

МАРІЯ КАМЕНСЬКА

Київський національний університет технологій та дизайну

<https://orcid.org/0009-0008-2771-2418>e-mail: mariakamenskayaa@gmail.com**ЛАВРИК ВОЛОДИМИР**

Київський національний університет технологій та дизайну

<https://orcid.org/0000-0002-6448-2470>e-mail: lavryk.vv@knuutd.edu.ua

КРИТЕРІЇ ВИБОРУ ПІДХОДІВ ДО УПРАВЛІННЯ ПРОЄКТАМИ У РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Предметом даної статті є порівняльний аналіз каскадної моделі (Waterfall) та гнучких методологій (Agile) у управлінні IT-проєктами. У статті розглядаються ключові принципи, переваги та недоліки кожної методології, їх застосування в реальних умовах, а також практичні рекомендації щодо вибору оптимального підходу залежно від специфіки проєкту. Дослідження базується на аналізі наукових джерел та кейсів, зокрема трансформації компанії Avira SOFT S.R.L, що перейшла від Waterfall до Agile.

Ключові слова: управління проєктами, Waterfall, Agile, Scrum, Kanban, IT-проєкти.

MARIA KAMENSKA**LAVRIK VOLODYMYR**

Kyiv National University of Technologies and Design

SELECTION CRITERIA FOR PROJECT MANAGEMENT APPROACHES IN SOFTWARE DEVELOPMENT

In today's dynamic technological landscape, effective project management is a critical success factor in software engineering. This paper presents a comparative analysis of three widely used project management methodologies: Waterfall, Scrum, and Kanban. Based on key evaluation criteria—such as flexibility, documentation, risk management, change management, teamwork, planning, and testing—each methodology's strengths and limitations are thoroughly examined. A structured comparison table highlights how these approaches differ and under what circumstances each proves most effective.

The research shows that the Waterfall model is a linear and sequential approach ideal for projects with clearly defined, stable requirements, where thorough documentation and predictability are essential. However, its rigidity makes it less adaptable to change. Scrum, an iterative framework, is highly responsive to changes and fosters close team collaboration, but it demands strong discipline and may not be suitable for projects with tight, inflexible deadlines. Kanban focuses on continuous workflow and visual task management, providing high flexibility and responsiveness but making time prediction and long-term planning more challenging.

The paper provides practical recommendations for selecting the appropriate methodology depending on project type, uncertainty level, and team priorities. It concludes that no single approach is universally optimal. Instead, choosing the right methodology requires a deep understanding of the project's nature and the ability to balance structure and adaptability. Applying the most suitable methodology can significantly enhance software development efficiency, reduce risks, and ensure successful project delivery in today's competitive environment.

Keywords: project management, Waterfall, Agile, Scrum, Kanban, IT projects.

Стаття надійшла до редакції / Received 21.06.2025

Прийнята до друку / Accepted 15.07.2025

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Сучасне бізнес-середовище та IT-сфера характеризуються високим рівнем динаміки, конкуренції та технологічних змін. В умовах швидкої трансформації зовнішнього середовища ефективне управління проєктами стає ключовим чинником досягнення стратегічних цілей компаній. Проєкти, як тимчасові ініціативи, потребують чіткого планування, скоординованих дій та оптимального використання ресурсів (див. рисунок 1). Це зумовлює необхідність вибору відповідної методології управління проєктами залежно від характеру завдань, рівня невизначеності та очікуваних результатів.

На практиці широко використовуються дві протилежні за підходом методології — каскадна модель (Waterfall) та гнучкі підходи (Agile). Кожна з них має свої переваги та обмеження, і їх ефективне застосування залежить від контексту проєкту. Однак вибір невідповідної методології може призвести до затримок, перевитрати ресурсів і навіть провалу проєкту. Це актуалізує проблему методологічної адаптивності в управлінні IT-проєктами.

Таким чином, постає наукове та практичне завдання — проаналізувати особливості застосування Waterfall і Agile, визначити критерії вибору між ними та сформулювати практичні рекомендації для фахівців, які працюють у сфері управління IT-проєктами різного масштабу та складності.

