ефективність нових підходів. Якщо щось не працює так, як очікувалося, зміни можуть бути адаптовані або скасовані.
- Закріплення практик: Якщо нові ідеї показали себе успішно, вони стають частиною регулярного процесу. Це може включати перегляд робочих інструкцій, оновлення документації або навчання команди.

Цей підхід дозволяє командам безперервно вдосконалювати свою роботу, надаючи змогу адаптуватися до нових умов і вирішувати проблеми ще на ранніх стадіях, що підвищує ефективність і гнучкість процесів.

Отже необхідно відзначити важливість системного підходу до оцінки робочих процесів. Кожен етап, починаючи з визначення позитивних аспектів і закінчуючи впровадженням нових ідей, сприяє безперервному вдосконаленню команди та проєкту. Визначення того, що працює добре, допомагає закріпити

успішні практики, а виявлення проблемних зон дає змогу ідентифікувати та вирішити перешкоди, які знижують ефективність. Пошук шляхів покращення та впровадження нових ідей дозволяють команді адаптуватися до змін, підтримуючи гнучкість та ефективність у довгостроковій перспективі.

3.3 Сучасний інструментарій менеджера в agile-середовищі

Команду в гнучкому менеджменті (agile команду) можна розглядати як соціальну групу, введемо поняття соціальної групи.

Соціальна група – це сукупність двох або більше осіб, що знаходяться в стані взаємодіють між собою. Стан соціальної взаємодії стосується взаємного впливу, який особи здійснюють під час взаємодії між собою [18].

Соціальна група за визначенням Роберта Мертона це сукупність двох або більше людей, які певним чином взаємодіють один з одним, усвідомлюють свою приналежність до цієї групи і вважаються членами цієї групи з погляду інших [19].

Соціальна група — сукупність осіб, об'єднаних спільною метою, ідеєю, працею [20].

Соціальні групи — це стійкі групи людей, яких поєднують одні й ті самі ознаки. Наприклад, інтереси, становище у суспільстві, цінності тощо [21].

Зробимо опис команди як соціальної групи:

Рекомендована кількість людей в команді від 3 до 9 людей, але в реальній практиці складається в середньому з 6-10 осіб.

Структура команди та функції:

Scrum master або team lead – дана особа відповідальна за координацію та ефективність роботи команди [22].

Product owner – відображає потреби клієнтів [22].

Team member – в agile команді це спеціалісти, залучені до розробки продукту [22].

Stakeholder – дані особи не приймають участі в проєктних завданнях, але грають важливу роль у визначені кінцевих результатів [22].

Agile команда організовуються для спільної роботи над створенням необхідного продукту, та отримання прибутку. Після виконання усіх робіт команда може бути розпущена або переформатована.

Мала соціальна група – невелика за чисельністю спільність, в якій індивіди безпосередньо контактують між собою, об'єднані спільною метою та завданнями, що є передумовою їх взаємодії, взаємовпливу, спільних норм, процесів та інтересів, міжособистісних відносин і тривалості їх існування [23].

Мала група - це контактна спільнота, яка взаємодіє на основі смислу спільності, має певний ступінь згуртованості, організованості й соціальної зрілості й володіє неповторною соціальною психікою [24].

Малої групою зазвичай називають невелике число людей, які добре знають один одного і постійно взаємодіють між собою [25].

Як описано вище agile команда має чітку структуру та лімітовану кількість членів. Взаємодія в команді заснована на принципах співпраці та комунікації між членами групи. Деякі члени команди можуть приймати участь в крос-тренінгах та працювати разом з колегами для вивчення та розвитку нових навичок [22].

Спираючись на опис вище agile команду можна вважати малою соціальною групою. Також agile команду можна розглядати як самоорганізуючу групу, так як самоорганізація в команді є одним зі стовпів методології Agile.

Самоорганізуюча група – це команда, яка має автономію вибирати, як найкраще виконувати свою роботу, замість того, щоб її керували інші люди поза командою [26].

