

<https://doi.org/10.31891/2307-5732-2025-353-33>

УДК 004.4

СКРИПНИК ТЕТЯНА

Хмельницький національний університет

<https://orcid.org/0000-0002-8531-5348>e-mail: tskripnik1970@gmail.com**ВОЗНІЮК ЛЕОНІД**

Хмельницький національний університет

<https://orcid.org/0009-0002-1152-3192>e-mail: leonid.voznuk@gmail.com**ЄЛИЗАВЕТА МАНДЗЮК**

Хмельницький національний університет

e-mail: Lizamandzuk@gmail.com

УПРАВЛІННЯ ІТ-ПРОЕКТАМИ: ВИКОРИСТАННЯ ГНУЧКИХ МЕТОДОЛОГІЙ

У статті розглянуто перспективи впровадження гнучких методологій управління проектами в ІТ-сфері та шляхи їх ефективного використання. Проаналізовано ключові аспекти Agile-менеджменту та наведено найбільш популярні інструменти гнучких підходів, зокрема Scrum і Kanban. Окремлено основні принципи Agile, їх вплив на підвищення командної ефективності та оптимізацію робочих процесів. Розглянуто переваги й обмеження Scrum як одного з провідних інструментів у гнучкому управлінні. Проведено огляд сучасних наукових джерел, присвячених організації управлінських процесів у галузі інформаційних технологій із використанням Agile-підходів. Проаналізовано ключові алгоритми та принципи, які застосовуються в ІТ-проектах для організації процесу створення програмного забезпечення. Визначено відмінності між методологіями Kanban та Scrum, розкрито їх зміст, специфіку використання та основні переваги. Акцентовано увагу на важливій ролі інструменту Jira, який сприяє ефективному розподілу завдань у команді та фіксації помилок у процесі розробки. Підкреслено актуальність подальших досліджень у напрямі використання гнучких методів, принципів і практик Agile у проектному менеджменті ІТ-компаній.

Ключові слова: гнучкі методології, ІТ-проект, Agile-менеджмент, Scrum, Kanban, Jira.

SKRYPNYK TETIANA, VOZNYUK LEONID, MANDZIUK YELYZAVETA

Khmelnytskyi National University

IT PROJECT MANAGEMENT: USING FLEXIBLE METHODOLOGIES

The rapid development of information technology and dynamic consumer needs changes force companies to optimize both external business conditions and internal organizational processes. One of the key issues in ensuring competitiveness in the IT services market is effective business management, which can adapt to rapid changes in the IT sector, customer requirements, oversaturation of the market with products and at the same time allows to work quickly, efficiently and cost-effectively. Accelerated development of the IT sector requires modern and advanced methodologies and tools for project management.

Flexible methodologies, such as Agile management and the Lean, Scrum and Kanban methods based on it, help organize work efficiently to produce a quality software product and adapt more quickly to new requirements. It is essential to consider flexible project management methodologies in the IT field, particularly Agile - management and derived methods Lean, Scrum, Kanban. An overview of key aspects of Agile - management will summarize this management methodology's main advantages and disadvantages and form a complete understanding of the current approach to effective management of the company and a particular project.

The article is devoted to identifying promising, agile project management methodologies in the IT sphere and ways of their successful application. It reviews basic aspects of Agile management and reveals the most popular tools of agile methodologies, such as Scrum and Kanban. The principles of Agile are highlighted, and the value of agile methodologies for developing teamwork and optimizing tasks in a team is shown. The advantages and disadvantages of Scrum as one of the most popular Agile tools are highlighted. The review of actual scientific works is carried out, defining questions, connected with organization and conducting of management processes in IT sphere, which are provided due to Agile methodologies. Attention is paid to the defining principles and algorithms that are used in IT projects, to organize work on the production of a software product. The difference between Kanban and Scrum methodologies is defined, the contents of the mentioned tools, key aspects and principles of their application and use are defined. The importance of the software product Jira is emphasized, which plays an important role in the distribution of functional tasks in the team and fixing the identified errors.

The need for further study of the application of Agile methodologies, principles and practices in the project management of IT companies was noted.

