

소프트웨어공학 활용



소프트웨어 개발 방법론



한국기술교육대학교
온라인평생교육원

학습내용

- 개발 방법론 진화 과정
- 개발 방법론 종류 및 선택 방법

학습목표

- 소프트웨어 개발 방법론 구성 절차를 설명할 수 있다.
- 소프트웨어 개발 방법론 종류를 이해하고 설명할 수 있다.

개발 방법론 진화 과정



1 개발 방법론 개요

1 정의

1

소프트웨어 공학 원리를 소프트웨어 개발 생명주기에 적용함



소프트웨어 개발 방법론 = 소프트웨어 공학 + 방법론

2

정보 시스템을 개발하기 위한 작업 활동, 절차, 산출물, 기법 등을 정리

3

품질을 높이고 납기일을 준수하여 고객의 만족도를 높이기 위한 효율적인 개발방법

2 출현 배경

소프트웨어의 수요가 규모와 복잡성으로 인해 기하급수적으로 증가하면서 소프트웨어 위기 발생



위기에 대한 인식에 따른 소프트웨어 공학의 발전

소프트웨어 프로젝트가 대형화되면서 개발 기간의 장기화



기간, 예산 및 품질상의 복합적인 문제가 발생

소프트웨어 이용범위 확대

정보시스템에 대한 이해확대, 실무자뿐만 아니라 관련된 사람들(관리자, 경영자)까지 이용

개발 방법론 진화 과정



1 개발 방법론 개요

3 진화 과정

1960년대

1970년대

1980년대

1990년대

2000년대

현재

- 컴퓨터는 대용량 계산을 위한 수단으로 쓰임
- 프로그래밍의 사용 폭이 좁고, 요구사항도 단순함
- 군 분야에서 주로 사용
- 기계어를 사용하였으며, 개발 방법론이 없음

1960년대

1970년대

1980년대

1990년대

2000년대

현재

- 컴퓨터가 점차 전문 영역으로 확대
- 고급언어를 사용하기 시작하였으며 고객의 요구사항 발생
- 군 분야에서 점차 민간 영역으로 확대
- 1968년 소프트웨어 위기가 언급되고 폭포수 방법론 탄생

1960년대

1970년대

1980년대

1990년대

2000년대

현재

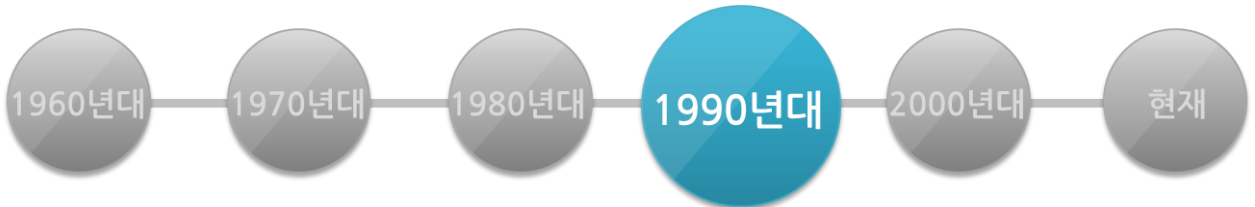
- 컴퓨터의 대중화 시작
- C 언어 탄생, 4세대 언어 툴이 생김
- 고객의 요구사항이 점차 복잡해짐
- SDLC(소프트웨어 개발 생명주기) 개발 방법론 대두
- 소프트웨어가 사업으로 인식되어 맞춤형 소프트웨어와 하드웨어 분야 발전

개발 방법론 진화 과정

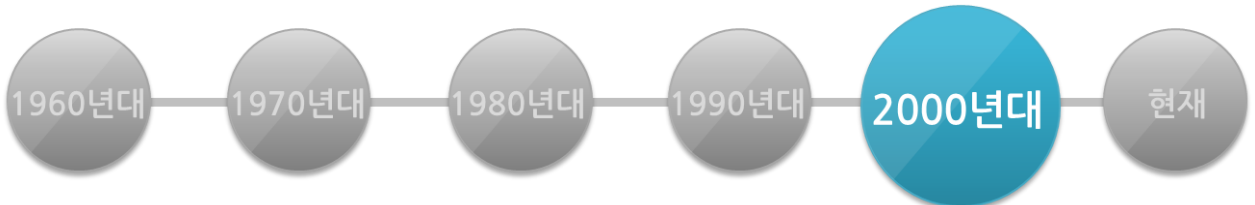


1 개발 방법론 개요

3 진화 과정



- 소프트웨어가 하드웨어보다 높은 부가가치를 창출하기 시작
- 기업들의 많은 투자 발생
- 대형 프로젝트들의 증가로 효율성과 생산성의 중요성 부각
- 개발 방법론은 위험관리와 고객 요구사항의 적합성, 생산성을 고려
- 객체지향 언어(JAVA, C++)의 전성기
- 반복과 통합 프로세스가 탄생하며 이를 토대로 RUP가 탄생
- 컴포넌트 기반 방법론(CBD)의 기반이 됨



