

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

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Гнучке управління створенням системи обробки замовлень

«AssetFlow» для фінансових ринків»

Ступінь вищої освіти – магістр

Спеціальність – 073 «Менеджмент»

Освітня програма «Agile-технології розробки програмного забезпечення»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Керівники: доцент кафедри комп'ютерних наук,
к.т.н.

Олександр ПОЛІЩУК

викладач кафедри інформаційного
менеджменту, математики та
статистики

Олег МУШИНСЬКИЙ

Виконав: здобувач

групи МЕН/Agile-24м

Сергій ТИХОНЕНКО

Засвідчую, що кваліфікаційна
робота оформлена відповідно до
ДСТУ 3008:2015 та не містить
запозичень з праць інших авторів
без відповідних посилань.

Здобувач: _____

(підпис)

Київ, 2026 р.

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

ЗАТВЕРДЖУЮ:

завідувач кафедри інформаційного
менеджменту, математики та статистики

_____ Денис БАЛДИК

«__» ____ 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Тихоненко Сергій Сергійович**

Тема роботи	Гнучке управління створенням системи обробки замовлень «AssetFlow» для фінансових ринків
Номер та дата наказу про затвердження теми	№ 109-2 від 14 жовтня 2025 року р.
Коротка постановка завдання	Обґрунтування бачення створюваного продукту для розв'язання проблеми в діяльності замовника на основі розробки моделі його організації. Детальний опис особливостей гнучкого управління розробкою системи обробки замовлень «AssetFlow». Розкриття для гнучкого управління створенням системи обробки замовлень «AssetFlow» для фінансових ринків
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	<ol style="list-style-type: none"> 1. Балдик Д., Мушинський О, Спічак Р. Застосування NLP моделей в управлінні бізнес-процесами електронної комерції будівельних матеріалів: моделювання та статистичний аналіз. Вчені записки Університету "КРОК". 2025. № 2(78). С. 297-305. DOI https://doi.org/10.31732/2663-2209-2025-78-297-305 2. Москвяк Я. Фінансові ринки як особливе середовище для ухвалення фінансових рішень. «Наука і техніка сьогодні. СЕРІЯ «Економіка». № 13(41) 2024 DOI: https://doi.org/10.52058/2786-6025-2024-13(41)-294-304 3. O. Runsewe, O. S. Osundare, S. Olaoluwa, and L. A. A. Folorunsho, “End-to-end systems development in agile environments: Best practices and case studies from the financial sector,” International Journal of Engineering Research and Development, vol. 20, no. 08, pp. 522–529, 2024
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має містити теоретичне та/або практичне дослідження за темою роботи, яку слід розглядати як складне спеціалізоване завдання або практичну проблематику в галузі управління та адміністрування, яка характеризується комплексністю та невизначеністю умов і потребує застосування Agile-технологій.

Дата видачі завдання «16» жовтня 2025 р.

Керівник

Олександр ПОЛЩУК

Керівник

Олег МУШИНСЬКИЙ

Здобувач

Сергій ТИХОНЕНКО

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напрямку дослідження та керівника.	01.09.2025 р.	виконано
2	Формування теми та призначення керівника.	22.09.2025 р.	виконано
3	Затвердження теми кваліфікаційної роботи.	14.10.2025 р.	виконано
4	Затвердження завдання на кваліфікаційну роботу.	16.10.2025 р.	виконано
Основний етап			
5	Розробка концепції та змісту кваліфікаційної роботи, погодження їх з науковим керівником	06.11.2025 р.	виконано
6	Підбір та вивчення джерел інформації з напрямку дослідження.	08.11.2025 р.	виконано
7	Теоретико-методичний аналіз предметної області. Підготовка та подання керівнику розділу 1 кваліфікаційної роботи.	13.11.2025 р.	виконано
8	Реалізація гнучкого управління розробкою продукту. Підготовка та подання керівнику розділу 2 кваліфікаційної роботи.	20.11.2025 р.	виконано
9	Розробка рекомендацій щодо вдосконалення управління із застосуванням Agile-технологій. Підготовка та подання керівнику розділу 3 кваліфікаційної роботи.	27.11.2025 р.	виконано
10	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи.	01.12.2025 р.	виконано
11	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівнику доопрацьованого варіанту кваліфікаційної роботи	03.12.2025 р.	виконано
Завершальний етап			
12	Представлення рукопису для перевірки на плагіат.	08.12.2025 р.	виконано
13	Підготовка презентації та доповіді на передзахист.	22.12.2025 р.	виконано
14	Передзахист кваліфікаційної роботи.	23-24.12.2025 р.	виконано
15	Технічна самоекспертиза роботи на відповідність вимогам до оформлення та виправлення недоліків.	12-16.01.2026 р.	виконано
16	Експертиза роботи керівником та зовнішнім експертом (рецензентом).	20.01.2026 р.	виконано
17	Доопрацювання доповіді та презентації для захисту.	22.01.2026 р.	виконано
18	Захист кваліфікаційної роботи.	26-30.01.2026 р.	виконано

Керівник

Керівник

Здобувач

Олександр ПОЛЩУК

Олег МУШИНСЬКИЙ

Сергій ТИХОНЕНКО

Тихоненко С.С. Гнучке управління створенням системи обробки замовлень «AssetFlow» для фінансових ринків

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, 2026 р.

В кваліфікаційній роботі досліджено особливості гнучкого управління розробкою системи обробки замовлень «AssetFlow». Розкрито гнучке управління створенням системи обробки замовлень «AssetFlow» для фінансових ринків. Проаналізовано діяльність фінансових ринків, здійснено планування та оцінювання інкрементів, проведено ретроспективний аналіз роботи команди на основі якого запропоновано модель оцінювання задач в беклозі.

Ключові слова: Agile, Scrum, управління проектами, фінансові ринки.

Табл. 22. Рис. 23. Бібліограф.: 16 найм.

Tikhonenko S.S. Agile management of developing the AssetFlow order processing system for financial markets

Qualification paper 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, 2026.

The thesis examines the features of flexible management of the development of the AssetFlow order processing system. It reveals flexible management of the creation of the AssetFlow order processing system for financial markets. The analysis of financial markets' activities, plans and evaluates increments, and conducts a retrospective analysis of the team's work, based on which a model for evaluating tasks in the backlog is proposed.

Keywords: Agile, Scrum, project management, financial markets.

Tabl. 22. Fig. 23. Bibliography: 16 items.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ДИЗАЙН БІЗНЕСУ ASSETFLOW	10
1.1 Аналіз стану та перспектив розвитку фондового ринку України і обґрунтування потреби у створенні системи AssetFlow	10
1.1.2. Нормативно-правове та інституційне середовище	12
1.1.3. Тенденції та перспективи розвитку фондового ринку України	15
1.1.4. Аналіз бізнес-процесів та потреб учасників ринку	17
1.1.5. Визначення цільової аудиторії системи AssetFlow	22
1.1.6. Аналіз конкурентів та аналогів (систем управління заявками та торговельних платформ)	24
1.1.7. Бізнес-модель AssetFlow (Business Model Canvas)	27
1.2. Постановка цілей та завдань створення системи AssetFlow	28
1.2.1. Нова проблема та потреба компаній на фондовому ринку	28
1.2.2. Загальна мета створення системи AssetFlow.....	29
1.2.3. Завдання проєкту створення системи AssetFlow.....	29
1.2.4. Канва ціннісної пропозиції системи AssetFlow (Value Proposition Canvas).....	30
1.2.5. Дошка бачення продукту AssetFlow (Product Vision Board).....	31
1.3. Визначення вимог до продукту AssetFlow	33
1.3.1. Зацікавлені сторони та їхні інтереси	33
1.3.2. Бізнес-вимоги до системи AssetFlow	34
1.3.3. Основні сценарії використання (Use case) та взаємодія із зовнішнім середовищем	36
1.3.4. Користувацькі історії (User story) для ключових ролей	39
1.3.5. Функціональні вимоги до системи AssetFlow.....	40
1.3.6. Нефункціональні вимоги	41
1.3.7. Вхідні та вихідні дані системи.....	42
Висновки до розділу 1 «Дизайн бізнесу AssetFlow».....	43
РОЗДІЛ 2 ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ СИСТЕМИ ASSETFLOW	45
2.1. Обґрунтування вибору гнучкого підходу управління проєктом	45
2.2. Планування процесів розробки системи (дорожня карта, ролі, реєстр вимог).....	49
2.2.1. Дорожня карта розробки системи «AssetFlow».....	49
2.2.2. План виконання та терміни проєкту.....	52
2.2.3. Бюджет проєкту (фаза розробки MVP системи AssetFlow)	55
2.2.4. Команда проєкту та ролі в Scrum.....	59
2.2.5. Аналіз зацікавлених сторін (Stakeholder Mapping)	61
2.3 Моніторинг виконання проєкту.....	68
2.3.1 Сутність гнучкого моніторингу у проєкті «AssetFlow».....	69

2.3.2 Організація моніторингу на рівні спринтів	71
2.3.4 Burndown Chart як основний інструмент контролю спринтів	73
2.3.5 Velocity команди та моделювання наступних спринтів	74
2.3.6 Інтерпретація результатів моніторингу та управлінські рішення	76
Висновки до розділу 2	77
РОЗДІЛ 3. РЕЗУЛЬТАТИ РОБОТИ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КОМАНДИ.....	78
3.1. Огляд реалізованого інкременту продукту та ретроспектива роботи команди	78
3.1.1. Оцінка досягнення цілей спринтів та якості інкременту.....	84
3.1.3. Організація та результативність роботи Scrum-команди під час AF Sprint 3.....	86
3.1.4. Ретроспектива роботи Scrum-команди (формат «Парусник») після AF Sprint 3.....	88
3.2. Аналіз управлінської проблемної ситуації та її причин	91
3.2.1. Формулювання управлінської проблемної ситуації.....	91
3.2.2. Аналіз причин за допомогою діаграми Ішікави (рибний скелет)....	92
3.2.3. Узагальнення результатів аналізу причин.....	95
3.3.3. Модель оцінювання задач беклогу за критеріями цінності, спроможності, ризику та технічної складності	96
Висновки до розділу 3.....	99
ВИСНОВКИ.....	101
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	103
ДОДАТКИ	106

ВСТУП

Актуальність теми. Сучасні фінансові ринки вимагають високої швидкості та надійності обробки біржових операцій, прозорого відображення статусів замовлень, своєчасної звітності та дотримання регуляторних вимог. Для професійних учасників ринку капіталу критичною є наявність ефективних систем обробки замовлень (Order Management System, OMS), які забезпечують наскрізний супровід угоди – від моменту ініціювання клієнтом до розрахунків і формування звітності. В умовах війни, посилення вимог регулятора, інтеграції із законодавством ЄС та загального зміщення акцентів у бік цифрових каналів обслуговування зростає потреба у створенні таких систем, адаптованих до специфіки українського ринку.

Наявні на ринку рішення переважно орієнтовані або на великих міжнародних гравців із високою вартістю володіння, або на вузькоспеціалізовані задачі (торгові термінали, звітні модулі тощо) без цілісного охоплення життєвого циклу замовлення в умовах української нормативної бази. Крім того, в науковій та прикладній літературі недостатньо представлено цілісні підходи до створення OMS саме для українського ринку капіталу, які поєднували б продуктивний дизайн, архітектуру системи та гнучкі методи організації розробки. Це зумовлює необхідність розробки системи обробки замовлень «AssetFlow», що враховує особливості національного ринку, та опису підходу до її гнучкого створення на основі сучасних Agile-практик.

Мета та завдання дослідження. Метою кваліфікаційної роботи є обґрунтування концепції та організаційних засад створення системи обробки замовлень «AssetFlow» для фінансових ринків України із застосуванням гнучких підходів до управління її розробкою, що забезпечують адаптивність, прозорість і орієнтацію на потреби користувачів. Для досягнення поставленої мети передбачається вирішити такі завдання:

- проаналізувати особливості функціонування сучасних фінансових ринків та роль систем обробки замовлень у їх інфраструктурі;

- дослідити теоретичні засади застосування гнучких підходів до управління проектами у фінансовому секторі;
- сформуванати продуктове бачення та концептуальну модель системи «AssetFlow» з урахуванням потреб основних стейкхолдерів;
- розробити продуктові артефакти управління розробкою системи (реєстр вимог, беклог продукту, план-графік розвитку, опис мінімально життєздатної версії продукту);
- запропонувати підхід до організації життєвого циклу розробки системи «AssetFlow» з використанням гнучких методів управління та з урахуванням українських реалій;
- оцінити практичні можливості та очікувані результати впровадження запропонованих підходів при створенні та розвитку системи обробки замовлень.

Об'єктом дослідження є процес створення системи обробки замовлень «AssetFlow» для фінансових ринків України.

Предметом дослідження є організаційні, методичні та інструментальні засади застосування гнучких підходів до управління розробкою системи обробки замовлень «AssetFlow» у фінансовому секторі.

Методи дослідження. У роботі використано комплекс методів: системний підхід і структурно-функціональний аналіз (для опису місця OMS в інфраструктурі ринку капіталу), методи бізнес-моделювання (для формування продуктового бачення та вимог до системи), елементи моделювання процесів (для опису життєвого циклу замовлення та організації розробки), методи проектного менеджменту й Agile-підходів (для побудови підходу до гнучкого управління розробкою), а також елементи економічного аналізу (для оцінки очікуваних ефектів від впровадження системи).

Практична значущість роботи полягає у формуванні комплексної концепції системи обробки замовлень «AssetFlow» та описі підходу до її гнучкого створення, який поєднує продуктові артефакти, організацію роботи Scrum-команди та інструменти моніторингу виконання. Отримані результати можуть бути використані як методична основа для планування, організації та контролю розробки систем обробки замовлень у банківських, брокерських та фінтех-

проектах, а також як база для подальшої технічної реалізації та розширення «AssetFlow» з урахуванням вимог регулятора та очікувань користувачів.

Апробація результатів дослідження здійснювалася під час Міжнародної V Наукової конференції Університету «КРОК» «Сучасний менеджмент організації: витоки, реалії та перспективи розвитку» (м. Київ, 17 квітня 2025 року);

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі наведено аналіз ринку та бізнес-дизайн системи «AssetFlow». У другому розділі розглянуто організацію гнучкого управління створенням системи. У третьому розділі проаналізовано результати реалізації інкременту продукту, виявлено управлінські проблеми та сформовано рекомендації щодо підвищення ефективності роботи команди.

РОЗДІЛ 1

ДИЗАЙН БІЗНЕСУ ASSETFLOW

1.1 Аналіз стану та перспектив розвитку фондового ринку України і обґрунтування потреби у створенні системи AssetFlow

Фондовий ринок України є складовою ринку капіталу та виступає особливим середовищем для ухвалення фінансових рішень, де взаємодіють економічні показники, психологічні фактори та технологічний розвиток [1].

Сучасний етап його розвитку характеризується трансформацією інфраструктури, зростанням ролі електронних торгів та посиленням регуляторних вимог з боку Національної комісії з цінних паперів та фондового ринку (НКЦПФР) [2].

У структурі (рис 1.1) організованого ринку домінує сегмент торгівлі борговими інструментами, насамперед облігаціями внутрішньої державної позики (ОВДП), частка яких у загальному обороті на ринках капіталу перевищує 80 % у окремі періоди [3].

Серед фінансових інструментів на ринках капіталу, найбільший обсяг торгів у II кварталі 2024 році припадає на операції з облігаціями внутрішніх державних позик України – 88,43% (на суму 360,56 млрд грн).



На ринках капіталу у II кварталі 2024 року відбувалась торгівля наступними фінансовими інструментами, крім ОВДП, зокрема найбільші за обсягом з:

- облігаціями іноземної держави – 16,28 млрд грн
- корпоративними облігаціями – 12,91 млрд грн
- інвестиційними сертифікатами ПФ – 6,43 млрд грн
- свопами – 3,84 млрд грн
- векселями – 2,16 млрд грн

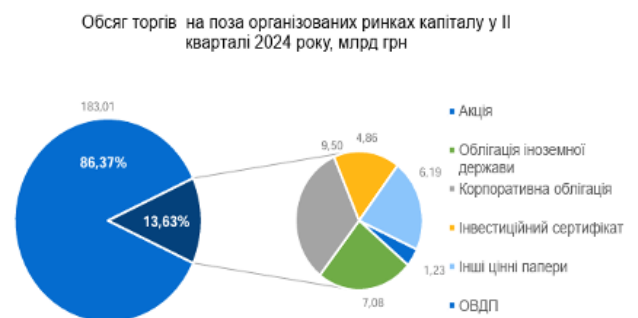


Рисунок. 1.1. - Структура торгів на ринках капіталу України (II кв. 2024)

Джерело: [3]

Ринки акцій та похідних інструментів розвинені істотно слабше, відзначаються низькою ліквідністю та обмеженим колом активних інструментів [4].

До ключових фінансових інструментів, що обертаються на фондовому ринку України, належать:

1. Акції. Український ринок акцій сформувався переважно внаслідок приватизації, що зумовило велику кількість емітентів із низькою біржовою активністю та обмеженим колом інвесторів [1]. Нині обіг зосереджений навколо небагатьох «якірних» емітентів (банки, енергетика, окремі промислові компанії), ринок загалом характеризується низькою ліквідністю, домінуванням інституційних інвесторів і недостатнім розвитком сервісів для роздрібних клієнтів [2], [4].

2. Облігації. Найбільш розвиненим сегментом є ринок облігацій внутрішньої державної позики (ОВДП), на який припадає основна частка біржових операцій; корпоративні та муніципальні облігації мають порівняно невеликі обсяги торгів [3]. Вони відіграють важливу роль у фінансуванні реального сектору й інфраструктурних проєктів, але стикаються з проблемами ліквідності та підвищеними кредитними ризиками, що стримує попит з боку широкого кола інвесторів [5].

3. Похідні фінансові інструменти (деривативи). Ринок деривативів перебуває на етапі формування: нормативна база вже дозволяє обіг стандартних біржових контрактів, однак фактичні обсяги торгів залишаються невеликими через брак ліквідних базових активів та обмежене використання хеджування [2]. За оцінками НКЦПФР, операції зі свопами, ф'ючерсами та іншими деривативами суттєво поступаються обсягам торгів ОВДП, хоча інтерес до цих інструментів поступово зростає [6].

Короткий огляд ключових параметрів фінансових інструментів, їхніх емітентів та ролі на ринку представлено в Таблиці 1.1 (Додаток А).

Місце фондового ринку у фінансовій системі України наведено на рис 1.2.

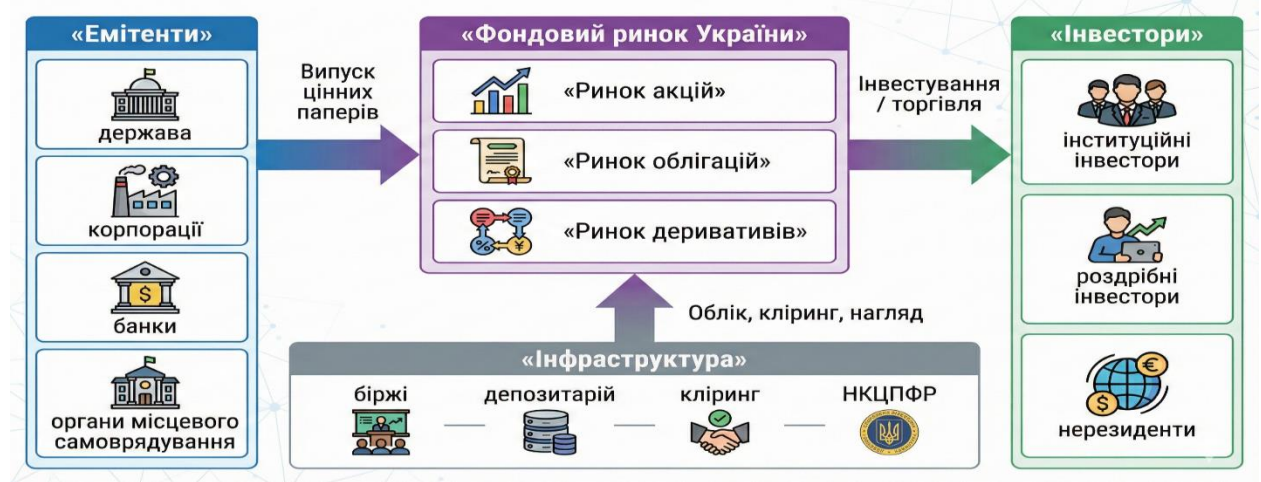


Рисунок 1.2 Місце фондового ринку у фінансовій системі України.

Джерело: розроблено автором

1.1.2. Нормативно-правове та інституційне середовище

Нормативно-правове та інституційне середовище ринку капіталу України формується системою спеціальних законів, підзаконних актів і стратегічних документів, що визначають правила випуску, обігу та обліку цінних паперів, а також повноваження органів нагляду й контролю. Ключовими суб'єктами державного регулювання є Національна комісія з цінних паперів та фондового ринку (НКЦПФР), Національний банк України (НБУ), Міністерство фінансів України, а також інституції ринкової інфраструктури – фондові біржі, центральний депозитарій, клірингові організації.

НКЦПФР виступає центральним органом, відповідальним за формування і реалізацію державної політики у сфері ринків капіталу. Комісія здійснює регулювання та нагляд за професійною діяльністю, ліцензування учасників, реєстрацію випусків цінних паперів і проспектів емісії, встановлює вимоги до розкриття інформації та ухвалює нормативні акти щодо організації торгів і роботи інфраструктурних учасників [7].

Національний банк України поєднує функції центрального банку (монетарна політика, фінансова стабільність, нагляд за банками) з участю в ринку капіталу – здійснює операції з державними цінними паперами, організовує розрахунки в платіжних системах і долучається до розроблення регуляторних рішень щодо інфраструктури ринку. Міністерство фінансів України виступає

емітентом облігацій внутрішньої державної позики (ОВДП) та визначає політику державних запозичень, що безпосередньо впливає на борговий сегмент ринку.

Інституційну інфраструктуру доповнюють Національний депозитарій України та інші суб'єкти депозитарної системи, фондові біржі та організатори торгівлі, клірингові установи й розрахункові центри, які забезпечують облік прав власності на цінні папери, організацію торгів та розрахунків за операціями відповідно до профільного законодавства.

Правова основа функціонування ринку капіталу спирається на низку базових законів, серед яких:

- Закон України «Про ринки капіталу та організовані товарні ринки» визначає засади випуску й обігу цінних паперів, професійної діяльності та функціонування організованих торговельних майданчиків;
- Закон України «Про державне регулювання ринків капіталу та організованих товарних ринків» – встановлює принципи й інструменти державного регулювання, у тому числі повноваження НКЦПФР;
- Закон України «Про депозитарну систему України» – регламентує структуру депозитарної системи, види депозитарної діяльності та порядок обліку прав власності на цінні папери.

Основні інституції та відповідні нормативні акти узагальнено в таблиці 1.2.

Таблиця 1.2 – Основні інституції та нормативні акти ринку капіталу України

Елемент середовища	Опис ролі	Ключові нормативні акти / документи
НКЦПФР	Державне регулювання та нагляд за ринками капіталу, ліцензування професійної діяльності, захист прав інвесторів	Закон «Про державне регулювання ринків капіталу...»; Указ Президента «Про НКЦПФР»
НБУ	Монетарна політика, нагляд за банками, операції з ОВДП, участь у розвитку ринкової інфраструктури	Закон «Про Національний банк України»; Стратегія розвитку фінансового сектору до 2025 року
Міністерство фінансів України	Емісія державних цінних паперів, політика державних запозичень	Бюджетний кодекс України; нормативні акти щодо ОВДП
Депозитарна система (НДУ, депозитарні установи)	Ведення обліку прав власності на цінні папери, розрахунки за операціями	Закон «Про депозитарну систему України»

Елемент середовища	Опис ролі	Ключові нормативні акти / документи
Фондові біржі та організатори торгівлі	Організація біржових торгів, допуск цінних паперів до обігу, формування біржових індикаторів	Закон «Про ринки капіталу та організовані товарні ринки»; правила бірж

Подальший розвиток ринку капіталу окреслено у «Стратегії розвитку фінансового сектору України до 2025 року», спільно підготовленій НБУ, НКЦПФР, Міністерством фінансів та іншими регуляторами [7]. Серед ключових цілей – зміцнення фінансової стабільності, розвиток ринків капіталу, технологічна модернізація інфраструктури, підвищення рівня захисту інвесторів та фінансової інклюзії.

Євроінтеграційний курс України передбачає гармонізацію законодавства про ринки капіталу з правом ЄС, зокрема імплементацію регламентів і директив EMIR, CSDR, MiFID II, MiFIR [9]. Реалізація положень Угоди про асоціацію між Україною та ЄС і адаптація *acquis* ЄС у сфері фінансових послуг розглядаються як необхідна умова формування конкурентоспроможного ринку капіталу та розширення доступу українських емітентів і інвесторів до європейських фінансових ресурсів [8].



Рисунок 1.3. Інституційна структура регулювання ринку капіталу України та розподіл повноважень між основними органами.

Джерело: розроблено автором

1.1.3. Тенденції та перспективи розвитку фондового ринку України

Сучасний етап розвитку фондового ринку України визначається поєднанням воєнних викликів та стратегічного курсу на євроінтеграцію. Стратегія розвитку фінансового сектору до 2025 року серед ключових напрямів виділяє модернізацію інфраструктури ринку капіталу, цифровізацію послуг та наближення регуляторних стандартів до норм ЄС [1].

