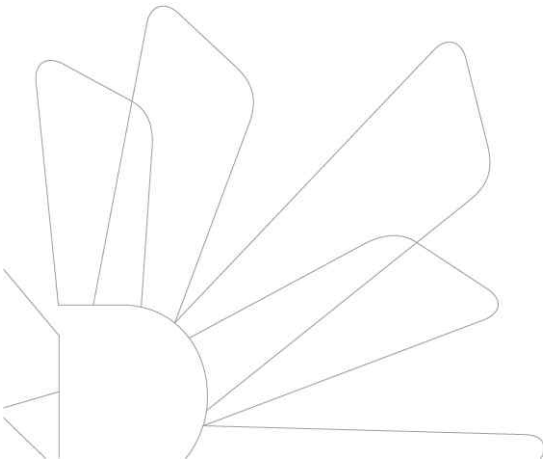


소프트웨어 공학 개론

강의 1: 소개

최은만
동국대학교 컴퓨터공학과



강좌의 목표

Programmer



Software Engineer



- **Goal #1**

- 품질 좋은 소프트웨어를 함께 개발할 수 있는 스킬을 계발

- **Goal #2**

- 프로젝트 사이클 경험

소프트웨어 공학을 잘 배우려면?

- Learning by doing
- Learning by making
- Learning by design
- Learning by coding
- Learning by testing
- Learning by tinkering
- Learning by engineering



