

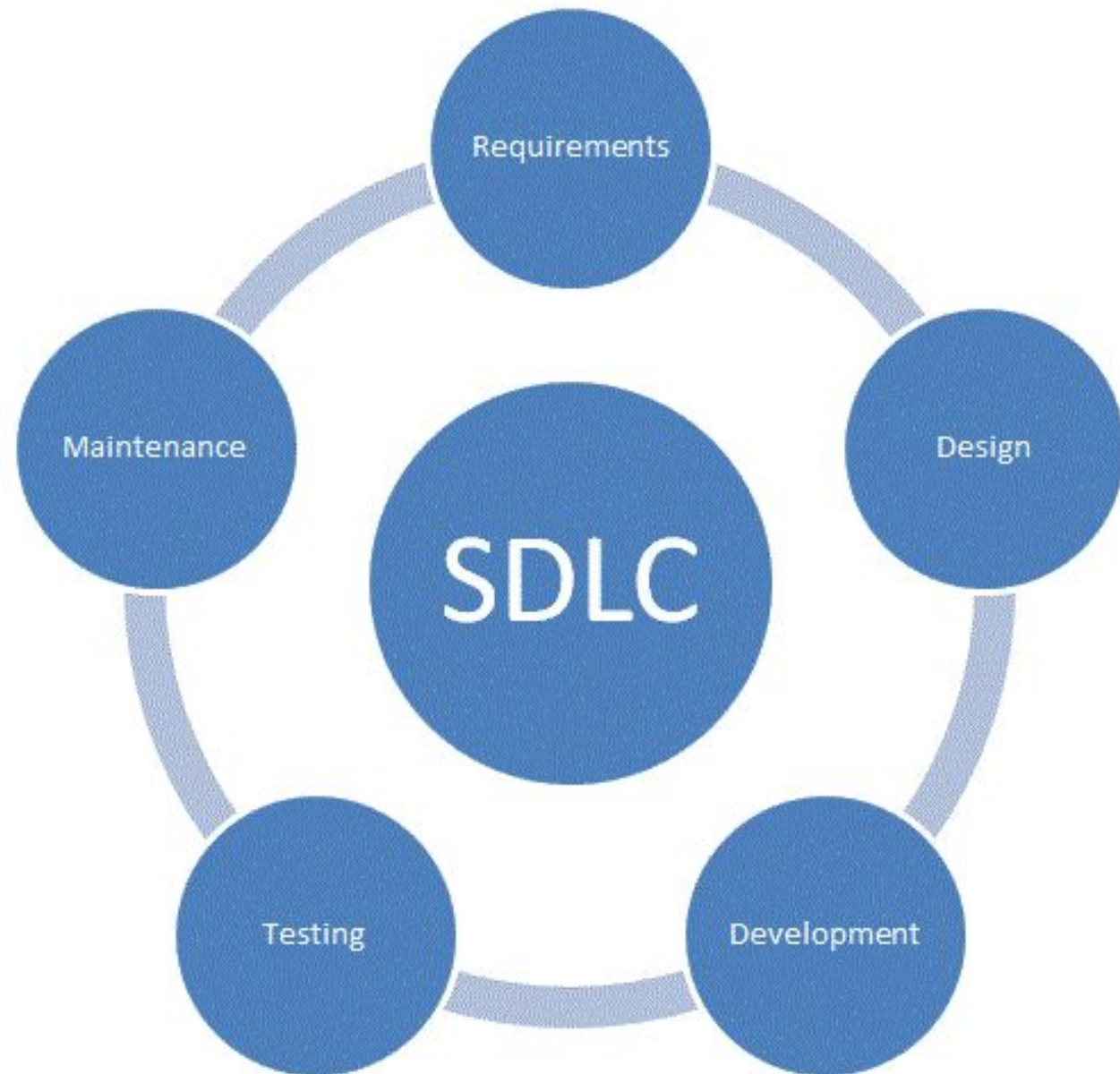
A person wearing a light blue button-down shirt is sitting at a dark wooden desk. They are holding a black pen and writing in an open, cream-colored notebook. To the right of the notebook, a silver laptop is partially visible. The background is a solid orange wall.