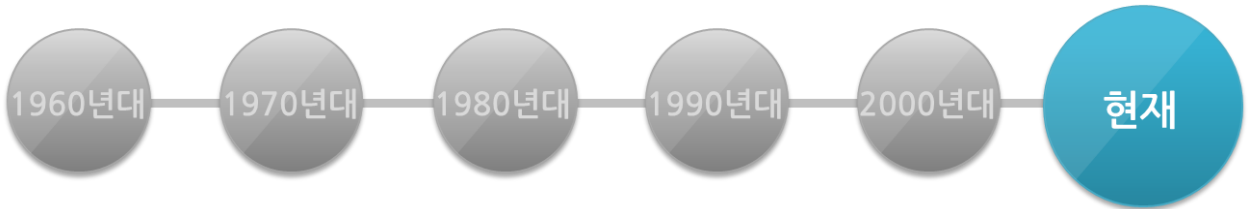
- 인터넷의 발전으로 웹 애플리케이션 중심으로 변화
- 고객과 서비스 중심으로 이동
- 고객의 요구사항을 예측하고 빠른 시간 안에 구축해야 하는 과제로 개발은 점차 생산성 중심으로 변화
- 쉬운 개발을 선호함으로써 자동화 툴 발전

개발 방법론 진화 과정



1 개발 방법론 개요

3 진화 과정



- 고객의 요구사항 + 개발자의 편의성을 고려(자동화 툴)
- 개발은 점차 기업, 그룹화에서 개인화로 변화
- 툴의 발전으로 다양한 개발 언어를 하나의 플랫폼상에서 동일하게 작동
- 계획부터 테스트까지 개발자 혼자 개발할 수 있는 시기

4 특징

1

개발단계를 정의하고 활동, 제품, 검증절차, 각 단계의 종결 기준 등을 상세히 서술

2

소프트웨어 개발에 관한 계획, 분석, 설계 및 구축에 대하여 정형화된 방법과 절차, 도구 등을 공학적인 기법으로 정리

3

소프트웨어 개발 방법 및 절차, 도구 등이 실무적인 관점에서 하나의 체계로 묶임

개발 방법론 진화 과정



1 개발 방법론 개요

5 필요성

개발 경험을 축적 및 재활용을 통한 개발 생산성 향상
(작업의 표준화/모듈화)

효과적인 프로젝트 관리(가시화)

정형화된 절차와 표준용어를 제공하여 의사소통 수단 사용

단계별 검증과 종결승인을 통한 일정 수준의 품질 보증

개발 방법론 진화 과정



2 개발 방법론 구성

1 작업 절차(Process)

1

프로젝트 수행 시 이루어지는 작업단계의 체계

2

단계별 활동, 활동별 세부작업 열거, 활동의 순서 명시

3

단계 - 활동 - 작업

2 작업 방법(Method)

각각의 단계에서 수행하는 구체적이고 명확한 설명

절차/작업방법(누가, 무엇을 언제 해야 한다는 내용으로 진행)

3 산출물

각 단계별로 만들어야 하는 산출물의 목록 및 양식(설계서)

매뉴얼/설정 방법 등

개발 방법론 진화 과정



2 개발 방법론 구성

4 기법과 도구

기법

각 단계별로 작업 수행 시 소요되는 기술 및 기법
설명(객체지향 모델링 기법)

도구

기법에 사용되는 지원 도구 혹은 사용표준 및
방법(CASE 등)

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

1 구조적 방법론

1 정의 및 개념



구조적
방법론이란?

전체 시스템을 기능에 따라 분할하여
개발하고, 이를 통합하는 분할과 정복
접근 방식의 방법론

업무 활동
중심의
방법론

프로세스
중심의 하향식
방법론

1970년대
기능 중심의
방법론

대표
방법론 :
폭포수

2 특징

1

전체 시스템을 기능에 따라 분할하고 이에 따른 개발과 통합하는
방식

2

순차, 선택, 반복으로 프로그램 로직을 처리하는 방식

3

정형화된 분석 절차에 따라 요구사항을 파악하고 도형 중심의
다이아그램을 이용하여 문서화

4

컨트롤 가능한 모듈로 구조화

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

1 구조적 방법론

3 장점과 단점

장점

- 일괄처리 방식인 자료변환을 중심으로 하는 응용소프트웨어개발에 적합(배치프로그램 등)
- 소프트웨어 및 시스템 개발의 전형적인 접근법(기능은 사용자의 1차적 요구사항)

단점

- 기능의 불안정요소
- 데이터 정보은닉 불가
- 재사용 보수성 낮음
- 시스템 분석/설계/개발 구분이 명확하여 단계별 전환 필요

4 원리

추상화

문제를 이해하고 표현하기 위해 개념화시켜 표현

구조화

기능을 계층적으로 분류

단계적 상세화

하향식으로 진행하면서 점차 내용을 구체화

모듈화

하나의 시스템을 서브 시스템, 프로그램, 모듈 등으로 구분하여 정의하고 개별적으로 설계(Divide & Conquer)