Однією з провідних тенденцій є прискорена цифровізація операцій на ринку капіталу. Як зазначають дослідники, цифрові платформи змінюють традиційні методи аналізу, а технології стають ключовим фактором конкурентоспроможності при ухваленні інвестиційних рішень [1]. Вона проявляється у розширенні використання електронних торговельних систем і віддаленого доступу до біржових майданчиків, переході до електронного документообігу та подання звітності через спеціалізовані кабінети НКЦПФР, а також у впровадженні автоматизованих систем обліку, клірингу й розрахунків, інтегрованих з платіжною інфраструктурою НБУ [3]. Переведення взаємодії професійних учасників з НКЦПФР у формат комплексної інформаційної системи з електронним кабінетом як основним каналом зв'язку підвищує прозорість операцій і створює передумови для подальшої інтеграції з державними реєстрами.

Другий важливий напрям – поглиблення інтеграції з міжнародними ринками та провайдерами ринкових даних. Котирування українських інструментів поступово повертаються на глобальні інформаційні платформи (Refinitiv, Bloomberg), що розширює коло потенційних інвесторів [4]. Розвиток інтегрованої інфраструктури ринку капіталу є предметом співпраці НКЦПФР, НБУ, Міністерства фінансів та ЄБРР; у спільних документах наголошується на створенні вертикально інтегрованої інфраструктури з узгодженими торговельними, розрахунковими та депозитарними системами [5]. Паралельно Україна рухається в напрямі приєднання до Цифрового єдиного ринку ЄС (Digital Single Market), що передбачає гармонізацію цифрових регуляторних стандартів, у тому числі у сфері фінансових послуг [6].

Третя тенденція пов'язана зі зростанням ролі технологічних платформ мультиактивної торгівлі та програмних інтерфейсів (API). Стратегія розвитку фінансового сектору підкреслює необхідність створення сучасної IT-інфраструктури ринку капіталу, сумісної з вимогами MiFID II, MiFIR, EMIR, CSDR [1]. НКЦПФР вже відкрила публічний інтерфейс доступу до баз даних емітентів та біржових операцій, що дозволяє автоматизувати отримання інформації через машинозчитувані канали [7], а оновлення регулювання щодо подання адміністративних даних стимулює використання стандартизованих електронних форматів [8]. Учасники ринку дедалі активніше застосовують модульні торговельні платформи, REST- та WebSocket-API для підключення алгоритмічних стратегій і клієнтських застосунків, а також хмарні рішення для зберігання історичних даних та побудови аналітичних моделей.

У сукупності ці тенденції формують попит на сучасні інтегровані програмні рішення для мультиактивної торгівлі й управління замовленнями. Схематично ключові тенденції та перспективи розвитку фондового ринку України наведено на рисунку 1.3.

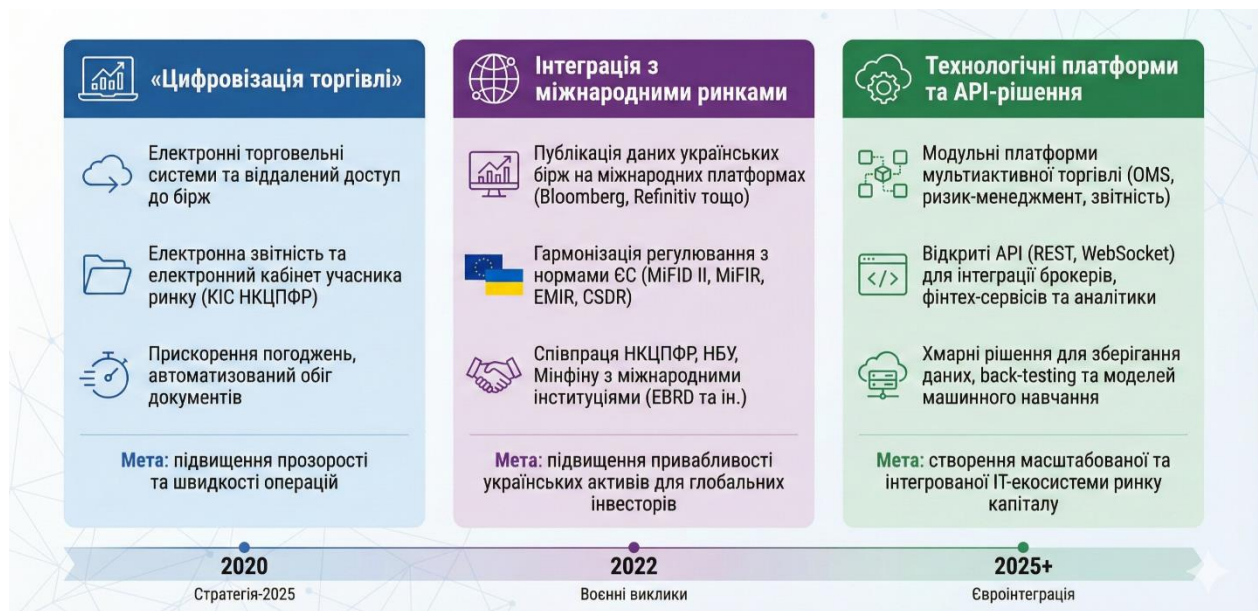


Рисунок 1.3 – Ключові тенденції та перспективи розвитку фондового ринку України.

Джерело: розроблено автором

1.1.4. Аналіз бізнес-процесів та потреб учасників ринку

Бізнес-процеси учасників фондового ринку України охоплюють повний життєвий цикл операцій – від формування інвестиційного рішення до розрахунків, обліку й регуляторної звітності. На практиці вони реалізуються через поєднання фронт-, мідл- та бек-офісних функцій брокерів, банків, інституційних і приватних інвесторів, а також інфраструктурних організацій [1]. Для проектування системи управління ордерами (OMS) ключовими є ті етапи й ролі, які впливають на створення, виконання та облік торгових доручень.

Брокери та банки як професійні посередники – виконують посередницькі функції між інвесторами та біржами. З погляду вимог до ІТ-рішень для них критичними є процеси:

- приймання та обробка торгових доручень клієнтів (онлайн-кабінети, API, телефонні доручення) з фіксацією повного аудиторного сліду;
- маршрутизація замовлень на різні біржові майданчики та інтерфейси виконання;
- інтеграція торгових процесів із внутрішніми системами ризик-менеджменту, казначейства та бухгалтерського обліку;
- пост-трейдовий облік, розрахунки та формування клієнтської й регуляторної звітності [2; 3].

У міжнародній практиці ці функції структуровані за схемою front–middle–back office: фронт-офіс працює з клієнтом і створює накази, мідл-офіс контролює ризики та ліміти, бек-офіс відповідає за розрахунки, облік і звітність [1; 2]. Відповідно, OMS має підтримувати інформаційні потоки між цими ланками без дублювання даних.

Інституційні та приватні інвестори (пенсійні фонди, страхові компанії, інвестиційні фонди, банки) сформульовують інвестиційні стратегії, управляють портфелями та генерують агреговані ордери, які передають брокерам [4]. Для них важливі: підтримка великих портфелів, гнучка аналітика, контроль лімітів і автоматизоване формування доручень.

Приватні (роздрібні) інвестори переважно працюють через онлайн-платформи та мобільні застосунки брокерів. Їхні ключові процеси – відкриття рахунку, поповнення/виведення коштів, розміщення ордерів, моніторинг позицій та отримання виписок і податкової інформації [5]. Відповідно, ядро OMS має бути готове до інтеграції з фронтендами партнерів та мобільними застосунками.

Аналіз практики використання програмно-апаратних комплексів на фондових ринках, у тому числі українському, показує низку типових проблем, які безпосередньо впливають на вимоги до нових систем [6; 8]:

1. Фрагментованість систем та дублювання функцій. Використання окремих рішень для різних класів активів і майданчиків, відсутність єдиного центру управління ордерами та позиціями, складність консолідації даних для ризик-менеджменту та звітності.

2. Низький рівень автоматизації пост-трейдових процесів. Значна частка ручних операцій у звірванні, розрахунках і формуванні звітності, підвищений ризик операційних помилок та затримок.

3. Обмежені можливості інтеграції через API. Застарілі або закриті протоколи підключення до бірж і постачальників даних, складність інтеграції з внутрішніми системами банків (core banking, ризики, бухгалтерія) та зовнішніми сервісами (регуляторні кабінети, провайдери ринкових даних).

4. Недостатні інструменти контролю відповідності нормам та аудиту. Відсутність повного машинозчитуваного аудиторного сліду по всьому життєвому циклу ордера, ускладнений контроль дотримання внутрішніх політик та регуляторних вимог (у тому числі MiFID II-подібних стандартів і вимог НКЦПФР).

5. Висока вартість та обмежена локалізація міжнародних платформ. Провідні OMS/EMS-системи орієнтовані на великі міжнародні ринки, мають високі ліцензійні та інфраструктурні витрати й недостатню адаптацію до українського законодавства, мови інтерфейсу та вимог локального регулятора.

Таблиця 1.3 – Учасники ринку, їх потреби та вимоги до системи AssetFlow

Учасник / роль	Основні бізнес-процеси	Ключові потреби	Поточні проблемні аспекти	Очікувані рішення в AssetFlow
Брокер	Приймання ордерів, маршрутизація, виконання, звітність клієнтам і НКЦПФР	Єдина платформа для всіх інструментів і майданчиків; швидке виконання; повний аудиторний слід	Фрагментовані системи, ручні операції, дублювання даних	Мультиактивна OMS із повним життєвим циклом ордера, наскрізна автоматизована обробка транзакцій, централізоване сховище даних
Банк / інституційний інвестор	Управління портфелями, формування агрегованих ордерів, контроль ризиків, звітність	Інтеграція з внутрішніми системами (core banking, ризики, бухгалтерія); гнучка аналітика; підтримка великих обсягів	Складність інтеграції між фронт-, мідл- та бек-офісними системами, обмежена аналітика	Відкрите API, модулі ризик-менеджменту та звітності, конектори до внутрішніх систем, підтримка великих портфелів
Приватний інвестор	Відкриття рахунку, поповнення, виставлення ордерів, моніторинг позицій, отримання звітів	Зручний інтерфейс, мобільний доступ, прозорі комісії, швидкий доступ до котирувань	Неінтегровані мобільні додатки, затримки в оновленні інформації, складні інтерфейси	API для платформ, що працюють під брендом партнера, реальні котирування, інтуїтивний інтерфейс користувача (через фронтенди партнерів)
Регулятор / НКЦПФР (опосередковано)	Отримання звітності, моніторинг операцій, аналіз ринкових даних	Стандартизовані формати даних, можливість автоматизованого аналізу	Нерівномірні якість і формат звітності від різних учасників	Уніфіковані формати експорту, готові звітні форми, підтримка машинозчитуваних інтерфейсів до регуляторних сервісів

У сучасних OMS життєвий цикл біржової операції складається з низки послідовних етапів, які бажано максимально автоматизувати [6]:

1. Формування інвестиційного рішення. Портфельний менеджер або клієнт, аналізуючи ринок і ліміти ризику, визначає інструмент, обсяг та напрям операції (купівля/продаж) [4].

2. Створення та реєстрація торгової заявки. У системі (через AssetFlow чи пов'язаний фронтенд) вводяться параметри заявки; перевіряються рахунок і повноваження клієнта, після чого заявка реєструється в OMS [6].

3. Перевірки та маршрутизація. Система автоматично перевіряє наявність коштів/цінних паперів, ліміти й регуляторні обмеження та направляє заявку на обраний торговельний майданчик [6; 7].

4. Виконання на біржі. Біржа (або інший майданчик) виконує заявку повністю чи частково; фіксуються параметри угоди, а інформація про виконання повертається в OMS/AssetFlow у режимі, наближеному до реального часу [6; 8].

5. Кліринг і розрахунки. На підставі підтверджених угод формуються інструкції на грошові розрахунки та поставку цінних паперів, здійснюються розрахунки й оновлюються позиції та залишки клієнтів [2; 6].

6. Облік і звітність. Дані про операції через API потрапляють до внутрішніх систем банку/брокера та регуляторних сервісів, на їх основі формуються клієнтські виписки, внутрішні та регуляторні звіти [2; 7; 8].

Схематично життєвий цикл біржової операції та точки інтеграції системи AssetFlow з учасниками й інфраструктурою ринку зображено на рисунку 1.5

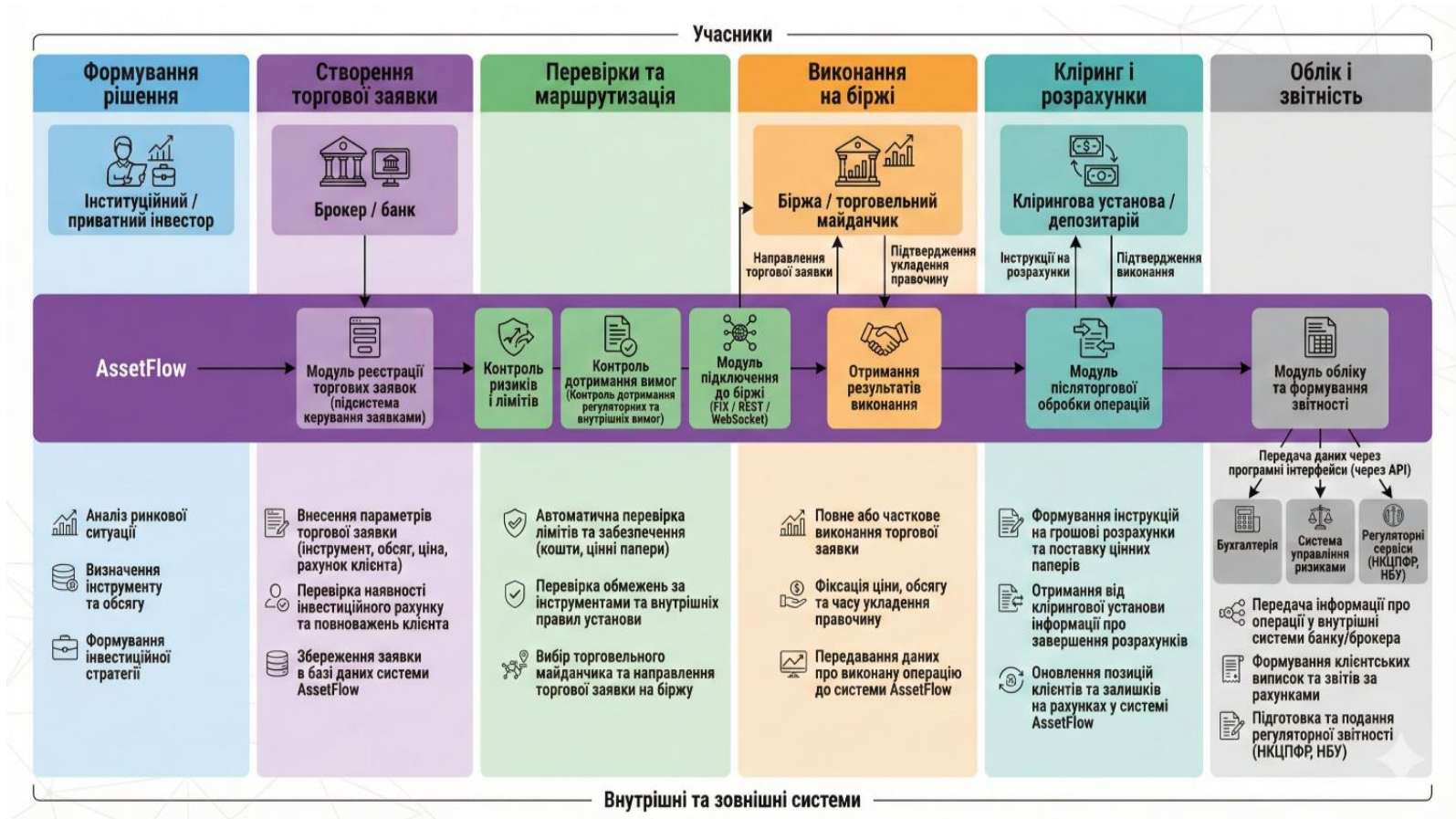


Рисунок 1.5 – Життєвий цикл біржової операції та основні точки інтеграції системи AssetFlow з учасниками та інфраструктурою фондового ринку.

Джерело: розроблено автором

Таким чином, система AssetFlow повинна:

- підтримувати повний життєвий цикл ордера (від створення до розрахунків) для різних класів активів;
- забезпечувати наскрізну автоматизовану обробку транзакцій з мінімальною кількістю ручних операцій;
- мати API-first архітектуру, сумісну з біржами, провайдерами даних, внутрішніми системами банків і регуляторними сервісами;
- відповідати актуальним та майбутнім вимогам регулювання (НКЦПФР, європейські стандарти).

1.1.5. Визначення цільової аудиторії системи AssetFlow

Цільова аудиторія системи AssetFlow охоплює професійних учасників ринку капіталу та кінцевих інвесторів. Незважаючи на відмінності бізнес-моделей, їх об'єднують подібні потреби: швидке та надійне виконання операцій, автоматизація рутинних процесів, інтеграція з іншими інформаційними системами й прозора звітність. Визначення ключових сегментів користувачів дає змогу конкретизувати вимоги до функціональності та архітектури системи. Основні сегменти користувачів і їхні очікування узагальнено в таблиці 1.4.

Таблиця 1.4 – Основні сегменти користувачів.

Сегмент користувачів	Коротка характеристика	Основні очікування / потреби щодо AssetFlow
Брокери	Професійні посередники між інвесторами та біржами, виконують клієнтські й власні операції	Швидка обробка торгових заявок; єдина платформа для різних інструментів і майданчиків; автоматизація післяторгових процесів; повний аудиторний слід; гнучкий розподіл прав доступу

Сегмент користувачів	Коротка характеристика	Основні очікування / потреби щодо AssetFlow
Фінансові установи (банки, КУА, фонди)	Працюють із великими портфелями, поєднують власні операції та послуги для клієнтів	Глибока інтеграція з внутрішніми системами (облік, ризики, платіжна інфраструктура); підтримка великих обсягів операцій; аналітика портфельів; автоматизована регуляторна звітність
Алготрейдери та фінтех-сервіси	Використовують програмні стратегії торгівлі та власні клієнтські застосунки	Надійний API для подання заявок і доступу до ринкових та історичних даних; стабільна робота без довільних обмежень; можливість тестування стратегій; гнучкі налаштування лімітів, сповіщень і журналів дій
Приватні інвестори (через сервіси брокерів та банків)	Фізичні особи, що інвестують самостійно через онлайн-сервіси	Зручний веб- або мобільний інтерфейс, що отримує дані з AssetFlow; швидке відображення результатів операцій; зрозумілі виписки та звіти; прозора інформація про комісії й ризики



Рисунок 1.6 – Сегменти цільової аудиторії системи AssetFlow та ключові напрями створення цінності для кожної групи користувачів.

Джерело: розроблено автором

1.1.6. Аналіз конкурентів та аналогів (систем управління заявками та торговельних платформ)

Для обґрунтування доцільності створення системи AssetFlow доцільно виокремити основні класи наявних рішень: глобальні інформаційно-торговельні комплекси (Bloomberg, Refinitiv), роздрібні та брокерські платформи (MetaTrader 5, Trader Workstation), а також локальні рішення – біржові термінали, системи типу QUIK, Exante й внутрішні розробки українських брокерів [1–11]. Вони забезпечують доступ до різних ринків і рівень автоматизації, але або надто дорогі й складні для більшості українських установ, або не забезпечують повної інтеграції та відповідності локальним регуляторним вимогам. Узагальнену порівняльну характеристику наведено в таблиці 1.5.

Таблиця 1.5 – Порівняльна характеристика торговельних платформ та систем управління заявками

Рішення / клас	Покриття ринків та інструментів	Автоматизація життєвого циклу операції	Інтеграція та відкритість	Відповідність українським реаліям	Орієнтовна вартість володіння
Bloomberg, Refinitiv (глобальні комплекси)	Широке охоплення міжнародних ринків, акції, облігації, деривативи, FX [1–4]	Високий рівень автоматизації всіх етапів операції, вбудований контроль обмежень	Є засоби інтеграції, але орієнтовані на великі міжнародні фінансові установи	Немає спеціальної адаптації під українські регуляторні вимоги; надлишковий функціонал для більшості	Дуже висока, потребує значних постійних витрат

Рішення / клас	Покриття ринків та інструментів	Автоматизація життєвого циклу операції	Інтеграція та відкритість	Відповідність українським реаліям	Орієнтовна вартість володіння
				локальних учасників	
MetaTrader 5, TWS (роздрібні та брокерські платформи)	Доступ до багатьох іноземних ринків; MT5 – фокус на FX та похідних, TWS – акції, деривативи, облігації [5–8]	Часткова автоматизація; зручні для окремого користувача, але не для повної установи	Мають технічні інтерфейси, але прив'язані до конкретних брокерів	Не адаптовані до роботи з українським і біржами та регуляторам і, обмежена відповідність локальній звітності	Середня/висока (залежно від моделі роботи з брокером)
Біржові термінали, QUIK, Exante (локальні/регіональні рішення)	Переважно один або кілька майданчиків; іноді доступ до іноземних бірж [9–11]	Основний акцент – введення заявок та базовий післяопераційний облік	Обмежена інтеграція з внутрішніми системами установи; часто відсутня єдина платформа для всіх напрямів діяльності	Частково відповідають українським умовам, але не закривають усі потреби щодо звітності, ризиків та аналітики	Відносно нижча, потребує доробок і підтримки з боку брокера
AssetFlow (запроєктована система)	Орієнтація на мультиактивну торгівлю (акції, облігації, деривативи)	Передбачена підтримка повного життєвого циклу операції: від заявки до	Концепція «відкритого» API для інтеграції з біржами, внутрішніми системами	Спеціальна адаптація до вимог НКЦПФР та НБУ, підтримка української	Помірна; за рахунок цільового функціоналу та локальної підтримки

Рішення / клас	Покриття ринків та інструментів	Автоматизація життєвого циклу операції	Інтеграція та відкритість	Відповідність українським реаліям	Орієнтовна вартість володіння
	з фокусом на українських і вибраних міжнародних ринках	обліку та звітності	банків/брокерів та регуляторних і сервісами	мови та локальної звітності	

Провідні міжнародні платформи забезпечують надзвичайно широкий функціонал і підходять для глобальних фінансових груп, однак їхня вартість, складність та відсутність локальної адаптації роблять їх малопридатними як базову систему для більшості українських брокерів і банків. Локальні біржові термінали та окремі платформи типу QUIK або Exante частково задовольняють потребу в доступі до торгів, але не забезпечують єдиного управління процесами, повноцінної інтеграції з внутрішніми системами та гнучкого формування звітності.

У цих умовах система AssetFlow може зайняти нішу мультиактивної платформи управління торговими заявками, яка:

- поєднує автоматизацію всіх етапів життєвого циклу операції;
- має відкритий програмний інтерфейс для інтеграції з біржами, даними та внутрішніми системами учасників;
- спеціально адаптована до українських регуляторних та операційних реалій;
- пропонує прийнятну вартість володіння для локальних учасників ринку капіталу.

Цей висновок безпосередньо пов'язує аналіз конкурентів із обґрунтуванням потреби у створенні системи AssetFlow.

1.1.7. Бізнес-модель AssetFlow (Business Model Canvas)

Бізнес-модель системи AssetFlow доцільно описати за підходом Business Model Canvas А. Остервалдера, що охоплює дев'ять елементів: сегменти клієнтів, ціннісні пропозиції, канали, взаємодію з клієнтами, джерела доходів, ключові ресурси, ключові види діяльності, каналів, взаємодію з клієнтами, джерела доходів, ключові ресурси, ключові види діяльності, ключових партнерів та структуру витрат. Узагальнену бізнес-модель наведено на рисунку 1.7.



Рисунок 1.7 – Бізнес-модель системи AssetFlow у форматі Business Model Canvas.

Джерело: розроблено автором

1.2. Постановка цілей та завдань створення системи AssetFlow

На основі аналізу фондового ринку України, бізнес-процесів його учасників та виявлених проблем (фрагментованість рішень, низький рівень автоматизації, відсутність уніфікованих програмних інтерфейсів, слабка аналітика тощо) постає потреба у створенні нового програмного продукту – системи AssetFlow.

1.2.1. Нова проблема та потреба компаній на фондовому ринку

Сучасні брокери, банки та інші професійні учасники ринку капіталу в Україні змушені працювати в умовах:

- розірваної IT-інфраструктури – окремі торговельні термінали для різних майданчиків, окремі системи обліку, окремі внутрішні інструменти для ризиків і звітності;
- значної частки ручної праці – звіряння операцій, формування реєстрів, підготовка звітності для клієнтів і регулятора;
- складних і дорогих інтеграцій – для зв'язку між внутрішніми системами, біржами, депозитаріями, сервісами регулятора;
- обмежених аналітичних можливостей – труднощі з оперативною оцінкою портфелів, ризику, результатів діяльності.

Водночас ринок стикається з новими викликами:

- цифровізація фінансових послуг і перехід клієнтів в онлайн;
- посилення вимог до прозорості, звітності та контролю з боку регулятора;
- інтеграція з європейським фінансовим простором та адаптація до практик розвинених ринків.

У цих умовах традиційні, розрізнені рішення (біржові термінали, саморобні бек-офісні системи, електронні таблиці) вже не відповідають потребам ринку.

Постає нова потреба: мати єдину, інтегровану платформу, яка:

- підтримує мультиактивну торгівлю та повний життєвий цикл операцій;

- забезпечує обмін даними з внутрішніми системами й зовнішніми сервісами;
- надає зручні інструменти аналітики та звітності.

Саме на цю потребу покликано відповісти проєкт створення системи AssetFlow.

1.2.2. Загальна мета створення системи AssetFlow

Метою створення системи AssetFlow є розробка та впровадження інтегрованої програмної платформи для учасників фондового ринку України, яка забезпечить:

- єдиний інформаційний простір для роботи з торговими операціями, позиціями та портфелями;
- автоматизацію ключових бізнес-процесів – від подання торгових заявок до обліку та звітності;
- підвищення прозорості та керованості ризиками за рахунок повного журналу дій і системи контролю обмежень;
- готовність до інтеграції з внутрішніми та зовнішніми системами, включаючи майбутні вимоги регулювання та міжнародні стандарти.

