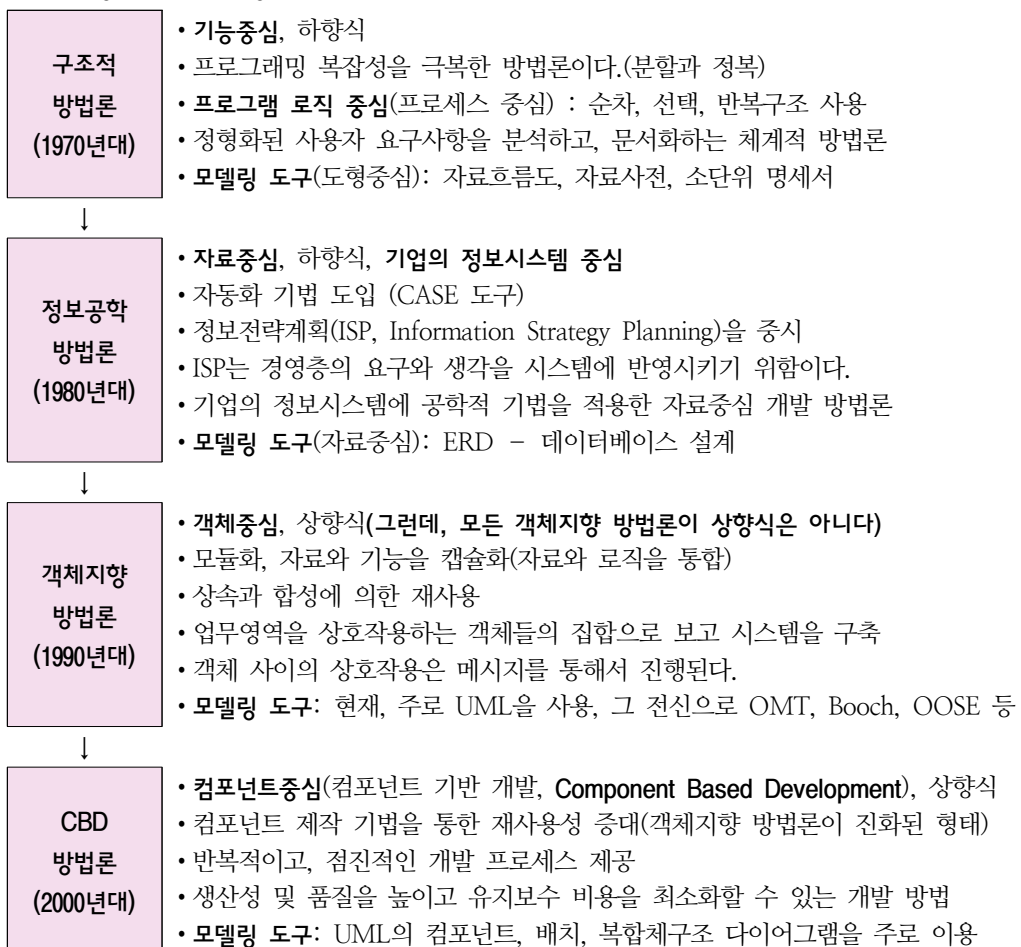


소프트웨어 개발론

소프트웨어 개발 방법론 정의

소프트웨어 공학 원리를 소프트웨어 개발 생명주기에 적용한 개념이다.
정보시스템을 개발하기 위한 활동, 절차, 산출물, 기법 등을 체계화한 것이다.

