

Chapter 01

소프트웨어 공학 소개



쉽게 배우는
소프트웨어 공학

01 소프트웨어의 이해

02 공학과 소프트웨어 공학의 이해

03 소프트웨어 개발 단계의 소개

요약

연습문제

- 소프트웨어의 특징을 살펴본다.
- 소프트웨어 공학의 뜻을 이해한다.
- 소프트웨어 개발 단계를 알아본다



Section 01 소프트웨어의 이해

1. 소프트웨어가 사용되는 곳

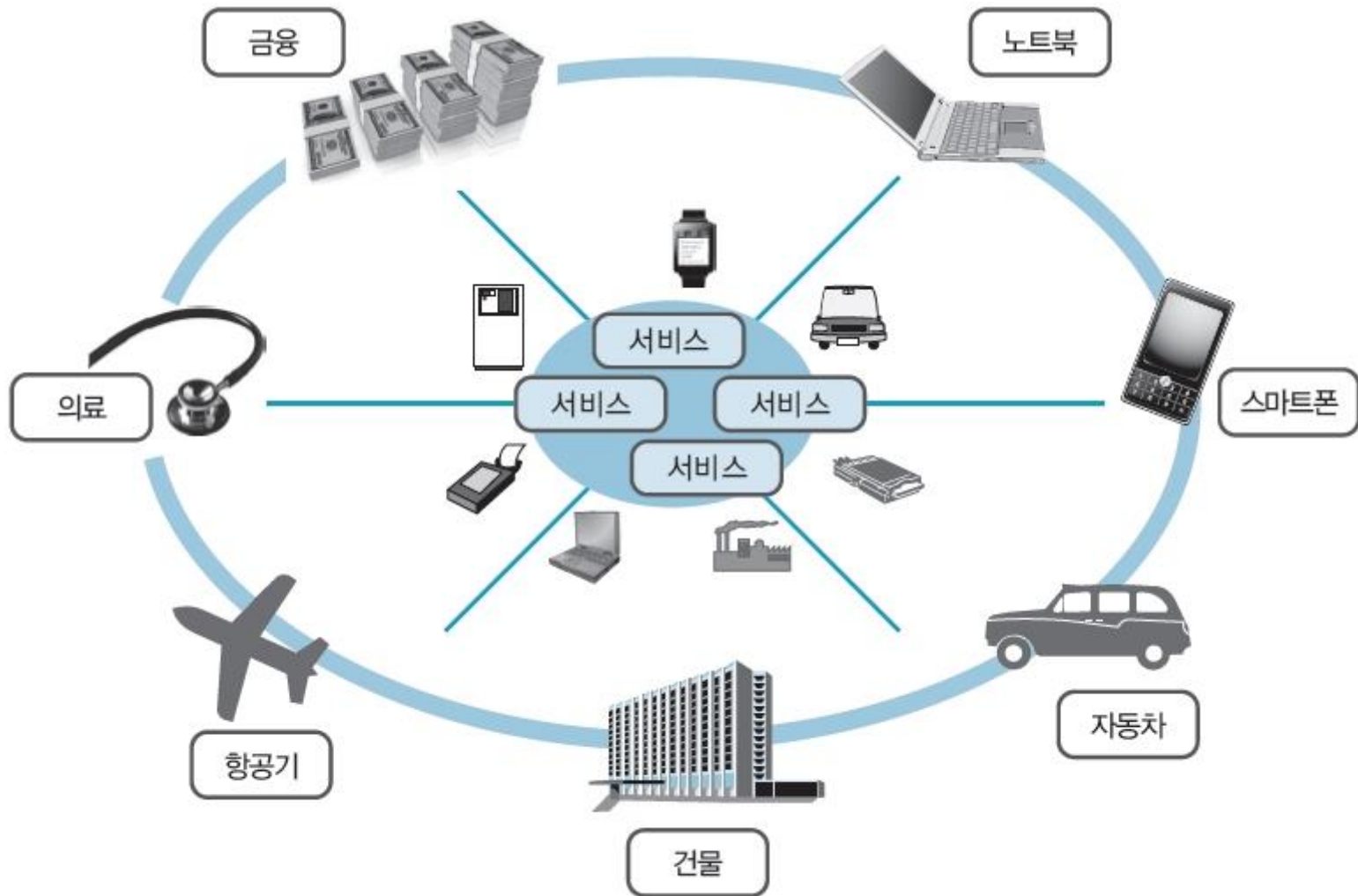


그림 1-1 소프트웨어가 사용되는 곳

2. 프로그램과 소프트웨어

■ 프로그램

- 원시코드 source code

■ 소프트웨어

- 원시코드 source code
- 모든 산출물(자료구조, DB구조, 테스트 결과 등)
- 각 단계마다 생산되는 문서
- 사용자 매뉴얼

→ 프로그램 뿐만 아니라 그 이상의 것도 포함하는 매우 포괄적인 개념

4. 소프트웨어의 특징

■ 제조가 아닌 개발

- 제조: 정해진 틀에 맞춰 일정하게 생산하는 것으로, 많은 인력이 필요하고 능력별 결과물 차이가 근소함
- 개발: 개인 능력 별 결과물 차이가 매우 큼

■ 소모가 아닌 품질 저하

- H/W : 오래 사용하면 부품이 닳고, 고장 발생 빈도 높고, 기능도 떨어짐
- S/W : 오래 사용해도 닳지 않고, 고장 발생 빈도 낮고, 기능도 동일 함

