



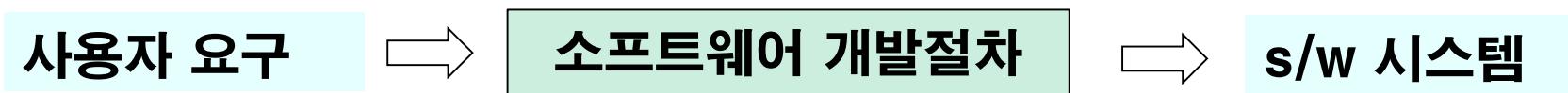
Unified Process

- Development Process
- Unified Process 역사
- Unified Process란?
- Unified Process 개발과정

Development Process

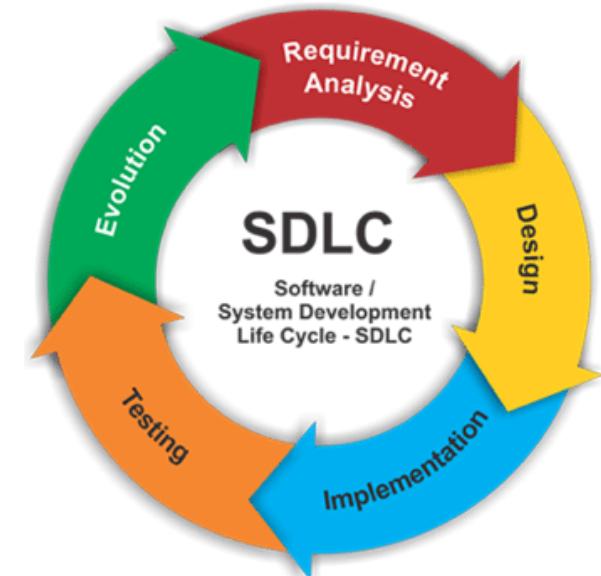
□ 개발절차(Process)란?

- 정의: 주어진 목표를 달성하기 위해서 누가, 언제, 무엇을, 어떻게 할 것인가에 관하여 정의해 놓은 것
- 목표: 소프트웨어 제품을 새로 개발하거나, 기존의 소프트웨어를 보강하는 것
- 사용자의 요구를 소프트웨어 시스템으로 변환시키는데 필요한 일련의 작업



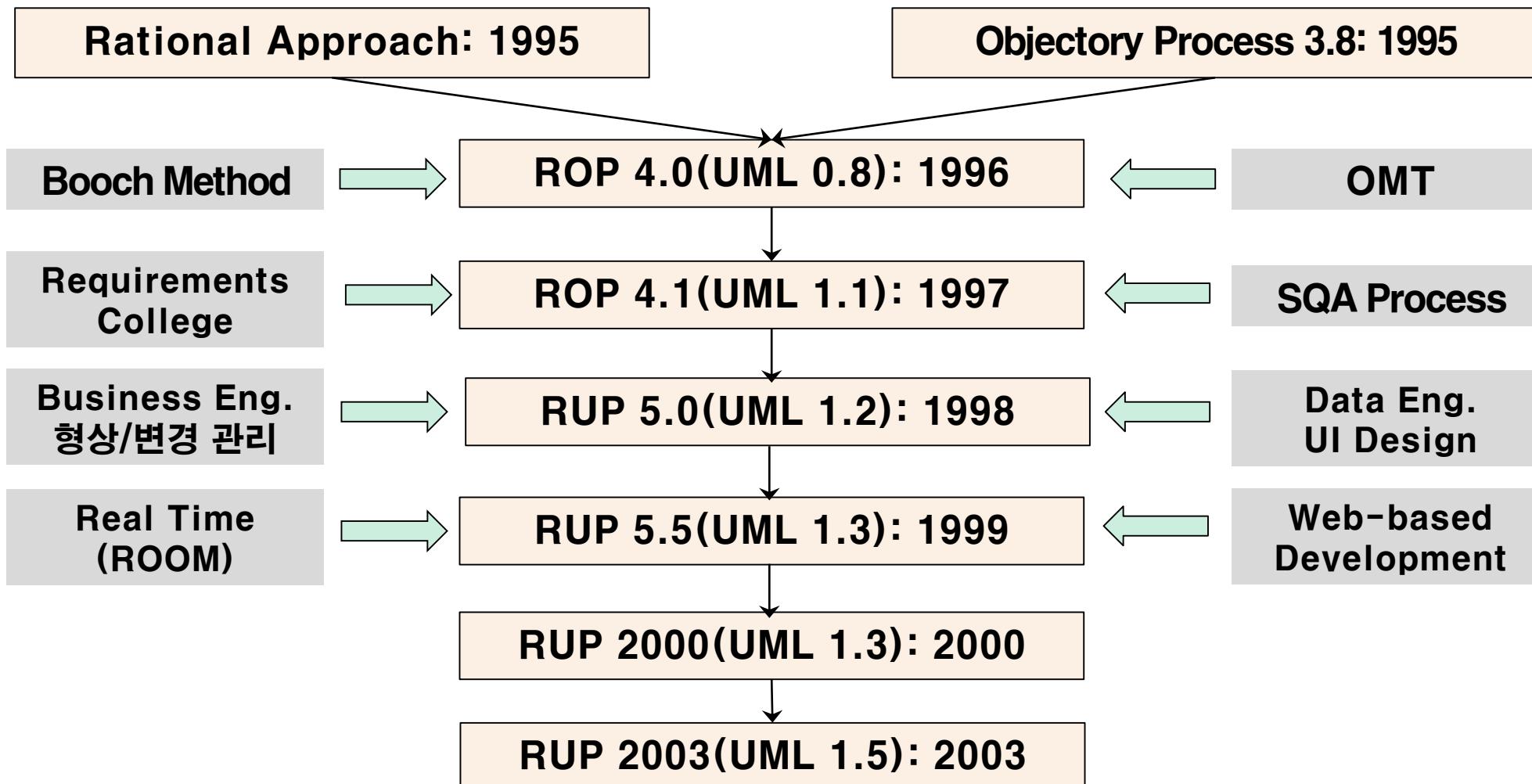
□ 효과적인 개발절차(effective process)

