

**СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ**

**Факультет інженерії
Кафедра дизайну та індустрії моди**

ПОЯСНЮВАЛЬНА ЗАПИСКА

**до кваліфікаційної роботи
II освітнього рівня магістр**

спеціальності 022 Дизайн

освітньої програми Графічний дизайн і фешн індустрія

на тему **ПРОЄКТНИЙ МЕНЕДЖМЕНТ У СФЕРІ
ВЕБРОЗРОБКИ З ВИКОРИСТАННЯМ ПЛАТФОРМ
ДЛЯ УПРАВЛІННЯ ЗАДАЧАМИ**

Виконала здобувачка
вищої освіти групи ГрД-24зм

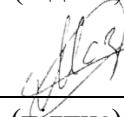
Анжела МИЦИК

(ім'я та ПРИЗВИЩЕ)



(підпис)

Керівник к.т.н., Євген МАЗНЄВ
(науковий ступінь, ім'я та ПРИЗВИЩЕ)



(підпис)

Завідувачка кафедри к.т.н., Галина РІПКА
(науковий ступінь, ім'я та ПРИЗВИЩЕ)



(підпис)

Рецензент к.т.н., Сергій КУДРЯВЦЕВ
(науковий ступінь, ім'я та ПРИЗВИЩЕ)



(підпис)

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет інженерії
Кафедра дизайну та індустрії моди
Освітній рівень магістр
Галузь знань 02 Культура та мистецтво
(шифр і назва)
Спеціальність 022 Дизайн
(шифр і назва)
Спеціалізація 022.01 Графічний дизайн
(шифр і назва)
освітня програма Графічний дизайн і фешн індустрія

ЗАТВЕРДЖУЮ
Завідувачка кафедри
 Галина РІПКА
22 грудня 2025 р.

ЗАВДАННЯ
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Мицик Анжела Юріївна
(прізвище, ім'я, по батькові)

1. Тема:

Проектний менеджмент у сфері веброзробки з використанням платформ для
управління задачами
спеціальне завдання:

керівник роботи Мазнів Євген Олександрович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання здобувачем вищої освіти роботи 10.12.2025 р.

3. Вихідні дані до роботи:

Компанії малого та середнього бізнесу у сфері аутсорсингу веброзробки
Основні РМ-платформи для аналізу
Системами контролю версій: Git
Інструментами комунікації: Microsoft Teams

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

ВСТУП

РОЗДІЛ 1.

РОЗДІЛ 2.

ЗАГАЛЬНІ ВИСНОВКИ

5. Перелік графічного матеріалу (слайдів презентації):

1. Титульний, 2. Мета роботи. 3. Методи дослідження. 4.

5. 6.

7. 8.

9. 10.

11. 12.

13. Загальні висновки

6. Дата видачі завдання 08.09.2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проектування	Термін виконання етапів	Примітка
1	Вибір та затвердження теми магістерської роботи	08.09.25	
2	Аналіз наукової літератури відповідно до обраної теми	09.10.25	
3	Написання та затвердження плану магістерської роботи	30.10.25	
4	Вступ	01.11.25	
5	Розділ 1	06.11.25	
6	Розділ 2	20.11.25	
7	Формулювання та оформлення загальних висновків	01.12.25	
8	Анотація до роботи	05.12.25	
9	Подача оформленої роботи на перевірку	10.12.25	

Здобувач
освіти

вищої



(підпис)

Анжела
МИЦИК

(ім'я та ПРІЗВИЩЕ)

Керівник роботи



(підпис)

Євген
МАЗНЄВ

(ім'я та ПРІЗВИЩЕ)

АНОТАЦІЯ

Мицик А.Ю. Проєктний менеджмент у сфері веброзробки з використанням платформ для управління задачами, 2025, загальний обсяг роботи становить 40 сторінок.

Магістерська робота присвячена дослідженню та вдосконаленню процесів проєктного менеджменту у сфері веброзробки з фокусом на інтеграцію графічного дизайну в Agile-середовище. Актуальність теми зумовлена необхідністю подолання розсинхронізації між творчими циклами створення візуального контенту та інженерними спринтами розробки.

У роботі проведено порівняльний аналіз провідних платформ управління задачами (Jira, Trello, Asana) за критеріями візуальної ергономіки та можливостей Design Ops. Обґрунтовано неефективність використання неспеціалізованих інструментів для комунікації в сучасних дизайн-командах.

Основним завданням стала розробка адаптивної моделі управління дизайн-спринтом (ADSM), яка включає структурований Workflow із вбудованими «воротами якості» (Quality Gates) та схему безшовної технічної інтеграції (Figma + PM-платформа).

Експериментальна апробація моделі на реальному вебпроєкті підтвердила її ефективність: зафіксовано зростання показника Design Velocity на 57 % та скорочення часу отримання зворотного зв'язку на 85 %. Доведено економічну доцільність впровадження розробленої методики для дизайн-студій та ІТ-компаній.

Ключові слова: проєктний менеджмент, Веброзробка, Agile, Scrum, Kanban, PM-платформи, Jira, Trello, Velocity, Lead Time, Git-інтеграція

ANNOTATION

Mitsyk A.Y. Project management in web development using task management platforms, 2025, total volume of work is 40 pages.

The master's thesis is devoted to the study and improvement of project management processes in web development with a focus on the integration of graphic design into the Agile environment. The relevance of the topic is due to the need to overcome the desynchronization between the creative cycles of visual content creation and engineering development sprints.

The thesis provides a comparative analysis of leading task management platforms (Jira, Trello, Asana) based on the criteria of visual ergonomics and Design Ops capabilities. It substantiates the ineffectiveness of using non-specialized tools for communication in modern design teams.

The main task was to develop an adaptive design sprint management model (ADSM), which includes a structured workflow with built-in quality gates and a seamless technical integration scheme (Figma + PM platform).

Experimental testing of the model on a real web project confirmed its effectiveness: a 57% increase in Design Velocity and an 85% reduction in feedback time were recorded. The economic feasibility of implementing the developed methodology for design studios and IT companies has been proven.

Keywords: project management, web development, Agile, Scrum, Kanban, PM platforms, Jira, Trello, Velocity, Lead Time, Git integration

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ОСНОВИ ІНТЕГРАЦІЇ ДИЗАЙН-ПРОЦЕСІВ У СИСТЕМУ ПРОЄКТНОГО УПРАВЛІННЯ	9
1.1. Аналіз сучасних методологій управління проєктами та їх адаптація до креативного циклу веброзробки	9
1.2. Огляд та порівняльний аналіз платформ управління задачами з позиції потреб графічного дизайну	17
1.3. Порівняльний аналіз за критеріями Design Ops	19
ВИСНОВКИ ДО РОЗДІЛУ 1.....	23
РОЗДІЛ 2. ПРОЄКТУВАННЯ ЕФЕКТИВНОЇ МОДЕЛІ УПРАВЛІННЯ ДИЗАЙН-СПРИНТОМ	24
2.1. Розробка моделі управління Design Ops: створення структури проєкту на обраній РМ-платформі	24
2.2. Оцінка ефективності впровадженої моделі на прикладі веброзробки	27
ВИСНОВКИ ДО РОЗДІЛУ 2.....	33
ЗАГАЛЬНІ ВИСНОВКИ.....	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36
ДОДАТКИ.....	40

ВСТУП

Актуальність теми зумовлена глибокою трансформацією бізнес-процесів в ІТ-галузі та нагальною потребою у підвищенні ефективності та прозорості управління проєктами. Сучасний ринок вимагає від команд веброзробки максимальної гнучкості, швидкої реакції на зміни вимог замовника та скорочення циклу випуску продукту, що можливо лише в рамках Agile-методологій (Scrum, Kanban). У цьому контексті використання неспеціалізованих інструментів (як-от Google Sheets, електронні пошти) призводить до критичних втрат часу на адміністрування, ускладнює моніторинг залежностей та унеможливорює точне прогнозування термінів, що безпосередньо впливає на нестабільність ключового показника — Velocity. Таким чином, актуальною стає науково-практична задача обґрунтування, вибору та розробки адаптованої моделі інтеграції професійних РМ-платформ (Jira, Trello, Asana) у робочий процес. Дослідження надасть чітке економічне обґрунтування інвестицій у ці інструменти, довівши, що автоматизація збору даних та безшовна інтеграція з технічними системами (Git) істотно знижує адміністративні витрати Project Manager та підвищує загальну успішність та якість веброзробних проєктів, що є критичним для конкурентоспроможності компаній малого та середнього бізнесу.

Метою кваліфікаційної роботи є розробка та обґрунтування комплексної моделі ефективного управління проєктами веброзробки шляхом адаптації Agile-методологій та інтеграції РМ-платформ, що підвищить успішність командної роботи.

Об'єктом дослідження є процеси управління проєктами у середовищі компаній, що спеціалізуються на веброзробці.

Предметом дослідження є методичні підходи, інструменти (РМ-платформи) та метрики оцінки ефективності управління проєктами, адаптовані до специфіки веброзробки

Для досягнення мети було поставлено такі **завдання**:

1. Проаналізувати сучасний стан проєктного менеджменту у веброзробці, особливості застосування гнучких методологій (Scrum, Kanban) та вимоги до РМ-платформ.

2. Провести порівняльний аналіз функціональних можливостей ключових РМ-платформ (Jira, Trello) за критеріями підтримки Agile-процесів та інтеграції з технічними інструментами (Git).

3. Розробити комплексну модель інтеграції обраної РМ-платформи у робочі процеси, включаючи схему налаштування, планування та моніторингу прогресу.

4. Експериментально оцінити ефективність розробленої моделі, провівши порівняльний аналіз ключових показників з базовим станом на прикладі пілотного проєкту.