5. H/W 실패 곡선(욕조 곡선)의 특징

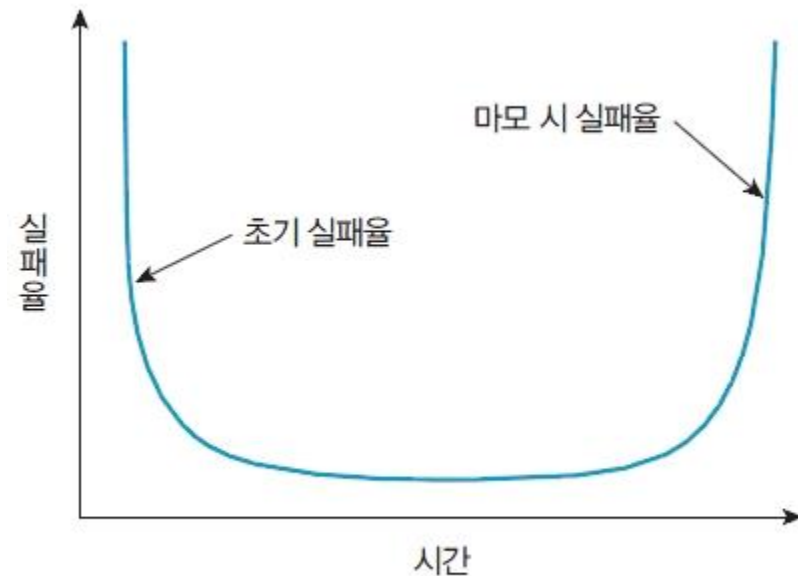
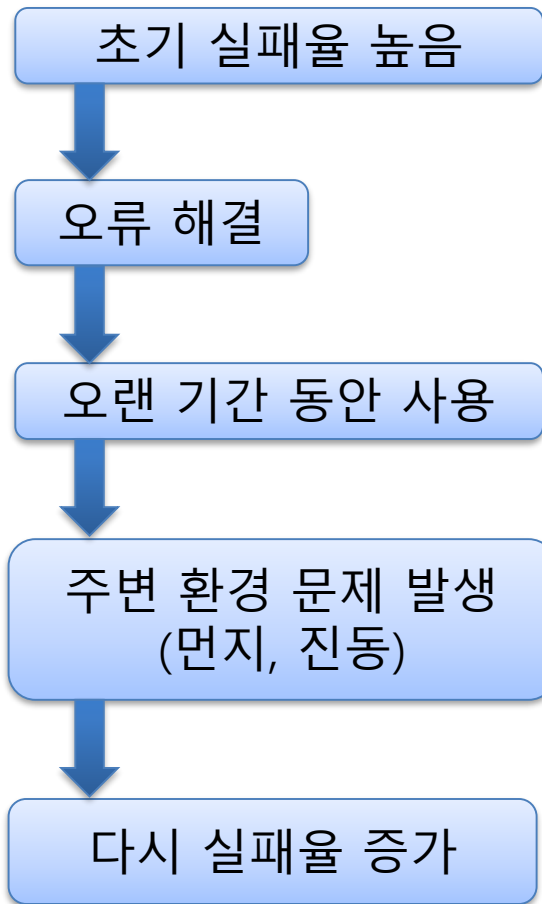