- 품질 좋은 소프트웨어의 효율적인 개발을 위한 지침을 제공
- 개발자들이 공통적인 시각과 문화의 공유를 진작
- 위험을 감소시키고, 예측 가능성을 증대시킴



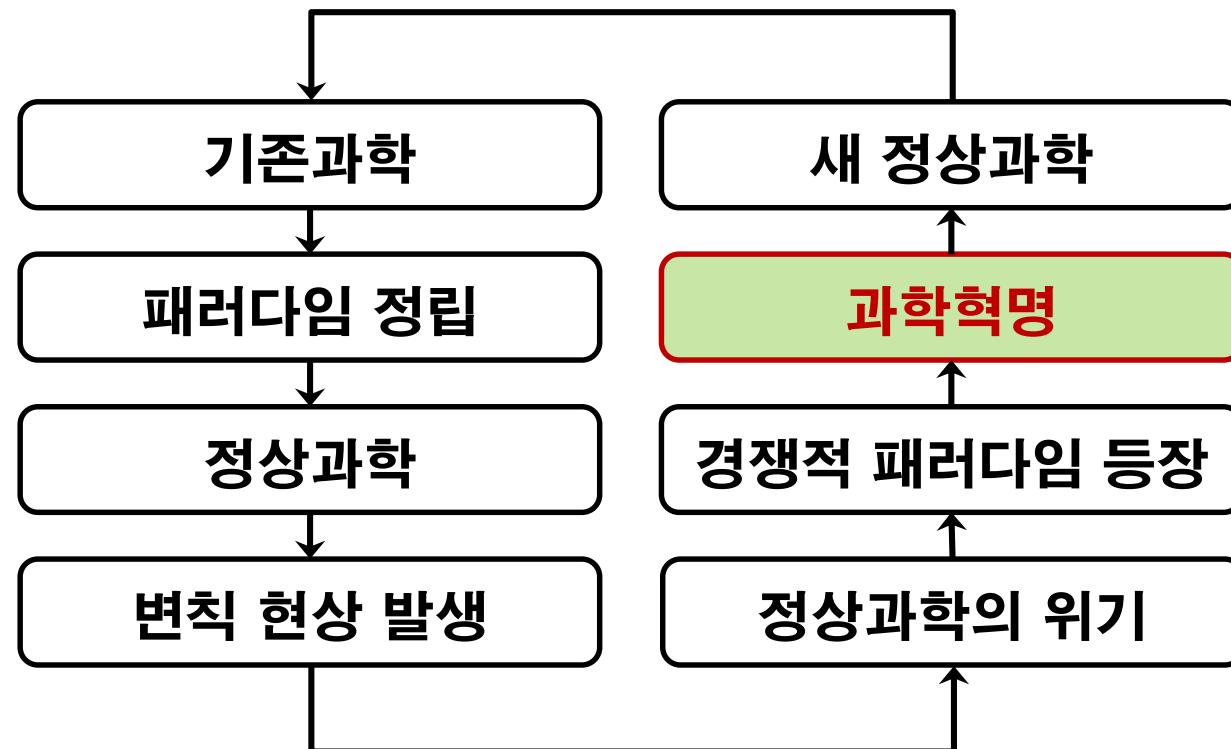
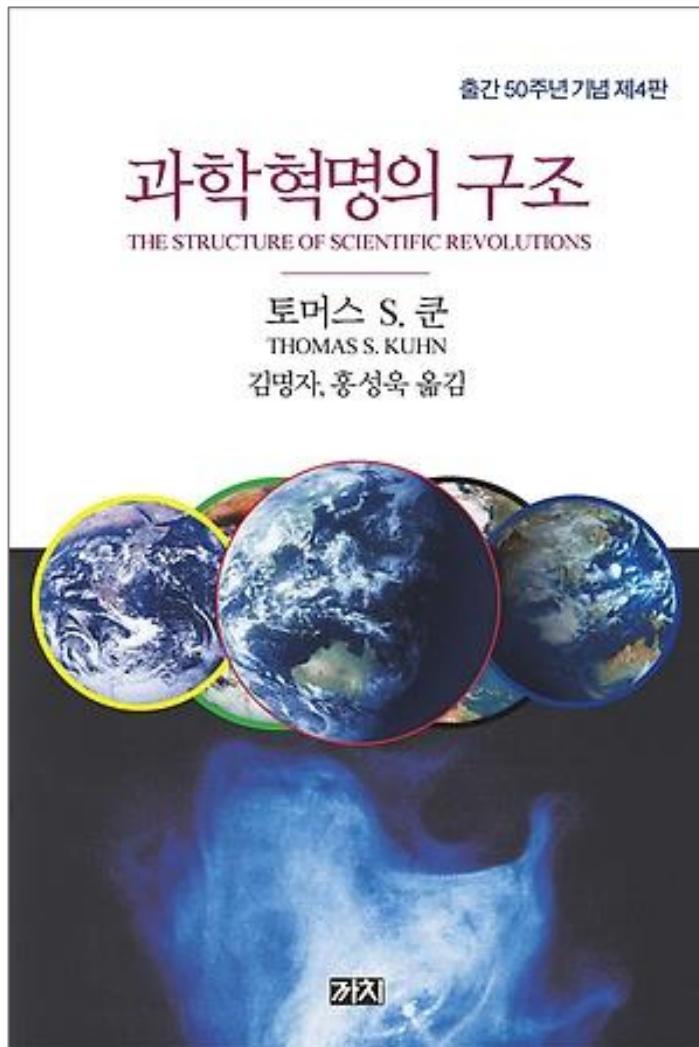
Unified Process의 역사(1)

□ Development of Unified Process(UP)



Unified Process의 역사(2)

□ Paradigm Shift



Unified Process의 역사(3)

- Ericsson 접근 방법(Ivar Jacobson이 제안)
 - 기본개념: 시스템은 서로 연관된 block들의 집합
 - 설계작업의 산출물: ①block diagrams, ②서브시스템의 구성
 - 개발과정의 핵심 문서: ①소프트웨어 구조 기술서, ②메시지 라이브러리
 - CCITT의 SDL(Specification and Description Language)에 강력한 영향을 미침
- Objectory (Object Factory의 약칭)
 - 1987년, Jacobson은 스톡홀름에서 『Objectory AB』 창립
 - 1987년, OOPSLA conference에서 유즈 케이스의 개념을 소개함으로 use-case-driven 방식을 정착
 - 1988년, Objectory 1.0 발표
 - 1995년, Objectory 3.8 발표

