

# Якість та тестування програмного забезпечення

## Лекція №3

Тема: Тестування, верифікація та валідація в життєвому циклі програмного забезпечення

---

## Питання лекції

1. Життєвий цикл розробки програмного забезпечення.
2. Життєвий цикл тестування програмного забезпечення.
3. Верифікація та валідація в контексті життєвого циклу розробки програмного забезпечення

# Життєвий цикл розробки програмного забезпечення

**Життєвий цикл програмного забезпечення** – період часу, який починається з моменту прийняття рішення про необхідність створення програмного продукту і закінчується в момент його повного вилучення з експлуатації.

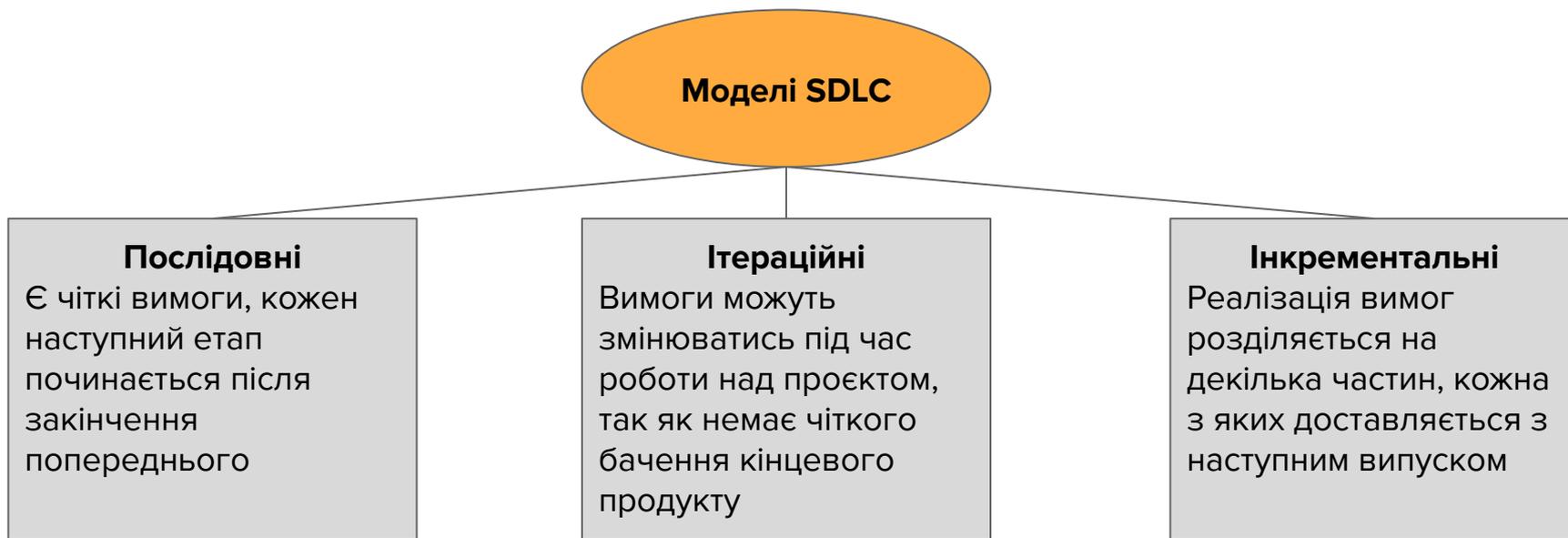
**Життєвий цикл розробки програмного забезпечення** (Software Development Life Cycle, SDLC) - це фреймворк, який визначає кроки, пов'язані з розробкою програмного забезпечення на кожному етапі. Він охоплює детальний план побудови, розгортання та підтримки програмного забезпечення.

Для полегшення проєктування, створення і випуску якісного програмного продукту існують різні **моделі життєвого циклу розробки ПЗ**.



