

Життєвий цикл розробки програмного забезпечення

SDLC

Порядок денний:

- Вступ
- SDLC Огляд
- SDLC Фази
- SDLC Потік
- SDLC Модель
- Модель водоспаду
- Гнучка модель
- Висновок

Вступ

Вступ

SDLC — це структура, яка описує дії, що виконуються на кожному етапі проекту розробки програмного забезпечення.

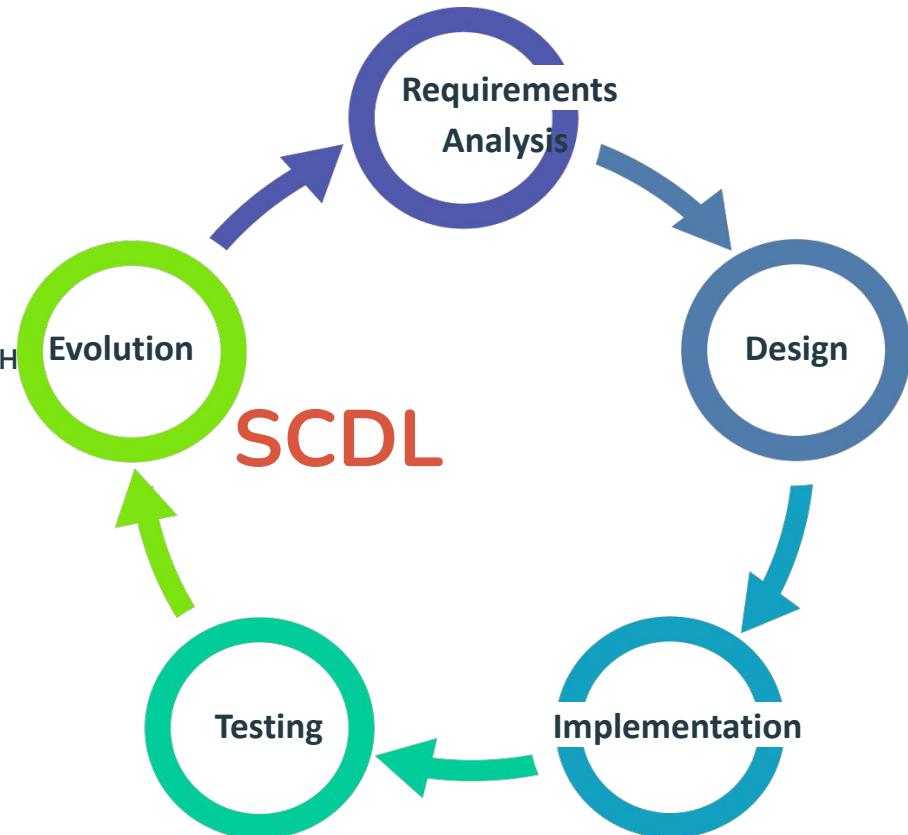
Процес SDLC використовується індустрією програмного забезпечення для проектування, розробки та тестування високоякісного програмного забезпечення. Вона спрямована на створення якісного програмного забезпечення, яке відповідає або перевершує очікування клієнтів, досягає завершення в межах часу та бюджету.

ISO/IEC 12207 — це міжнародний стандарт для процесів життєвого циклу програмного забезпечення. Він визначає всі необхідні завдання для розробки та супроводу програмного забезпечення, забезпечуючи єдиний підхід до управління життєвим циклом.

Software Engineering Process Technology Company (SEPT) — це фірма, яка спеціалізується на задоволенні потреб професійної спільноти в інформації про стандарти процесів програмного забезпечення, зокрема щодо ISO/IEC 12207.

SCDL фази

1. Планування та аналіз вимог
2. Визначення вимог
3. Проектування програмного забезпечення
4. Створення або розробка програмного забезпечення
5. Тестування програмного забезпечення
6. Розгортання та обслуговування



Планування та аналіз вимог

Аналіз вимог є найважливішим і фундаментальним етапом в SDLC.