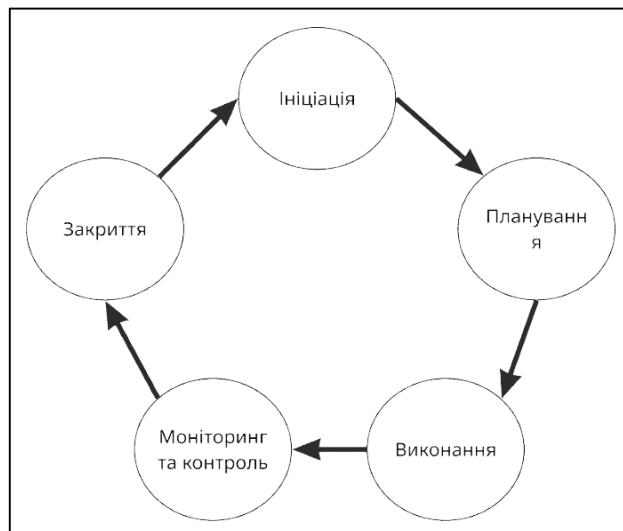


Рис.1. Фази життєвого циклу проєкту (створено автором)

Аналіз досліджень та публікацій

У роботі [10] наведено результати досліджень, присвячених аналізу підходу "Code and Fix" як одного з найперших методів розробки програмного забезпечення. Показано, що цей метод дозволяє швидко отримати перший результат за рахунок відсутності планування та документації. Але залишилися невирішеними питання, пов'язані з масштабованістю та підтримкою таких систем у довгостроковій перспективі. Причиною цього можуть бути як об'єктивні труднощі, пов'язані з браком контролю за якістю, так і принципова неможливість керувати ризиками без чіткого процесу. Варіантом подолання відповідних труднощів може бути застосування більш структурованих моделей розробки.

Саме такий підхід використано у роботах [5, 11], де представлено каскадну модель (Waterfall), що базується на послідовному проходженні фаз розробки. Показано, що така модель забезпечує високу передбачуваність і контроль, особливо для великих проєктів із чітко визначеними вимогами. Проте у цих роботах також залишилися відкритими питання щодо здатності Waterfall адаптуватися до змін вимог у процесі розробки. Причиною цього є жорстка лінійна структура, яка не передбачає зворотних ітерацій. Як наслідок, виникає потреба у більш гнучких підходах до розробки ПЗ.

У статтях [1, 6] наведено результати досліджень у сфері гнучких методів розробки програмного забезпечення, що стали основою для створення Agile-методологій. Показано, що використання принципів Agile забезпечує адаптивність, орієнтацію на замовника та зменшення витрат на зміни. Проте залишилися невирішеними питання щодо ефективного впровадження Agile в умовах великих організацій із усталеними процесами. Причиною цього можуть бути як організаційна інертність, так і витратна частина в плані навчання персоналу та зміни внутрішніх процедур. Варіантом подолання цих труднощів може бути поступове впровадження гнучких елементів через гібридні моделі.

У роботах [15, 16] досліджено застосування фреймворку Scrum, як однієї з реалізацій принципів Agile. Показано, що Scrum дозволяє ефективно управляти динамічними вимогами завдяки ітеративному підходу. Проте, як зазначено у роботах, виникають складнощі із застосуванням Scrum у проєктах з нечітко визначеними ролями або в командах з низькою автономією. Все це зумовлює необхідність адаптації фреймворку до конкретного середовища.

У дослідженнях [2, 8] розглянуто Kanban як альтернативу Scrum. Показано, що візуальне управління потоком роботи дозволяє краще контролювати навантаження та швидше виявляти вузькі місця в процесі. Але залишилися невирішеними питання масштабованості цього підходу у великих мультидисциплінарних командах. Причиною цього може бути відсутність чіткої структури або надто спрощена модель планування. Варіантом подолання відповідних труднощів є комбінування Kanban із елементами інших фреймворків, що й демонструють деякі практичні кейси.

Все це дозволяє стверджувати, що доцільним є проведення дослідження, присвяченого порівняльному аналізу традиційних та гнучких методологій розробки ПЗ з урахуванням практичного контексту впровадження, організаційних обмежень та рівня зрілості команд. Такий підхід дозволить не лише краще зрозуміти особливості кожної методики, а й обґрунтувати вибір оптимальної моделі для конкретного проєкту.

Формулювання цілей статті

Метою роботи є дослідження та порівняльний аналіз традиційних (Waterfall) і гнучких (Agile, Scrum, Kanban) методологій управління проєктами у сфері розробки програмного забезпечення з урахуванням їх переваг, обмежень та умов ефективного застосування залежно від специфіки проєкту.