5. Надати практичні рекомендації щодо впровадження моделі та обґрунтувати її економічну доцільність для ІТ-компаній.

Практичне значення роботи полягає у розробці та впровадженні структурованої системи управління творчими процесами, що є критичним для забезпечення якості та термінів у дизайн-проєктах.

Структура кваліфікаційної роботи складається зі вступу, двох розділів, висновків, списку використаної літератури, додатків.

У першому розділі проведено аналіз специфіки застосування гнучких методологій (Agile, Scrum, Kanban) у сфері графічного дизайну та веброзробки. Виявлено проблему розсинхронізації ритмів роботи дизайнерів та розробників, для вирішення якої обґрунтовано доцільність використання підходу Dual-Track Agile. Здійснено порівняльний аналіз функціональних можливостей РМ-платформ (Jira, Trello, Asana) за критеріями візуальної ергономіки, інтеграції з графічними редакторами та наявності інструментів для рецензування макетів (Proofing).

У другому розділі розроблено авторську адаптивну модель ADSM (Adaptive Design Sprint Model), яка структурує творчий процес через систему статусів та «воріт якості» (Quality Gates). Спроектовано схему технічної інтеграції інструментарію (Figma + РМ-платформа + Slack) для реалізації принципу «Єдиного джерела істини». Проведено експериментальну апробацію моделі на пілотному вебпроєкті, яка довела її ефективність: зафіксовано зростання показника Design Velocity на 57%, скорочення часу отримання зворотного зв'язку на 85% та підтверджено економічну доцільність впровадження. Сформульовано загальні висновки по роботі.

РОЗДІЛ 1. ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ОСНОВИ ІНТЕГРАЦІЇ ДИЗАЙН-ПРОЦЕСІВ У СИСТЕМУ ПРОЄКТНОГО УПРАВЛІННЯ

1.1 Аналіз сучасних методологій управління проєктами та їх адаптація до креативного циклу веброзробки

Сучасна парадигма веброзробки характеризується відходом від лінійних моделей (Waterfall) на користь ітеративних підходів, що зумовлено високою невизначеністю вимог на старті проєкту та необхідністю швидкого реагування на зміни ринкової кон'юнктури. Для фахівця з графічного дизайну та UX/UI це означає фундаментальну зміну робочого процесу: замість створення завершеного макету всього сайту перед початком кодування, дизайнер стає учасником безперервного циклу розробки. У цьому контексті найбільш кращими методологіями є Scrum та Kanban, проте їх застосування у креативному середовищі вимагає специфічної адаптації, відомої як концепція Design Operations (Design Ops).

Scrum – це не просто методологія, а фреймворк (каркас) для управління складними продуктами, який базується на принципах Agile. Він дозволяє командам вирішувати нетипові задачі через адаптивний підхід, доставляючи цінність продукту ітеративно (частинами) та інкрементально (з нарощуванням функціоналу) (рис. 1.1).

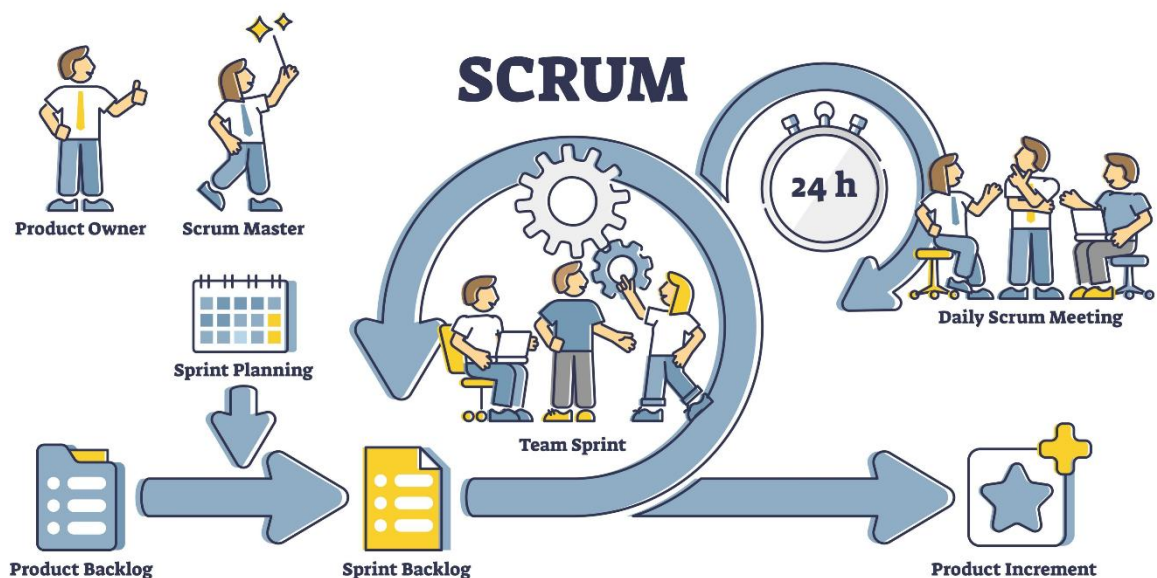


Рис. 1.1. Методологія Scrum: сутність, ролі та процеси

Автори фреймворку, Кен Швабер та Джефф Сазерленд, визначають Scrum як легкий для розуміння, але важкий для досконалого опанування інструмент, що базується на теорії емпіричного управління процесами.

1. Три стовпи Scrum (Теорія емпіризму)

Ефективність Scrum базується на трьох фундаментальних принципах, які мають бути реалізовані у кожному вебпроєкті:

1. Прозорість (Transparency): Значущі аспекти процесу мають бути видимими для тих, хто відповідає за результат. У контексті веброзробки це означає, що Backlog (список задач) та статус макетів у РМ-платформі (Jira/Trello) мають бути доступні всім учасникам команди.
2. Інспекція (Inspection): Користувачі Scrum повинні часто перевіряти артефакти Scrum і прогрес на шляху до мети, щоб виявити небажані відхилення.
3. Адаптація (Adaptation): Якщо інспекція виявила проблеми (наприклад, дизайн не відповідає технічним обмеженням верстки), процес або матеріали повинні бути скориговані якнайшвидше.

2. Команда Scrum (Scrum Team)

У Scrum немає суворої ієрархії. Це самоорганізована одиниця, що складається з трьох ролей:

Власник Продукту (Product Owner, PO): відповідає за максимізацію цінності продукту та управління Беклогом Продукту. Це особа, яка комунікує із замовником, збирає вимоги до сайту, розставляє пріоритети (що малюємо спочатку: головну сторінку чи кошик) та приймає роботу команди.

Scrum-майстер (Scrum Master): слуга-лідер (Servant Leader), який допомагає команді зрозуміти та застосовувати Scrum. Він усуває перешкоди (impediments), фасилітує зустрічі та захищає команду від зовнішнього втручання.

Розробники (Developers): люди, які безпосередньо створюють Інкремент (готовий продукт).

У Scrum термін "Developers" означає всіх, хто працює над продуктом. У вашому випадку сюди входять UI/UX дизайнери, Frontend/Backend розробники та QA-інженери.

3. Артефакти Scrum (Artifacts) – це матеріальні результати роботи, які забезпечують прозорість (рис. 1.2).

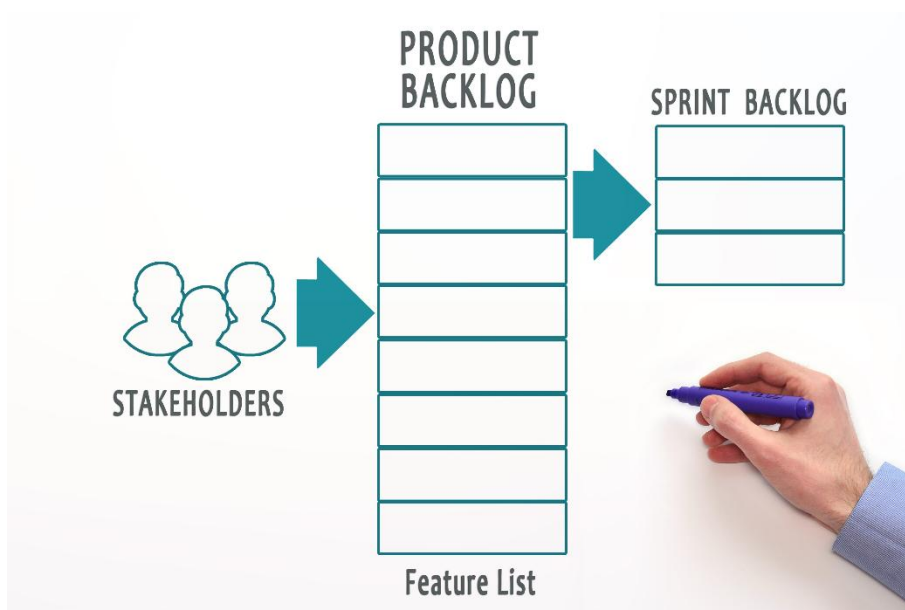


Рис. 1.2. Артефакти Scrum

1. Беклог Продукту (Product Backlog) впорядкований список усього, що може знадобитися у продукті. Це єдине джерело вимог. Для дизайнера це список усіх сторінок, станів кнопок та адаптивів, які треба намалювати.
2. Беклог Спринту (Sprint Backlog) набір елементів з Беклогу Продукту, відібраних для виконання у поточному Спринті, плюс план того, як створити Інкремент.
3. Інкремент (Increment) сума всіх виконаних елементів беклогу за поточний та попередні спринти. Для вебпроєкту це може бути не просто набір макетів, а, наприклад, повністю зверстана та працююча частина сайту.

