

소프트웨어공학

소프트웨어 개발 방법론 1

1 소프트웨어 개발 방법론 개요

2 소프트웨어공학의 패러다임

3 소프트웨어 개발 방법론

소프트웨어 개발 방법론 개요

1. S/W 개발 방법론

- **패러다임**
 - 어떠한 시대 사람들의 견해나 사고를 근본적으로 규정하고 있는 테두리로서의 인식 체계 또는 사물에 대한 이론적인 틀이나 체계
 - 사물을 바라보는 관점, 기본 틀, 접근방법, 스타일 등
- **S/W 공학 paradigm의 정의**
 - S/W 개발 시 고려해야 하는 개발 방법, 개발 환경, 개발 관리 등에 대한 이론적인 체계나 접근 방법

01

소프트웨어 개발 방법론 개요

1. S/W 개발 방법론

S/W 개발 방법	S/W를 어떻게 만들 것인가를 결정하는 기술적인 요소를 제시. 프로젝트에 대한 계획과 추정, 요구사항 분석, 코딩 등 개발 프로젝트 진행 단계에서 요구되는 기법과 수행되어야 할 과제를 포함
S/W 개발 환경	개발 방법론을 지원해 주기 위해 필요한 CASE(Computer-Aided Software Engineering), DBMS 등을 포함. CASE, DBMS 등은 개발 환경을 개선하지만, 논리적인 것을 결정하는 사람을 대체할 수 없음
S/W 개발 관리	S/W 개발 방법과 개발 환경을 묶어 시스템을 효율적으로 적시에 개발할 수 있도록 공정 과정과 절차를 제시. 예) 개발에 필요한 공정 단계, 각 단계별로 요구되는 입력과 결과물 (문서, 보고서, 회의 결과), 품질보증을 위한 검증과 제어 장치 등에 대한 정의 등이 필요.

소프트웨어 개발 방법론 개요

2. 건축공학 VS S/W 공학



소프트웨어공학의 패러다임

1. S/W 개발 방법론

소프트웨어 개발 방법론	<p>소프트웨어 개발, 유지보수 등에 필요한 여러가지 일들의 수행 방법과 이러한 일들을 효율적으로 수행하려는 과정에서 필요한 각종 기법 및 도구를 체계적으로 정리하여 표준화한 것이다.</p>
-------------------------	--



분야	의미	사례	요리 비유
방법 (method)	소프트웨어 제작에 사용하는 기법이나 절차	- 구조적 분석 - 객체지향 분석 - 설계 방법	익히는 방법
도구 (tool)	자동화된 시스템	- 설계 도구 - 프로그래밍 도구 - 테스트 도구	요리 도구
프로세스 (process)	도구와 기법을 사용하여 작업하는 순서	- unified process - eXtreme programming	조리 순서
패러다임 (paradigm)	접근 방법, 스타일	- 구조적 방법론 - 객체지향 방법론	음식 스타일

그림 2-2 네 가지 소프트웨어공학 패러다임

02

소프트웨어공학의 패러다임

2. 시대별 개발 방법론 도입 현황

	1970년대	1980년대	1990년대	2000년대	2010년대
주요기술	Mainframe Computing • 중앙화, 집중화, 메뉴기반	Client/Server Computing • 분산화, 다계층 GUI, 이벤트 기반	Internet/Web Computing • 분산화, 협업적 (Collaborative), 클라이언트 플랫폼 독립적	Ubiquitous Computing • 임베디드 단말기 • 무선/모바일 네트워크 • 웹 서비스 • 지능화된 에이전트	
개발 방법론			CBD 방법론	객체지향 방법론	정보공학 방법론
구조적 방법론					

소프트웨어공학의 패러다임

1. S/W 개발 방법론

CBD방법론 :