# Життєвий цикл розробки програмного забезпечення

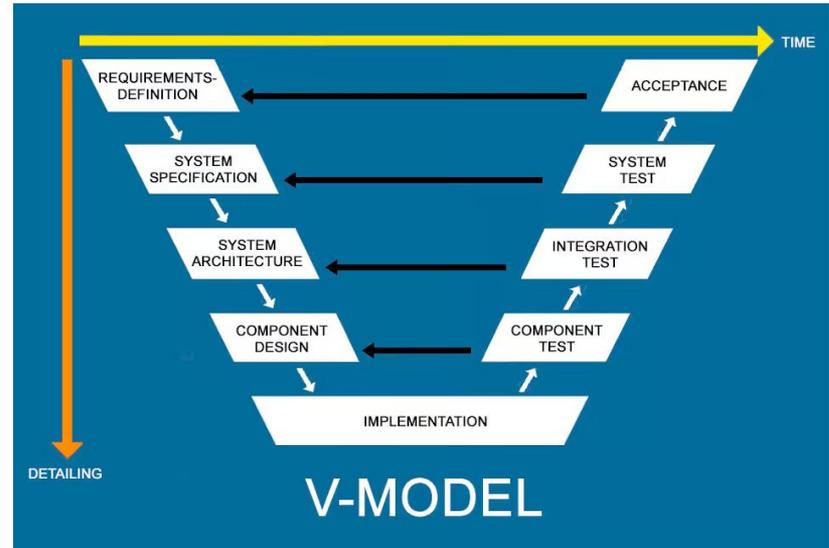
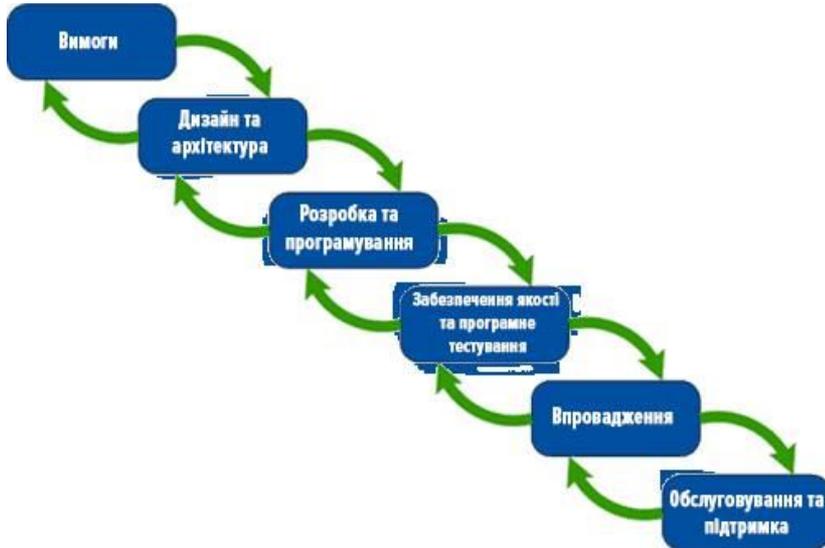
**Модель SDLC** — це абстрактне, високорівневе представлення процесу розробки програмного забезпечення. Модель SDLC визначає, як різні фази розробки та типи дій, що виконуються в рамках цього процесу, пов'язані між собою, як логічно, так і хронологічно.



# Життєвий цикл розробки програмного забезпечення

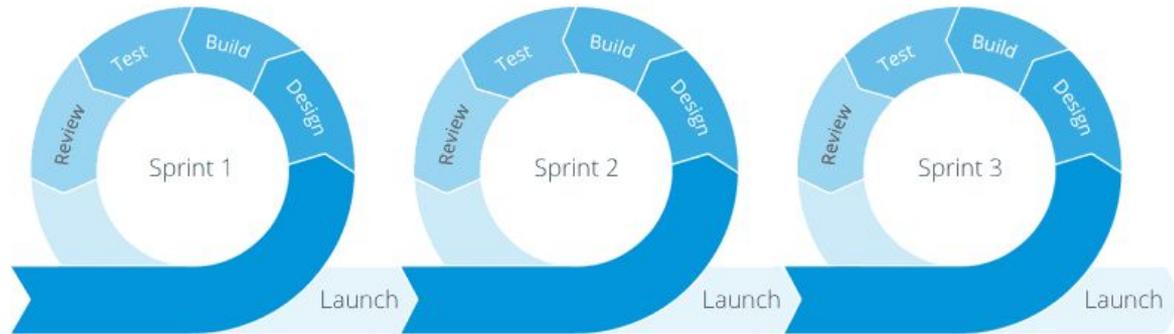
## Послідовні моделі SDLC:

- Каскадна (Waterfall)
- V-модель



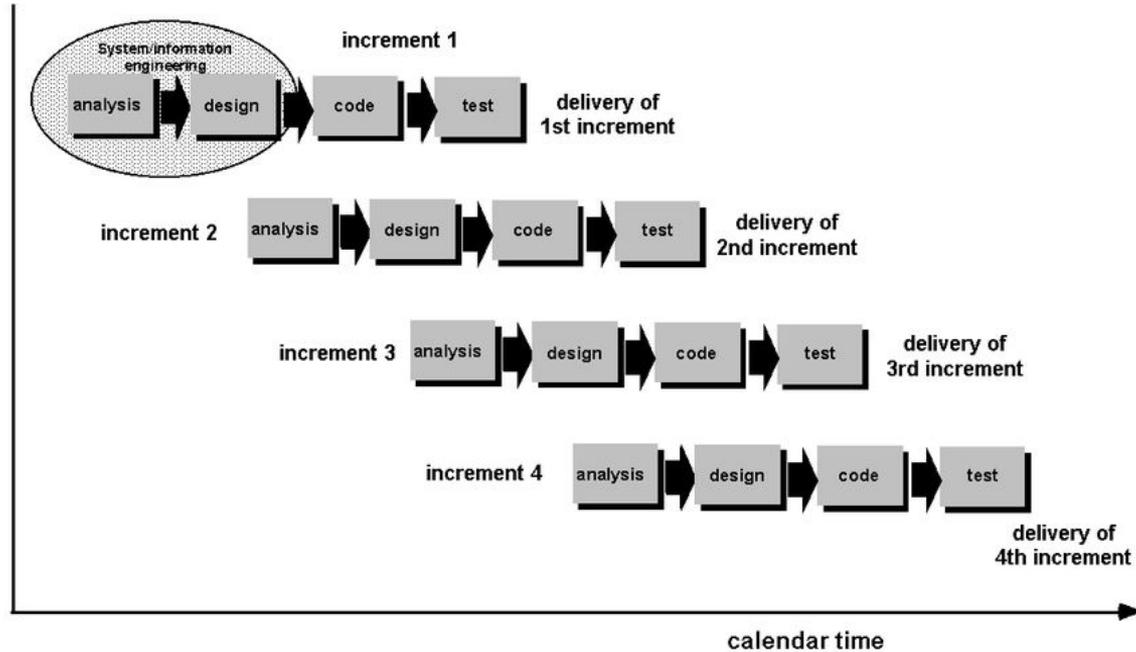
# Життєвий цикл розробки програмного забезпечення

## Ітераційна модель SDLC



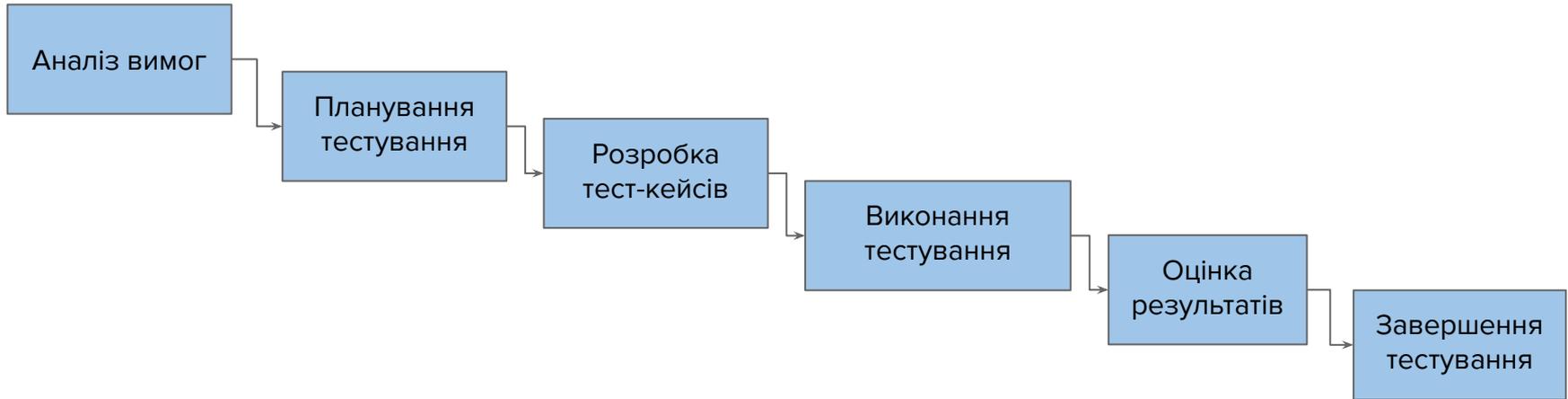
# ЖИТТЄВИЙ ЦИКЛ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Інкрементальна модель SDLC