Agile команда сама вирішує як краще виконати їх роботи, замість того щоб очікувати на вказівки від іншої людини поза командою. Scrum master в такій команді слугує команді розробників в декількох напрямках, в таких як: навчання розробників в самоорганізації та крос-функціональності [27].

Команда у гнучкому менеджменті це крос-функціональна команда у якій кожен член в agile команді має свій набір навичок, але всі члени команди працюють для досягнення спільної мети [22].

Тому agile команду можна вважати групою що самоорганізується. За параметр групи взято самоорганізацію.

Крос-функціональна група – це група людей з різних департаментів в організації, які працюють для досягнення спільної цілі. Часто крос-функціональні групи організуються для виконання проєкту, але їх також можна створити з більш простою метою [28].

Крос-функціональна команда – це група людей з різними функціональними знаннями, які працюють разом для досягнення спільної мети [29].

В agile середовищі роль лідера команди виконує Scrum-майстер, також лідер в даному середовищі виступає наставником, його обов'язками є навчання, психологічна допомога, моніторинг команди та направлення команди розробників для ефективного виконання поставлених задач.

Найбільше для agile середовища підходять такі стилі управління:

- Демократичний стиль - стиль управління характеризується тим, що різні повноваження, ініціативи і навіть відповідальність розподіляються між керівником і його підлеглими. Демократичний стиль управління передбачає, що в ході постійного спілкування і обговорення відбувається своєчасне інформування з усіх основних питань, що так чи інакше стосуються функціонування компанії [15].

- Ліберальний стиль являє собою щось середнє між демократичним і адміністративно правовим стилями, де проповідуються ліберальні норми управління особистістю і колективом. Ліберальний керівник характеризується мінімальною участю керівника, підлеглим дається майже повна воля у визначенні їх цілей і контролю за своєю власною роботою [16].

- Наставницький стиль в якому лідери допомагають підлеглим побачити свої слабкі й сильні сторони й адекватно оцінити свої перспективи -

особисті й кар'єрні. Вони заохочують підлеглих ставити собі довгострокову мету й допомагають їм досягати її. Лідери володіють чудовою здатністю делегувати повноваження, давати своїм підлеглим цікаві, сприятливі для їхнього зростання завдання, навіть якщо знають, що в результаті робота буде виконана повільніше, ніж звичайно [17].

Для виконання своїх обов'язків scum-майстер має володіти широким інструментарієм для управління та налагодження взаємодії в команді. Одним з таких інструментів є практика трьох С's, яка складається з 9 принципів приведених на Рисунок 3.1.



Communication (комунікація)

- **Developing** – лідер працює над собою перш ніж над іншими.
- **Reflecting** – лідер на пряму бере участь у вирішенні проблеми.
- **Learning** – лідер постійно навчається та виправляє свої помилки

Commitment (зобов'язання)

- **Inspiring** – лідер розробляє та поширює спільне розуміння та мету
- **Engaging** – лідер надихає співробітників для досягнення найкращого результату
- **Unifying** – лідер реалізує потенціал своїх підлеглих



Collaboration (співпраця)

- **Empowering** – лідер слідкує за можливостями співробітників та надає нові повноваження
- **Achieving** – лідер будує спільноту на основі довіри, поваги та робочої культури
- **Innovating** – лідер має бути відкритим до ідей усіх співробітників

Рисунок 3.4 – Візуалізація принципу 3С

Джерело: розроблено автором

Крім цих практик Пітер Конінг в своїй книжці «Agile Leadership Toolkit» описує 4 необхідні лідеру навички:

- Co-create – це здатність створення спільно з командою створювати бачення та напрямок, в якому необхідно прямувати. Разом з agile командою, лідер створює фокус на наданні цінностей для клієнтів та компанії [30];
- Facilitate – це здатність сприянню владі не шляхом її примусу, а через полегшення процесу та постійного вдосконалення влади [30];
- Experiment – це здатність agile лідера до створення безпечного середовища, що дозволить та дасть місце команді для експериментів [30];
- Lead the culture – це здатність лідера до створення «здорової» культури і вести людей за собою не за допомогою вказівок що робити, а за допомогою створеної культури [30].