Він виконується старшими членами команди за участю всіх зацікавлених сторін і експертів у галузі чи малих і середніх підприємств галузі.

Планування вимог щодо забезпечення якості та визначення ризиків, пов'язаних із проектом, також здійснюється на цьому етапі. SDLC від Манохара Прасада

Аналіз вимог

- Вимоги до бізнесу
- Вимоги до зацікавлених сторін
- Вимоги до рішення
 - Функціональні вимоги
 - Нефункціональні вимоги
- Вимоги до переходу

Визначення вимог

Після аналізу вимог наступним кроком є чітке визначення та документування вимог до програмного забезпечення та отримання їх схвалення від зацікавлених сторін проекту.

SRS (Software Requirement Specification) — це документ, який детально описує всі вимоги до програмного забезпечення, яке потрібно спроектувати та розробити протягом життєвого циклу проєкту. Це основа для узгодження між замовниками, розробниками та іншими зацікавленими сторонами.

Визначення вимог

- Аналіз підприємства
- Планування та моніторинг бізнес-аналізу
- Збір вимог
- Аналіз вимог
- Управління вимогами та комунікація
- Оцінка та перевірка рішення

Проектування програмного забезпечення

На основі вимог, зазначених у SRS, зазвичай пропонується більше ніж один підхід до проектування архітектури продукту та документується в DDS — специфікації проектної документації(Design Document Specification).

Цю DDS перевіряють усі зацікавлені сторони, і на основі таких параметрів, як оцінка ризику, модульність дизайну, бюджет і часові обмеження, вибирається найкращий підхід до розробки програмного забезпечення.

Розробка програмного забезпечення

На цьому етапі SDLC починається фактична розробка та створення продукту. На цьому етапі генерується програмний код відповідно до DDS.

Розробники мають дотримуватися вказівок щодо кодування, визначених їхньою організацією, а інструменти програмування, такі як компілятори, інтерпретатори, налагоджувачі тощо, використовуються для створення та впровадження коду.

Тестування програмного забезпечення

Цей етап зазвичай є підмножиною всіх етапів, оскільки в сучасних моделях SDLC діяльність з тестування здебільшого задіяна на всіх етапах SDLC.