정보 은닉

각 모듈은 다른 모듈에 독립적이며 하나의 모듈 변경이 다른 모듈의 세부내용에 영향을 미치지 않음

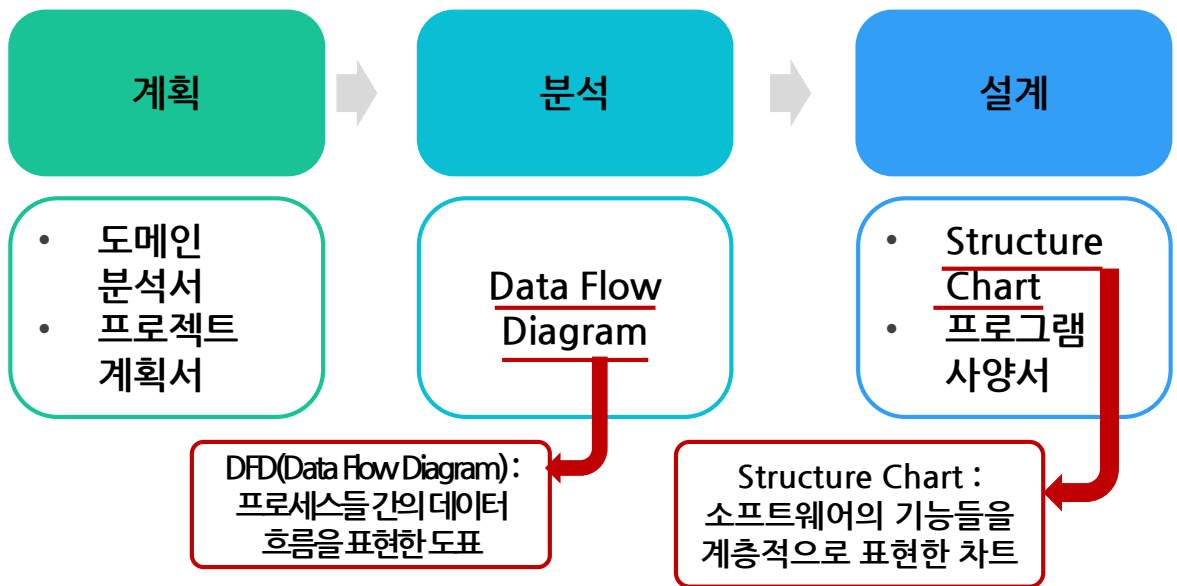
개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

1 구조적 방법론

5 단계별 산출물



6 지원 CASE 툴과 주요 지원 언어



주요 지원 언어 : COBOL, C, VB, PASCAL

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

1 구조적 방법론

7 구성 요소

구조적 분석

구조적 설계

구조적 프로그래밍

Data Dictionary :
데이터의 상세 사항이 기술

- 도형 중심 : DFD(Data Flow Diagram), DD(Data Dictionary), Mini-Spec(소단위명세서) 이용
- 정형화된 분석 절차, 사용자 요구파악, 문서화하는 체계적 기법
- 기본원칙 : 분할과 정복, 추상화, 정형화, 구조적 조직화, 하향식기능 분해

소단위 명세서 : DFD에 표시된
처리에 대한 상세한 절차를 표현한
문서

구조적 분석

구조적 설계

구조적 프로그래밍

구조적 언어

- 소프트웨어 기능과 프로그램 구조, 모듈을 설계 전략, 평가 지침, 문서화 도구를 지원하는 체계적 설계 기법
- Flow Chart, HIPO(Hierarchical Input Process Output) Chart, N-S(Nassi-Schneiderman) Chart, 프로그램 명세서 이용
- 기본원칙 : 복합 설계의 기본 원칙(결합도, 응집도)

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

1 구조적 방법론

7 구성 요소

구조적 분석

구조적 설계

구조적
프로그래밍

구조적 언어

- Dijkstra에 의해 정형화
- 계층적 형식, 제한된 제어 구조, 작성 순서대로 프로그램 실행
- 연속(Sequence) 구조
- 선택(Selection or IF - Then - Else) 구조
- 반복(Repetition) 구조

구조적 분석

구조적 설계

구조적 프로그래밍

구조적 언어

- Structured COBOL
- Fortran 77
- PL/1
- Pascal

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

2 정보공학 방법론

1 정의 및 개념



정보공학
방법론이란?

기업 전체 또는 주요 부분을 대상으로
정보시스템 계획 수립, 분석, 설계 구축에
정형화된 기법들을 상합, 적용하는
데이터 중심 방법론

데이터 중심
방법론

1980년대
자료구조
중심의
방법론

구조적
방법론

문제점 극복을
위해 등장



대표 방법론 : 프로토타입, 정보공학방법론

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

2 정보공학 방법론

2 특징

1 기업 중심으로 정보전략 계획(ISP) 포함

2 데이터 중심의 분석과 설계를 진행

3 도형 중심의 산출물

4 프로젝트를 관리 가능한 단위로 분할, 정복

5 공학적 접근방식을 사용함

6 적극적인 사용자 참여를 유도함

7 정보시스템 개발의 자동화 지향

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

2 정보공학 방법론

3 장점과 단점

장점

- 경쟁우위 확보의 전략적 기회 식별 및 방안 제공
- 일관성 있고 통일된 정보시스템 구축 가능
- 시스템의 장기적인 진화, 발전 허용
- 데이터 중심으로 업무절차 및 환경변화에 유연

단점

- 정보공학의 효과를 위해 장기간 필요
- 소규모의 자동화 요구 사업영역에는 시간이 오래 걸림
- 특정 사업영역으로부터 독립된 시스템 개발에는 부적합

4 원리 및 구성요소

- 정보전략계획(ISP - Information Strategy Planning)