Одна з головних задач scrum-майстра – спілкування з колективом, з метою навчання, підтримки, покращення колективної атмосфери та знайомства з кожним членом команди.

Для виконання даної задачі scrum-майстер використовує ряд методів комунікації. Методи комунікації описані нижче.

Осмотичне спілкування – це метод спілкування коли члени команди знаходяться поряд один з одним, і вони піддаються широкому спектру явищ, звуків, дебатів, дискусій та розмов. У фоновому режимі прослуховування, члени команди, ймовірно будуть отримувати інформації з розмов навіть якщо вони не приділяють увагу до них [31].

Активне слухання представляє фізичне та ментальне зусилля щоб обережно зрозуміти ціле повідомлення, правильно інтерпретувати його значення через вербальну комунікацію, тон голосу та мову тіла, та показати спікеру, через зворотній зв'язок, те що ми зрозуміли. За допомогою активного слухання лідер розпізнає проблеми, цілі та фактори що мотивують членів команди. Це дозволяє розвивати робочі та персональні стосунки з співробітниками, друзями сім'єю та іншими [32].

Двостороння комунікація – це інтерактивна форма комунікації в якій дві або більше людей обмінюються між собою думками та почуттями. Вирішення проблем та прийняття рішень це приклади двосторонньої комунікації. На відміну від односторонньої комунікації, в двосторонній комунікації увага приділяється до всіх точок зору через що даний тип комунікації являється більш точним та ефективним під час прийняття важливих рішень [33].

Під час спілкування з членами команди, скрам-майстер може допомогти у прийнятті важливих рішень або вирішенні проблем і непорозумінь між членами команди враховуючи деталі та коментарі отримані від усіх зацікавлених сторін з приводу ситуації.

Зворотній зв'язок – це відповідь, реакція або інформація яка надається отримувачем повідомлення відправнику. Це процес надання інформації про які області необхідно покращити. За допомогою зворотного зв'язку, скрам-майстер допомагає зрозуміти допущені помилки, додає нові ідеї та відповідає на питання що виникають у членів команди [34].

Окрім описаних методів комунікації, скрам-майстер повинен використовувати різні технології для комунікації з командою. Такими технологіями можуть бути:

Мітинги (зустрічі) з контактним видом спілкування – це процес коли члени команди, знаходячись поряд один з одним, обговорюють різні питання щодо поточних завдань з метою отримання інформації та відгуків, вирішення проблем, покращення процесів у команді. В методології Scrum існує декілька видів мітингів які представлені на Рисунок 3.2.