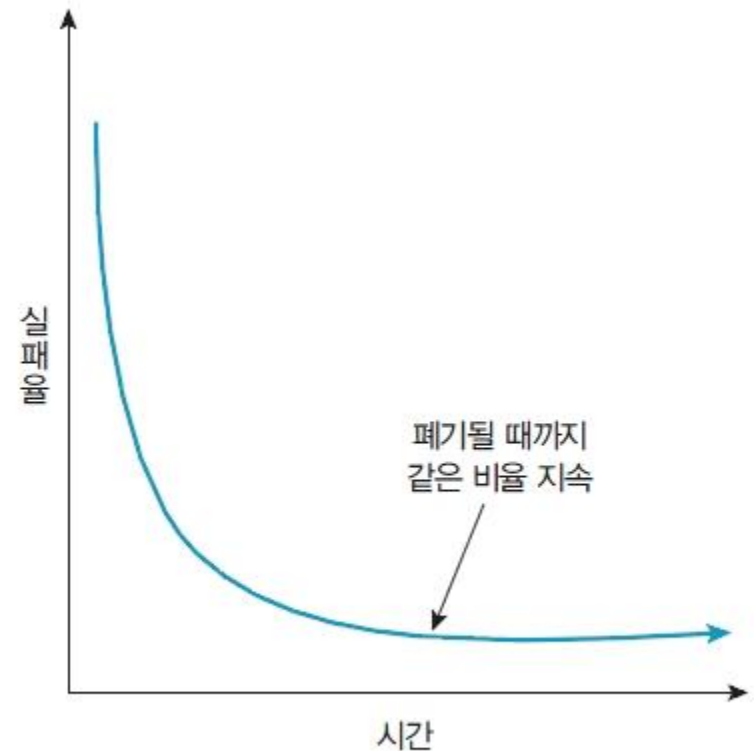
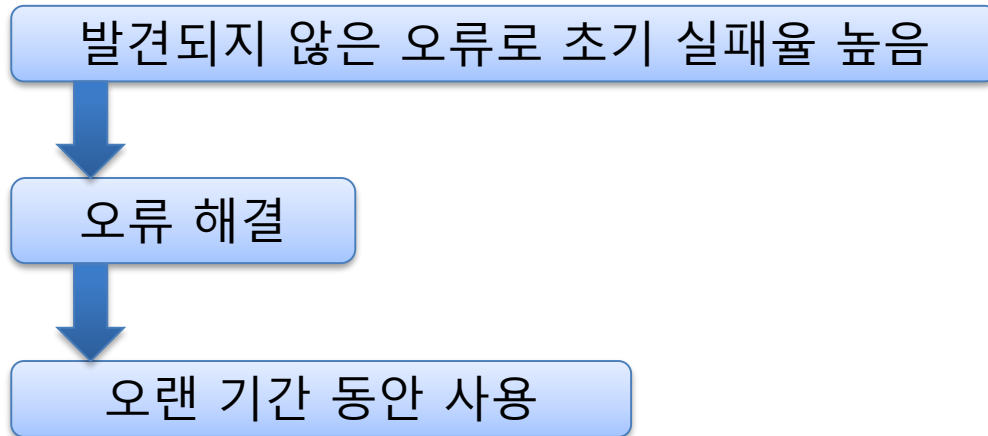