Unified Process의 역사(4)

□ Rational Approach

- 1981년, Interactive Environment 개발 시작
 - 객체지향 설계, 추상화, 정보 은닉, 재사용, 프로토타이핑의 개념을 도입
- Architecture-driven, Incremental & Iterative 개념 정착

□ Rational Objectory Process(ROP)

- 1995년, Rational사는 Objectory AB를 인수
- 1997년, ROP 4.1 발표
 - Booch Method, Rumbaugh의 OMT(Object Modeling Technique)
 - Phase & Controlled iterative Approach 도입
 - Architecture의 개념을 명시적으로 도입
- ROP의 모든 모델에서 UML을 사용

Unified Process의 역사(5)

□ Rational Unified Process(RUP)

- Rational는 Requisite Inc., SQA Inc., Pure-Atria, Performance Awareness, Vigortech 등의 소프트웨어 개발사를 인수 합병
- 1998년 6월, RUP 5.0 출시
 - ROP를 일반적인 업무를 모델링하는데 적합하도록 확장
- 1999년, RUP 5.5 출시
 - Real Time의 개념 및 Web 기반 방법을 도입
- 2000년, RUP 2000 출시

Unified Process의 역사(6)

□ UP의 다양화

- **프로젝트를 수행하는 과정에서 준수해야 할 원칙과 작업절차에 따라 다양한 형태로 변화**
- **잘 알려진 UP의 변종**
 - Agile Unified Process(AUP) by Scott W. Ambler(2005.9)
 - Basic Unified Process(BUP) by IBM (2005.10)
 - Enterprise Unified Process(EUP) – RUP의 확장판
 - Essential Unified Process(EssUP) by Ivar Jacobson
 - Open Unified Process(OpenUP) by Eclipse Process Framework (2006.3)
 - RUP by IBM / Rational Software
 - Oracle Unified Method(OUM) by Oracle
 - RUP-System Engineering(RUP-SE) by Rational Software

Unified Process란? (1)

□ Unified Process

- 객체지향 시스템을 위한 가장 보편적인 소프트웨어 개발 절차
 - 누가(**worker**), 무슨 활동(**activity**)을, 언제(**workflow**) 수행하여, 무엇을(**artifacts**) 산출하는지를 정의
 - 컴포넌트 기반
 - UML을 모델링 언어로 사용
- 널리 알려지고 문서화가 잘 되어 있는 것 -> **RUP**

□ Unified Process의 동적 특성

- **Use Case Driven**: 유즈 케이스를 축으로 개발 진행
- **Architecture Centric**: 개발과정에서 구조를 중시
- **Iterative and Incremental**: 반복적, 점진적인 개발
- **Risk Focused**: 위험요소를 조기에 파악하여 대처

Unified Process란? (2)

□ 유의할 용어

- **Worker(or Role)**

- 특정 업무를 수행하는 직책(role)
 - 구체적인 사람 또는 직위를 지칭하는 개념이 아님

- **Workflows:** 요구 추출, 분석, 설계, 구현, 테스트

- **Phases:** 도입, 정련, 구축, 전이

- **Milestone:** 개발절차에서 규정된 산출물이 모두 작성되었는가?

