

# < 1라목 >

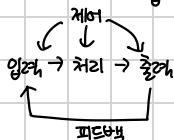
## 01. 소프트웨어 공학

### ① 소프트웨어 특징

- 상품성, 복잡성, 변경가능성(필드), 복제성

### ② 시스템의 개요와 기본요소

② 입력, 출력, 제어, 피드백



### ③ 소프트웨어 위기

- 개발비용 ↑, 기간 ↑, 인력 부족, 성능 및 신뢰성 부족, 유지보수의 어려움

### ④ 소프트웨어 공학 : 신뢰성 ↑, 유지보수성 ↑, 지속적인 성장을 하기 위한 절차

## 02. 재공학

### ① 재공학

- 기존에 만든 것을 다시사용 ⇒ 시간 ↓, 비용 ↓, 품질 ↑, 생산성 ↑, 신뢰성 ↑, 프로젝트 위험 ↓ (현장 검증됨)

유지보수성 향상이 최우선

- 과정 : 분석 → 구성 → 역공학 → 이식 (적용할)  
(구조 변경) (기능 및 알고리즘 기존 분석, 재공학)

### ② CASE : 소프트웨어 공학을 도울

⇒ 개발 신속, 정확도 향상, 품질 ↑, 생명주기 전체 관계 연결, 고도화, 기능 제공, 민중성, 흐름도 제공

⇒ 장점 : 기간 ↓, 비용 ↓, 생산성 ↑

⇒ 종류 : 상위 : 추상화, 설계

하위 : 구현

통합 : 소프트웨어 개발주기 전체 과정 지원

⇒ 도구 : SADT ⇒ Software에서 개발함, 블록 다이어그램을 채택함

## 03. 소프트웨어 개발 방법론

① 생명주기 : 타당성 검토 → 개발계획 → 요구사항 분석 → 설계 → 구현 → 테스트 → 운용 → 유지보수

### ② 연립

- 폭발 : 각 단계가 순차적 실행, 소규모, 이벤트가 발생하면 변형 X

- 나선형 : 계획 → 구현 (이벤트 발생하면) → 고도화 → 개발 및 검증

- 하향식 설계 : 중요한 기능 먼저 리듬 보이기 가능

- 상향식 설계 : 기본적인 기능을 통합함

- 프로토타입 : 고객과 의사소통에 좋음

- HIPO : 계층적 입력-처리-출력, 시스템도, 출력 다이어그램, 세부적 다이어그램 → 이식성, 가시성

· V모델 : HIPPO 테스트를 최우선

· 예시일 개발 방법론 : 정확하게 목표한 것 위함

(예) 종류 : XP, 스크럼, RUP, FDD

이점들

⑦ XP : 빠르게 양질의 소프트웨어를 만들, 소통, 피드백, 용이, 존중, 간소성

\* XP의 12가지 실천사항

1. Pair Programming : 두 개발 - 두 컴퓨터
2. Planning game : 목표를 수립함
3. Test Driven Development : 단위테스트
4. Whole team : 협업
5. Continuous Integration : 배포할수 있는 상태 유지
6. Design Improvement : 재구성
7. Small release : 짧은 주
8. Coding standard : 코딩 규칙
9. Collective code ownership : <sup>공동</sup>소스 코드 수정 가능
10. Simple design : 간결하게 재현
11. System Metaphor : 시스템 구조
12. Sustainable pace : 과도한 일하지

⑧ 프론트엔드 개발 요소 : UI, 로직, 프레임워크 (개)

04. SCRUM

① 역할

1. 제품 책임자: 이해관계 개발 의뢰자, 담당자. 운영 간섭X
2. 스프린트 매스터: 개발 프로세스 방해 요소 제거
3. 스토리점: 디자이너, 제품 관리자, Client 등의 대표

② 진행

1. product backlog : 작업 목록
2. sprint : 2~4주 진행

05. 현대형 시스템 빌딩

(\*) 회사가 어떻게 구성되어 있는지

- ① 파악할라 : 1단계 시스템 구성 파악 (부피 파악) → 기능 파악 → 시스템 인터페이스 영향 파악  
2단계 이해관계자 파악 → SW 구성 파악  
3단계 하드웨어 파악 → 네트워크 구성 파악

운영  
시스템 (SW)  
HW

→ 상용시스템, 하용시스템들이  
이러한 관계로 상호 작용하는지

② EAT : 기업 내의 컴퓨터 애플리케이션들을 현대화 하는 동안에,  
조정되는 것을 목표로 하는 계획, 단계를 일컫는 것?

③ 플랫폼 : 응용소프트웨어 + (하드웨어 + 시스템소프트웨어)

↳ 소프트웨어를 만들기 위한 기반

1. 특성분석항목 : 응답시간, 가용성, 사용률

2. 특성분석 방법 : 기능테스트, 인터뷰, 문서검정

④ OS 분석 : 종류, 버전, 비용(TCO), 가용성, 성능, 지원(하드웨어/소프트웨어)

⑤ 인프라인 : 공개하트

↳ Apache 2.0 ⇒ HADOOP 라이선스

⑥ DBMS : 데이터관리, 종속성, 중복성 해결

↳ 가용성, 성능, 기술지원, 상인화성, 구축비용

## 06. 요구사항 개발

: 원할한 의사소통 제공, 누락방지, 상충해제, 용이성

① SWE Bok : 요구사항 추출 → 분석 → 명세 → 검증

② 도출방법 : 고객의 발표, 문제지, 설문, 인터뷰, 워크숍, Use Case

③ 요구사항을 위한 개발 : 사용자의 견해, 사용자 인터페이스, 설문조사

↳ 문제인식 → 전개 → 검토 → 문제해결

④ 기능재 vs 비기능재

↳ 동태 vs 보안, 품질, 성능

⑤ 명세속성 : 정확성, 명확성, 일관성, 일관성, 수정용이성

## 07. UML

↳ 시스템분석

① 컴퓨터 객체지향 분석 방법

객체모델링 : 객체 다이어그램  
동작 // : 상호작용, 용량성, 상태  
기능 // : 자료 흐름

② Use Case 작성방법

분석식별 → Use Case 식별 → 관계명치 → Use Case 구현

③ 특징 : 비구조적, 문이나, 추상

④ public +

private -

protected #

package ~

⑤ → 상속, 일반화

→ 일반화 — 일반화