그림 1-2 하드웨어의 실패 곡선

6. 이상적인 소프트웨어 실패 곡선

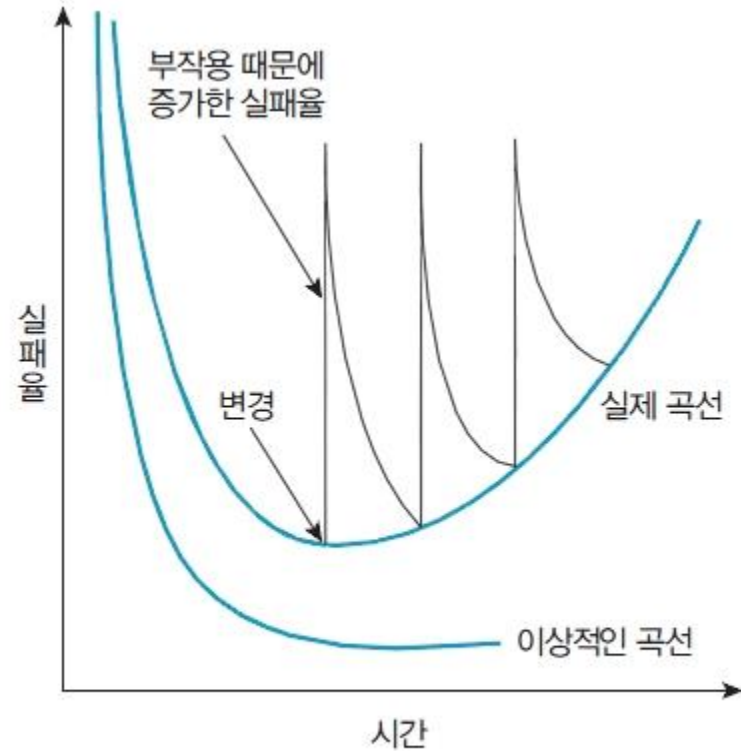
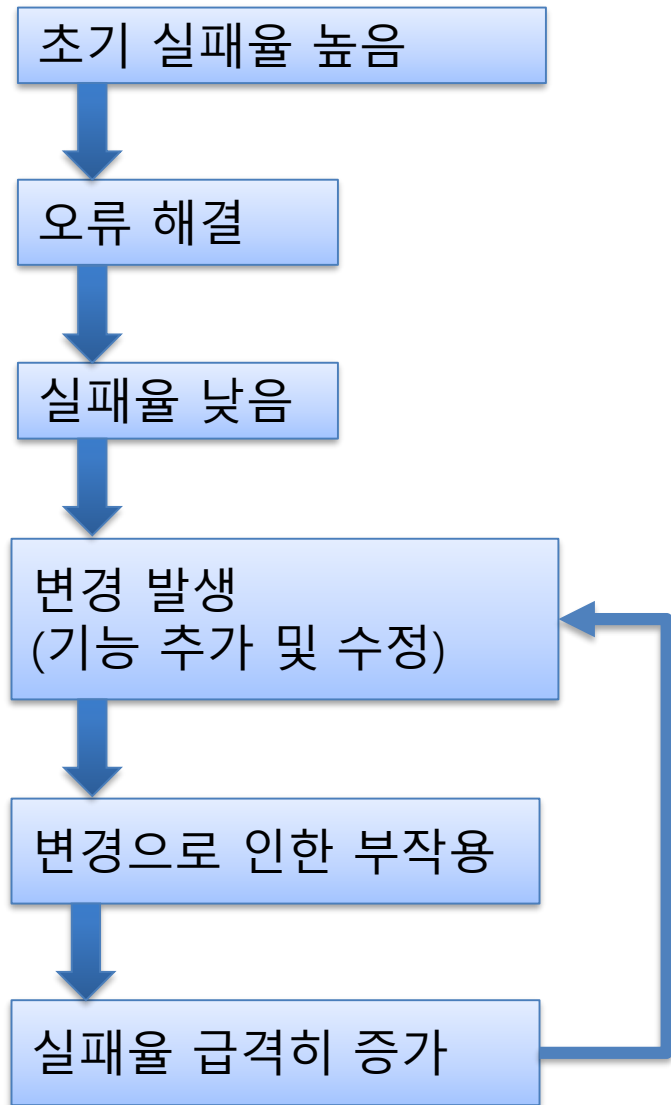
■ 특징 : 이상적인 상황

- 개발 완료 후 변경 사항 없어야 함
- 개발 완료 후 환경 변화 없어야 함



(a) 이상적인 소프트웨어의 실패 곡선

7. 실제 소프트웨어 실패 곡선



(b) 실제 소프트웨어의 실패 곡선

8. 소프트웨어의 당면 과제(1)