- **Increment**

- iteration의 결과로 만들어진 시스템의 일부
 - 직전 iteration까지 만들어진 시스템과 결합하여 실행 가능

- **Artifacts**

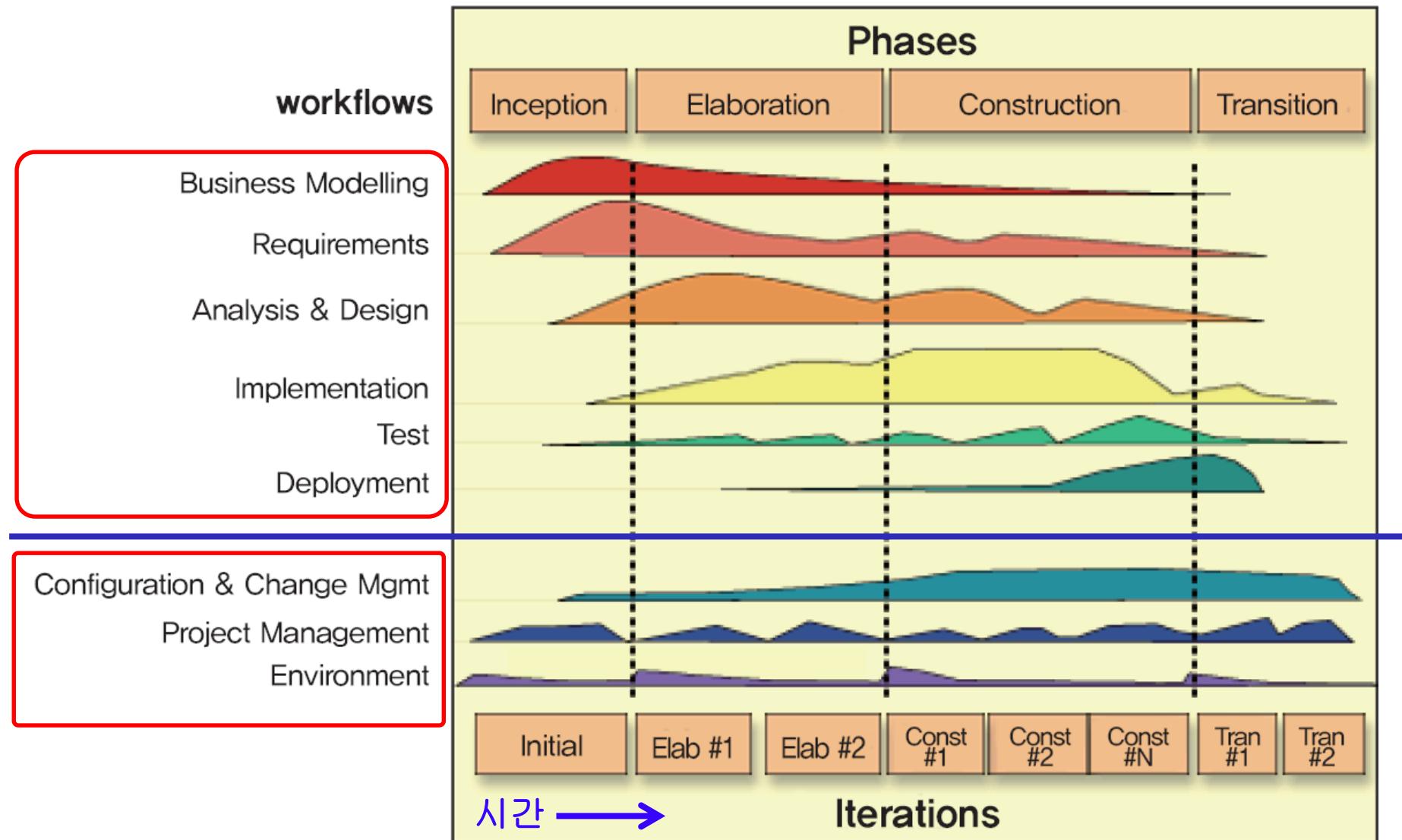
- 실행 가능한 Source code

- 매뉴얼(Manuals) 및 관련된 교부 문서(deliverables)

- 사용자 요구(기능적/비기능적), 유즈 케이스, 테스트 케이스

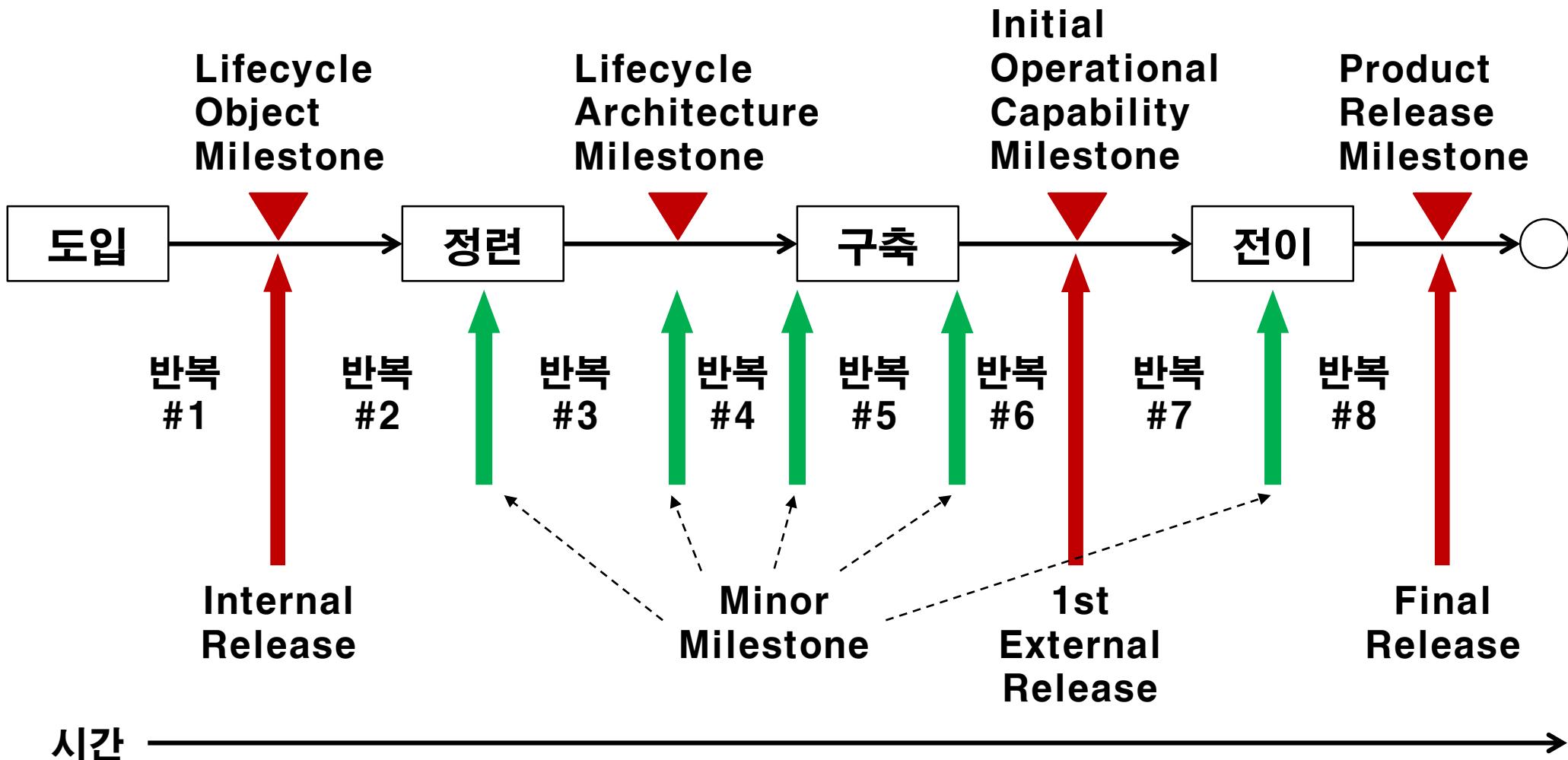
- UML로 기술된 소프트웨어 구조(Architecture) 및 가시적 모델

Unified Process란? (3)