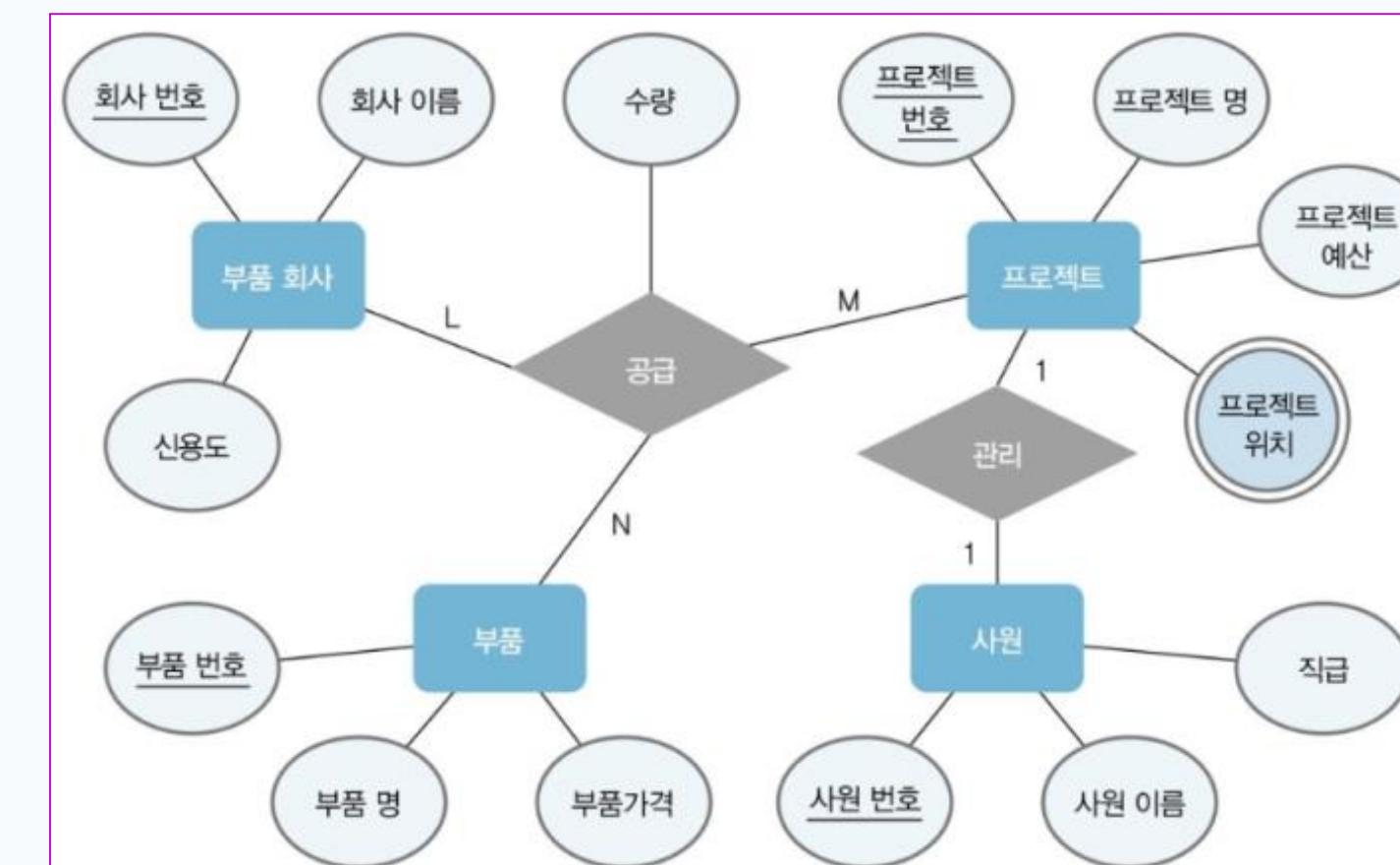
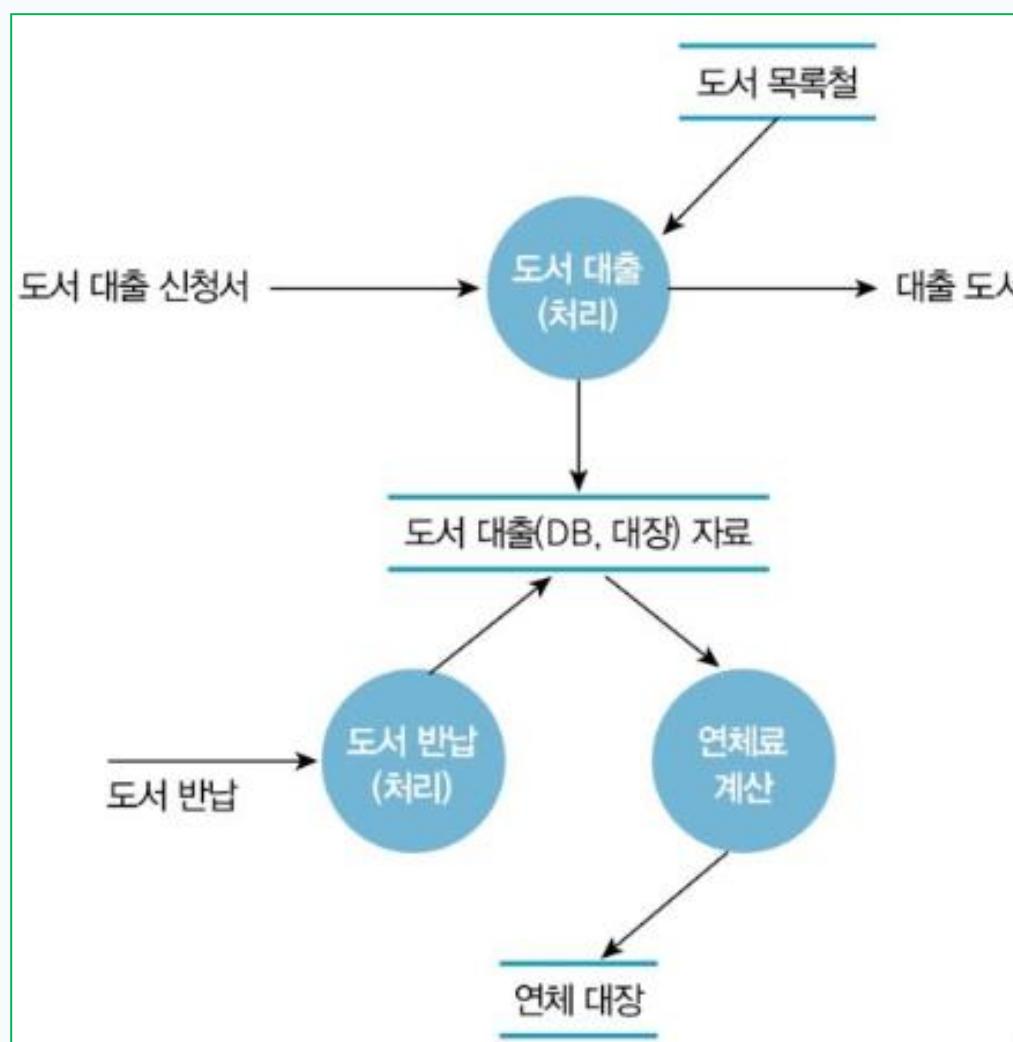
[Component Based Development 방법론]
컴포넌트(재사용이 가능하도록 구현된 요소)
기반으로 개발