Іншими словами, AssetFlow має стати ядром ІТ-екосистеми брокера чи фінансової установи, яке об'єднує торгівлю, облік, аналітику та взаємодію з інфраструктурою ринку.

1.2.3. Завдання проєкту створення системи AssetFlow

Для досягнення поставленої мети необхідно розв'язати низку конкретних завдань:

1. Проаналізувати бізнес-процеси брокерських компаній, банків та інших учасників ринку й визначити, які з них доцільно підтримати в рамках системи AssetFlow.

2. Сформувати детальні вимоги до функціональності системи (управління торговими заявками, облік операцій, аналітика, звітність, інтеграція з іншими системами).
3. Розробити архітектуру системи, що забезпечить:
 - можливість масштабування за кількістю користувачів і обсягом операцій;
 - надійність і відмовостійкість;
 - захист інформації.
4. Реалізувати прототип системи з ключовими модулями:
 - керування торговими заявками та угодами;
 - облік позицій і залишків;
 - базова аналітика та звітність.
5. Забезпечити можливість інтеграції з:
 - торговельними системами бірж;
 - внутрішніми системами обліку та ризик-менеджменту;
 - сервісами подання звітності для регулятора.
6. Підготувати рекомендації щодо подальшого розвитку системи (розширення функціональності, підключення нових ринків, впровадження розширеної аналітики).

Ці завдання будуть деталізовані у наступних підрозділах через вимоги, сценарії використання та архітектурні рішення.

1.2.4. Канва ціннісної пропозиції системи AssetFlow (Value Proposition Canvas)

Канва ціннісної пропозиції використовується для того, щоб зіставити потреби клієнтів із тим, яку цінність пропонує продукт. У межах даного проєкту доцільно зосередитися на ключовому сегменті – брокерські компанії / інвестиційні підрозділи банків.



Рисунок 1.8 – Канва ціннісної пропозиції системи AssetFlow для брокерів та інвестиційних підрозділів банків

Джерело: розроблено автором

Таким чином, канва ціннісної пропозиції демонструє, що система AssetFlow безпосередньо відповідає ключовим потребам брокерів та інвестиційних підрозділів банків. Вона знімає основні «болі» клієнта – фрагментованість IT-інфраструктури, високу частку ручної праці, складність інтеграцій та слабку аналітику – за рахунок єдиної платформи для роботи з операціями, вбудованих модулів обліку, звітності й контролю ризиків, а також стандартизованих програмних інтерфейсів.

Одночасно AssetFlow формує додаткову цінність: створює основу для запуску нових інвестиційних продуктів і клієнтських сервісів, підвищує прозорість діяльності установи для керівництва й регулятора, а також дозволяє масштабувати бізнес без пропорційного зростання витрат на персонал та підтримку IT-інфраструктури. Це підтверджує доцільність розробки системи як стратегічного елемента цифрової трансформації учасників фондового ринку України.

1.2.5. Дошка бачення продукту AssetFlow (Product Vision Board)

Дошка бачення продукту використовується для компактного опису для кого, який продукт створюється, які потреби він закриває та які бізнес-цілі переслідує.



Рисунок 1.9 – Дошка бачення продукту AssetFlow (Product Vision Board)

Джерело: розроблено автором

Дошка бачення продукту демонструє, що система AssetFlow розробляється для конкретних груп користувачів: брокерів, інвестиційних підрозділів банків та пов’язаних із ними фінансово-технологічних компаній. Саме їхні повсякденні задачі – оперативне й безпечне проведення операцій з цінними паперами, повний огляд портфелів та стану рахунків, спрощення підготовки звітів і контроль ризиків – покладено в основу концепції продукту.

Водночас AssetFlow розглядається не лише як робочий інструмент, а як елемент зміни бізнес-моделі установи. Система дає можливість скоротити витрати на рутинні дії персоналу, підвищити рівень сервісу для клієнтів, безболісно нарощувати обсяги операцій, виконуючи вимоги регулятора, та поступово переходити до європейських підходів до звітності й нагляду. Таким чином, AssetFlow стає платформою, що підтримує цифрову трансформацію учасників фондового ринку та допомагає їм утримувати й посилювати конкурентні позиції.

1.3. Визначення вимог до продукту AssetFlow

У підрозділах 1.1–1.2 було показано, що учасники фондового ринку України працюють у фрагментованому ІТ-середовищі, мають значні обсяги ручної роботи та стикаються з новими вимогами цифровізації й регулювання. У цьому підрозділі ці висновки перетворюються на конкретні вимоги до системи AssetFlow: визначаються зацікавлені сторони, їхні інтереси, формуються бізнес-вимоги, описуються ключові сценарії використання (Use case), користувацькі історії (User story) та основні вхідні/вихідні дані.

1.3.1. Зацікавлені сторони та їхні інтереси

До системи AssetFlow дотичні кілька груп зацікавлених сторін:

- Брокерські компанії (оператори, бек-офіс, ризик-менеджери, керівництво).
- Інвестиційні підрозділи банків та компанії з управління активами.
- Приватні та інституційні інвестори, які користуються сервісами брокерів і банків.
- Фінансово-технологічні компанії та алгоритмічні трейдери, що підключаються до платформи.
- Інфраструктурні установи – біржі, депозитарії, клірингові організації.
- Органи регулювання та нагляду (НКЦПФР, НБУ).
- Державні цифрові сервіси (Дія, BankID та інші системи е-ідентифікації й е-підпису).
- Команда експлуатації та підтримки системи в самій установі (адміністратори, ІТ-фахівці).

Для узагальнення їхніх потреб можна використати таблицю.

Таблиця 1.7 Зацікавлені сторони та їхні інтереси щодо системи AssetFlow

Зацікавлена сторона	Основні інтереси та очікування
Брокерські компанії, інвестиційні підрозділи банків	Швидке й безпомилкове виконання торгових заявок; єдина картина по

	рахунках і портфелях; автоматизація післяопераційних процесів; зручна звітність; виконання регуляторних вимог; можливість дистанційного обслуговування клієнтів.
Приватні та інституційні інвестори	Надійність і прозорість операцій; оперативне відображення результатів угод; зрозумілі виписки та звіти; можливість дистанційної роботи без відвідування офісу.
Фінтех-компанії, алгоритмічні трейдери	Програмний інтерфейс доступу (API) до торгівлі й ринкових даних; стабільність роботи; можливість тестування стратегій на історичних даних.
Біржі, депозитарії, клірингові установи	Коректний обмін даними в стандартизованих форматах; своєчасне передавання заявок і результатів розрахунків; дотримання вимог безпеки та конфіденційності.
Регуляторні органи (НКЦПФР, НБУ)	Достовірна та своєчасна звітність; можливість перевірки правильності операцій; слідкування за дотриманням лімітів та обмежень.
Державні цифрові сервіси (Дія, BankID)	Коректне використання каналів е-ідентифікації та е-підпису; відповідність законодавчим вимогам щодо захисту персональних даних.
Команда експлуатації та підтримки	Простота адміністрування; можливість моніторингу роботи системи; зручні засоби резервування та відновлення; чітка журналізація подій.

1.3.2. Бізнес-вимоги до системи AssetFlow

Бізнес-вимоги описують що саме має давати система організації в цілому, не заглиблюючись у технічну реалізацію. Їх доцільно пронумерувати (BR – Business Requirement).

BR-1. Єдина платформа для мультиактивної торгівлі. Система має забезпечувати роботу з основними видами фінансових інструментів (акції, облигації, похідні) та різними торговельними майданчиками в межах єдиної платформи, з уніфікованою логікою роботи для користувачів.

BR-2. Повний життєвий цикл операції. AssetFlow має підтримувати всі етапи операції з цінними паперами: від створення торгової заявки клієнтом до розрахунків, обліку й формування звітності.

BR-3. Автоматизація післяопераційних процесів. Має бути мінімізована ручна робота бек-офісу: формування реєстрів, оновлення позицій, нарахування доходу, підготовка клієнтських та внутрішніх звітів.

BR-4. Інтеграція з внутрішніми системами установи. Система повинна мати програмні інтерфейси для обміну даними з бухгалтерським обліком, системами управління ризиками, платіжними та казначейськими модулями.

BR-5. Інтеграція з біржами, депозитаріями, кліринговими установами. AssetFlow має підтримувати обмін даними в узгоджених форматах, забезпечуючи своєчасне передавання заявок і отримання результатів виконання та розрахунків.

BR-6. Підтримка державних цифрових сервісів. Система має інтегруватися з сервісами е-ідентифікації та е-підпису (Дія, BankID тощо), щоб забезпечувати дистанційне відкриття рахунків і підписання документів відповідно до законодавства.

BR-7. Прозорість і контроль дотримання вимог. Усі дії з торговими заявками й операціями мають фіксуватися в журналі подій; повинні діяти правила контролю лімітів та обмежень, з можливістю відстеження їх виконання.

BR-8. Аналітика та підтримка прийняття рішень. Система має надавати засоби аналізу портфелів, оцінки дохідності та ризику, а також оглядові панелі для керівництва, ризик-менеджерів і бек-офісу.

BR-9. Підтримка клієнтських онлайн-сервісів. AssetFlow повинна виступати джерелом даних для веб-кабінетів і мобільних застосунків клієнтів, забезпечуючи оперативне відображення операцій та стану рахунків.

BR-10. Масштабованість і довгостроковий розвиток. Архітектура системи має дозволяти розширення функціональності, підключення нових ринків та збільшення обсягів операцій без радикальної перебудови.

1.3.3. Основні сценарії використання (Use case) та взаємодія із зовнішнім середовищем

Сценарії використання (Use case) описують типові ланцюжки дій користувачів і зовнішніх систем, у яких задіяна AssetFlow. Це допомагає зрозуміти поведінку системи «ззовні».

Таблиця 1.3.1 – Основні актори системи «AssetFlow»

Актор	Опис ролі
Клієнт-інвестор	Фізична/юридична особа, що відкриває рахунок, подає торгові доручення, отримує звітність.
Оператор брокера / банку	Вводить заявки клієнтів, ініціює операції в системі, комунікує з клієнтом.
Співробітник бек-офісу	Обробляє реєстри операцій, формує виписки, готує звітність.
Ризик-менеджер	Налаштовує ліміти та обмеження, контролює ризики операцій.
Адміністратор системи	Підтримує роботу AssetFlow, керує правами доступу та налаштуваннями.
Біржа / торговельний майданчик	Приймає та виконує торгові заявки, повертає результати виконання.
Депозитарій / клірингова установа	Забезпечує облік прав власності, кліринг та розрахунки за операціями.
Сервіси регулятора	Канали подання обов'язкової звітності (НКЦПФР, НБУ тощо).
Державні сервіси е-ідентифікації	Дія, BankID – забезпечують дистанційну ідентифікацію та підтвердження даних клієнта.

Приклад Use case 1 – «Відкриття рахунку клієнта з використанням Дія / BankID»

Мета: дистанційно ідентифікувати клієнта та відкрити інвестиційний рахунок.

Актори: Клієнт-інвестор, Оператор брокера/банку, Система AssetFlow, Дія/BankID, внутрішня облікова система.

Передумови: у брокера/банку налаштована інтеграція з Дія/BankID; клієнт має доступ до цих сервісів.

Результат: створений профіль клієнта та інвестиційний рахунок, дані передано в облікову систему

Таблиця 1.3.2 – Сценарій UC-1 «Відкриття рахунку через Дія / BankID»

№ кроку	Актор	Опис дії
1	Оператор	Створює в AssetFlow заявку на відкриття рахунку клієнта.
2	Система AssetFlow	Пропонує варіант дистанційної ідентифікації через Дія/BankID.
3	Клієнт, Дія/BankID	Клієнт отримує запит та підтверджує ідентифікацію у Дія/BankID.
4	Дія/BankID	Передає до AssetFlow персональні дані клієнта та статус ідентифікації.
5	Система AssetFlow	Створює профіль клієнта та інвестиційний рахунок, фіксує джерело даних і згоду клієнта.
6	Система AssetFlow	Передає інформацію про рахунок у внутрішню облікову систему.
7	Оператор	Отримує підтвердження про успішне відкриття рахунку.

Альтернативи: у разі відмови або помилки в Дія/BankID оператор переходить до ручної процедури з очною ідентифікацією.

Приклад Use case 2 – «Подання та виконання торгової заявки»

Мета: виконати торгову операцію клієнта на біржі.

Актори: Клієнт-інвестор, Оператор брокера, Система AssetFlow, Біржа,

Ризик-менеджер (непрямо).

Таблиця 1.3.3 – Сценарій UC-2 «Подання та виконання торгової заявки»

№ кроку	Актор	Опис дії
1	Клієнт	Подає брокеру доручення (онлайн-кабінет, телефон або інший канал).
2	Оператор / API	Вводить торгову заявку в AssetFlow або заявка надходить через програмний інтерфейс.
3	Система AssetFlow	Перевіряє наявність рахунку, залишків коштів/цінних паперів, лімітів та регуляторних обмежень.
4	Система AssetFlow	У разі успішних перевірок направляє заявку до біржі через торговий шлюз.
5	Біржа	Виконує заявку повністю/частково та повертає результат виконання.
6	Система AssetFlow	Оновлює позиції клієнта, фіксує параметри операції, формує внутрішні записи для розрахунків.
7	Система AssetFlow	Генерує повідомлення клієнту про виконання доручення.

Варіанти: відмова заявки через порушення лімітів або технічну помилку, часткове виконання, відкликання заявки.

Приклад Use case 3 – «Формування клієнтської виписки»

Мета: надати клієнту повну й зрозумілу інформацію про стан рахунку та операції.

Актори: Клієнт-інвестор, Співробітник бек-офісу, Система AssetFlow.

Таблиця 1.3.4 – Сценарій UC-3 «Формування клієнтської виписки»

№ кроку	Актор	Опис дії
1	Клієнт / Бек-офіс	Ініціює формування виписки за обраний період у кабінеті або в інтерфейсі бек-офісу.
2	Система AssetFlow	Збирає дані про залишки, позиції, рух коштів і цінних паперів клієнта.
3	Система AssetFlow	Формує документ заданого формату (PDF/Excel) з поясненнями комісій та показниками дохідності.
4	Система AssetFlow	Надає виписку клієнту через онлайн-кабінет або надсилає погодженим каналом.

1.3.3.5. Приклад Use case 4 – «Формування регуляторної звітності»

Актори: Співробітник бек-офісу, Представник служби контролю відповідності нормам, Система AssetFlow, Сервіси НКЦПФР/НБУ.

Мета: сформувати та подати обов'язкові звіти до регулятора в потрібному форматі.

Таблиця 1.3.5 – Сценарій UC-4 «Формування регуляторної звітності»

№ кроку	Актор	Опис дії
1	Бек-офіс / Контроль	Обирає тип звіту та звітний період.
2	Система AssetFlow	Збирає необхідні дані та виконує базову перевірку повноти й коректності.
3	Система AssetFlow	Формує файл у форматі, наближеному до вимог відповідного регулятора.
4	Відповідальні особи	Переглядають, за потреби коригують та затверджують звіт.
5	Система AssetFlow / Сервіс	Передає звіт через канал подання звітності, фіксує факт відправлення та результат у журналі.

1.3.4. Користувацькі історії (User story) для ключових ролей

Користувацькі історії відображають вимоги «мовою користувача». Формат: «Як <роль>, я хочу <дія>, щоб <очікувана користь>».

Нижче наведено приклади для основних ролей.

Таблиця 1.3.6 – Користувацькі історії для основних ролей системи «AssetFlow»

Код	Роль	Користувацька історія (формат «Як ..., я хочу ..., щоб ...»)
US-01	Оператор брокера	Як оператор брокера, я хочу швидко вводити та змінювати торгові заявки клієнтів, щоб скорочувати час від моменту доручення до його виконання.
US-02	Оператор брокера	Як оператор брокера, я хочу бачити історію дій із кожною заявкою, щоб у разі претензій клієнта можна було відновити повну картину подій.
US-03	Ризик-менеджер	Як ризик-менеджер, я хочу налаштовувати ліміти на клієнтів, інструменти та види операцій, щоб система автоматично блокувала ризиковані заявки.
US-04	Співробітник бек-офісу	Як співробітник бек-офісу, я хочу автоматично отримувати реєстри операцій за кінець дня, щоб уникнути ручного звіряння з біржами та депозитаріями.
US-05	Клієнт-інвестор	Як клієнт-інвестор, я хочу мати змогу переглядати свій портфель і історію операцій у кабінеті, щоб контролювати стан інвестицій.
US-06	Клієнт-інвестор	Як клієнт-інвестор, я хочу проходити ідентифікацію через Дія або BankID, щоб відкривати рахунок і підписувати документи дистанційно.
US-07	Керівник інвестиційного підрозділу	Як керівник інвестиційного підрозділу, я хочу мати оглядові панелі зі зведеною інформацією про портфелі та дохідність, щоб оперативно приймати управлінські рішення.
US-08	Регулятор / аудитор	Як представник служби контролю відповідності нормам чи аудитор, я хочу мати доступ до журналу дій і звітів у стандартизованому вигляді, щоб перевіряти дотримання внутрішніх і зовнішніх вимог.
US-09	Адміністратор системи	Як адміністратор системи, я хочу гнучко налаштовувати ролі та права доступу, щоб кожен користувач бачив тільки ті функції та дані, які йому потрібні.

Ці історії ляжуть в основу деталізації функціональних вимог і планування реалізації.

1.3.5. Функціональні вимоги до системи AssetFlow

Функціональні вимоги описують конкретні дії, які система повинна виконувати. Їх можна згрупувати за підсистемами.

Таблиця 1.3.7 – Функціональні вимоги до системи «AssetFlow»

Код	Підсистема	Опис функціональної вимоги	Пов'язано з
FR-1	Управління клієнтами та рахунками	Система повинна забезпечувати створення та зберігання профілів клієнтів із необхідними реквізитами (персональні дані, контакти, KYC-атрибути).	UC-1, US-05
FR-2	Управління клієнтами та рахунками	Система повинна підтримувати дистанційну ідентифікацію клієнтів через Дія/BankID з фіксацією джерела даних та статусу перевірки.	UC-1, US-06
FR-3	Управління клієнтами та рахунками	Система повинна створювати й обслуговувати кілька рахунків клієнта в різних валютах та інструментах (грошові, брокерські, депозитарні тощо).	UC-1, US-05
FR-4	Торгові заявки та операції	Система повинна дозволяти створення, зміну та скасування торгових заявок як вручну користувачами, так і через програмні інтерфейси (API).	UC-2, US-01
FR-5	Торгові заявки та операції	Перед надсиланням заявки на біржу система повинна автоматично перевіряти наявність рахунку, залишків коштів/цінних паперів, лімітів і обмежень.	UC-2, US-03
FR-6	Торгові заявки та операції	Система повинна передавати торгові заявки до біржових або інших торговельних систем і приймати відповіді про виконання (повне/часткове/відмову).	UC-2, US-01
FR-7	Торгові заявки та операції	Система повинна відображати статуси заявок та результати їх виконання в режимі, наближеному до реального часу.	UC-2, US-01, US-05
FR-8	Післяопераційний облік і розрахунки	Система повинна автоматично формувати інструкції на розрахунки за операціями й передавати їх у відповідні системи/депозитарії.	UC-2, UC-4, US-04
FR-9	Післяопераційний облік і розрахунки	Після отримання підтвердження розрахунків система повинна оновлювати позиції та залишки клієнтів.	UC-2, US-05
FR-10	Звітність і аналітика	Система повинна формувати клієнтські виписки за обрані періоди в установлених форматах (наприклад, PDF, Excel).	UC-3, US-05

Код	Підсистема	Опис функціональної вимоги	Пов'язано з
FR-11	Звітність і аналітика	Система повинна створювати внутрішні звіти за портфелями, інструментами, клієнтами, доходами та витратами.	UC-3, US-07
FR-12	Звітність і аналітика	Система повинна формувати файли звітності для регулятора відповідно до затверджених шаблонів та форматів.	UC-4, US-08
FR-13	Звітність і аналітика	Система повинна надавати оглядові панелі (дашборди) для керівництва та ризик-менеджерів із ключовими показниками портфелів і ризиків.	UC-3, UC-4, US-07
FR-14	Інтеграція та програмні інтерфейси	Система повинна мати документований програмний інтерфейс (API) для інтеграції з внутрішніми системами банку/брокера та зовнішніми сервісами.	UC-1–UC-4, US-01–06
FR-15	Інтеграція та програмні інтерфейси	Система повинна підтримувати імпорт/експорт довідників, історичних даних та результатів розрахунків у машинозчитуваних форматах.	UC-3, UC-4
FR-16	Управління доступом і безпекою	Система повинна підтримувати багаторівневу модель ролей і прав доступу, з можливістю гнучкої конфігурації для різних категорій користувачів.	US-09
FR-17	Управління доступом і журнал подій	Система повинна фіксувати всі суттєві дії користувачів та системні події в журналі з можливістю пошуку, фільтрації та експорту записів.	UC-4, US-08, US-09

1.3.6. Нефункціональні вимоги

Нефункціональні вимоги характеризують якість роботи системи.

Таблиця 1.3.8 – Нефункціональні вимоги до системи «AssetFlow»

Код	Категорія	Опис вимоги	Орієнтовний показник / примітка
NFR-1	Продуктивність	Система повинна опрацьовувати типові операції (створення/зміна заявки, перегляд портфеля, формування короткої виписки) із прийнятним часом відгуку.	Напр., час відгуку ≤ 1 с для 95 % запитів у межах типової завантаженості.
NFR-2	Доступність	Система повинна забезпечувати доступність у межах торгових днів відповідно до узгоджених SLA.	Напр., доступність не менше 99 % часу в межах торгових сесій.
NFR-3	Безпека	Має бути забезпечено автентифікацію користувачів, розмежування доступу,	Використання захищених протоколів (TLS), ролей, політик доступу.

Код	Категорія	Опис вимоги	Орієнтовний показник / примітка
		шифрування чутливих даних і захист каналів зв'язку.	
NFR-4	Надійність та відмовостійкість	Передбачається резервування критичних компонентів та регулярне резервне копіювання даних із можливістю відновлення після збоїв.	RPO та RTO узгоджуються із замовником; наявність плану відновлення.
NFR-5	Масштабованість	Система повинна підтримувати збільшення кількості користувачів і обсягу операцій без суттєвого погіршення продуктивності.	Можливість горизонтального/вертикального масштабування компонентів.
NFR-6	Зручність використання	Інтерфейси користувачів мають бути інтуїтивно зрозумілими, із підтримкою української мови (за можливості – кількох мов).	Проведення UX-тестування; локалізація інтерфейсів українською.
NFR-7	Супроводжуваність	Архітектура та документація системи повинні забезпечувати можливість подальшого розвитку, виправлення помилок і інтеграції нових компонентів.	Наявність технічної документації, стандартів коду, описаних API.

1.3.7. Вхідні та вихідні дані системи

Для завершення визначення вимог потрібно описати, які дані надходять до AssetFlow та що система повертає назовні.

Таблиця 1.8 – Основні вхідні та вихідні дані системи AssetFlow

Категорія	Вхідні дані	Вихідні дані
Клієнти та рахунки	персональні дані клієнтів (з форми або через Дія/BankID); дані про договори, відкриті рахунки	підтвердження відкриття рахунків; оновлені дані профілю; журнали ідентифікації та підписання документів
Торгові операції	доручення клієнтів; торгові заявки із зовнішніх застосунків; параметри інструментів та ринків	підтвердження виконання заявок; реєстри операцій; дані для післяопераційних розрахунків
Ринкові дані	котирування, обсяги, довідники інструментів з бірж/постачальників	агреговані дані для відображення клієнтам та аналітики
Розрахунки й облік	повідомлення від депозитаріїв і клірингових установ про виконання розрахунків	оновлені позиції та залишки; записи для бухгалтерського обліку
Регуляторна інформація	вимоги до звітності, шаблони файлів, довідники регулятора	сформовані файли звітності; підтвердження відправлення та статуси прийняття
Системна інформація	настройки лімітів, ролей, параметрів безпеки	журнали подій, діагностичні дані, показники для моніторингу

Таким чином, у підрозділі 1.3 було здійснено перехід від загальної проблематики та бачення продукту до формалізованих вимог. Описані зацікавлені сторони, їхні інтереси, бізнес-вимоги, ключові сценарії використання, користувацькі історії, а також функціональні й нефункціональні вимоги та основні потоки даних. Це створює основу для подальшого проектування архітектури та технічної реалізації системи AssetFlow у наступних розділах роботи.

Висновки до розділу 1 «Дизайн бізнесу AssetFlow»

У першому розділі проведено всебічний аналіз предметної області, в якій створюється система AssetFlow. Показано, що фондовий ринок України є ключовим елементом ринку капіталу, проте його структура залишається дисбалансованою: домінує сегмент державних облігацій, тоді як ринки акцій та похідних інструментів характеризуються низькою ліквідністю й обмеженим колом активних емітентів та інвесторів. Окреслено основні класи фінансових інструментів, інституційне та нормативно-правове середовище ринку, а також стратегічний курс на гармонізацію регулювання із законодавством ЄС і цифрову трансформацію інфраструктури.

На основі аналізу тенденцій (цифровізація торгівлі, інтеграція з міжнародними майданчиками й провайдерами ринкових даних, зростання ролі технологічних платформ та програмних інтерфейсів взаємодії) деталізовано бізнес-процеси брокерів, банків, інституційних та приватних інвесторів. Показано, що їхня щоденна діяльність спирається на повний життєвий цикл біржової операції – від формування інвестиційного рішення до клірингу, розрахунків, обліку й регуляторної звітності. Виявлено типові проблемні аспекти чинних ІТ-рішень: фрагментованість систем, значна частка ручної праці в післяторгових процесах, складність інтеграції з біржами, депозитаріями, внутрішніми системами установ і державними сервісами, а також висока вартість і слабка локалізація міжнародних платформ.