# Життєвий цикл тестування програмного забезпечення

**Життєвий цикл тестування ПЗ** (Software Testing Life Cycle, STLC) – це визначена послідовність запланованих етапів та дій, що виконуються в процесі тестування програмного забезпечення. При цьому, кожний етап (фаза) виконується систематично, маючи власні цілі та результати. В різних компаніях етапи STLC можуть відрізнятися, проте основа залишається незмінною.



# Життєвий цикл тестування програмного забезпечення

## Аналіз вимог

Тестування на етапі аналізу вимог, має за мету переконатися, що вимоги правильно інтерпретовані, зрозумілі та послідовні. Чітко сформульовані вимоги, допомагають визначити правильні цілі для тестування, та заощадити **час разом з зусиллями та грошима**, в процесі розробки та тестування. Саме на цьому етапі, виправлення потенційних недоліків проекту має відносно невелику вартість, в порівнянні з наступними.

# Життєвий цикл тестування програмного забезпечення

## Планування тестування

Основні завдання:

- Визначення обсягів і ризиків
- Визначення цілей тестування та підходів тестування
- Визначення необхідних ресурсів
- Планування тест дизайну, впровадження та виконання тестування
- Визначення "вихідних" критеріїв
- Створення, узгодження та поширення тест плану

Артефакти даного етапу:

- Test Strategy
- Test Plan
- Configuration Matrix
- Test Hardware List



# ЖИТТЄВИЙ ЦИКЛ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Розробка тест-кейсів

Основні завдання:

- Розробка та пріоритизація тестових випадків
- Створення тестових комплектів для ефективного виконання

Артефакти даного етапу:

- Test Cases

## Виконання тестування

Основні завдання:

- Виконання тест комплектів і окремих тестів
- Перевиконання попередньо неуспішних тестів
- Порівняння актуальний/очікуваний результат
- Звітування відмінностей як дефект
- Збереження результатів тест виконання

Артефакти даного етапу:

- Defect Reports



# Життєвий цикл тестування програмного забезпечення

## Оцінка результатів

Основні завдання:

- Перевірка чи результати задовольняють вихідні критерії
- Оцінка чи не потрібно додаткових тестових випадків
- Збір та аналіз щільності дефектів
- Написання фінального звіту для зацікавлених осіб

Артефакти даного етапу:

- Звіт результату, що включає статус тестування, метрики, аналіз та заключення



# Життєвий цикл тестування програмного забезпечення

## Завершення тестування

Основні завдання:

- Перевірка, чи все що мало бути зроблено, виконано
- Переконались, що всі дефект репорти проаналізовані
- Завершити та зберегти об'єкти що використовувались в тестуванні: тест скрипти, тестові дані та інше
- Передача тестового обладнання при здачі проекту
- Оцінка того як проходило тестування та засвоєння отриманих уроків для майбутніх проектів



# Верифікація та валідація в контексті життєвого циклу розробки програмного забезпечення

Верифікація виконується на кожному етапі **SDLC**.