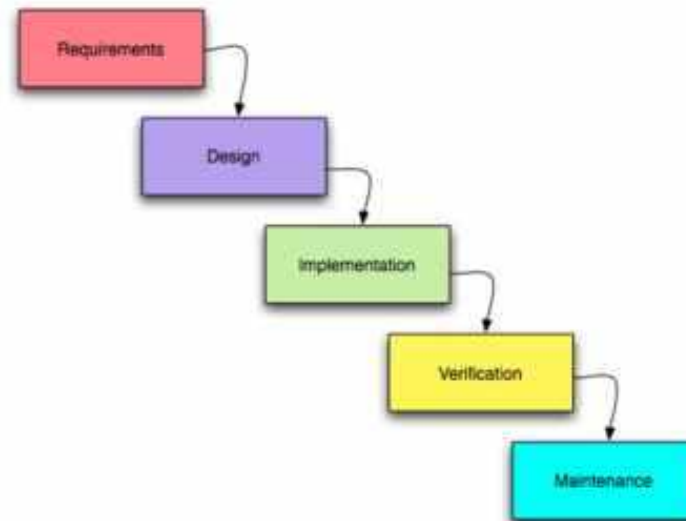
소프트웨어 개발

- 프로그래밍

- 상상력
- 구현력

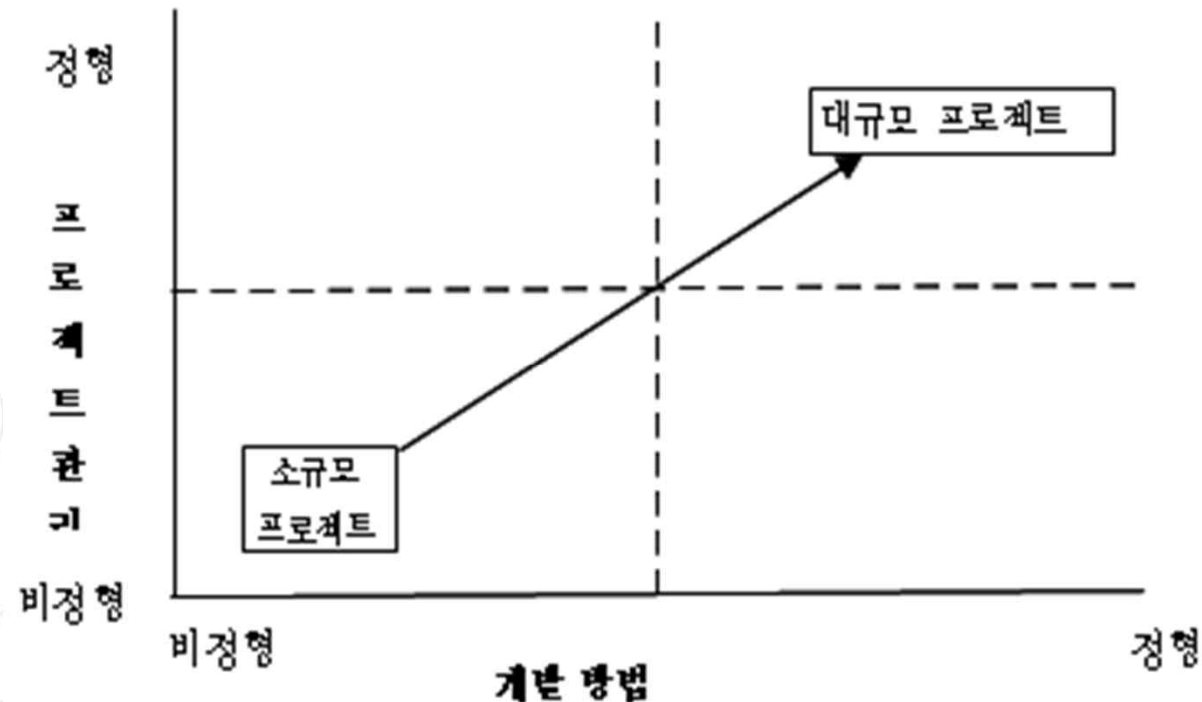
- 경험이 중요

- 문제의 발생부터 분석, 통합
- 검증



다루는 문제 - 규모

- 수만 줄의 소프트웨어를 개발할 때는 수백 줄의 프로그램을 개발하는 데 사용하는 방법과는 다른 방법을 적용
- 엔지니어링 식 접근 방법 - 방법, 절차, 도구 사용



품질과 생산성

- 엔지니어링 작업에서는 비용, 일정, 품질과 같은 변수가 중요

- 비용

- Man-Month로 측정

- 일정

- 짧은 time-to-market

- 품질

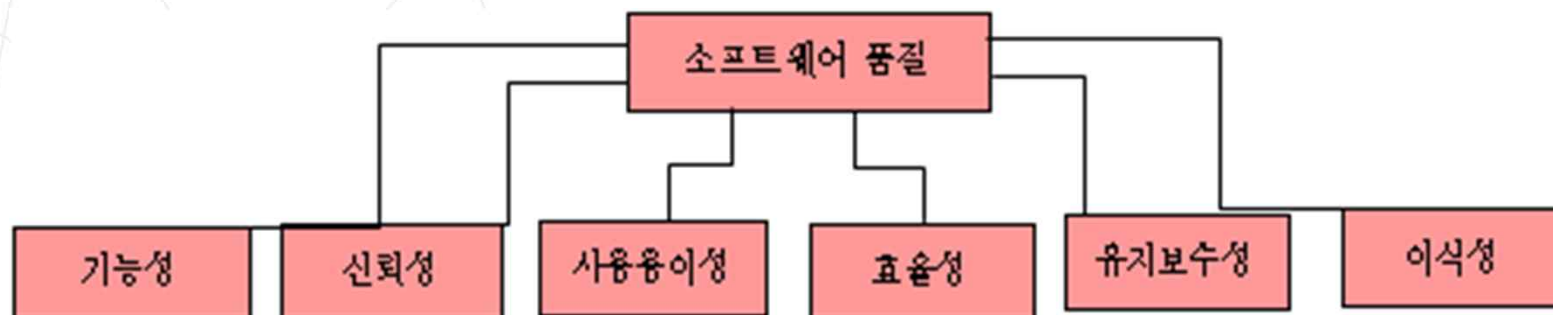


품질



생산성

그림1.8 소프트웨어 공학의 목표



일관성과 재현성

- **일관성**

- 프로젝트의 결과를 어느 정도 정확하게 예측가능
- 더 높은 품질의 제품을 생산

- **프로세스의 표준화가 필요**

- ISO 9001
- CMM(Capability Maturity Model)

