

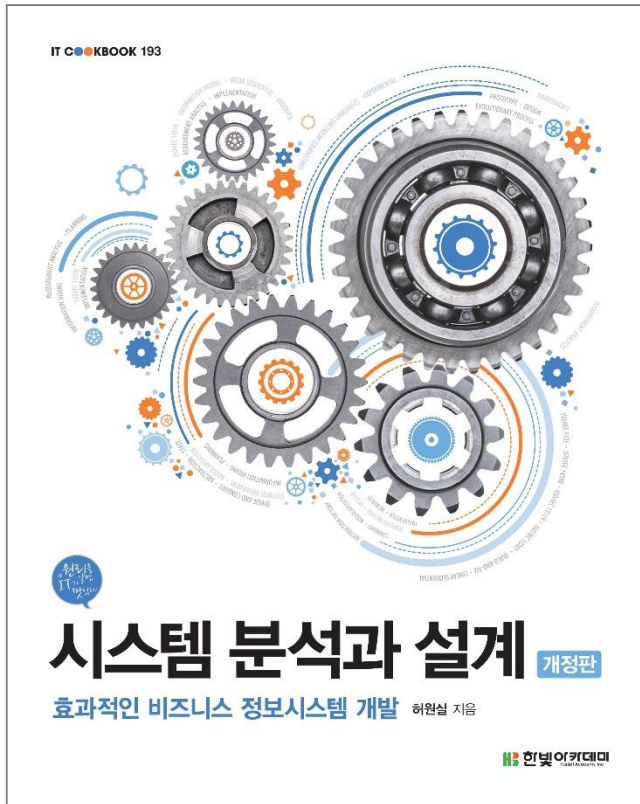


시스템 분석 설계

김 지 혜

010-8745-8484

wisdom8093@naver.com



- 도서명 : 시스템 분석과 설계 개정판
- ISBN : 979-11-5664-206-0 93000
- 저자 : 허원실
- 출판사 : 한빛아카데미(주)
- 페이지 / 정가 : 408p / 25,000원

1부 시스템 분석/설계 개요

- 시스템 개발 과정의 이해
- 시스템 분석/설계 방법론과 관련 문서

2부 구조적 분석 방법론

- 구조적 분석 방법론의 이해
- 자료흐름도
- 자료사전
- 소단위 명세서
- 미니 프로젝트 1

5부 소프트웨어 품질관리

- 소프트웨어 품질관리 절차와 기준

3부 정보공학 방법론

- 통합 분석/설계 방법론
- 데이터모형 구축 방법
- 관계형 DB 설계 과정
- 미니 프로젝트 2

4부 객체지향 방법론

- 객체지향 방법론의 이해
- UML 모델링
- 미니 프로젝트 3

강의 계획표



주	해당 장	주제
1	1장	시스템 개발 과정의 이해
2	2장	시스템 분석/설계 방법론과 관련 문서
3	3, 4장	구조적 분석 방법론, 자료흐름도
4	5장	자료흐름도 작성단계
5	6장	자료사전
6	7장	소단위 명세서
7	8장	미니 프로젝트(1)
8	필기/실기	중간고사
9	9장	통합 분석/설계 방법론
10	10장	데이터모형 구축 방법
11	11, 12장	관계형 DB 설계 과정, (미니 프로젝트(2))
12	13장	객체지향 방법론
13	14장	UML 모델링
14	15장	미니 프로젝트(3)
15	16장	소프트웨어 품질관리
16	필기/실기	기말고사

평가개요		배점	
	진단평가	-	
	출석	10	
	과제	30	
	중간고사	30	
	기말고사	30	

시스템 분석 설계

Chapter 01 시스템 개발 과정의 이해

목차

- 01 소프트웨어 공학
- 02 시스템과 시스템 개발자
- 03 SDLC 모형
- 04 프로토타입 모형
- 05 프로젝트 관리

학습목표

- 소프트웨어 위기와 소프트웨어 공학의 출현 배경을 이해한다.
- 시스템(소프트웨어) 개발 과정에 참여하는 사람들에 대해 알아본다.
- 시스템 개발 단계를 이해하기 위해 SDLC 모형 및 프로토타입 모형을 학습한다.
- 프로젝트 관리의 개념을 이해하고 절차를 학습한다.

01. 소프트웨어의 이해

■ 소프트웨어의 정의

- 프로그램: 프로그래밍한 원시 코드(source code)
- 소프트웨어
 - 프로그램(코드)을 비롯해 개발 과정에서 생성되는 모든 산출물과 각 단계에서 만들어지는 문서와 사용자 매뉴얼 등
» (자료 구조, 데이터베이스 구조, 테스트 결과 등)



그림 1-1 소프트웨어가 사용되는 곳

01. 소프트웨어의 이해



■ 소프트웨어의 특징

• 제조가 아닌 개발

- 제조: 개인 능력에 따라 차이가 크므로 능력에 따른 결과물의 차이는 크지 않음
- 소프트웨어 개발 과정은 제조와 달리 개인 능력에 따라 차이가 큼

• 소모가 아닌 품질 저하

- 하드웨어: 오래 사용하면 부품이 닳고 기능도 떨어짐
- 소프트웨어: 닳지 않으며 또한 시간이 지나도 고장 빈도가 높지 않음, 사용 시작 단계부터 사용자의 요구가 계속 발생