경영전략 분석

기업의 내/외부 환경 분석 및 기업 비전, 전략 도출

현행 업무 프로세스 분석

문제점 및 개선방안 도출

현 시스템 분석, 평가

현 시스템 문제점 도출 평가 후 개선 방안

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

2 정보공학 방법론

4 원리 및 구성요소

- 정보전략계획(ISP - Information Strategy Planning)

아키텍처 개발

후속 작업 지원을 위해 프로세스, 데이터 모델 개발

전략계획

프로젝트 정의 및 우선순위 부여

- 업무영역분석(BAA - Business Area Analysis) :
ISP 과정에서 수집된 정보를 이용하여 세부적으로 확장

데이터 모델
다이아그램

ISP 과정에서 만들어진 ERD를 상세하게 확장한
다이아그램

ERD(Entity-Relationship
Diagram) : 데이터 모델과
제약 조건들을 표현한 도표

프로세스 분할
다이아그램(PHD)

업무영역 내의 기능들을 프로세스들로 분할하여
트리구조의 분할도를 만들

프로세스의존
다이아그램(PDD)

서로 간의 다른 프로세서들 간의 의존관계를
나타냄

프로세스/데이터
매트릭스

프로세스와 데이터 간에 일어나는 행위를
매트릭스로 보여줌

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

2 정보공학 방법론

4 원리 및 구성요소

- 업무시스템설계(BSD) : 업무절차 정의, Presentation 설계, 분산 설계

엔티티 관계 다이어그램 (Entity-Relation Diagram)

- 정보전략계획(ISP)에서부터 시스템 구축까지 개발 전 과정에 걸쳐 사용하는 다이어그램
- 추상화된 데이터를 간의 함수적 관계를 그림으로 나타냄

분할 다이어그램 (Decomposition Diagram)

- 정보공학 피라미드 최상위 단계
- 전략수립 단계에서부터 하위단계로 진행해 나가면서 나타낸 기능 분할도, 프로세스 분할도, 프로시저 분할도 등

액션 다이어그램 (Action Diagram)

- CASE Tool을 사용해 분할 다이어그램을 액션 다이어그램으로 변환
- 향후 실행 가능한 코드로 전환됨

의존 다이어그램 (Dependency Diagram)

- 분할 다이어그램의 약점(프로세스 간의 우선순위를 나타낼 수 없음)을 보완한 다이어그램
- 프로세스 상호 간의 연관을 나타냄

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

2 정보공학 방법론

4 원리 및 구성요소

- 업무시스템설계(BSD) : 업무절차 정의, Presentation 설계, 분산 설계

데이터 흐름도 (Data Flow Diagram)

- 프로시저 의존도의 특별한 형태
- 각 프로시저에서 사용되는 입출력 데이터의 흐름을 각 데이터의 입출력과 함께 그림으로 나타냄

결정 트리(Decision Tree)

- 프로그램 로직의 분기점과 분기 조건 및 결과를 기술하기 위함

대화구조(Dialogue Structure)의 표현

- 계층적인 메뉴: 컴퓨터와 사용자 간의 대화방식을 기술하기 위해 액션 다이어그램을 사용
- 수평적인 대화: 팝업 윈도우를 나타내기 위해서 수평적인 대화 흐름도를 사용

자료구조 다이어그램 (Data Structure Diagram)

- 업무영역분석(BAA) 단계에서 만들어진 데이터 모델을 해당 DBM에 맞도록 Diagram을 생성
- 예) RDB, HDB 등

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

2 정보공학 방법론

4 원리 및 구성요소

- 기술설계 및 구축(SC - System Construction)

데이터 사용 분석

물리적 데이터베이스
설계

분산분석

- Data Usage Analysis
- 트랜잭션의 발생량을 토대로 부하를 최적 분산하기 위해 응용 프로그램별 발생량을 액션 다이어그램에 주석으로 표기하거나 ERD에서 각 경로에 대한 관계비를 숫자로 표현

데이터 사용 분석

물리적 데이터베이스 설계

분산분석

- 데이터베이스 설계자가 시스템의 비용, 성능, 응답시간 등을 고려하여 복잡한 시스템이 서로 균형을 이루면서 동작할 수 있도록 최적의 답을 찾아 설계