Всі події у Scrum мають часові рамки (time-boxed), тобто максимальну тривалість. Спринт (The Sprint): "Серце" Scrum. Це фіксований відрізок часу (зазвичай 1–4 тижні), протягом якого створюється готовий до випуску Інкремент. Планування Спринту (Sprint Planning): Зустріч на початку спринту, де команда визначає: Що можна зробити? Як це буде зроблено? Дизайнери тут оцінюють складність макетів. Щоденний Scrum (Daily Scrum / Stand-up): 15-хвилинна зустріч для синхронізації дій розробників та дизайнерів. Кожен відповідає на 3 питання: Що я робив вчора? Що

робитиму сьогодні? Які є перешкоди?

Огляд Спринту (Sprint Review): Демонстрація результатів роботи стейкхолдерам. Дизайнери показують затверджені макети, розробники – працюючий функціонал.

Ретроспектива Спринту (Sprint Retrospective): Зустріч після огляду, де команда обговорює процес: що пішло не так у комунікації, як покращити використання РМ-платформи тощо. Критерії готовності (Definition of Done - DoD) це спільне розуміння того, що означає "робота завершена". Для дизайн-задачі DoD може включати: Макет затверджено арт-директором. Створено адаптивні версії (Mobile, Tablet, Desktop). Усі асети експортовані у Zeplin/Figma для розробників. Отримано погодження від розробника (технічна реалізованість).

РМ-платформи (Jira, Trello) слугують цифровим середовищем для реалізації цього фреймворку. Вони візуалізують Беклог, забезпечують прозорість під час Спринту та допомагають проводити Інспекцію через автоматизовані звіти (Burndown Charts).

Kanban (з японської «рекламний щит» або «вивіска») – це метод управління інтелектуальною працею, який фокусується на візуалізації робочого процесу, обмеженні кількості незавершеної роботи (WIP) та оптимізації потоку виконання задач. На відміну від Scrum, який ділить роботу на фіксовані часові інтервали (спринти), Kanban є поточною (flow-based) методологією, що дозволяє змінювати пріоритети "на льоту" та випускати готовий продукт у міру його готовності.

Цей метод еволюціонував із виробничої системи Toyota (Lean Manufacturing) і чудово підходить для дизайн-команд, які працюють у режимі підтримки продукту або мають потік нерівномірних за складністю запитів.

Базові принципи Kanban. Kanban не вимагає революційної зміни процесів з першого дня. Він базується на еволюційному підході:

1. Почніть з того, що є зараз: Не потрібно змінювати ролі (як у Scrum) чи переписувати всі процеси. Просто візуалізуйте поточний стан справ.
2. Домовтеся про еволюційний розвиток: Зміни впроваджуються поступово, щоб уникнути опору команди.

3. Поважайте поточні ролі та обов'язки: Це знижує страх змін.
4. Лідерство на всіх рівнях: Ініціатива щодо покращення процесу може виходити від звичайного дизайнера, а не лише від менеджера.

Ефективність методу забезпечується дотриманням шести практик:

1. Візуалізація (Visualize): Ви не можете керувати тим, чого не бачите. Основою є Канбан-дошка (фізична або цифрова в Trello/Jira). Кожна задача — це картка, кожен етап роботи – стовпчик (наприклад, *Backlog* → *Design* → *Review* → *Done*). Це дозволяє миттєво побачити, на якому етапі знаходиться кожен макет.
2. Обмеження незавершеної роботи (Limit WIP — Work In Progress): Це ключова відмінність Kanban. Встановлюються ліміти на кількість карток у кожному стовпчику (наприклад, не більше 2-х макетів у колонці "В роботі" на одного дизайнера). Це змушує команду завершувати почате ("Stop starting, start finishing") перед тим, як брати нове, і запобігає перевантаженню та вигоранню.
3. Управління потоком (Manage Flow): Мета — забезпечити рівномірний рух карток зліва направо. Якщо картки накопичуються в колонці "Review", це сигнал про "вузьке місце" (bottleneck). Менеджер має перерозподілити ресурси, щоб "розкоркувати" цей етап.
4. Явні правила процесу (Make Process Policies Explicit): Правила гри мають бути чіткими. Наприклад: "Що саме має бути зроблено, щоб картку можна було перемістити з *Design* у *Ready for Dev*?" (наявність усіх станів кнопок, експорт svg-іконок тощо).
5. Петлі зворотного зв'язку (Feedback Loops): Регулярні зустрічі для синхронізації (каденції). У Kanban це може бути щоденний стендап біля дошки, огляд надання сервісу (Service Delivery Review) та зустрічі з поповнення черги (Replenishment).
6. Спільне покращення (Improve Collaboratively): Використання наукового підходу та метрик для постійного вдосконалення (Kaizen).

У Kanban ефективність вимірюється не в Story Points (як у Scrum), а в часових показниках потоку (рис 1.3).

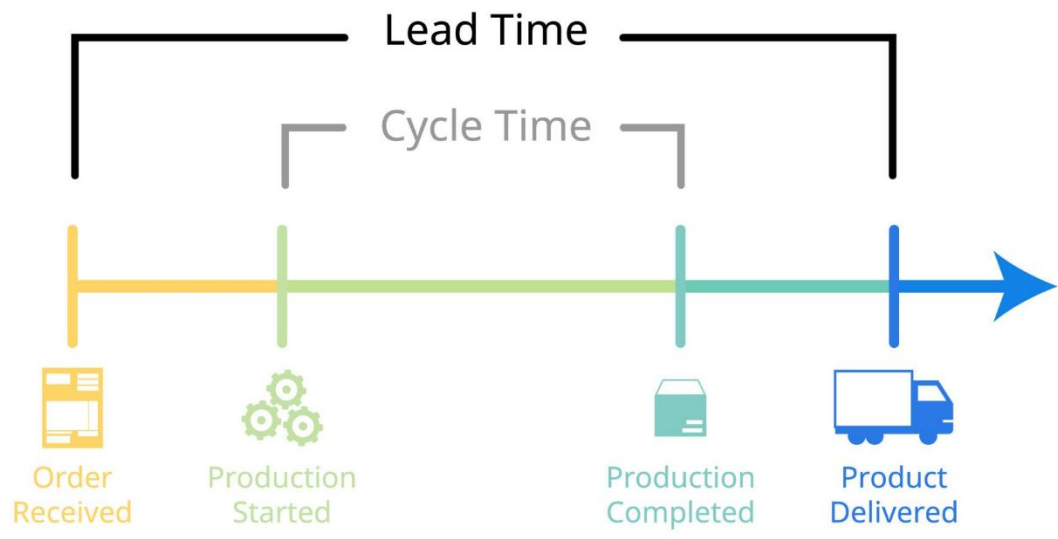


Рис. 1.3. Ключові метрики Kanban

- 1. Lead Time (Час виконання): Загальний час від моменту появи запиту (ідеї) до моменту його повної реалізації. Це те, що цікавить замовника ("Як швидко я отримаю дизайн?").
- 2. Cycle Time (Час циклу): Час, протягом якого команда безпосередньо працює над задачею (від переміщення в "In Progress" до "Done"). Це показник продуктивності команди.
- 3. Throughput (Пропускна здатність): Кількість задач, завершених за одиницю часу (наприклад, 10 банерів на тиждень).
- 4. Cumulative Flow Diagram (CFD): Графік, що показує стабільність потоку та допомагає прогнозувати терміни.

Таблиця 1.1

Kanban vs Scrum у дизайні

Характеристика	Scrum	Kanban
Ритм роботи	Спринти (фіксований час, 2 тижні)	Безперервний потік (Continuous Flow)
Зміни	Небажані під час	Можливі в будь-який

Характеристика	Scrum	Kanban
	спринту	момент (якщо є слот)
Ролі	Чіткі (SM, PO, Dev Team)	Необов'язкові (зберігаються поточні)
Метрика	Velocity (швидкість)	Lead Time (час проходження)
Для кого краще	Розробка нового продукту (Project)	Підтримка, дрібні задачі, дизайн-студії (Service)

Kanban часто використовується для етапу Design Support або для команд, які обслуговують багато дрібних запитів від маркетингу (банери, лендінги). У вашій моделі ADSM елементи Kanban (візуалізація, ліміти WIP) інтегровані всередину спринтів для кращого контролю потоку задач.

Ключовою проблемою інтеграції дизайну в Agile є розсинхронізація ритмів роботи розробників та дизайнерів. Розробка, як правило, оперує чіткими спринтами (фіксованими відрізками часу), тоді як творчий пошук рішень (Ideation phase) важко піддається жорсткому таймінгу. Для вирішення цього протиріччя в управлінні вебпроектами застосовується модель Dual-Track Agile («Двоколіїний Agile»), де процеси дослідження та дизайну (Discovery) випереджають процеси технічної реалізації (Delivery) на один або кілька спринтів.

На рисунку 1.4 візуалізовано паралельну роботу двох потоків. Дизайнерська команда працює в потоці Discovery, перевіряючи гіпотези та створюючи макети, які після затвердження передаються в Backlog розробки (Delivery) (рис. 1.4).

Ефективність такої взаємодії можна оцінити через показник Design Velocity (V_d), який розраховується як обсяг затверджених дизайн-макетів, переданих у розробку за одиницю часу. Для математичного опису трудомісткості дизайн-задач у середовищі Agile пропонується використовувати формулу розрахунку зваженої складності (W_c):

$$W_c = (T_e \cdot K_u) + \sum_{i=1}^n T_{fb_i} \quad (1.1)$$

де: T_e – оціночний час виконання задачі дизайнером (в годинах або Story Points); K_u – коефіцієнт невизначеності вимог (1,0 – чітке ТЗ, 1,5 – розмиті вимоги, 2,0 – пошуковий концепт); T_{fb} – час, витрачений на ітерації зворотного зв'язку (Feedback Loop) та внесення правок.