Завдання:

1. Описати історичний розвиток та особливості традиційних і гнучких підходів до управління проєктами.

2. Проаналізувати принципи та інструменти методологій Waterfall, Agile, Scrum і Kanban.
3. Порівняти переваги й обмеження кожного підходу залежно від типу проекту та рівня змінності вимог.
4. Сформулювати рекомендації щодо вибору оптимальної методології або їх поєднання для підвищення ефективності проектної діяльності.

Основний матеріал

Одним із найперших детально описаних підходів до управління розробкою програмного забезпечення є каскадна (**Waterfall**) модель, запропонована Вінстоном В. Ройсом у 1970 році в роботі «Managing the Development of Large Software Systems» [13]. Розробка поділяється на строгі фази, які йдуть одна за одною, кожен етап починається лише після завершення попереднього, через таку каскадність він є доволі не гнучкий, його було створено для стабільних вимог. Модель пропонує чітко визначений набір критеріїв та вимог ще до початку фази проектування та реалізації проекту, таким чином команда розробників заощаджує час у процесі розробки.

На рисунку 2 ми бачимо, що проект А не має методології та наповнений процесами, а також проблемами, які фактично зростають у міру просування проекту. На відміну від цього, проект Б, який має структуровану методологію, мінімізує кількість проблем, які можуть виникнути в проекті.

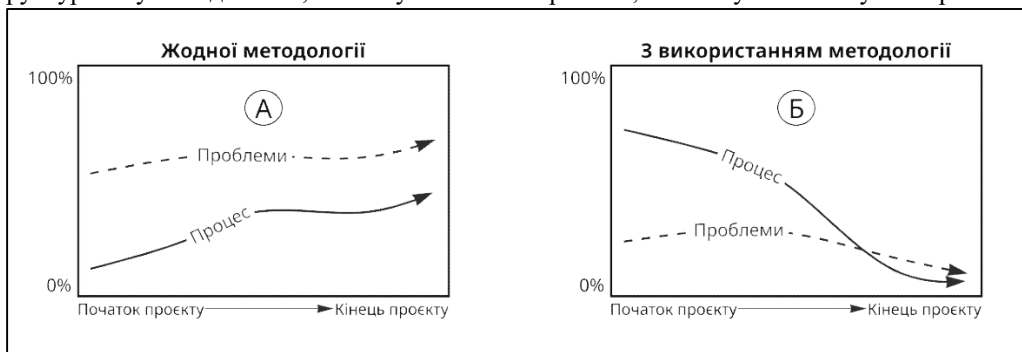


Рис.2. Різниця у використанні методологій (створено автором)

Фази Waterfall моделі включають в себе:

1. Аналіз вимог – визначення функціональних та нефункціональних вимог.
2. Проектування – розробка архітектури ПЗ: алгоритми, схеми даних, логічні структури.
3. Реалізація – написання коду, створення БД та розгортання системи.
4. Тестування – перевірка відповідності вимогам (верифікація) та коректності роботи (валідація). Виправлення помилок.
5. Супровід – виправлення багів, адаптація до змін та підвищення ефективності після впровадження [3].

Під час роботи над невеликими проектами з фіксованими термінами, бюджетом і масштабом Waterfall може допомогти команді краще організувати, оскільки вся документація пишеться вичерпно: описується кожен модуль та графік робіт. Втрата досвіду та знань, якщо учасник залишить проект, буде зведена до мінімуму, а час можна розподілити оптимально, оскільки кожна фаза має фіксований проміжок часу, проте в разі, якщо одне завдання запізнюється, усі завдання будуть запізнюватися [4]. Через таку детальну пророботку проект займає багато часу і грошей, проте це дає високе розуміння того, що відбувається.

Однак, як доводить Ройс на основі аналізу реальних проектів, класична Waterfall-модель непрацездатна для великомасштабних систем через критичні недоліки [13]:

1. Тестування в кінці призводить до виявлення помилок проектування надто пізно.
2. Зміни вимог руйнують весь процес.
3. Якщо щось пішло не так, виправлення коштують дуже дорого.

У підсумку, методологія Waterfall найкраще працює, коли є чітке уявлення про кінцевий продукт і чітко визначені вимоги, які не змінюватимуться часто. Коли час не проблема, а кінцевий продукт головне. Переваги та недоліки методології наведено у таблиці 1.