Key words: agile methodologies, IT project, Agile - management, Scrum, Kanban, Jira.

Стаття надійшла до редакції / Received 14.04.2025

Прийнята до друку / Accepted 03.05.2025

Постановка проблеми

Стрімкий розвиток інформаційних технологій та постійна зміна потреб споживачів змушують компанії адаптувати як зовнішні умови ведення бізнесу, так і внутрішні організаційні процеси. В умовах жорсткої конкуренції на ринку ІТ-послуг особливої актуальності набуває питання ефективного управління ІТ-підприємствами, здатними оперативно реагувати на зміни в галузі, нові вимоги клієнтів та високий рівень насичення ринку цифровими продуктами. У таких умовах успішність діяльності ІТ-компанії залежить від здатності швидко, якісно та економічно ефективно реалізовувати ІТ-проекти.

Активне зростання ІТ-сектору зумовлює необхідність використання сучасних, перспективних підходів до управління проєктами. Серед них особливе місце займають гнучкі методології – Agile, а також розроблені на його основі підходи, такі як Scrum та Kanban. Ці інструменти дозволяють ефективно організувати командну роботу, забезпечити якісну розробку програмного забезпечення та оперативно адаптуватися до змін.

У цьому контексті доцільним є ґрунтовне вивчення гнучких методологій управління проєктами в ІТ-сфері, зокрема Agile-менеджменту та його похідних. Аналіз основ Agile-підходу дає змогу визначити його ключові переваги та обмеження, а також сформувати цілісне уявлення про сучасні інструменти ефективного управління як проєктами, так і підприємствами в цілому.

Швидка трансформація ІТ-галузі вимагає постійного оновлення підходів до організації управлінських процесів, що передбачає поглиблене вивчення та вдосконалення гнучких методологій.

Метою статті є розкриття основ управління ІТ-проєктами із застосуванням гнучких підходів (Agile) та аналіз ключових інструментів – Scrum і Kanban.

Виклад основного матеріалу

У сучасних умовах глобалізації та високої конкуренції на ринку ІТ-послуг ІТ-компанії змушені постійно вдосконалювати свою діяльність. Щоб забезпечити сталий розвиток та зберегти конкурентні переваги, ІТ-компанії мають оперативно реагувати на динамічні зміни як у зовнішньому, так і у внутрішньому середовищі. У зв'язку з цим виникає потреба у впровадженні новітніх, інноваційних підходів до управління, які дозволяють працювати швидше, продуктивніше й економічно ефективніше.

Науковці наголошують, що ІТ-проєкти мають низку характерних рис, зокрема: орієнтацію на досягнення чітко визначених цілей, координацію взаємозалежних дій, обмеженість у часі, з визначеним початком і завершенням [1, 2]. Висока складність таких проєктів вимагає чіткого розподілу ролей, ефективної координації та паралельного виконання багатьох завдань. Досягати цього можливо завдяки раціональному використанню ресурсів та зусиль упродовж проєктного циклу.

Для досягнення ефективності в управлінні ІТ-проєктами необхідно застосовувати гнучкі методології, які забезпечують гнучке формування вимог і сприяють їх своєчасній реалізації. Такий підхід дозволяє налагодити ефективну взаємодію між членами команди та фахівцями різного профілю. В умовах постійної зміни вимог до програмного продукту саме гнучкі методології стають запорукою успішного виконання проєктних завдань [2].

Основу Agile-підходу в розробці програмного забезпечення становлять принципи переваги комунікації між розробниками над дотриманням суворих процедур, а також пріоритет співпраці із замовником над формальними умовами контракту. Гнучкість та здатність адаптуватися до змін мають більшу цінність, ніж фіксований початковий план, а команда розробників функціонує як самокерована система, яка здатна ефективно реагувати на зміни.

Парадигма гнучкої розробки тісно пов'язана із сучасними концепціями внутрішнього маркетингу в ІТ-компаніях. Її впровадження сприяє гармонійному поєднанню інтересів команди та клієнта. При такому підході співробітники розглядаються як внутрішні клієнти компанії, чия задоволеність і мотивація безпосередньо впливають на якість надання послуг. Іншими словами, якщо компанія прагне задовольнити зовнішнього клієнта, вона має створити сприятливе середовище для власних працівників.

