

1. 소프트웨어 구축

1-1. 소프트웨어 공학

1. 소프트웨어 공학 정의

- 소프트웨어 위기를 극복하고 품질 높은 소프트웨어를 효율적으로 개발하기 위한 학문
- 소프트웨어 개발에 적용되는 방법, 도구, 이론을 포괄적으로 다룬다.

2. 소프트웨어 공학의 3R

- 완성된 소프트웨어를 기반으로 하여 역공학, 재공학, 재사용을 통해 소프트웨어의 생산성을 극대화하는 기법

(1) 역공학(Reverse Engineering) :

- 이미 개발된 시스템을 분석하여 요구 분석서, 설계서 등의 문서를 추출하는 작업
- 개발 단계를 역순으로 거슬러 올라가면서 기존의 코드나 데이터로부터 설계 명세서나 요구 분석서 등을 도출

(2) 재공학(Re-Engineering) :

- 유지보수의 생산성을 통해 소프트웨어의 위기를 해결하기 위한 한 방법
- 기존 소프트웨어를 폐기하지 않고, 기능을 개선하거나 새로운 소프트웨어로 재활용하는 공법

재공학 과정

과정	설명
분석 (Analysis)	- 기존 소프트웨어의 명세서를 검토하여 재공학 대상을 선정
재구성 (Restructuring)	- 소프트웨어의 구조를 개선하기 위해 코드를 재구성 - 소프트웨어의 외적인 기능은 변경되지 않음

과정	설명
역공학 (Reverse Engineering)	- 소프트웨어의 소스코드를 분석하여 설계 수준을 도출
이관 (Migration)	- 기존 소프트웨어를 다른 운영체제, 프레임워크 등에서 사용할 수 있도록 변환

(3) 재사용(Reuse) :

- 이미 개발된 소프트웨어의 전체 또는 일부를 다시 사용하는 것을 의미한다.
- 재사용의 범위

함수와 객체 재사용	클래스나 함수 단위로 구현된 소스코드의 재사용
컴포넌트 재사용	독립적인 소프트웨어 컴포넌트의 재사용
애플리케이션 재사용	기존 애플리케이션 또는 그 일부를 새로운 소프트웨어 개발에 재사용

- 재사용 방법

합성중심 (Composition Based, 블록구성)	모듈을 조합하여 소프트웨어를 완성하는 방법
생성중심 (Generation Based, 패턴구성)	추상화된 형태의 명세를 구체화하여 프로그램을 만드는 방법

3. 소프트웨어 개발 단계

계획(Planning) → 요구사항분석(Requirements Analysis) → 설계(Design) → 구현(Development) → 테스트(Test) → 유지보수(Maintenance)

1-2소프트웨어 개발 방법론

소프트웨어 개발 방법론 종류

(1) 구조적 방법론

- 절차지향적인** 소프트웨어 개발 방법론(하향식)
- 제한된 구조 안에서 코드를 생성하고 순차적으로 실행하는 방식