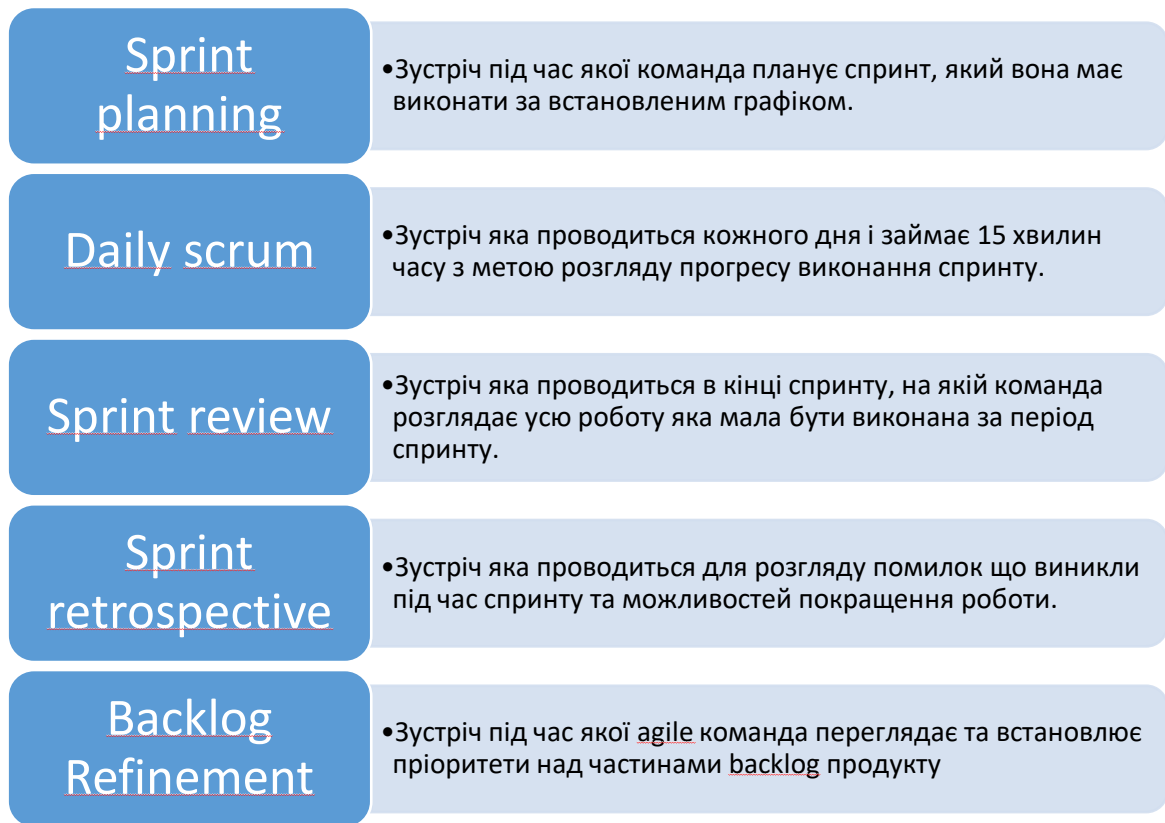


Рисунок 3.5 – Види мітингів у гнучкому менеджменті

Джерело: розроблено автором

Месенджери – це програми для обміну повідомленнями, вони дозволяють швидко та легко спілкуватися між членами команди без необхідності писати великі електронні листи чи телефонних дзвінків [35].

За допомогою месенджерів скрам-майстер може швидко зв'язатись з необхідним йому членом команди та розпочати з ним комунікацію без фізичного спілкування, а також швидко доносити важливу інформацію до усіх членів команди. Також месенджери можна використовувати для інформування скрам-майстра про непередбачувані проблеми зі сторони членів команди, як от вихід на лікарняний.

Інструменти відео зв'язку – дозволяє командам спілкуватися віч-на-віч незалежно від того, де вони знаходяться. Це допомагає зміцнити довіру та товариськуність серед членів команди, які можуть не мати змоги регулярно зустрічатися особисті [35].

Whiteboards – даний інструмент представляє з себе платформу яка надає віртуальний простір для мозкового штурму ідей і візуальної організації інформації. Також цей інструмент дозволяє кільком членам команди працювати над одним проєктом одночасно, підвищуючи ефективність і співпрацю [35].

Порівняння видів комунікації приведено в Таблиці 3.3.

