

11. UP / RUP

1. 개요

〈UP와 RUP 관계〉

먼저, UP는 통합 프로세스(unified process)를 의미한다.

UP는 Jacobson, Booch, Rumbaugh에 의해 개발되었다.

UP는 소프트웨어 객체지향 개발 방법론 중 하나이다.

UP는 반복적이고 점진적인 소프트웨어 개발 프로세스 프레임워크이다.

RUP는 래셔널(Rational)에서 제안한 가장 잘 알려진 UP 기반의 통합 프로세스이다.

그 외 통합 프로세스로 OpenUP와 애자일 통합 프로세스가 있다.

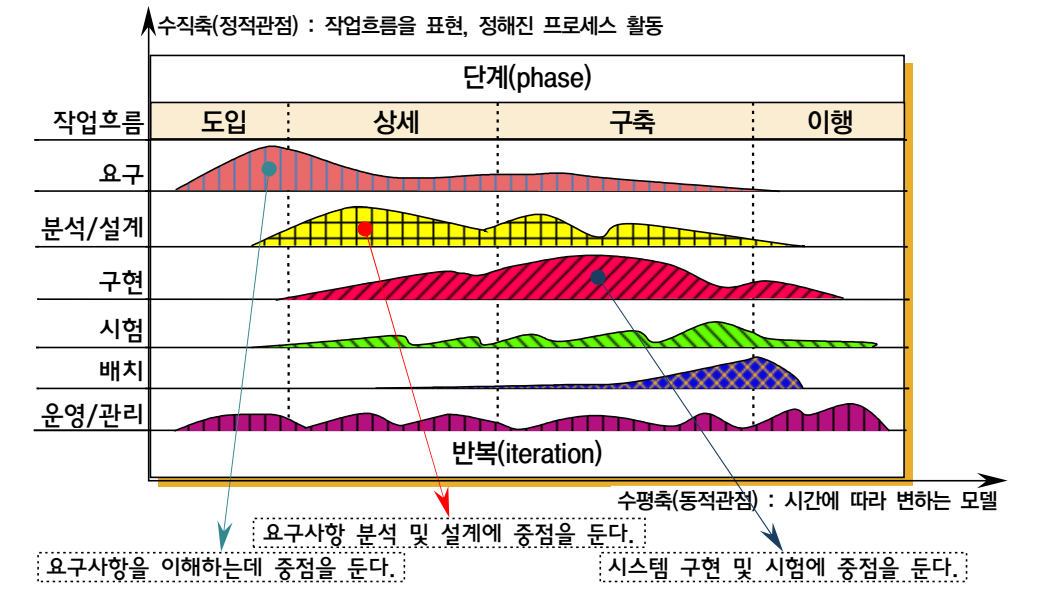
〈UP와 RUP 특징〉

- UML의 유스케이스 기반으로 사용자의 요구사항을 식별하고, 정의한다.
 - UML의 유스케이스 중심으로 기술적인 관점보다는 비즈니스 관점에 집중한다.
 - 아키텍처(architecture) 중심의 개발을 지향한다.
 - 시스템 전체를 표현한 아키텍처는 참여자들에게 최종 산출물의 모습을 인지하게 한다.
 - 참여자들에게 공통된 시각을 갖도록 한다.
 - 위험관리를 중시한다.
 - 프로젝트 성공에 장애가 될 수 있는 위험요소들을 파악한다.
 - 위험도가 높을수록 프로젝트 초기에 처리 방안을 찾아 해결한다.
-

