



소프트웨어 공학

제1장 소프트웨어 공학 소개

2023. 2. 1

컴퓨터공학

허훈식

Learning Objectives

1. 소프트웨어 공학의 접근 방법을 알아본다.
2. 소프트웨어 공학의 주제와 도전 과제를 알아본다.
3. 소프트웨어 공학이 다루는 지식 영역을 알아본다.

1장. 소프트웨어 공학 소개

1.1 소프트웨어

➤ 소프트웨어

- 프로그램 + 프로그램의 개발, 운용, 보수에 필요한 관련 정보 일체
- 프로그램 + 모든 엔지니어링 산출물(설계 문서, 자료구조, DB 구조, 테스트 결과 등)

프로그램 뿐만 아니라
그 이상의 것도 포함하는 매우 포괄적인 개념

1.1 소프트웨어

➤ 소프트웨어 특징

- 복잡성(**complexity**)
 - 소프트웨어 내부 요소 간의 상호 작용이 매우 복잡
- 변경성(**changeability**)
 - 개발 과정에서 잦은 변경이 발생
- 비가시성(**invisibility**)
 - 개념적이고 무형적이기 때문에 쉽게 파악하기 힘들다.
- 순응성(**conformity**)
 - 외부 요구 또는 환경과 데이터 변화에 적절히 변형되는 특징

1장. 소프트웨어 공학 소개

1.1 소프트웨어

➤ 소프트웨어 3가지 유형

- 주문형, 패키지, 임베디드 소프트웨어

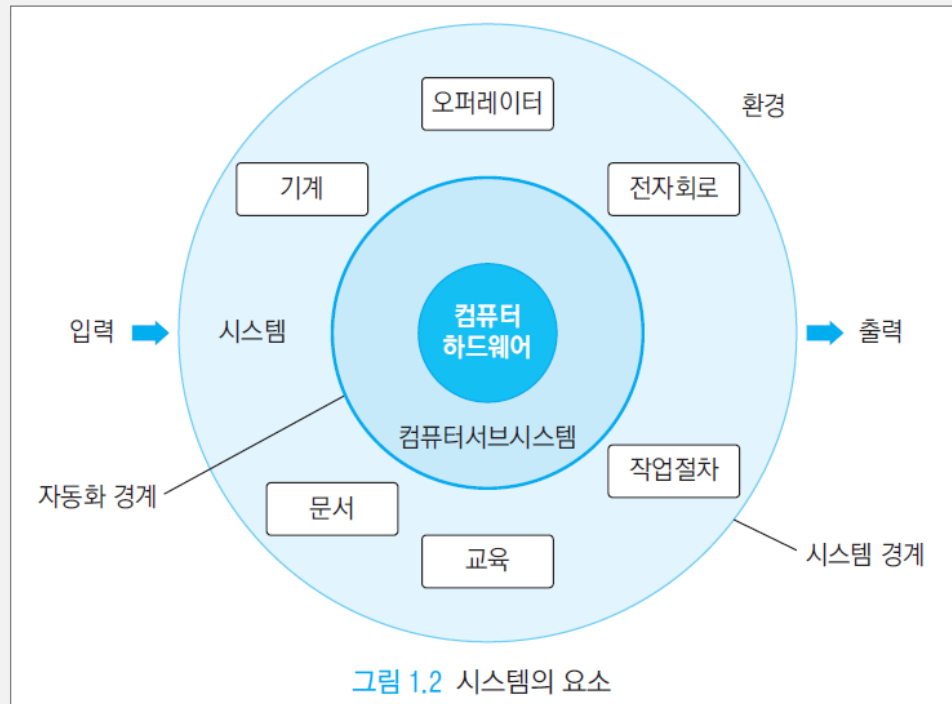
소프트웨어 분류	특징	사용되는 카피의 수	요구되는 하드웨어 성능	개발 인력
주문형 소프트웨어 	• 특정 고객 또는 기업의 요구를 만족시키기 위하여 제작한 소프트웨어	적음	낮음	많음
패키지 소프트웨어 	• 패키지화 하여 상업적으로 판매하는 소프트웨어 • 워드프로세서, 스프레드시트, 유통업체의 POS(Point of Sales) 시스템, 재정 분석, 주문 관리, 회계 관리 시스템	중간	높음	중간
임베디드 소프트웨어 	• 다른 시스템에 내장된 소프트웨어	많음	중간	적음