// 개발 방법론 진화 과정



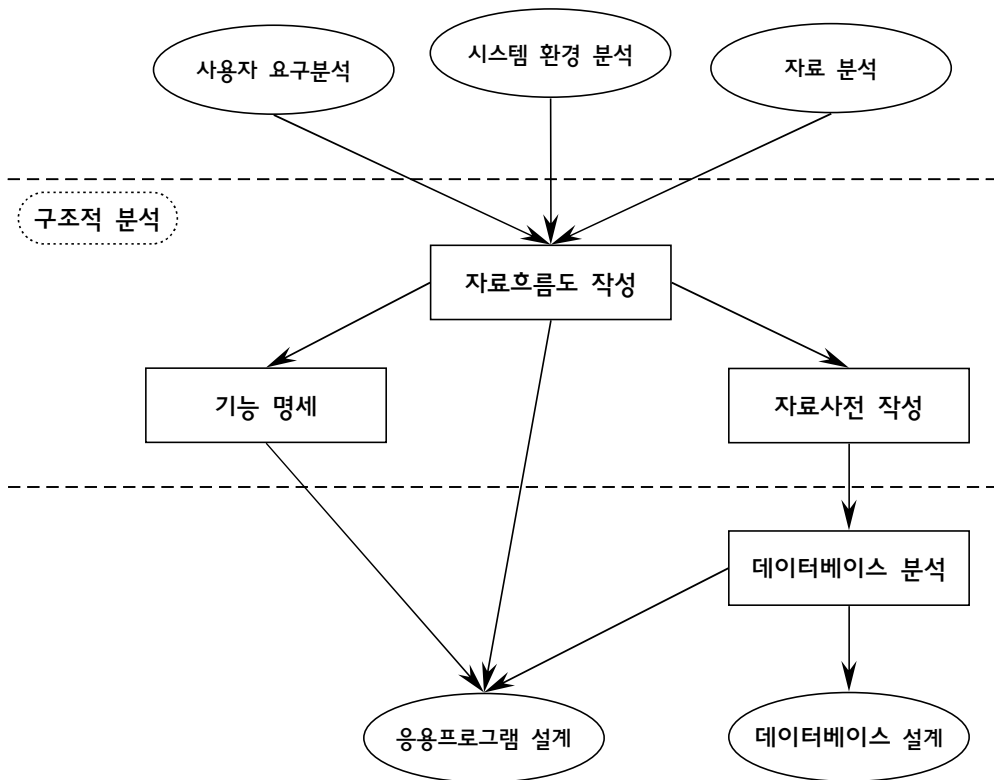
1. 구조적 방법론

구조적 방법론 정의

전체 시스템을 기능에 따라 모듈 단위로 분할하여 개발하고, 이를 통합하는 분할과 정복 접근방식과 **프로세스 중심**의 개발 방법론이다.

↓ 구조적 분석과 설계

— <구조적 분석과 설계 흐름> —



- 사용자 요구사항을 분석하고, 이를 통해 시스템의 논리적 모델링을 수행하고
- 논리적 모델을 물리적 형태로 변환하고, 최종 시스템으로 구체화시켜 나간다.

// 구조적 방법론의 한계점

- 기업 전체에 대한 거시적 관점 부족과 단위 프로젝트 위주의 접근 방법
- 설계와 코딩을 강조하고, 업무활동 위주의 접근 및 데이터 모델링 방법의 미흡
- 명확한 방법론적인 지침이 미흡하다.

2. 정보공학 방법론

정보공학 방법론 정의

- 기업의 정보시스템 구축을 위한 정형화시킨 절차 및 방법론이다.
- 계획, 분석, 설계 등 전 과정을 정형화시킨 절차 및 방법론이다.

// 정보공학 방법론 등장 배경

- 1990년대 초, 구조적 개발 방법론의 문제점을 극복하기 위해 개발되었다.
 - C.Finkelstein이 처음 사용하였고, James Martin에 의해 폭넓게 전파
- 정보시스템 개발을 공학적으로 접근하기 위해 체계화시킨 개발 방법론
- 기업 전체 데이터 통합 어려움 등으로 새로운 정보시스템 개발 방법이 필요