■ 소프트웨어 개발의 느린 발전 속도

- H/W의 발전: PC 및 스마트폰의 발전 속도(크기, 속도, 성능)
- S/W 발전 속도: DOS ~ Windows 10

■ 새로운 소프트웨어에 대한 사용자 요구의 증가

- S/W의 발전 속도가 미처 따라가지 못함
- H/W와 S/W의 개발 방법의 근본적인 차이 때문
 - H/W: 검증 받은 부품을 조립하는 형태의 생산
 - S/W: 처음부터 만들어가는 개발 형태
- (해결 방안) CBD개발 방법론

8. 소프트웨어의 당면 과제(2)

■ 관리 기술의 부분적 활용

- 기계: 닦고, 조이고, 기름치고 => 수명 연장
 - S/W 개발에도 관리가 필요
 - 비용 관리
 - 일정 관리
 - 개발자 관리
- 도구를 활용한 적극적인 프로젝트 관리 필요

9. 소프트웨어 개발의 어려움(1)

■ 개집 짓기

- 필요 도구: 망치, 톱, 줄자 등
- 설계 도면 필요 없음, 머릿속 구상만으로도 충분
- 혼자 가능, 만드는 과정 단순



그림 1-4 개집 짓기

9. 소프트웨어 개발의 어려움(2)

■ 단독주택 짓기

- 필요 도구: 레미콘과 같은 장비, 시멘트 등의 수 많은 자재
- 설계 도면, 건축 설계사 필요
- 많은 사람 참여, 만드는 공정 과정 필요



그림 1-5 단독주택 짓기

9. 소프트웨어 개발의 어려움(3)

■ 대형 빌딩 짓기

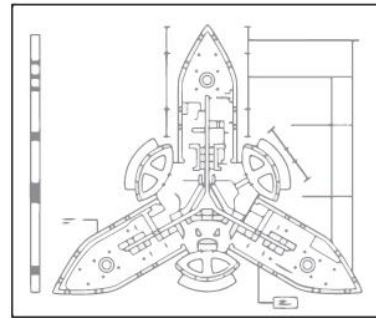
- 필요 도구: 레미콘뿐만 아니라 크레인과 같은 대형 장비
- 설계 도면, 건축 설계사뿐만 아니라 내진 설계 필요
- 많은 사람이 참여할 뿐만 아니라 통제와 조정할 수 있는 조직(부서)이 필요
- 하중 문제 등 고려 사항이 많음



그림 1-6 대형 빌딩 짓기



① 기획 단계



② 설계 단계



③ 시공 단계

그림 1-47 부르즈 할리파와 건축 과정

9. 소프트웨어 개발의 어려움(4)

개발 과정이 복잡하다

무엇이든지 복잡하면 문제가 많이 발생할 수 있는데 소프트웨어 개발도 예외가 아니다. 그래서 소프트웨어 공학에서는 개발의 복잡함을 줄이기 위한 방법과 기술을 제시한다.

참여 인력이 많다

인력이 많으면 의사소통 경로가 많아져 의사 결정 과정도 복잡할 것이다. 또한 협력도 쉽지 않다. 그리고 중간에 이직하는 사람, 새로 투입되는 사람 등 변화도 많이 발생한다. 그래서 소프트웨어 공학에서는 개발에 참여하는 팀을 구성하고 관리하는 효율적인 방법을 제시한다.

개발 기간이 길다

개발 기간이 길면 프로젝트 진행 상황을 파악하기 쉽지 않고 개발 비용 산정도 어렵다. 그래서 소프트웨어 공학에서는 프로젝트를 효율적으로 관리하기 위한 프로젝트관리지식체계^{PMBOK}를 소개한다.