Таблиця 3.3 – Порівняння видів комунікації

Назва інструменту	Зоровий контакт	Швидкість вирішення питань	Зручність використання	Ситуаційна ефективність інструменту
Мітинги з контактним видом спілкування	так	Висока	Висока (необхідна фізична присутність співробітника)	Ефективний для будь-яких процесів
Месенджери	ні	Низька	Змішана (Залежить від ситуації використання)	Ефективний для обміну необхідною інформацією
Інструменти відео зв'язку	так	Висока	Висока (Високий вплив технічних проблем)	Ефективний для проведення зустрічі при неможливості відвідати офлайн мітинг
Whiteboard	ні	Низька	Висока (Необхідно навчити членів команди корисуванию)	Ефективний для організації роботи команди, розподілення завдань та ідей

Agile-команда може розглядатися як мала соціальна група, де взаємодія між її учасниками заснована на принципах співпраці та комунікації. Команда самоорганізується і крос-функціональна, де кожен член має свій набір навичок, але працює задля спільної мети. Важливими складовими успішної роботи є спілкування, активне слухання, взаємодія через різні канали (наприклад, мітинги, месенджери), а також підтримка здорової робочої атмосфери. Scrum-майстер виконує роль лідера, координуючи команду та забезпечуючи її ефективність.

Висновки по розділу 3

Розробка інкременту системи SmartAgri є ключовим етапом у реалізації Agile-підходу, який забезпечує поступове вдосконалення та адаптацію системи до потреб користувачів. Інкременти, що створюються наприкінці кожного спринту, складаються з функціонально готових частин системи, які можуть бути протестовані та впроваджені, що дозволяє надавати користувачам додану вартість на кожному етапі.

Інкрементний підхід до розробки SmartAgri не лише покращує управління аграрними процесами, але й забезпечує гнучкість та адаптивність системи до нових викликів. Поступове розширення функціональності через інфраструктурні, аналітичні та інтерфейсні інкременти створює цінність для користувачів на кожному етапі, що в кінцевому результаті сприяє підвищенню ефективності фермерського господарства.

Ретроспектива роботи команди є критично важливим етапом у процесі гнучкого менеджменту, оскільки вона надає можливість команді систематично оцінювати свою діяльність, аналізувати результати спринтів і виявляти шляхи для постійного вдосконалення. Під час ретроспективи команда не тільки оцінює, що працювало успішно, але й ідентифікує проблеми, які потребують вирішення. Це дозволяє їй адаптуватися до нових викликів і підвищувати загальну продуктивність.

Системний підхід до ретроспективи не лише покращує ефективність командної роботи, але й сприяє створенню культури безперервного вдосконалення. Завдяки цьому, команди можуть швидко реагувати на зміни в умовах роботи, що особливо важливо у динамічному середовищі сучасного програмування та управління проектами. Ретроспективи стають основою для формування сталого розвитку команди і підвищення якості продукту, оскільки дозволяють уникати повторення помилок і запроваджувати найкращі практики у щоденній діяльності.

Команда в agile-середовищі, за визначенням, має чітко визначену структуру, що включає такі ролі, як Scrum Master, Product Owner та члени

команди. Scrum Master виконує функцію координації та навчання, підтримуючи культуру співпраці і комунікації. Різноманітні стилі управління, такі як демократичний, ліберальний та наставницький, сприяють створенню атмосфери, де члени команди можуть вільно ділитися своїми ідеями та вносити пропозиції.

Для забезпечення ефективної комунікації команда використовує різні методи, зокрема осмотичне спілкування, активне слухання та двосторонню комунікацію. Ці методи дозволяють створити продуктивний діалог і сприяти вирішенню проблем. Використання сучасних технологій, таких як відео зв'язок, месенджери та віртуальні дошки, дозволяє покращити комунікаційні процеси, особливо в умовах віддаленої роботи.

ВИСНОВОК

Система SmartAgri являє собою інноваційне рішення для управління сільськогосподарськими процесами, що охоплює широкий спектр функцій — від моніторингу посівів до аналітики врожайності. Використання Agile-підходу у розробці системи забезпечує гнучкість, адаптивність і тісну співпрацю з користувачами, що дозволяє швидко реагувати на зміни в умовах ринку та вимогах споживачів.