// 정보공학 방법론의 특징

- **기업중심의 방법론**
 - 적용 대상이 기업의 비즈니스 시스템으로 단순한 소프트웨어 개발이 아니다.
 - 기업의 전략경영을 지원하기 위해 SIS(Strategic Information System)에 초점을 둔다.
- **ISP(Information Strategy Planning, 정보전략계획)를 중시**
 - 정보공학 4단계 중 가장 처음에 시작하는 단계이다.
 - 경영층의 요구와 생각을 시스템에 반영시키기 위함이다.
- **데이터 중심의 분석과 설계**
 - 데이터와 프로세스를 분리하여 분석 및 설계를 진행하고, 상관분석을 통해 검증한다.
 - 기업 업무 프로세스는 수시로 변하지만 데이터는 대부분 변하지 않는다.
 - 해서, 시스템 유지보수를 줄이고 잦은 변화에 적극 대응하고자 하는 개념이다.
- **업무 영역을 분할과정복 방식으로 분석(Divide and Conquer)**
 - 기업의 전략경영을 지원하기 위해 SIS(Strategic Information System)에 초점을 둔다.
 - 문제영역을 세분화해 가면서 최종적으로 완성(Top-Down 방식 적용)
 - 진행절차 : 문제영역 → 업무영역 → 업무시스템 → 구현시스템
- **공학적 접근 방식**
 - 분석, 설계 등 초기단계를 중요시한다.(초기단계를 철저하게 작업)
 - 후속단계에서는 원시코드를 자동으로 생성하는 방식(정보시스템 개발의 자동화 지향)
- **사용자의 적극적인 참여 유도**
 - 작업 초기단계부터 사용자를 참여시켜 불명확한 요구사항을 없앤다.
 - 개발 완료 후 사용자 불만을 최소화하기 위함이다.

// 정보공학 4단계

ISP(정보전략계획) : 기업의 경쟁력 확보를 위한 정보전략 수립



업무영역분석 : 업무영역별로 데이터 모델링



업무시스템설계 : 시스템 중심으로 업무 설계



시스템구축 : 응용프로그램 관점에서 시스템 구현

3. 객체지향 방법론

객체지향 방법론 정의

프로그램을 객체와 객체 사이의 인터페이스 형태로 구성하기 위하여 문제영역에서 객체와 클래스, 이들 사이의 관계를 식별하여 설계모델로 변환하는 방법론이다.

- 복잡한 메커니즘의 실세계를 인간이 이해하는 방식으로 시스템에 적용시키는 개념이다.
- 소프트웨어를 자료와 프로세스로 분리하지 않고 실세계에 존재하는 사물 그대로 구현한다.
- 즉, 객체를 인간이 이해하는 방식 그대로 시스템을 구현하는데 적용하는 기술이다.

// 객체지향 방법론의 산출물

단계	주요 산출물
계획	• 프로젝트 계획서 • 객체지향 개발 방법들도 기존의 폭포수 모델을 토대로 함
분석	• Use Case diagram • Sequence diagram • Class diagram
설계	• Class diagram • Sequence diagram • Component diagram • Deployment diagram

- 객체지향 방법론은 분석, 설계, 구현의 벽이 없다.(분석은 문제영역, 설계는 솔루션 영역)

많은 객체지향 개발 방법론이 소개되고 있다. 다음은 대표적인 객체지향 개발 방법론이다.

부치(booch)의 OOD	• OOD는 Object Oriented Design 약어이다. • 정적 모델과 동적 모델로 표현
코트(coad)/유던(yourdon) OOAD	• OOAD는 Object-oriented analysis and design 약어이다. • 객체지향 = 객체 + 분류화 + 상속 + 객체 통신
실레어(shlaer)/멜러(mellor) OOSA	• OOSA는 Object Oriented Structured Analysis 약어이다. • 시스템, 도메인, 서브시스템, 객체, 상태, 프로세스의 6계층으로 분석 • 시스템을 최상위 수준의 시스템부터 최하위 프로세스 수준까지 표현
럼보(rumbaugh)의 OMT	• OMT는 Object Modeling Technique 약어이다. • 객체 모델링 / 동적 모델링 / 기능 모델링 등을 이용

〈럼보(rumbaugh) 객체지향 분석〉

- 소프트웨어 구성요소를 그래픽 표기로 모델링하는 기법이다.
 - 럼보의 객체지향 분석 기법을 **OMT**(Object Modeling Technique)라고도 한다.
 - 진행 순서 : 객체 모델링 → 동적 모델링 → 기능 모델링
-

① 객체 모델링(object modeling)

- 객체 모델링은 정보 모델링이라고도 한다.
- 객체 식별, 관계 정의, 클래스의 속성과 연산 기능을 정의한다.
- 시스템에 필요한 객체를 찾아내어 객체들 관계를 **객체 다이어그램**으로 나타내는 것
- 객체 다이어그램은 연관성, 집단화, 일반화 관계를 중심으로 정의한다.