데이터 사용 분석

물리적 데이터베이스
설계

분산분석

- 데이터와 프로세스를 여러 곳의 서버에 분산시켜 부하를 평준화시키려는 방법
- 지역, 프로세스, 데이터를 매트릭스로 구성하여 분석

개발 방법론 종류 및 선택 방법

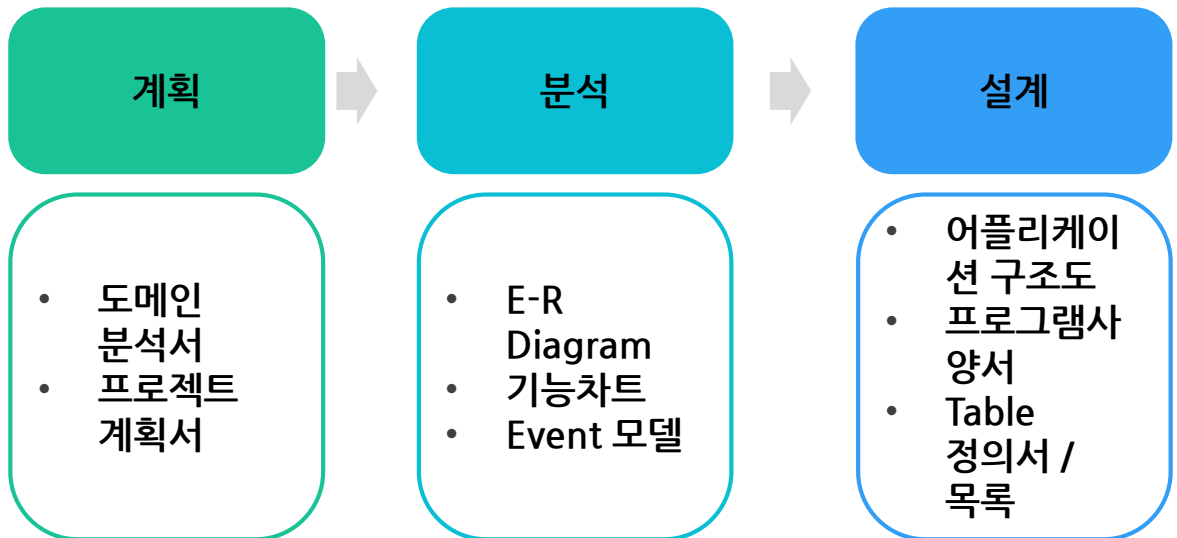


1 개발 방법론 종류

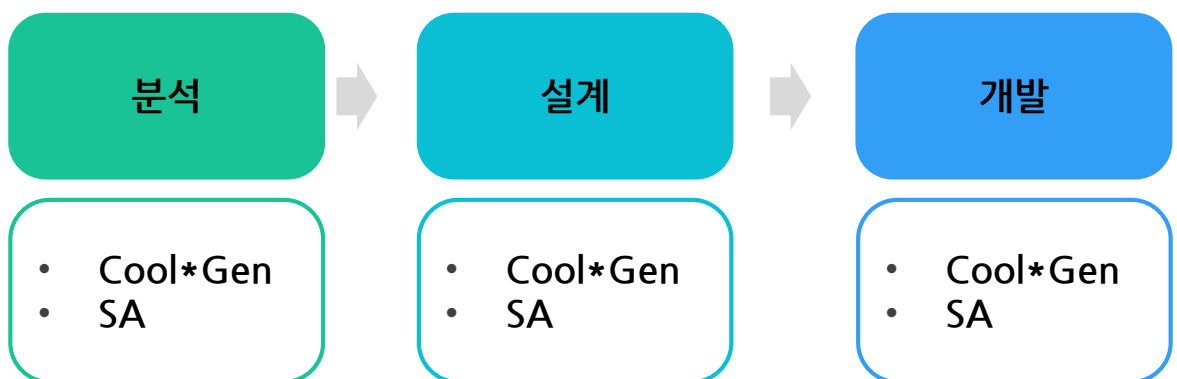
2 정보공학 방법론

4 원리 및 구성요소

▪ 단계별 산출물



▪ 지원 CASE 툴과 주요 지원 언어



주요 지원 언어 : COBOL, C, PASCAL

개발 방법론 종류 및 선택 방법

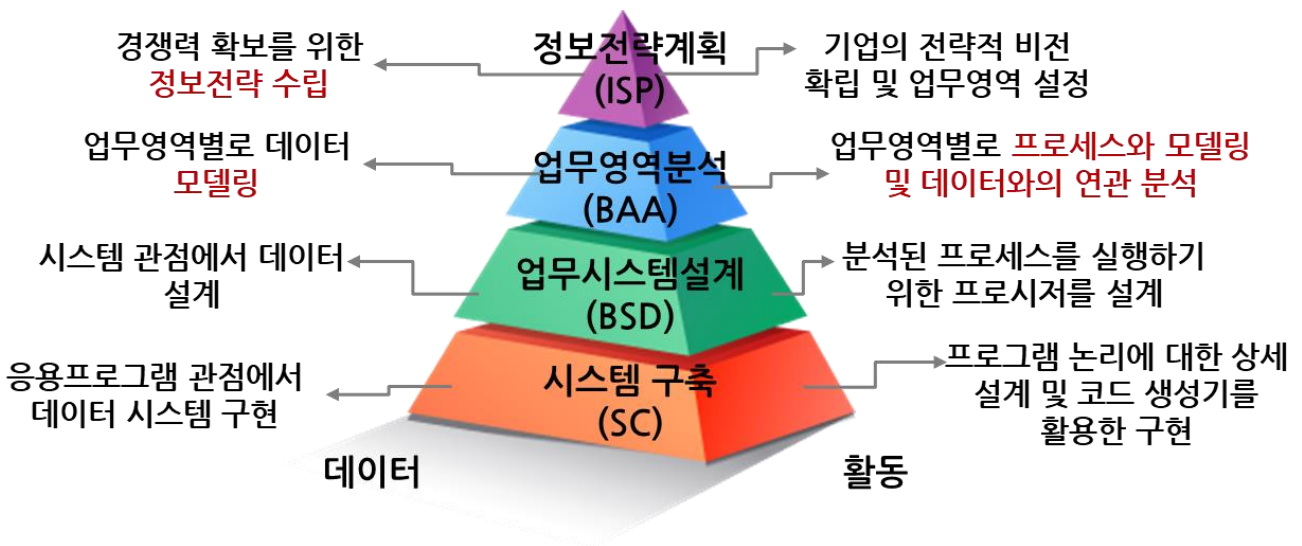


1 개발 방법론 종류

2 정보공학 방법론

5 정보공학의 목적

- 생산성이 높은 System을 품질이 좋고, 유지보수가 쉽게



- 정보 공학 방법론 수행 단계



개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

3 객체지향 방법론

1 정의 및 개념

요구분석, 설계, 구현, 시험 등의 소프트웨어 생명주기에 객체지향 개념을 접목해 일관된 모델을 가지고 소프트웨어를 개발하는 방법론

객체, 클래스 및 이들 간의 관계를 식별하여 설계모델로 변환하는 방법론

실 세계의 문제 영역에 대한 표현을 소프트웨어 해결영역으로 접목하는 방법으로 객체 간에 메시지를 주고받는 형태로 시스템 구성

1990년대 객체
중심 개발론

대표 방법론 :
반복적 개발

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

3 객체지향 방법론

2 특징

모형의 적합성

현실 세계 및 인간의 사고방식과 유사

모형의 일관성

공정별 단계 전환이 자연스럽게 신속함

모형의 추적성

공정별 추적이 편리

고도의 모듈화

상속에 의한 재사용

3 원리

업무요건 정의

요구사항 수집 → 업무 요건 정의

객체지향 분석

객체모델링 → 동적 모델링 → 기능 모델링