- **재현성**

- 개발하는 시스템 마다 높은 품질과 생산성을 갖도록 만드는 것
- 개발 능력, 결과의 재현성

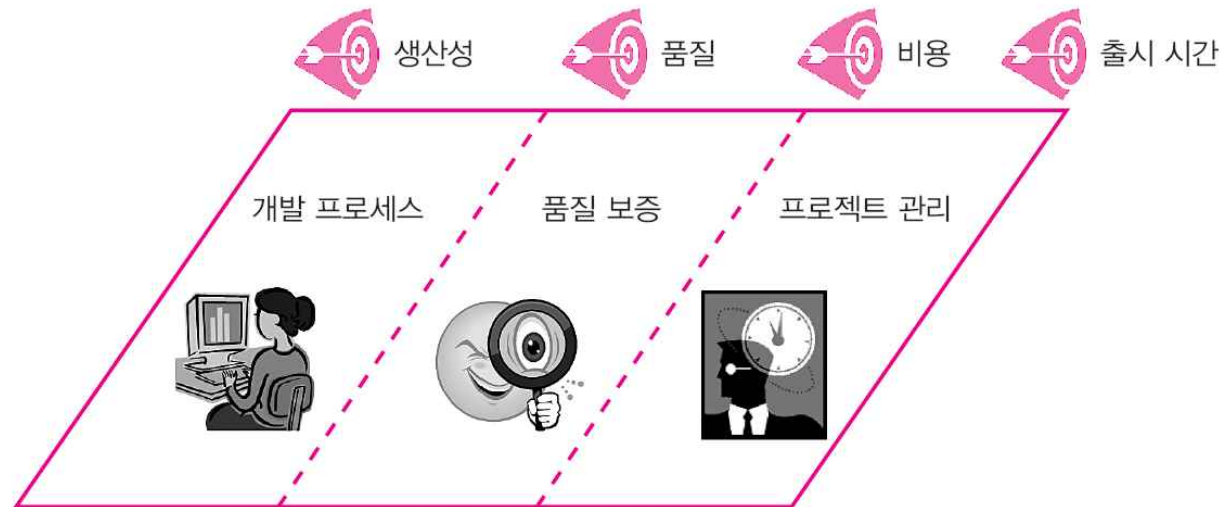
소프트웨어 공학의 접근 방법

- 프로젝트를 수행하는 동안 얻은 품질과 생산성은 여러 가지 요인에 좌우됨
- 품질을 좌우하는 세가지 : 인력, 프로세스, 기술
 - 프로젝트 삼각 균형



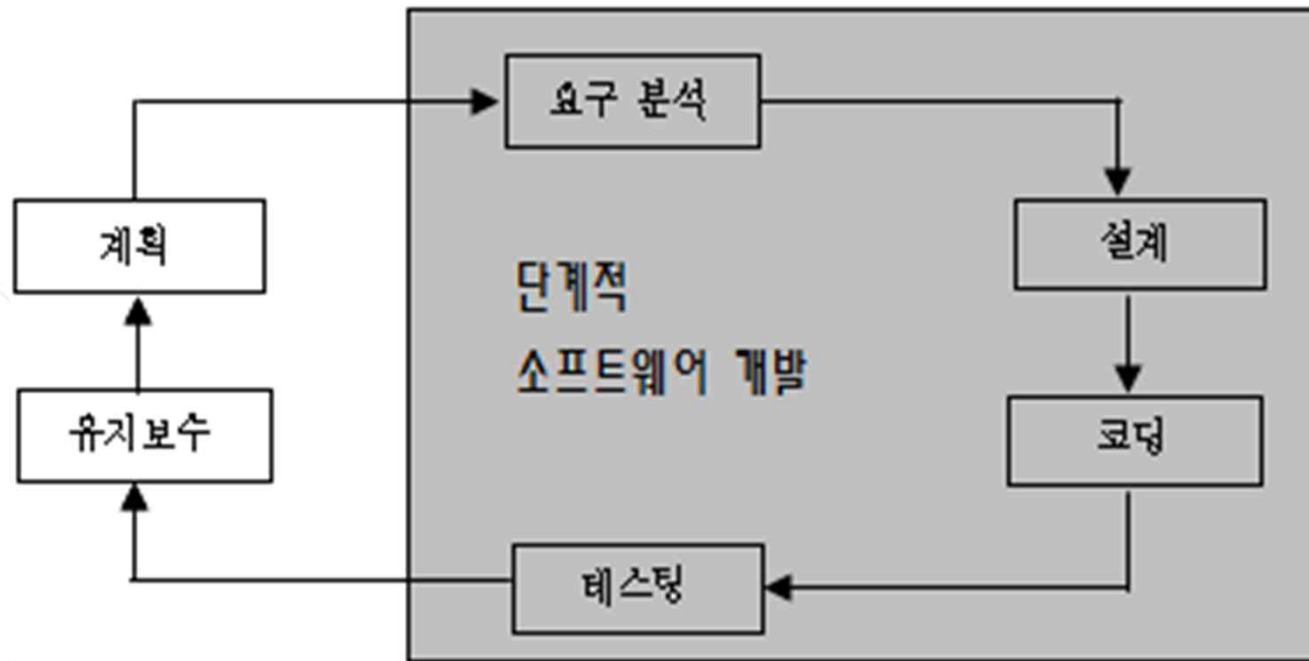
소프트웨어 공학의 접근 방법

- 소프트웨어를 개발하는 프로세스를 소프트웨어와 분리
- 소프트웨어 공학은 소프트웨어 제작과정에 집중
 - 알고리즘, 운영 체제, 데이터베이스 등은 소프트웨어 제품 자체에 초점
- 소프트웨어 엔지니어링 작업의 종류