구 분	구조적 방법론	정보공학 방법론	객체지향 방법론	CBD 방법론
목 표	비즈니스 프로세스 자동화	경영전략적 정보시스템 구축	재사용 시스템	컴포넌트 개발 및 활용
접근방법	프로세스 중심	데이터 중심	객체중심 (데이터+프로세스)	컴포넌트 중심
특 징	<ul style="list-style-type: none"> - 분할과 정복의 원칙 - 하향식 접근 - 모형적 설계 - 추상화 → 정형화 → 구체화 	<ul style="list-style-type: none"> - 기업 업무시스템 지원 방법론 - Data 모델 중시 - 프로그램 로직은 data 구조에 종속(CRUD) - 전사적 통합 데이터 모델 	<ul style="list-style-type: none"> - 반복적, 점증적 개발 방식 - 데이터와 로직을 통합(객체) - 공학적 접근 - 상속에 의한 재사용 	<ul style="list-style-type: none"> - 객체지향 진화된 형태 - Interface 중시 (구현에 제약 없음) - 인터페이스 구현이 컴포넌트 - Black Box Reuse 지향
산업구조	소품종 다량 생산	다품종 소량생산	인터넷 비지니스	모바일 비즈니스
SDLC 모델	폭포수	폭포수 / 프로토타이핑	반복적 개발	반복적 개발
개발방식 자동화	Top-down 수작업 가능	Top-down 자동화 도구 요구	Bottom-up 자동화 도구 필요	Bottom-up 자동화 도구 필수
모델링	기능 모델링	데이터 모델링 프로세스 모델링	객체 모델링	객체 모델링 컴포넌트 모델링