Проведений аналіз конкурентних рішень (глобальні інформаційно-торговельні комплекси, роздрібні торговельні платформи, біржові термінали та

власні програмні розробки українських брокерів) показав, що жоден із наявних класів продуктів не забезпечує поєднання мультиактивної торгівлі, глибокої інтеграції з українською інфраструктурою ринку капіталу, адаптації до вимог НКЦПФР та НБУ і прийнятної вартості володіння для локальних учасників. Саме на перетині цих незадоволених потреб формується ринкова ніша для системи AssetFlow.

У межах розділу сформовано бізнес-модель AssetFlow з використанням шаблону бізнес-моделі (Business Model Canvas): визначено ключових партнерів (біржі, депозитарії, банки, постачальники даних, державні цифрові сервіси, ІТ-інтегратори), основні види діяльності й ресурси, сегменти клієнтів, ціннісні пропозиції для кожної групи користувачів, канали збуту та структуру витрат і доходів. Додатково через канву ціннісної пропозиції та дошку бачення продукту уточнено, які завдання, проблеми та очікувані вигоди мають брокери, інвестиційні підрозділи банків, фінансово-технологічні компанії та кінцеві інвестори, і як саме AssetFlow має ці запити закривати.

На підставі отриманих результатів у підрозділі 1.2 сформульовано нову потребу ринку – створення єдиної інтегрованої платформи управління торгами, позиціями та звітністю – та визначено мету й завдання проєкту розробки системи AssetFlow. У підрозділі 1.3 цю потребу трансформовано у формалізовані вимоги: окреслено зацікавлені сторони й їхні інтереси, сформульовано бізнес-вимоги, описано основні сценарії використання, функціональні потреби користувачів, функціональні та нефункціональні вимоги, а також ключові потоки вхідних і вихідних даних.

Отже, розділ 1 виконує функцію концептуального фундаменту роботи: він одночасно обґрунтовує актуальність створення системи AssetFlow для українського фондового ринку та задає рамки майбутнього проєктування й реалізації. Це створює підґрунтя для подальшого розгляду вибраних гнучких підходів до управління розробкою, планування дорожньої карти продукту та побудови архітектури системи у наступних розділах.

РОЗДІЛ 2

ГНУЧКЕ УПРАВЛІННЯ СТВОРЕННЯМ СИСТЕМИ ASSETFLOW

2.1. Обґрунтування вибору гнучкого підходу управління проєктом

Система обробки замовлень «AssetFlow» створюється для використання у високодинамічному середовищі фінансових ринків, де змінюються торгові стратегії, вимоги регуляторів, біржова інфраструктура та очікування професійних учасників ринку. За таких умов класичні каскадні моделі (Waterfall) не дозволяють оперативно реагувати на зміни: значна частина вимог уточнюється вже в процесі розробки або після перших спроб дослідної експлуатації. Це узгоджується з підходами Agile-спільноти, за якими цінність створюється через ранню й безперервну поставку працюючого програмного забезпечення та готовність реагувати на зміну вимог [10].

У фінансовому секторі, зокрема на ринках капіталу, перехід до гнучких підходів вже продемонстрував підвищення адаптивності та конкурентоспроможності банків і інвестиційних компаній. Сучасні дослідження підтверджують ефективність наскрізної (end-to-end) розробки систем у гнучких середовищах саме для фінансового сектору. Використання Agile дозволяє мінімізувати ризики при створенні складних транзакційних систем та забезпечити відповідність динамічним вимогам ринку [16]. Це підтверджує доцільність застосування гнучкого управління й для проєкту створення OMS.

Сутність фреймворку Scrum. Для організації гнучкого управління в проєкті «AssetFlow» обрано фреймворк Scrum. Згідно з офіційним Посібником зі Scrum, це фреймворк для розроблення та підтримки складних продуктів, який ґрунтується на емпіричному управлінні процесами та ітераційно-інкрементальній поставці результату Scrum Guide [12].

Сутність Scrum полягає у тому, що розробка продукту організовується як послідовність коротких ітерацій (спринтів) сталої тривалості (наприклад, 2 тижні), в межах яких кросфункціональна команда створює інкремент продукту, потенційно

готовий до доставки користувачам. Scrum реалізує підхід емпіричного процес-контролю, що спирається на три ключові стовпи: прозорість, інспекція та адаптація [13].

Основними елементами фреймворку є:

- Ролі (accountabilities):
 - o *Product Owner* – відповідає за максимізацію цінності продукту й управління Product Backlog;
 - o *Scrum Master* – забезпечує розуміння й застосування Scrum, усуває перешкоди в роботі команди;
 - o *Developers* – кросфункціональна команда, що безпосередньо створює інкремент продукту.
- Артефакти:
 - o *Product Backlog* – впорядкований перелік вимог до продукту;
 - o *Sprint Backlog* – вибірка елементів беклогу та план роботи на поточний спринт;
 - o *Increment* – сукупність завершених елементів, що додають цінність продукту;
- Події (events):
 - o *Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective* – структурований набір зустрічей, що забезпечує прозорість, регулярну інспекцію результатів і адаптацію планів.

Таким чином, Scrum задає не жорсткий процес розробки, а рамки (framework), всередині яких команда може застосовувати конкретні технічні практики, обрані відповідно до домену (розробка високонавантажених фінансових систем, інтеграції з біржами, забезпечення відмовостійкості тощо).

Переваги Scrum для створення продукту AssetFlow

Вибір саме Scrum як основного фреймворку для гнучкого управління створенням системи обробки замовлень «AssetFlow» зумовлений низкою переваг, критично важливих для фінтех-продуктів:

1. Орієнтація на цінність для стейкхолдерів. Product Owner керує Product Backlog'ом, впорядковуючи вимоги за бізнес-цінністю. Це дозволяє в першу чергу реалізовувати функціонал, який створює максимальний ефект для користувачів AssetFlow: ядро обробки ордерів, ключові інтеграції з торговими майданчиками, базові механізми ризик-менеджменту.

2. Рання поставка MVP та поетапне нарощування функціональності. Завдяки коротким спринтам команда може вже на ранніх етапах побудувати та надати MVP-версію OMS для пілотної експлуатації з обмеженим набором клієнтів (наприклад, обробка лімітних і ринкових ордерів на обмеженому переліку інструментів), а далі ітераційно розширювати систему: додавати нові типи ордерів, звітність, аналітичні модулі. Це відповідає принципам Agile про «ранню й безперервну поставку цінного програмного забезпечення» [10].

3. Здатність працювати з невизначеними та змінними вимогами. Після кожного спринту Product Backlog переглядається й уточнюється з урахуванням:

- змін ринкової кон'юнктури;
- появи нових регуляторних вимог;
- зворотного зв'язку від трейдерів, ризик-менеджерів, операційної підтримки.

Це дозволяє системі AssetFlow еволюціонувати разом із середовищем, а не фіксуватися на застарілих вимогах, визначених лише на старті проєкту.

4. Раннє виявлення технічних і бізнес-ризиків. Ітераційно-інкрементальна побудова продукту дає змогу відпрацьовувати критичні елементи (продуктивність маршрутизації ордерів, стійкість до відмов, затримки на інтеграційних інтерфейсах) уже на перших релізах, а не в кінці проєкту. Практика показує, що використання Agile-підходів у фінансових і фінтех-проєктах позитивно впливає на успішність ініціатив та зниження ризиків [14].

5. Прозорість і передбачуваність процесу розробки. Використання фіксованої тривалості спринтів, оцінювання задач у story points та вимірювання

швидкості команди (velocity) створюють основу для більш точного прогнозування термінів реалізації етапів дорожньої карти AssetFlow й управління очікуваннями стейкхолдерів.

6. Фокус на командній взаємодії та безперервному вдосконаленні. Регулярні події Scrum (Daily Scrum, Sprint Review, Sprint Retrospective) стимулюють відкриту комунікацію, спільне прийняття рішень і постійний аналіз того, як покращити як продукт, так і процеси команди. Для довготривалого проєкту зі складним доменом (мультиактивна торгівля, регуляторні обмеження, висока відповідальність за помилки) це є ключовою умовою підтримки стабільної якості.

Альтернативні гнучкі підходи (Kanban, гібридні моделі) також застосовуються у фінансових організаціях, передусім для процесів підтримки й операційної діяльності. Однак у випадку AssetFlow головний акцент робиться на розробці нового продукту зі складною архітектурою та послідовними релізами. Scrum у цьому контексті надає більш чітко структурований набір ролей, подій і артефактів, що спрощує запуск команди, формування дорожньої карти та організацію планування інкрементів. Порівняння обраного підходу з альтернативами наведено в Таблиці 2.1.

Таблиця 2.1 Порівняльна характеристика підходів до управління проєктом створення AssetFlow

Критерій порівняння	Каскадна модель (Waterfall)	Гнучкий підхід (Scrum)	Перевага Scrum для AssetFlow
Реакція на зміни	Зміни дорогі та складні після затвердження ТЗ	Зміни вітаються на початку кожного спринту	Можливість швидко адаптувати систему до нових регуляторних вимог
Постачання цінності	Готовий продукт лише в кінці проєкту	Працюючий інкремент продукту кожні 2–4 тижні	Можливість раннього тестування функцій брокерами
Участь замовника	Обмежена етапом збору вимог	Постійна взаємодія та зворотний зв'язок	Зниження ризику створення незручного інтерфейсу
Контроль ризиків	Ризики стають явними на пізніх етапах	Раннє виявлення ризиків завдяки частим релізам	Швидке виявлення помилок у розрахунках чи логіці

Отже, фреймворк Scrum обрано як базовий інструмент гнучкого управління створенням системи обробки замовлень «AssetFlow», оскільки він поєднує:

- емпіричний підхід до управління складними продуктами;
- ітераційно-інкрементальну поставку цінності;
- здатність адаптуватися до змін у середовищі фінансових ринків;
- розвинуті механізми прозорості, інспекції та адаптації.

2.2. Планування процесів розробки системи (дорожня карта, ролі, реєстр вимог)

Після обґрунтування вибору гнучкого підходу на основі фреймворку Scrum у підрозділі 2.1 необхідно визначити, як саме буде організовано планування процесів розробки системи обробки замовлень «AssetFlow». На відміну від традиційних каскадних моделей, у гнучкому підході планування має багаторівневий характер:

- на стратегічному рівні формується дорожня карта розвитку продукту;
- на тактичному рівні підтримується реєстр вимог і журнал вимог до продукту;
- на операційному рівні здійснюється планування ітерацій (спринтів).

Такий підхід дозволяє поєднати довгострокове бачення розвитку системи обробки замовлень (англ. *Order Management System, OMS*) з гнучким реагуванням на зміни ринку та вимог користувачів.

2.2.1. Дорожня карта розробки системи «AssetFlow»

Дорожня карта розвитку продукту (англ. *product roadmap*) — це високорівневий план, який відображає послідовні етапи еволюції системи, основні віхи та очікувані результати для бізнесу. На відміну від детального календарного графіка, дорожня карта у гнучкому підході є «живим» документом: вона регулярно переглядається та уточнюється на основі результатів ітерацій, зворотного зв'язку стейкхолдерів і змін зовнішнього середовища. Приклади підходів до побудови дорожніх карт для гнучких продуктів наведено, зокрема, на ресурсі компанії Atlassian [15].

Основні етапи (епіки) дорожньої карти для системи «AssetFlow» такі:

– Етап «Ініціація проєкту та середовище розробки». Містить роботи підготовчого, нульового спринту: налаштування середовища розробки, створення репозиторію коду, проєкту в Jira, базових дошок та журналу вимог до продукту (Product Backlog), формування початкового реєстру вимог. Цей етап забезпечує технічну та організаційну готовність до подальшої розробки.

– Етап «Моделювання процесів та архітектура системи обробки замовлень». На цьому етапі уточнюються бізнес-цілі й вимоги до системи, описані у розділі 1; моделюються бізнес-процеси обробки замовлень «від клієнта до виконаного ордеру»; розробляється цільова архітектура системи обробки замовлень (Order Management System, OMS) з урахуванням інтеграції з біржами, брокерською інфраструктурою та внутрішніми системами фінансової установи. Епік AF-1 відповідає за завершення концептуального проєктування і переходу до реалізації.

– Етап «Клієнти та рахунки, інтеграція з Дія / BankID». Охоплює реалізацію підсистеми роботи з клієнтами та торговими рахунками: реєстрацію клієнта, відкриття й ведення рахунків, базові атрибути клієнтського профілю. Також передбачено інтеграцію з державним застосунком «Дія» та системою BankID для віддаленої ідентифікації та верифікації клієнтів (процедури «Знай свого клієнта» — *Know Your Customer, KYC*). Цей етап забезпечує юридично коректну роботу системи з реальними клієнтами.

– Етап «Ядро системи обробки замовлень». Відповідає за створення мінімально життєздатного ядра системи: реалізацію моделі ордеру, життєвого циклу заявки (створення, зміна, скасування), базової маршрутизації на один або кілька торгових майданчиків, журналювання операцій. У межах цього епіка формується перший варіант мінімально життєздатної версії системи (MVP), придатної для пілотної експлуатації в обмеженому середовищі.

– Етап «Інтеграції із зовнішніми та внутрішніми системами». Містить роботи з під'єднання системи «AssetFlow» до зовнішньої торгової інфраструктури (біржі, брокерські шлюзи) та внутрішніх систем фінансової установи (облік, звітність, управлінські системи). До цього епіку належать налаштування форматів повідомлень, механізмів обміну даними, вимог до часового

лагу й надійності передавання інформації. Фактично це етап розширення інтеграцій та підвищення корисності системи в реальному середовищі.

– Етап «Ризик-менеджмент та звітність». Передбачає впровадження контролю торгових і кредитних лімітів, базових передторгових перевірок, а також формування операційних та первинних регуляторних звітів. На цьому етапі система доповнюється механізмами управління ризиками та прозорості операцій, що є критичним для роботи на фінансових ринках.

– Етап «Інтерфейс користувача та моніторинг». Охоплює розробку інтерфейсів для трейдерів і операційних співробітників: перегляд черги ордерів, деталізацію стану заявки, фільтри й пошук, відображення повідомлень про помилки. Також сюди входить створення панелей моніторингу стану системи (черги, помилки, затримки), налаштування сповіщень для технічної та операційної підтримки.

– Етап «Аналітичні та інтелектуальні модулі». Завершальний етап дорожньої карти, який передбачає розвиток аналітичних можливостей системи: аналіз якості виконання ордерів, виявлення аномальної активності, побудову модулів підтримки прийняття рішень. Цей епік забезпечує поступовий перехід від базової функціональності OMS до інтелектуальних сервісів для учасників ринку. Впровадження інтелектуальних моделей обробки даних у бізнес-процеси дозволяє трансформувати систему з облікової в консультаційну, підвищуючи якість управлінських рішень [15]

На часовій шкалі Jira Roadmap ці епіки розміщені послідовно й частково перекриваються: підготовчі роботи та моделювання з архітектурою припадають на початковий період; далі реалізуються блоки «Клієнти та рахунки» і «Ядро системи обробки замовлень», що відповідає створенню мінімально життєздатної системи; наступні епіки відображають розширення інтеграцій, механізмів ризик-менеджменту й користувацьких інтерфейсів; останній епік – розвиток аналітичних та інтелектуальних модулів. Така структура дорожньої карти дозволяє чітко пов'язати бачення продукту з етапами реалізації в часі та надалі деталізувати їх у вигляді історій користувачів і задач у Product Backlog.

2.2.2 План виконання та терміни проєкту

Для отримання мінімально життєздатної версії продукту (MVP) системи обробки замовлень «AssetFlow» передбачено 5 спринтів, тривалістю 2 тижні кожен. Нульовий підготовчий спринт (ініціалізація середовища розробки та налаштування Jira) винесено за межі основного плану виконання та розглядається як стартовий етап.

- Кожен спринт триває два тижні: з понеділка з 9:00 до понеділка через два тижні до 14:00, оскільки огляд результатів спринту (Sprint Review) та демонстрація інкременту проводяться щопонеділка з 14:00 до 15:00.
- Планування наступного спринту (Sprint Planning) відбувається щопонеділка з 9:00 до 10:00 перед початком нового спринту.
- Перший спринт починається у понеділок 06-10-2025 о 9:00.
- Планові спринти закінчуються 15-12-2025 о 14:00 після завершення п'ятого спринту.

Розглянемо дорожню карту (roadmap) проєкту з Jira та описи цілей спринтів (Sprint Goal) для кожного з них (див. умовно рисунок 2.2 – 2.4 – Дорожня карта розробки продукту «AssetFlow» у Jira).

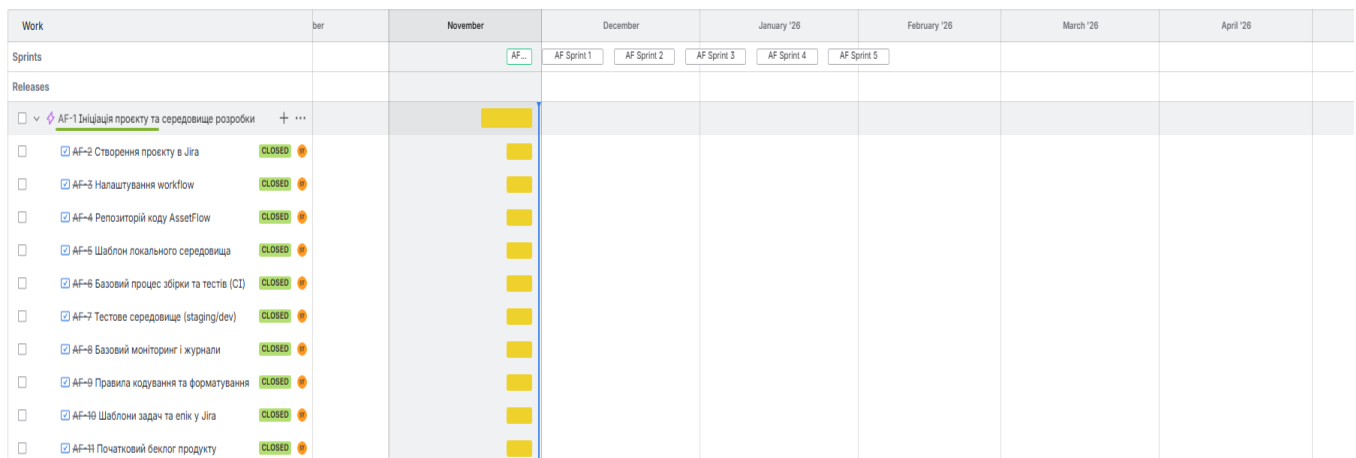


Рисунок 2.2 - Дорожня карта розробки продукту «AssetFlow» у Jira

Джерело: розроблено автором

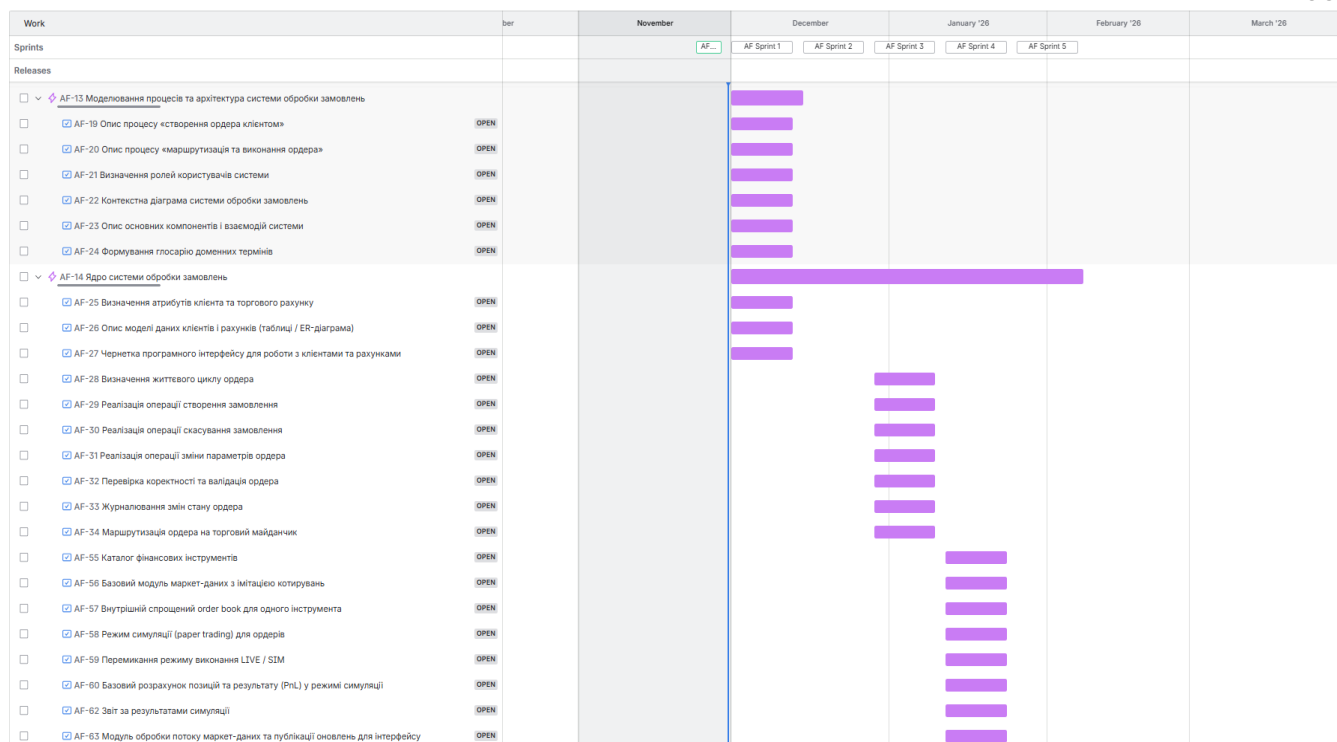


Рисунок 2.3 - Дорожня карта розробки продукту «AssetFlow» у Jira

Джерело: розроблено автором

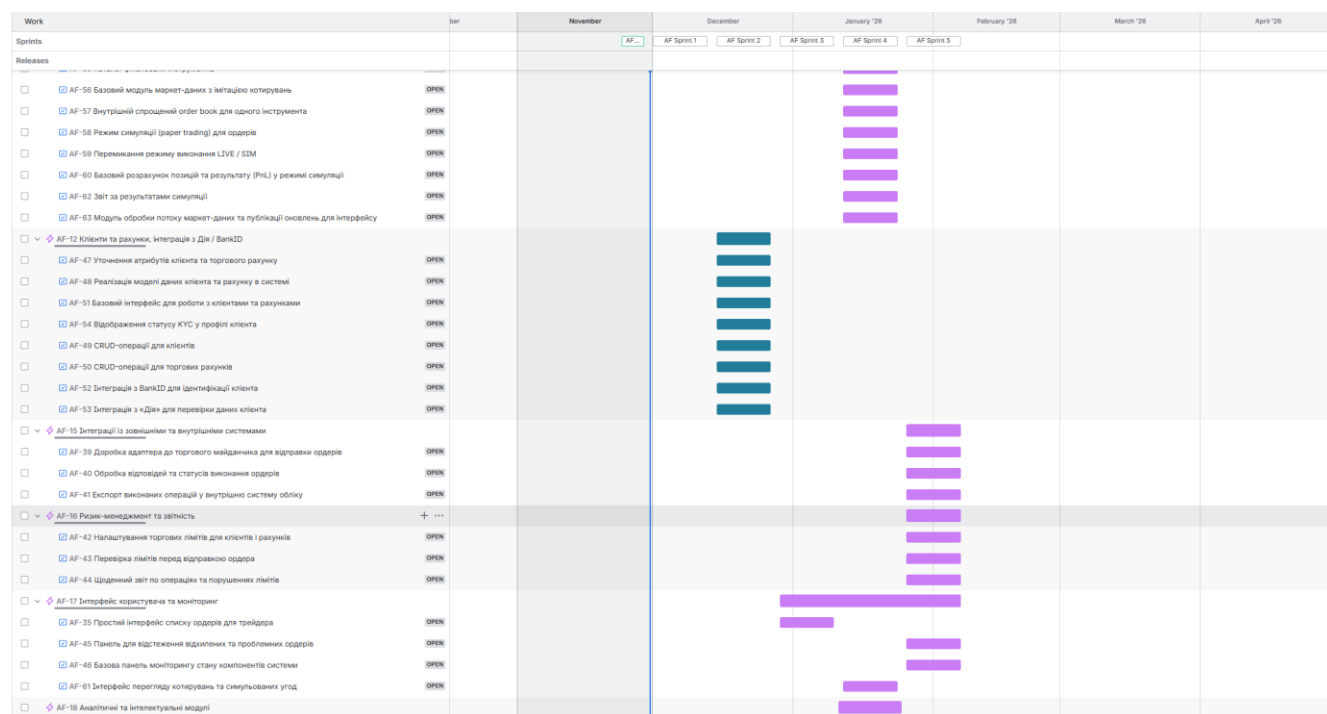


Рисунок 2.4 - Дорожня карта розробки продукту «AssetFlow» у Jira

Джерело: розроблено автором

Sprint 1, 06-10-2025|09:00 : 20-10-2025|14:00:

– Ціль спринту. Отримати погоджені бізнес-процеси обробки замовлень «від клієнта до виконаного ордера», базову архітектуру системи «AssetFlow» та первинну модель даних для клієнтів і торгових рахунків. Інкрементом спринту є пакет аналітичних артефактів (діаграми процесів, контекстна діаграма, перелік компонентів системи) і початковий опис сутностей «клієнт» та «торговий рахунок».

Sprint 2, 20-10-2025|15:00 : 03-11-2025|14:00:

– Ціль спринту. Реалізувати працездатну підсистему клієнтів і торгових рахунків із підтримкою процедур ідентифікації клієнтів (KYC) через інтеграції з сервісами BankID та «Дія». Інкрементом спринту є модуль управління клієнтами та рахунками (модель даних, CRUD-операції, внутрішній інтерфейс оператора) зі збереженням статусу проходження KYC у профілі клієнта.