그림 1-7 대규모 소프트웨어 개발의 어려움과 소프트웨어 공학



Section 02

공학과 소프트웨어 공학의 이해

1. 공학

■ 공학의 사용 예

- 전기공학과, 건축공학과, 토목공학과 등의 대학교에서 학과 명으로 사용

■ 공학의 특성

- 제약 사항: 정해진 기간, 주어진 비용
→ 과학적 지식을 활용하여 문제를 해결하는데 한정된 기간과 비용의 제약을 받음

■ 소프트웨어 공학

- 소프트웨어 + 공학
- 취지: '소프트웨어 개발 과정에 공학적인 원리를 적용하여 소프트웨어를 개발'
- 목적:
 - S/W 개발의 어려움 해결
 - 효율적 개발을 통한 생산성 향상
 - 고품질 소프트웨어 제품

2. 소프트웨어 개발 과정

■ 소프트웨어 개발 생명주기(SDLC Software Development Life Cycle)

- 계획 단계에서 유지보수 단계에 이르기까지 일어나는 일련의 과정

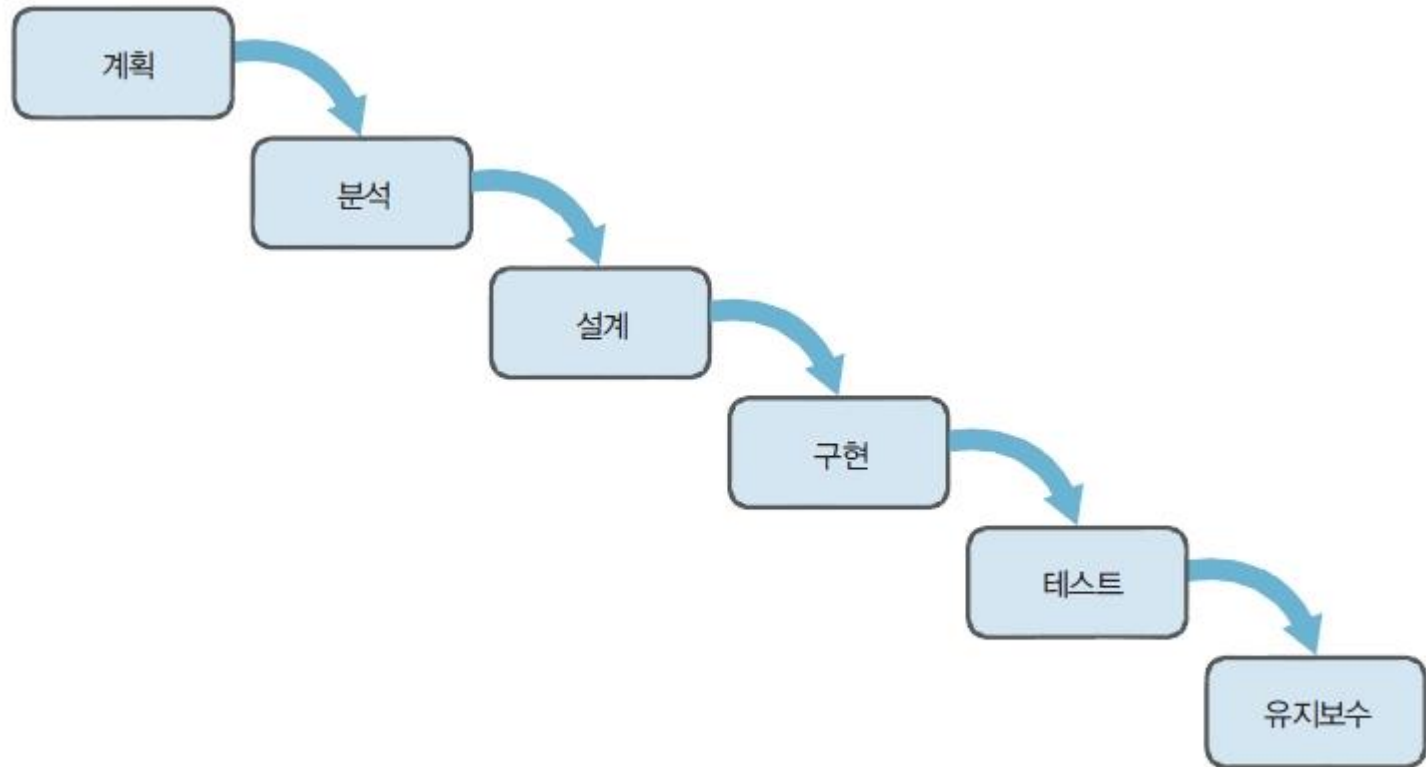


그림 1-8 소프트웨어 개발 생명주기 SDLC: Software Development Life Cycle

3. 소프트웨어 공학

■ 정의

품질 좋은 소프트웨어를 경제적으로 개발하기 위해
계획을 세우고, 개발하며,
유지 및 관리하는 전 과정에서
공학, 과학 및 수학적 원리와 방법을 적용하여
필요한 이론과 기술 및 도구들에 관해
연구하는 학문

