

# Тестування та верифікація програмного забезпечення

## Лекція №8

Тема: Командна робота, підходи до розробки програмного забезпечення

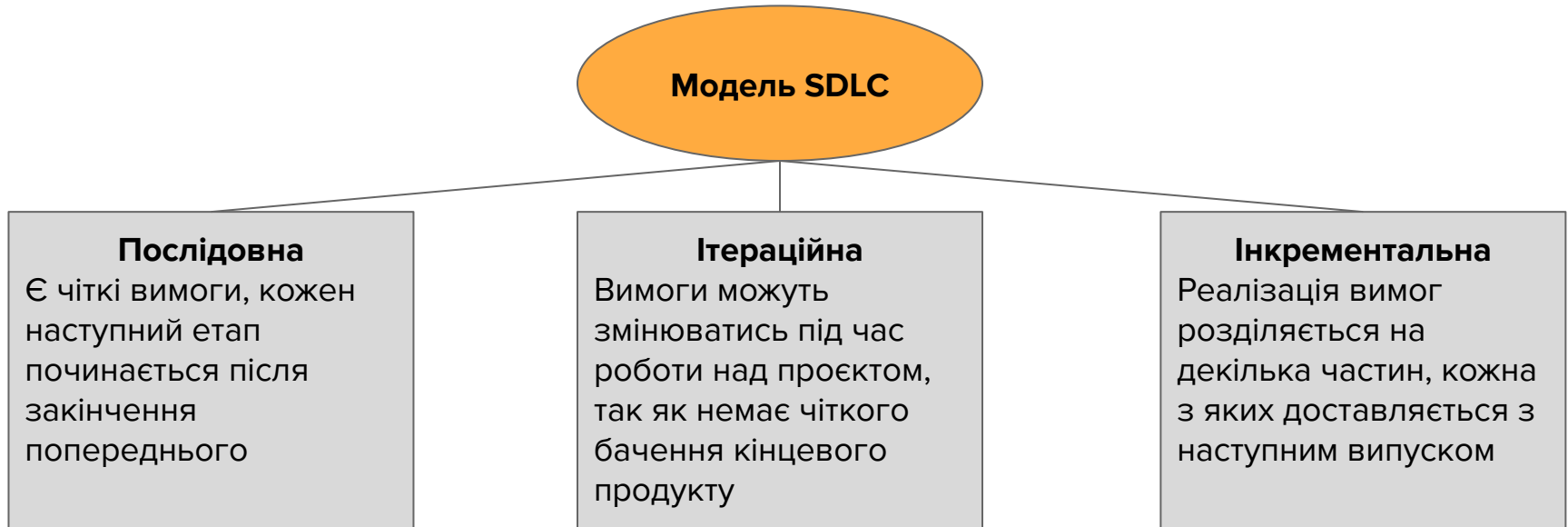
---

## Питання лекції

1. Класичні моделі розробки ПЗ: каскадна (водоспадна), ітераційна, інкрементна.
2. Промислові технології розробки ПЗ: RUP, MSF, Agile, Scrum, Extreme Programming (XP), Kanban.
3. Ролі та обов'язки у команді проєкту, переваги командної роботи, ризики та складність такої співпраці.
4. Основні етапи планування і виконання ІТ проєкту. Життєвий цикл ІТ проєкту.