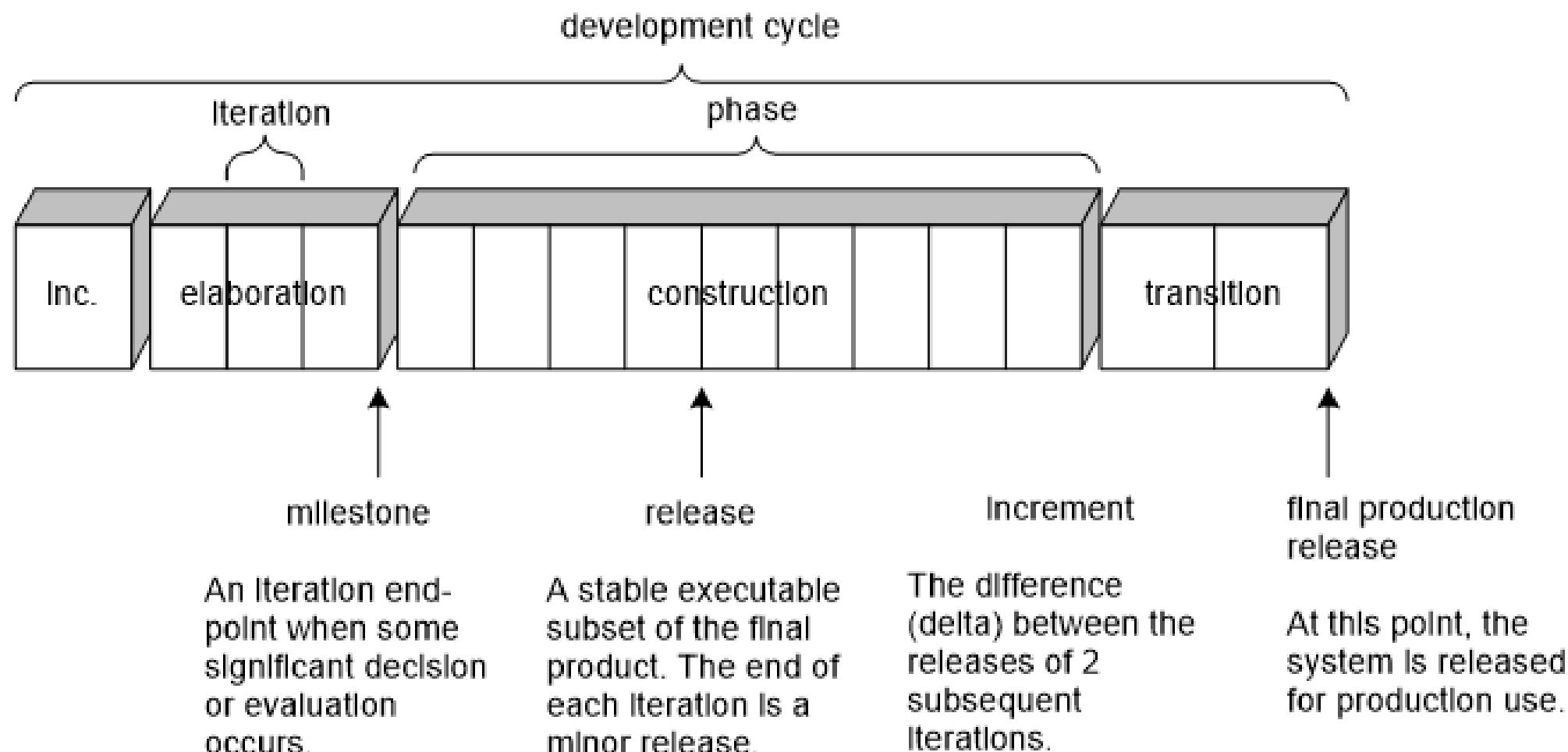
Unified Process란? (4)

□ UP Lifecycle & Milestone



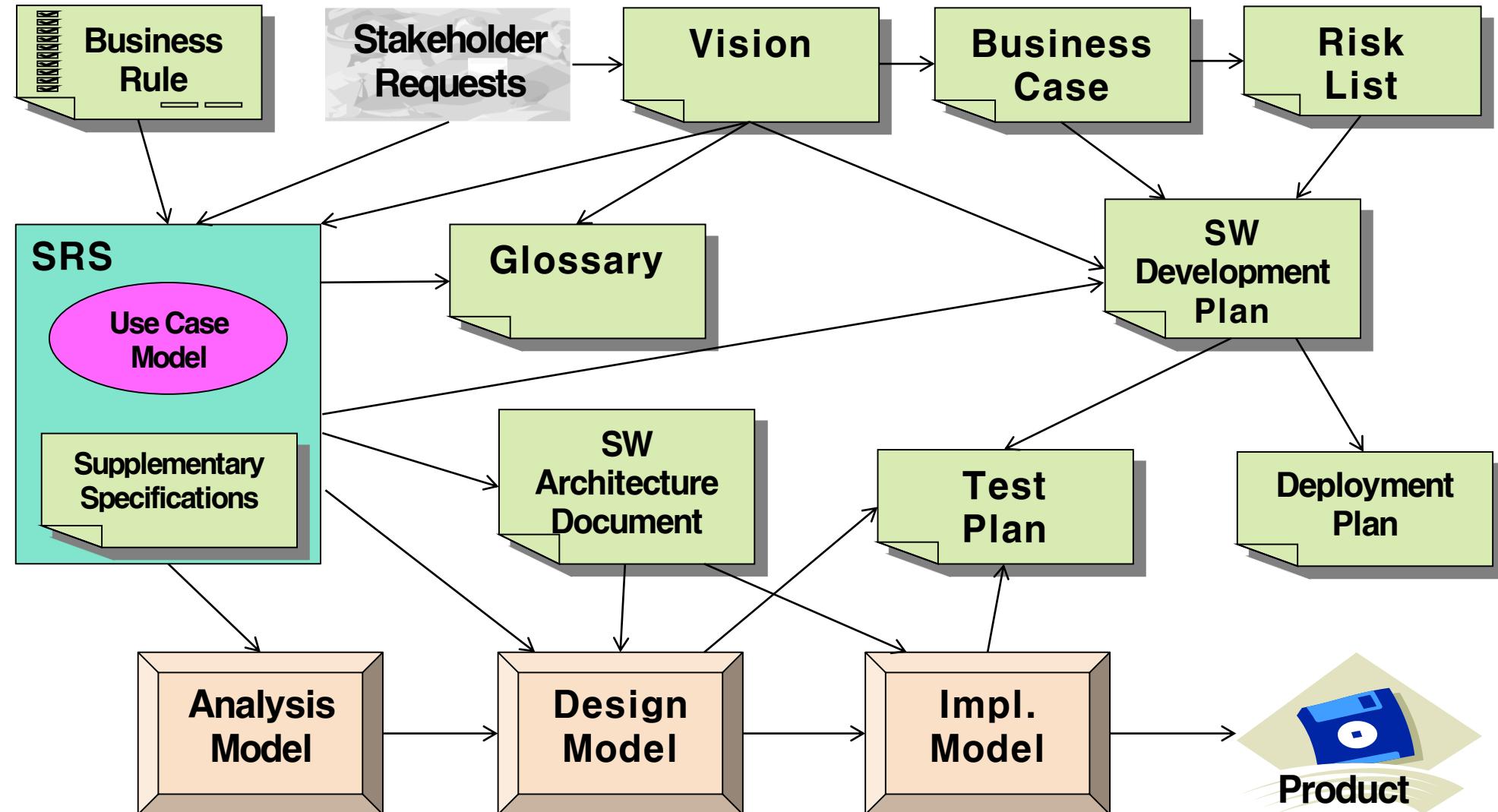
Unified Process란? (5)