Тиждень

2

Тиждень 2. Життєвий цикл розробки ПЗ

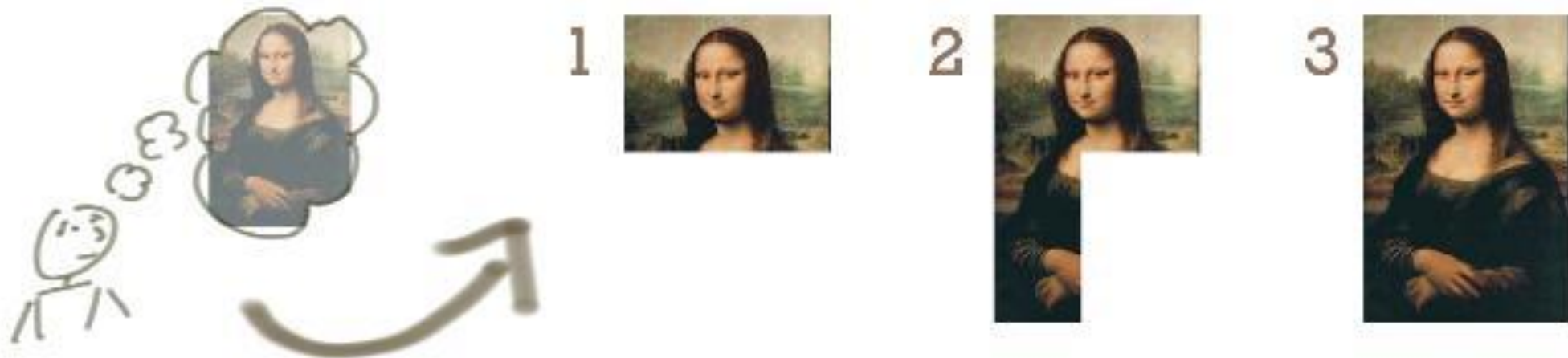
1. Збір та аналіз вимог
2. Дизайн
3. Розробка
4. Тестування
5. Підтримка



Тиждень 2. Моделі розробки ПЗ

Інкрементальна модель

процес розробки, при якому вимоги розділяються на кілька частин, кожна з яких доставляється з наступним випуском



Тиждень 2. Моделі розробки ПЗ

Ітеративна модель

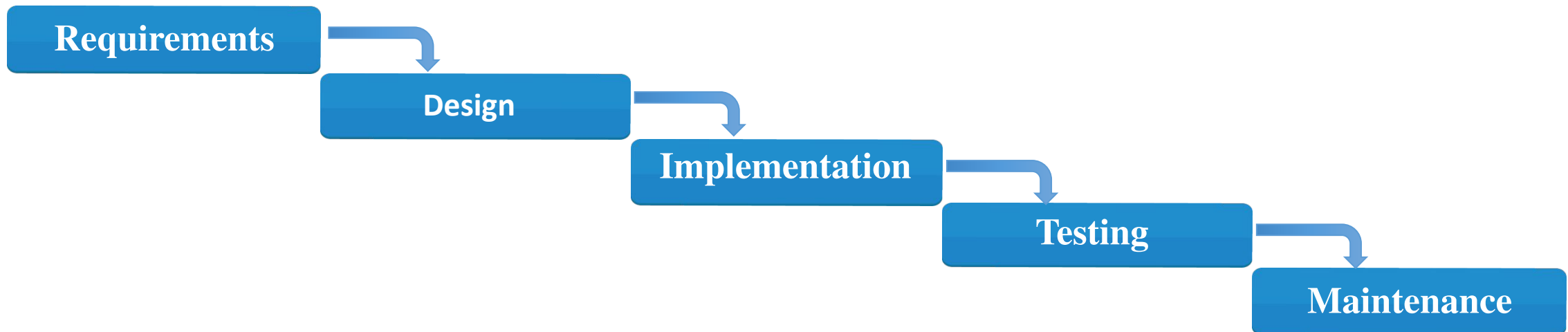
процес розробки, при якому немає чіткого бачення кінцевого продукту, натомість подальші вимоги узгоджуються в процесі розробки