Завдяки чітко визначеним цілям, таким як розробка інтуїтивно зрозумілого інтерфейсу, оптимізація управління ресурсами та інтеграція нових технологій, проєкт прагне підвищити продуктивність і знизити витрати в аграрному секторі. Бізнес-вимоги, включаючи ефективну аналітику даних і інтеграцію з існуючими системами, сприяють досягненню стратегічних цілей і підтримці стійкого зростання на ринку.

Реалізація SmartAgri має велике стратегічне значення, оскільки вона вирішує актуальні проблеми, такі як неефективне управління ресурсами і відсутність єдиної платформи для моніторингу. Проєкт складається з восьми основних етапів, серед яких критично важливим є розробка мінімального життєздатного продукту (MVP) через фреймворк Scrum. Чітке планування та систематизація процесів, таких як Scrum-церемонії, гарантують ефективну комунікацію та моніторинг прогресу команди.

Успішна реалізація проєкту вимагає ретельного управління вимогами та активного виявлення можливих ризиків, таких як технічні складнощі та питання безпеки. Систематичний підхід до ретроспективи дозволяє команді оцінювати свою діяльність, ідентифікувати проблеми і запроваджувати стратегії для їх вирішення, що підвищує загальну продуктивність.

Таким чином, система SmartAgri має потенціал суттєво підвищити ефективність аграрного сектора, сприяючи покращенню управління агропроцесами, підвищенню конкурентоспроможності та забезпеченню сталого розвитку в умовах сучасних викликів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алькема В. Г., Кириченко О. С. Менеджмент організацій: навчальний посібник. Кн.1. Київ: Університет «КРОК», 2023. 276 с. Електронний ресурс КРОК. URL: <https://library.krok.edu.ua/ua/kategoriji/navchalni-posibniki/1440-menedzhment-orhanizatsii>
2. Горковець, Д. О. Балдик. Управління ресурсами в ІТ проєктах. Держава, регіони, підприємництво: інформаційні, суспільно-правові, соціально-економічні аспекти розвитку: тези доповідей V Міжнародної конференції (Київ, 7 грудня 2023 р.). Київ: Університет "КРОК", 2023. Електронний ресурс КРОК. URL: <https://conf.krok.edu.ua/SRE/SRE-2023/paper/view/1635>
3. Данченко О.Б. Практичні аспекти реінжинірингу бізнес-процесів / О.Б. Данченко. Київ. Університет економіки та права «КРОК», 2017. – 238 с. Електронний ресурс КРОК. URL: <https://library.krok.edu.ua/ua/kategoriji/navchalni-posibniki/529-praktychni-aspekty-reinzhyrynihu-biznesprotsesiv>
4. Кириченко О. С. Інвестиційно-інноваційне забезпечення промисловості України в умовах Четвертої промислової революції: монографія. Київ. Університет економіки та права «КРОК», 2021. 368 с. - Електронний ресурс КРОК. URL: <https://library.krok.edu.ua/ua/kategoriji/monografiji/1351-investytsiinoinnovatsiine-zabezpechennia-promyslovosti-ukrainy-v-umovakh-chetvertoi-promyslovoi-revoliutsii>
5. Мічківський С. М. Системи та методи прийняття рішень: методичні вказівки / С. М. Мічківський, Р. Ю. Подольський, Т.К. Талапов. - Старобільськ: ЛНАУ, 2020.- 80 с. Електронний ресурс. URL: <http://dspace.lgnau.edu.ua/xmlui/handle/123456789/1456>