У свою чергу, **Scrum** розглядає процес розробки як емпіричний (передбачає неповне визначення процесів), на відміну від "теоретичних" методологій, таких як Waterfall, які намагаються детально планувати всі етапи. Більш ретельно методологію Scrum можна подивитися на рисунку 3.

Ключові принципи Scrum:

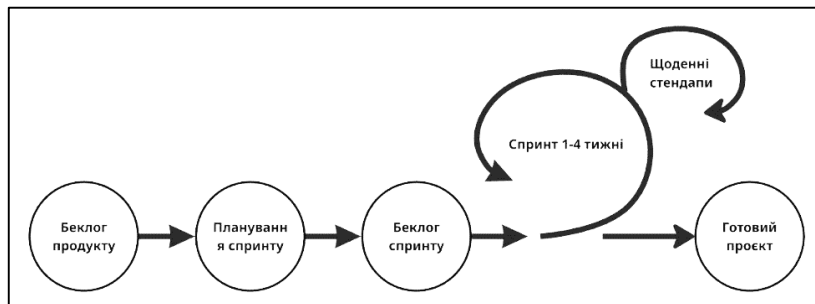
1. Гнучкість: зміни в проекті можливі на будь-якому етапі.
2. Ітеративність: робота поділена на спринти (1–4 тижні), під час яких команда адаптується до нових вимог.
3. Контроль ризиків: регулярні ретроспективи та корективи замість жорсткого дотримання початкового плану.
4. Командна робота: малі групи (3–6 осіб) з кроспрофільною взаємодією.

Таблиця 1

Переваги та недоліки методології Waterfall (створено автором)

	Переваги	Недоліки	Де використовувати
Waterfall	Надає структуру для організації та контролю проекту, має чіткі процедури, які слід дотримуватися, дозволяє розподіл обов'язків та управлінський контроль.	Важко передбачити всі специфікації, оскільки вимоги до проекту можуть змінюватися, погано підходить для великих проектів.	Коли є чітке уявлення про кінцевий продукт, вимоги чітко визначені та не зміняться, коли час не є критичним фактором.

Основними характеристиками Scrum є гнучкість, ітеративність, активна командна взаємодія та механізми управління ризиками через регулярні ретроспективи [7]. Розробка ведеться через серії коротких циклів — спринтів, кожен з яких завершується створенням інкременту функціонального продукту.

**Рис.3. Фреймворк методології Scrum (створено автором)**

Scrum дозволяє оперативно реагувати на зміну вимог і сприяє високій залученості як команди, так і замовників. Водночас метод вимагає високої самодисципліни учасників та певної зрілості команди. Незважаючи на гнучкість, Scrum не завжди є ефективним у проєктах із фіксованими вимогами та жорсткими термінами виконання.

Scrum виник як відповідь на обмеження бюрократизованих систем, пропонуючи гнучкість, швидкість та орієнтацію на реальні потреби клієнтів, проте незважаючи на переваги, які можуть надати гнучкі підходи, багато компаній не поспішають повністю відмовлятися від своїх традиційних практик. Переваги та недоліки методології наведено у таблиці 2.

Таблиця 2

Переваги та недоліки Scrum (створено автором)

	Переваги	Недоліки	Де використовувати
Scrum	Гнучкість і адаптивність до змін вимог на будь-якому етапі проєкту, прозорість процесу завдяки щоденним зустрічам і демо-оглядам, висока залученість команди та замовника	Вимагає високого рівня самодисципліни та зрілості команди, не підходить для проєктів із жорсткими фіксованими вимогами	Коли вимоги змінюються в процесі, потрібна швидка адаптація до ринку, проєкт має інноваційний або дослідницький характер, важлива постійна взаємодія зі стейкхолдерами.

Kanban у розробці програмного забезпечення спонукає команди візуалізувати робочий процес, обмежувати кількість завдань у роботі на кожному етапі та вимірювати час проходження завдань. Метод реалізується як візуальна дошка з колонками (див. рис. 4), що відображають етапи розробки. Завдання подаються у вигляді карток і переміщуються між колонками згідно з прогресом. Для кожного етапу встановлюється ліміт на кількість завдань у роботі, що дозволяє контролювати навантаження та виявляти вузькі місця.