Ця формула демонструє, що в управлінні дизайн-проектами критичним фактором є не лише час безпосереднього малювання, а й комунікаційна складова (правки), яка в РМ-платформах часто ігнорується, що призводить до зриву дедлайнів. Інтеграція методології Kanban дозволяє нівелювати цей ризик шляхом візуалізації "вузьких місць" (bottlenecks) на етапі затвердження макетів артдиректором або замовником.

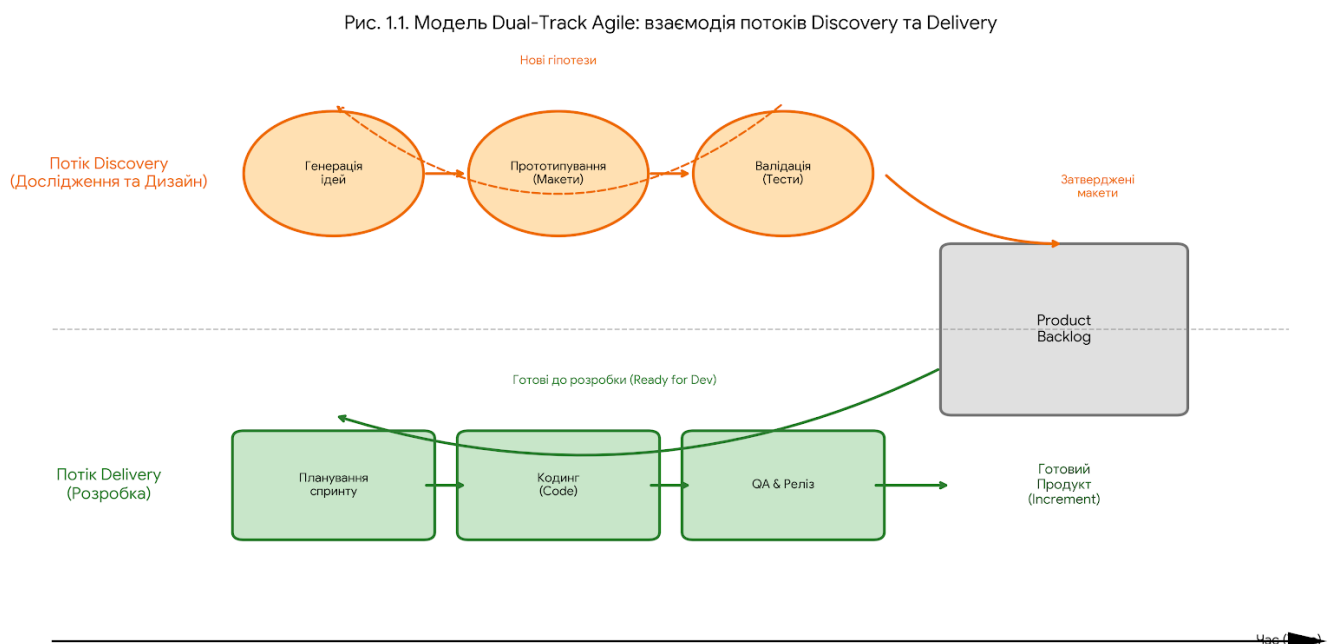


Рис. 1.4. Модель Dual-Track: взаємодія потоків Discovery та Delivery

Рисунок 1.4 демонструє, як дизайн-процес інтегрується в інженерний цикл, не блокуючи його.

1.2. Огляд та порівняльний аналіз платформ управління задачами з позиції потреб графічного дизайну

Вибір інструментарію для управління проєктами (PM-платформи) у сфері веброзробки є критичним фактором, що визначає швидкість комунікації (Time-to-Market) та якість кінцевого візуального продукту. На ринку існує широкий спектр рішень, проте не всі вони адаптовані до специфіки роботи графічного дизайнера, для якого пріоритетними є візуалізація завдань, підтримка графічних файлів великого обсягу та інтеграція з профільним софтом (Figma, Sketch, Adobe CC).

У рамках дослідження проведено порівняльний аналіз трьох найбільш поширених платформ: Jira Software, Trello та Asana.

Jira Software є галузевим стандартом для великих команд розробки. Її архітектура дозволяє налаштовувати складні робочі процеси (Workflows) з автоматичними переходами статусів. Проте для дизайнерів інтерфейс Jira часто є перевантаженим технічною інформацією, що знижує UX самої системи управління. Trello, що базується на чистій методології Kanban, забезпечує максимальну наочність. Карткова система ідеально підходить для візуального відображення станів макетів, проте функціонал Trello обмежений у можливостях звітності та складного планування залежностей між екранами інтерфейсу. Asana займає проміжну позицію, пропонуючи гнучкість у перемиканні між режимами відображення (Список, Дошка, Таймлайн), що дозволяє дизайнеру бачити свої завдання в зручному форматі, а менеджеру – контролювати загальний графік Ганта.

Результати порівняльного аналізу функціональних можливостей платформ зведено у таблицю 1.2.

Таблиця 1.2.

Порівняльна характеристика PM-платформ для дизайн-команд

Критерій порівняння	Jira Software	Trello	Asana
Тип візуалізації	Scrum/Kanban дошки, Списки	Тільки Kanban дошки	Список, Дошка, Таймлайн, Календар

Критерій порівняння	Jira Software	Trello	Asana
Інтеграція з Figma	Висока (плагіни, embed-фрейми)	Середня (посилання, вкладення)	Висока (двостороння синхронізація)
Управління версіями макетів	Реалізовано через додаткові плагіни	Обмежено (коментарі)	Вбудований функціонал (Proofing)
Крива навчання (Learning Curve)	Висока (складно для новачків)	Низька (інтуїтивно зрозуміло)	Середня
Вартість (для малих команд)	Висока	Умовно безкоштовна	Середня
Design Ops відповідність	Для масштабних Enterprise-проектів	Для фрілансу та малих студій	Для дизайн-студій середнього розміру

Окрему увагу слід приділити процесу передачі макетів у розробку (Design Handoff). Ефективність цього етапу в РМ-платформі визначається рівнем автоматизації. Сучасні системи дозволяють налаштувати тригери: наприклад, зміна статусу задачі в Jira на "Ready for Dev" автоматично надсилає сповіщення розробнику в Slack з посиланням на конкретний фрейм у Figma.

Ефективність впровадження платформи (E_{pm}) можна оцінити за формулою скорочення операційного часу:

$$E_{pm} = \frac{T_{sync_before} - T_{sync_after}}{T_{sync_before}} \cdot 100\% \quad (1.2)$$

де T_{sync} – сумарний час, витрачений командою на синхронізацію статусів, пошук актуальних версій макетів та з'ясування вимог до та після впровадження платформи.

На схемі (рис. 1.2, яка буде розроблена у практичній частині) показано ідеалізований потік даних: Дизайнер оновлює статус у плагіні Figma > Задача оновлюється в Jira > Менеджер бачить зміну на діаграмі згорання задач (Burndown Chart).

Такий підхід мінімізує "людський фактор" та забезпечує прозорість процесу, що є головною метою проєктного менеджменту у веброботі.

В умовах сучасної диджитал-індустрії графічний дизайн перестав бути ізольованим процесом створення статичних зображень і трансформувався у складну систему взаємодії, що включає етапи дослідження, прототипування, тестування та інтеграції візуальних активів у програмні продукти. Ефективність управління цими процесами безпосередньо залежить від обраного інструментарію — платформ управління проєктами (Project Management Platforms, далі — РМ-платформи). Специфіка роботи дизайнера, на відміну від розробника програмного коду, вимагає від РМ-платформи наявності розвинених засобів візуалізації, підтримки роботи з графічними файлами великого обсягу та інструментів для ітеративного затвердження макетів (Proofing).

У роботі проведено аналіз трьох найбільш поширених на ринку платформ – Jira Software, Trello та Asana з метою визначення їхньої відповідності потребам дизайн-команд.

1.3. Порівняльний аналіз за критеріями Design Ops

Для об'єктивної оцінки платформ було виділено ряд критеріїв, критичних саме для графічного дизайну та Design Ops (операційного управління дизайном). Результати аналізу зведено у Таблицю 1.3.

Таблиця 1.3

Порівняльна характеристика РМ-платформ для дизайн-проєктів

Критерій порівняння	Jira Software	Trello	Asana
Тип візуалізації	Scrum/Kanban дошки, складні таблиці	Тільки Kanban дошки	Список, Дошка, Таймлайн, Календар

Критерій порівняння	Jira Software	Trello	Asana
User Experience (UX)	Низький (складний, технічний)	Високий (інтуїтивний)	Високий (сучасний, чистий)
Інтеграція з Adobe CC	Через сторонні плагіни	Обмежена	Висока (нативна інтеграція)
Proofing (Коментарі на макеті)	Ні (потрібні плагіни)	Ні	Так (вбудовано)
Управління дизайн-спринтами	Відмінно (Backlog, Epics)	Посередньо (вручну)	Добре (Timeline)
Придатність для веброзробки	Ідеально (синхронізація з кодом)	Низька (для простих сайтів)	Середня

Для формалізації процесу вибору оптимальної платформи (Port) розроблена математична модель вибору платформи для конкретного дизайн-проєкту. Пропонується використовувати метод зважених оцінок.