〈UP와 RUP에서 소프트웨어 개발 단계〉

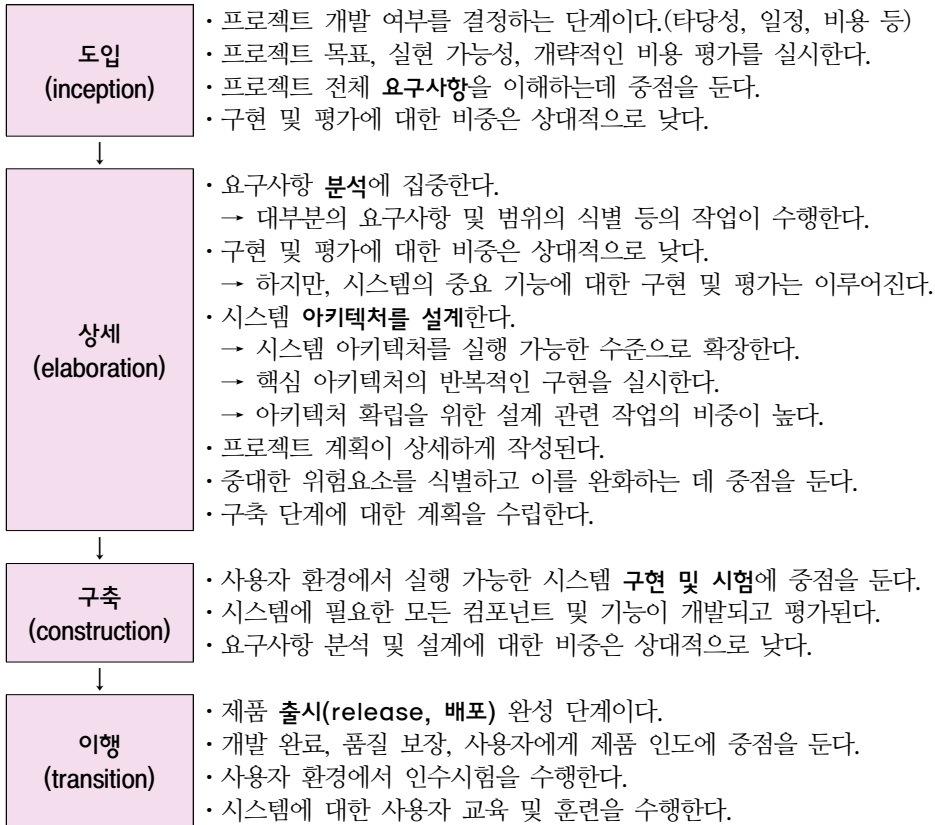
- 소프트웨어 개발 단계는 시간순서에 따라 4개의 범주로 구분한다.(도입, 상세, 구축, 이행)
 - 각 범주는 요구사항 도출, 분석, 설계, 구현, 시험, 배치 등 개발 생명주기를 포함한다.
-

2. 소프트웨어 개발 단계



- 먼저, UP에서 단계(phase)는 도입, 상세, 구축, 이행을 말한다.
 - UP에서는 요구/분석/설계/구현/테스트/배포 등이 단계가 아니다.
 - 단지, 폭포수 모델 등에서는 요구단계, 분석단계라는 말을 사용하고 있다.
- 모든 작업흐름(workflow)은 요구사항 도출, 분석, 설계, 구현, 시험을 포함한다.
 - 즉, 작업흐름 자체가 하나의 작은 폭포수 모델의 프로젝트와 같다.
- 소프트웨어 개발의 시작부터 모든 영역의 활동을 수행한다.(반복적 개발)
 - 중복되지만 시간이 지나도 대부분의 활동을 모두 수행합니다.
 - 단, 시간이 지나면서 비중이 요구, 분석/설계, 구현, 시험, 배포로 이동할 뿐이다.
- 시간이 지나면서 각 단계에서 집중해야 하는 역영은 변한다.
 - 그림에서 색이 칠해져 있는 부분이 각 단계에서 노력을 집중하는 정도를 의미한다.
 - 프로젝트 초반에는 요구사항 도출에 많은 노력을 기울임을 알 수 있다.
 - 그렇다고, 분석/설계/구현/테스트/배포를 수행하지 않는 것은 아니다.
- 수평축(시간)에서 볼 때, 구축(construction) 단계가 가장 많은 비중을 차지한다.(그림 참조)
 - 소프트웨어 개발은 실제 코딩하는 시간이 절대적으로 많아야 한다는 것이다.
 - 아무리 요구/분석/설계를 잘해도 구현 시점에서 이전 수행한 것을 변경할 수도 있다.