Методологія Kanban найкраще підходить для проєктів з постійно змінюваними пріоритетами та невизначеними обсягами роботи, де важлива гнучкість і оперативне реагування на зміни. Вона ефективна, коли потрібно оптимізувати потік задач, а не фіксувати строки завершення або жорсткі етапи. Kanban особливо корисний у командах підтримки, обслуговування або неперервної доставки продукту. Переваги та недоліки методології наведено у таблиці 3.

З метою кращого розуміння особливостей методологій Waterfall та Agile доцільно розглянути їхнє практичне застосування на прикладі стратегічної трансформації компанії Avira SOFT S.R.L, що відбулася у 2023 році. Аналіз базується на матеріалах наукової статті «Product Development Models in the IT Sector – From Waterfall to Agile Project Management Models in the Case of AVIRA SOFT S.R.L» [9], яка висвітлює перехід компанії від традиційної каскадної моделі до гнучкого підходу в управлінні проєктами у відповідь на зростаючі вимоги клієнтів та динаміку IT-ринку. Зміна підходу до розробки

програмного забезпечення була зумовлена необхідністю реагування на підвищення вимог з боку замовників і посилення конкуренції в IT-секторі.

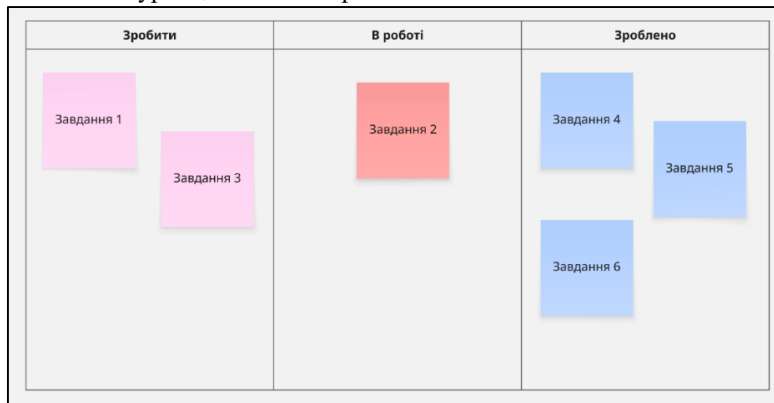


Рис.4. Дошка Kanban (створено автором)

Таблиця 3

Переваги та недоліки методології Kanban (створено автором)

	Переваги	Недоліки	Де використовувати
Kanban	Візуалізація робочого процесу в реальному часі, обмеження WIP запобігає перевантаженню, немає фіксованих спринтів, задачі виконуються безперервно, підходить для підтримки та обслуговування	Відсутність чіткої структури для планування, може не забезпечувати достатньої регулярності зворотного зв'язку, важче прогнозувати терміни завершення	Для проєктів із постійним потоком завдань, техпідтримки, DevOps, багфіксингу, обслуговування ПЗ, або в умовах з частими пріоритетними змінами

У рамках дослідження було проведено інтерв'ю та опитування, які охоплювали такі аспекти, як загальний процес розробки антиспам-програмного забезпечення, застосування моделі Waterfall у цьому процесі, а також перехід до моделі Agile, що був ініційований на початку 2011 року відповідно до стратегічного рішення керівництва компанії.

Розробка антивірусного продукту в компанії AVIRA SOFT S.R.L включала такі основні етапи:

1. Первинний аналіз отриманого електронного листа – оцінка загальних характеристик, таких як форматування (наприклад, виділений червоним заголовок), розмір шрифту, обсяг листа або наявність вкладень.

2. Аналіз вмісту – розбиття листа на логічні частини, застосування правил до речень чи слів, перевірка посилань за допомогою Real-time Blackhole List (RBL) – бази даних URL, які часто зустрічаються у спамі. Кожному посиланню присвоювався бал, що вказував на його ймовірну приналежність до спаму.

3. Застосування фільтра Байєса – використання двох баз даних: однієї зі спам-словами та іншої зі словами зі звичайних листів. Фільтр постійно вдосконалювався, додаючи нові слова з аналізованих листів.

4. Фінальна класифікація – на основі сумарного балу лист визначався як спам або легальний.

До початку 2023 року компанія використовувала Waterfall-модель (див. рис. 5 (а)). Головним недоліком Waterfall-моделі була її лінійність: виявлені помилки на пізніх етапах вимагали повернення до попередніх фаз, що сповільнювало процес.