Таким чином, використання гнучких методів розробки програмного забезпечення формує клієнтоорієнтовану корпоративну культуру та створює потужне підґрунтя для підвищення конкурентоспроможності ІТ-компанії.

Одним із провідних напрямів сучасного управління проєктами є *Agile-менеджмент*. Це інноваційний підхід до створення продуктів і послуг, який ґрунтується на проєктному форматі роботи та застосуванні методик гнучкої розробки, таких як *Scrum*, *Kanban* та інші. Основна ідея цих методологій полягає у створенні сприятливого середовища, що забезпечує максимальну ефективність і цінність результатів праці кожного учасника команди.

Фахівці зазначають, що головною особливістю Agile є поділ масштабного процесу реалізації проєкту на невеликі етапи або ітерації, тривалістю, як правило, 2–3 тижні. Такий підхід дозволяє значно знизити ризики та оперативно вносити необхідні корективи [3, 4].

Гнучка методологія також забезпечує прозорість у процесі розробки, даючи клієнтам змогу відстежувати поточний стан проєкту в режимі реального часу. Це сприяє більш точному врахуванню змін у вимогах замовника та підвищує ефективність роботи команди. У результаті компанія отримує продукт вищої якості, уникаючи надмірного бюрократичного поділу ролей, що може стримувати потенціал досвідчених фахівців і обмежувати розвиток молодих працівників.

Для ефективного впровадження Agile-менеджменту важливо дотримуватись ключових принципів, закладених у Маніфесті Agile, підписаному 17 провідними розробниками програмного забезпечення з різних шкіл гнучкої розробки (зокрема Extreme Programming, Scrum, DSDM, Adaptive Software Development, Crystal Clear, Feature-Driven Development, Pragmatic Programming) в 2001 році в штаті Юта, США [5].

На тлі викликів, з якими стикалася ІТ-галузь — таких як швидкі зміни, нестабільність ринку, необхідність точного реагування на вимоги замовника та ефективного розподілу ресурсів — створення Маніфесту стало кроком до подолання кризових явищ у сфері ІТ.

Agile Manifesto базується на дванадцяти принципах гнучкого програмного забезпечення:

1. Наш найвищий пріоритет — задовольнити потреби клієнта шляхом швидкої та безперервної доставки цінного програмного забезпечення;
2. Ласкаво просимо до зміни вимог, навіть на пізній стадії розробки. Гнучкі процеси використовують зміни для конкурентної переваги клієнта;
3. Доставляйте робоче програмне забезпечення часто, від кількох тижнів до кількох місяців, віддаючи перевагу коротшим часовим рамкам;
4. Бізнесмени та розробники повинні працювати разом щодня протягом усього проекту;
5. Створюйте проекти навколо мотивованих людей. Забезпечте їм необхідне середовище та підтримку та довірте їм виконання роботи;
6. Найефективнішим і дієвим методом передачі інформації команді розробників і всередині неї є бесіда віч-на-віч;
7. Працююче програмне забезпечення є основним показником прогресу;
8. Гнучкі процеси сприяють сталому розвитку. Спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп нескінченно довго;
9. Постійна увага до технічної досконалості та гарного дизайну підвищує маневреність;
10. Простота — мистецтво максимізувати кількість невиконаної роботи — є важливою;
11. Найкращі архітектури, вимоги та проекти виходять із самоорганізованих команд;
12. Через регулярні проміжки часу команда розмірковує над тим, як стати більш ефективною, а потім відповідно налаштовує та коригує свою поведінку.

Відповідно до цих принципів:

- люди та взаємодія важливіші за процеси й інструменти;
- робочий продукт важливіший за вичерпну документацію;
- співпраця з клієнтом має більшу цінність, ніж дотримання формальних контрактних умов;
- гнучкість до змін важливіша за сліпе дотримання плану.

Отже, акцент у проектному управлінні слід робити на ефективну комунікацію як усередині команди, так і з клієнтом. Документація не повинна бути самоціллю, а лише інструментом підтримки. Здатність адаптуватися до змін — одна з основних ознак гнучкого підходу.