객체지향 설계

시스템설계 → 객체 설계 → 구현

테스트/배포

테스트 → 패키지 → 프로젝트 평가

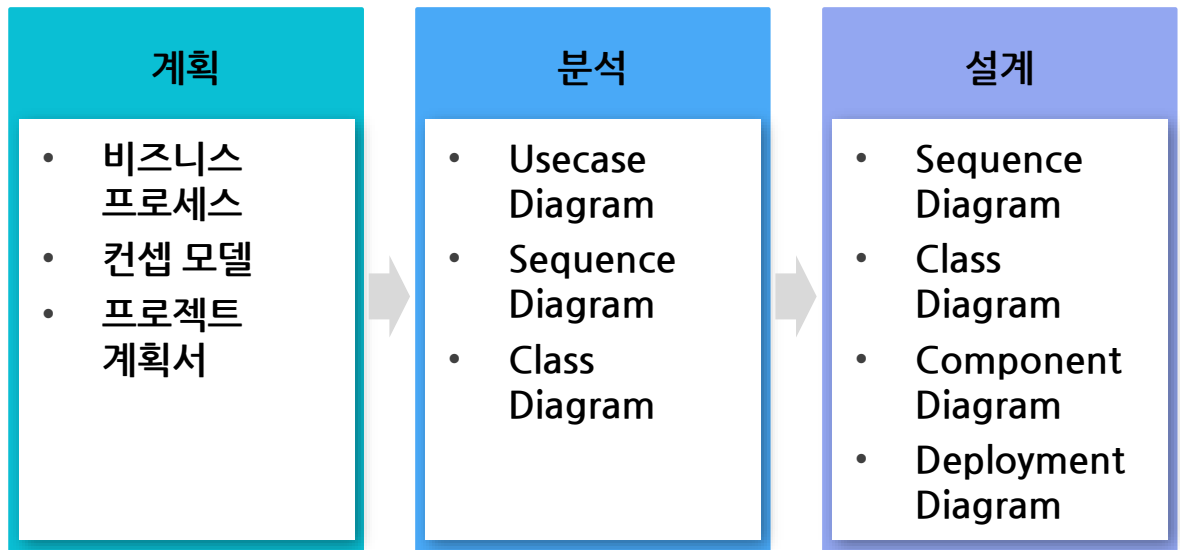
개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

3 객체지향 방법론

4 단계별 산출물



5 장점과 단점



개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

3 객체지향 방법론

6 지원 CASE 툴과 주요 지원 언어

분석

• Rose, SA, Plastic

설계

• Rose, SA, Plastic

개발

• Rose, SA, Plastic



C++, JAVA, C#, VB

7 절차와 단계별 작업 항목

요건정의	객체지향 분석	객체지향설계/구현	테스트/배포
업무요건 정의	객체 모델링 ↓ 동적 모델링 ↓ 기능 모델링	구현 ↑ 객체 설계 ↑ 시스템 설계	테스트 ↓ 패키지 ↓ 프로젝트 평가

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

3 객체지향 방법론

7 절차와 단계별 작업 항목

▪ 객체 지향 분석

객체 모델링

- 시스템 정적 구조 포착
- 추상화, 분류화, 일반화, 집단화

동적 모델링

- 시간의 흐름에 따라 객체 사이의 변화를 조사
- 상태, 사건, 동작

기능 모델링

- 입력에 대한 처리 결과에 대한 확인

▪ 객체 지향 설계와 객체 지향 구현

시스템 설계

- 시스템구조를 서브 시스템으로 분해
- 성능 최적화 방안, 자원 분배 방안

객체 설계

- 상세 내역을 모형으로 개발의 상세화
- 구체적인 자료구조와 알고리즘 구현

객체 지향 구현

- 객체 지향 언어(객체, 클래스)로 상속 지원
- C++, JAVA

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

3 객체지향 방법론

8 객체 지향 방법론 종류

OOSE (Jacobson)