◆ 소프트웨어 개발 단계 세부적 설명



↓
↓ 각 단계의 목표
↓

<각 개발 단계의 목표>

- 도입** : 프로젝트 전반적인 범위를 정의하고, 최종제품과 관련된 비즈니스 비전(vision) 정의
- 상세** : 산출물을 정제하고 구조를 정의하며 개발 및 배치를 위한 정확한 계획수립
- 구축** : 최종사용자에게 인도될 제품을 구현
- 이행** : 최종제품을 사용자환경에 이전(제품 제조, 인도, 교육, 지원, 유지보수를 포함)

4 <http://cafe.daum.net/pass365>(홍재연)

3. RUP에서 프로세스를 나타내는 3가지 관점

① 동적관점(수평축)

- 시간에 따라 변하는 모델을 나타내는 관점이다.
- 동적 생명주기를 나타내며, 시간 흐름에 따라 단계를 나누고 단계별 목적을 제시한다.

② 정적관점(수직축)

- 정해진 대로 행동하는 프로세스 활동을 나타내는 관점이다.
- 핵심적인 **작업흐름(workflow)**을 표현한다.
- 작업흐름은 단계별 목적 달성을 위해 전체 개발 과정에 걸쳐 수행될 수 있다.

③ 기술관점(실무 관점)

- 프로세스 동안에 사용되는 **실무 관행**을 제시하는 관점이다.
- 프로세스 동안 사용할 좋은 기술들을 제안한다.

〈래셔널에서 선정한 소프트웨어 공학에 필요한 6가지 교훈〉

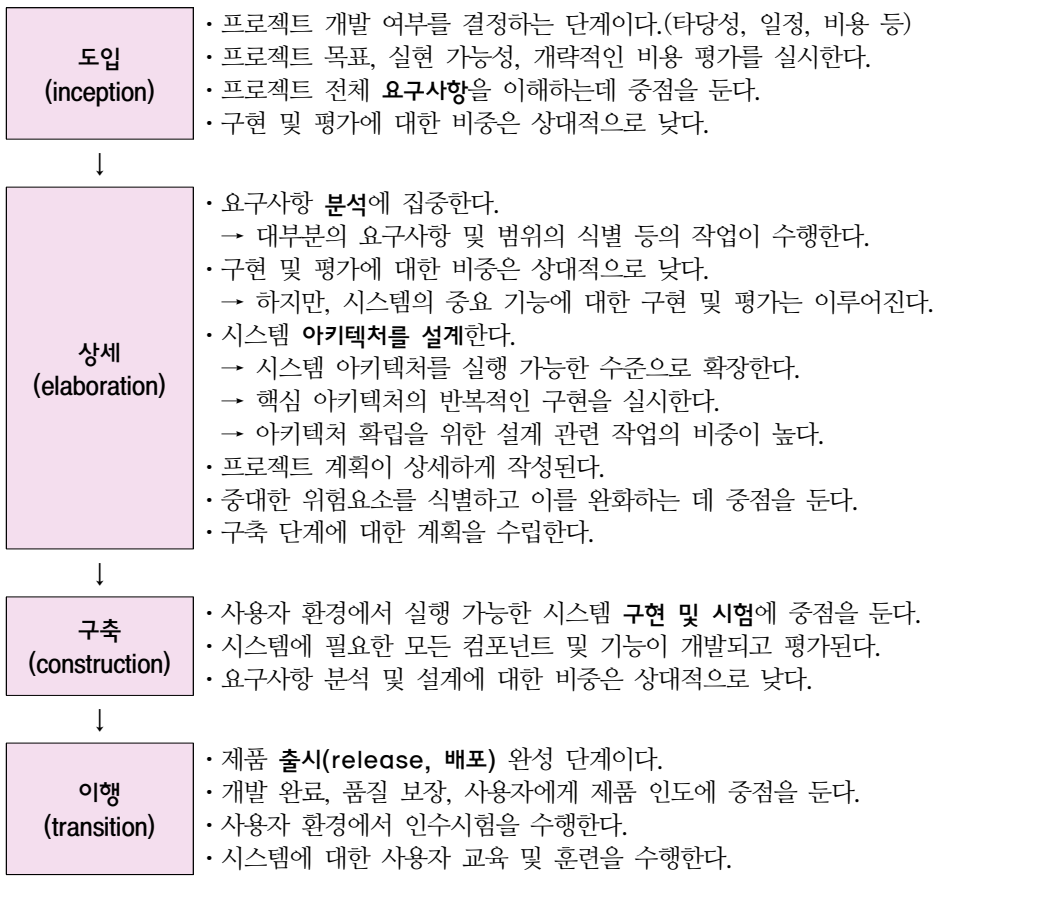
- 변화를 통제하라.
 - 필수사항을 관리하라.
 - 소프트웨어를 시각화하라.
 - 품질을 지속적으로 확인하라.
 - 컴포넌트 기반의 구조를 도입하라.
 - 발견된 위험요소를 원동력으로 반복적으로 개발하라.
-