**Аналіз вимог** → перевірка специфікацій (чи вимоги чіткі, повні, несуперечливі).

**Проектування і дизайн** → review діаграм, моделей, прототипів.

**Розробка** → code review, статичний аналіз, відповідність стандартам.

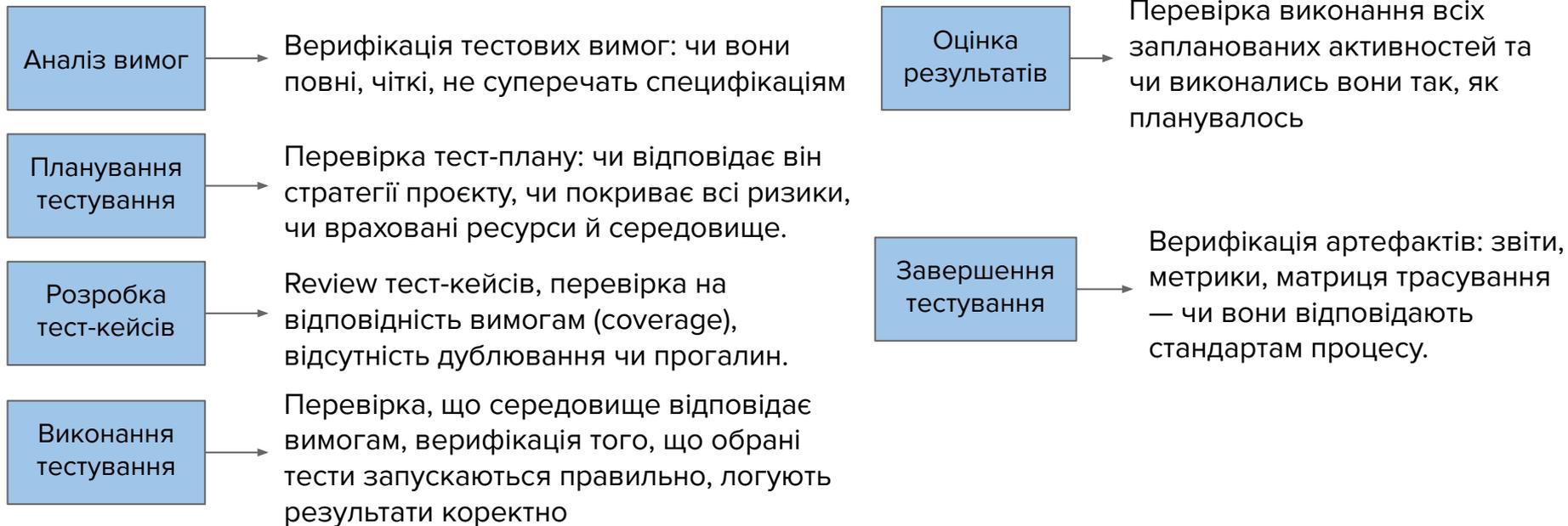
**Тестування** → перевірка, що тести покривають усі вимоги.

**Підтримка** → верифікація змін і патчів (чи не зламали іншу функціональність).

Верифікація забезпечує контроль відповідності артефактів їх специфікаціям на кожному кроці.

# Верифікація та валідація в контексті життєвого циклу розробки програмного забезпечення

Верифікація також виконується на кожному етапі **STLC**.



# Корисні ресурси

1. [https://istqb.org/wp-content/uploads/2024/11/ISTQB\\_CTFL\\_Syllabus\\_v4.0.1.pdf](https://istqb.org/wp-content/uploads/2024/11/ISTQB_CTFL_Syllabus_v4.0.1.pdf)
2. <https://glossary.istqb.org/>
3. <https://www.maxzosim.com/software-development-life-cycle-sdlc/>
4. <https://www.it-notes.wiki/software-testing/software-testing-life-cycle/>
5. <https://stfalcon.com/uk/blog/post/How-to-Write-a-Software-Requirements-Specification>
6. <https://training.gatetestlab.com/blog/technical-articles/popular-software-development-life-cycles/>
7. <https://www.parasoft.com/blog/verification-vs-validation-in-embedded-software/>

Дякую за увагу!