OMT (Rumbaugh)

Booch

- Object Oriented SW Engineering
- Usecase에 의한 접근 방법으로 Usecase를 모든 모델의 근간으로 활용
- 분석, 설계, 구현 단계로 구성
- 기능적 요구사항 중심의 시스템

OOSE (Jacobson)

OMT (Rumbaugh)

Booch

- Object Modeling Technology
- 객체 지향 분석, 시스템 설계, 오브젝트 설계/구현의 4단계로 구성
- 객체 모델링 : 시스템의 정적 구조 표현(산출물 : 객체도(Object Diagram))
- 동적 모델링 : 객체의 제어 흐름/상호반응 표현
- 기능 모델링 : 데이터값의 변화 과정 표현
- 복잡한 대형 프로젝트에 유용
- 기업 업무의 모델링 편리 및 사용자와 의사소통 편리

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

3 객체지향 방법론

8 객체 지향 방법론 종류

OOSE
(Jacobson)

OMT
(Rumbaugh)

Booch

- OOD(Object Oriented Design)로 Design 부분만 존재
- 설계 문서화를 강조하여 다이어그램 중심으로 개발하는 방법론
- 분석과 설계 분리 안 됨
- 분석하는데 이용된 객체 모델의 설계 시 적용

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

4 컴포넌트 기반 방법론(CBD)

1 정의 및 개념



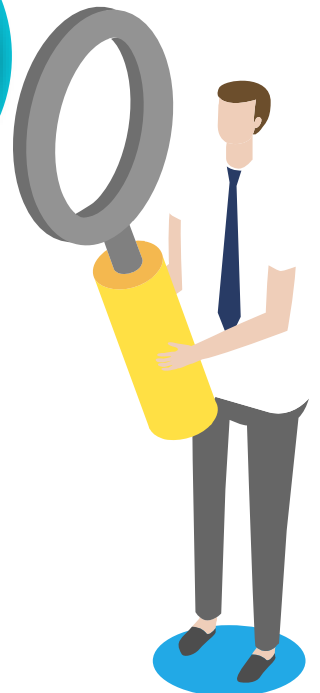
컴포넌트 기반
방법론이란?

- 재사용 가능한 컴포넌트의 개발 또는 컴포넌트들을 조합하여 어플리케이션 개발 생산성, 품질을 높이고 시스템 유지보수 비용을 최소화할 수 있는 혁신 개발방법론
- 컴포넌트 기반의 소프트웨어 개발이라는 뜻

소프트웨어를 개발할 때 기존에 존재하거나 새로 만든 컴포넌트들을 잘 조합함으로써 하나의 새로운 소프트웨어를 만드는 개발 방법

2000년대
컴포넌트 중심
개발론

대표 방법론 :
RUP, XP, Agile,
마르미



개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

4 컴포넌트 기반 방법론(CBD)

2 특징

1

객체개발방법론의 진화된 형태

2

Interface 중시

3

컴포넌트 기반, 재사용성 중시, 반복과 통합 중시

4

생산성 : 부품의 조립을 통한 개발 시간 단축

5

변경용이성 : 독립적인 컴포넌트 단위의 관리로 복잡성을 최소화

6

재사용성 : 실행기반의 재사용, 모델의 재사용

7

고품질 : 품질이 검증된 컴포넌트 사용

8

Black Box Resuse 지향

개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

4 컴포넌트 기반 방법론(CBD)

3 장점과 단점

장점

- 생산성, 품질, 비용 위험개선

단점

- 컴포넌트 유통환경 개선 필요
- 컴포넌트 평가
- 인증 환경 미흡

4 원리

요구분석

개발 시스템에 대한 요구사항 분석

분석

아키텍츠 정의, Usecase 모델링

설계

Rose, SAUI 설계, 컴포넌트 정의/설계, DB, 컨버전, 테스트 설계, Plastic

분석

코딩, 테스트

구현

릴리즈, 교육

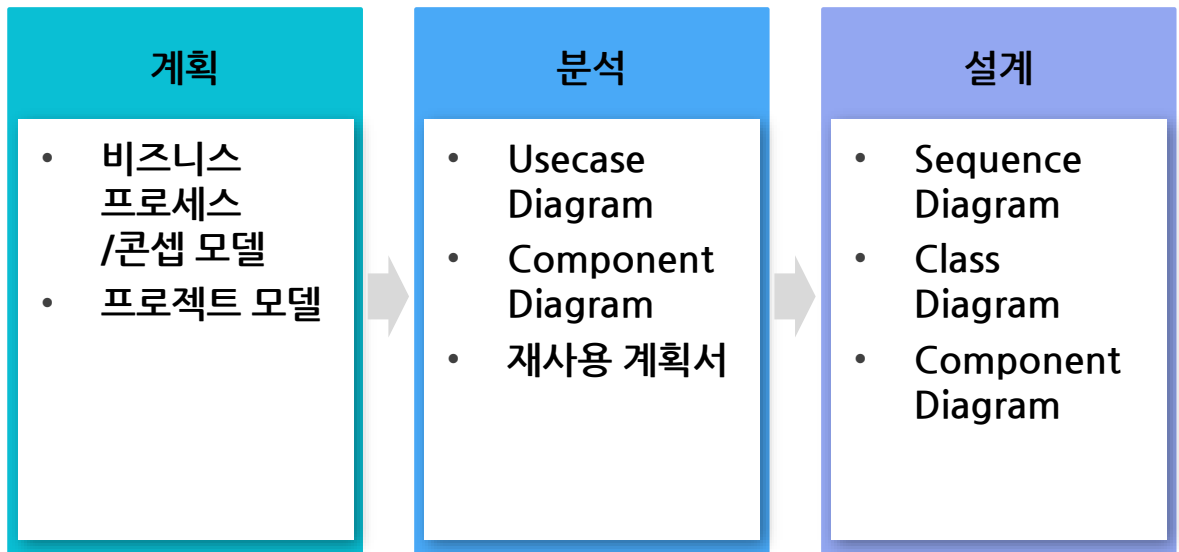
개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