기출문제 분석

1. 객체지향개발방법론인 UP(Unified Process)에서 정제된 비전, 핵심 아키텍처의 반복적인 구현, 대부분의 요구사항 및 범위의 식별 등의 작업이 수행되는 단계는? [2008년 국가 7급]

- ① construction 단계
- ② elaboration 단계
- ③ inception 단계
- ④ transition 단계

☞ UP(unified process)



정답 : ②

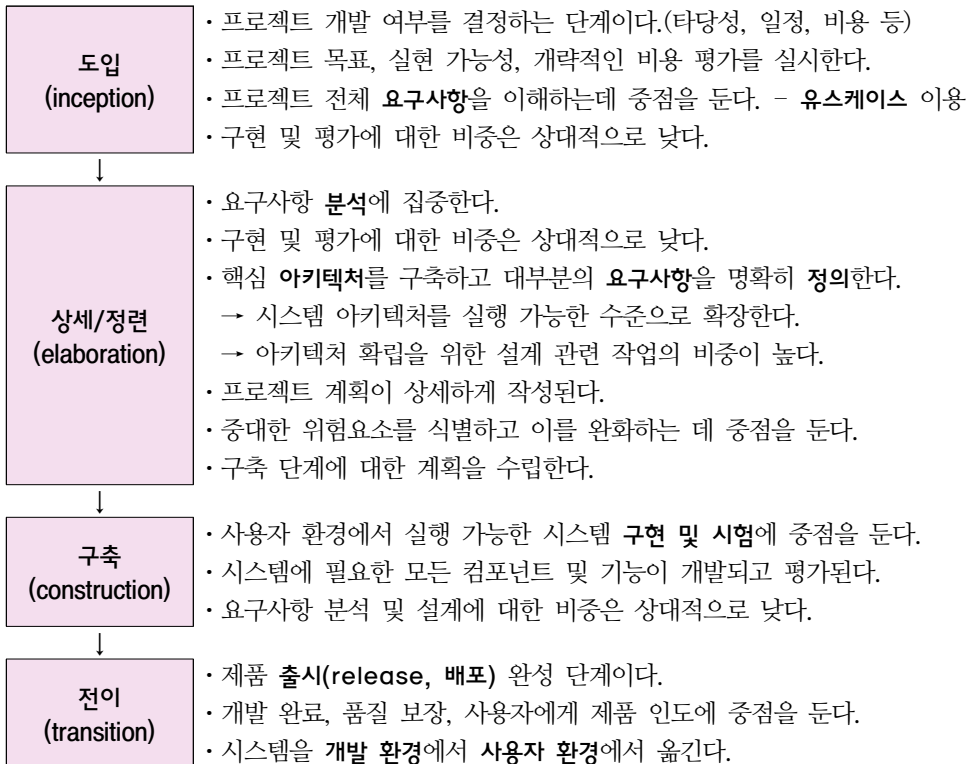
2. <보기>에서 설명하는 UP(unified process)의 단계 (가)~(다)를 옳게 짝지은 것은? [2022년 서울 7급]

-----<보기>-----

- (가) 소프트웨어 시스템을 개발 환경에서 사용자 환경으로 옮긴다.
- (나) 핵심 아키텍처를 구축하고 대부분의 요구사항을 명확히 정의한다.
- (다) 주요 요구사항들을 나열하고 전체적으로 10% 정도의 유스케이스(use case)를 상세히 작성한다.