1장. 소프트웨어 공학 소개

1.1 소프트웨어 - 시스템

➤ 시스템

- 필요한 기능을 실현시키기 위하여 관련 요소를 어떤 법칙에 따라 조합한 집합체
 - 서브 시스템, 기능적 분할, 시스템 경계, 자동화 경계

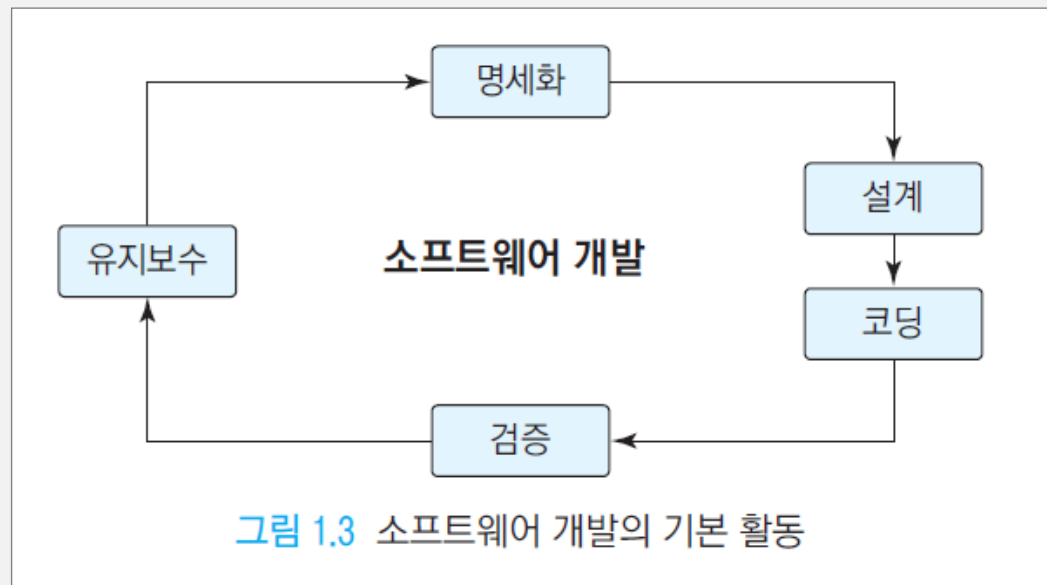


1장. 소프트웨어 공학 소개

1.2 소프트웨어 개발 작업

➤ 개발 기본 활동

- 명세화(Specification)
- 구현(Coding)
- 검증(Verification)
- 유지보수(Maintenance)