До основних засад Agile підходу належать такі ключові принципи:

Гнучкість передбачає здатність адаптуватися до змін у середовищі та швидко реагувати на нові вимоги до кінцевого продукту;

Безперервність означає підтримання стабільного робочого ритму, який відповідає очікуванням клієнтів без втрати якості;

Єдність полягає в тісній співпраці розробників і замовників, що дозволяє краще розуміти потреби останніх;

Мотивація передбачає розвиток корпоративної культури, підтримку в команді, чесність і натхнення, що підвищують ефективність роботи;

Простота означає створення зрозумілих і доступних продуктів;

Системність виявляється у злагодженій взаємодії учасників команди як елементів цілісної системи, що працює на досягнення спільної мети;

Автономність забезпечує здатність членів команди до самостійного прийняття рішень, самоорганізації та взаємообміну досвідом;

Відповідальність акцентує увагу на персональній відповідальності за результати діяльності та якість виконаних завдань [6].

Щоб забезпечити максимальну результативність використання Agile-менеджменту, необхідно неухильно дотримуватись цих принципів, які розкривають його сутність як сучасного й ефективного підходу до управління.

Впровадження принципів Agile-розробки має низку позитивних наслідків. Зокрема, гнучкі команди демонструють вищу продуктивність, скорочується кількість непотрібної чи неефективної роботи, а завдяки ітеративному підходу помилки виявляються та усуваються на ранніх етапах проекту [6-9].

Однією з найпоширеніших моделей, побудованих на принципах Agile-менеджменту, є Scrum. Цей метод відзначається високою динамічністю, а його основною метою є досягнення максимальної якості кінцевого продукту. Заснований на ітераційному підході до планування та реалізації проектів, Scrum дозволяє у стислі терміни з мінімальними витратами створювати готові версії продукту за допомогою коротких і уточнюючих ітерацій.

Постійна комунікація з клієнтами в процесі роботи забезпечує поступове коригування функціоналу відповідно до актуальних потреб замовника. Завдяки цьому підходу команда може поетапно планувати реалізацію проекту, своєчасно виявляти потенційні ризики і оперативно на них реагувати.

Важливою перевагою Scrum є здатність гнучко змінювати поточний план дій, швидко виправляти помилки та адаптувати розробку до нових вимог, не втрачаючи ефективності. Такий покроковий підхід забезпечує вищу якість продукту та кращий контроль над усім процесом створення.

Scrum-команда складається лише з трьох ключових ролей: Власника Продукту, Scrum-майстра та Команди розробників. Механізм реалізації Scrum у сфері розробки IT-проектів включає кілька етапів. Він передбачає створення невеликих, мультифункціональних і самоорганізованих команд. Самоорганізація означає, що команда бере на себе відповідальність за повний цикл створення продукту, самостійно визначаючи шляхи досягнення поставлених цілей. Робочий процес при цьому структурується шляхом розбиття проекту на невеликі, чітко окреслені завдання.

Варто зазначити, що у процесі реалізації Scrum значну увагу приділяють пріоритезації задач — завдання сортується відповідно до їхньої важливості. Для кожного елемента додатково визначається обсяг необхідних робіт, що дозволяє ефективно планувати витрати часу на розробку інформаційного проекту.

На цьому етапі чітко визначаються часові рамки, які будуть витрачені на створення проекту. Час поділяється на короткі, однакові за тривалістю відрізки — спринти (ітерації). Для кожного спринту формуються завдання, які команда має виконати протягом цього періоду.

Після завершення кожної ітерації проводиться демонстрація частково готового продукту, який уже потенційно придатний до використання. На основі зворотного зв'язку з клієнтом здійснюється аналіз отриманих результатів, коригується план доопрацювань, а також вдосконалюється сам процес розробки. Важливу роль на цьому етапі відіграє ретроспективний аналіз, що дозволяє виявити недоліки, оптимізувати робочі підходи й визначити нові пріоритети задач на наступний спринт.