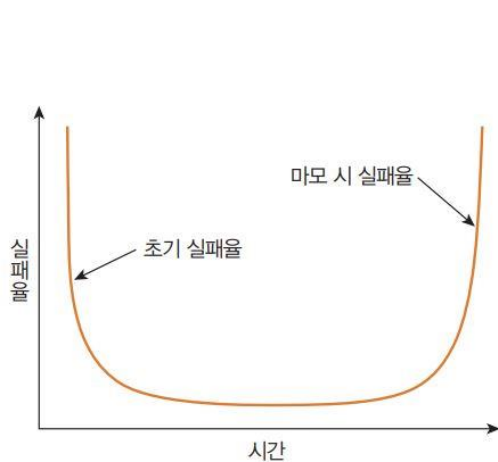
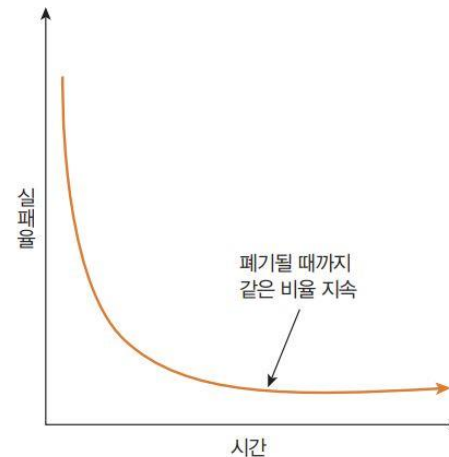
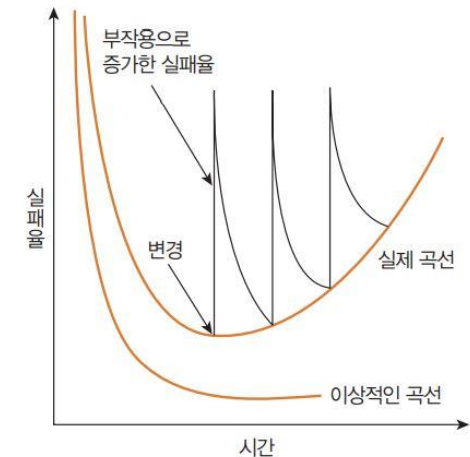


그림 1-2 하드웨어 실패 곡선



(a) 이상적인 소프트웨어의 실패 곡선

그림 1-3 소프트웨어 실패 곡선



(b) 실제 소프트웨어의 실패 곡선

■ 소프트웨어 개발의 어려움

① 개집 짓기

- 연장과 나무만 있으면 똑딱똑딱 빠른 시간에 지을 수 있음
- 도면을 따로 그리지 않고 머릿속으로 구상만 해도 가능
- 지은 집이 균형이 조금 맞지 않아도 사용하는 데 큰 문제가 없음



01. 소프트웨어의 이해

■ 소프트웨어 개발의 어려움

② 단독주택 짓기

- 개집과 달리 여러 사람이 참여해야 하며 설계와 공정 과정도 필요
- 단순한 연장만으로는 안 되고 전문적인 도구들이 필요



■ 소프트웨어 개발의 어려움

③ 대형 빌딩 짓기

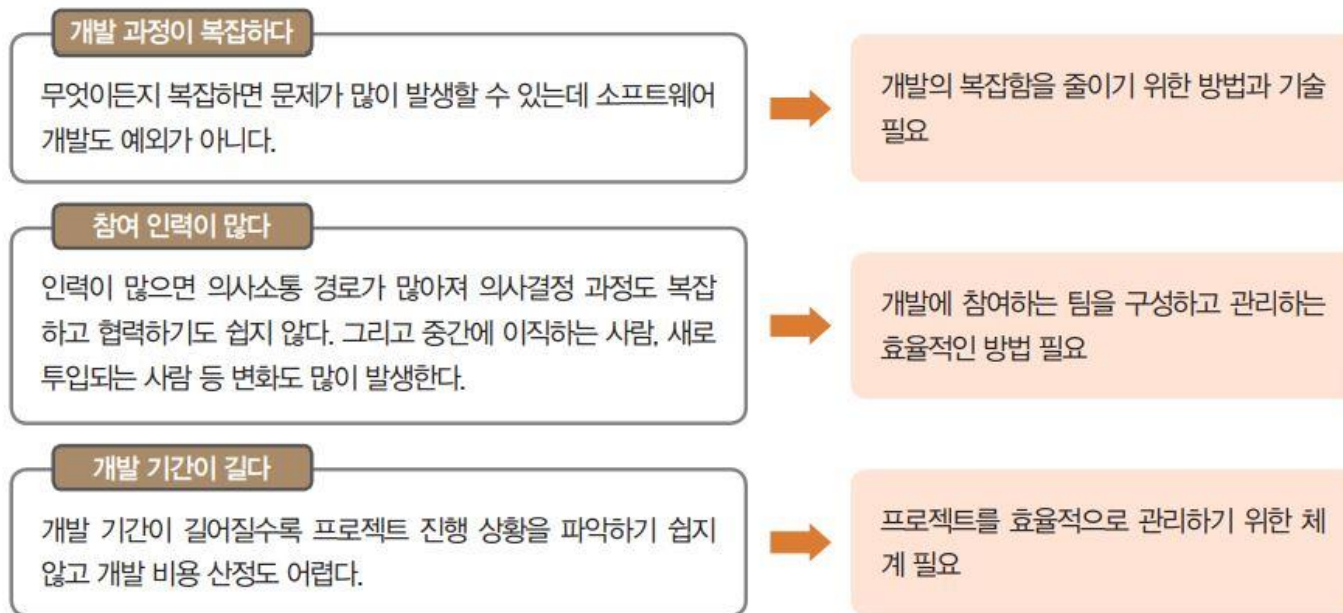
- 지반이 안전한지부터 확인한 후에 토목공사가 이루어짐
- 지진에 대비한 내진 설계, 하중 문제 등 고려해야 할 사항도 훨씬 많고 복잡
- 그래서 수많은 사람과 여러 작업을 중앙에서 조정하고 통제하는 팀이나 부서가 필요