② 동적 모델링(dynamic modeling)

- 시간흐름에 따라 변하는 객체들 사이의 제어흐름, 상호작용, 연산순서 등을 표현
- 동적 모델링은 동적 행위를 **상태 다이어그램**으로 나타내는 것이다.

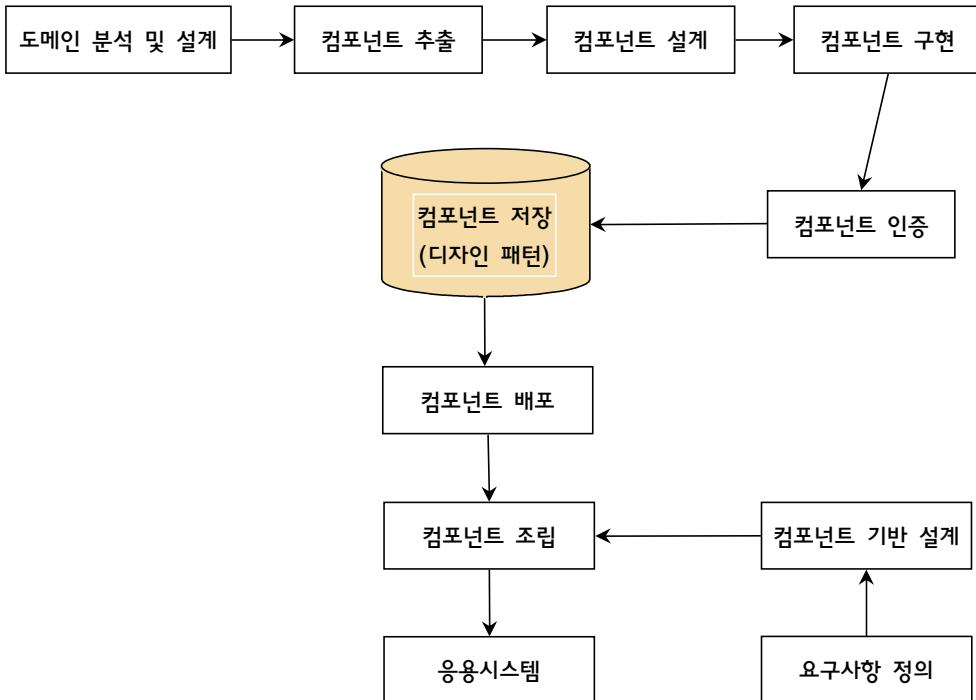
③ 기능 모델링(function modeling)

- 시스템 내에서 Data 값이 변하는 과정을 표현한다.(자료흐름)
- 즉, 어떤 데이터를 입력하여 어떤 결과를 구할 것인지를 표현한다.
- UML의 **활동(activity) 다이어그램**이나 자료흐름도(DFD)를 사용할 수 있다.

4. CBD 방법론

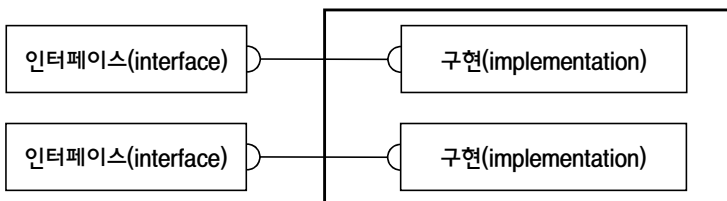
컴포넌트 기반 개발(component based development)은 컴포넌트를 조립하는 개발 방식이다.

〈CBD 개발 절차〉



- 컴포넌트 개발(component development)은 소프트웨어 개발에 필요한 부품을 만드는 것이다.
- 컴포넌트 개발을 위해서는 비즈니스 영역에 대한 이해와 기술이 선행되어야 한다.