У 2023 році компанія перейшла на Agile-модель, а саме Scrum, яка запровадила циклічний підхід (див. рис. 5 (б)).

Agile-підхід забезпечив більш тісну взаємодію з клієнтами завдяки частим релізам. Щоденні stand-up зустрічі та щотижневі наради менеджерів (Scrum of Scrums) покращили комунікацію. Однак це також призвело до підвищеного тиску на працівників через короткі терміни та високі очікування, що спричинило певну плінність кадрів.

У підсумку, Agile має переваги у швидкому виявленні помилок, орієнтації на клієнта та конкурентоспроможності. Проте він також створив виклики, такі як збільшене навантаження та необхідність швидкої адаптації, що вплинуло на корпоративну культуру.

Порівняльний аналіз методологій. Методології Waterfall, Scrum та Kanban є одними з найуживаніших підходів у галузі, кожна з яких характеризується специфічними принципами, структурою та механізмами реалізації.

Порівняльний аналіз методологій наведений у таблиці 4. Він дозволяє визначити ключові відмінності між зазначеними методологіями, їх переваги та обмеження, а також обґрунтувати доцільність використання тієї чи іншої моделі в конкретному проєктному середовищі.

Практичні рекомендації до обрання методології. Як показано в таблиці 4, кожна з розглянутих методологій – Waterfall, Scrum та Kanban – має свої унікальні переваги та обмеження, що

робить їх ефективними для різних умов та типів проєктів. При виборі методології управління проєктами слід враховувати не лише технічні аспекти, а й організаційну культуру, готовність команди до змін, обсяг проєкту, а також ступінь взаємодії із замовником.

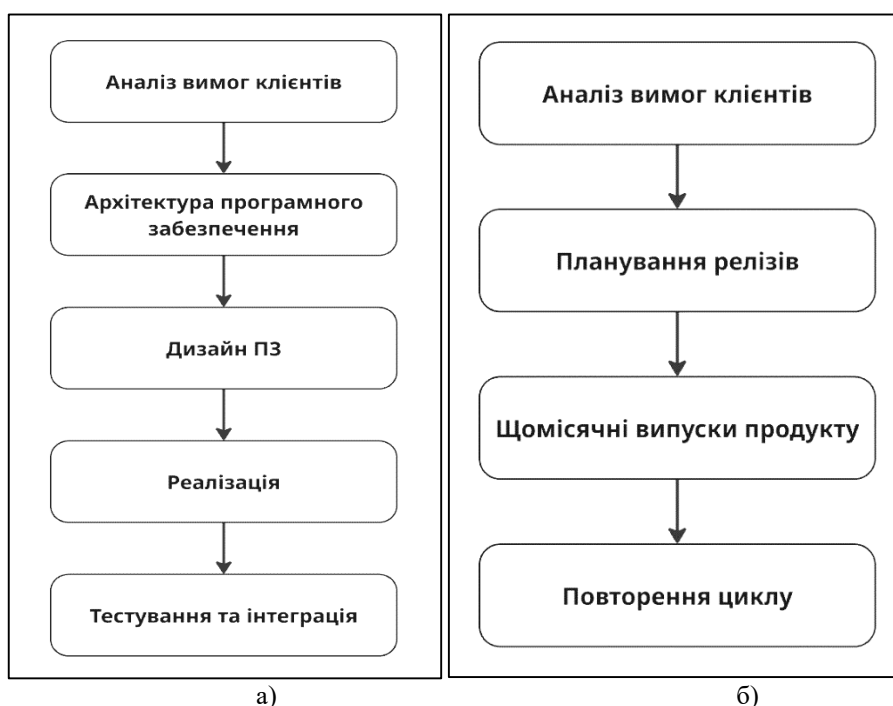


Рис.5. Моделі компанії AVIRA SOFT S.R.L: а) Waterfall - модель ; б) Agile-модель

Таблиця 4

Порівняння методологій Waterfall, Scrum та Kanban (створена автором)