6. Стаття «Smart Agriculture Market | Driven by increasing food demand and adoption of technology in agriculture», URL: <https://huecanada.ca/smart-agriculture-market-driven-by-increasing-food-demand-and-adoption-of-technology-in-agriculture/>
7. Стаття «Farm Management Software Market Size, Share & Trends Analysis Report By Agriculture Type», URL: <https://www.grandviewresearch.com/industry-analysis/farm-management-software-market>
8. «What is scrum and how to get started», URL: <https://www.atlassian.com/agile/scrum>
9. «The 2020 Scrum Guide», URL: <https://scrumguides.org/scrum-guide.html#scrum-values>
10. Стаття: «Що таке управління вимогами: визначення, переваги, найкращі інструменти», URL: <https://visuresolutions.com/uk/>
11. Стаття «Burndown chart» на сайті Вікіпедія, URL: https://en.wikipedia.org/wiki/Burndown_chart#CITEREFProject_Management_Institute2021
12. Velocity Chart, URL: <https://www.agile-scrum.be/whats-great-scrum-methodology/velocity-chart/>
13. Стаття «Scrum термінологія.», URL: <https://training.gatestlab.com/blog/technical-articles/scrum-terminology/>
14. Стаття «3 agile-методи, які повинен знати кожен менеджер (частина 3): Ретроспектива», URL: <https://echometerapp.com/uk/%D1%80%D0%B5%D1%82%D1%80%D0%BE%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D0%B8%D0%B2%D0%B0/>
15. «Демократичний стиль управління та його особливості», URL: <https://vcf.vn.ua/demokratichnij-stil-upravlinnya-ta-jogo-osoblivosti/>
16. Мельник І. І. «Інженерний менеджмент.» / Мельник І. І., Тивоненко І. Г., Фришев С. Г., Бабій В. П., Бондар С. М – Нова книга, 2007 р. – 536 с.

17. «Зразковий стиль», URL: https://pidru4niki.com/16150816/menedzhment/zrazkoviy_stil
18. «Social group», URL: <https://www.studyandexam.com/social-group-types.html>
19. «Соціальні групи», URL: https://pidru4niki.com/18340719/sotsiologiya/sotsialni_grupi
20. «Соціальні група», URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%BE%D1%86%D1%96%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%B3%D1%80%D1%83%D0%BF%D0%B0
21. «Соціальні групи», URL: <https://tsikavi-fakty.com.ua/sotsialni-grupy/>
22. «Buildng an Agile Team Structure», URL: <https://www.wrike.com/agile-guide/agile-team-structures/>
23. «Мала група у контексті соціальної психології», URL: <https://studentam.net.ua/content/view/3188/97/>
24. «Малі соціальні групи», URL: https://pidru4niki.com/12210605/psihologiya/mali_sotsialni_grupi
25. «Мала група», URL: https://stud.com.ua/74344/sotsiologiya/mala_grupa
26. «What is a Self-Organizing Team in Scrum?» з веб-сайту «Visual paradigm», URL: <https://www.visual-paradigm.com/scrum/what-is-self-organizing-team-in-scrum/>
27. «About self-organizing teams», URL: <https://www.scrum.org/resources/blog/about-self-organizing-teams>
28. «What Are Cross-Functional Teams? Everything You Need To Know», URL: <https://www.forbes.com/advisor/business/cross-functional-teams/>
29. Стаття «What Are Crossed Functional Teams in Agile? Why Do They Matter?», URL: <https://kanbanize.com/blog/cross-functional-teams/>

30. Peter Koning «Agile Leadership Toolkit» / Peter Koning – Addison-Wesley, 2019 p. – 240 c.
31. Sumanta Boral «Ace the PMI-ACP® Exam» – Apress 2016 p. – 441c.
32. Rodrigo Ortiz Crespo «The Active Listener» – Lulu Press, Incorporated 2010 p. – 106 c.
33. Debra L. Nelson, James Campbell Quick «ORGB4» – Cengage Learning 2014 p. – 368 c.
34. Leanne E. Atwater, David Andrew Waldman «Leadership, Feedback, and the Open Communication Gap» / Leanne E. Atwater, David Andrew Waldman – Lawrence Erlbaum Associates 2008p. – 164 c.
35. «Powerful Tools for a Scrum Master», URL: <https://www.linkedin.com/pulse/powerful-tools-scrum-master-sunny-uday-kiran/>