- | | | |
|--------------------|------------------|-----------------|
| (가) | (나) | (다) |
| ① 전이(transition) | 정련(elaboration) | 도입(inception) |
| ② 도입(inception) | 구축(construction) | 정련(elaboration) |
| ③ 전이(transition) | 구축(construction) | 정련(elaboration) |
| ④ 구축(construction) | 전이(transition) | 도입(inception) |

☞ UP

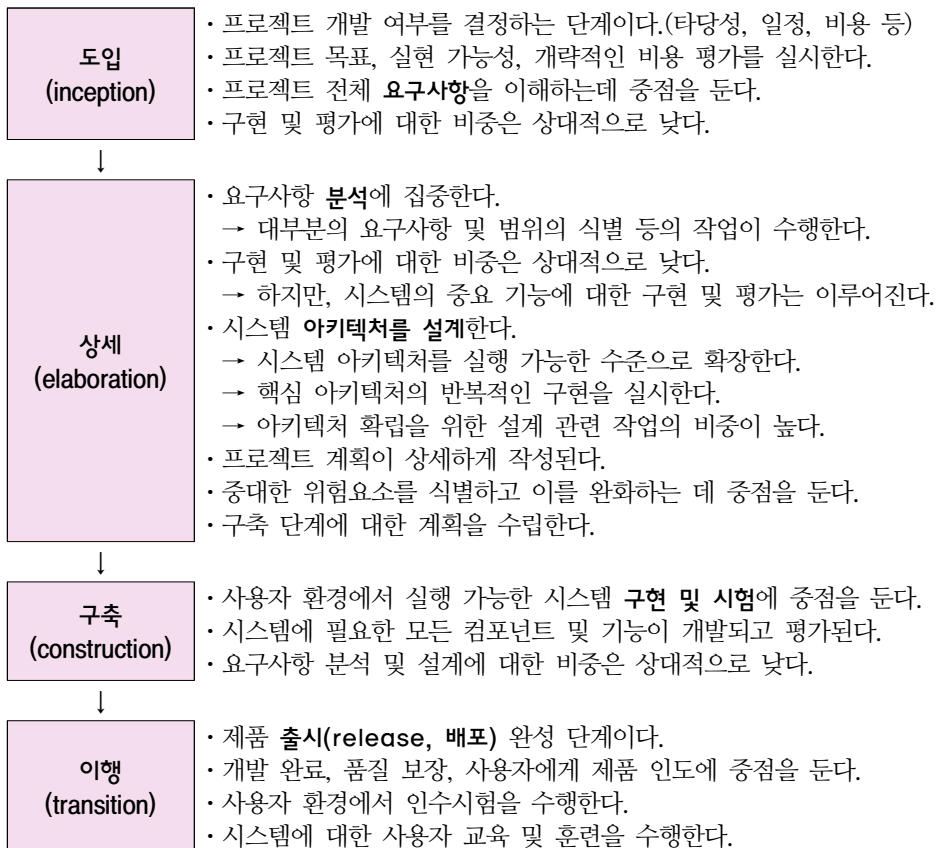


3. 다음은 UP(Unified Process)의 어떤 단계를 설명한 것인가? [2012년 국가 7급]

시스템 아키텍처 확립을 위한 설계 관련 작업의 비중이 크게 나타나고 프로젝트 계획이 상세히 작성된다.

- ① 도입 단계(inception phase)
- ② 정련 단계(elaboration phase)
- ③ 구축 단계(construction phase)
- ④ 전이 단계(transition phase)