// 컴포넌트 구조



- 인터페이스(interface) : 서비스를 외부에서 접근할 수 있도록 하는 가상함수(virtual function)
- 구현(implementation) : 인터페이스를 상속 받아 실제 서비스를 구현한 부분(객체, 클래스와 동일)

기출문제 분석

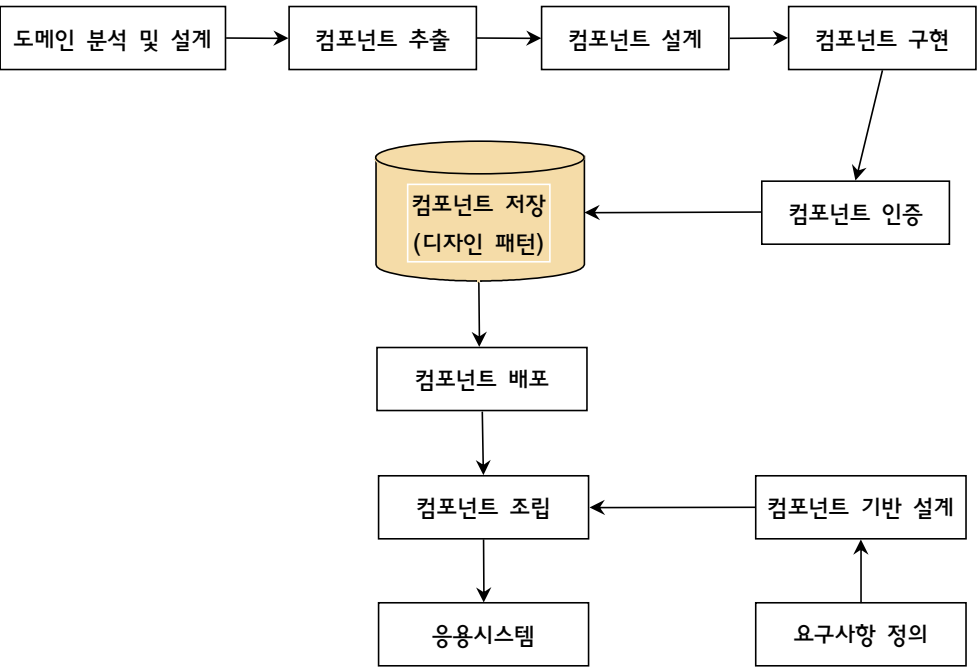
1. 컴포넌트 기반 소프트웨어 개발(CBSD)에 대한 설명으로 가장 옳지 않은 것은? [2021년 서울 7급]

- ① 고객이 원하는 요구들을 지속적으로 피드백한다.
- ② 이미 개발된 컴포넌트를 조립하여 시스템을 개발하는 과정이다.
- ③ 도메인 영역에서 재사용이 가능한 기능적 요구사항을 명확하게 정의하는 것이 필요하다.
- ④ 기능의 추가나 변경이 필요한 경우 요구에 따라 커스터마이징이 가능하다.

☞ 컴포넌트 기반 소프트웨어 개발

• 고객이 원하는 요구들을 지속적으로 피드백한다.(x) → 애자일 소프트웨어 개발

// 컴포넌트 기반 개발은 컴포넌트를 조립하는 개발 방식이다.



- 컴포넌트 개발(component development)은 소프트웨어 개발에 필요한 부품을 만드는 것이다.
- 컴포넌트 개발을 위해서는 비즈니스 영역에 대한 이해와 기술이 선행되어야 한다.
- 커스터마이징은 '주문 제작하다'라는 뜻이다. 'customize'에서 따온 것이다.

2. 다음 소프트웨어 개발 프로세스 모델로 가장 적절한 것은? [2025년 군무 7급]

