



Управління проектами Waterfall vs Agile Вступ

Вступ

Проектне управління – це ключова дисципліна, яка допомагає організовувати та координувати процеси для досягнення поставлених цілей у визначені терміни та з оптимальним використанням ресурсів.

У сучасному світі бізнесу та технологій ефективне управління проектами є критично важливим фактором успіху компаній та організацій. Воно включає планування, розподіл ресурсів, управління ризиками та контроль виконання робіт.

Аспекти управління проектами

Основні аспекти управління проектами:

- Цілі та завдання проекту** – чітке визначення очікуваних результатів.
- Життєвий цикл проекту** – послідовність етапів, через які проходить проект (ініціація, планування, виконання, контроль та завершення).
- Методології управління проектами** – популярні підходи, такі як Waterfall, Agile, Scrum, Kanban, XP.
- Роль керівника проекту** – управління командою, комунікація та прийняття рішень.

Цілі та завдання проекту

Перший і найважливіший крок у проектному управлінні – це визначення його **цілей та завдань**. Вони мають бути чітко сформульовані, вимірювані та досяжні.

Для адекватного визначення цілей та завдань використовується акронім **SMART**

- **SMART-критерії** (Specific, Measurable, Achievable, Relevant, Time-bound):
 - *Specific* (конкретні) – цілі повинні бути чітко визначені.
 - *Measurable* (вимірювані) – повинен бути спосіб оцінки прогресу.
 - *Achievable* (досяжні) – цілі мають бути реалістичними.
 - *Relevant* (актуальні) – вони повинні відповідати загальній стратегії компанії.
 - *Time-bound* (обмежені в часі) – необхідно встановити конкретні дедлайни.

SMART-критерії - Specific

Ціль має бути **чіткою та однозначною**, щоб кожен член команди розумів, **що саме потрібно досягти**. Вона не повинна містити загальних формулювань або розмитих понять.

- Визначити **хто** буде виконувати завдання.
- Вказати **що саме** потрібно зробити.
- Описати **де та коли** має відбутися реалізація.
- Уточнити **чому** це важливо.

«Поліпшити обслуговування клієнтів» – занадто загально.

«Зменшити середній час відповіді на запит клієнта до 2 хвилин шляхом автоматизації процесу підтримки» – конкретно, зрозуміло, що потрібно зробити.

SMART-критерії - Measurable

Ціль повинна мати чіткі **кількісні** або **якісні** показники, за якими можна оцінити її виконання.

- Додати **конкретні числові значення** (відсотки, суми, терміни).
- Використовувати **чіткі метрики** для оцінки результату.

«Збільшити продажі» – немає можливості оцінити, чи досягнута ціль.

«Збільшити продажі на 15% у наступні 6 місяців за рахунок впровадження нової маркетингової стратегії» – можна відстежити прогрес і результат.

SMART-критерії - Achievable

Ціль має бути **реалістичною** з урахуванням ресурсів, часу, навичок команди та можливостей компанії.

- Оцінити **наявні ресурси** та **можливості**.
- Врахувати **зовнішні фактори**, що можуть **впливати на результат**.
- Переконатися, що ціль є **реалістичною**, але водночас **мотивуючою**.

«Збільшити продажі на 300% за 1 місяць» – занадто амбітно й нереально.

«Збільшити продажі на 10% за три місяці шляхом проведення рекламної кампанії» – ціль є складною, але досяжною.

SMART-критерії - Relevant

Ціль повинна бути **важливою та відповідати загальній стратегії компанії** або бізнесу. Вона має приносити реальну користь і бути пов'язаною з ключовими пріоритетами.

- Переконатися, що вона **відповідає місії компанії**.
- Оцінити, чи буде її досягнення **корисним для бізнесу**.
- Визначити, **як вона впливає на загальну стратегію**.

«Запустити нову лінію продукції в категорії, якою компанія не займається» – не відповідає основному напрямку діяльності.