1.2 소프트웨어 개발 작업

➤ 개발 작업 특징

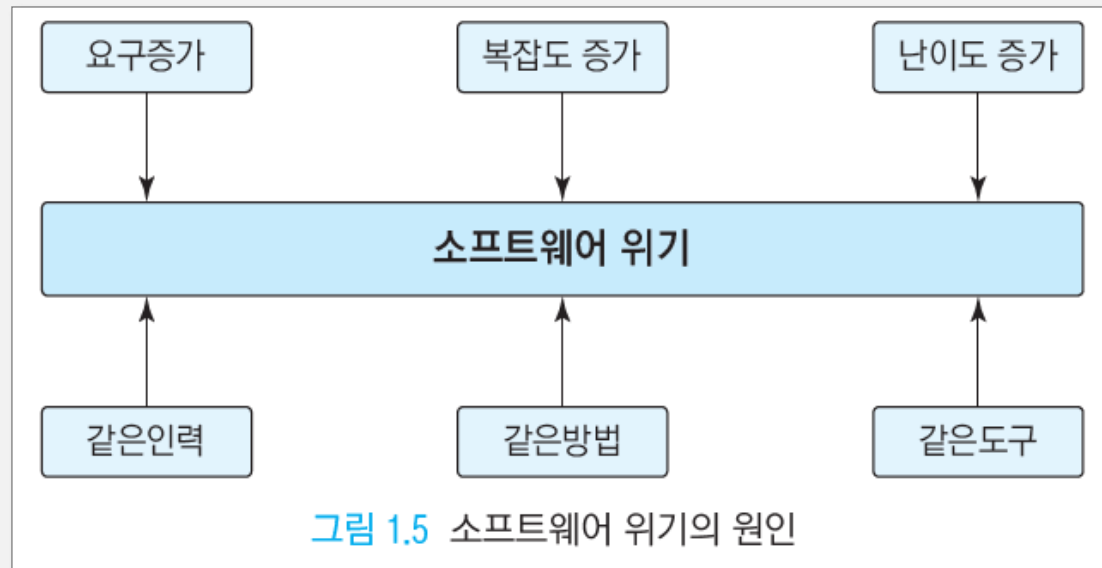
- 명세화의 어려움
 - 고객의 의도를 처음부터 정확하게 이해하기 쉽지 않다.
- 예측의 어려움
 - 외부 요소(업무 절차, 하드웨어, 데이터 형식, 성능 조건 등)의 영향으로 프로젝트 규모를 예측하기 어렵다.
- 재사용의 어려움
 - 전에 만들어 놓은 컴포넌트를 다른 소프트웨어에 적용할 수 없다.
- 유지보수의 어려움
 - 이미 만들어진 소프트웨어를 다른 유지보수 인력이 쉽게 이해하고 변경하기 어렵다.
- 고품질의 어려움
 - 한 줄의 코드를 수정하더라도 관련된 요소 간의 실행 조합을 모두 테스트하기 어렵다.

1장. 소프트웨어 공학 소개

1.2 소프트웨어 개발 작업

➤ 소프트웨어 위기(software crisis)

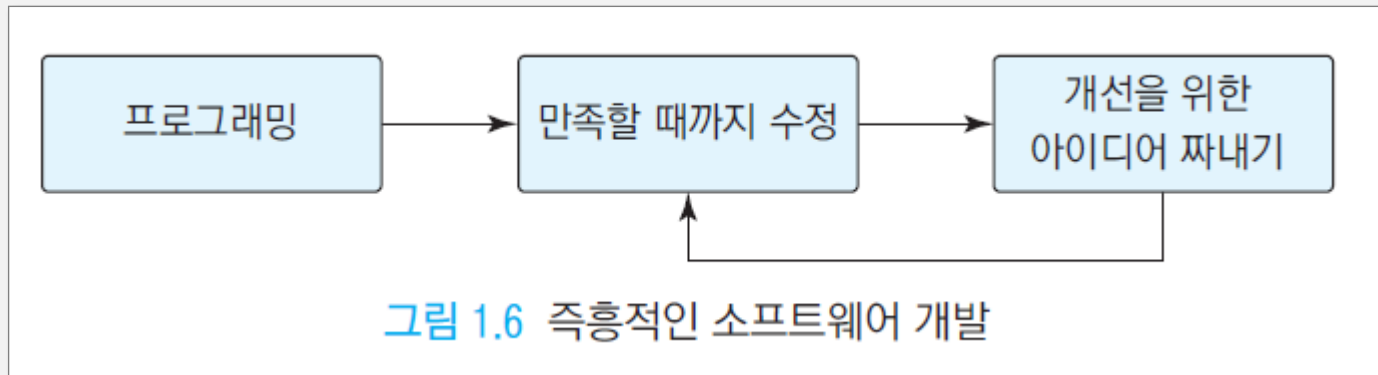
- 1968년 NATO 소프트웨어공학 학회에서 처음 등장
- 소프트웨어 수요가 급격히 증가하고 그 복잡성이 증가함에 따라 기존 방법이 충분하지 않아 발생한 문제 (문제의 복잡성 + 소프트웨어 공학 미성숙)