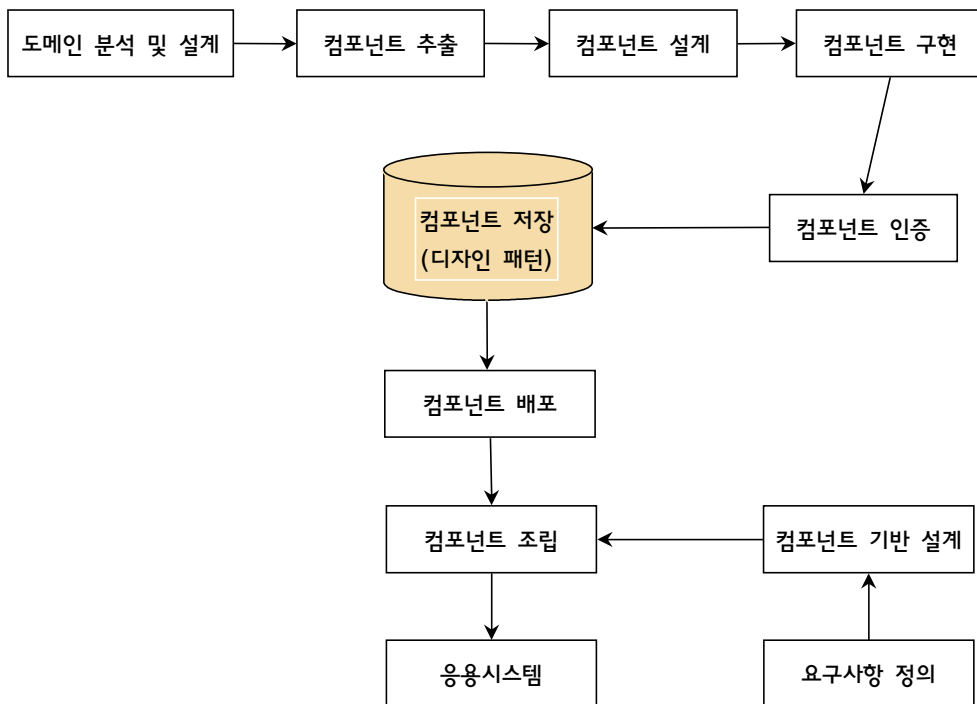
- 독립적인 서비스를 제공하는 소프트웨어 모듈을 조합해 재사용함으로써 개발 생산성과 품질을 높이고 시스템 유지보수 비용을 절감할 수 있는 개발 방법론
- 재사용성을 높이기 위해서는 결합도를 낮추고 응집도를 높이는 설계 전략이 요구됨

- ① 폭포수 모델
- ② 프로토타이핑 모델
- ③ 컴포넌트 기반 개발 모델
- ④ 애자일 개발 모델

☞ 컴포넌트 기반 개발 모델

- 컴포넌트 기반 개발은 컴포넌트를 조립하는 개발 방식이다.(컴포넌트 재사용)

// 컴포넌트 기반 개발 절차



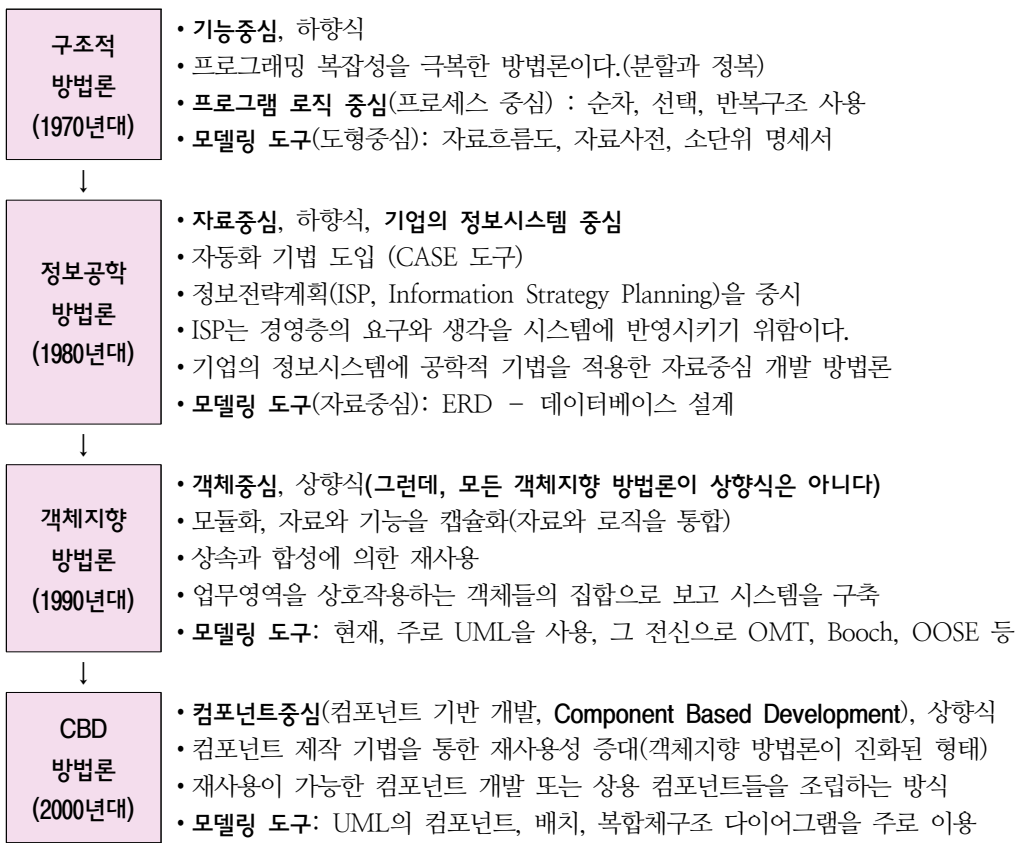
- 컴포넌트 기반 개발 모델은 소프트웨어를 재사용 가능한 컴포넌트 단위로 나누어 개발한다.
- 개발한 컴포넌트들을 조립하여 하나의 응용프로그램을 구축하는 소프트웨어 개발 방법론이다.
- 컴포넌트 개발을 위해서는 비즈니스 영역에 대한 이해와 기술이 선행되어야 한다.
- 컴포넌트는 독립적으로 동작하고 특정 기능을 수행하는 소프트웨어 모듈이다.
- 이 모델은 재사용성 향상, 개발 생산성 증대, 시스템 유지보수 용이성 등의 장점을 가진다.