«Запустити новий канал продажів у соцмережах для розширення аудиторії та підвищення прибутку на 20%» – ціль відповідає стратегії розвитку бізнесу.

SMART-критерії - Time-bound

Будь-яка ціль повинна мати **чіткий дедлайн** або часовий проміжок для її реалізації. Це допомагає контролювати процес виконання та розподіляти ресурси.

- Встановити **кінцевий термін** виконання.
- Визначити **проміжні етапи** (якщо ціль довгострокова).
- Врахувати **реальні строки** для досягнення результату.

«Зменшити витрати» – не вказано, до якого часу.

«Зменшити витрати на логістику на 10% протягом наступних 6 місяців шляхом оптимізації маршрутів» – є чіткий термін для реалізації.

SMART-критерії - Приклад

non-SMART ціль:

"Збільшити трафік на сайт"

SMART-ціль:

"Збільшити трафік на сайт на 25% за 4 місяці шляхом SEO-оптимізації та запуску рекламних кампаній у соціальних мережах"

-Specific – чітко зазначено, що потрібно зробити.

-Measurable – є точний показник (+25%).

-Achievable – реалістично для середнього бізнесу.

-Relevant – відповідає стратегії компанії.

-Time-bound – визначено термін (4 місяці).

Життєвий цикл проекту

Кожен проект проходить через **кілька основних етапів**, які разом утворюють **життєвий цикл проекту**.

Основні фази життєвого циклу:

1. Ініціація
2. Планування
3. Виконання
4. Контроль та моніторинг
5. Завершення

Життєвий цикл проекту - Ініціація

На цьому етапі відбувається **зародження проекту**: аналізується його доцільність, визначаються цілі та зацікавлені сторони. Основна мета – оцінити, **чи варто** реалізовувати проект і **чи відповідає він** стратегічним цілям компанії.

- Визначення **основної ідеї проекту**.
- Аналіз **доцільності та можливих вигод**.
- Виявлення **зацікавлених сторін** (*stakeholders*), які впливають на проект.
- Визначення **основних ризиків** та способів їх уникнення.
- Формулювання **бізнес-кейсу** (опису економічного обґрунтування проекту).
- Розробка **первинного проектного документа – Project Charter** (статуту проекту), в якому фіксуються ключові параметри.

Життєвий цикл проекту - Планування

Цей етап є критичним, оскільки від якісного планування **залежить успіх виконання проекту**. Планування визначає **графік робіт, ресурси, бюджет, ризики та методи управління проектом**.

- Визначення **детального переліку завдань**.
- Побудова **графіка виконання** (діаграма Гантта, etc).
- Розподіл **ролей та відповідальності** в команді.
- Оцінка **бюджету** та необхідних ресурсів.
- Визначення **ключових показників ефективності (KPIs)**.
- Аналіз **можливих ризиків** та розробка плану їх мінімізації.
- Вибір **методології управління проектом** (Agile, Waterfall, Scrum, etc).

Життєвий цикл проекту - Виконання

Це **найбільш активний етап**, коли команда приступає до роботи. Тут відбувається **безпосередня реалізація** завдань згідно з планом.

Основні завдання фази виконання:

- Виконання **запланованих завдань**.
- **Контроль** витрат та використання ресурсів.
- Управління **змiнами в проєкті** (якщо вони необхідні).
- **Постійна комунікація** між членами команди та зацікавленими сторонами.
- Використання **інструментів моніторингу** та управління (Jira, Trello, MS Project).

Життєвий цикл проекту - Контроль та моніторинг

Ця фаза проходить паралельно з виконанням проекту. Основне завдання – відстежувати прогрес та вчасно реагувати на відхилення від плану.

Основні завдання фази контролю:

- Відстеження графіку виконання робіт (згідно з діаграмою Гантта).
- Аналіз відхилень у бюджеті та ресурсах.
- Оцінка якості проміжних результатів.
- Виявлення ризиків та усунення проблем.
- Проведення регулярних зустрічей команди (*stand-up meetings*).