Цей процес циклічно повторюється: знову відбувається постановка задач для наступного спринту, їх реалізація, презентація результатів та відповідна адаптація стратегії.

Таким чином, великий проект реалізується невеликою командою шляхом виконання послідовних коротких ітерацій, у межах яких команда працює над окремими частинами продукту. Застосування методології Scrum дозволяє окреслити як переваги, так і потенційні виклики цього підходу (таблиця 1).

Таблиця 1

Переваги та недоліки Scrum

Переваги	Недоліки
Гнучкість та адаптивність до змін вимог і пріоритетів	Потребує високої дисципліни та самоорганізації команди
Регулярний зворотний зв'язок від замовника	Може бути складно застосовувати для великих або багатокомандних проектів
Раннє виявлення помилок завдяки ітеративному підходу	Відсутність чіткого завершеного плану на початку проекту може викликати невизначеність
Підвищення прозорості процесів завдяки щоденним зустрічам та спринтам	Часті зустрічі можуть бути обтяжливими та неефективними без належної підготовки
Мотивація та залученість команди завдяки автономії та самоорганізації	Залежність від досвіду команди – слабка команда може не досягти очікуваних результатів
Чітке бачення пріоритетів завдяки Product Backlog'у	Scrum-майстер може бути перевантажений обов'язками в малих командах
Постійне вдосконалення процесу через ретроспективи	Не завжди підходить для проектів із жорсткими регламентами або фіксованим бюджетом
Швидке отримання працездатного продукту вже на ранніх етапах	Потребує повної залученості замовника в процес розробки

Kanban – це методологія, яка ідеально підходить для невеликих проектів, що не потребують тривалого планування, а також для довготривалих проектів, у яких вимоги формуються поступово в процесі розробки.

Загалом Kanban є системою організації завдань і управління робочими процесами з метою досягнення максимальної ефективності. Вона належить до методів Agile і відзначається високим рівнем прозорості, адаптивності й постійного вдосконалення. Основою Kanban є візуалізація робочого процесу, де всі етапи виконання завдань (workflow) позначаються у вигляді стовпців на дошці, а кожне завдання – у вигляді картки. Ці картки поступово переміщуються по дошці, відображаючи прогрес завдання від планування до завершення.

Ключові принципи Kanban включають:

Фокус на якість – управління якістю через поєднання класичних та гнучких підходів, аналітичні інструменти та спільне проектування бізнес-процесів;

Мінімізація незавершених завдань – концентрація на зменшенні обсягу одночасно виконуваних задач;

Часті релізи – регулярне випускання готових до використання частин продукту підвищує загальну якість;

Баланс між навантаженням і спроможністю команди – тягуча система дозволяє самостійно визначати оптимальний ритм роботи;

Пріоритизація – ефективно розміщення пріоритетів допомагає максимально використовувати можливості команди та компанії.

Застосування методології Kanban також має переваги та недоліки (таблиця 2).

Таблиця 2

Переваги та недоліки Kanban

Переваги	Недоліки
Простота у впровадженні та використанні	Може бути неефективним для складних проєктів з чіткими дедлайнами
Підходить для постійних процесів та довгострокових проєктів	Відсутність чітких ролей може призводити до невизначеності відповідальностей
Візуалізація робочого процесу (легко контролювати статус завдань)	Перевантаження дошки завданнями ускладнює управління
Гнучкість у зміні пріоритетів та завдань	Висока залежність від самодисципліни команди
Мінімізація незавершеної роботи (WIP – work in progress)	Відсутність чітких часових рамок може затягнути виконання завдань
Підходить для команд будь-якого розміру	Погано працює при невизначеності з ресурсами або великими залежностями між задачами
Полегшує виявлення вузьких місць у процесі	Може потребувати додаткового інструментарію для інтеграції з іншими системами

Ця методологія не потребує великих ресурсів і дозволяє краще контролювати хід розробки, ефективно впроваджувати Agile-практики та знижувати витрати. Kanban добре працює у динамічному середовищі, де важлива адаптація, але не завжди підходить для проєктів з фіксованими термінами або складною структурою. Kanban має низку відмінностей від Scrum (таблиця 3).