소프트웨어 개발 방법론

1. S/W 개발 방법론의 종류

구 분	구조적 방법론	정보공학 방법론	객체지향 방법론	CBD 방법론
요구분석 산출물	DFD	ERD	Usecase Diagram	



< DFD >

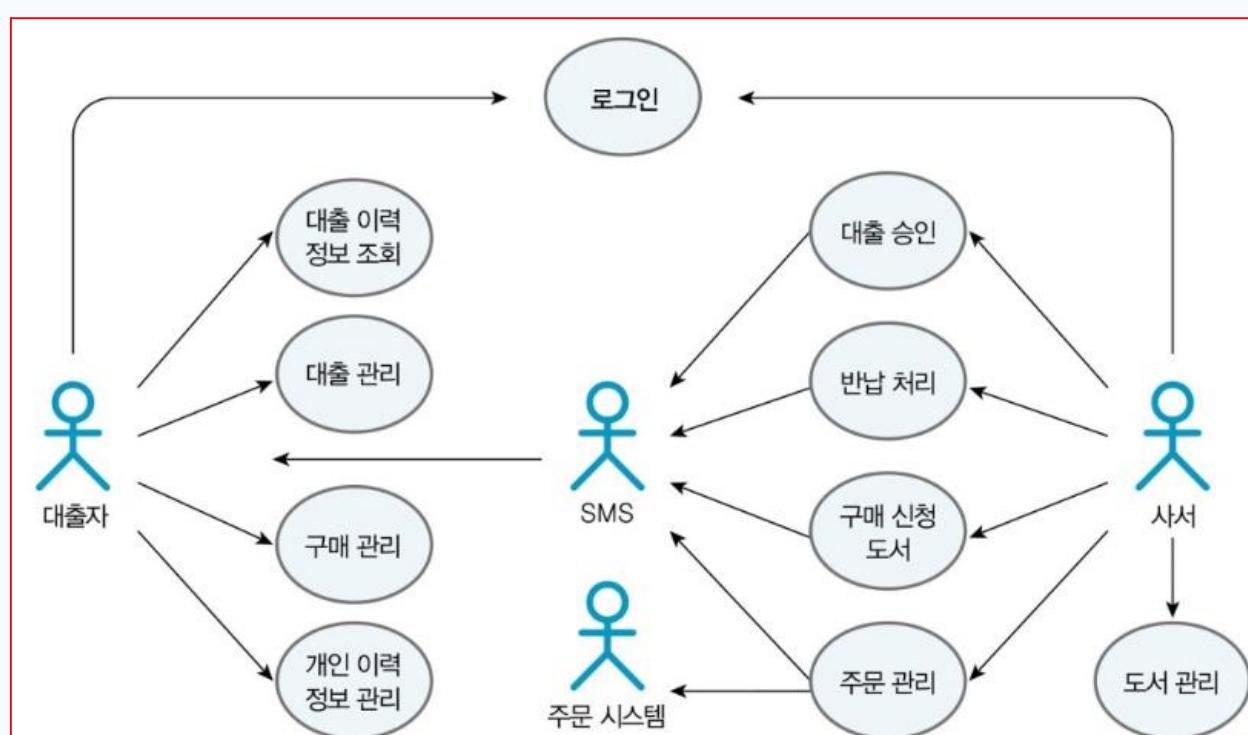
< ERD >

03

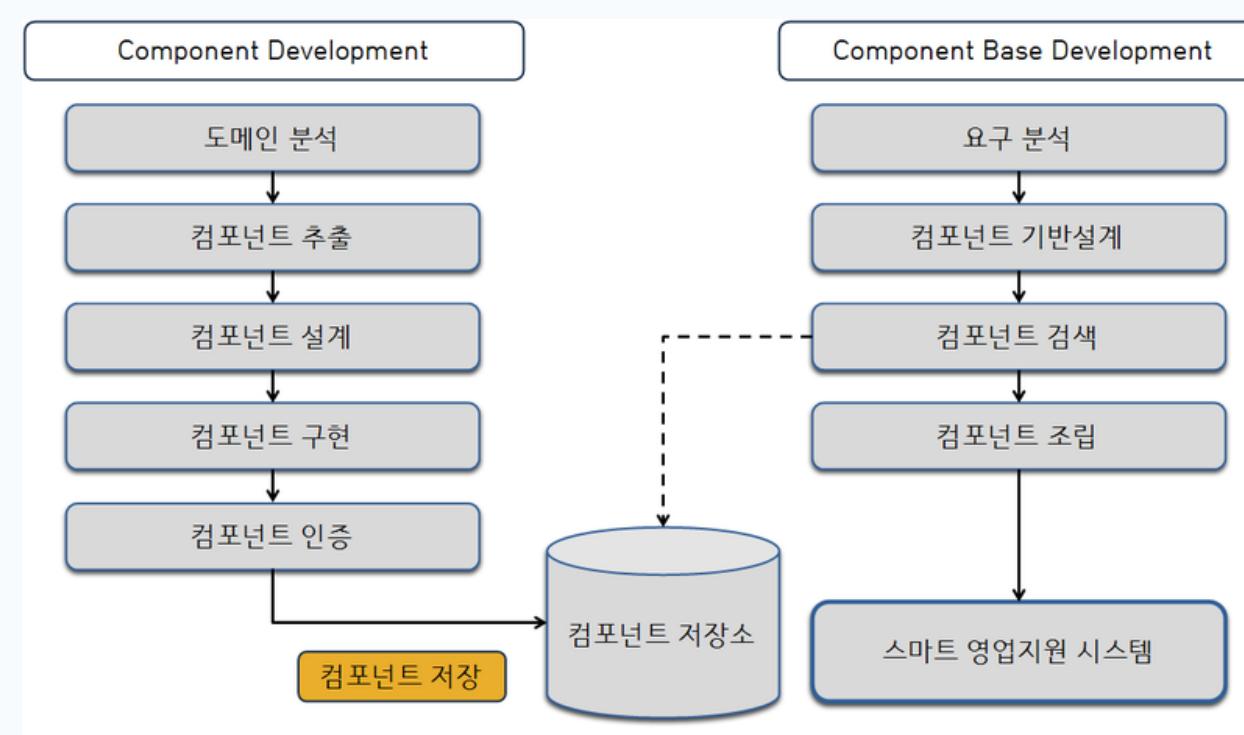
소프트웨어 개발 방법론

1. S/W 개발 방법론의 종류

구 분	구조적 방법론	정보공학 방법론	객체지향 방법론	CBD 방법론
요구분석 산출물	DFD	ERD	Usecase Diagram	



< Usecase Diagram >



< CBD 개발 단계 사례 >

[http://wiki.hash.kr/index.php/CBD_개발방법론] 참조

소프트웨어 개발 방법론

1. S/W 개발 방법론의 종류

구성 요소	내 용	비 고
작업 절차	<ul style="list-style-type: none"> - 프로젝트 수행시 이루어지는 작업단계의 체계 - 단계별 활동, 활동별 세부 작업 나열, 활동의 순서 명시 	작업순서 명시
작업 방법	<ul style="list-style-type: none"> - 각 단계별 수행해야 하는 작업 - 절차/작업 방법(누가, 언제, 무엇을 작업하는지 가술) 	작업 절차
산출물	<ul style="list-style-type: none"> - 각 단계별 만들어야 하는 산출물의 목록 및 양식 	
관 리	<ul style="list-style-type: none"> - 프로젝트의 진행 기록 - 계획 수립, 진행 관리, 품질, 외주, 예산, 인력 관리 등을 기록 	일정 및 위험관리
기 법	<ul style="list-style-type: none"> - 각 단계별 작업 수행시 기술 및 기법의 설명 	
도 구	<ul style="list-style-type: none"> - 각 기법별 지원 도구에 대한 구체적인 사용 표준 및 방법 	CASE, Tool등