Життєвий цикл проекту - Завершення проекту

На цьому етапі проект **офіційно завершується**, продукт передається замовнику, команда **аналізує результати та робить висновки**.

Основні завдання фази завершення:

- Оцінка досягнення поставлених цілей.
- Передача проекту або продукту кінцевому користувачеві.
- Закриття фінансових питань та підбиття підсумків бюджету.
- **Аналіз** виконаних робіт та **документування** отриманого досвіду.
- Проведення **ретроспективи** для визначення, що можна покращити у майбутніх проектах.

Життєвий цикл проекту - Підсумовуємо

Життєвий цикл проекту допомагає структурувати процеси та ефективно керувати всіма його етапами. Кожна фаза має свою важливу роль:

- **Ініціація** – визначення цілей та обґрунтування проекту.
- **Планування** – створення дорожньої карти реалізації.
- **Виконання** – втілення проекту в життя.
- **Контроль** – моніторинг прогресу та коригування плану.
- **Завершення** – підбиття підсумків та аналіз результатів.

Методології управління проектами

Методологія управління проектами **визначає, яким способом буде організована робота над проектом**. Існує кілька основних підходів:

- Каскадна модель (Waterfall)
- Гнучка методологія (Agile)
- Scrum
- Kanban

Методології управління проектами - Waterfall

Концепція Waterfall була вперше формально описана у **1970 році Вінстоном Ройсом**, американським комп'ютерним науковцем, який працював у компанії Lockheed Martin.

До 1970-х років розробка програмного забезпечення **не мала чіткої структури**, що часто призводило до **хаосу**, перевищення бюджету, затримок та низької якості. Вінстон Ройс у своїй знаменитій статті "**Managing the Development of Large Software Systems**" запропонував **послідовний підхід** до управління проектами.

Назва "Waterfall" (у перекладі – "водоспад") походить від **графічного зображення процесу**, де кожен етап плавно "стікає" вниз до наступного, подібно до водоспаду.

Методології управління проектами - Waterfall

У Waterfall кожен етап виконується **послідовно**, і наступний етап починається тільки після завершення попереднього.

Особливості методології:

- **Лінійна структура** – кожен етап слідує за попереднім без повернення назад.
- **Повна документація** – перед початком роботи необхідно створити всі технічні специфікації.
- **Жорстке управління змінами** – внесення змін після запуску етапу дуже складне.
- **Чітке визначення вимог на початку проекту.**
- **Проект завершується повністю, перш ніж передати продукт замовнику.**

Методології управління проектами - Waterfall

Waterfall складається з **шести ключових етапів**:

- Визначення вимог (Requirements Analysis)
- Проектування (System & Software Design)
- Реалізація (Implementation / Coding)
- Тестування (Verification / Testing)
- Впровадження та запуск (Deployment & Release)
- Підтримка та обслуговування (Maintenance & Support)

Методології управління проектами - Waterfall - Requirements Analysis

Основна мета етапу:

- **Чітко визначити**, що потрібно створити.
- Зібрати **всі вимоги** до системи від замовника, користувачів та інших зацікавлених сторін.
- Задокументувати всі вимоги у вигляді **Software Requirement Specification (SRS)**.

Що роблять на цьому етапі?

- Аналізують потреби бізнесу або кінцевих користувачів.
- Фіксують технічні, функціональні та нефункціональні вимоги.
- Визначають обмеження (бюджет, терміни, технології).
- Складають специфікацію вимог, яка затверджується замовником.

Методології управління проектами - Waterfall - System & Software Design

Основна мета етапу:

- **Перетворити вимоги** на чітку архітектуру майбутнього продукту.
- **Створити детальний** план розробки.
- **Визначити** технології, інструменти та бази даних.

Що роблять на цьому етапі?

- Розробляють **архітектуру системи** (включаючи модулі, компоненти, інтерфейси).
- Визначають **технологічний стек** (наприклад, Java + Spring для бекенду, React для фронтенду).
- Готують макети UI/UX** (дизайн інтерфейсу).
- Створюють **технічну документацію** для програмістів.