Інтегральний показник ефективності платформи (E_p) розраховується за формулою:

$$E_p = \sum_{i=1}^n (C_i \cdot W_i) \cdot K_{adpt} \quad (1.2)$$

де: C_i – бальна оцінка платформи за i -м критерієм (від 1 до 10), де критеріями виступають: візуалізація (C_v), інтеграція з дизайн-софтом (C_{int}), зручність комунікації (C_{com}), вартість (C_{cost}); W_i – ваговий коефіцієнт важливості критерію для конкретної команди (де $W_i = 1$). Наприклад, для команди UI-дизайнерів важливість інтеграції з розробкою (W_{dev}) буде вищою, ніж для ілюстраторів; K_{adpt} – коефіцієнт адаптивності інтерфейсу (ергономічність для візуального сприйняття).

Якщо застосувати цю модель для типової команди веброзробки (де важливі і візуалізація, і зв'язок з кодом), розрахунок показує наступний розподіл ефективності (рис. 1.5).

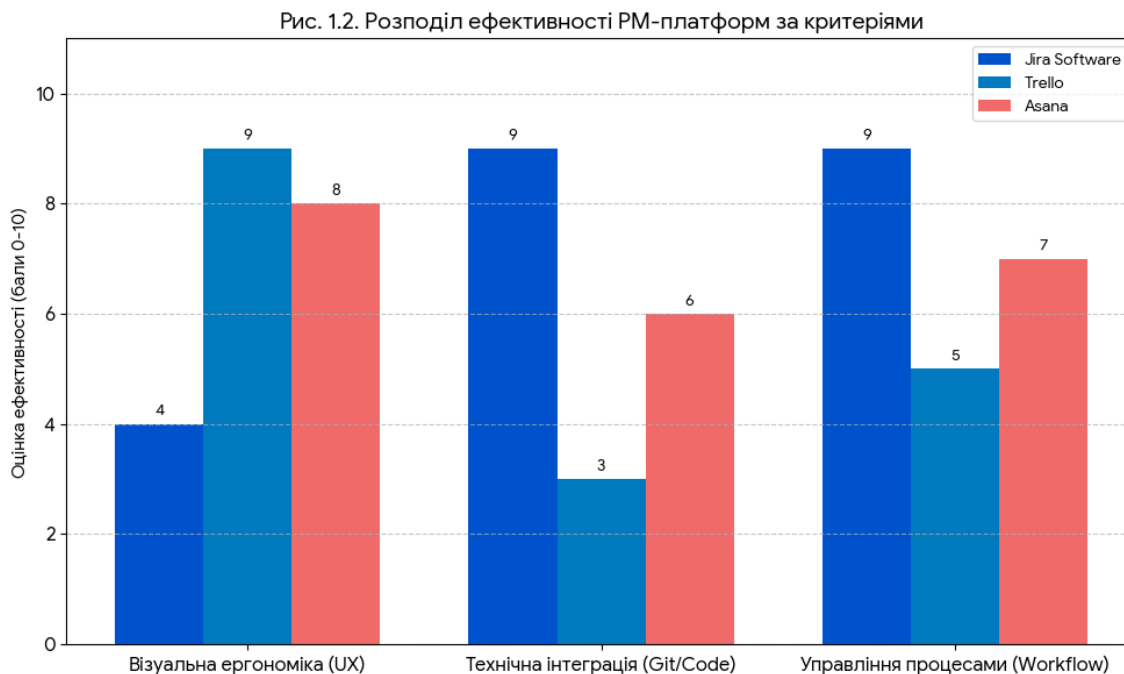


Рис. 1.5. Розподіл ефективності PM-платформ за критеріями

Аналіз графіку (рис. 1.5) демонструє, що Jira отримує найвищий бал за технічними критеріями, але програє в ергономіці. Trello лідирує в простоті, але не підходить для складних процесів. Asana демонструє найбільш збалансований результат для дизайн-команд, поєднуючи візуальну зручність з потужним функціоналом управління.

Проведений аналіз дозволяє стверджувати, що вибір платформи має базуватися на масштабі проєкту та специфіці дизайн-задач. Для задач веброзробки, де критичною є передача макету верстальнику (Design Handoff), найбільш доцільним є використання гібридної моделі: використання Jira для загального трекінгу проєкту в зв'язці зі спеціалізованими плагінами для дизайнерів, або використання Asana як основного інструменту для дизайн-команди з налаштованою синхронізацією статусів. Відсутність спеціалізованого інструментарію призводить до збільшення часу на

комунікацію («Time-to-Feedback») та зростання кількості помилок у фінальному інтерфейсі.

Теоретичний аналіз засвідчив, що просте копіювання інженерних підходів Agile на дизайн-процеси є неефективним без врахування специфіки творчої праці. Оптимальним рішенням є використання гібридних методологій (Dual-Track Agile) та вибір РМ-платформи, яка підтримує глибоку інтеграцію з дизайн-інструментами, забезпечуючи баланс між творчою свободою та управлінським контролем.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Встановлено, що пряме застосування інженерних методологій Agile (Scrum, Kanban) до креативних процесів є неефективним через розсинхронізацію ритмів роботи дизайнерів та розробників. Обґрунтовано необхідність використання гібридної моделі Dual-Track Agile (Discovery + Delivery), де дизайн-процес випереджає розробку, забезпечуючи перевірку гіпотез та готовність макетів до початку спринту.
2. Визначено, що ефективність дизайн-процесу (V_d) залежить не лише від часу, витраченого на створення макетів (T_e), але й від комунікаційної складової (зворотного зв'язку). Для кількісної оцінки трудомісткості дизайн-задач запропоновано використовувати формулу зваженої складності (W_c), яка враховує час на ітерації правок (T_{fb}) та коефіцієнт невизначеності (K_u).
3. Проведений порівняльний аналіз РМ-платформ (Jira, Trello, Asana) з позиції потреб графічного дизайну засвідчив, що ключовими критеріями вибору є не лише підтримка базових функцій Agile, але й висока якість візуалізації (Kanban), гнучкість інтеграції з дизайнерським софтом (Figma) та можливість ефективного управління версіями макетів і процесом затвердження (Proofing).
4. Головною метою інтеграції РМ-платформ є створення ефективної системи Design Operations (Design Ops), що забезпечує прозорість робочого процесу та мінімізує «людський фактор» на етапі Design Handoff (передачі макетів розробникам) через автоматизацію синхронізації статусів та даних між РМ-платформою, Figma та Git-системами.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ЕФЕКТИВНОЇ МОДЕЛІ УПРАВЛІННЯ ДИЗАЙН-СПРИНТОМ

2.1 Розробка моделі управління Design Ops: створення структури проєкту на обраній РМ-платформі

На основі теоретичного аналізу, проведеного у першому розділі, встановлено, що критичним фактором успішності дизайн-проєкту є не лише талант виконавця, а й архітектура робочого процесу. У цьому підрозділі розроблено авторську модель управління дизайн-спринтом, адаптовану для інтеграції в середовище веброзробки. Запропонована модель, яку надалі будемо називати ADSM (Adaptive Design Sprint Model), базується на принципах Design Ops і спрямована на стандартизацію етапів створення візуального контенту та автоматизацію рутинних операцій менеджера.

2.1.1. Архітектура робочого процесу (*Workflow Design*)

Фундаментом моделі є розроблена схема життєвого циклу дизайн-задачі (Issue Lifecycle). На відміну від стандартного циклу розробки («To Do» → «In Progress» → «Done»), креативний процес вимагає деталізації етапів валідації та затвердження.

Для реалізації в РМ-платформі (на прикладі Jira/Asana) спроектовано наступний ланцюжок статусів:

1. Backlog (Сховище ідей): Усі гіпотези, референси та вимоги стейкхолдерів.
2. To Do (Спринт): Задачі, відібрані для виконання у поточному тижневому циклі.
3. Discovery & Drafting (Пошук): Етап мудбордів, скетчів та низькорівневих прототипів (Lo-Fi).
4. Design in Progress (Розробка): Створення чистових макетів (Hi-Fi UI).
5. Internal Review (Арт-дирекшн): Етап перевірки якості (QA дизайну) арт-директором або лід-дизайнером.
6. Client/Stakeholder Approval: Зовнішнє затвердження. Критична точка, де часто виникають затримки.

7. Ready for Dev (Handoff): Фінальна підготовка макету, експорт ассетів, написання специфікацій.

8. Done: Задача передана в розробку.

Графічну схему запропонованого робочого процесу (Workflow) наведено на рисунку 2.1.

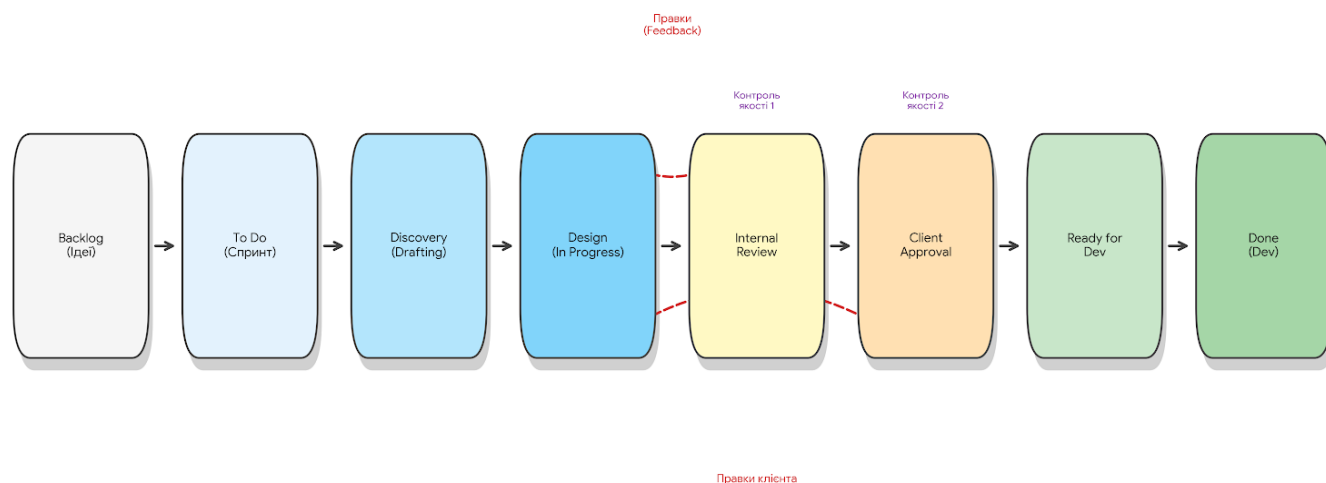


Рис. 2.1. Схема робочого процесу Design Ops Workflow

На схемі (рис. 2.1) виділено так звані «ворота якості» (Quality Gates) між етапами «Internal Review» та «Client Approval». Це контрольні точки, які унеможливають перехід задачі на наступний етап без виконання чеклиста (наприклад, перевірка відповідності сітці, наявність адаптивних версій тощо).