□ UP Development Cycle



Unified Process란? (6)

□ UP Major Artifacts



UP에서의 개발과정(1)

□ 개발과정은 n개의 사이클로 구성

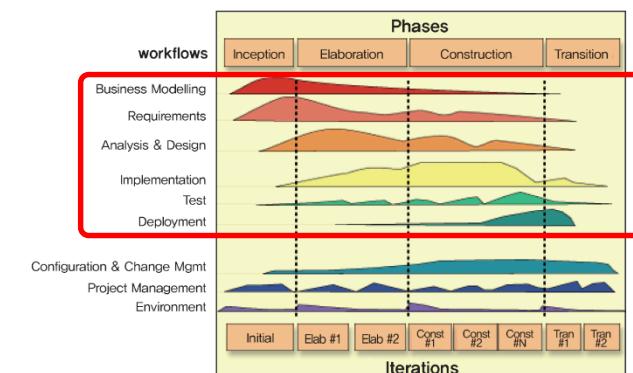
- 사이클은 독자적으로 실행 가능한 제품(product release)이 완성되면 종결
 - 개발 대상의 크기에 따라 사이클의 개수를 설정

▪ 각각의 사이클은 4개의 단계(phases)로 구성

- Inception(개념 형성), Elaboration(정제), Construction(구축), Transition(이전)
- 각각의 단계에서는 5개의 핵심 Workflow(작업흐름)을 수행하며, milestone(이정표)에 도달하면 종결