Sprint 3, 03-11-2025|15:00 : 17-11-2025|14:00:

– Ціль спринту. Побудувати ядро системи обробки замовлень, яке забезпечує повний життєвий цикл ордера (створення, зміна, скасування, зміна статусів) та базовий інтерфейс для трейдера. Інкрементом спринту є працездатний прототип ядра OMS: ордери пов'язані з клієнтами та їхніми рахунками, проходять через визначені стани, журналюються основні події, а в інтерфейсі відображається список ордерів із ключовими параметрами та статусами.

Sprint 4, 17-11-2025|15:00 : 01-12-2025|14:00:

– Ціль спринту. Забезпечити базову інтеграцію системи «AssetFlow» із зовнішньою торговою інфраструктурою та внутрішніми системами обліку, а також впровадити перевірку торгових лімітів і засоби оперативного моніторингу. Інкрементом спринту є версія системи, яка вміє відправляти ордери на один торговий майданчик (або його тестовий адаптер), отримувати статуси виконання, експортувати виконані операції у внутрішню облікову систему, перевіряти ліміти перед відправкою ордера та відображати проблемні заявки й стан компонентів на панелі моніторингу.

Sprint 5, 01-12-2025|15:00 : 15-12-2025|14:00:

– Ціль спринту. Побудувати тестове ринкове середовище всередині «AssetFlow» з модулем маркет-даних і режимом симуляції виконання ордерів, щоб можна було реалістично перевіряти роботу ядра системи та торгових стратегій без підключення до реальних бірж. Інкрементом спринту є модуль маркет-даних (каталог інструментів, імітація котирувань), спрощений внутрішній order book, режим симуляції (paper trading) з перемиканням між режимами LIVE/SIM, базовий розрахунок позицій та фінансового результату (P&L) у режимі симуляції, а також інтерфейс для перегляду котирувань, симульованих угод і результатів тестування стратегії.

Представлена дорожня карта та структура спринтів відображають логіку фазування розробки MVP системи «AssetFlow» та послідовність нарощування її функціоналу: від ядра обробки замовлень до інтеграцій, звітності та інтерфейсів користувачів. Вона слугує основою для планування навантаження Scrum-команди та подальшого коригування підходу до гнучкого управління створенням системи на основі фактичних результатів спринтів.

2.2.3. Бюджет проєкту (фаза розробки MVP системи AssetFlow)

Метою цього підрозділу є оцінка витрат на фазу розробки MVP системи «AssetFlow» за заданою дорожньою картою та структурою спринтів. Розрахунок бюджету дозволяє перевірити реалістичність прийнятих рішень щодо складу команди, тривалості спринтів і обсягу робіт, а також слугує основою для порівняння планових показників із фактичними витратами часу на етапі створення MVP.

2.2.3.1. Вихідні припущення для розрахунків

У розрахунках прийнято такі припущення:

- Тривалість фази MVP:
- 5 спринтів × 2 тижні = 10 тижнів.
- Робоче навантаження однієї штатної одиниці (1 FTE):
10 тижнів × 5 робочих днів × 8 год/день = 400 годин.

- Склад Scrum-команди на фазі MVP:
 - Власник продукту (Product Owner) – 0,5 ставки (часткова зайнятість).
 - Скрам-майстер – 0,5 ставки.
 - Розробники (2 особи) – 2,0 ставки (повна зайнятість).
 - Тестувальник (QA) – 0,5 ставки.
 - UX/UI-дизайнер – 0,25 ставки.
 - DevOps-інженер – 0,5 ставки (налаштування середовищ, CI/CD, моніторинг).
- Умовні середні погодинні ставки:
 - Product Owner – 17 USD/год.
 - Скрам-майстер – 15 USD/год.
 - Розробник – 20 USD/год.
 - Тестувальник (QA) – 15 USD/год.
 - UX/UI-дизайнер – 15 USD/год.
 - DevOps-інженер – 20 USD/год.

Валюта розрахунків – долари США (USD), що спрощує подальші порівняння.

Таблиця 2.2 – Трудові витрати команди на фазу розробки MVP AssetFlow

Роль	Частка зайнятості	Загальна кількість годин	Примітка
Product Owner	0,5 FTE	$400 \times 0,5 = 200$ год	Беклог, пріоритезація, валідація вимог
Скрам-майстер	0,5 FTE	200 год	Події Scrum, координація команди
Розробники (2 особи)	2,0 FTE	$400 \times 2 = 800$ год	Back-end, front-end, технічна логіка
Тестувальник (QA)	0,5 FTE	200 год	Планування та виконання тестів
UX/UI-дизайнер	0,25 FTE	$400 \times 0,25 = 100$ год	Прототипи та дизайн основних екранів
DevOps-інженер	0,5 FTE	200 год	CI/CD, середовища, моніторинг, інфраструктурні скрипти
Разом	–	1 700 год	–

2.2.3.2. Прямі витрати на оплату праці

Прямі витрати на оплату праці визначаються як добуток кількості годин і погодинної ставки для кожної ролі.

Таблиця 2.3 – Розрахунок прямих витрат на оплату праці команди

Роль	Години	Ставка, USD/год	Вартість, USD
Product Owner	200	17	$200 \times 17 = 3\,400$
Скрам-майстер	200	15	$200 \times 15 = 3\,000$
Розробники (2 особи)	800	20	$800 \times 20 = 16\,000$
Тестувальник (QA)	200	15	$200 \times 15 = 3\,000$
UX/UI-дизайнер	100	15	$100 \times 15 = 1\,500$
DevOps-інженер	200	20	$200 \times 20 = 4\,000$
Разом пряма зарплата	–	–	30 900 USD

Для урахування податків, адміністративних і накладних витрат (HR, бухгалтерія, офісна інфраструктура тощо) застосовано умовний коефіцієнт накладних витрат 20 % від суми прямих витрат:

$$\text{Накладні витрати} = 30\,900 \times 0,20 = 6\,180 \text{ USD.}$$

Тоді загальні витрати на персонал:

$$30\,900 + 6\,180 = 37\,080 \text{ USD.}$$

2.2.3.3. Інфраструктурні та програмні витрати

Окрім витрат на команду, для фази розробки MVP потрібні інфраструктурні ресурси та інструменти:

1. Хмарна інфраструктура / серверні ресурси

(середовище розробки, тестове середовище, CI/CD):

$$\text{умовно } 400 \text{ USD/місяць} \times 3 \text{ місяці} \approx 1\,200 \text{ USD.}$$

2. Системи управління проектом та документацією

(Jira, Confluence чи аналогічні сервіси):

$$10 \text{ USD/користувача/місяць} \times 7 \text{ користувачів} \times 3 \text{ місяці} = 210 \text{ USD.}$$

3. Хостинг репозиторіїв коду та CI/CD-сервіси (платні тарифи GitLab/GitHub, runner-и тощо):
приблизно 50 USD/місяць × 3 місяці = 150 USD.
4. Моніторинг, логування, допоміжні сервіси (умовні витрати на APM/лог-сервіси та ін.):
60 USD/місяць × 3 місяці = 180 USD.

Таблиця 2.4 – Інфраструктурні та програмні витрати на фазу MVP

Стаття витрат	Орієнтовна сума, USD
Хмарна інфраструктура / сервери (Dev + Test)	1 200
Jira/Confluence (проектний менеджмент, документація)	210
CI/CD, хостинг репозиторіїв коду	150
Моніторинг, логування, допоміжні сервіси	180
Разом інфраструктура й ПЗ	1 740

2.2.3.4. Резерв на ризики та підсумкова оцінка бюджету

З огляду на можливі ризики (уточнення вимог у процесі роботи, додаткові доопрацювання за результатами тестування, технічні затримки) доцільно передбачити резерв на непередбачені витрати. Для фазового MVP-етапу умовно приймемо резерв у розмірі 1 000 USD.

Тоді орієнтовний бюджет фази розробки MVP системи AssetFlow становить:

- Витрати на персонал (з накладними): 37 080 USD
- Інфраструктура та програмне забезпечення: 1 740 USD
- Резерв на ризики: 1 000 USD

Разом:

$$37\,080 + 1\,740 + 1\,000 = 39\,820 \text{ USD,}$$

що для подальших розрахунків може бути округлено до 40 000 USD.

Таким чином, розрахований бюджет відображає витрати на фазу розробки MVP (5 спринтів по 2 тижні), яка є об'єктом моделювання в цій кваліфікаційній роботі. Повномасштабне промислове впровадження AssetFlow вимагатиме

додаткових інвестицій у подальші релізи, розширення функціональності, інтеграції з новими контрагентами та експлуатаційну підтримку, що виходить за межі даного дослідження.

2.2.4. Команда проєкту та ролі в Scrum

Для фази розробки MVP системи AssetFlow використовується одна крос-функційна Scrum-команда, яка об'єднує фахівців із різними компетенціями та спільно відповідає за створення інкременту продукту в кожному спринті. Такий підхід відповідає принципам гнучкого управління, оскільки забезпечує саморганізацію, швидкий обмін знаннями та мінімізацію залежностей між окремими підрозділами.

Scrum-команда складається з трьох груп ролей: Власник продукту (Product Owner), Скрам-майстер (Scrum Master) та Команда розробки (Developers), до якої входять розробники, тестувальник, DevOps-інженер і UX/UI-дизайнер.

2.2.4.1. Склад Scrum-команди на фазі MVP

Склад команди узгоджено з бюджетними розрахунками для п'яти спринтів тривалістю два тижні. Ролі та орієнтовна зайнятість наведені в таблиці 2.5.

Таблиця 2.5 – Склад Scrum-команди проєкту AssetFlow на фазі розробки MVP

Роль / компетенція	Належність у Scrum	Кількість осіб / зайнятість	Основні функції в проєкті
Власник продукту (Product Owner)	Окрема роль Scrum	0,5 FTE	Формування бачення продукту; ведення та пріоритезація беклогу; постановка цілей спринтів; приймання результатів робіт.
Скрам-майстер (Scrum Master)	Окрема роль Scrum	0,5 FTE	Фасилітація подій Scrum; усунення перешкод; коучинг команди щодо принципів гнучкого управління; захист команди від зовнішніх відволікань.
Розробники (Software Developers)	Команда розробки (Developers)	2,0 FTE	Реалізація функціоналу бекенду й фронтенду; участь в оцінюванні й декомпозиції користувацьких історій; технічний дизайн.

Роль / компетенція	Належність у Scrum	Кількість осіб / зайнятість	Основні функції в проєкті
Тестувальник (QA-інженер)	Команда розробки (Developers)	0,5 FTE	Планування та виконання тестування; підготовка тест-кейсів; регресійні перевірки перед релізами інкременту.
DevOps-інженер	Команда розробки (Developers)	0,5 FTE	Налаштування середовищ (Dev/Test); CI/CD-пайплайни; базовий моніторинг і логуювання; підтримка інфраструктури MVP.
UX/UI-дизайнер	Команда розробки (Developers)	0,25 FTE	Проектування інтерфейсів кабінету користувача та робочих місць операторів; підготовка макетів і прототипів.

Усі технічні ролі (розробники, QA, DevOps, UX/UI) в межах Scrum розглядаються як єдина команда розробки (Developers), яка колективно несе відповідальність за готовий інкремент продукту наприкінці кожного спринту.

2.2.4.2. Ролі та зона відповідальності

Таблиця 2.6 – Ролі Scrum-команди проєкту AssetFlow та зона відповідальності

Роль	Короткий опис ролі	Основні зони відповідальності
Власник продукту (Product Owner)	Представник стейкхолдерів, відповідає за цінність продукту та зміст Product Backlog	Формування й підтримання Product Backlog в актуальному та прозорому стані; Опис і пріоритезація користувацьких історій відповідно до цілей MVP; Визначення критеріїв приймання (acceptance criteria) та очікуваних результатів спринту; Взаємодія зі стейкхолдерами (брокер, банк, ризик-менеджмент, IT-підрозділи) та відображення їхніх потреб у беклозі; Приймання інкрементів на Sprint Review та ухвалення рішення щодо готовності функціоналу.
Скрам-майстер (Scrum Master)	Сервісний лідер команди, відповідальний за коректне застосування Scrum	Фасилітація ключових подій: Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective; Допомога Product Owner у формулюванні цілей спринтів і підготовці беклогу до планування; Виявлення та усунення перешкод (impediments), що заважають команді виконувати запланований обсяг робіт; Підтримка культури безперервного вдосконалення (інспекція та адаптація на ретроспективах); Навчання команди принципам гнучкого управління та самоорганізації.

Роль	Короткий опис ролі	Основні зони відповідальності
Команда розробки (Developers)	Крос-функційна команда, що створює інкремент продукту (розробники, QA, DevOps, UX/UI)	Спільна оцінка користувацьких історій і декомпозиція їх на технічні підзадачі; Розробка, тестування та інтеграція функціоналу, необхідного для досягнення цілей спринту; Забезпечення належної якості інкременту (автотести, код-рев'ю, базові практики CI/CD); Підтримка актуальності технічної документації та артефактів конфігурації.
DevOps-інженер (член команди розробки)	Відповідає за інфраструктуру, середовища й автоматизацію поставки	Налаштування процесів безперервної інтеграції та доставки (CI/CD); Уніфікація середовищ розробки й тестування (Dev/Test); Забезпечення базового моніторингу й логування працездатності компонентів AssetFlow.
Тестувальник (QA) (член команди розробки)	Забезпечує системний підхід до контролю якості продукту	Підготовка тест-кейсів і сценаріїв тестування; Проведення функціонального, інтеграційного й регресійного тестування інкременту; Фіксація дефектів і взаємодія з розробниками щодо їх усунення.
UX/UI-дизайнер (член команди розробки)	Проектує користувацькі інтерфейси та сценарії взаємодії	Проектування прототипів інтерфейсу кабінету клієнта та робочих місць операторів; Уточнення вимог до UX спільно з Product Owner та стейкхолдерами; Забезпечення зрозумілості й зручності ключових сценаріїв для цільових сегментів користувачів.

2.2.5. Аналіз зацікавлених сторін (Stakeholder Mapping)

Ефективна реалізація MVP системи AssetFlow потребує врахування інтересів і впливу різних груп зацікавлених сторін (стейкхолдерів): внутрішніх підрозділів брокера/банку, кінцевих користувачів, регуляторів, бірж, інфраструктурних організацій та IT-партнерів. Stakeholder Mapping дає змогу:

- ідентифікувати ключові групи стейкхолдерів;
- оцінити їх вплив на проєкт та рівень зацікавленості;
- визначити стратегію взаємодії з кожною групою.

2.2.5.1. Ідентифікація основних стейкхолдерів

У таблиці 2.7 наведено основні групи зацікавлених сторін для проєкту AssetFlow, їхній тип, інтереси, рівень впливу на результати проєкту та бажаний формат залучення.

Таблиця 2.7 – Основні стейкхолдери проєкту AssetFlow та їх інтереси

ID	Стейкхолдер / група	Тип (внутрішній / зовнішній)	Основний інтерес щодо AssetFlow	Рівень впливу	Рівень зацікавленості	Бажаний формат залучення
S1	Топ-менеджмент компанії-ініціатора (керівництво брокера/банку)	Внутрішній	Отримання робочого MVP, що підвищує конкурентоспроможність та ефективність операцій	Високий	Високий	Спільне прийняття рішень, регулярні звіти, Steering-комітет
S2	Власник продукту AssetFlow (Product Owner)	Внутрішній	Максимізація цінності продукту для цільових сегментів ринку, відповідність баченням стейкхолдерів	Високий	Високий	Активна участь, постійна комунікація з усіма групами
S3	Scrum-команда розробки (розробники, QA, DevOps, UX/UI)	Внутрішній	Успішна реалізація MVP у межах спринтів, технічна якість, стійка архітектура	Середній	Високий	Щоденна операційна взаємодія, внутрішні зустрічі Scrum
S4	Оператори фронт-офісу (оператори брокера/банку)	Внутрішній / бізнес-користувачі	Зручне робоче місце для введення та контролю заявок, мінімум ручної роботи	Середній	Високий	Спільна розробка вимог, інтерв'ю, юзабіліті-тести
S5	Співробітники бек-офісу	Внутрішній / бізнес-користувачі	Автоматизація післяторгових процедур, звіряння, формування реєстрів і звітів	Середній	Середній	Консультації, уточнення вимог, участь у прийманні функціоналу
S6	Підрозділ ризик-менеджменту	Внутрішній	Наявність лімітів, контролів, звітів по ризиках, прозорий аудиторний слід	Високий	Високий	Спільне формування правил і лімітів, погодження ключових рішень
S7	Підрозділ IT-безпеки та комплаєнсу	Внутрішній	Відповідність вимогам безпеки, захист даних, відповідність внутрішнім політикам та нормам	Високий	Середній	Консультації, review архітектури та рішень
S8	Приватні клієнти-інвестори	Зовнішній	Зручний доступ до портфеля, швидке відображення операцій, прозорі звіти	Низький	Високий	Опосередкований зворотний зв'язок через брокера, опитування

ID	Стейкголдери / група	Тип (внутрішній / зовнішній)	Основний інтерес щодо AssetFlow	Рівень впливу	Рівень зацікавленості	Бажаний формат залучення
S9	Інституційні інвестори (фонди, КУА, страхові, банки)	Зовнішній	Можливість роботи з великими портфелями, надійність, можливості інтеграції з їхніми системами	Середній	Високий	Пілотні впровадження, демо, спільне планування розширень
S10	Регулятори (НКЦПФР, НБУ)	Зовнішній	Коректність регуляторної звітності, прозорість операцій, контроль ринкових ризиків	Високий	Середній / Високий	Погодження форматів звітності, дотримання нормативних вимог
S11	Фондові біржі та організатори торгів	Зовнішній	Стабільне технічне підключення, коректна робота торговельних інтерфейсів	Середній	Середній	Технічні інтеграційні проекти, узгодження протоколів
S12	Депозитарні та клірингові установи	Зовнішній	Коректний обмін даними для розрахунків та обліку прав власності	Середній	Середній	Узгодження форматів повідомлень, тестові обміни даними
S13	Постачальники ринкових даних	Зовнішній	Стабільне споживання їхніх сервісів, дотримання умов ліцензування	Низький	Середній	Технічна інтеграція, супровід договорів
S14	ІТ-партнери та інтегратори	Зовнішній	Можливість впровадження AssetFlow у своїх клієнтів, отримання дохідних проектів	Низький / Середній	Середній	Партнерські угоди, спільні впровадження

2.2.5.2. Матриця «вплив – зацікавленість» (Stakeholder Mapping)

Для наочності аналізу доцільно представити стейкголдерів на матриці «рівень впливу / рівень зацікавленості». Це дозволяє визначити, кого необхідно:

- тісно залучати до ухвалення рішень,
- кого забезпечувати підтримку
- кого регулярно інформувати,
- за ким достатньо спостерігати.

У квадраті «високий вплив – висока зацікавленість» (Manage Closely) розміщуються стейкхолдери S1 (топ-менеджмент), S2 (Product Owner), S6 (ризик-менеджмент), які залучаються до ключових рішень щодо вимог і пріоритизації беклогу. Стейкхолдери з високим впливом, але нижчою зацікавленістю (S10 – регулятори) відносяться до групи «Keep Satisfied» тощо.



Рисунок 2.5 - «Карта стейкхолдерів проекту AssetFlow»

Джерело: розроблено автором

2.2.6. Структура декомпозиції робіт (Work Breakdown Structure, WBS)

Для планування змісту фази розробки MVP системи AssetFlow використано структуру декомпозиції робіт (Work Breakdown Structure, WBS). WBS відображає, що саме потрібно зробити в межах проекту, у вигляді ієрархічного дерева: від цілого проекту до окремих пакетів робіт, які можуть бути призначені команді в спринтах.

На рівні 1 виділяється проект розробки MVP AssetFlow, на рівні 2 – основні блоки робіт, що відповідають дорожній карті, на рівні 3 – конкретні пакети робіт (work packages), які стають основою для формування Product Backlog та Sprint Backlog.

Таблиця 2.8 – Ієрархічна структура робіт (WBS) фази розробки MVP системи

AssetFlow

Код WBS	Рівень	Назва роботи	Короткий зміст / результат (deliverable)
1.0	1	Проект розробки MVP системи AssetFlow	Фаза створення мінімально життєздатної версії (MVP) OMS-платформи
1.1	2	Ініціація та організація робіт	Налаштований репозиторій, Jira-проект, базові правила роботи Scrum
1.1.1	3	Налаштування середовищ Dev/Test	Базова інфраструктура для розробки та тестування
1.1.2	3	Початкове заповнення Product Backlog	Початковий список епіків та користувацьких історій для MVP
1.2	2	Аналіз та проектування	Опис цільових процесів та архітектури системи
1.2.1	3	Уточнення бізнес-процесів та сценаріїв (Use Case)	Оновлені діаграми процесів і сценарії використання AssetFlow
1.2.2	3	Проектування архітектури MVP	Контекстна та компонентна схеми, технічні рішення для ядра OMS
1.3	2	Модуль «Клієнти та рахунки»	Базовий функціонал управління клієнтами та їхніми інвестиційними рахунками
1.3.1	3	Профілі клієнтів і рахунки	CRUD-операції, зберігання реквізитів клієнтів і рахунків
1.3.2	3	Підтримка дистанційної ідентифікації (Дія/BankID)	Інтеграція для отримання даних клієнта та фіксації джерела даних
1.4	2	Ядро OMS: управління торговими заявками	Реалізація основних сценаріїв роботи із заявками
1.4.1	3	Реєстрація та зміна торгових заявок	Створення, редагування, скасування заявок (manual + API)
1.4.2	3	Контроль лімітів та доступності активів	Автоматичні перевірки коштів, паперів, внутрішніх обмежень
1.4.3	3	Маршрутизація заявок на біржу	Логіка вибору майданчика та підготовка повідомлень для торгового шлюзу
1.5	2	Інтеграція з біржею та післяторговою інфраструктурою	Базова інтеграція для виконання угод і отримання результатів
1.5.1	3	Підключення до біржової системи (FIX/REST/WebSocket)	Канал обміну заявками та повідомленнями про виконання
1.5.2	3	Обмін даними з кліринговою/депозитарною системою	Передача інструкцій на розрахунки та отримання підтверджень
1.6	2	Звітність та облік	Базові звіти для клієнтів та внутрішнього обліку
1.6.1	3	Клієнтські виписки	Формування виписок за рахунком за період
1.6.2	3	Внутрішні реєстри операцій	Реєстр угод для бек-офісу та подальших розрахунків
1.7	2	Забезпечення якості та тестування	Організований процес контролю якості MVP

Код WBS	Рівень	Назва роботи	Короткий зміст / результат (deliverable)
1.7.1	3	Підготовка тест-кейсів і сценаріїв	Набір тестових сценаріїв для ключових користувацьких історій
1.7.2	3	Функціональне та інтеграційне тестування	Протоколи тестування, перелік дефектів та результати їх усунення
1.8	2	DevOps та CI/CD	Автоматизація збірки, тестування та розгортання
1.8.1	3	Налаштування пайплайнів CI/CD	Автоматичний білд, запуск тестів, деплой на Test
1.8.2	3	Моніторинг та журналювання	Базові дашборди й логи для спостереження за інфраструктурою
1.9	2	Демонстрація та приймання MVP	Підготовка й проведення демонстрації стейкхолдерам
1.9.1	3	Підготовка демонстраційних сценаріїв	Набір сценаріїв для показу ключових можливостей AssetFlow
1.9.2	3	Сесія приймання MVP та збір зворотного зв'язку	Протокол Sprint Review / демо та список запланованих поліпшень

2.2.7. Product Backlog фази розробки MVP системи AssetFlow

Product Backlog системи AssetFlow формується як впорядкований список робіт, необхідних для реалізації мінімально життєздатної версії (MVP) системи обробки замовлень.

У практичній частині проєкту беклог ведеться в середовищі Jira Software у вигляді:

- епіків (Epic), що відповідають основним блокам робіт та елементам WBS;
- користувацьких історій і технічних задач (Story/Task), прив'язаних до відповідних епіків і запланованих спринтів (AF Sprint 0 – AF Sprint 5).

У цій роботі Product Backlog подається у вигляді узагальнених таблиць, а на окремих рисунках демонструється фрагмент беклогу безпосередньо в інтерфейсі Jira Software.

2.2.7.1. Епіки Product Backlog

Епіки задають основні блоки робіт фази MVP і напряму відповідають елементам WBS (підрозділ 2.2.6): ініціація, моделювання процесів, ядро системи обробки замовлень, робота з клієнтами та рахунками, інтеграції, ризик-менеджмент,

інтерфейси та моніторинг, а також окремий епік для подальшого розвитку аналітичних модулів.

Таблиця 2.9 – Епіки Product Backlog фази розробки MVP системи AssetFlow

Key	Summary
AF-1	Ініціація проєкту та середовище розробки
AF-13	Моделювання процесів та архітектура системи обробки замовлень
AF-14	Ядро системи обробки замовлень
AF-12	Клієнти та рахунки, інтеграція з Дія / BankID
AF-15	Інтеграції із зовнішніми та внутрішніми системами
AF-16	Ризик-менеджмент та звітність
AF-17	Інтерфейс користувача та моніторинг
AF-18	Аналітичні та інтелектуальні модулі

Примітка: епік AF-18 виходить за межі фази MVP і планується до реалізації в наступних хвилях розвитку продукту.

2.2.7.2. Фрагмент Product Backlog (історії та задачі)

Користувацькі історії та технічні задачі деталізують епіки та розподіляються між спринтами AF Sprint 0 – AF Sprint 5. У Jira Software для кожного елемента Product Backlog задаються ідентифікатор (Key), батьківський епік (Parent), короткий опис (Summary) та спринт (Sprint), що забезпечує прозоре планування робіт фази розробки MVP системи AssetFlow.