01. 소프트웨어의 이해

■ 소프트웨어 개발의 어려움

- 대규모 소프트웨어를 개발하는 것은 대형 빌딩 짓기와 유사



01. 소프트웨어의 이해



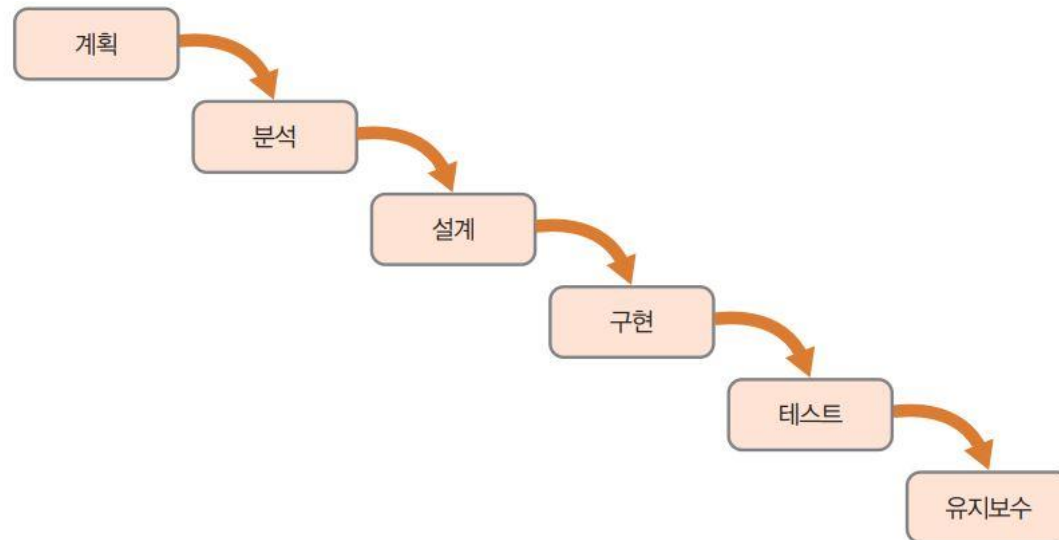
■ 소프트웨어 공학

- 소프트웨어 공학의 학문적 정의

- » 품질 좋은 소프트웨어를 경제적으로 개발하기 위해 계획을 세우고, 개발하며, 유지 및 관리 하는 전 과정에서 공학, 과학 및 수학적 원리와 방법을 적용해 필요한 이론과 기술 및 도구 등에 관해 연구하는 학문

- 소프트웨어 개발 생명주기

- 소프트웨어를 만들기 위해 계획 단계에서 유지보수 단계에 이르기까지 일어나는 일련의 과정
 - (계획, 분석, 설계, 구현, 테스트, 유지보수)



02. 소프트웨어 개발 프로세스



■ 프로세스

- 프로세스

- 프로세스: 흔히 일을 처리하는 과정 또는 순서
- 주어진 일을 해결하기 위한 목적으로 그 순서가 정해져 수행되는 일련의 절차
- 어떤 일을 해결하고자 할 때는 해당 프로세스만 잘 따르면 목적을 달성할 수 있음
 - » 예) 요리, 빨래, 소프트웨어 설치 등...