# Класичні моделі розробки ПЗ

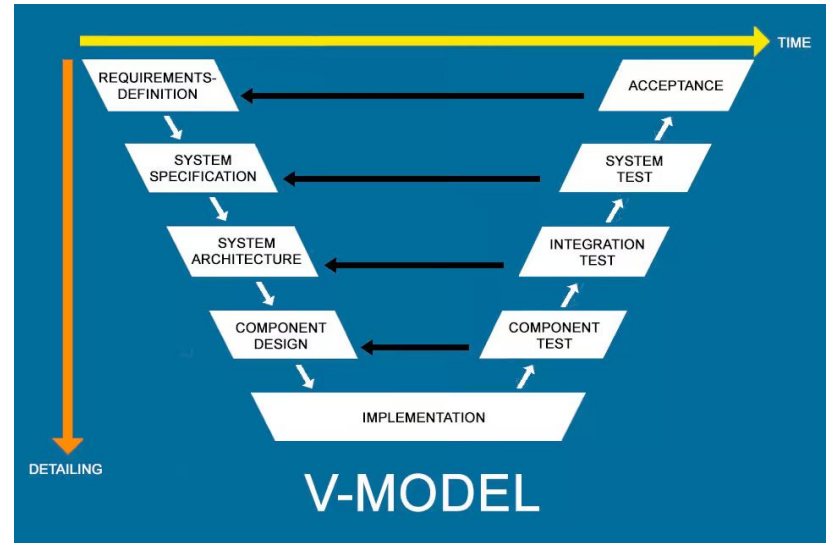
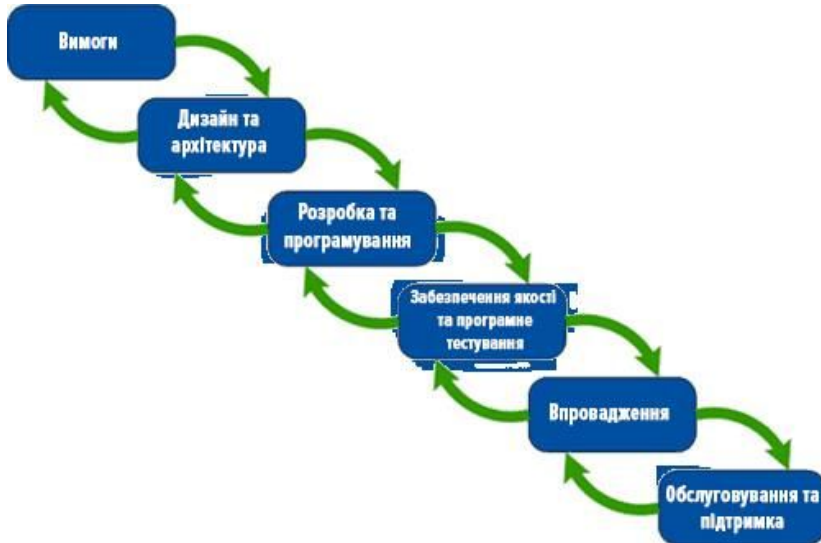
**Модель SDLC** — це абстрактне, високорівневе представлення процесу розробки програмного забезпечення. Модель SDLC визначає, як різні фази розробки та типи дій, що виконуються в рамках цього процесу, пов'язані між собою, як логічно, так і хронологічно.



# Класичні моделі розробки ПЗ

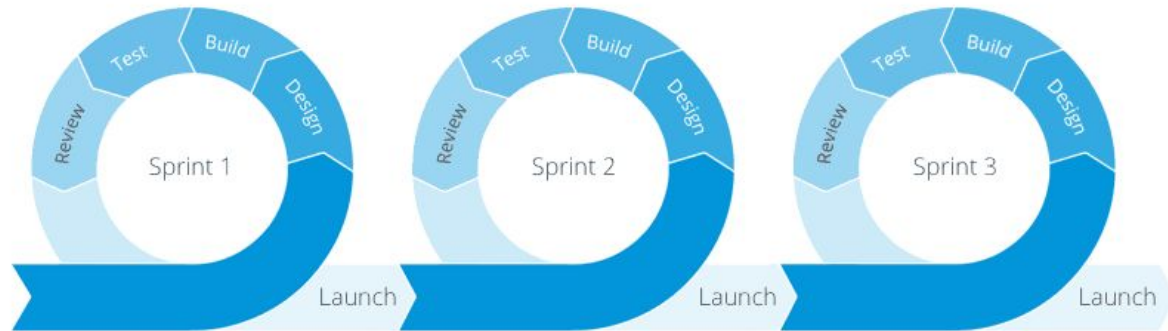
## Послідовні моделі SDLC:

- Каскадна (Waterfall)
- V-модель



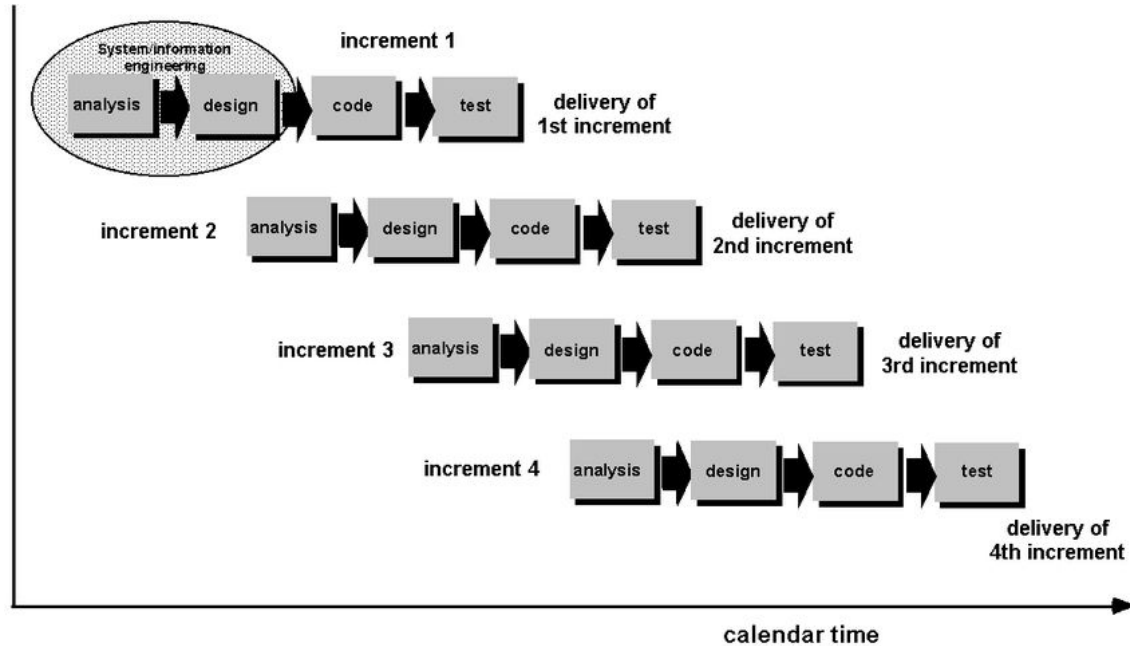
# Класичні моделі розробки ПЗ

## Ітераційна модель SDLC



# Класичні моделі розробки ПЗ

## Інкрементальна модель SDLC



# Промислові технології розробки ПЗ

## Rational Unified Process

Rational Unified Process (RUP) є ітеративним процесом розробки програмного забезпечення створеним Rational Software — підрозділом IBM з 2003.

RUP заснований на наборі будівельних блоків, чи містить елементи, що описують те, що повинно бути зробленим, необхідні навички, та покрокове пояснення того, як досягаються конкретні цілі розробки. Основними будівельними блоками, чи елементами вмісту, є наступні:

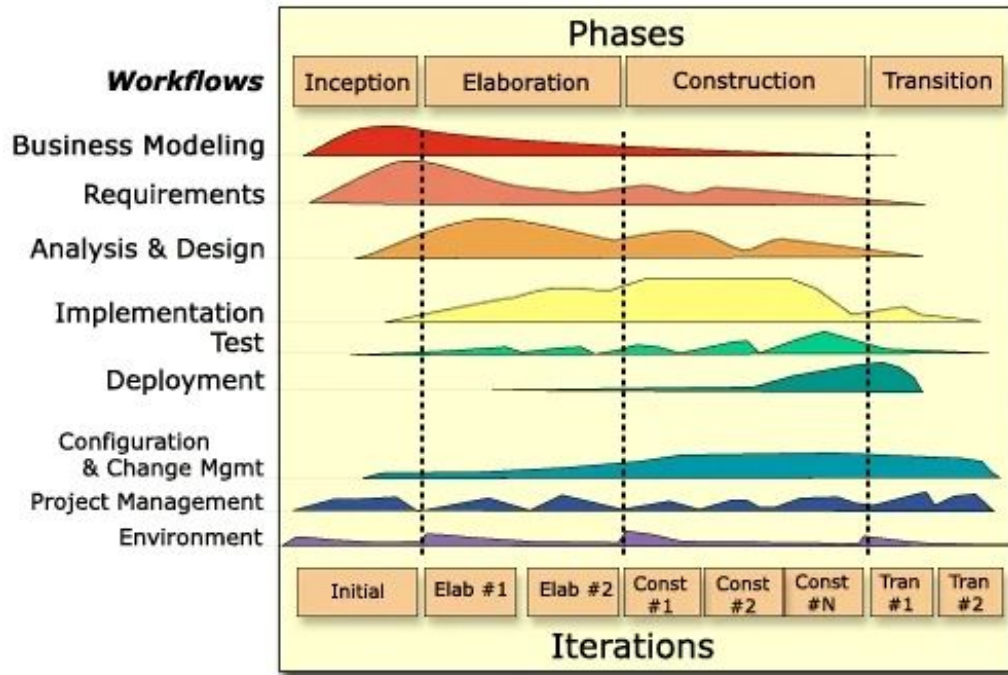
- **Ролі** (хто). Роль визначає набір навичок, компетенції та відповідальності.
- **Робочі продукти** (що). Робочий продукт являє собою щось отримане з завдання, в тому числі всі документи і моделі, випущені під час роботи впродовж процесу.
- **Завдання** (як). Завдання описує одиницю роботи, яка доручена ролі, яка забезпечує значущий результат.

У кожній ітерації, завдання діляться на дев'ять дисциплін: шість «інженерних дисциплін» (бізнес-моделювання, вимоги, аналіз і проектування, реалізація, тестування, розгортання) і трьох допоміжних дисциплін (конфігурація і керування змінами, управління проєктами, середовища).



# Промислові технології розробки ПЗ

## Rational Unified Process



# Промислові технології розробки ПЗ

## Microsoft Solutions Framework

Microsoft Solutions Framework (MSF) — це гнучка методологія розробки програмного забезпечення, створена компанією Microsoft, яка поєднує найкращі практики управління проектами та розробки ПЗ.

Процес MSF орієнтований на «віхи» (milestones) — ключові точки проекту, що характеризують досягнення у його якогось суттєвого (проміжного чи кінцевого) результату. Цей результат може бути оцінений і проаналізований, що має на увазі відповіді на запитання: «Чи прийшла проектна група до однозначного розуміння цілей і рамок проекту?», «Чи достатньою мірою готовий план дій?», «Чи відповідає продукт затвердженій специфікації?», « Чи задовольняє рішення потреб замовника?» і т.д.

Модель процесів MSF враховує постійні зміни проектних вимог. Вона виходить із того, що розробка рішення повинна складатися з коротких циклів, що створюють поступальний рух від найпростіших версій рішення для його остаточного вигляду.

Модель процесів MSF тісно пов'язані з базовими принципами MSF, розглянутими вище. Взагалі кажучи, трьома особливостями моделі процесів MSF є:

- Підхід, заснований на фазах та віхах.
- Ітеративний підхід.
- Інтегрований підхід до створення та впровадження рішень.



# Промислові технології розробки ПЗ

## Microsoft Solutions Framework



# Промислові технології розробки ПЗ

## Agile

Гнучка розробка програмного забезпечення (англ. Agile software development) — клас методологій розробки програмного забезпечення, що базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між багатофункціональними командами здатними до самоорганізації. Гнучка розробка — засіб для підвищення продуктивності розробників програмного забезпечення.

Основні принципи Agile:

- Люди та взаємодія важливіші за процеси та інструменти.
- Працюючий продукт важливіший за вичерпну документацію.
- Співпраця із замовником важливіша, ніж умови контракту.
- Готовність до змін важливіша за дотримання плану.