1.3 소프트웨어 공학의 접근 방법

➤ 즉흥적인 개발의 문제점

- 개발 지연과 예산 초과 - 수정 반복으로 인한 일정 지연
- 낮은 품질 - 소프트웨어 결함
- 유지보수 곤란 - 발견되지 않은 결함으로 많은 수정이 필요
- 재작업 - 잘못된 구조로 수정 또는 기능 추가시 이전 코드를 재작업하여 성능 저하

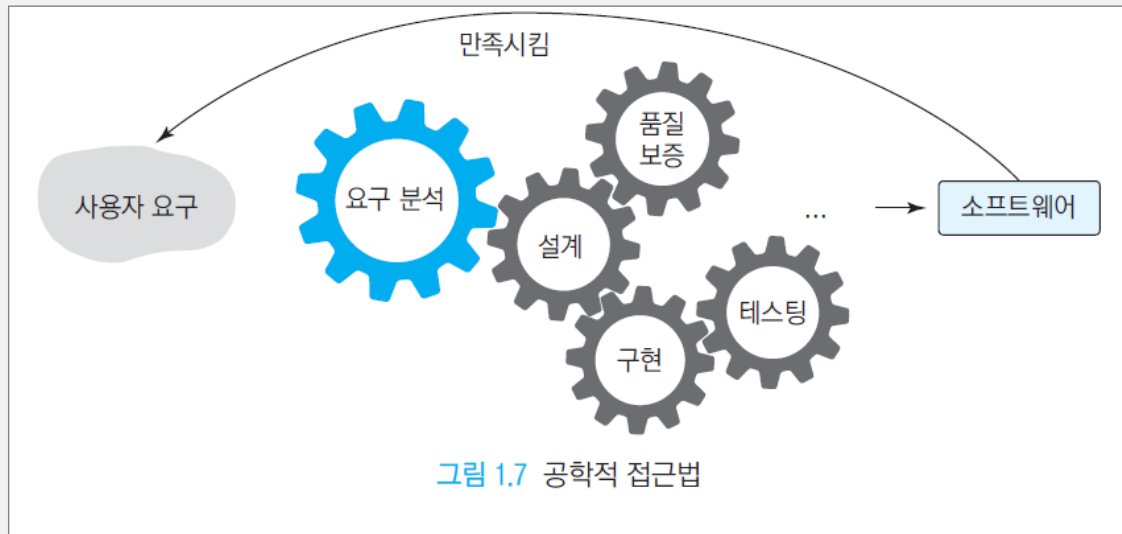


1장. 소프트웨어 공학 소개

1.3 소프트웨어 공학의 접근 방법

➤ 소프트웨어 공학 정의

- 소프트웨어의 개발과 운영, 유지보수, 소멸에 대한 체계적인 접근 방법
- 소프트웨어 개발 과정에 공학적인 원리를 적용하여 품질 좋은 소프트웨어를 개발
 - 소프트웨어 개발에 사용되는 방법이 일회성이 아닌 반복 사용이 가능함



1장. 소프트웨어 공학 소개

1.3 소프트웨어 공학의 접근 방법

➤ 소프트웨어 공학 목표

- 여러 가지 '원리와 방법'을 적용하여 품질 좋은 소프트웨어를 최소의 비용으로 계획된 일정에 맞추어 개발하는 것
 - 복잡도 낮춤, 비용 최소화, 개발 기간 단축, 대규모 프로젝트 관리, 고품질, 효율성



1.4 소프트웨어 공학의 주제

➤ 소프트웨어 공학의 목표를 달성하기 위해 크게 3가지 주제로 구성된다.