AF Sprint 0 охоплює підготовку інфраструктури розробки: створення проєкту в Jira, налаштування workflow, репозиторію коду, шаблонів середовища, базового CI, тестового середовища, моніторингу, правил кодування та початкового беклогу. Наступні спринти послідовно фокусуються на: формалізації вимог та архітектури системи (AF Sprint 1), моделюванні даних, KYC та довідників (AF Sprint 2), реалізації CRUD-операцій з ордерами та базового інтерфейсу трейдера (AF Sprint 3), запуску режиму симуляції та розрахунку PnL (AF Sprint 4), а також інтеграції з торговим майданчиком, впровадженні торгових лімітів, звітності та панелей

моніторингу (AF Sprint 5). Таким чином, беклог відображає еволюцію від налаштування середовища до отримання функціонального MVP OMS.

На рисунку 2.Z наведено фрагмент екрана Backlog у системі Jira Software, де відображаються епіки та задачі, що складають Product Backlog фази розробки MVP системи AssetFlow.

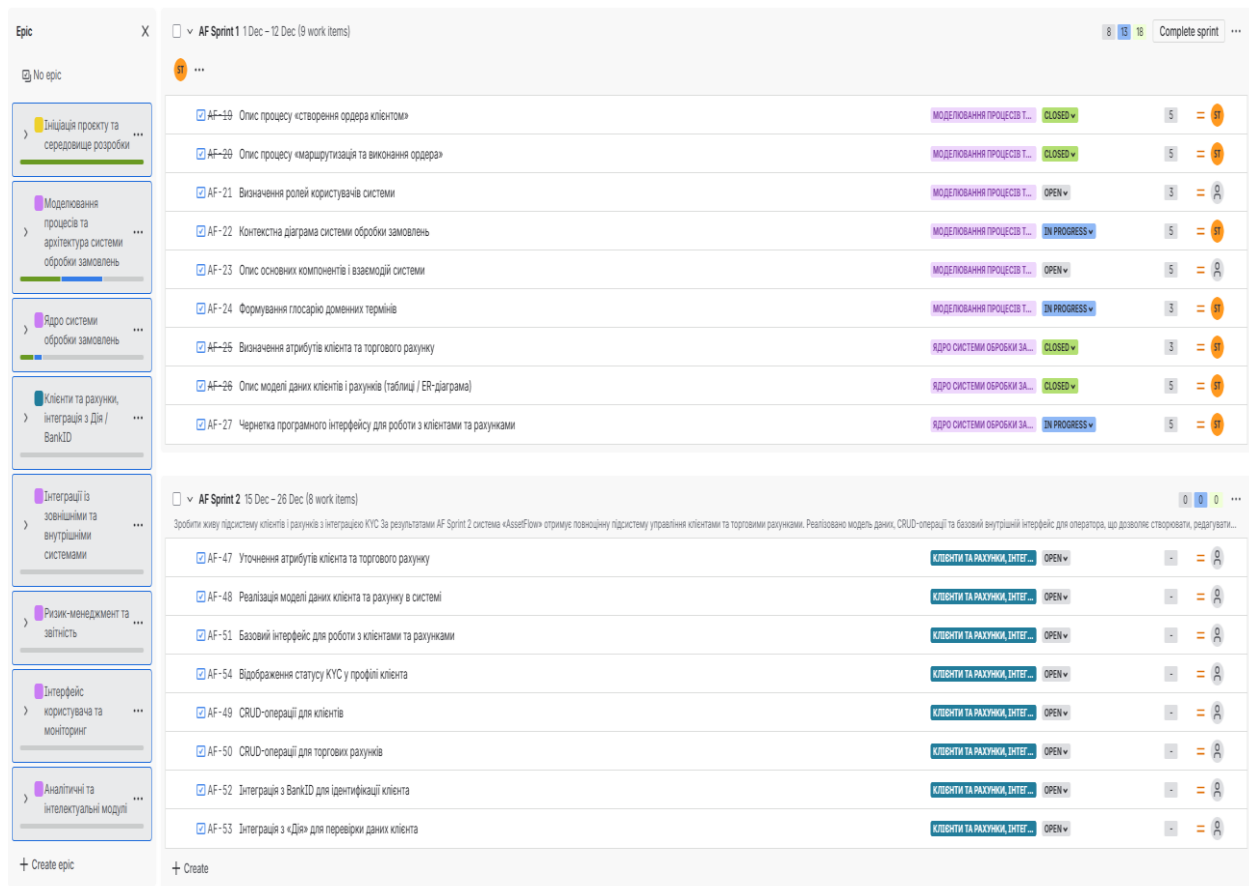


Рисунок 2.6 – Фрагмент Product Backlog системи AssetFlow у середовищі Jira Software

Джерело: розроблено автором

2.3 Моніторинг виконання проєкту

Моніторинг виконання робіт у проєкті зі створення системи обробки замовлень «AssetFlow» є центральним елементом гнучкого управління, оскільки дозволяє не лише відстежувати ступінь готовності окремих модулів, а й оцінювати

стабільність доставки цінності для кінцевих користувачів – брокерів, інвесторів та операційного персоналу.

На відміну від традиційних каскадних підходів, де контроль здійснюється шляхом порівняння факту з жорстким календарним планом, у проєкті «AssetFlow» імплементовано адаптивний моніторинг. Він базується на емпіричних даних з Jira, візуалізації ключових метрик (Burndown, Velocity) та ітеративному моделюванні ходу робіт у межах коротких спринтів.

У межах розробки OMS «AssetFlow» моніторинг дозволяє:

- прозоро відслідковувати прогрес по епіках («Клієнти та рахунки», «Торгові операції», «Ринкові дані та аналітика», «Звітність і комплаєнс», «Технічна платформа»);
- контролювати виконання спринтів та дотримання взятих командою зобов'язань;
- виявляти місця концентрації ризиків (інтеграції, складні бізнес-процеси, регуляторні вимоги);
- на основі фактичної швидкості команди планувати релізи та побудову MVP.

2.3.1 Сутність гнучкого моніторингу у проєкті «AssetFlow»

Гнучкий моніторинг у проєкті розглядається як безперервний цикл «планування – виконання – інспекція – адаптація», що реалізується в рамках кожного спринту. Його ключові принципи:

- прозорість. Усі учасники (Product Owner, Scrum-майстер, команда розробки, стейкхолдери з боку бізнесу) мають доступ до актуальної інформації щодо стану задач у Jira: які історії виконуються, які блокуються зовнішніми залежностями, які вже створюють інкремент продукту (наприклад, повністю реалізований сценарій створення клієнта та відкриття рахунку).
- вимірюваність. Для кожної історії встановлюються оцінки у story points, а прогрес вимірюється не у відсотках «готовності», а в кількості завершених одиниць

цінності. Це дозволяє об'єктивно оцінювати навантаження, переглядати реалістичність планів та порівнювати спринти між собою.

– орієнтація на бізнес-цінність. Моніторинг зосереджується на завершенні функціонально завершених історій («end-to-end» сценаріїв), а не на часткових технічних активностях. Наприклад, створення таблиці в базі даних або окремого API-методу фіксується як проміжний крок, а в центрі уваги – чи може брокер фактично провести торгову операцію в системі.

– короткий цикл зворотного зв'язку. Після кожного спринту команда не лише демонструє інкремент, а й аналізує метрики Burndown і Velocity, що дає змогу коригувати підходи до планування, декомпозиції задач та управління залежностями.

Таким чином, гнучкий моніторинг у «AssetFlow» реалізує безперервний цикл «планування – виконання – інспекція – адаптація». З теоретичної точки зору це відповідає циклу Демінга PDCA (Plan–Do–Check–Act), адаптованому до контексту Scrum:

- Plan – формування та пріоритизація продуктового й спринт-беклогу, оцінювання історій у story points під час Sprint Planning;
- Do – виконання задач у межах спринту, щоденні стендапи, розробка, тестування та інтеграція інкременту OMS «AssetFlow»;
- Check – перевірка результатів на Sprint Review, аналіз метрик Burndown та Velocity, виявлення відхилень і блокерів;
- Act – оновлення беклогу, перегляд планування та підходів до оцінювання, корекція процесів за підсумками Sprint Retrospective.

У такій інтерпретації моніторинг не зводиться до формального звітування, а стає механізмом циклічного навчання команди: кожен спринт проходить повний PDCA-цикл, завдяки чому поступово зростає зрілість процесу розробки та якість керування проєктом.

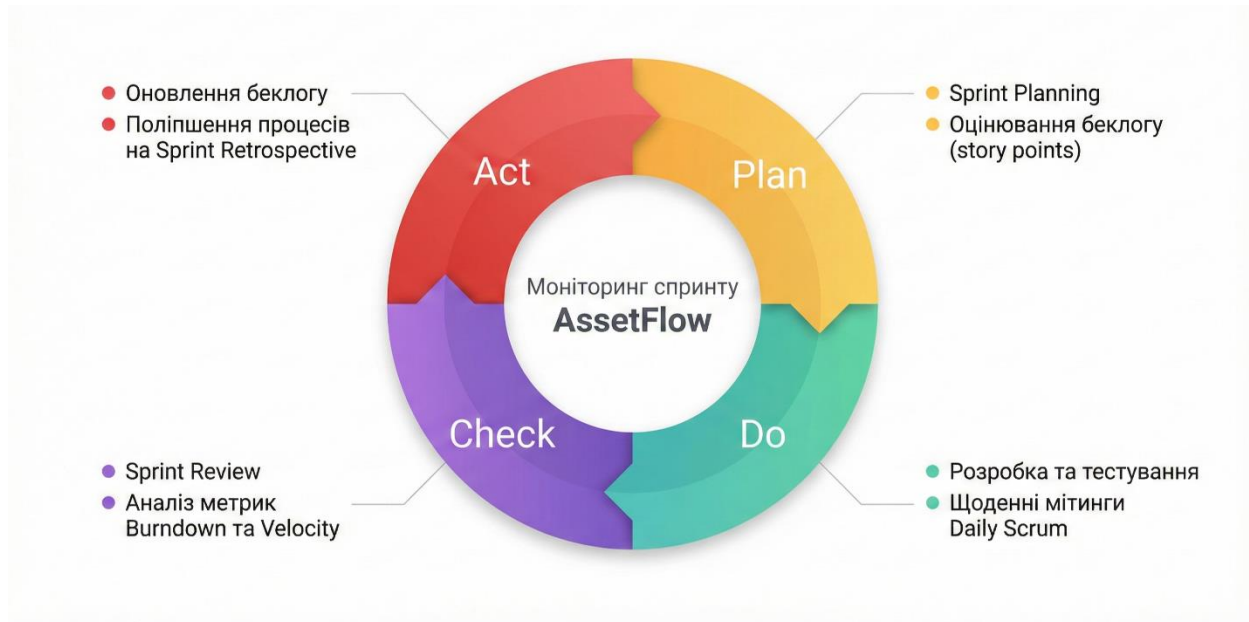


Рисунок 2.7 – Цикл Демінга PDCA в організації моніторингу спринтів проєкту «AssetFlow»

Джерело: розроблено автором

2.3.2 Організація моніторингу на рівні спринтів

Розробка OMS «AssetFlow» виконується кросфункціональною командою за Scrum-підходом. Структура робіт відображається в Jira у вигляді продуктового беклогу, згрупованого за епіками, та послідовності спринтів фіксованої тривалості.

У спринтах виконуються такі основні типи робіт:

- реалізація бізнес-функціоналу: onboarding клієнта, відкриття рахунків, створення та обробка торгових заявок, облік операцій, формування звітів для регулятора;
- розробка інтеграцій: підключення до джерел ринкових даних, налагодження обміну з зовнішніми сервісами (KYC, платіжні шлюзи тощо);
- технічні задачі: налаштування середовищ, логування, моніторинг продуктивності, оптимізація бази даних.

Кожна історія в беклозі спринту має:

- прив'язку до конкретного епіку (для забезпечення трасованості від вимоги до реалізації);
- оцінку у story points;

- статус (To Do, In Progress, Code Review, Testing, Done);
- відповідального виконавця.

Моніторинг виконується на кількох рівнях:

- щоденні стендапи – оперативний контроль прогресу та блокерів;
- візуальний контроль через дошку Jira та Burndown-графік спринту;
- аналітичний огляд за результатами спринту (Sprint Review і Retrospective), де розглядаються фактична Velocity, відхилення від планів і причини таких відхилень.

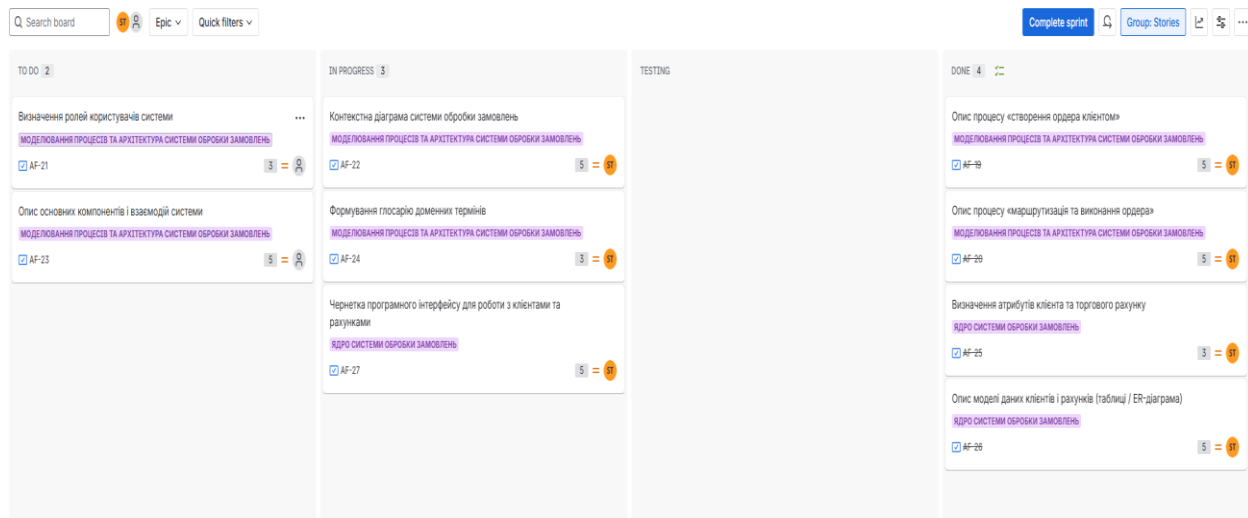


Рисунок 2.7 – Приклад дошки спринту проєкту «AssetFlow» у Jira.

Джерело: розроблено автором

2.3.3 Використання Jira як центра моніторингу

Jira виконує роль єдиного «джерела правди» щодо стану проєкту. У ній формуються всі основні артефакти моніторингу:

- продуктовий беклог та беклог спринту;
- дошка спринту зі статусами задач;
- автоматично згенеровані звіти, зокрема Burndown Chart для кожного спринту та звіт Velocity, який показує сумарну кількість story points, завершених у спринті.

Саме ці вбудовані звіти Jira використовуються як основа для кількісного аналізу перебігу спринтів у проєкті «AssetFlow».

2.3.4 Burndown Chart як основний інструмент контролю спринтів

Burndown Chart застосовується для щоденного контролю виконання плану спринту. По осі X відкладаються календарні дні спринту, по осі Y – обсяг роботи, що залишився до виконання (у story points). Сіра діагональна лінія Guideline показує ідеальну траєкторію «згоряння» задач, а червона ламана Remaining Values відображає фактичний залишок роботи.

У проєкті «AssetFlow» реальний Burndown-графік було сформовано для спринту AF Sprint 1 (рисунок 2.3.2). На початку спринту сумарний обсяг робіт становив 39 story points. На графіку видно, що команда працювала достатньо рівномірно: у перші дні залишок роботи поступово зменшувався, у вихідні спостерігається коротке плато, після чого прогрес відновлюється. До завершення спринту фактична крива практично збігається з ідеальною, тобто всі 39 story points були виконані в межах спринту.

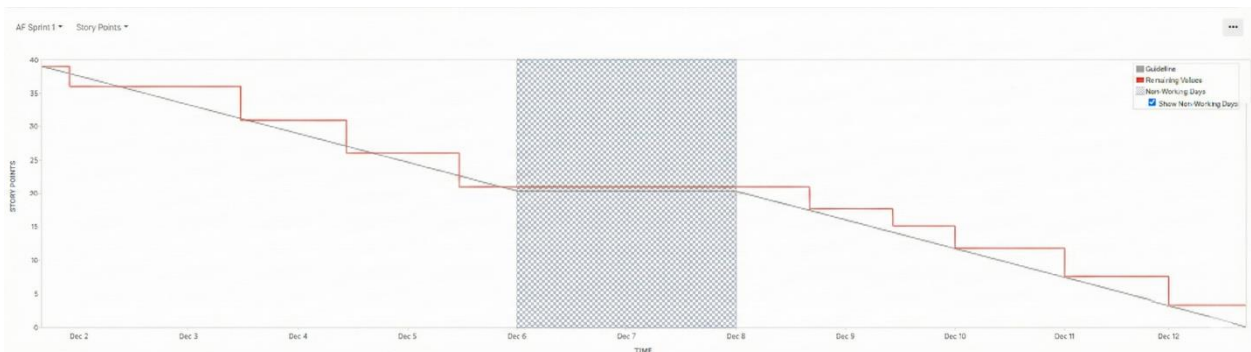


Рисунок 2.8 – Графік згоряння завдань у AF Sprint 1 проєкту «AssetFlow».

Джерело: розроблено автором

Таким чином, Burndown AF Sprint 1 виступає еталонним прикладом здорового спринту для даної команди: коректно обраний обсяг робіт, відсутність тривалих плато та передбачуване завершення всіх задач. У наступних підрозділах аналіз перебігу спринтів AF Sprint 2–AF Sprint 4 проводиться вже на рівні сумарних метрик

Velocity, що дозволяє оцінити, як реальний хід робіт відхиляється від цієї «еталонної» картини.

2.3.5 Velocity команди та моделювання наступних спринтів

Velocity команди у проєкті «AssetFlow» визначається як кількість story points, яку команда фактично завершує у межах одного спринту. Для аналізу було розглянуто п'ять послідовних спринтів: підготовчий AF Sprint 0 та робочі AF Sprint 1–AF Sprint 4.

Плановий обсяг (Commitment) та фактичний обсяг виконаного (Completed) мали такі значення:

- AF Sprint 0: 39 / 39 story points;
- AF Sprint 1: 39 / 39 story points;
- AF Sprint 2: 36 / 28 story points;
- AF Sprint 3: 36 / 22 story points;
- AF Sprint 4: 39 / 30 story points.

На основі цих даних побудовано Velocity-чарт (рисунок 2.3.3), де сірі стовпчики відображають плановий обсяг робіт, а зелені – фактичну Velocity кожного спринту.

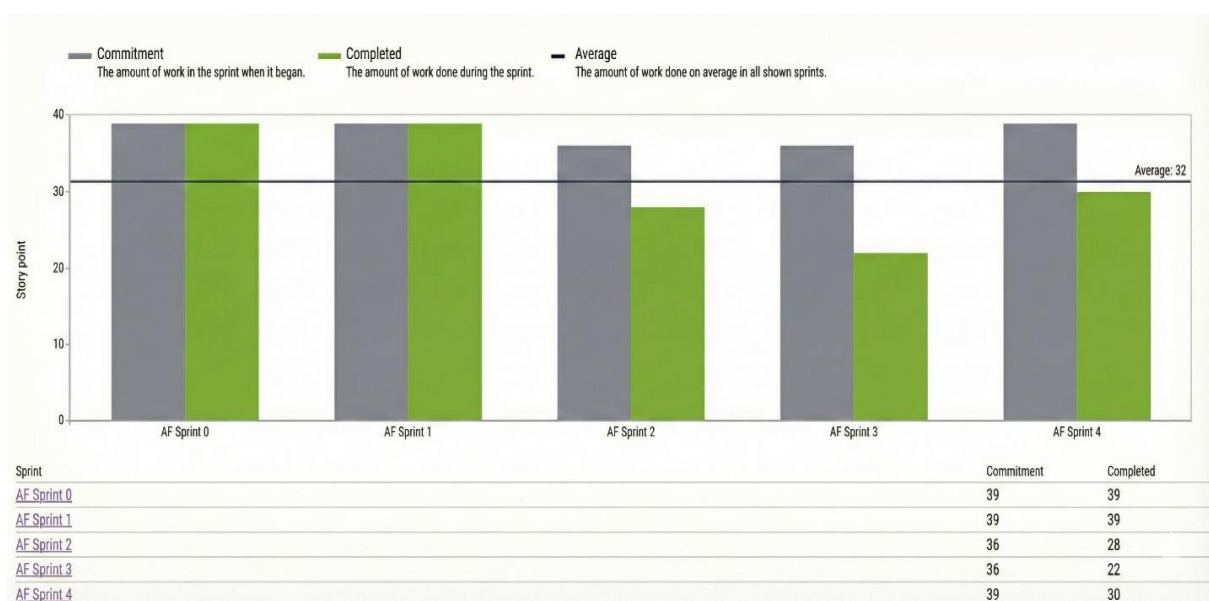


Рисунок 2.9 – Діаграма Velocity команди розробки «AssetFlow» за кілька спринтів.

Джерело: розроблено автором

Аналіз отриманої діаграми показує таке:

– У AF Sprint 0 та AF Sprint 1 команда повністю виконала запланований обсяг робіт (39 story points), що корелює з «ідеальною» формою Burndown-графіка AF Sprint 1.

– Починаючи з AF Sprint 2 Velocity знижується до 28 story points при плані 36. Частина задач переноситься в наступний спринт, що свідчить про зростання складності робіт (особливо в модулях «Клієнти та рахунки» та «Торгові операції»).

– У AF Sprint 3 падіння Velocity продовжується: при тому ж плані 36 story points команда завершує лише 22 story points. Свою роль відіграли накопичення перенесених задач, святковий період та інтеграційні ризики ядра OMS.

– У AF Sprint 4, після аналізу причин провалів попередніх спринтів, Velocity зростає до 30 story points при плановому обсязі 39. Команда покращує декомпозицію великих історій, уточнює критерії готовності та обмежує додавання незапланованих задач у середині спринту. Хоча план усе ще не виконується повністю, розрив між Commitment і Completed зменшується.

За підсумками цих п'яти спринтів було зроблено висновок, що стійка пропускна здатність команди становить близько 30 story points на спринт. Саме це консервативне значення Velocity ($V = 30$) було взято за базове для подальшого моделювання.

Наприклад, для епіку «Торгові операції» залишок оціненого беклогу становив $R = 120$ story points. Застосувавши розрахунок кількості необхідних ітерацій за формулою

$$N = R / V,$$

де N – кількість спринтів, R – залишок беклогу в story points, V – середня Velocity (30 SP), було отримано $N = 120 / 30 = 4$ спринти.

Таким чином, історична Velocity стала базисом для моделювання наступних спринтів та уточнення дорожньої карти продукту. Перехід від календарного планування «згори вниз» до такого емпіричного підходу дозволив, за оцінками

команди, орієнтовно на 20 % підвищити точність прогнозування дати MVP-релізу порівняно з початковим планом.

2.3.6 Інтерпретація результатів моніторингу та управлінські рішення

Отримані в проєкті «AssetFlow» дані Burndown-графіка AF Sprint 1, Velocity-чарту за спринтами AF Sprint 0–AF Sprint 4 та результати моделювання за формулою $N = R/V$ використовуються не лише для констатації факту виконання чи невиконання плану. Вони стали основою для ухвалення низки управлінських рішень, які безпосередньо вплинули на хід розробки та якість планування.

По-перше, сформовано еталонний профіль спринту. Burndown AF Sprint 1 задає цільову картину: рівномірне зниження залишку робіт, короткі плато у вихідні, відсутність різких «обвалів» наприкінці ітерації та повне виконання запланованого обсягу. Саме з цим профілем порівнювалися подальші спринти при аналізі Velocity: будь-яке системне відхилення (недовиконання плану, зниження фактичних story points) трактувалося як сигнал про проблеми у плануванні, декомпозиції або управлінні ризиками.

По-друге, скориговано підходи до планування спринтів. Після того як було зафіксовано падіння Velocity в AF Sprint 2–AF Sprint 3 та обчислено стійку пропускну здатність команди на рівні 30 SP, планування спринтів почали будувати «від Velocity», а не «від бажань стейкхолдерів». Це означало свідоме обмеження запланованого обсягу робіт, пріоритизацію епіків та перенесення частини функцій (зокрема, низькопріоритетних сценаріїв торгових операцій) у наступні релізи.

По-третє, результати моніторингу стали підставою для посилення практик декомпозиції та оцінювання задач. Виявлене у спринтах AF Sprint 2–AF Sprint 3 відставання від плану показало, що великі й слабо деталізовані історії суттєво збільшують ризик недовиконання спринту. У відповідь команда запровадила обов'язкове розбиття складних вимог на менші історії та підзадачі, використання референтних історій для калібрування story points, а також жорсткіший контроль за дотриманням Definition of Done.

По-четверте, покращено управління ризиками та блокерами. Хоча наявні дані не давали повної картини всіх Burndown-графіків, саме коливання Velocity вказували на вплив зовнішніх залежностей (тестові дані, інтеграційні точки, уточнення регуляторних вимог). Для таких випадків у беклозі стали створювати окремі задачі «task/спайк», що дозволило явним чином фіксувати й відпрацьовувати невизначеність, замість того щоб приховано переносити її між спринтами.

Нарешті, використання формули $N = R/V$ для моделювання кількості необхідних спринтів дало змогу узгодити очікування щодо дорожньої карти продукту. Прогноз завершення епіків і MVP почали будувати не на основі абстрактних дат, а виходячи з реального темпу команди. Це підвищило прозорість комунікацій з бізнес-стейкхолдерами та дозволило своєчасно приймати рішення про зміну обсягу релізів, не очікуючи накопичення критичних відставань.