Критерій	Waterfall	Scrum	Kanban
Основні принципи	Лінійний, послідовний підхід з чіткими фазами.	Ітеративний підхід з фіксованими спринтами (1-4 тижні).	Безперервний потік робіт з візуалізацією задач на дошці.
Гнучкість	Низька. Зміни вимог важко впровадити після початку проєкту.	Висока. Зміни можливі між спринтами.	Дуже висока. Зміни можна вносити в будь-який момент.
Документація	Детальна документація на кожному етапі.	Мінімальна документація, акцент на робочий продукт.	Мінімальна, акцент на візуалізацію процесу.
Управління ризиками	Ризики виявляються пізно (на етапі тестування).	Ризики виявляються швидко завдяки регулярним ретроспективам.	Ризики виявляються в реальному часі через візуалізацію потоків.
Підхід до змін	Зміни небажані та дорогі.	Зміни приймаються між спринтами.	Зміни можливі в будь-який час.
Командна робота	Чіткі ролі та відокремлені етапи.	Кроспрофільні команди з щоденними стендапами.	Гнучка командна робота без жорстких ролей.
Підхід до планування	Жорстке планування на початку проєкту.	Планування на рівні спринтів.	Планування гнучке, орієнтоване на пріоритети.
Підхід до тестування	Тестування на завершальному етапі.	Тестування під час кожного спринту.	Безперервне тестування в міру надходження задач.
Найкраще підходить для:	Проєкти з чіткими, незмінними вимогами (державні або банківські системи).	Інноваційні проєкти зі змінними вимогами (стартапи, вебдодатки).	Проєкти з постійним потоком задач.
Недоліки	Важко адаптуватися до змін, великі витрати на виправлення помилок.	Вимагає високої дисципліни команди, не підходить для жорстких дедлайнів.	Важко прогнозувати терміни, відсутність чіткої структури для великих проєктів.

Waterfall – ідеальний для проєктів із чітко визначеними вимогами, стабільним бюджетом та фіксованими термінами. Такий підхід доцільний для державних закупівель, банківських систем, інженерних і будівельних проєктів, де детальне планування та документація є критично важливими. Проте слід уникати Waterfall в динамічному середовищі, де очікується постійна зміна вимог.

Scrum – найкращий вибір для інноваційних, дослідницьких або IT-проєктів, де важлива швидка адаптація до змін, гнучкість у плануванні та активна участь замовника. Scrum вимагає високого рівня самодисципліни та самоорганізації команди, що має бути готова до постійної комунікації, аналізу та вдосконалення процесів. Його доцільно застосовувати у стартапах, розробці мобільних і веб-додатків, R&D-напрямах.

Kanban – підходить для команд, що працюють у режимі підтримки, обслуговування або виконують поточні задачі без чітко визначених термінів завершення. Його доцільно застосовувати в технічній підтримці, DevOps, системному адмініструванні, контент-менеджменті, де важливі оперативність і гнучкість. Kanban дозволяє плавно інтегруватися в уже існуючі процеси, не змінюючи їх кардинально.

Гібридні підходи – в деяких випадках доцільним є поєднання елементів кількох методологій. Наприклад, Water-Scrum-Fall (Waterfall для планування та фінансування, Scrum – для розробки) або Scrum+Kanban (Scrumban), коли команда працює в спринтах, але використовує Kanban-дошку для візуалізації потоку задач.

При виборі методології важливо враховувати:

1. Рівень невизначеності (чи є зрозумілими вимоги до продукту?)
2. Тривалість і розмір проєкту (короткостроковий, довгостроковий, великомасштабний)
3. Залучення замовника (чи готовий клієнт до регулярної взаємодії?)
4. Зрілість команди (чи має команда досвід самоорганізації?)
5. Необхідність у швидкому зворотному зв'язку та гнучкості
6. Збалансоване врахування цих чинників дозволяє ефективно підібрати методологію, що найкраще відповідає реальним потребам проєкту.

Висновок

Вибір методології управління проєктами – стратегічне рішення, яке безпосередньо впливає на успішність реалізації проєкту, задоволеність замовника та ефективність роботи команди. Жодна методологія не є універсальною – кожна має свої переваги та обмеження, тому ключовим фактором є здатність адаптувати підхід до специфіки конкретного проєкту.

Waterfall забезпечує чіткість, передбачуваність і високу якість документації, проте є недостатньо гнучким в умовах змін. Scrum забезпечує високу адаптивність і залученість команди, але вимагає високої зрілості та самоорганізації. Kanban дозволяє оптимізувати потоки робіт і швидко реагувати на зміни, хоча й не підходить для проєктів із чіткими дедлайнами та складною структурою.