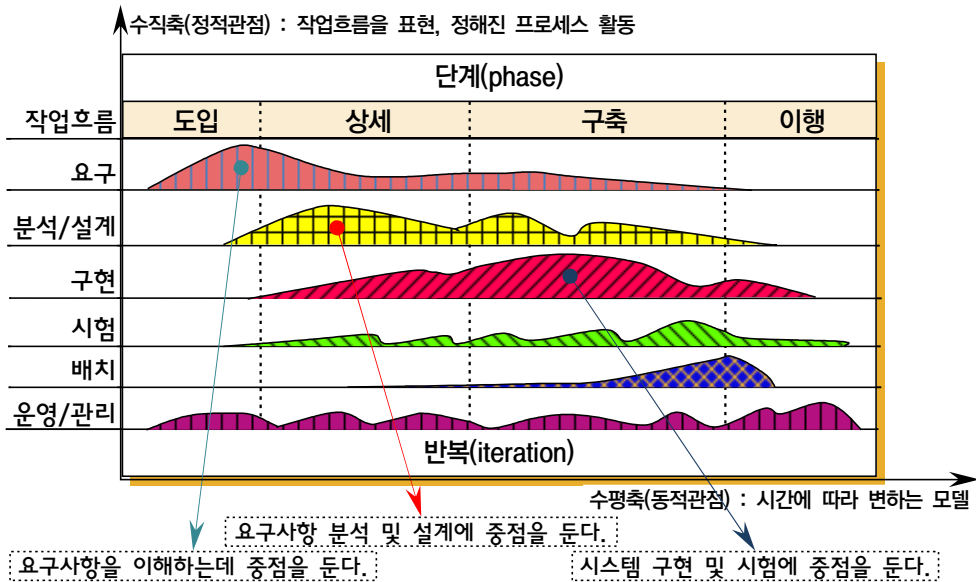
☞ UP(Unified Process)



4. 소프트웨어 개발 단계를 시간의 흐름에 따라 네 개의 범주(도입, 상세, 구축, 이행)로 나누고, 각 범주에는 요구사항 도출부터 설계, 구현, 평가까지의 개발 생명주기가 포함되어 있는 방법론은? [2020년 국가 7급]

- ① XP(eXtreme Programming)
- ② UP(Unified Process)
- ③ CMM(Capability Maturity Model)
- ④ SPICE(Software Process Improvement and Capability dEtermination)

☞ 통합 프로세스(unified process)



- 먼저, UP에서 단계(phase)는 도입, 상세, 구축, 이행을 말한다.
 - UP에서는 요구/분석/설계/구현/테스트/배포 등이 단계가 아니다.
 - 단지, 폭포수 모델 등에서는 요구단계, 분석단계라는 말을 사용하고 있다.
- 모든 작업흐름(workflow)은 요구사항 도출, 분석, 설계, 구현, 시험을 포함한다.
 - 즉, 작업흐름 자체가 하나의 작은 폭포수 모델의 프로젝트와 같다.
- 소프트웨어 개발의 시작부터 모든 영역의 활동을 수행한다.(반복적 개발)
 - 중복되지만 시간이 지나도 대부분의 활동을 모두 수행합니다.
 - 단, 시간이 지나면서 비중이 요구, 분석/설계, 구현, 시험, 배포로 이동할 뿐이다.

5. 통합 프로세스 모델(UP, unified process)의 특징을 설명한 것으로 옳지 않은 것은? [2016년 국가 7급]

- ① 반복적(iterative)이고, 점진적(incremental)인 개발 방법이다
- ② 유스케이스를 기반으로 한다.
- ③ 아키텍처 중심의 개발을 지향한다.
- ④ 위험관리는 포함되어 있지 않다.

☞ UP(unified process) - 통합 프로세스

- UP는 UML의 유스케이스 기반으로 사용자의 요구사항을 식별하고, 정의한다.
 - UP는 UML의 유스케이스 중심으로 기술적인 관점보다는 비즈니스 관점에 집중한다.
 - UP는 아키텍처(architecture) 중심의 개발을 지향한다.
 - 시스템 전체를 표현한 아키텍처는 참여자들에게 최종 산출물의 모습을 인지하게 한다.
 - 참여자들에게 공통된 시각을 갖도록 한다.
 - UP는 위험관리를 중시한다.
 - 프로젝트 성공에 장애가 될 수 있는 위험요소들을 파악한다.
 - 위험도가 높을수록 프로젝트 초기에 처리 방안을 찾아 해결한다.
 - UP는 반복적이고 점진적인 소프트웨어 개발 프로세스 프레임워크이다.
-