■ 목표

- 개발 과정에서의 생산성 향상
- 고품질의 소프트웨어 생산 → 사용자 만족



Section 03

소프트웨어 개발 단계의 소개

1. 소프트웨어 개발 단계

■ 소프트웨어 개발 프로세스

- 1단계 : 계획
- 2단계 : 요구분석
- 3단계 : 설계
- 4단계 : 구현
- 5단계 : 테스트
- 6단계 : 유지보수

■ 소프트웨어 개발 프로세스

■ 품질 관리

■ 프로젝트 관리

2. 계획/요구분석 단계

■ 1단계 : 계획(3장에서 구체적으로 설명)

- 개발 비용 산정: COCOMO모델, 기능점수(FP)모델 사용
- 일정 계획: 작업분할구조도^{WBS}, CPM 사용
- 위험 관리

■ 2단계 : 요구분석 (4장에서 구체적으로 설명)

- 기존 시스템의 문제점 파악 → 새로운 요구사항 도출 → 다이어그램 작성
- 개발 방법론에 따른 표현 도구
 - 구조적 방법론: DFD, DD, Mini Spec
 - 정보공학 방법론: E-R 다이어그램
 - 객체지향 방법론: UML의 유스케이스 다이어그램
- 최종 산출물: 요구 분석 명세서

3. 설계/구현 단계

■ 3단계 : 설계 (5-6장에서 구체적으로 설명)

- 설계 원리: 분할과 정복, 추상화, 단계적 분해, 모듈화, 정보은닉
- 소프트웨어 아키텍처, 객체지향 설계
- 아키텍처 스타일
- GoF의 디자인 패턴
- 모듈 평가 기준: 응집도와 결합도

■ 4단계 : 구현 (7장에서 구체적으로 설명)

- 간략한 프로그래밍 언어의 역사
- 표준 코딩 규칙

4. 테스트/유지보수 단계

■ 5단계 : 테스트 (8장에서 구체적으로 설명)

- 테스트의 절차
- 개발자 또는 사용자 시각에 따른 분류
- 사용되는 목적에 따른 분류
- 품질 특성에 따른 분류
- 소프트웨어 개발 단계에 따른 분류

■ 6단계 : 유지보수 (10장에서 구체적으로 설명)

- 수정 유지보수
- 적응 유지보수
- 기능보강 유지보수
- 예방 유지보수

5. 소프트웨어 개발 프로세스

■ 소프트웨어 개발 프로세스(2장에서 구체적으로 설명)

- 주먹구구식 개발 모델
- 선형순차적 모델(폭포수 모델)
- V 모델
- 진화적 프로세스 모델(프로토타입 모델)
- 나선형 모델
- 단계적 개발 모델
- 통합 프로세스 모델(UP)
- 애자일 프로세스 모델

6. 품질 관리

■ 품질 관리(8장에서 구체적으로 설명)

■ 제품 품질 특성 평가

- ISO/IEC 9126 모델
- ISO/IEC 9126 모델
- ISO/IEC 9126 모델
- ISO/IEC 9126 모델

■ 프로세스 품질 특성 평가

- ISO/IEC 9000 모델
- ISO/IEC 12207 모델
- CMMI 모델
- SPICE 모델

7. 프로젝트 관리

- 프로젝트 관리 (10장에서 구체적으로 설명)

- 형상 관리

- PMBOK프로젝트관리지식체계의 9가지 관점

- ① 프로젝트 통합 관리
- ② 프로젝트 범위 관리
- ③ 프로젝트 일정 관리
- ④ 프로젝트 비용 관리
- ⑤ 프로젝트 품질 관리
- ⑥ 프로젝트 인적자원 관리
- ⑦ 프로젝트 의사소통 관리
- ⑧ 프로젝트 위험 관리
- ⑨ 프로젝트 조달 관리



Thank You