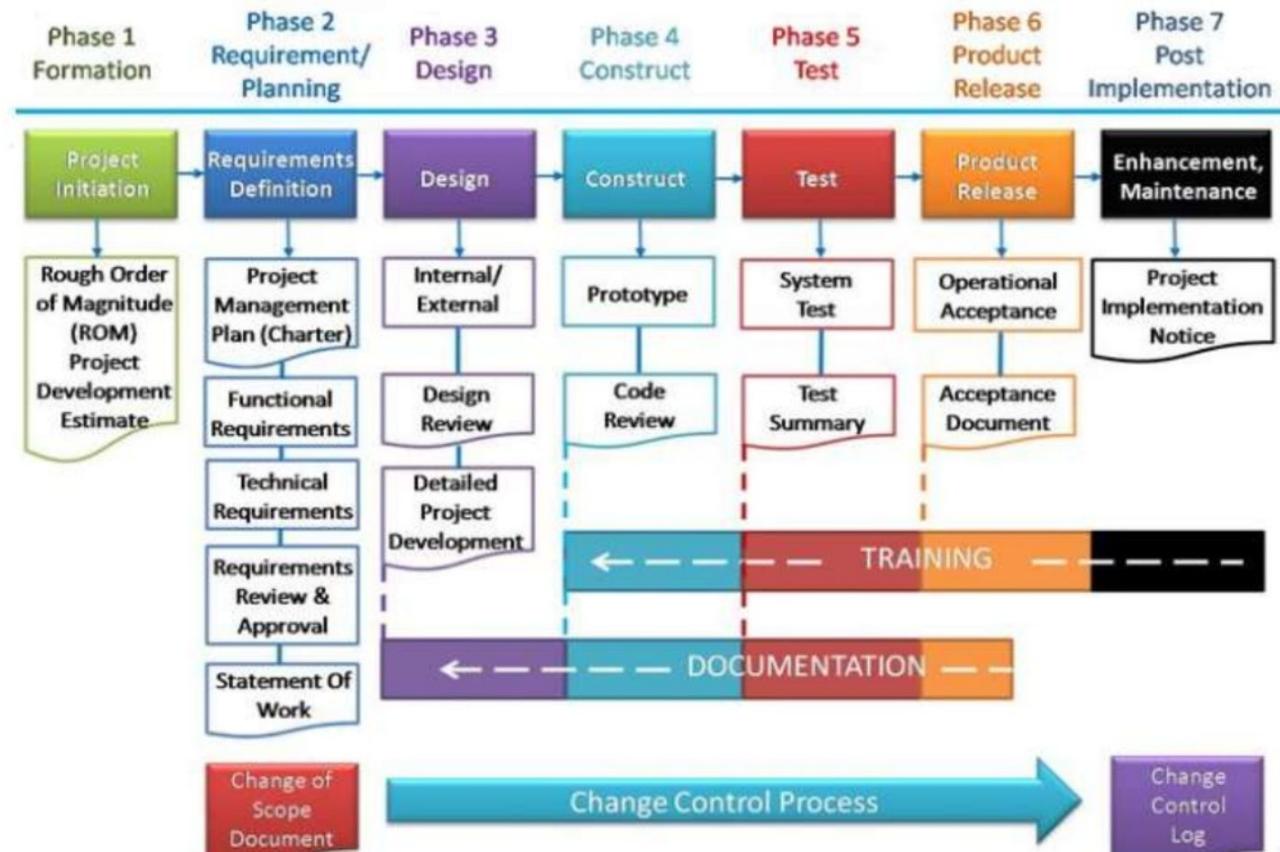
Однак цей етап стосується **тестування**, яке охоплює лише ту частину програмного забезпечення, де виявляються дефекти, ведеться їх реєстрація, виправлення та повторне тестування, доки програмне забезпечення не досягне визначених у **SRS** (**Software Requirement Specification**) стандартів якості.

Розгортання та обслуговування

Як тільки програмне забезпечення проходить тестування і не залишається звітів про помилки чи дефекти, його впроваджують у робоче середовище.

Потім на основі відгуків програмне забезпечення може бути випущено як є або із запропонованими вдосконаленнями в цільовому сегменті.

Після розгортання програмного забезпечення починається його обслуговування.



SDLC Моделі

Для полегшення розуміння та впровадження фаз **життєвого циклу розробки програмного забезпечення (SDLC)** було створено різні моделі SDLC провідними експертами в галузі розробки програмного забезпечення, університетами та організаціями зі стандартизації.

Причини використання моделей SDLC

- Забезпечує основу для планування проекту, оцінки та планування.
- Забезпечує основу для стандартного набору термінології, заходів і результатів.
- Забезпечує механізм відстеження та контролю проекту.
- Збільшує видимість прогресу проекту для всіх зацікавлених сторін.

Переваги правильного вибору SDLC

- Підвищена швидкість розвитку
- Підвищена якість продукції
- Покращене відстеження та контроль
- Поліпшення відносин з клієнтами
- Зниження ризику проекту
- Зменшення накладних витрат на управління проектом

SDLC Моделі

- *Модель водоспаду*
- Ітеративна модель
- Спіральна модель
- *Гнучка модель*
- V – Модель
- Модель великого вибуху

Висновок

- Проектне управління веде до успішного завершення завдання.
- **Планування** — це своєрідна карта або керівництво, особливо важливe для команди. (Відносно прості та корисні техніки)
- SDLC здебільшого стосується людей, процесів, управління часом і спілкування.
- **Ризики неминучі**, але планування допомагає уникнути безглуздих помилок (Важливий досвід).
- Оцінка обсягу робіт, термінів, ризиків і ресурсів призведе до успіху проекту.

Питання та відповіді