-та хто такий цей ваш Agile



# Промислові технології розробки ПЗ

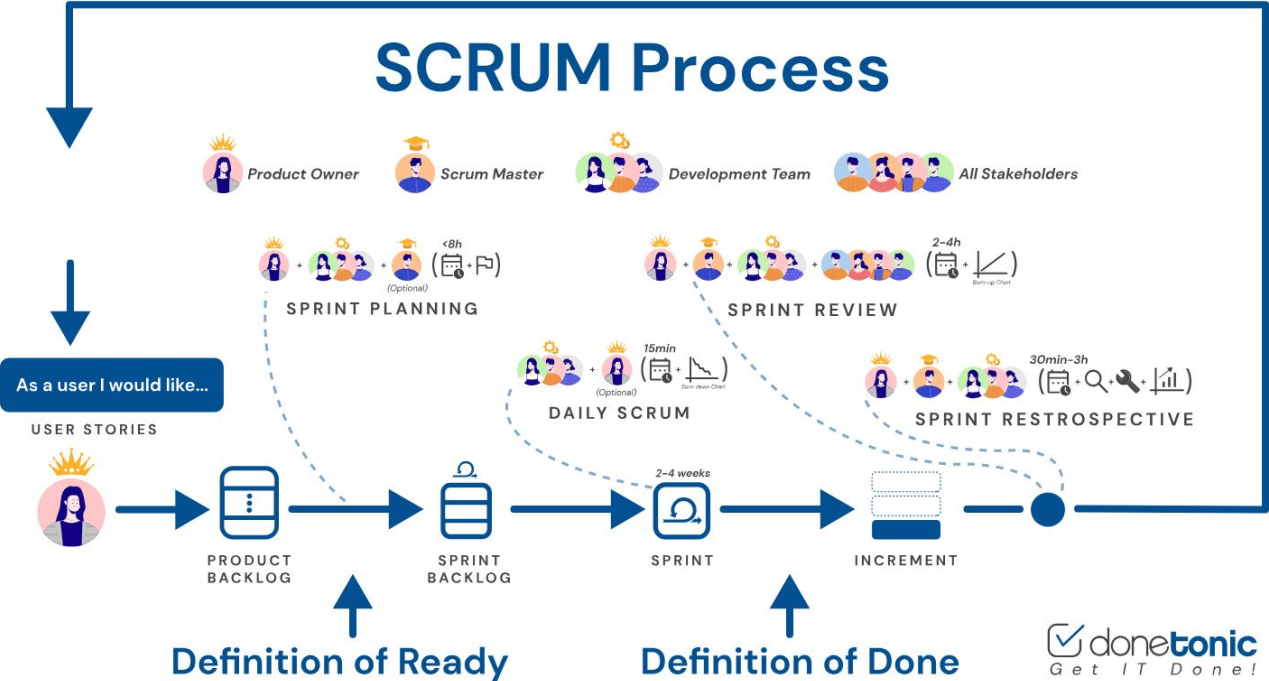
## Scrum

Scrum (Скрам) — це гнучкий фреймворк (методологія) для управління проектами, що базується на ітеративному підході, де кросфункціональні команди створюють продукт поетапно (інкрементами) за короткі цикли — спринти (зазвичай 1–4 тижні). Це допомагає швидко реагувати на зміни, підвищувати якість та прозорість розробки, особливо в умовах невизначеності.



# Промислові технології розробки ПЗ

## Scrum



# Промислові технології розробки ПЗ

## Extreme Programming

Екстремальне програмування (XP) — це гнучка (Agile) методологія розробки ПЗ, спрямована на покращення якості продукту та адаптацію до змінних вимог. Вона базується на частих релізах, парному програмуванні, постійному тестуванні (TDD) та тісній взаємодії із замовником. Основна мета — швидка розробка надійного коду, що відповідає поточним потребам клієнта. Extreme Programming Explained описує екстремальне програмування як дисципліну розробки програмного забезпечення яка змушує людей створювати високоякісне ПЗ якомога швидше.