Таблиця 3

Порівняння методологій Kanban та Scrum

Критерій	Scrum	Kanban
Підхід	Ітеративний (спринти зі сталим терміном)	Потоковий (без чітко фіксованих ітерацій)
Ролі	Визначені: Product Owner, Scrum Master, Команда розробки	Ролі не обов'язкові, команда працює гнучко
Структура	Чітко структурований процес з регламентованими подіями	Гнучка структура, завдання додаються будь-коли
Планування	Планування на початку кожного спринту	Завдання додаються по мірі готовності
Візуалізація процесу	Scrum-дошка зі спринтом і беклогом	Kanban-дошка з етапами виконання завдань
Зміна обсягу задач	Не допускається в межах спринту	Можна змінювати у будь-який момент
Частота релізів	Наприкінці кожного спринту	По завершенню кожного завдання
Пріоритетність завдань	Встановлюється до спринту	Може змінюватися в реальному часі
Контроль над обсягом роботи	Обсяг фіксується на спринт	Обмежується кількість завдань у роботі одночасно (WIP)
Найкраще підходить для	Проєктів з чіткими етапами	Безперервних процесів або проєктів з мінливою специфікацією

Для організації роботи з **Scrum** та **Kanban** дуже часто використовується спеціалізоване програмне забезпечення, зокрема **Jira** — популярний інструмент управління проєктами, розроблений компанією Atlassian. Його функціонал створено з урахуванням потреб **Agile-команд**, тому він ідеально підходить як для роботи за методологією Scrum, так і Kanban [10].

У контексті Scrum, Jira дозволяє:

- планувати спринти, створювати беклог продукту та беклог спринту;

- вести історії користувача (user stories), завдання та підзадачі;
 - фіксувати швидкість виконання завдань (velocity);
 - здійснювати щоденні стендапи, демо, спринт-рев'ю та ретроспективи;
 - генерувати звіти про прогрес, зокрема Burndown Charts.
- У межах Kanban, Jira допомагає:
- створювати канбан-дошки з візуалізацією робочого процесу (workflow);
 - обмежувати кількість завдань у кожному стовпці (WIP limits);
 - контролювати цикл часу (cycle time) та час виконання (lead time);
 - забезпечувати гнучке додавання завдань без потреби чекати нового спринту;
 - підтримувати неперервний потік роботи.

Крім того, Jira має гнучкий інтерфейс, підтримує інтеграції з іншими інструментами (Confluence, Bitbucket, Slack тощо), а також дозволяє налаштовувати ролі, доступи, нагадування, сповіщення, плагіни та автоматизації, що робить її універсальним середовищем для Agile-менеджменту.

Таким чином, використання Jira значно підвищує прозорість, ефективність і керованість проєктів, незалежно від того, яка методологія застосовується — Scrum чи Kanban.

Висновки

Отже, поєднання моделей Scrum та Kanban (Scrum-ban) демонструє високу гнучкість і ефективність у реагуванні на зміни, що виникають під час реалізації IT-проєктів. Використання цих підходів дозволяє впроваджувати принципи Agile-менеджменту, які орієнтовані на підвищення продуктивності командної роботи, забезпечення якості кінцевого продукту, а також на оптимізацію часових та ресурсних витрат компанії.

Завдяки інтеграції таких інструментів, як багтрекінгова система Jira, IT-фахівці отримують можливість ефективно керувати проєктами: відслідковувати помилки, координувати завдання між учасниками команди, контролювати пріоритети, а також покращувати прозорість та прогнозованість процесів розробки.

У перспективі подальші дослідження мають бути зосереджені на аналізі реальних кейсів використання гнучких методологій в IT-середовищі. Це дозволить не лише узагальнити ефективні практики впровадження, але й розробити інструменти для мінімізації потенційних недоліків та бар'єрів їх застосування у різних організаційних контекстах.