1.1 소프트웨어 위기

■ 주먹구구식 개발로 개발 기간이 지연되거나 실패로 돌아감

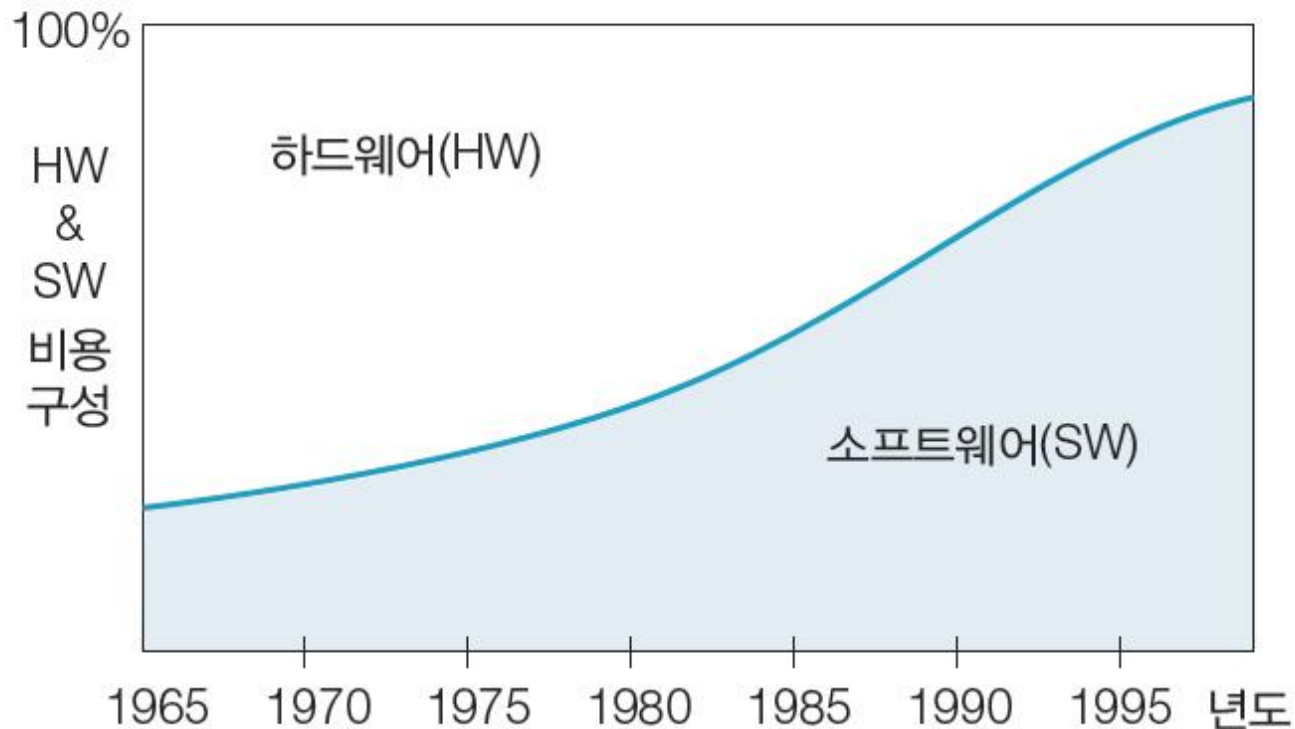


그림 1-1 하드웨어와 소프트웨어 비용 구성률

1.2 소프트웨어 공학의 출현

■ IEEE의 <소프트웨어 공학>에 대한 정의

소프트웨어의 개발, 운용, 유지보수 및 파기에 대한 체계적인 접근 방법

→ 품질이 좋은 소프트웨어를 최소한의 비용으로 계획된 일정에 맞추어 개발하는 것

1.3 소프트웨어 공학의 계층 구조



그림 1-2 소프트웨어 공학에서 다루는 주제의 계층 구조

■ 도구

- 프로그램 개발 과정에서 사용되는 여러 가지 방법을 자동화한 것
- CASE는 소프트웨어 개발 전 단계를 지원하는 대표적인 도구

1.3 소프트웨어 공학의 계층 구조

■ 방법론

■ 프로세스 중심 방법론

- 1970년대 제시됨
- 자료의 변환과정과 프로세스를 강조하여 프로그램을 개발하는 방법

■ 자료 중심 방법론

- 프로그램을 개발할 때 사용할 자료를 규명하고 자료와 자료 간의 관계를 분석한 후 자료구조를 정의하고 이를 토대로 프로세스 구조를 고안하는 방법
- 데이터베이스에 기반한 쿼리 중심의 프로그래밍 방식

■ 객체지향 방법론

- 객체를 캡슐화함으로써 좀 더 쉽게 프로세스의 모듈화, 정보 은닉, 코드 재사용의 효율성을 꾀할 수 있는 방법
- 프로세스 중심 방법론과 자료 중심 방법론의 장점을 묶어 진화한 방법론

1.3 소프트웨어 공학의 계층 구조

■ 프로세스

- 소프트웨어 개발에 필요한 작업 이름, 작업 내용, 결과물, 절차, 지시사항 등을 작업 사이의 선후 관계와 더불어 나타낸 것
- 다음 네 가지 영역으로 구분됨
 - 소프트웨어 명세
 - 소프트웨어 개발
 - 소프트웨어 검증
 - 소프트웨어 진화

■ 품질

- 소프트웨어 품질을 평가하는 기준은 아래와 같음
 - 정확성
 - 유지보수성
 - 무결성
 - 사용성

2.1 시스템의 개념

■ 시스템

컴퓨터에 의해 처리가 가능한 형태로 자료를 변환하여 입력하고, 그 자료를 저장, 처리, 가공하여 필요한 시점에 정보를 출력할 수 있도록 설계되고 구현된 정보체계