Методології управління проектами - Waterfall - Implementation / Coding

Основна мета етапу:

- Написати програмний код **відповідно до вимог і дизайну**.
- **Перетворити** технічну документацію в **працюючий продукт**.

Що роблять на цьому етапі?

- Програмісти пишуть код згідно з **технічними вимогами**.
- Використовують **систему контролю версій** (Git, GitHub).
- Інтегрують **різні компоненти** (наприклад, база даних, API, бекенд, фронтенд).
- Ведуть **внутрішнє тестування** на рівні модулів.

Методології управління проектами - Waterfall - Verification / Testing

Основна мета етапу:

- Переконатися, що продукт **працює правильно, без помилок.**
- Гарантувати відповідність** системи вимогам, визначеним у першому етапі.

Що роблять на цьому етапі?

- Юніт-тестування** (перевірка окремих модулів).
- Інтеграційне тестування** (чи коректно взаємодіють різні компоненти).
- Системне тестування** (повна перевірка всього продукту).
- Регресійне тестування** (перевірка після виправлення помилок).
- Безпекове тестування** (виявлення вразливостей).

Методології управління проектами - Waterfall - Deployment & Release

Основна мета етапу:

- Запустити продукт у **реальне використання**.
- Передати систему замовнику або кінцевим користувачам.

Що роблять на цьому етапі?

- Розгортають продукт на сервері або публікують у маркетплейсах (App Store, Google Play).
- Надають користувачам доступ до системи.
- Проводять навчання персоналу (за потреби).

Методології управління проектами - Waterfall - Maintenance & Support

Основна мета етапу:

- Підтримувати працездатність системи після запуску.
- Виправляти помилки, які не були виявлені на етапі тестування.
- Вносити оновлення та вдосконалення.

Що роблять на цьому етапі?

- Моніторять продуктивність системи.
- Виправляють знайдені помилки (баг-фікси).
- Додають нові функції за запитами користувачів.

Методології управління проектами - Waterfall - Підсумовуємо

Фази Waterfall чітко структуровані:

- Визначення вимог** → що потрібно створити.
- Проектування** → як це зробити.
- Реалізація** → написання коду.
- Тестування** → перевірка працездатності.
- Впровадження** → запуск продукту.
- Підтримка** → оновлення та виправлення помилок.

Цей метод підходить для **стабільних, передбачуваних проектів**, де вимоги зрозумілі заздалегідь.

Методології управління проектами - Agile

До появи Agile у 1990-х роках більшість ІТ-проектів керувалися **традиційними підходами (Waterfall)**. Це часто призводило до того, що:

- Продукт **розроблявся роками**, перш ніж користувачі могли його побачити.
- Зміни вимог **вимагали значних витрат** і переробок.
- Велика кількість бюрократії **уповільнювала процес**.

Через ці проблеми у 90-х роках з'явилися **нові підходи до розробки програмного забезпечення**, які зосереджувалися на **швидких ітераціях, взаємодії з клієнтами та постійних оновленнях**.

Методології управління проектами - Agile

У 2001 році група з **17 розробників** зібралася в гірському курорті Сноуберд (штат Юта, США), щоб обговорити проблеми традиційного управління проектами. Серед них були **Кент Бек, Джефф Сазерленд, Кен Швабер, Мартін Фаулер та інші.**

Вони розробили **Agile Manifesto** (<https://agilemanifesto.org>) – документ, що містить **чотири основні цінності та 12 принципів Agile.**

Методології управління проектами - Agile - Agile Manifesto

Agile Manifesto складається з **12** принципів та **4** ключових цінностей:

Цінності:

- **Люди та взаємодія важливіші, ніж процеси та інструменти.**
- **Працюючий продукт важливіший, ніж вичерпна документація.**
- **Співпраця із замовником важливіша, ніж узгодження контракту.**
- **Готовність до змін важливіша, ніж дотримання початкового плану.**

Методології управління проектами - Agile - Agile Manifesto

Принципи:

- Задоволення потреб клієнта завдяки швидкому та безперервному постачанню продукту.
- Приймаємо зміни, навіть на пізніх етапах розробки.
- Часте постачання працюючого продукту (кожні кілька тижнів або місяців).
- Співпраця між бізнесом та розробниками протягом усього проекту.
- Побудова проекту навколо мотивованих людей, які отримують необхідні ресурси.
- Найкращий спосіб передати інформацію – особисте спілкування.