Висновки до розділу 2

У другому розділі було обґрунтовано вибір гнучкого фрейму управління розробкою системи «AssetFlow» та описано організацію роботи Scrum-команди: визначено ролі й відповідальність учасників, структуру беклогу та епік-слендів, дорожню карту розвитку системи, WBS-структуру, календарний план і бюджет фази розробки MVP.

Окремо було описано систему моніторингу ходу робіт, яка спирається на використання Jira, Burndown-графіка, метрики Velocity та моделювання кількості необхідних спринтів для досягнення стану MVP. У сукупності це формує практичний підхід до гнучкого управління створенням системи обробки замовлень «AssetFlow», який у третьому розділі проаналізовано на прикладі конкретного інкременту продукту та роботи Scrum-команди в AF Sprint 3.

РОЗДІЛ 3.

РЕЗУЛЬТАТИ РОБОТИ ТА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ КОМАНДИ

3.1. Огляд реалізованого інкременту продукту та ретроспектива роботи команди

У межах третього спринту (AF Sprint 3) було реалізовано ключовий інкремент системи обробки замовлень «AssetFlow», який поєднує базове ядро OMS та перший користувацький інтерфейс для роботи трейдера із замовленнями. Цей інкремент забезпечує повний життєвий цикл біржового замовлення – від моменту створення до відправлення на торговий майданчик, часткового/повного виконання, скасування або відхилення, а також відображення історії змін стану.

У спринт увійшли user stories, згруповані в епіки «Ядро системи обробки замовлень» та «Інтерфейс замовлень для трейдера». Основні робочі елементи інкременту:

- AF-28 – визначення життєвого циклу замовлення (модель станів і допустимих переходів між ними);
- AF-29 – реалізація операції створення замовлення;
- AF-30 – реалізація операції скасування замовлення;
- AF-31 – реалізація операції зміни параметрів замовлення;
- AF-32 – перевірка коректності та валідація замовлення (бізнес-правила, ліміти клієнта, технічні обмеження);
- AF-33 – журналювання змін стану замовлення;
- AF-34 – маршрутизація замовлення на торговий майданчик;
- AF-35 – прототип інтерфейсу списку замовлень для трейдера.

На концептуальному рівні життєвий цикл замовлення описано як послідовність станів: «Нове», «Подане», «Частково виконане», «Виконане», «Скасоване», «Відхилене». Для кожного стану визначено дозволені переходи та події, які їх ініціюють (створення, успішна валідація, часткове/повне виконання, відхилення майданчиком, скасування за ініціативою клієнта або брокера). Відповідна діаграма

станів у поєднанні з діаграмою послідовності повідомлень між інтерфейсом трейдера, ядром OMS, модулем маршрутизації та торговим майданчиком наведено на рисунку 3.1.

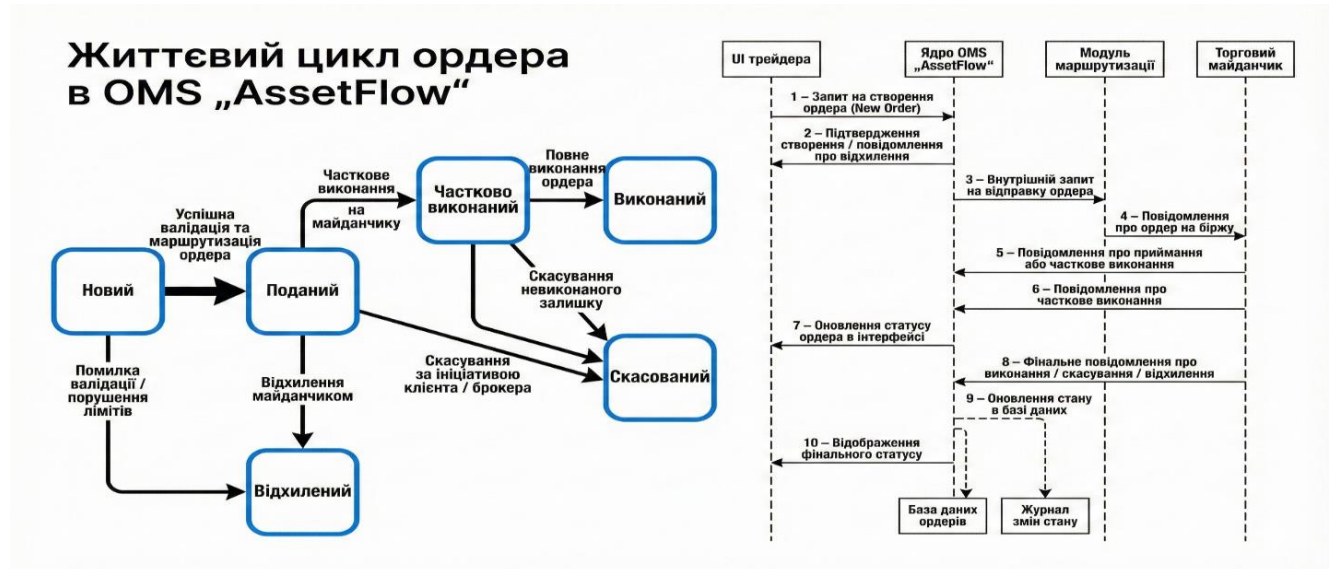


Рисунок 3.1 – Життєвий цикл ордеру в OMS «AssetFlow» та обмін повідомленнями з торговим майданчиком

Джерело: розроблено автором

На рівні середовища розробки інкремент було інтегровано в єдине середовище з попередніми модулями: використано спільну базу даних для зберігання інформації про ордери, налаштовано сервіси для журналювання подій та тестові конфігурації інтеграційного шлюзу. Це дозволило продемонструвати узгоджену роботу всіх компонентів та підтвердити, що отриманий інкремент може розглядатися як основа мінімально життєздатного продукту OMS «AssetFlow» для підтримки операцій з ринковими ордерами.

З технічного боку в інкременті реалізовано такі функціональні можливості ядра OMS «AssetFlow»:

- Створення замовлення. Ядро приймає запит від користувачького інтерфейсу, виконує валідацію обов’язкових параметрів (інструмент, сторона, кількість, ціна для лімітних заявок, тип замовлення, термін дії), перевіряє ліміти

клієнта та коректність торгового режиму. У разі успіху замовлення реєструється в базі даних у стані «Нове» та повертається підтвердження у UI.

- валідація та обробка помилок. У разі порушення бізнес-правил або технічних обмежень замовлення не приймається до обробки, фіксується відповідна причина відхилення, а користувачу повертається повідомлення про помилку.

- маршрутизація на майданчик. Для замовлень, що пройшли валідацію, формується внутрішній запит до модуля маршрутизації. Останній визначає цільовий торговий майданчик (біржу) та надсилає на нього повідомлення у погодженому форматі.

- обробка відповідей майданчика. Повідомлення про приймання, часткове або повне виконання, а також відхилення замовлення повертаються до ядра OMS через модуль маршрутизації. На їх основі оновлюється стан замовлення в базі даних та формуються записи в журналі подій.

- журналювання життєвого циклу. Кожна зміна стану замовлення супроводжується окремим записом у журналі із зазначенням часу, типу події, користувача та торгового майданчика, що забезпечує можливість подальшого аудиту та формування регуляторної звітності.

- операції зміни та скасування. Для активних замовлень (наприклад, у стані «Нове» або «Подане») реалізовано можливість зміни окремих параметрів та повного скасування з фіксацією відповідних подій у базі даних та журналі.

Для демонстрації реалізованих можливостей ядра в межах інкременту розроблено прототип користувацького інтерфейсу трейдера. Основним робочим екраном є «Список замовлень», який дозволяє трейдеру:

- переглядати перелік замовлень поточного робочого дня з основними параметрами (час створення, ID, інструмент, сторона, кількість, ціна, тип, статус, виконано/залишок, майданчик);

- виконувати пошук за ID замовлення або інструментом;

- фільтрувати замовлення за інструментом, стороною, статусом, часовим інтервалом;

– обирати конкретне замовлення та переглядати його деталі, включно з таймлайном подій життєвого циклу.

У правій частині екрана розміщено панель «Деталі замовлення», яка відображає розгорнуту інформацію щодо виділеного в таблиці запису: інструмент, сторона операції, кількість, ціна, тип замовлення, ідентифікатор клієнта, підсумок виконання та хронологію подій. Там же передбачена кнопка «Скасувати замовлення», що ініціює відповідний бізнес-процес. Структура екрана та основні елементи інтерфейсу наведені на рисунку 3.2.

Другим важливим артефактом інкременту є форма створення нового замовлення, яка відкривається при натисканні кнопки «Нове замовлення». Вона забезпечує введення всіх основних параметрів: інструмента, сторони операції (купівля/продаж), типу замовлення (лімітне або ринкове), кількості, терміну дії, торгового майданчика та клієнта. У формі реалізовано елементи первинної валідації, а також короткий підсумок заявки перед підтвердженням, що знижує ризик помилкового введення параметрів. Приклад прототипу вікна створення нового замовлення показано на рисунку 3.3.

У сукупності наведені функції дозволяють розглядати реалізований інкремент як основу мінімально життєздатної версії системи обробки замовлень OMS «AssetFlow». Він забезпечує повний цикл роботи із замовленнями на стороні брокера, прозору фіксацію всіх змін стану та базовий, але достатньо зручний інтерфейс для роботи трейдера.

AssetFlow OMS Demo Trader UA01 TU

Список замовлень
Ручні та алгоритмічні замовлення поточного робочого дня

[+ Нове замовлення](#) [Зберегти фільтр](#)

Пошук за ID замовлення або інструментом...

Інструмент: Усі Інструменти | Сторона замовлення: Усі | Статус замовлення: Усі | Дата від: Дата від | Дата до: Дата до | [Застосувати](#) | [Скинути](#)

Час	ID замовлення	Інструмент	Сторона	Кількість	Ціна	Тип замовлення	Статус замовлення	Виконано / Залишок	Майданчик	Дії
10:44:15	AF-ORD-2025-000122	AAPL	SELL	500	178.25	Ринкове	Частково виконане	250 / 250	PFTS	Очі П В
10:40:05	AF-ORD-2025-000120	UX-03.25	BUY	1 000	2450.50	Лімітне	Виконане	1000 / 0	UX	Очі П В
10:35:22	AF-ORD-2025-000119	TSLA	SELL	200	750.00	Лімітне	Скасоване	0 / 200	NASDAQ	Очі П В
10:35:22	AF-ORD-2025-000119	AAPL	SELL	500	750.00	Лімітне	Виконане	0 / 200	NASDAQ	Очі П В
10:40:05	AF-ORD-2025-000117	UX-03.25	BUY	1 000	2450.50	Лімітне	Виконане	50 / 0	UX	Очі П В

Деталі замовлення AF-ORD-2025-000122
Частково виконане
Останнє оновлення 10:44:15

Основна інформація

Інструмент: AAPL
Сторона: SELL
Кількість: 500
Ціна: 178.25
Тип замовлення: Ринкове
Клієнт: Client_01

Виконання

Виконано: 250
Залишок: 250
Середня ціна виконання: 178.20
50% 250 / 250

Таймлайн подій

- 10:44:15 Замовлення створено трейдером UA01
- 10:44:15 Замовлення подано на майданчик PFTS
- 10:44:25 Часткове виконання: 250

[Скасувати замовлення](#) [Закрити](#)

Рисунок 3.2 – «Список замовлень в OMS «AssetFlow»»

Джерело: розроблено автором

Нове замовлення
Створення торгового замовлення для клієнта

Інструмент: AAPL | Торговий майданчик: PFTS

Сторона: Купівля / Продаж

Клієнт: Client_01

Тип замовлення: Ринкове | Доступний ліміт: 100 000 USD

Кількість: 500
Вкажіть коректну кількість, перевищено ліміт.

Ціна: Визначається ринком

Термін дії: На сьогодні

Підсумок
Інструмент: AAPL
Операція: Продаж 500 шт
Тип замовлення: Ринкове

Після підтвердження замовлення буде відправлено на торговий майданчик згідно з налаштуваннями маршрутизації.

Скасувати | Створити замовлення

Деталі замовлення AF-ORD-2025-000122
Частково виконано
Останнє оновлення 10:44:15

Основна інформація

Інструмент	AAPL
Сторона	SELL
Кількість	500
Ціна	178.25
Тип замовлення	Ринкове
Клієнт	Client_01

Виконання

Виколопо	250
Залишок	250
Середня ціна експонація	178.20
50%	250 / 250

Таймлайн подій

- 10:44:15 Замовлення створено трейдером UAD1
- 10:44:15 Замовлення подало на майданчик PFTS
- 10:44:25 Часткове виконання: 250

Скасувати замовлення | Закрити

Рисунок 3.3 – «Форма створення нового замовлення в OMS «AssetFlow»

Джерело: розроблено автором

3.1.1. Оцінка досягнення цілей спринтів та якості інкременту

Третій спринт (AF Sprint 3) планувався як ключовий для отримання цілісного інкременту OMS «AssetFlow», у межах якого мали бути поєднані ядро системи обробки замовлень та перший користувацький інтерфейс для роботи трейдера. На початку спринту команда взяла на себе зобов'язання обсягом 36 story points, що включали всі user stories AF-28 – AF-35.

Аналіз діаграми velocity Scrum-команди, що відображає співвідношення між запланованим обсягом робіт (Commitment) та фактично виконаним (Completed) по спринтах AF Sprint 0–4, показує, що розвиток продуктивності команди був нерівномірним. У нульовому та першому спринтах команда повністю виконала взяті зобов'язання (39/39 story points), що створило враження стабільного темпу. Починаючи зі спринту AF Sprint 2, з'явився недовиконаний обсяг (36/28 story points), а у спринті AF Sprint 3 розрив став найбільшим: із запланованих 36 story points було завершено лише 22, тобто близько 60 % від початкового зобов'язання. У AF Sprint 4 частина перенесених робіт була завершена (39/30 story points), проте повністю наздогнати план попереднього спринту не вдалося.

Причини відхилення від плану, виявлені під час інспекції результатів та обговорені на ретроспективі, можна звести до таких груп:

- недооцінка складності інтеграції ядра та UI. Частина user stories вимагала одночасних змін у бізнес-логіці, базі даних і прототипах інтерфейсу, що призвело до переробок і додаткових погоджень вимог.
- залежності між задачами. Реалізація операцій зміни та скасування замовлень виявила приховані залежності від механізмів журналювання та маршрутизації, які спочатку планувалися як відносно незалежні.

– перенасичене планування. Команда орієнтувалася на velocity попередніх спринтів (39 story points) і зберегла майже такий самий обсяг, не врахувавши зростання складності задач та появу інтеграційних ризиків.

– частина робіт була виконана лише частково. У спринт потрапили задачі, які до кінця ітерації перебували в стані «In progress» (наприклад, доопрацювання валідаційних повідомлень, розширені фільтри в UI, підготовка додаткових автоматизованих тестів) і були перенесені в наступний спринт.

Попри недовиконання плану за обсягом, критичні для інкременту сценарії були реалізовані: створення замовлення, базова валідація, маршрутизація на майданчик, обробка часткового та повного виконання, фіксація змін стану в журналі, відображення списку замовлень і детальної інформації по кожному з них у UI. Саме ці задачі сформували «скелет» інкременту, який можна демонструвати стейкхолдерам як робочу версію продукту.

Оцінка якості інкременту здійснювалася через виконання узгодженого Definition of Done для завершених user stories. DoD включав:

- реалізацію бізнес-логіки згідно з описаними сценаріями та моделлю життєвого циклу замовлення;
- збереження даних у базі та запис ключових подій у журнал;
- ручове тестування основних сценаріїв (створення, часткове та повне виконання, скасування, відхилення);
- актуалізацію прототипів інтерфейсу («Список замовлень», «Нове замовлення») та опису взаємодії з майданчиком;
- відсутність критичних дефектів, які блокують демонстрацію інкременту.

Для завершених задач Definition of Done був виконаний, але частина запланованого score залишилася незавершеною й була перенесена до

наступного спринту. Це означає, що формально цілі AF Sprint 3 були досягнуті частково: команда отримала працездатний інкремент ядра та UI, придатний для демонстрації та подальшого розвитку, однак не змогла повністю реалізувати початковий обсяг функцій, запланований у спринті. Цей факт став одним із ключових предметів аналізу на ретроспективі та підґрунтям для корекції підходів до планування та оцінювання робіт у наступних ітераціях.

3.1.3. Організація та результативність роботи Scrum-команди під час AF Sprint 3

Роботу над інкрементом OMS «AssetFlow» у спринті AF Sprint 3 виконувала Scrum-команда, до складу якої входили Product Owner, Scrum Master та кросфункціональна команда розробки (бекенд-розробники, інженер інтерфейсу, тестувальник). Команда працювала за стандартними подіями Scrum і використовувала Jira як основний інструмент планування та відстеження задач.

Під час Sprint Planning було визначено ціль спринту: отримати робочу версію ядра системи обробки замовлень та інтерфейс трейдера, який дозволяє створювати, переглядати й управляти замовленнями. На основі пріоритизації Product Owner команда відібрала user stories AF-28 – AF-35, оцінила їх у story points і взяла зобов'язання обсягом 36 story points. Уже на етапі планування частина ризиків інтеграції ядра та UI була відома, однак команда недооцінила їхній вплив на тривалість виконання задач.

Упродовж спринту робота будувалася навколо канбан-дошки з колонками «To Do», «In Progress», «Code Review / Testing», «Done». Бекенд-розробники зосередилися на реалізації життєвого циклу замовлення, валідації, маршрутизації та журналювання (AF-28 – AF-34). Інженер інтерфейсу відповідав за екран «Список замовлень», панель «Деталі замовлення» та

форму «Нове замовлення» (AF-35), синхронізуючи макети з моделлю даних і API бекенду. Тестувальник проводив ручні перевірки наскрізних сценаріїв (створення – виконання – скасування) та фіксував дефекти в Jira.

Команда проводила щоденні стендапи, де фіксувалися прогрес і блокери. Саме на цих зустрічах стало очевидно, що обраний обсяг робіт надто амбітний для одного спринту:

- частина задач зі зміни й скасування замовлень виявилася жорстко пов'язаною з механізмом журналювання та обробкою відповідей майданчика;
- у процесі розробки UI з'явилися додаткові узгодження щодо форматів даних та поведінки статусів, що призвело до переоцінки окремих stories.

Scrum Master відстежував блокери, ініціював додаткові технічні обговорення, допомагав розбивати надто великі задачі на менші та координував актуалізацію спринт-беклогу. Product Owner, у свою чергу, уточнював критерії приймання та пріоритети: у другій половині спринту фокус був зміщений на завершення сценаріїв, критичних для демонстрації інкременту, навіть за рахунок відтермінування менш важливих покращень. За результатами спринту команда перевела в статус «Done» задачі, що формують ключову цінність інкременту:

- модель життєвого циклу замовлення та переходів між станами;
- створення, базову валідацію, маршрутизацію замовлень і обробку відповідей майданчика;
- журналювання змін стану;
- екран «Список замовлень» із панеллю деталей та інтеграцією зі сценаріями бекенду.

Частина запланованого функціоналу (розширені перевірки, окремі сценарії редагування, додаткові UX-удосконалення) залишилася на етапі «In Progress»

і була перенесена в AF Sprint 4. Це напряду відобразилося на velocity: при зобов'язанні 36 story points у AF Sprint 3 команда змогла завершити лише 22. Узагальнюючи, AF Sprint 3 показав, що команда здатна поставляти цінний інкремент продукту й ефективно співпрацювати між ролями, але потребує більш консервативного планування обсягу робіт, ретельнішого опрацювання залежностей між задачами та регулярнішого спільного refinement-процесу перед плануванням. Ці висновки стали основою для ретроспективи, результати якої наведено в підрозділі 3.1.4.

3.1.4. Ретроспектива роботи Scrum-команди (формат «Парусник») після AF Sprint 3

Після завершення AF Sprint 3 команда провела ретроспективу у форматі «Парусник» на онлайн-дошці (за шаблоном, аналогічним Mural Sailboat Retrospective). Робоче поле було поділено на чотири зони:

- «Вітер у вітрилах» – що допомагало команді рухатися вперед;
- «Маяки – цілі на горизонті» – до яких результатів команда прагнула в цьому спринті;
- «Якір – що нас гальмує» – що сповільнювало виконання задач;
- «Перешкоди – ризики попереду» – потенційні загрози для наступних спринтів.

За результатами обговорення команда сформувала окремий артефакт – «План дій за результатами ретроспективи AF Sprint 3», у якому визначено, що саме потрібно змінити в процесі роботи, хто за це відповідає та коли очікується впровадження відповідних кроків. Узагальнений вигляд цього артефакту наведено в таблиці 3.2.

Таблиця 3.2 – План дій за результатами ретроспективи AF Sprint 3

№	Запланована дія	Очікуваний ефект	Відповідальні	Термін виконання
1	Планувати обсяг спринту, орієнтуючись на середній фактичний velocity, а не на максимальне значення	Реалістичніші зобов'язання, зменшення розриву між «заплановано» та «виконано», менше перенесених задач у наступні спринти	Product Owner, Scrum Master	Починаючи з планування AF Sprint 4
2	Проводити регулярні сесії refinement'у та дробити великі user stories на менші з прозорими критеріями	Краще розуміння обсягу робіт, менше великих «незакритих» stories, швидше досягнення Definition of Done	Вся команда, модерація – Scrum Master	Щонайменше одна сесія refinement на тиждень, починаючи з AF Sprint 4
3	Явно позначати залежності між задачами в Jira (зв'язки типу «blocks / is blocked by»)	Зниження кількості блокерів у середині спринту, краща послідовність виконання задач, прозорість інтеграційних залежностей	Усі розробники, контроль – Scrum Master	З AF Sprint 4 – на постійній основі
4	Обмежити кількість задач у статусі «In Progress» на одного розробника (WIP-ліміти 1–2 задачі)	Менше розпорошення уваги, швидше доведення задач до «Done», зменшення обсягу незавершених робіт наприкінці спринту	Команда розробки	Запровадити в AF Sprint 4, переглядати ефект на наступних ретроспективах
5	Починати тестування ключових сценаріїв не пізніше середини спринту	Раннє виявлення дефектів у наскрізних сценаріях, більше часу на їх виправлення без авралів наприкінці ітерації	Тестувальник, бекенд-розробники, інженер інтерфейсу	З AF Sprint 4 – як стандарт командної практики

Структуру та наповнення дошки ретроспективи у форматі «Парусник» для AF Sprint 3 наведено на рисунку 3.4.

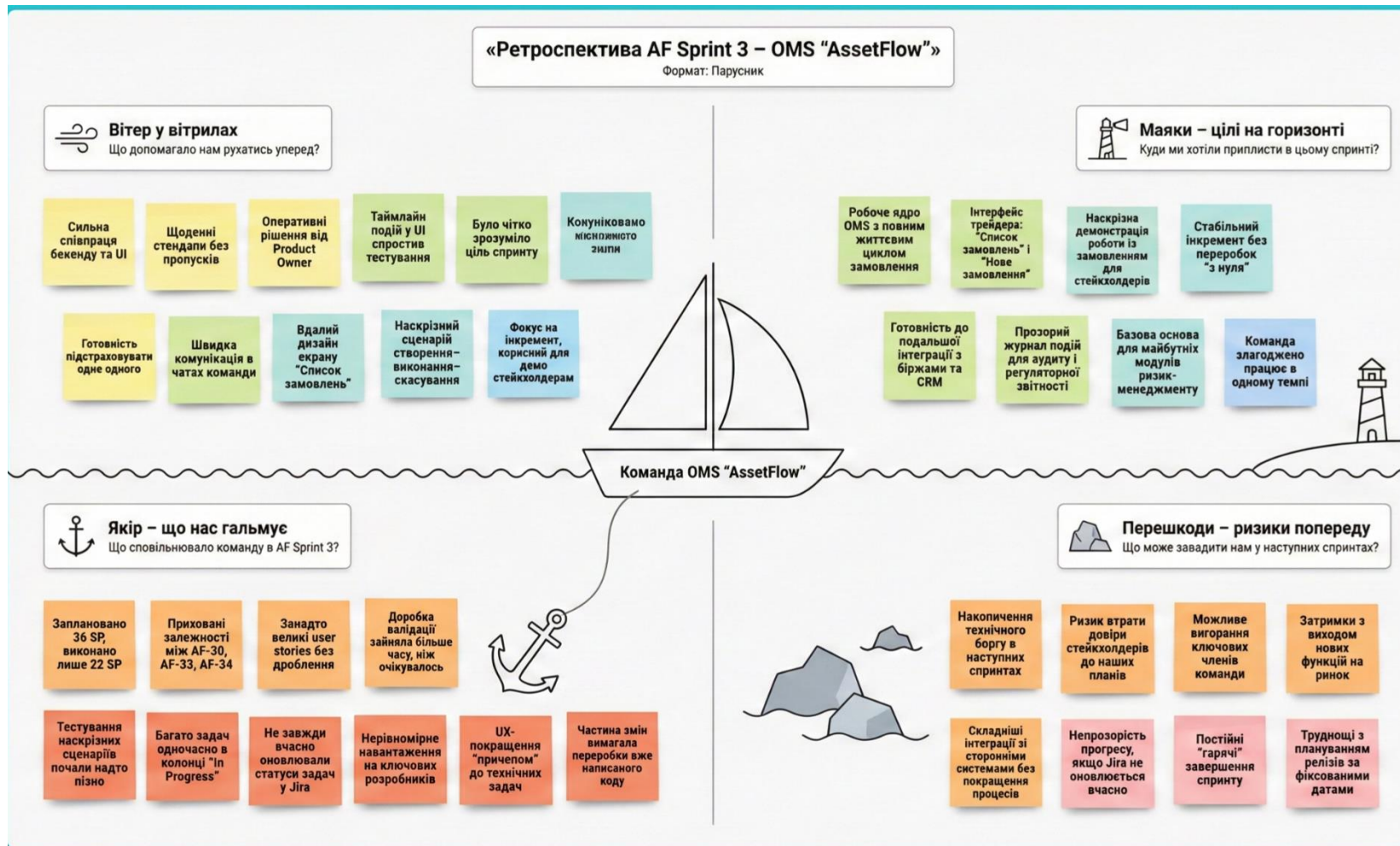


Рисунок 3.4 – Ретроспектива AF Sprint 3 у форматі «Парусник» для Scrum-команди проекту «AssetFlow»
Джерело: розроблено автором

3.2. Аналіз управлінської проблемної ситуації та її причин

У цьому підрозділі розглядається управлінська проблемна ситуація, пов'язана зі зниженням ефективності керованого об'єкта – Scrum-команди проєкту «AssetFlow», а також здійснюється аналіз її причин за допомогою діаграми Ішікави («рибний скелет»).