정답 : ④

6. 소프트웨어 통합 프로세스(UP) 모델의 구체화 단계(elaboration phase)에서 하는 주요 활동으로 옳지 않은 것은? [2018년 국가 7급]

- ① 사용자 환경에서 인수 테스트 수행
- ② 아키텍처 설계
- ③ 요구사항 분석
- ④ 중대한 위험요소 식별 및 해결

☞ 통합 프로세스(UP)

- 사용자 환경에서 인수 테스트 수행(×)
 - 사용자 환경에서 인수 테스트는 이행(transition) 단계에서 실시한다.
-

정답 : ①

7. 객체지향 소프트웨어 개발 프로세스 UP(unified process)의 특징으로 가장 옳지 않은 것은?
[2018년 서울 7급]

- ① 반복적(iterative)이고 점진적(incremental)이다.
- ② 유스케이스 기반(use-case driven)으로 한다.
- ③ 아키텍처 중심(architecture centric)이다.
- ④ E-R 다이어그램 기반으로 되어 있다.

☞ UP(unified process) - 통합 프로세스

- E-R 다이어그램 기반으로 되어 있다.(×)
→ 요구사항을 식별하고, 정의하는데 UML의 유스케이스를 사용한다.
-

정답 : ④

8. 객체지향 개발 프로세스인 RUP(Rational Unified Process)에 대한 설명으로 옳지 않은 것은?
[2009년 국가 7급]

- ① 반복적 개발 프로세스이다.
- ② 비즈니스 관점보다는 기술적인 관점에 초점을 맞춘다.
- ③ 고객의 요구사항을 명확하게 문서화하고 그 요구사항의 변경을 추적한다.
- ④ 구축(construction)단계에서도 요구사항을 기술한다.

☞ RUP(Rational Unified Process) - 래셔널 통합 프로세스

- RUP는 유스케이스 기반으로 사용자의 요구사항을 반복적이고 점진적으로 개발한다.
 - RUP는 UML의 유스케이스 중심으로 기술적인 관점보다는 **비즈니스 관점에 집중**한다.
 - RUP는 아키텍처 중심(architecture-centric) 시스템 개발이다.
 - 시스템에 대한 다양한 뷰를 모아 놓은 것, 소프트웨어 전체적인 형태(4+1 View)
 - UML의 4+1 View가 표준이다.
 - Logical View / Implementation View / Process View / Deployment View
 - 유스케이스 관점(UseCase View) - 시나리오
 - RUP는 통합 개발 방법론이다.(4 단계 범주 : 도입, 상세, 구축, 이행)
 - 각 범주는 요구사항 도출, 분석, 설계, 구현, 평가까지 개발 생명주기를 포함한다.
 - RUP는 구축(construction)단계에서도 요구사항을 기술한다.
 - 단지, 요구사항 분석 및 설계에 대한 비중은 상대적으로 낮다.
-

정답 : ②

9. RUP(Rational Unified Process)에서 프로세스를 나타내는 관점이 아닌 것은? [2013년 국가 7급]

- ① 시간에 따라 변하는 모델을 나타내는 동적관점
- ② 정해진 대로 행동하는 프로세스 활동을 나타내는 정적관점
- ③ 프로세스 동안에 사용되는 실무 관행을 제시하는 실무관점
- ④ 프로세스 진행 중 이해당사자(stakeholder)들의 조율을 위한 중재관점

♣ RUP에서 프로세스를 나타내는 3가지 관점

① 동적관점(수평축)

- 시간에 따라 변하는 모델을 나타내는 관점이다.
- 동적 생명주기를 나타내며, 시간 흐름에 따라 단계를 나누고 단계별 목적을 제시한다.

② 정적관점(수직축)

- 정해진 대로 행동하는 프로세스 활동을 나타내는 관점이다.
- 핵심적인 **작업흐름(workflow)**을 표현한다.
- 작업흐름은 단계별 목적 달성을 위해 전체 개발 과정에 걸쳐 수행될 수 있다.

③ 기술관점(실무 관점)

- 프로세스 동안에 사용되는 **실무 관행**을 제시하는 관점이다.
 - 프로세스 동안 사용할 좋은 기술들을 제안한다.
-

정답 : ④