XP намагається зменшити ціну зміни вимог до ПЗ завдяки малим циклам розробки, а не одним довгим циклом. Екстремальне програмування сприймає зміни до вимог як звичайні, немінучі та бажані аспекти розробки ПЗ, і ці зміни мають бути очікуваними. Основна ідея полягає в тому, що неможливо розробити самодостатній пакет вимог до ПЗ, зміни в вимогах — немінучі.

Екстремальне програмування також вводить набір практик та принципів на основі методології гнучкої розробки програмного забезпечення.



# Промислові технології розробки ПЗ

## Extreme Programming



# Промислові технології розробки ПЗ

## Kanban

Канбан (Kanban) — це метод візуального управління робочими процесами (потоками завдань), спрямований на підвищення прозорості, обмеження незавершеної роботи (WIP) та забезпечення принципу «точно в строк». Він базується на використанні дощок зі стікерами (картками), які рухаються між етапами «Backlog», «В роботі» та «Готово».

- Візуалізація: Робота представлена на дошці, що робить процес прозорим для всієї команди.
- Обмеження незавершеної роботи (WIP): Встановлення лімітів на кількість завдань, що виконуються одночасно, щоб уникнути перевантаження.
- Управління потоком: Завдання рухаються по мірі виконання, що допомагає виявити вузькі місця та оптимізувати час.
- Безперервне вдосконалення (Kaizen): Метод спрямований на еволюційні зміни, зменшуючи опір команди та ризику.

**ВЕСЕЛО ПО ПОЛЮ БІГАЄ** канбанчик

# Ролі та обов'язки у команді проєкту. Командна робота

Основні ролі в IT-проєкті:

- Керівник проєкту (Project Manager)
  - Планує та контролює виконання завдань.
  - Розподіляє ресурси та координує команду.
  - Відповідає за дотримання термінів та бюджету.
- Аналітик (Business/Systems Analyst)
  - Збирає та документує вимоги замовника.
  - Формує специфікації та сценарії використання системи.
- Розробник (Developer)
  - Реалізує функціональні вимоги у коді.
  - Пише тестовані та підтримувані програмні модулі.
- Тестувальник (QA Engineer / Tester)
  - Перевіряє продукт на відповідність вимогам.
  - Виконує валідацію, верифікацію та автоматизоване тестування.
- Дизайнер (UI/UX Designer)
  - Створює зручний та зрозумілий інтерфейс.
  - Забезпечує позитивний досвід користувачів.
- DevOps / Інженер з розгортання
  - Налаштовує середовище розробки та продуктивне середовище.
  - Автоматизує процеси розгортання та підтримки системи.

## EVERY WORKPLACE TEAM



# Основні етапи планування і виконання ІТ проєкту. Життєвий цикл ІТ проєкту.

**Проєкт – це тимчасове зусилля (підприємство), яке здійснюється з метою створення унікального продукту, послуги або результату.**

**Етап 1. Ініціація.** Це фундамент усього проєкту. Тут визначаються основні цілі та завдання, формується вихідна концепція проєкту. У процесі ініціації важливо чітко розуміти, що саме необхідно досягти, і які ресурси будуть задіяні. Основне питання, на яке потрібно відповісти: «Що ми хочемо отримати в результаті?»

**Етап 2. Планування.** На цьому етапі розробляється детальний план проєкту, включно з термінами, бюджетуванням і розподілом ресурсів. Тут також встановлюються метрики успіху проєкту. Ефективне планування — ключ до успішного релізу.

**Етап 3. Виконання.** Тут проєкт переходить від теорії до практики. Відбувається безпосередня реалізація всіх запланованих завдань. Особлива увага приділяється управлінню командами, ресурсами та часом.

**Етап 4. Моніторинг і контроль.** На цій стадії важливо постійно відстежувати його перебіг і порівнювати реальне виконання завдань із планом. Мета цього етапу — своєчасно виявляти та коригувати відхилення від плану, керувати змінами.

**Етап 5. Завершення проєкту.** Останній етап охоплює підбиття підсумків роботи, оцінку досягнутих результатів і надання звітів зацікавленим сторонам. Це час аналізувати, що було зроблено добре і що можна поліпшити в майбутніх проєктах.

Дякую за увагу!