2.1.2. Структура та таймінг Дизайн-Спринту

Для синхронізації з командою розробки пропонується використовувати двотижневі дизайн-спринти. Структуру спринту та розподіл активностей у РМ-платформі наведено у Таблиці 2.1.

Ключовою метою впровадження моделі є підвищення операційної ефективності (Eops). Для кількісної оцінки запропонованої структури вводиться формула розрахунку «чистого творчого часу».

У традиційній (хаотичній) моделі значна частина часу дизайнера (T_{total}) витрачається на комунікацію та пошук інформації. В моделі ADSM ми прагнемо максимізувати час безпосереднього проєктування (T_{design}).

$$E_{ops} = \frac{T_{design}}{T_{total}} = \frac{T_{total} - (T_{sync} + T_{search} + T_{admin})}{T_{total}} \cdot 100\% \quad (2.1)$$

де: T_{sync} – час на синхронізацію статусів та мітинги; T_{search} – час на пошук актуальних версій макетів та ТЗ; T_{admin} - час на ручне створення задач та звітів.

Таблиця 2.1

Регламент проведення Дизайн-Спринту в моделі ADSM

День спринту	Етап	Активність дизайнера	Дії в РМ-платформі (Jira/Asana)	Артефакти
День 1	Планування	Декомпозиція задач, оцінка часу	Переміщення карток з Backlog в To Do, призначення Story Points	Sprint Backlog
День 2-5	Виконання	Створення концептів, UI-дизайн	Оновлення статусів, лінування фреймів Figma до тікетів	UI Mockups (Draft)
День 6-7	Рев'ю	Презентація команді, внутрішній критикал	Коментування в картках (Proofing), повернення на доопрацювання	Feedback Log
День 8-9	Полірування	Внесення правок, підготовка UI-кіта	Зміна статусу на "Ready for Handoff", прикріплення специфікацій	Final UI, Assets

День спринту	Етап	Активність дизайнера	Дії в РМ-платформі (Jira/Asana)	Артефакти
День 10	Ретроспектива	Аналіз ефективності, передача в Dev	Закриття спринту, аналіз Velocity chart	Звіт спринту

Впровадження РМ-платформи дозволяє мінімізувати Tsearch та Tadmin за рахунок інтеграції.

Ефективність моделі ADSM забезпечується технічною зв'язкою «РМ-платформа + Графічний редактор». Пропонується наступна схема інтеграції даних (рис. 2.2):

1. Figma/Sketch: Є джерелом істини для візуалу. Використовується плагін (наприклад, "Jira for Figma"), який дозволяє розробнику бачити статус розробки прямо в макеті.
2. РМ-платформа: Є джерелом істини для вимог та строків. У картку задачі автоматично підтягується «живе» прев'ю (Live Embed) макету.
3. Комунікатор (Slack/Teams): Отримує тригерні сповіщення лише про критичні зміни статусів (наприклад, «Макет затверджено», «Потрібні правки»).

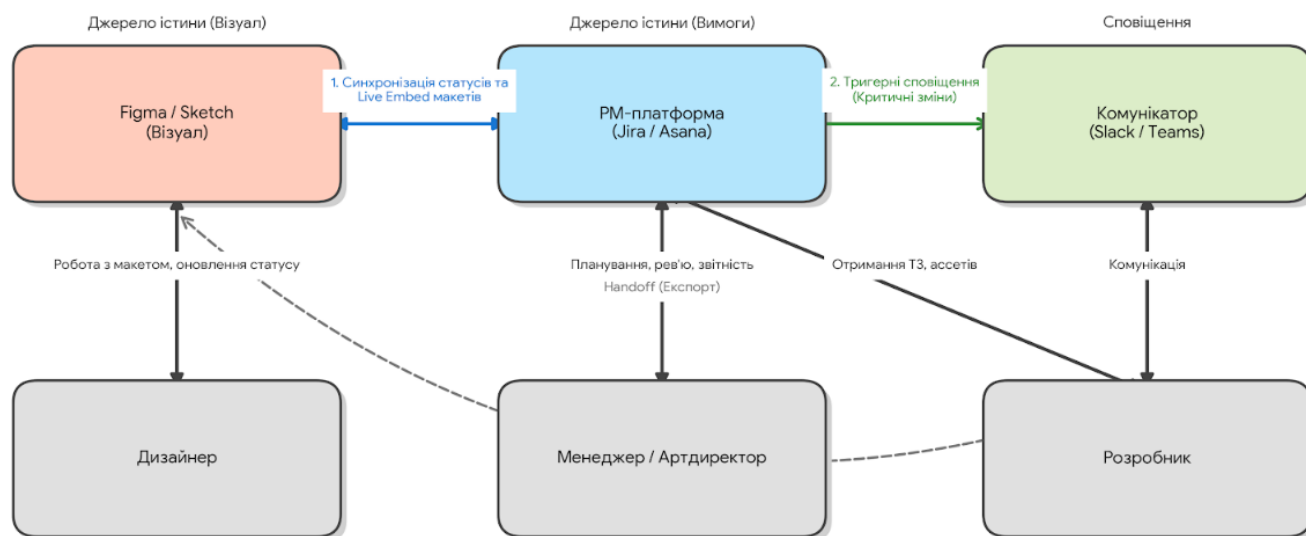


Рис. 2.2.Схема інтеграції даних та інструментів

Схема демонструє потік даних між ключовими інструментами та взаємодію учасників процесу з ними: верхній ряд (інструменти) показує технічну інтеграцію. Figma/Sketch (джерело візуалу) має двосторонню синхронізацію з РМ-платформою (джерело вимог), яка, в свою чергу, надсилає тригерні сповіщення в Комуникатор (Slack/Teams). Нижній ряд (ролі) показує, як кожен учасник команди взаємодіє з цими інструментами. Дизайнер працює переважно у Figma, Менеджер – у Jira/Asana, а Розробник отримує інформацію з усіх трьох джерел, включаючи прямий доступ до макетів для Handoff (пунктирна стрілка). Кольорові стрілки (синя та зелена) виділяють автоматизовані потоки даних, що є основою запропонованої моделі інтеграції.

Така конфігурація дозволяє реалізувати принцип Single Source of Truth (Єдине джерело істини), усуваючи проблему дублювання файлів та роботи з неактуальними версіями дизайну («final_v2_new.png»), що є типовою проблемою графічного дизайну.

Отже, розроблена модель ADSM трансформує хаотичний творчий процес у структурований потік створення цінності. Чітка архітектура статусів та регламентований таймінг спринту створюють умови для прогнозованої роботи, а інтеграція інструментів знижує адміністративне навантаження, вивільняючи ресурс для креативних завдань. Наступним кроком дослідження є експериментальна перевірка цієї моделі на реальному проєкті.

2.2. Оцінка ефективності впровадженої моделі на прикладі веброзробки

Верифікація теоретичних положень та практичної цінності розробленої моделі ADSM (Adaptive Design Sprint Model) здійснювалася шляхом проведення педагогічного експерименту в умовах реального виробничого процесу. Базою для дослідження виступив пілотний проєкт із розробки редизайну корпоративного веб-порталу (e-commerce), що реалізовувався командою у складі одного арт-директора, двох UI/UX-дизайнерів та одного проєктного менеджера. Експеримент тривав 6 тижнів і був розділений на два рівних етапи: констатувальний (3 спринти за

традиційною методологією без спеціалізованої РМ-платформи) та формувальний (з спринти з використанням інтегрованого середовища Jira + Figma + Slack згідно з розробленою моделлю).

2.2.1. Методика оцінювання та порівняльний аналіз показників. Для об'єктивізації результатів було виокремлено систему ключових показників ефективності (KPI), специфічних для дизайн-процесу. Головним індикатором продуктивності обрано Design Velocity (Vd) — кількість успішно закритих Story Points (умовних одиниць складності макетів) за один спринт. Динаміка цього показника дозволяє оцінити вплив структурованого беклогу та чітких статусів на пропускну здатність команди.

Другим критичним показником став середній час отримання зворотного зв'язку (Tfb) – часовий інтервал між зміною статусу задачі на «Internal Review» та отриманням першого змістовного коментаря або затвердження. Скорочення цього часу прямо вказує на ефективність налаштованої системи сповіщень та інтеграції (Proofing).

Результати вимірювань на обох етапах експерименту зведено у Таблицю 2.2.