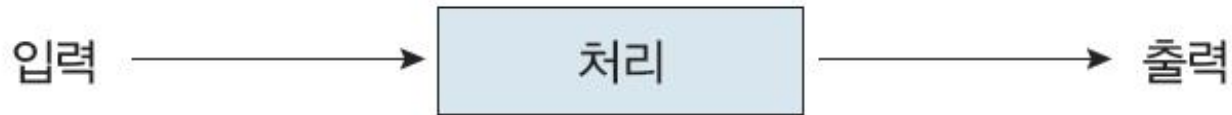


그림 1-3 시스템의 구조

2.2 시스템 개발에 참여하는 사람들

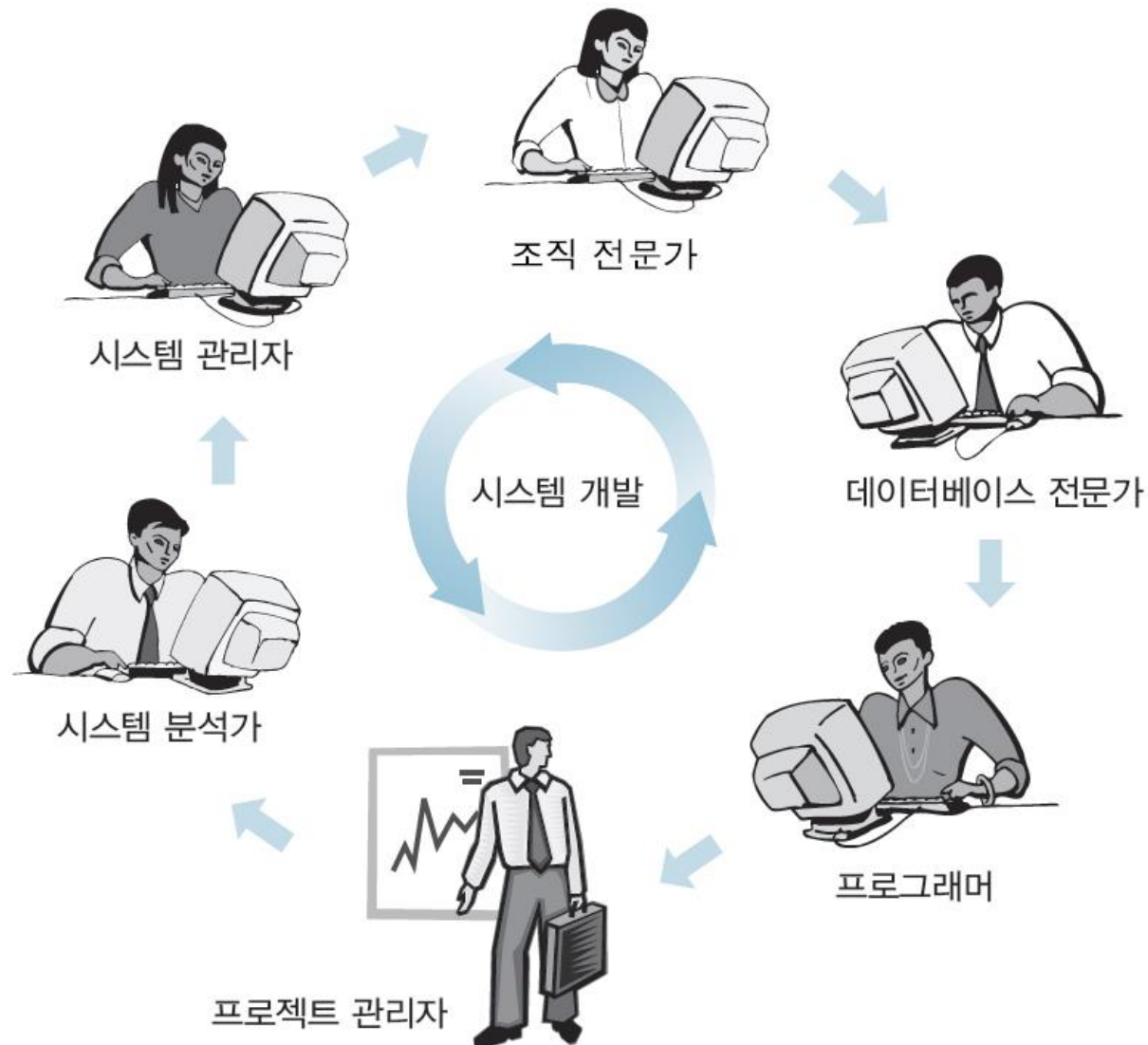


그림 1-4 시스템 개발 참여 인력

3.1 SDLC 모형의 5단계

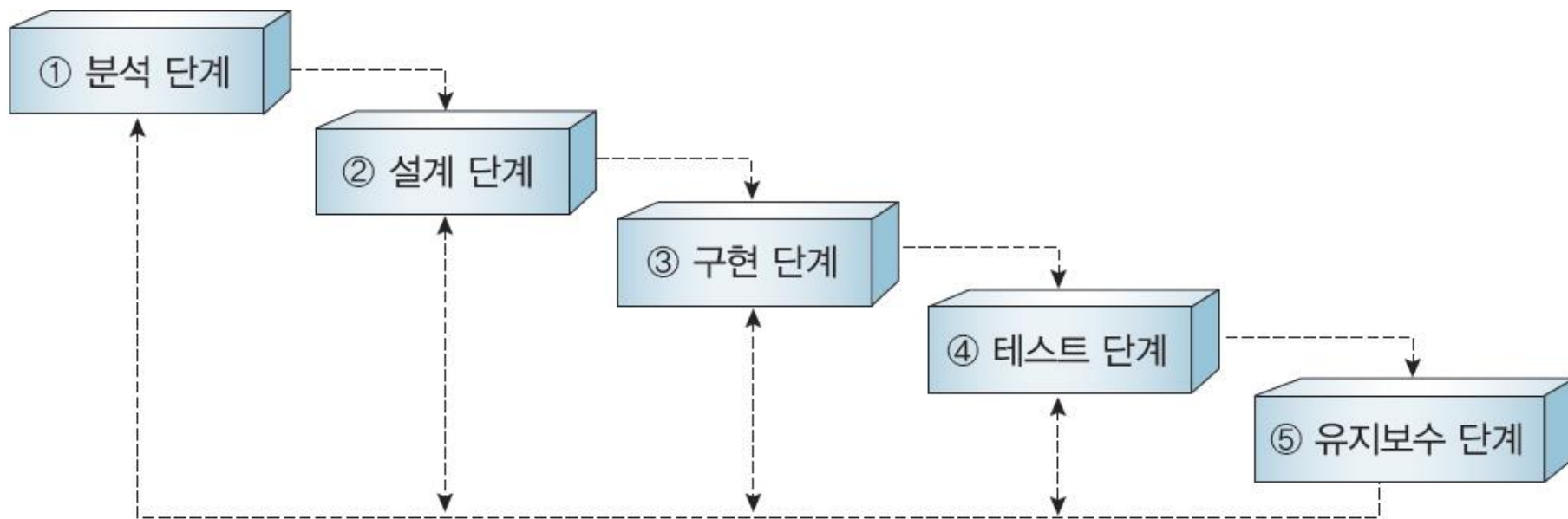


그림 1-5 SDLC 모형 5단계

3.2 SDLC 모형과 건축 과정의 비교

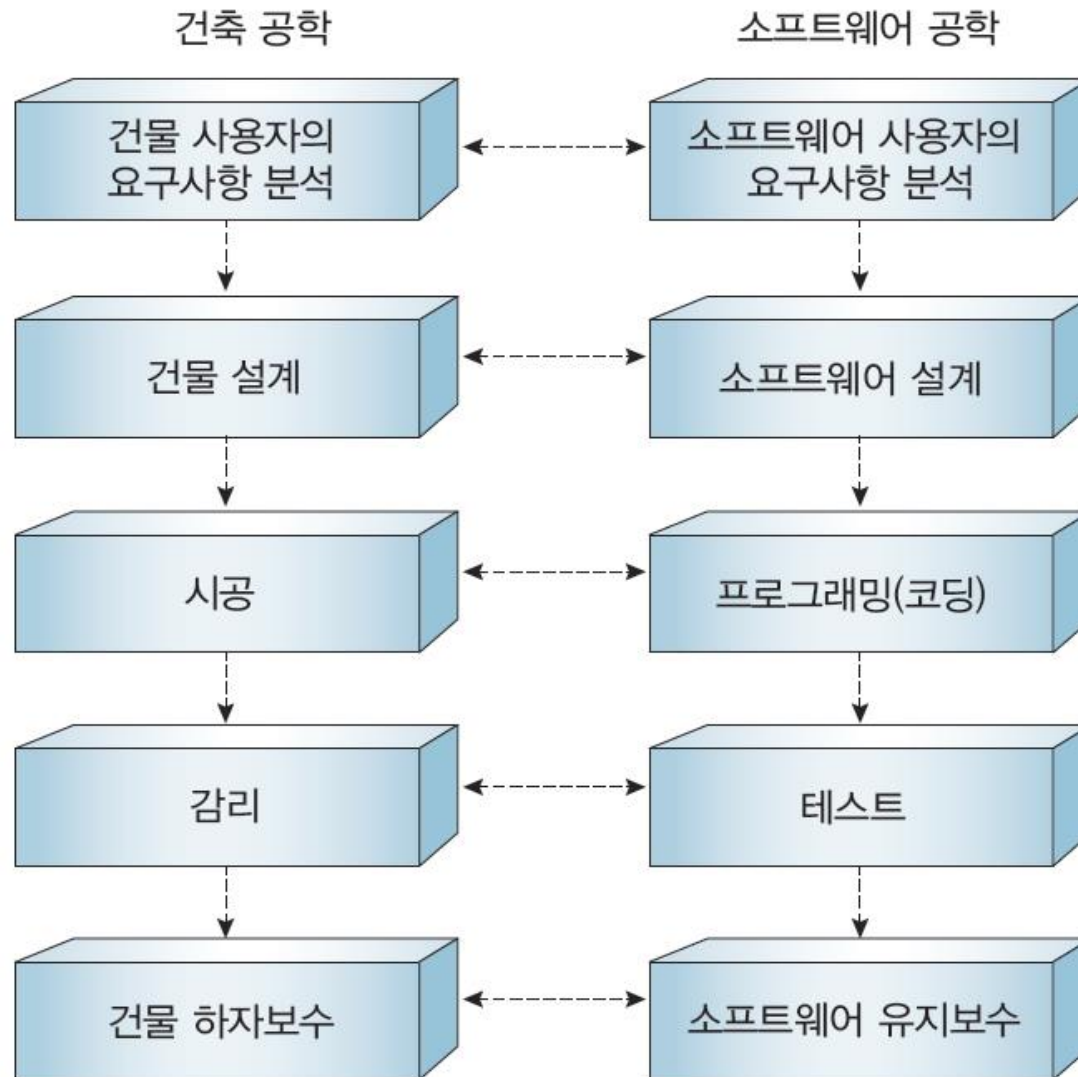


그림 1-6 건축 단계와 시스템 개발 단계의 비교

3.3.1 SDLC 모형의 단계별 인력소요

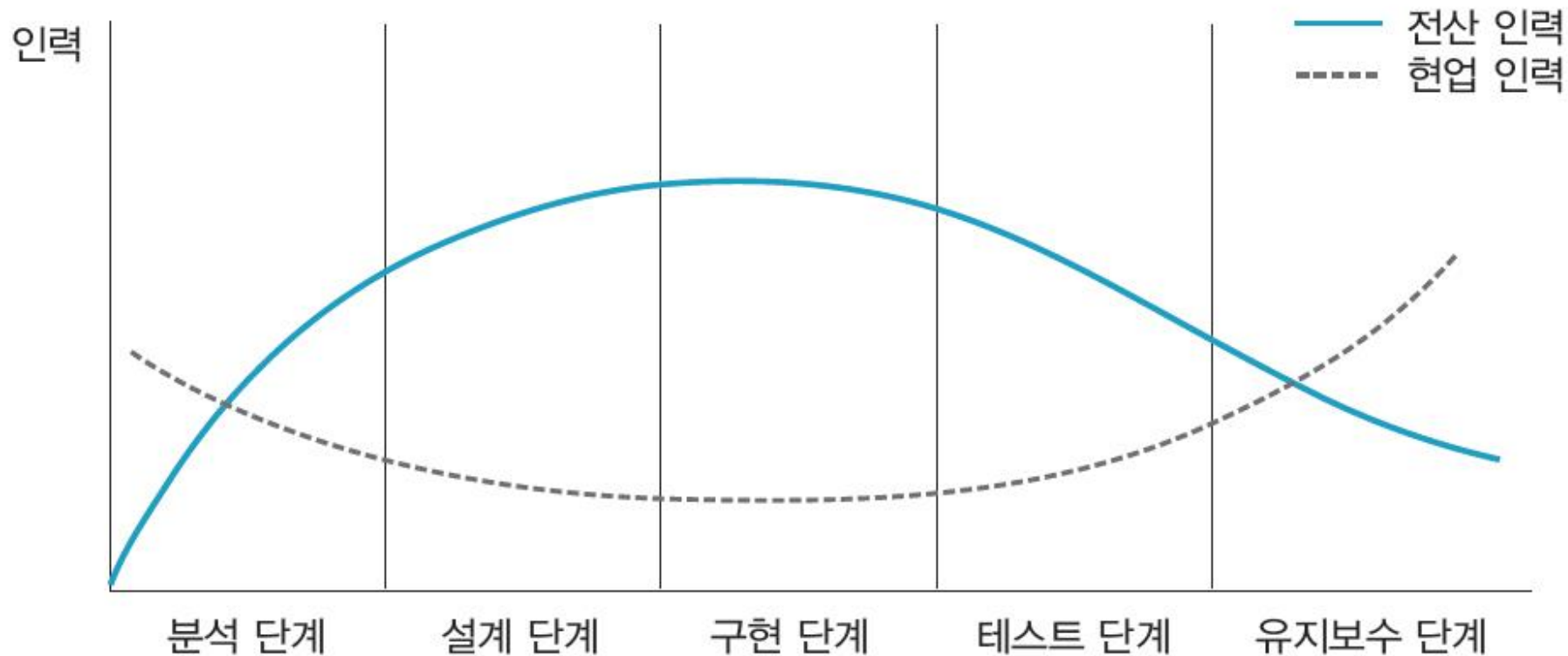