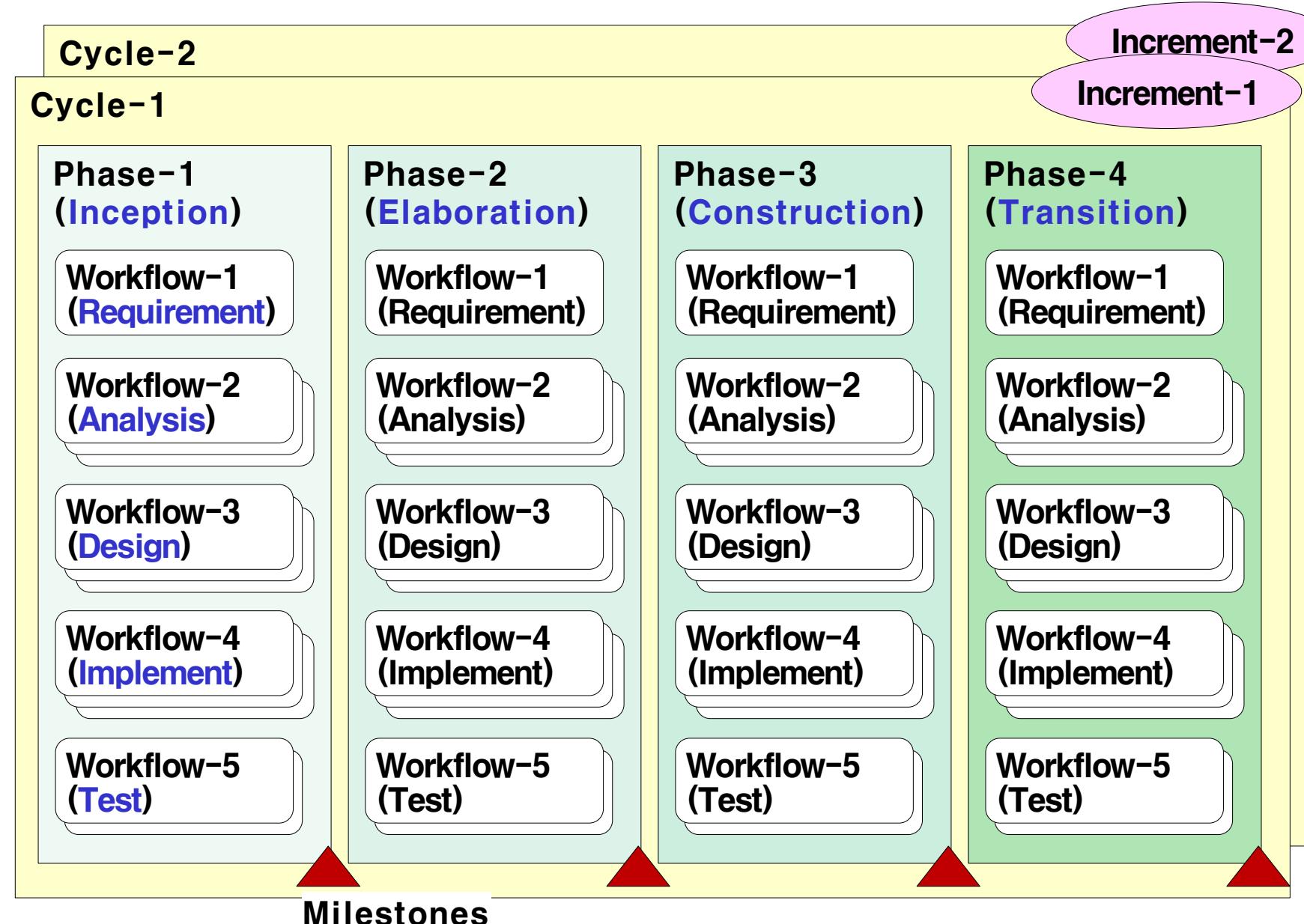
▪ 각 단계는 여러 개의 iteration으로 분할 가능

- 단계별로 수행할 작업의 양에 따라 적절한 개수로 분할
- increment 가 완성되면 iteration은 종결



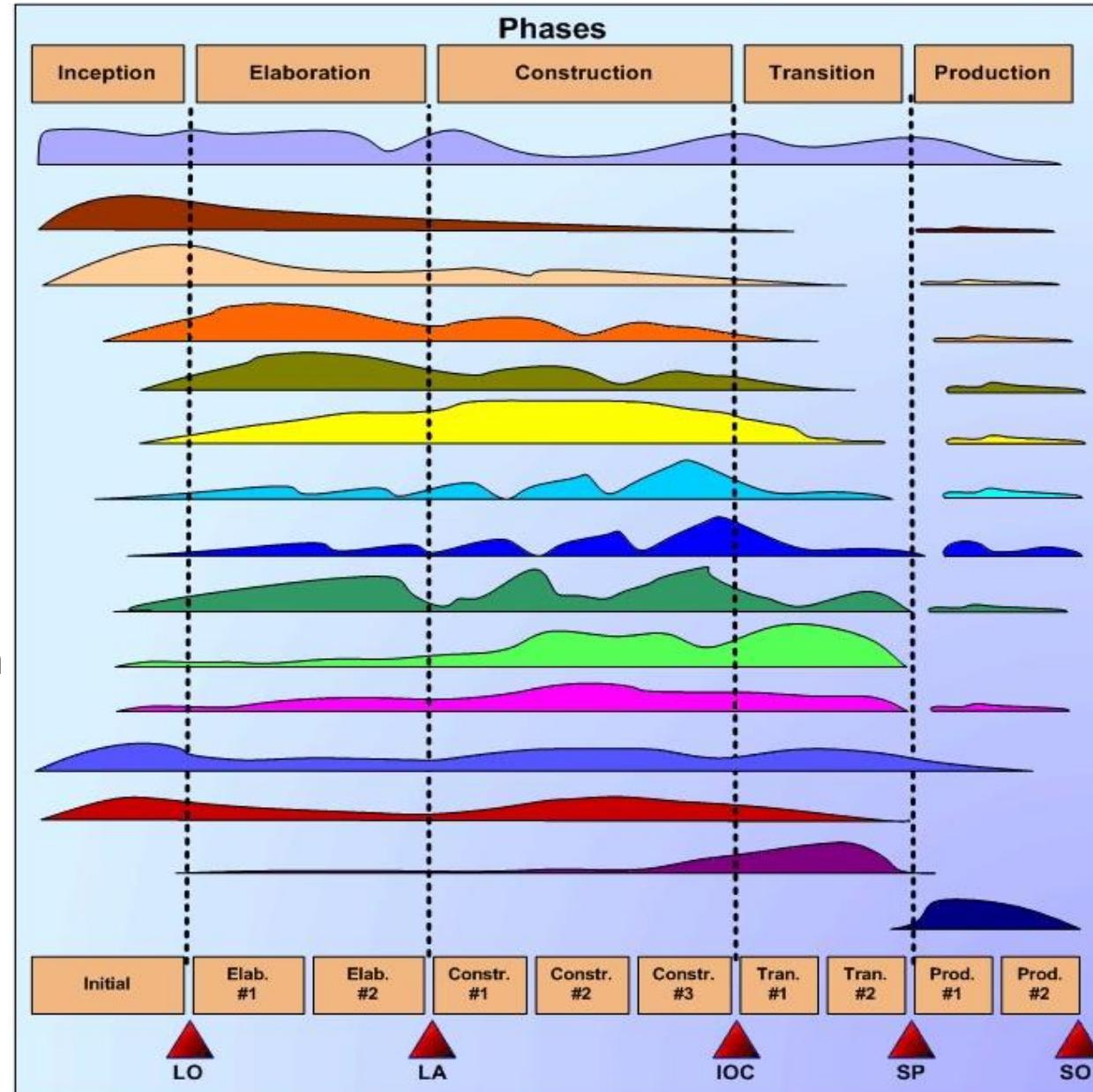
n-Cycles, 4-Phases, 5-Workflows, m-Iterations

UP에서의 개발과정(2)



UP에서의 개발과정(3)