Література

1. Brewer, J.L., & Dittman, K.C. *Methods of IT Project Management*. 4th ed. West Lafayette: Purdue University Press, 2022. 582 p.
2. Кузьмич, В. О., Коваль, О. В., & Тараненко, Р. А. *Моделі та засоби управління IT проєктами: Навчальний посібник*. Київ: КПІ ім. Ігоря Сікорського, 2023.
3. Davidson, B. *Project Management Unlocked: An Easy Guide to Mastering Essential Principles*. 1st ed. It Was A Dark And Starry Night, 2024. 174 p.
4. Волинець, Д. *CURLY менеджмент. Як управляти проєктами, не зраджуючи собі*. Київ: Yakaboo Publishing, 2023.
5. Agile Manifesto. *Manifesto for Agile Software Development* [Електронний ресурс]. – Режим доступу: <http://agilemanifesto.org>
6. Brewer, J.L. *Methods of IT Project Management*. 4th ed. West Lafayette: Purdue University Press, 2022. 582 p.
7. Radiuk, P.M., Skrypnyk, T.K., & Karlechuk, D.T. Applying mental models to making controlled critically safe decisions in IT project management. *Herald of Khmelnytskyi National University. Technical Sciences*, 2021, Vol. 301, no. 5, pp. 32–35. DOI: 10.31891/2307-5732-2021-301-5-32-35.
8. Скрипник, Т., Багрий, Р., Мазурець, О., Вознюк, Л., & Ільчишин, В. Автоматизований розподіл процесів при управлінні іт-проєктами. *Вісник Хмельницького національного університету. Технічні науки*, 2025, №349(2), с. 398–406. DOI: [10.31891/2307-5732-2025-349-58](https://doi.org/10.31891/2307-5732-2025-349-58)
9. Яковчук, М.В., Міхалевський, В.Ц., Скрипник, Т.К. Порівняння ефективності управління в децентралізованих та централізованих системах. *Вісник Хмельницького національного університету. Технічні науки*, 2021, №5. DOI: [10.31891/2307-5732-2021-301-5-41-4](https://doi.org/10.31891/2307-5732-2021-301-5-41-4)
10. Інструменти для роботи в команді: від стартапу до великої корпорації [Електронний ресурс]. – Режим доступу: <https://www.atlassian.com/software/jira>

References

1. Brewer, J.L., & Dittman, K.C. *Methods of IT Project Management*. 4th ed. West Lafayette: Purdue University Press, 2022. 582 p.
2. Kuzminykh, V.O., Koval, O.V., & Taranenko, R.A. *Models and Tools for IT Project Management: A Textbook*. Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2023.
3. Davidson, B. *Project Management Unlocked: An Easy Guide to Mastering Essential Principles*. 1st ed. It Was A Dark And Starry Night, 2024. 174 p.

4. Volynets, D. *CURLY Management: How to Manage Projects Without Betraying Yourself*. Kyiv: Yakaboo Publishing, 2023.
5. Agile Manifesto. Manifesto for Agile Software Development [Electronic resource]. – Available at: <http://agilemanifesto.org>
6. Brewer, J.L. *Methods of IT Project Management*. 4th ed. West Lafayette: Purdue University Press, 2022. 582 p.
7. Radiuk, P.M., Skrypnyk, T.K., & Karlechuk, D.T. Applying Mental Models to Making Controlled Critically Safe Decisions in IT Project Management. *Herald of Khmelnytskyi National University. Technical Sciences*, 2021, Vol. 301, no. 5, pp. 32–35. DOI: 10.31891/2307-5732-2021-301-5-32-35
8. Skrypnyk, T., Bahrii, R., Mazurets, O., Vozniuk, L., & Ilchyshyn, V. Automated Process Distribution in IT Project Management. *Herald of Khmelnytskyi National University. Technical Sciences*, 2025, No. 349(2), pp. 398–406. DOI: 10.31891/2307-5732-2025-349-58
9. Yakovchuk, M.V., Mikhaliivskyi, V.Ts., Skrypnyk, T.K. Comparison of Management Efficiency in Decentralized and Centralized Systems. *Herald of Khmelnytskyi National University. Technical Sciences*, 2021, No. 5. DOI: 10.31891/2307-5732-2021-301-5-41-4
10. Team Collaboration Tools: From Startup to Enterprise [Electronic resource]. – Available at: <https://www.atlassian.com/software/jira>