У сучасному конкурентному середовищі ефективність проєктної діяльності залежить від здатності гнучко підходити до управління, комбінуючи найкращі практики залежно від потреб. Критично важливо не лише вибрати методологію, а й постійно переглядати її ефективність у процесі реалізації проєкту, коригуючи підхід за потреби. Саме така гнучкість і здатність до адаптації є запорукою стабільного розвитку й успіху в динамічному ринку.

Література

1. Agile-маніфест розробки програмного забезпечення [Електронний ресурс]. – Режим доступу: <https://agilemanifesto.org/iso/uk/manifesto.html> (дата звернення: 23.05.2025).
2. Anderson D. J., Shalloway A., Denning S. D. Lessons In Agile Management On The Road To Kanban. – Blue Hole Press, 2012. – 434 с.
3. Aroral H. K. Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis // International Journal of Applied Business and Management Studies. – 2021. – Т. 6, № 1. – С. 91–99. – Режим доступу: http://www.ijabms.com/wp-content/uploads/2021/05/05_ARORAL_PB.pdf (дата звернення: 23.05.2025).
4. A study on using Waterfall and Agile methods in software project management / B.-A. Andrei та ін. // Journal of Information Systems & Operations Management. – 2019. – Т. 13, № 1. – С. 125–135. – Режим доступу: <https://web.rau.ro/websites/jisom/Vol.13%20No.1%20-%202019/JISOM-SU19-A12.pdf> (дата звернення: 23.05.2025).
5. Waterfall model – Wikipedia [Електронний ресурс] // Wikipedia, the free encyclopedia. – Режим доступу: https://en.wikipedia.org/wiki/Waterfall_model#cite_note-5 (дата звернення: 23.05.2025).
6. Highsmith J., Cockburn A. Agile software development: the business of innovation // Computer. – 2001. – Т. 34, № 9. – С. 120–127. – DOI: <https://doi.org/10.1109/2.947100> (дата звернення: 23.05.2025).
7. John. Understanding Scrum [Електронний ресурс] // Institute of Project Management. – Режим доступу: <https://projectmanagement.ie/blog/understanding-scrum/> (дата звернення: 23.05.2025).
8. Lee Q. Toyota Survives Bombing & Economic Chaos... {Strategos} [Електронний ресурс] // Strategos, Inc, Engineers-Consultants-Strategists. – Режим доступу:

https://www.strategosinc.com/RESOURCES/04-Lean_History/toyota_crises.htm (дата звернення: 23.05.2025).

9. Maassen M. A. Product development models in the IT sector – From Waterfall to Agile Project Management Models in the case of AVIRA SOFT S.R.L. // Proceedings of the International Conference on Business Excellence. – 2023. – Т. 12, № 1. – С. 568–578. – DOI: <https://doi.org/10.2478/picbe-2018-0051> (дата звернення: 23.05.2025).

10. Mkrtchyan R. The Code and Fix Model [Електронний ресурс] // Medium. – Режим доступу: <https://medium.productcoalition.com/the-code-and-fix-model-2cabd4c48166> (дата звернення: 23.05.2025).

11. Panel U.S.N.M.C.A. Symposium on advanced programming methods for digital computers: Washington, D.C., June 28, 29, 1955. – Washington, D.C.: Office of Naval Research, Dept. of the Navy, 1956. – 83 с.

12. Richardson T., Marion J. Managing Projects with PMBOK 7: Connecting New Principles with Old Standards. – Business Expert Press, 2022.

13. Royce W. W. Managing the Development of Large Software Systems (1970) // Ideas That Created the Future. – 2021. – С. 321–332. – DOI: <https://doi.org/10.7551/mitpress/12274.003.0035> (дата звернення: 23.05.2025).

14. Sachdeva S. Scrum Methodology // International Journal Of Engineering And Computer Science. – 2016. – DOI: <https://doi.org/10.18535/ijecs/v5i6.11> (дата звернення: 23.05.2025).

15. Schwaber K. SCRUM Development Process // Business Object Design and Implementation. – London, 1997. – С. 117–134. – DOI: https://doi.org/10.1007/978-1-4471-0947-1_11 (дата звернення: 23.05.2025).

16. Takeuchi H., Nonaka I. The New New Product Development Game // Harvard Business Review. – 2023. – С. 137–146. – Режим доступу: <http://damiantgordon.com/Methodologies/Papers/The%20New%20Product%20Development%20Game.pdf> (дата звернення: 23.05.2025).