1) 단계적 프로세스

- 소프트웨어를 구현할 때 정해진 순서의 작업을 수행한다.

2) 품질 보증

- 개발 작업이 적절히 수행되었는지 확인한다.

3) 프로젝트 관리

- 개발과 품질보증 작업을 관리 감독한다.

1장. 소프트웨어 공학 소개

1.4 소프트웨어 공학의 주제

➤ 주제 1 : 단계적 프로세스

- 코딩에 치중하지 않고 요구 분석, 설계, 코딩, 테스트 등 정해진 절차를 따라 작업하는 것 : 정해진 시점에 품질과 생산성을 효율적으로 점검할 수 있다.

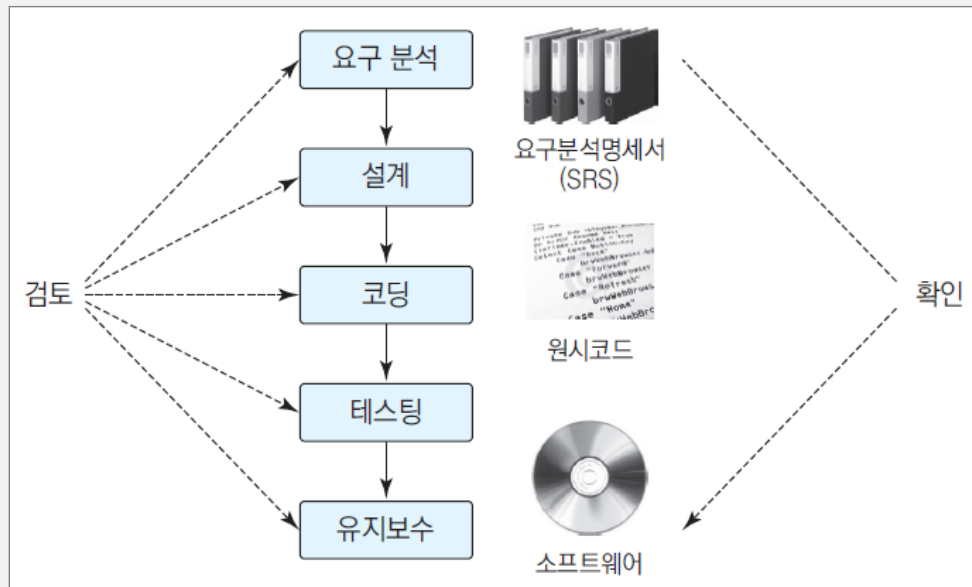


단계	초점	작업	결과물
분석	무엇을 만들 것인가	요구사항 결정 유스케이스 분석	요구 분석 명세서
설계	어떻게 구축할 것인가	아키텍처 설계 인터페이스, DB설계 프로그램 상세 설계	설계 명세서
구현	코딩 및 단위 테스트	프로그래밍 단위 테스트	코드
테스팅	요구에 맞게 실행되는가	통합 테스트, 시스템 테스트 인수 테스트	테스트 결과 보고서