4 컴포넌트 기반 방법론(CBD)

5 단계별 산출물



6 지원 CASE 툴



개발 방법론 종류 및 선택 방법



1 개발 방법론 종류

4 컴포넌트 기반 방법론(CBD)

7 분류

요소
컴포넌트

더 이상 나누어질 수 없는 최소단위의 컴포넌트

기능
컴포넌트

하나 이상의 요소 컴포넌트가 결합하여
하나의 기능 구현

서비스
컴포넌트

하나의 사용자 서비스를 수행하는 컴포넌트

어플리케이션
컴포넌트

여러 서비스를 수행하는 시스템 컴포넌트

8 구성

요구분석

AS-IS 모델링, TO-BE 모델링, 요구사항
정의

분석

아키텍처정의, Usecase 모델링, UI
프로토타이핑

설계

UI 설계, 컴포넌트 정의/설계

개발

코딩, 테스트

구현

릴리즈, 교육

개발 방법론 종류 및 선택 방법



2 개발 방법론 선택

1 선택 기준

1

프로젝트 환경 고려(응용 분야, 시스템 규모, 복잡도, 성격 등)

2

수작업을 최소화하고 자동화되어 있을수록 좋음(시간과 비용)

3

개발자들이 공감하고 적절히 이용할 수 있는 방법론
(방법과 도구, 경험)

2 선택 시 문제점과 개선 대책

문제점

- 프로젝트 특성을 무시한 특정 방법론 강요
- 형식적인 적용에 그쳐 필요 없는 문서만 양산
- 소규모 프로젝트에 방대한 규모의 방법론 적용

개선 대책

- 기업 차원의 품질관리 인식 제고 및 교육과 효과적 활용 도모
- 융통성 있게 개발 방법론 적용, CMM, SPICE 연계

학습정리

1. 개발 방법론 진화 과정



- 소프트웨어 개발 방법론은 소프트웨어 공학과 방법론의 이론이 합쳐진 것
- 정보 시스템을 개발하기 위한 작업 활동, 절차, 산출물, 기법 등을 정리한 것
- 소프트웨어의 위기 이후 소프트웨어의 이용범위가 확대되고 대형 프로젝트가 생기면서 이를 극복한 방법론이 등장
- 1960년대에 주로 군 분야에서 사용
- 1970년대에 민간영역으로 확대, 폭포수 방법론 탄생
- 1980년대에 SDLC 개발 방법론이 대두
- 1990년대에 객체지향 언어의 전성기가 도래하면서, 반복과 통합 프로세스가 탄생
- 2000년대에 인터넷의 발전으로 웹 어플리케이션 중심으로 변화, 자동화 툴이 발전
- 현재는 다양한 자동화 툴의 발전으로 계획부터 테스트까지 혼자 개발할 수 있음
- 작업 절차(Process), 작업방법(Method), 단계별 산출물, 단계별 작업 기법, 도구로 개발방법론이 구성

학습정리

2. 개발 방법론 종류 및 선택 방법



- 구조적 방법론은 1970년대 기능 중심, 프로세스 중심의 하향식 방법론으로 전체 시스템을 기능에 따라 분할하여 개발하고, 이를 통합하는 분할과 정복 접근 방식의 방법론
- 구조적 방법론의 구성요소는 구조적 분석, 구조적 설계, 구조적 프로그래밍, 구조적 언어로 구성
- 정보공학 방법론은 1980년대 자료구조 중심의 방법론으로 정보시스템 계획 수립, 설계 구축에 정형화된 기법들을 상합, 적용하는 데이터 중심 방법론
- 정보공학 방법론은 정보전략계획(ISP), 업무영역분석(BAA), 업무시스템설계(BSD), 기술설계 및 구축(SC)단계로 구성
- 객체지향 방법론은 1990년대 객체 중심 개발론으로 객체, 클래스 및 이들 간의 관계를 식별하여 설계모델로 변환하는 방법론
- 객체지향 방법론의 작업 진행 단계
 - 요건정의 → 객체지향분석 → 객체지향 설계/구현 → 테스트/배포
- 컴포넌트 기반 방법론은 2000년대 컴포넌트 중심 개발론으로, 재사용 가능한 컴포넌트의 개발 또는 컴포넌트들을 조합하여 어플리케이션 개발을 통하여 생산성 품질을 높이고 유지보수 비용을 최소화할 수 있는 개발 방법론
- 개발 방법론을 선택할 때 개발자들이 동의할 수 있는 방법론과 프로젝트의 여러 환경(구축 기간, 비용, 고객의 요구사항)을 고려하여 선택