단계적 개발 프로세스

- 단계적 개발 프로세스를 따르는 이유
 - 소프트웨어의 문제를 나눠 여러 개발 단계에서 다른 관점을 다루기 때문
 - 개발하는 동안 정해진 시점에 품질과 진행을 체크할 수 있음



단계적 개발 프로세스

Requirements

Design

Implementation

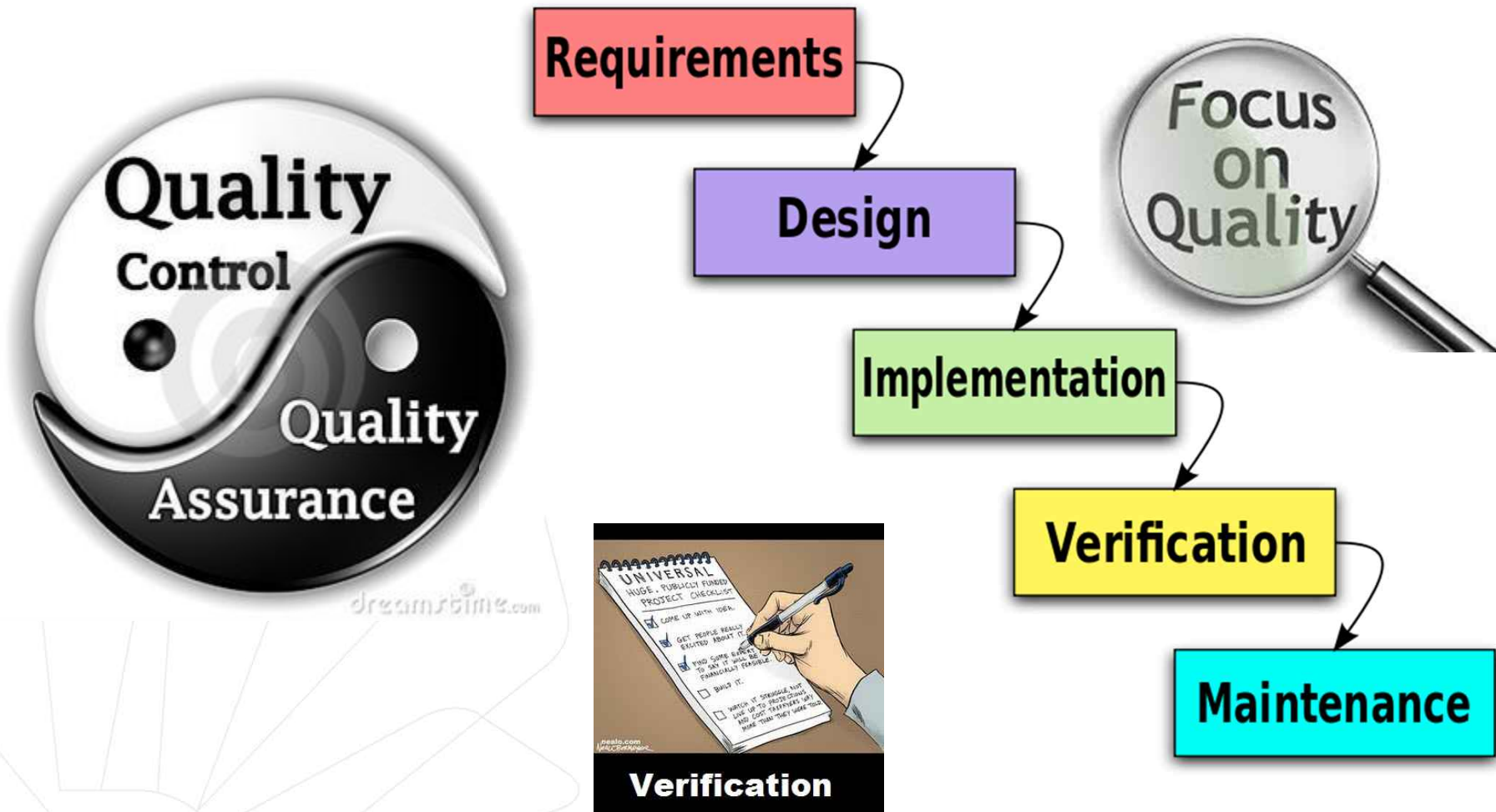
Verification

Maintenance

개발 단계

단계	초점	주요작업과 기술	결과물
분석	● 시스템을 위하여 무엇을 만들 것인가?	1. 분석 전략 수립(3장) 2. 요구 결정(3장) 3. 사용 사례 분석(4장) 4. 구조적 모델링(6장) 5. 동적 모델링(6장)	요구 명세서
설계	● 시스템을 어떻게 구축할 것인가?	1. 설계 전략 수립(7장) 2. 아키텍처 설계(5장) 3. 인터페이스 설계(7장) 4. 프로그램 설계 5. 데이터베이스, 파일 설계(7장)	설계 명세서
구현	● 시스템의 코딩과 단위 시험	1. 프로그래밍(8장) 2. 단위 테스트(9장) 3. 시스템 안정화 및 유지보수(10장)	코드, 테스트 계획
테스팅	● 시스템이 요구에 맞게 실행되나?	1. 통합 테스트(9장) 2. 시스템 테스트(9장) 3. 인수 테스트(9장) 4. 시스템의 설치(9장) 5. 프로젝트 관리 계획	테스팅 결과 보고서

품질 보증



프로젝트 관리



강의 내용과 방법

- 소프트웨어 개발 기술의 습득
 - 요구 분석 방법
 - 설계 기법
 - 코딩 스타일, 테스트 기법
 - 릴리스 및 유지보수 기술
 - 객체지향 설계 기술(UML)
- 플립드 러닝(6주)
 - 강의 듣기 - Out of class room