1.4 소프트웨어 공학의 주제

➤ 주제 2 : 품질보증

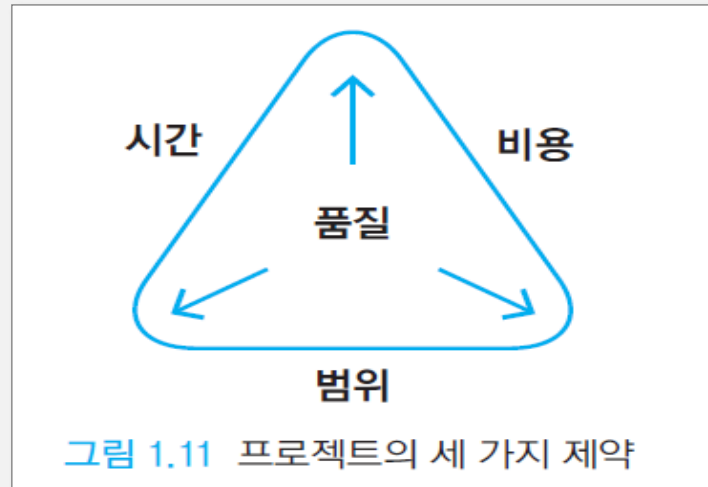
- 개발하고 있는 소프트웨어가 요구와 품질 수준을 만족시키는지 검토, 확인, 테스트
 - **검토** : 각 단계의 작업이 제시된 절차와 방법에 맞게 진행되었는지 체크하는 작업
 - **확인** : 개발 완료된 결과물이 품질 수준에 맞게 생산되었는지 검사하는 작업
 - **테스팅** : 구현된 소프트웨어를 실행하여 예상된 결과를 보이는지 확인하는 작업



1.4 소프트웨어 공학의 주제

➤ 주제 3 : 프로젝트 관리

- 프로젝트의 제약 조건들을 파악하고 적절히 관리하는 것이 필수적
- 프로젝트 관리 활동
 - 프로젝트 계획 : 개발 전, 범위 결정, 타당성 검토를 통해 개발의 일정, 방법 등을 계획
 - 자원 관리 : 개발하는 데 사용될 자원을 산정하고 할당 및 관리
 - 리스크 관리 : 위험요소를 예측하고 이를 식별, 분석하여 대비책을 세우는 관리 활동
 - 프로젝트 수행과 모니터링 : 프로젝트 작업이 계획에 맞게 진행되고 있는지 확인



프로젝트 관리 삼각형

1.4 소프트웨어 공학의 주제

➤ 소프트웨어 공학의 연구 결과

주제	의미	사례	요리에 비유
방법 (method)	소프트웨어 제작에 사용하는 기법이나 절차	<ul style="list-style-type: none">• 구조적 분석, 설계 방법• 객체지향 분석, 설계 방법	익히는 방법 (구이, 찜, 훈제 등)
도구 (tool)	자동화된 시스템	<ul style="list-style-type: none">• 설계 도구• 프로그래밍 도구• 테스트 도구	요리 도구 (프라이팬, 압력 솥, 오븐 등)
프로세스 (process)	도구와 기법을 사용하여 작업하는 순서	<ul style="list-style-type: none">• Unified Process• eXtreme Programming	조리 순서(레시피)
패러다임 (paradigm)	접근 방법, 스타일	<ul style="list-style-type: none">• 구조적 방법론• 객체지향 방법론	음식 스타일 (한식, 일식, 중식, 퓨전 등)