3.2.1. Формулювання управлінської проблемної ситуації

У попередніх підрозділах було показано, що в AF Sprint 3 Scrum-команда проєкту OMS «AssetFlow» зіткнулася зі значним відхиленням між запланованим та фактично виконаним обсягом робіт: при зобов'язанні 36 story points було завершено лише 22 story points. Частина задач залишилася у статусах «In Progress» або «Code Review / Testing» і була перенесена до наступного спринту.

З погляду управління це означає:

- зниження передбачуваності поставки інкрементів (команда не виконує повністю власні зобов'язання);
- виникнення ризику накопичення технічного та процесного боргу (незавершені задачі, відкладені покращення, відтерміновані перевірки);
- потенційний ризик втрати довіри стейкхолдерів до планів спринтів і термінів реалізації ключового функціоналу OMS.

Отже, керований об'єкт – Scrum-команда проєкту «AssetFlow» – продемонстрував тимчасову втрату ефективності: команда, маючи стабільний досвід попередніх ітерацій, не змогла в межах AF Sprint 3 поставити повний запланований обсяг робіт, що негативно вплинуло на терміни завершення функціоналу ядра OMS та інтерфейсу трейдера.

Ключова управлінська проблема полягає у зниженні показників ефективності Scrum-команди та погіршенні прогнозованості інкрементальних поставок. Зазначена ситуація зумовлена дисбалансом між обсягом взятих зобов'язань та пропускнуою здатністю команди, низьким рівнем формалізації процесів планування та уточнення вимог (refinement), а також недоліками в координації залежностей та забезпеченні якості.

Для подальшого опрацювання цієї проблеми було проведено аналіз причин за допомогою діаграми Ішікави.

3.2.2. Аналіз причин за допомогою діаграми Ішікави (рибний скелет)

Для виявлення ключових причин втрати ефективності команди в AF Sprint 3 було побудовано діаграму Ішікави, де в «голові» риби розміщено сформульовану вище проблему:

«Недосягнення цілей AF Sprint 3 та зниження ефективності Scrum-команди».

Основними «кістками» (категоріями причин) діаграми стали:

1. Планування та оцінювання
2. Процеси та практики Scrum
3. Команда та розподіл навантаження
4. Інструменти та інформаційна прозорість
5. Комунікація та робота зі стейкхолдерами
6. Зовнішні чинники та контекст проєкту

Кожна категорія містить конкретні причини, виявлені під час обговорення з командою та аналізу ходу AF Sprint 3.

1. Планування та оцінювання

У цій гілці були виділені такі причини:

– орієнтація на максимальний показник velocity попередніх спринтів, а не на середнє значення, що призвело до завищеного обсягу

зобов'язань (36 story points);

- недостатня деталізація user stories до рівня відносно невеликих, однорідних задач, через що окремі історії виявилися «занадто великими» для завершення в межах одного спринту;

- відсутність резерву на інтеграційні ризики (поєднання ядра OMS та UI, синхронізація з журналюванням і маршрутизацією).

2. Процеси та практики Scrum

У межах цієї гілки було відзначено:

- недостатню регулярність сесій деталізації беклогу (refinement), унаслідок чого окремі історії були недостатньо чітко описані перед плануванням;

- запізнілий початок тестування наскрізних сценаріїв (основна хвиля перевірок припала на кінець спринту);

- відсутність чітко зафіксованих обмежень на обсяг роботи в процесі (WIP-обмежень), що призвело до великої кількості задач у статусі «In Progress».

3. Команда та розподіл навантаження

У цій гілці були виділені:

- нерівномірний розподіл задач між учасниками (перевантаження ключових розробників бекенду, відносно менше навантаження на інших членів команди);

- часті переключення між задачами, що знижували концентрацію та збільшували час на завершення кожної окремої задачі;

- недостатнє використання можливостей взаємопідстраховки у частині тестування й доопрацювань користувацького інтерфейсу.

4. Інструменти та інформаційна прозорість

До цієї групи причин віднесено:

- не всі залежності між задачами були явно позначені в Jira (зв'язки типу «blocks / is blocked by»), через що частина робіт блокувалася вже під час виконання;

- невчасне оновлення статусів задач у Jira окремими членами команди, що знижувало прозорість реального стану спринту;

- обмежене використання аналітичних можливостей Jira (діаграми, аналіз блокерів) при оперативному управлінні спринтом.

5. Комунікація та робота зі стейкхолдерами

У цій гілці відзначено:

- окремі уточнення вимог і пріоритетів під час спринту, які призводили до зміни обсягу робіт уже в межах взятих історій;

- не завжди достатньо чітке фіксування домовленостей щодо поведінки системи в нестандартних ситуаціях, що вимагало повернення до вже реалізованих частин функціоналу.

6. Зовнішні чинники та контекст проєкту

Серед зовнішніх чинників виділено:

- поява непередбачених завдань (дрібні виправлення, допоміжні задачі з інфраструктури), які відволікали членів команди від основного спринтового беклогу;

- загальна складність предметної області (біржові операції, життєвий цикл замовлень, регуляторні вимоги), що підвищувала «ціну помилки» і змушувала витратити більше часу на узгодження деталей.

Узагальнене графічне представлення наведених причин у вигляді діаграми Ішікави наведено на рисунку 3.5.

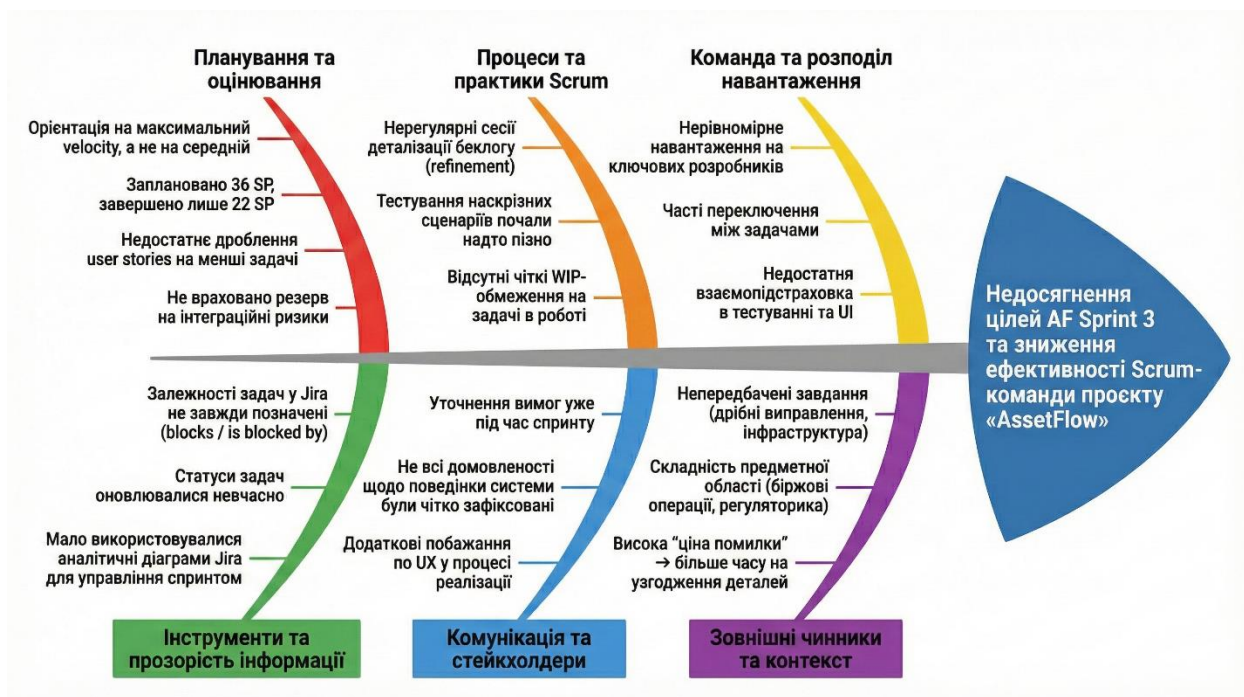


Рисунок 3.5 – Діаграма Ішікави причин недосягнення цілей AF Sprint 3 та втрати ефективності Scrum-команди проекту «AssetFlow»

Джерело: розроблено автором

3.2.3. Узагальнення результатів аналізу причин

Проведений аналіз показав, що втрата ефективності Scrum-команди в AF Sprint 3 має комплексний характер і не зводиться до одного чинника. Водночас більшість виявлених причин можна згрупувати в кілька ключових блоків:

- стратегія планування та структурування робіт – завищений обсяг зобов'язань, орієнтація на максимальний, а не середній рівень продуктивності, недостатнє дроблення user stories та неповний облік інтеграційних ризиків.
- якість процесів Scrum – нерегулярні сесії деталізації беклогу, нечіткі обмеження на обсяг роботи в процесі, запізнений старт тестування наскрізних сценаріїв.

- управління залежностями та прозорість виконання – неформалізовані в Jira залежності між задачами, невчасне оновлення статусів, обмежене використання аналітики для прийняття управлінських рішень.

Саме ці фактори в сукупності призвели до того, що команда, маючи високий рівень мотивації та конструктивну взаємодію, не змогла повністю виконати запланований обсяг робіт у AF Sprint 3 й частково втратила керованість темпом поставки інкрементів.

Результати аналізу за допомогою діаграми Ішікави слугують відправною точкою для розроблення сучасних підходів до вирішення управлінської ситуації. У наступному підрозділі 3.3 на основі виявлених причин буде сформовано набір управлінських інструментів і практичних рекомендацій щодо вдосконалення процесу гнучкого управління проектом «AssetFlow».

3.3.3. Модель оцінювання задач беклогу за критеріями цінності, спроможності, ризику та технічної складності

Щоб зменшити ризик завищення обсягу спринтів і врахувати не лише трудомісткість задач, а й управлінські аспекти (корисність для бізнесу, відповідність можливостям команди, ризику та технічний вплив), доцільно застосувати просту багатокритеріальну модель оцінювання задач беклогу.

Для кожної задачі пропонується оцінювати чотири критерії:

- Ц – цінність задачі: наскільки реалізація цієї функції важлива для бізнесу та кінцевих користувачів, чи дозволяє вона наблизити продукт до визначених цілей.
- С – відповідність спроможності команди: наскільки задача відповідає наявним компетенціям команди, обмеженням за часом та доступним ресурсам.

- Р – рівень ризику: ступінь невизначеності вимог, залежність від зовнішніх систем, складність предметної області, імовірність переробок.
- Т – технічна складність та вплив на архітектуру: очікуваний обсяг змін у технічній частині, можливий вплив на архітектуру системи та технічний борг.

Для всіх критеріїв використовується єдина порядкова шкала від 1 до 5, де 1 означає мінімальний рівень, а 5 – максимальний. Узагальнена інтерпретація подана в таблиці 3.4.

Таблиця 3.4 – Критерії оцінювання задач беклогу та їх шкала

Критерій	Зміст	1 – низький рівень	3 – середній рівень	5 – високий рівень
Ц	Цінність задачі для бізнесу та користувача	Невеликий вплив; другорядне покращення, майже непомітне користувачу	Помітне покращення окремих сценаріїв; бажана, але не критично необхідна функція	Критичний функціонал; без нього інкремент втрачає значну частину цінності або неможлива повноцінна демонстрація стейкхолдерам
С	Відповідність спроможності команди	Команда майже не має досвіду; потрібні сторонні експерти або суттєве додаткове навчання	Частковий досвід є, але потрібні додаткові узгодження чи дослідження	Команда має достатній досвід і напрацьовані рішення; задача добре вписується в поточну спроможність
Р	Рівень ризику виконання	Низький ризик; вимоги стабільні, залежностей мінімум	Середній ризик; є окремі невизначеності або залежності	Високий ризик; багато невідомих, сильна залежність від зовнішніх чинників або складної предметної області
Т	Технічна складність та вплив на архітектуру	Технічно проста задача, не впливає на архітектуру	Помірна складність; локальні зміни в кількох компонентах	Висока складність; значний вплив на архітектуру або технічний борг

Оцінювання критеріїв Ц, С, Р, Т здійснюється командою під час сесій деталізації беклогу. Для підтримки ухвалення рішень щодо пріоритизації задач може бути розрахований інтегральний показник пріоритету.

Один із можливих варіантів розрахунку:

- задаються вагові коефіцієнти для кожного критерію, наприклад:
 $w_{Ц} = 0,4$; $w_{С} = 0,2$; $w_{Р} = 0,2$; $w_{Т} = 0,2$;
- щоб нижчий ризик підвищував пріоритет задачі, у формулі використовується вираз $(6 - Р)$;
- інтегральний показник пріоритету обчислюється так:

$$P_{\text{пріор}} = 0,4 \cdot Ц + 0,2 \cdot С + 0,2 \cdot (6 - Р) + 0,2 \cdot Т.$$

Задачі з вищим значенням $P_{\text{пріор}}$ мають вищий управлінський пріоритет: вони поєднують значну цінність для бізнесу, відповідають спроможності команди, дають передбачуваний технічний результат і не несуть надмірних ризиків. Саме такі задачі доцільно насамперед включати до спринтів в умовах обмеженої місткості.

Для ілюстрації практичного застосування моделі розглянемо три задачі беклогу OMS «AssetFlow»:

1. реалізація повного життєвого циклу замовлення з журналом подій;
2. розширені фільтри та сортування у списку замовлень;
3. оптимізація продуктивності журналу замовлень за рахунок індексів у базі даних.

Припустимо, що команда оцінила їх за критеріями цінності, спроможності, ризику та технічної складності, а також розрахувала інтегральний показник пріоритету $P_{\text{пріор}}$ (таблиця 3.5).

Таблиця 3.5 – Приклад оцінювання задач беклогу за критеріями Ц, С, Р, Т

№ задачі	Короткий опис задачі	Ц	С	Р	Т	$P_{\text{пріор}}$
1	Повний життєвий цикл замовлення з журналом подій	5	4	2	4	4,4

№ задачі	Короткий опис задачі	Ц	С	Р	Т	Р_пріор
2	Розширені фільтри та сортування у списку замовлень	4	3	3	3	3,4
3	Оптимізація продуктивності журналу замовлень за рахунок індексів у базі даних	3	4	2	2	3,2

Як видно з таблиці, найвищий інтегральний пріоритет має задача 1 ($P_{\text{пріор}} = 4,4$), далі – задача 2 ($P_{\text{пріор}} = 3,4$), тоді як задача 3 ($P_{\text{пріор}} = 3,2$) має нижчий пріоритет і в умовах обмеженої місткості спринту може бути перенесена до наступної ітерації. Це демонструє, що запропонована модель дозволяє приймати більш обґрунтовані рішення щодо включення задач до спринту, поєднуючи кілька важливих управлінських критеріїв в одному показнику.

Висновки до розділу 3

У розділі 3 було досліджено, як реалізований інкремент продукту OMS «AssetFlow» та організація роботи Scrum-команди відображають практичне застосування гнучких підходів до управління. На прикладі AF Sprint 3 показано, що команда змогла створити демонстрабельний інкремент із робочим ядром системи та інтерфейсом трейдера (екрани «Список замовлень» і «Нове замовлення»), який підтримує повний життєвий цикл замовлення. Це підтвердило можливість побудови мінімально життєздатного продукту (MVP) для подальшого розширення функціоналу та інтеграцій.

Водночас аналіз виконання спринту засвідчив часткове недосягнення планових показників (36 проти 22 story points) та тимчасове зниження ефективності команди. За допомогою ретроспективи у форматі «Парусник» та діаграми Ішікави було виявлено, що основними причинами стали завищений

обсяг зобов'язань, недостатня деталізація беклогу, слабо формалізоване управління залежностями, запізнений старт тестування та відсутність чітких обмежень на обсяг роботи в процесі. Таким чином, проблема мала комплексний характер і стосувалася насамперед якості процесів планування й організації спринтів, а не професійного рівня окремих членів команди.

У відповідь на це в розділі було запропоновано використання сучасного інструментарію менеджера в agile-середовищі: систематизацію планування на основі середнього темпу команди, регулярні сесії деталізації беклогу, формалізацію WIP-обмежень, активне застосування аналітичних можливостей інструментів управління (зокрема Jira), а також запровадження багатокритеріальної моделі оцінювання задач за цінністю, відповідністю спроможності команди, ризиком і технічною складністю. Реалізація цих підходів має підвищити передбачуваність поставки інкрементів, зменшити ризик накопичення технічного й процесного боргу, а також зміцнити довіру стейкхолдерів до процесу розробки OMS «AssetFlow».

ВИСНОВКИ

У кваліфікаційній роботі розв'язано прикладну задачу гнучкого управління створенням системи обробки замовлень «AssetFlow» для фінансових ринків України. Обґрунтовано концепцію та організаційні засади її розробки із застосуванням Agile-підходів, що й становило основну мету дослідження.

У першому розділі на основі аналізу стану українського ринку капіталу, нормативного середовища та наявних OMS-рішень сформовано продуктове бачення системи «AssetFlow». Визначено ключові сегменти користувачів, їхні потреби, основні бізнес-процеси та вимоги до системи, що створило підґрунтя для подальшого проєктування та планування розробки.

Другий розділ присвячено організації гнучкого управління створенням системи. Обґрунтовано вибір фреймворку Scrum, описано структуру та ролі Scrum-команди, побудовано дорожню карту розвитку продукту, WBS, календарний план і бюджет фази MVP. Сформовано Product Backlog системи в Jira Software та систему моніторингу на основі Burndown-графіків, Velocity і PDCA-циклу, що забезпечує прозорість планування й відстеження прогресу.

У третьому розділі продемонстровано застосування розробленого підходу на прикладі AF Sprint 3: реалізовано інкремент із робочим ядром OMS та інтерфейсом трейдера, проаналізовано результати спринту, виявлено причини відхилень (завищене планування, недостатній refinement, управління залежностями, відсутність WIP-лімітів) і сформовано конкретні рекомендації щодо вдосконалення процесу. Запропоновано використання задач типу «risk/спайк», багатокритеріального оцінювання пріоритетів і діаграми Ішікави для аналізу управлінських проблемних ситуацій.

Наукова новизна роботи полягає в поєднанні бізнес-дизайну продукту OMS для українського ринку капіталу з організаційними рішеннями щодо гнучкого управління його створенням і системою моніторингу, побудованою на даних Jira та PDCA-циклі. Окремий внесок становить підхід до моделювання кількості спринтів для досягнення MVP та використання багатокритеріальної моделі оцінювання задач беклогу в контексті розробки OMS.

Практична значущість результатів полягає в можливості їх використання як методичної основи для планування, організації та контролю розробки систем обробки замовлень у банківських, брокерських та фінтех-проектах. Запропоновані продуктова концепція «AssetFlow», набір артефактів управління та рекомендації з удосконалення процесу можуть бути адаптовані у інших проектах, орієнтованих на український ринок капіталу, а також слугувати базою для подальшої технічної реалізації й розширення функціональності системи.

З урахуванням викладеного можна зробити висновок, що мета кваліфікаційної роботи досягнута, а поставлені завдання вирішені в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Москв'як Я. Є. Фінансові ринки як особливе середовище для ухвалення фінансових рішень. *Наука і техніка сьогодні. Серія «Економіка»*. 2024. № 13(41). С. 294–304. DOI: [https://doi.org/10.52058/2786-6025-2024-13\(41\)-294-304](https://doi.org/10.52058/2786-6025-2024-13(41)-294-304).Кривов'язюк І.В. Проблеми та перспективи розвитку фондового ринку України в сучасних умовах. Сучасний стан та перспективи розвитку фондового ринку України: матеріали V Міжнародної науково-практичної конференції (27 червня 2025 р., Дніпро, Україна). Вінниця: ТОВ «УКРЛОГОС Груп», 2025. URL: <https://archives.mcnd.org.ua/index.php/conference-proceeding/article/download/845/849>
2. Національна комісія з цінних паперів та фондового ринку. Аналітична довідка за II квартал 2024 року. URL: https://www.nssmc.gov.ua/wp-content/uploads/2025/04/analitychna-dovidka_2kv24_.pdf
3. Голюк В.Я., Кузьмінський В.З., Чумаченко О.Г. Сучасний стан фондового ринку України та перспективи його розвитку. *Економіка та суспільство*. 2024. № 64. С. 464–469. DOI: <https://doi.org/10.32782/2524-0072/2024-64-96>
4. Котик О.В., Клим'юк Ю.Ю. Особливості випуску та розміщення державних цінних паперів. *Вісник Національного університету водного господарства та природокористування. Серія «Економічні науки»*. 2023. Вип. 1(101). С. 42–54. DOI: <https://doi.org/10.31713/ve120235>
5. Національна комісія з цінних паперів та фондового ринку. Аналітична довідка за II квартал 2025 року. URL: <https://www.nssmc.gov.ua/wp-content/uploads/2025/09/ii-kvartal.pdf>

6. Національний банк України. Стратегія розвитку фінансового сектору України до 2025 року (припинена у 2023 році). URL: <https://bank.gov.ua/ua/about/develop-strategy/develop-strategy2023>Климченко К.В. Напрями реформування ринку капіталу в Україні. Наукові праці НДФІ. 2018. № 1(82). С. 108–117. URL: https://npndfi.org.ua/docs/NP_18_01_108_uk.pdf
7. Національний банк України. Стратегія розвитку фінансового сектору України до 225 року. Київ: Національний банк України, 2020. 102 с. URL: https://bank.gov.ua/admin_uploads/article/Strategy_FS_2025_2020.pdf
8. Manifesto for Agile Software Development. URL: <https://agilemanifesto.org/>
9. Association for Financial Markets in Europe. Adopting Agile Work Practices at Scale in European Capital Markets. 2020. URL: <https://www.afme.eu/media/n52ek254/afmeagiletransformation202006.pdf>
10. Schwaber K., Sutherland J. The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. 2020. URL: <https://scrumguides.org/scrum-guide.html>
11. Atlassian. 3 pillars of Scrum. URL: <https://www.atlassian.com/agile/project-management/3-pillars-scrum>
12. Manifesto for Agile Software Development: Principles. URL: <https://agilemanifesto.org/principles.html>
13. Abu Irtaymah H., Shaari H., Ahmed N. Agile Project Management Approaches: A Case Study with Respect to Their Application in Finance Technology Projects. Sebha University Journal of Pure & Applied Sciences (JOPAS). 2022. Vol. 21, No. 4. Pp. 25–29. DOI: <https://doi.org/10.51984/JOPAS.v21i4.2118>
14. Atlassian. Product roadmaps. URL: <https://www.atlassian.com/agile/product-management/product-roadmaps>

15. Балдик Д., Мушинський О., Спічак Р. Застосування NLP моделей в управлінні бізнес-процесами електронної комерції будівельних матеріалів: моделювання та статистичний аналіз. *Вчені записки Університету «КРОК»*. 2025. № 2(78). С. 297–305. DOI: <https://doi.org/10.31732/2663-2209-2025-78-297-305>.

16. Runsewe O., Osundare O. S., Olaoluwa S., Folorunsho L. A. A. End-to-end systems development in agile environments: Best practices and case studies from the financial sector. *International Journal of Engineering Research and Development*. 2024. Vol. 20, no. 08. P. 522–529.

ДОДАТКИ

Додаток А

Таблиця 1.1 – Основні інструменти фондового ринку України

Інструмент	Коротка сутність	Основні емітенти	Типові інвестори	Роль на українському ринку
Акції	Пайові цінні папери, що підтверджують участь у статутному капіталі та право на дивіденди	Акціонерні товариства, банки, промислові та сервісні компанії	Інституційні інвестори, інвестиційні фонди, досвідчені приватні інвестори	Сегмент із невисокою біржовою ліквідністю, але значним потенціалом зростання в період відбудови економіки [1]
ОВДП (державні облигації)	Боргові папери, що підтверджують зобов'язання держави сплатити купон та погасити номінал	Держава в особі Міністерств а фінансів України	Банки, страхові компанії, інвестиційні фонди, роздрібні інвестори	Домінуючий інструмент за обсягами торгів, орієнтир безризикової дохідності в національній валюті [3]
Корпоративні облигації	Боргові папери компаній, що використовуються для залучення довгострокового фінансування	Приватні та державні корпорації, банки	Інституційні інвестори, досвідчені приватні інвестори	Менша частка ринку, вищий кредитний ризик порівняно з ОВДП, важливий інструмент фінансування бізнесу [4]
Муніципальні облигації	Облігації органів місцевого самоврядування, спрямовані на фінансування регіональних проектів	Міські, обласні ради та інші органи місцевого самоврядування	Банки, фонди, інвестори, орієнтовані на інфраструктурні проекти	Поки що невеликий, але перспективний сегмент для фінансування відбудови громад [5]
Деривативи (ф'ючерси, опціони тощо)	Похідні інструменти, вартість яких залежить від ціни базового активу	Біржі, клірингові організації, інколи банки	Професійні учасники, спекулянти, хеджери	Ринок перебуває на етапі становлення; розвиток деривативів визначено одним із пріоритетів регулятора [2]