Методології управління проектами - Agile - Agile Manifesto

Принципи:

- Працюючий продукт – головний показник прогресу.
- Стабільний темп роботи команди протягом усього проекту.
- Постійна увага до технічної досконалості та гарного дизайну.
- Простота – ключовий елемент ефективної роботи.
- Самоорганізовані команди створюють найкращі рішення.
- Регулярний аналіз і коригування процесів для підвищення ефективності.

Методології управління проектами - Agile - Основні методології

Agile – це **не одна методологія**, а **набір принципів**, на основі яких розроблено різні фреймворки:

- **Scrum** – найпопулярніший підхід, що базується на коротких спринтах.
- **Kanban** – візуальне управління завданнями за допомогою дошки.
- **Extreme Programming (XP)** – орієнтований на якість коду та часте тестування.
- **Lean** – фокус на мінімізації витрат та постійному вдосконаленні.

Методології управління проектами - Agile - Scrum

Scrum — це найпопулярніший Agile-фреймворк, який дозволяє **розробляти продукт короткими циклами (спринтами)** тривалістю 1-4 тижні. Команда **регулярно аналізує** результати роботи та **адаптується** до змін.

Основні ролі в Scrum:

- **Scrum Master** – контролює процес, усуває перешкоди.
- **Product Owner** – визначає вимоги, пріоритети та взаємодіє з клієнтом.
- **Development Team** – безпосередньо виконує завдання.

Методології управління проектами - Agile - Scrum

Цикл Scrum:

- 1.Формується **беклог** (список завдань).
- 2.Команда обирає завдання для найближчого спринту (1-4 тижні).
- 3.Щоденні **Scrum-зустрічі** для обговорення прогресу.
- 4.Після спринту команда проводить **демо-презентацію**.
- 5.Відбувається **ретроспектива** для покращення процесу.

Використовується:

- ІТ (розробка програмного забезпечення)
- Фінансовий сектор
- Стартапи

Методології управління проектами - Agile - Kanban

Kanban — це метод управління завданнями, який використовує **візуальну дошку (Kanban board)** для відстеження **процесу виконання роботи**.

Основні елементи:

- **Kanban-дошка** – поділяється на три основні стовпці:
 - **To Do** – завдання, які потрібно виконати.
 - **In Progress** – завдання, які зараз у роботі.
 - **Done** – виконані завдання.
- **Ліміти робіт (WIP Limits)** – обмеження кількості завдань, що виконуються одночасно.

Методології управління проектами - Agile - Extreme Programming

Extreme Programming (XP) — це методологія, яка зосереджується на **якісному написанні коду та частому тестуванні**.

Основні практики:

- **Парне програмування** — два розробники пишуть код разом.
- **Тестування (Test-Driven Development, TDD)** — спочатку пишуться тести, потім код.
- **Часті релізи (Continuous Integration)** — оновлення продукту кілька разів на день.

Використовується:

- Розробка програмного забезпечення
- Стартапи

Методології управління проектами - Agile - Lean

Lean — це підхід, який фокусується на **усуненні зайвих процесів та витрат**.

Основні принципи Lean:

- **Усунення** всіх непотрібних дій (оптимізація роботи).
- Використання **зворотного зв'язку** від клієнта.
- Постійне **вдосконалення** процесів.

Використовується:

- Виробництво
- Логістика
- ІТ та бізнес

Методології управління проектами - Agile - Підсумовуємо

- **Scrum** – для ітеративного підходу (найпопулярніший).
- **Kanban** – для оптимізації робочого процесу.
- **XP** – для підвищення якості коду.
- **Lean** – для усунення зайвих витрат.