3. 다음 소프트웨어 개발 방법론과 모델링 도구를 바르게 연결한 것은? [2025년 국가 7급]

(가) 정보공학 방법론
(나) 구조적 방법론
(다) 객체지향 방법론
A. 유스케이스 다이어그램(Usecase Diagram)
B. E - R 다이어그램(E-R Diagram)
C. 자료흐름도(Data Flow Diagram)

- ① (가) : A (나) : B (다) : C
- ② (가) : A (나) : C (다) : B
- ③ (가) : B (나) : C (다) : A
- ④ (가) : C (나) : B (다) : A

☞ 소프트웨어 개발 방법론 진화 과정



4. 소프트웨어 개발 방법에 대한 설명으로 옳은 것은? [2015년 국가 7급]

- ① 구조적 분석 방법은 자료와 함수의 관점으로 요구분석을 수행하며 함수보다는 자료에 중점을 둔다.
- ② 객체지향 방법은 객체가 가지는 자료와 오퍼레이션을 정의하며 자료흐름도(data flow diagram)가 대표적인 표기법으로 사용된다.
- ③ 정보공학 방법은 자료에 중점을 두어 자료와 프로세스를 별개의 작업으로 병행 진행한 뒤에는 서로 간의 오류를 상관 분석을 통해 검증한다.
- ④ 정형명세 방법에서는 시스템의 상태와 상태 변화를 논리적으로 명확하게 표현하는 것이 중요하며 UML이 대표적인 정형명세 방법의 표기법이다.

☞ 소프트웨어 개발 방법

-
- ① 구조적 분석 방법은 자료와 함수의 관점으로 요구분석을 수행하며 함수보다는 자료에 중점을 둔다.(×)
→ 구조적 분석에는 자료흐름도, 자료사전, 소단위 명세서, 개체-관계도(ERD)를 사용
 - ② 객체지향 방법은 객체가 가지는 자료와 오퍼레이션을 정의하며 자료흐름도(Data Flow Diagram)가 대표적인 표기법으로 사용된다.(×)
→ 객체지향 방법은 UML이 대표적인 표기법으로 사용된다.
 - ③ 정보공학 방법은 자료에 중점을 두어 자료와 프로세스를 별개의 작업으로 병행 진행한 뒤에는 서로 간의 오류를 상관 분석을 통해 검증한다.(○)
 - ④ 정형명세 방법에서는 시스템의 상태와 상태 변화를 논리적으로 명확하게 표현하는 것이 중요하며 UML이 대표적인 정형명세 방법의 표기법이다.(×)
→ 정형명세 방법은 요구사항 기술을 수학 및 논리학에 기반을 둔다.
-

정답 : ③