그림 1-7 SDLC 모형의 단계별 인력소요

3.3.2 SDLC 모형의 장단점

■ 장점

- 시스템 개발의 각 단계가 비교적 명확
- 각 단계들 간에 유기적인 연관성을 가지고 있어 쉽게 적용할 수 있음

■ 단점

- 충분한 분석을 기반으로 개발이 진행되지 않았을 경우 테스트 단계 또는 유지 보수 단계에서 문제점이 노출되어 이를 개선하는 데 많은 비용과 시간이 소요 됨
- 대형 프로젝트의 경우 긴 개발기간 동안 외부환경이나 내부 정책이 변화할 소 지가 크고, 이를 개선하기 위해 이전 단계로 되돌아가 변경관리를 해야 하므로 막대한 시간과 비용이 들어감

4.1 프로토타입 모형

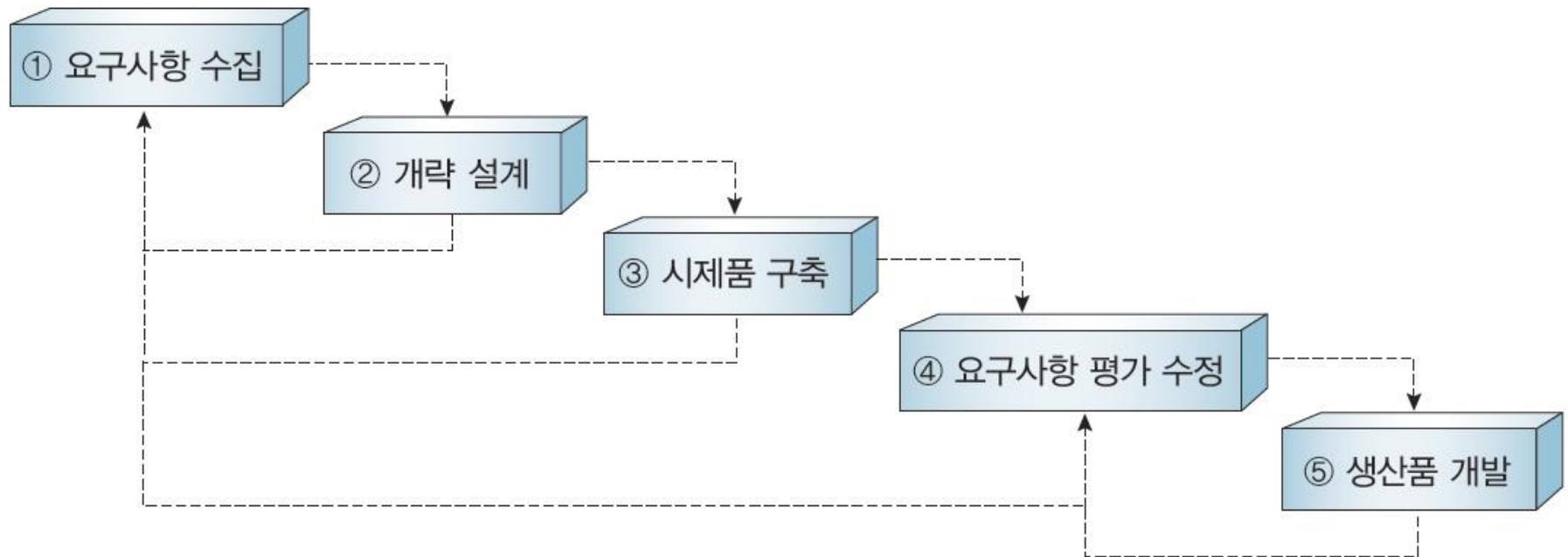


그림 1-8 프로토타입 모형의 개발 과정

4.2 프로토타입 모형의 장단점

■ 장점

- 개발 초기에 미리 결과물을 확인할 수 있다는 점에서 사용자의 이해를 도움
- 개발의 초기 단계에서 수정·보완할 사항을 미리 파악할 수 있음
- 분석 및 설계 과정에 사용자가 동참하여 즉각적인 피드백을 줄 수 있음

■ 단점

- 일회적 프로젝트나 대규모 프로젝트의 개발에는 적용하기 쉽지 않음
- 불완전한 요구사항을 바탕으로 시제품이 만들기 때문에 결과적으로 불완전한 시스템을 산출하여 수정과 보완에 많은 인력과 시간이 투입됨

5.1 프로젝트 관리자의 활동

- 제안서 작성
- 프로젝트 계획과 일정 수립
- 프로젝트 비용 산정
- 프로젝트 모니터링과 중간평가 실시
- 실무자 선정과 평가
- 보고서 작성과 발표