Таблиця 2.2

Порівняльна характеристика показників ефективності дизайн-процесу

Показник	Базовий етап (Google Sheets + Telegram)	Експериментальний етап (Модель ADSM)	Динаміка змін (Δ)
Середній Design Velocity (V_d), pts/спринт	14	22	+57%
Час на отримання фідбеку (T_{fb}), годин	28.5	4.2	-85%

Показник	Базовий етап (Google Sheets + Telegram)	Експериментальний етап (Модель ADSM)	Динаміка змін (Δ)
Кількість ітерацій правок на один макет ($\$N_{\{rev\}}\$$)	4.5	1.8	-60%
Час на адміністративні задачі РМ ($\$T_{\{admin\}}\$$), хв/день	90	25	-72%
Рівень "втрачених" вимог/коментарів	12%	0%	-100%

Аналіз даних таблиці демонструє суттєве зростання продуктивності. Зокрема, інтеграція РМ-платформи дозволила скоротити час очікування затвердження макетів майже у 7 разів (з 28,5 до 4,2 годин). Це пояснюється автоматизацією процесу: замість ручного написання листів чи повідомлень у месенджери, арт-директор отримував миттєве сповіщення про готовність макету з прямим посиланням на фрейм у Figma.

Графічну інтерпретацію зростання показника Design Velocity по спринтах наведено на рисунку 2.3.

Як видно з графіка (рис. 2.3), на базовому етапі спостерігалася нестабільність («пилкоподібний» графік), спричинена хаотичним надходженням правок та відсутністю пріоритезації. Після впровадження моделі ADSM (спринти 4-6) графік демонструє стабільне зростання, що свідчить про адаптацію команди до нових умов та вихід на плато ефективності.

2.2.2. Розрахунок економічної ефективності. Окрім організаційних переваг, впровадження моделі має виражений економічний ефект, який полягає у мінімізації

невиробничих витрат часу кваліфікованих фахівців. Розрахунок економії бюджету (S_{save}) на адміністративних операціях проводився за формулою:

$$S_{save} = (T_{admin_base} - T_{admin_new}) \cdot C_h \cdot D_{sprint} \quad (2.2)$$

де: T_{admin_base} – щоденні витрати часу на адміністрування у базовому періоді (90 хв $\approx$ 1,5 год); T_{admin_new} – щоденні витрати часу на адміністрування після впровадження платформи (25 хв $\approx$ 0,4 год); C_h – середня погодинна ставка спеціаліста (Project Manager або Art Director). Для розрахунку прийнято середньоринкову ставку в Україні на рівні 600 грн/год; D_{sprint} – тривалість одного спринту в робочих днях (стандартний спринт – 10 днів).

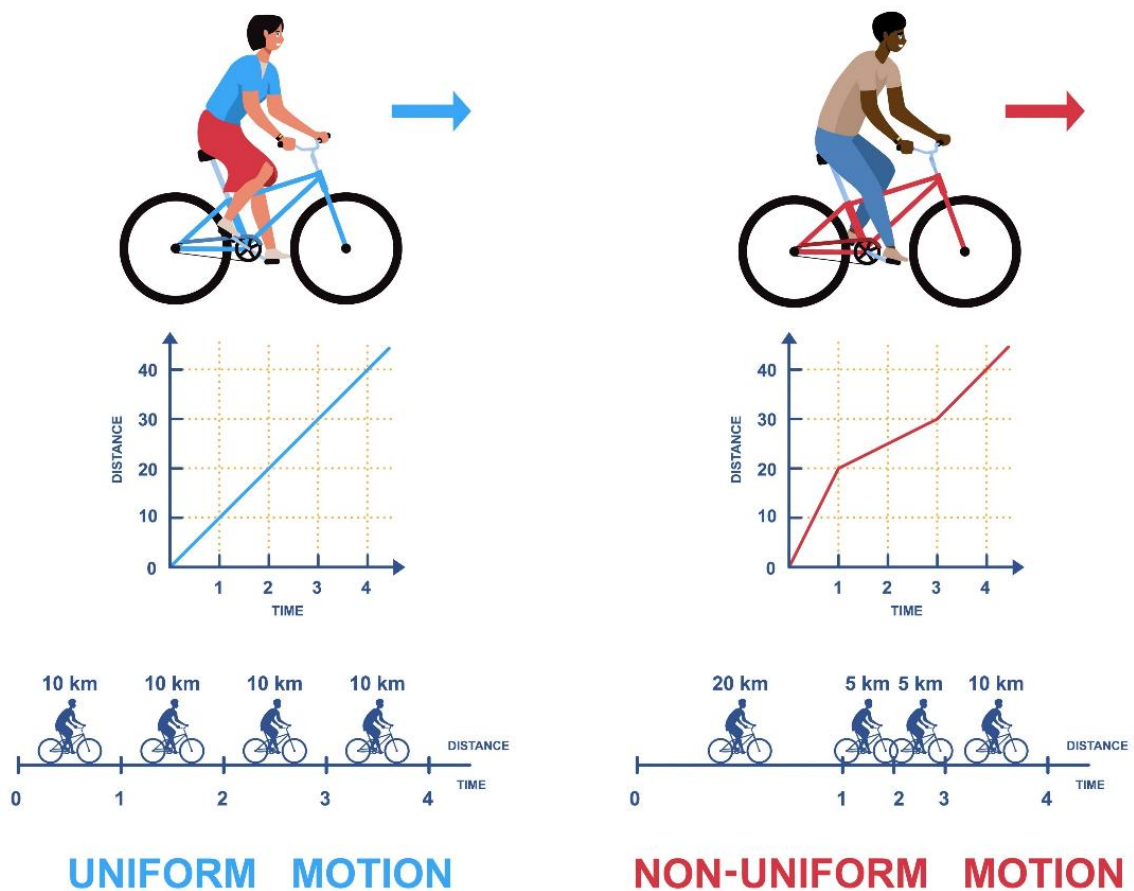


Рис. 2.3. Інтерпретація зростання показника Design Velocity по спринтах

Підставивши отримані експериментальні дані у формулу, отримуємо розрахунок економії за один спринт:

$$S_{save} = (1.5 - 0.4) \cdot 600 \cdot 10 = 1.1 \cdot 6000 = 6\,600 \text{ грн} \quad (2.3)$$

Таким чином, лише за рахунок автоматизації адміністративних процесів економія складає 6 600 грн за один спринт.

Для повного пілотного проєкту, який складався з 3-х спринтів, сумарний економічний ефект (E_{total}) становить:

$$E_{total} = S_{save} \cdot 3 = 6\,600 \cdot 3 = 19\,800 \text{ грн} \quad (2.4)$$

Впровадження моделі дозволило заощадити майже 20 000 грн бюджету проєкту лише на оптимізації роботи одного менеджера. У масштабах дизайн-студії, яка веде паралельно 5-6 подібних проєктів, місячна економія може досягати 100 000 грн, що робить інвестиції в підписку на платні тарифи РМ-платформ (вартість яких зазвичай не перевищує 2000-4000 грн/міс на команду) високорентабельними.

Експериментальне впровадження запропонованої моделі ADSM підтвердило її високу ефективність. Перехід від неспеціалізованих інструментів комунікації до професійної екосистеми управління (РМ-платформа + Design Tools) дозволив підвищити швидкість розробки макетів на 57 % та радикально зменшити комунікаційні затримки. Це доводить, що інвестиції часу в налаштування Design Ops процесів є виправданими та необхідними для сучасних дизайн-команд, що працюють у сфері веброботи.

ВИСНОВКИ ДО РОЗДІЛУ 2

1. Спроектовано авторську модель управління дизайн-спринтом ADSM (Adaptive Design Sprint Model), яка адаптує методологію Agile до специфіки графічного дизайну. Модель структурує творчий хаос шляхом введення чіткого життєвого циклу задачі з обов'язковими точками контролю якості (Quality Gates) на етапах внутрішнього рев'ю та затвердження клієнтом.

2. Розроблено схему безшовної інтеграції програмного середовища (Figma + Jira/Asana + Slack), що реалізує принцип «Єдиного джерела істини» (Single Source of Truth). Це дозволило автоматизувати синхронізацію візуальних макетів з технічними вимогами, усунувши ризики роботи команди з неактуальними версіями дизайну.

3. Результати формувального етапу експерименту зафіксували зростання ключового показника ефективності – Design Velocity — на 57 % (з 14 до 22 story points за спринт). Це підтверджує, що впровадження структурованого беклогу та Kanban-дошки сприяє кращій концентрації дизайнерів на пріоритетних задачах.

4. Впровадження системи автоматичних сповіщень та інструментів візуального коментування (Proofing) призвело до скорочення середнього часу отримання зворотного зв'язку (Tfb) на 85% (з 28,5 до 4,2 годин), а кількість ітерацій правок на один макет зменшилася з 4.5 до 1.8.

5. Розрахунок економічної ефективності довів, що автоматизація адміністративних процесів дозволила заощадити 19 800 грн бюджету пілотного проєкту (за 3 спринти). Це досягнуто завдяки скороченню невиробничих витрат часу менеджера та арт-директора на 72 %, що робить запропоновану модель високорентабельною для впровадження в дизайн-студіях.

ЗАГАЛЬНІ ВИСНОВКИ

1. На основі аналізу сучасних методологій управління встановлено, що пряме застосування інженерних підходів Agile (Scrum/Kanban) до креативних процесів є неефективним через розсинхронізацію ритмів роботи дизайнерів та розробників. Обґрунтовано доцільність переходу до концепції Design Ops та використання гібридної моделі, яка дозволяє інтегрувати творчий цикл у жорсткі рамки спринтів веброзробки.