Тиждень 2. Waterfall

Waterfall

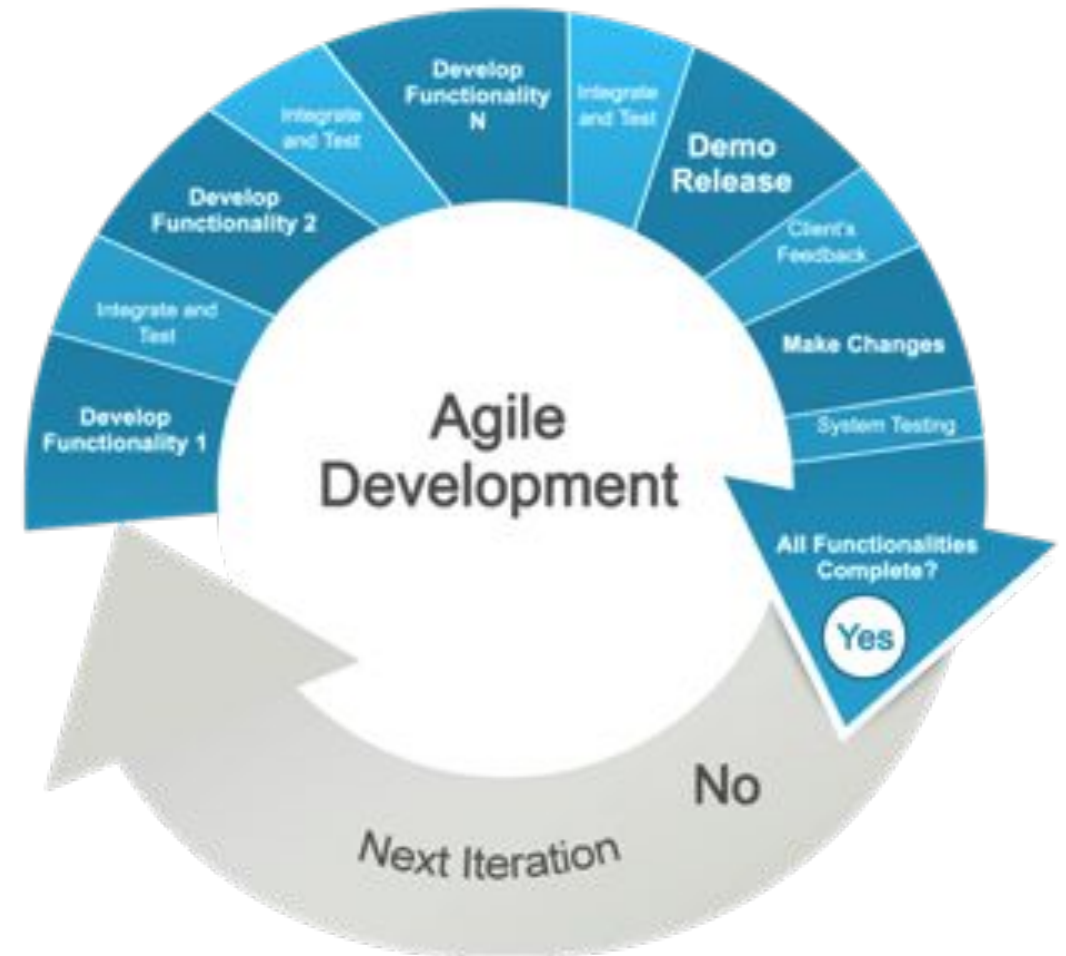
Методологія, при якій етапи розробки йдуть послідовно, кожна наступна починається по завершенні попередньої



Тиждень 2. Agile

Agile

Підхід в розробці програмного забезпечення, при якому використовується ітеративна модель, динамічне формування вимог, а також повна взаємодія в середині команди



Тиждень 2. Планування і контроль

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

Артефакт

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



Тиждень 2. Аналіз та дизайн

Основні

- **Завдання:**
 - Оцінка основи для тестування
 - Визначення тестових умов
 - Дизайн тестових випадків
 - Оцінка можливості тестування вимог та системи
 - Розробка тестових даних, налаштування та конфігурація тестового обладнання

Артефакт

- **И** Test Design Specification
- Peer Review Records



Тиждень 2. Впровадження та виконання

Основні завдання

- **впровадження:**
 - Вибір та пріоритизація тестових випадків
 - Створення тестових комплектів для ефективного виконання
 - Остаточне впровадження оточення

Основні завдання

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

Артефакт

- **И** Test Cases
- Defect Reports



Тиждень 2. Оцінка результатів

Основні завдання:

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

Артефакт

- **З**віт результату, що включає статус тестування, метрики, аналіз та заключення



Тиждень 2. Завершення тестування

Основні завдання:

- Перевірка, чи все що мало бути зроблено, виконано
- Переконатись, 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