 - 디자인 스튜디오 - In class room
- 프로젝트 중심 교육
 - 문제 해결에 집중
 - 배운 스킬을 적용

프로젝트

- 프로젝트 팀 구성
 - 다음 주 액티비티
 - 주제는 간단한 인터넷 쇼핑몰/출석체크 시스템 어떤 것을 할 것인지 스스로 정함
- 4명의 학생으로 팀 구성
 - 실제 프로젝트와 같이 팀 구성(리더, Writer, 엔지니어,...)
 - 정확한 역할 분담과 팀 스피릿 발휘
- 코드 포함 7개의 프로젝트 과제
 - 요구한 결과물 제출
- 평가 요소
 - 성취 결과(burndown chart), 완성도, 기여도

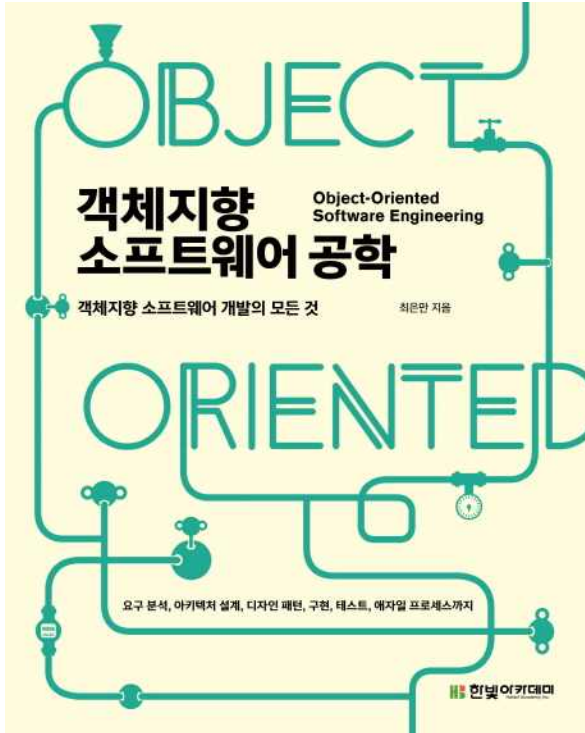
성적 평가 방법

시험 60%	중간시험 (30%) 기말시험 (30%)
팀 프로젝트 25%	#1 – 팀 구성 및 계획(5점) #2 – 사용 사례(15점) #3 – 클래스 다이어그램(30점) #4 – 시퀀스 다이어그램(30점) #5 – UI 설계(20점) #6 – 소스코드, 데모(80점) #7 – 테스트(20점)
플립러닝 10%	플립러닝 연습 문제 발표 및 Activity 참여
출석 참여율 5%	출석 1/4 불참자는 F

- 추가 시험 없음, 성적 올려주기 없음
- 팀 프로젝트는 팀 단위로도 평가하지만 개인 기여도도 반영

교재

-
- 소프트웨어 공학**
- 최은환 지
- SOFTWARE ENGINEERING**
- 정익사 JUNGSA PUBLISHING CO.



Smart work and study

- 효과적인 협력
- 효율적인 시간 관리
- 꾸준한 학습 – Long-term memory !!!!!





Questions?