2. Проведено порівняльний аналіз провідних PM-платформ (Jira, Trello, Asana) за критеріями візуальної ергономіки та технічної інтеграції. Визначено, що для дизайн-команд критичним є не лише функціонал трекінгу задач, а й можливості візуального рецензування (Proofing) та двостороння синхронізація з графічними редакторами (Figma/Sketch), що дозволяє реалізувати принцип «Єдиного джерела істини».

3. Розроблено модель управління дизайн-спринтом ADSM (Adaptive Design Sprint Model), яка структурує робочий процес через систему статусів із вбудованими «воротами якості» (Quality Gates). Модель формалізує етапи від ідеї до передачі макетів у розробку (Handoff), мінімізуючи ризики повернення задач через невідповідність технічним вимогам.

4. Запропоновано та реалізовано схему автоматизованої екосистеми «PM-платформа + Графічний редактор + Месенджер». Налаштування автоматичних тригерів та сповіщень дозволило усунути проблему роботи з неактуальними версіями макетів та забезпечило прозорість процесу для всіх учасників команди без необхідності постійних мітингів.

5. Результати апробації моделі на реальному вебпроекті засвідчили значне підвищення продуктивності команди: показник Design Velocity зріс на 57%, а середній час отримання зворотного зв'язку скоротився на 85% (з 28.5 до 4.2 годин), що підтверджує гіпотезу про позитивний вплив структурованого менеджменту на швидкість креативних процесів.

6. Доведено економічну ефективність впровадження розробленої моделі для студій веброзробки. За рахунок автоматизації рутинних адміністративних операцій витрати часу менеджера скоротилися на 72%, що в грошовому еквіваленті дозволило заощадити близько 20 000 грн бюджету лише в рамках одного пілотного проекту, роблячи інвестиції в РМ-інструментарій високорентабельними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ ISO 21500:2013. Настанови щодо управління проектами (ISO 21500:2012, IDT). – [Чинний від 2014-07-01]. – Київ : Мінекономрозвитку України, 2014. – 54 с.
2. Настанова з питань управління проектами (PMBOK® Guide). – 7-ме вид. – Pennsylvania : Project Management Institute, Inc., 2021. – 370 p.
3. Burgess D. Design Ops Handbook / D. Burgess, J. Sherwin. — New York : DesignBetter.Co by InVision, 2020. — 165 p.
4. Gothelf J. Lean UX: Designing Great Products with Agile Teams / J. Gothelf, J. Seiden. — 3rd ed. — Sebastopol : O'Reilly Media, 2021. — 200 p.
5. Hall E. Just Enough Research / E. Hall. — New York : A Book Apart, 2019. — 154 p.
6. Lewrick M. The Design Thinking Playbook: Mindful Digital Transformation of Teams, Products, Services, Businesses and Ecosystems / M. Lewrick, P. Link, L. Leifer. — Hoboken : Wiley, 2018. — 320 p.
7. Rehkopf M. Jira Strategy Admin Workbook: Templates for the Process Improvement of Your Jira Instance / M. Rehkopf. — Birmingham : Packt Publishing, 2020. — 298 p.
8. Skinner J. Creative Project Management: Templates and guides for managing creative projects from initiation to close / J. Skinner. — London : Kogan Page, 2020. — 240 p.
9. Design, user experience, and usability. user experience design practice / ed. by A. Marcus. Cham: Springer International Publishing, 2014. URL: <https://doi.org/10.1007/978-3-319-07638-6>
10. Design, user experience, and usability. user experience design practice / ed. by A. Marcus. Cham : Springer International Publishing, 2014. URL: <https://doi.org/10.1007/978-3-319-07638-6>
11. Elsayed A. M. Web content strategy in higher education institutions. Information development. 2016. Vol. 33, no. 5. P. 479–494. URL: <https://doi.org/10.1177/0266666916671387>

12. Golander G., Tractinsky N., Kabessa-Cohen I. Trends in website design. AIS transactions on human-computer interaction. 2012. Vol. 4, no. 3. P. 169–189. URL: <https://doi.org/10.17705/1thci.00045>
13. Notess M. Usability, user experience, and learner experience. ELearn. 2001. Vol. 2001, no. 8. P. 3. URL: <https://doi.org/10.1145/566933.566938>
14. Usability and user experience / C. Rusu et al. International journal of information technologies and systems approach. 2015. Vol. 8, no. 2. P. 1–12. URL: <https://doi.org/10.4018/ijitsa.2015070101>
15. Usability and user experience / C. Rusu et al. International journal of information technologies and systems approach. 2015. Vol. 8, no. 2. P. 1–12. URL: <https://doi.org/10.4018/ijitsa.2015070101>
16. Work and life of stanley kubrick. Work and life of stanley kubrick. URL: <https://kubrick.life/>
17. Академія ITSTEP (2023, 14 квітня). Фігма – базовий інструмент для роботи дизайнерів. URL: <https://shorturl.at/Pr1Na>
18. Atlassian Team. The Agile Coach: Scrum, Kanban, and DevOps guides [Електронний ресурс]. Режим доступу: <https://www.atlassian.com/agile>. — Дата звернення: 10.12.2025.
19. Design Operations (DesignOps) 101 [Електронний ресурс] / Nielsen Norman Group. Режим доступу: <https://www.nngroup.com/articles/design-operations-101>. — Дата звернення: 10.12.2025.
20. Figma for Agile Teams: How to integrate design into your workflow [Електронний ресурс]. Режим доступу: <https://www.figma.com/best-practices/agile-design>. Дата звернення: 10.12.2025.
21. How to use Asana for design project management [Електронний ресурс] / Asana Guide. Режим доступу: <https://asana.com/guide/examples/marketing/design-project-management>. — Дата звернення: 10.12.2025.
22. Jira Software for Designers [Електронний ресурс] / Atlassian Documentation. Режим доступу: <https://www.atlassian.com/software/jira/guides/use-cases/designers>. — Дата звернення: 10.12.2025.
23. Malewicz M. The ultimate guide to Design Handoff [Електронний ресурс] / M. Malewicz. — Режим доступу: <https://uxdesign.cc/the-ultimate-guide-to-design-handoff-9c042796245>. — Дата звернення: 10.12.2025.
24. Scrum Guide 2020 [Електронний ресурс] / K. Schwaber, J. Sutherland. Режим доступу: <https://scrumguides.org/scrum-guide.html>. Дата звернення: 10.12.2025.

25. State of Agile Report [Електронний ресурс]. Режим доступу: <https://stateofagile.com>. — Дата звернення: 10.12.2025.
26. Маркетинговий статистичний аналіз веб-сайту університету митної справи та фінансів / Маркетинг і цифрові технології URL: https://mdt-opu.com.ua/index.php/mdt/article/view/305?utm_source=chatgpt.com
27. Сходінка у веб-дизайні. URL: <https://skhodynka-marathon.com/>
28. Андерсон Д. Канбан. Альтернативний шлях до Agile / Д. Андерсон ; пер. з англ. Київ : Фабула, 2019. — 304 с.
29. Бушуєв С. Д. Креативні технології в управлінні проектами та програмами : монографія / С. Д. Бушуєв, Н. С. Бушуєва, Р. Ф. Ярошенко. — Київ : Саміт-Книга, 2019. — 208 с.
30. Верба В. А. Управління проектами : навч. посіб. / В. А. Верба, О. М. Загородніх. — Київ : КНЕУ, 2020. — 362 с.
31. Кнапп Д. Спринт. Як вирішити великі проблеми та протестувати нові ідеї всього за п'ять днів / Д. Кнапп, Д. Зерацьки, Б. Ковітц ; пер. з англ. — Київ : Наш Формат, 2019. 288 с.
32. Кон М. Scrum. Навчись робити вдвічі більше за менший час / М. Кон. Харків : КСД, 2021. 256 с.
33. Рубін К. С. Основи Scrum. Практичний путівник з гнучкої розробки програмного забезпечення / К. С. Рубін. — Київ : Брайт Стар Паблішинг, 2018. 560 с.
34. Сазерленд Д. Scrum. Навчись робити вдвічі більше за менший час / Джефф Сазерленд ; пер. з англ. Харків : Клуб Сімейного Дозвілля, 2016. 280 с.
35. Тесленко П. О. Управління командою проекту : навч. посіб. / П. О. Тесленко. Одеса : ОДАБА, 2021. 184 с.
36. Чумаченко І. В. Управління ІТ-проектами : підручник / І. В. Чумаченко, В. В. Морозов, Н. В. Доценко. Київ : Університет «Україна», 2018. 340 с.
37. Білощицька С. В. Методи та засоби управління ІТ-проектами з використанням гнучких методологій / С. В. Білощицька, О. С. Кучанський // Вісник Житомирського державного технологічного університету. Серія: Технічні науки. 2019. № 2 (84). С. 132–139.
38. Гринчук А. В. Адаптація методології Scrum для управління дизайн-проектами у сфері ІТ / А. В. Гринчук // Управління розвитком складних систем. — 2021. — № 46. — С. 54–60.

39. Карпенко М. О. Особливості застосування Kanban-дошок у розподілених командах веброзробки / М. О. Карпенко // Комп'ютерні системи та мережі. – 2022. – Т. 4, № 1. – С. 22–29.
40. Ковтун Т. О. Порівняльний аналіз систем управління проєктами: Jira, Trello, Asana / Т. О. Ковтун // Сучасні проблеми управління підприємствами: теорія та практика : матеріали міжнар. наук.-практ. конф. Харків : НТУ «ХПІ», 2020. С. 145–147.
41. Федорова Ю. В. Впровадження підходів Design Operations у процеси розробки програмного забезпечення / Ю. В. Федорова // Інформаційні технології в економіці та управлінні. 2023. № 12. С. 88–94.

ДОДАТКИ