Методології управління проектами - Waterfall vs Agile

Параметр	Waterfall	Agile
Підхід	Лінійний, послідовний	Гнучкий, ітераційний
Зміни в проекті	Дуже складно внести після початку	Легко адаптувати на будь-якому етапі
Взаємодія з клієнтом	Мінімальна	Постійна комунікація
Час отримання результату	Тільки після завершення всіх етапів	Робочий продукт уже після першого циклу
Тестування	В кінці розробки	Постійне тестування
Використання	Будівництво, медицина, виробництво	ІТ, стартапи, маркетинг, фінанси

Методології управління проектами - Програмні продукти - Waterfall

Програма	Опис	Функції
Microsoft Project	Потужний інструмент від Microsoft для управління проектами	Діаграми Ганта, Управління ресурсами, Детальне планування
Wrike	Хмарний сервіс для управління проектами	Чітке відстеження дедлайнів, Документообіг, Гнучке налаштування статусів
Oracle Primavera	Використовується для великих корпоративних	Планування термінів, Аналіз ризиків, Управління бюджетом

Методології управління проектами - Програмні продукти - Agile

Програма	Опис	Функції
Jira	Найпопулярніший інструмент для Agile-команд	Scrum-дошки, Kanban, Звіти про ефективність
Trello (Agile режим)	Візуальна Kanban-система	Дошки завдань, Гнучка структура, Автоматизація процесів
Asana	Гнучкий інструмент для управління завданнями	Scrum, Kanban, Колективна робота, Відстеження прогресу
ClickUp	Інструмент для управління Agile-командами	Scrum-борди, Управління беклогом, Вбудований чат

Методології управління проектами - Програмні продукти - Універсальні інструменти

Програма	Опис	Waterfall функції	Agile функції
Jira	Потужний інструмент для гнучкої розробки	Діаграми Ганта	Scrum, Kanban
Microsoft Project	Класичний корпоративний інструмент	Планування Waterfall	Дошки Agile
ClickUp	Інтуїтивне управління командами	Відстеження прогресу	Гнучке управління

Роль керівника проекту

Керівник проекту (Project Manager, PM) – це людина, яка відповідає за успішне виконання проекту.

Основні обов'язки PM:

- **Планування та управління ресурсами.**
- **Організація роботи команди.**
- **Комунікація із зацікавленими сторонами.**
- **Контроль якості виконання завдань.**
- **Управління ризиками та усунення проблем.**

Методології управління проектами - Висновок

Життєвий цикл проекту включає п'ять основних фаз: ініціацію, планування, виконання, контроль та завершення. **Кожен етап важливий** для успішної реалізації проекту.

Методології управління проектами діляться на **традиційні (Waterfall)** та **гнучкі (Agile)**.

- **Waterfall** – підходить для стабільних, передбачуваних проектів.
- **Agile** – забезпечує гнучкість і швидку адаптацію до змін.

Waterfall (каскадна модель) – послідовний підхід, де кожен етап завершується перед початком наступного. Він підходить для **будівництва, виробництва, медичних та державних проектів**, де важлива чітка документація та контроль над процесами.

Методології управління проектами - Висновок

Agile (гнучка методологія) – підхід, що базується на ітераціях, швидких оновленнях та постійному зворотному зв'язку. Він ідеально підходить для розробки програмного забезпечення, стартапів і швидкозмінних ринків.

Роль керівника проекту (PM) – критично важлива. PM відповідає за планування, контроль, управління ресурсами, комунікацію та розв'язання проблем. Успішний керівник проекту поєднує технічні, аналітичні та лідерські навички.

Інструменти управління проектами

- Для **Waterfall: Microsoft Project, Oracle Primavera.**
- Для **Agile: Jira, Trello, Asana, Monday.com.**
- Для **гібридного підходу: Wrike, ClickUp, Microsoft Project.**