5.2 프로젝트 계획

■ 프로젝트 계획과 함께 수립해야 할 계획

- 품질계획 : 품질을 위한 계획과 표준을 설명
- 검증계획 : 시스템 검증을 위한 접근방법, 자원, 일정을 설명
- 구성관리계획 : 구성관리 방안과 구조를 설명
- 유지보수계획 : 유지보수를 위한 요구사항, 비용, 노력을 설명
- 인력개발계획 : 팀원들의 경험 축적과 기술개발 설명

5.3 프로젝트 일정 수립

■ 프로젝트 일정 수립 예

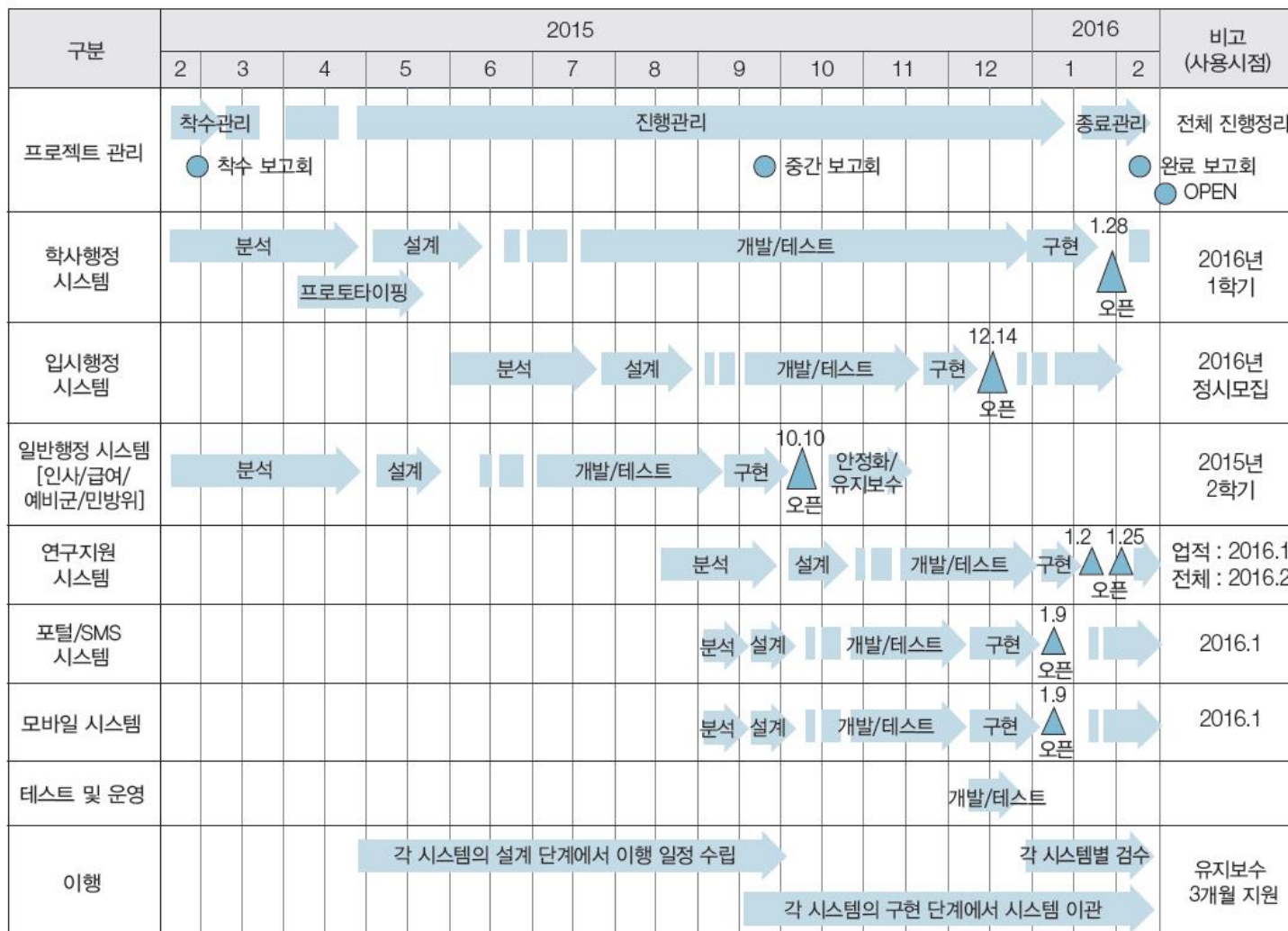


그림 2-12 일정계획 예

5.4 품질관리

- 개발주기 전체에 걸쳐 시행하는 검열, 검토, 테스트 등을 말함
- 품질은 설계 품질과 적합 품질로 나뉨
 - 설계 품질 : 사용자의 요구를 반영하여 설계자가 시스템 설계 시 설계 문서에 적용하는 품질 특성
 - 적합 품질 : 구현할 때 설계 명세대로 이행하는 정도

5.5 위험관리



표 1-1 소프트웨어의 위험 요소

종류	위험 요소	내용
프로젝트 위험	인력부족	경험 있는 실무자가 프로젝트 완료 전에 사직
	관리변화	조직 내에서 관리의 우선순위 변화
	하드웨어 미비	개발에 필수적인 하드웨어 수급이 배당지연 등으로 부족
	규모의 과소 평가	시스템 크기의 과소 평가
생산품 위험	요구변경	계속적인 요구사항 변경
	명세서 지연	기본적인 인터페이스 명세서가 일정대로 작성되지 못함
비즈니스 위험	기술변화	기반 기술을 신기술로 대체하지 못한 기술적 취약
	생산품 경쟁	경쟁 상품이 먼저 시장에 나옴

5.5 위험관리

■ 위험관리 프로세스

- 1단계 위험 인식
- 2단계 위험 분석
- 3단계 위험 계획
- 4단계 위험 모니터링



Thank You