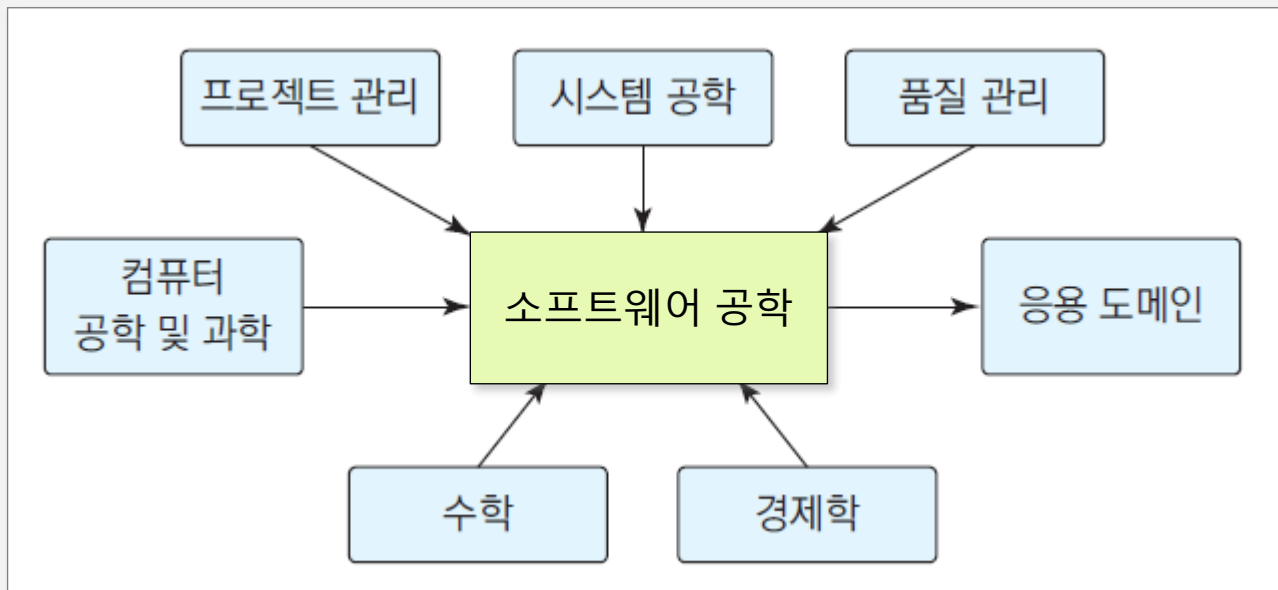
1.4 소프트웨어 공학의 주제

- 소프트웨어 공학 지식 체계(SWEBOK)
 - SWEBOK(software Engineering Body of knowledge)
 - IEEE 산하의 소프트웨어 공학 표준위원회와 ACM이 작업, 소프트웨어 공학을 10개의 주요 지식 영역과 5개의 연관 학문 지식 영역을 정리하고 각 지식 영역과 관련된 활동을 규정한 표준 문서로 ISO/IEC 24773으로 채택됨
- SWEBOK의 10개의 주요 지식 영역
 - SW 엔지니어링 측면
 - SW 요구사항, SW 설계, SW개발, SW테스트, SW유지보수
 - S/W 관리 측면
 - SW형상관리, SW관리, SW프로세스, SW툴&방법론, SW품질

1.5 연관 분야

➤ 소프트웨어 공학의 두 가지 연관 분야

- 컴퓨터 공학의 원리나 기술과 관련된 여러 원리를 다루는 분야
- 이를 적용하여 특정한 문제를 해결하려는 응용 도메인 분야



1.5 연관 분야

- SWEBOOK v3.0은 5가지 영역을 연관 학문 지식 영역으로 추가 정의
 - 소프트웨어 공학 전문가 기량
 - 전문성, 집단 역학과 심리학, 소통 기술
 - 소프트웨어 공학 경제학
 - 기초, 생명주기 경제학, 위험과 불확실성, 경제학 분석 방법론, 실무적 고려사항
 - 컴퓨팅 기반
 - 문제 해결, 추상화, 프로그래밍 기초, 언어 기초, 디버깅 툴, 자료구조와 표현, 알고리즘
 - 수학적 기반
 - 집합, 관계, 함수, 논리 기초, 검증 기초, 계수 기초, 그래프와 트리, 확률 밀도 함수 등
 - 공학적 기반
 - 경험적 기법, 통계 분석, 측정, 공학 설계 등

1.5 연관 분야

➤ 컴퓨터 과학과 소프트웨어 공학

- 컴퓨터 과학

- 계산적 효율, 자원 공유, 정확성, 최적화, 성능을 강조(원리와 기초)
- 자료구조, 데이터베이스, 알고리즘, 운영체제 – 기술적 측면, 즉시 측정 가능

- 소프트웨어 공학

- 생산성, 품질, 비용, 시간을 강조(소프트웨어 개발에 컴퓨터 원리를 문제 해결에 적용)
- 기술적 측면과 기술 외적인 측면 고려 – 방법, 도구, 프로세스 측정에 긴 시간 필요