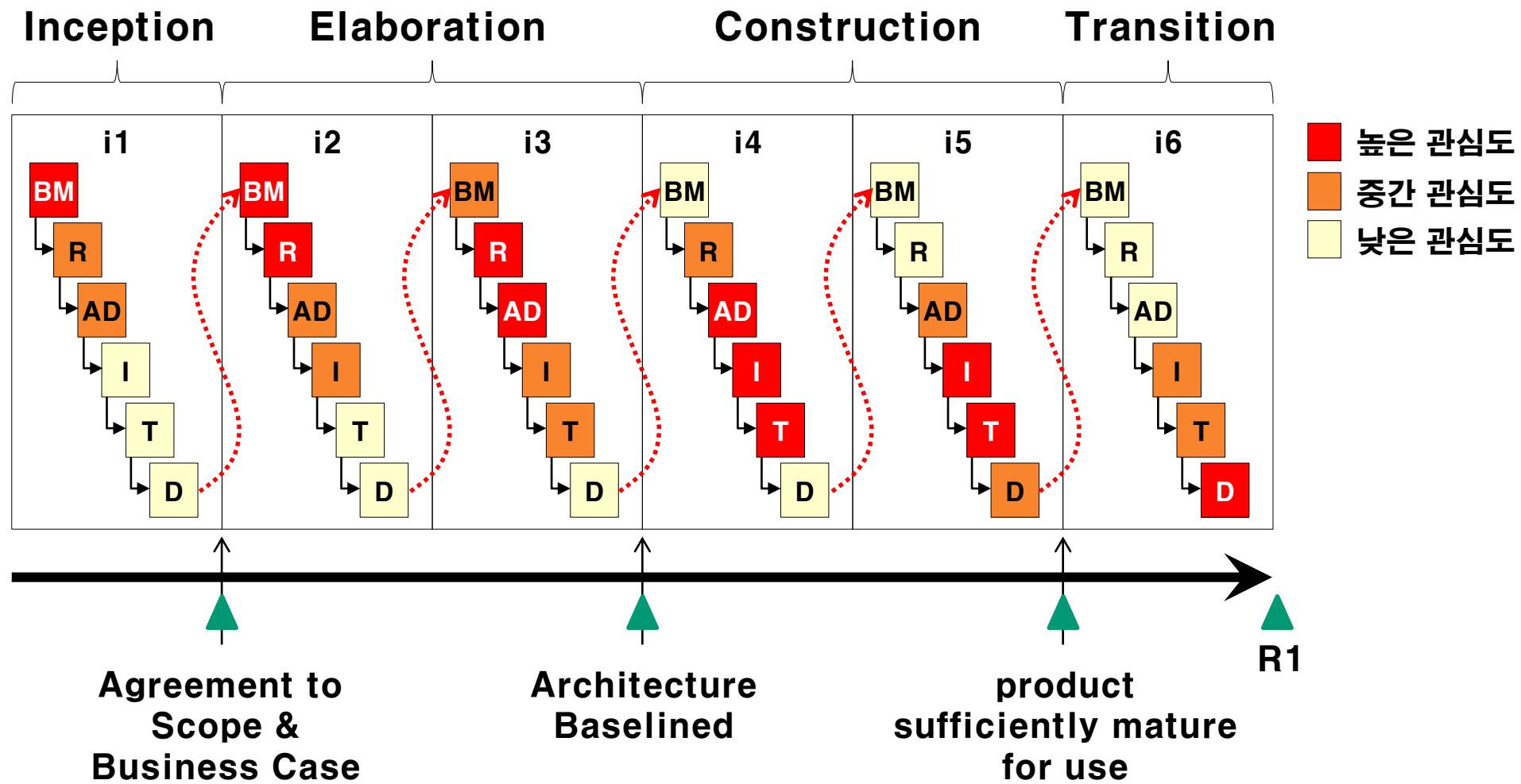
Project Management
Business Requirements
Requirements Analysis
Analysis
Design
Implementation
Testing
Performance Management
Technical Architecture
Data Acquisition & Conversion
Documentation
Change Management
Training
Transition
Operations & Support
Iterations
Milestones



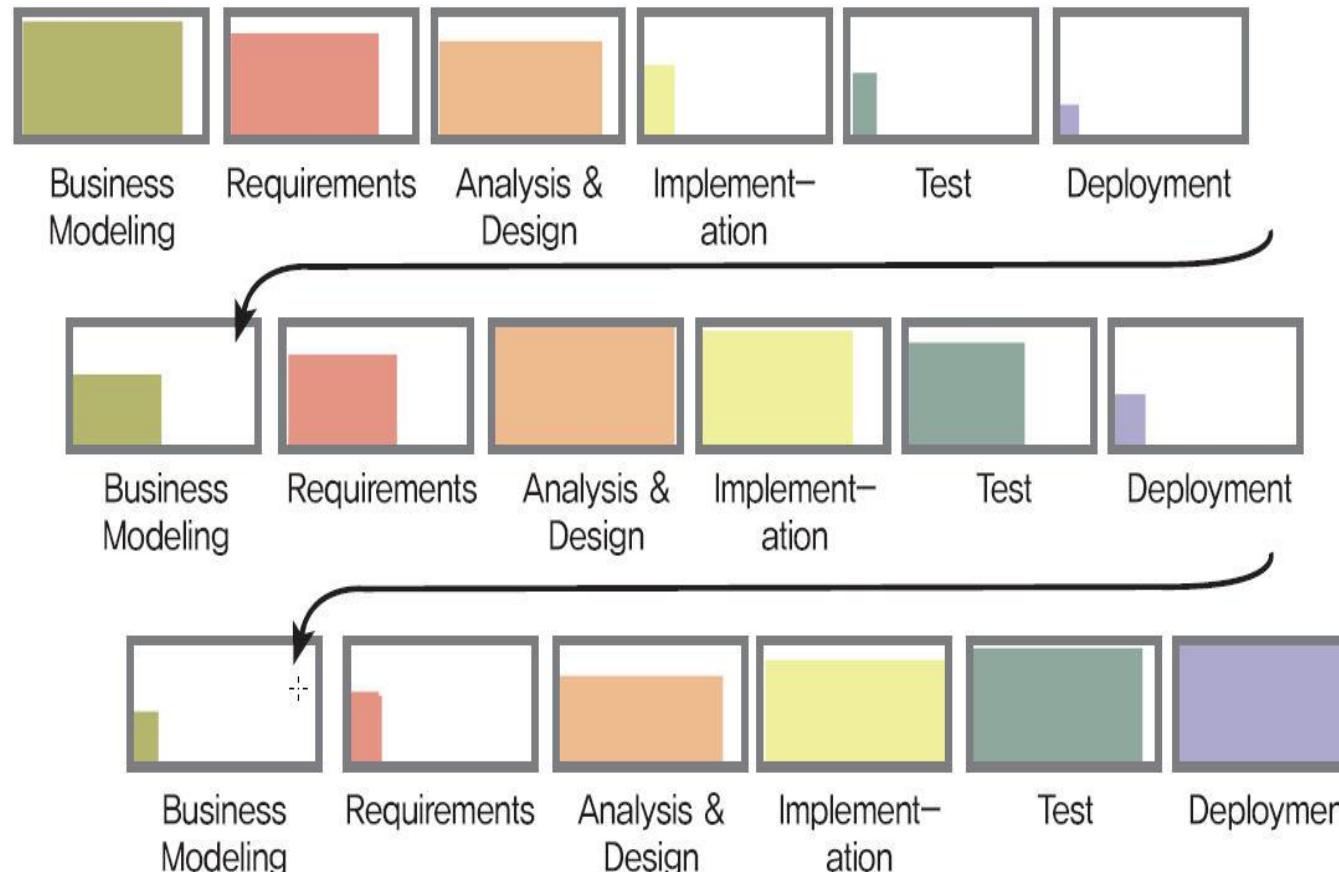
*LO(Lifecycle Objective), LA(Lifecycle Architecture), IOC(Initial Operating Capability), SP(System Production), SO(Sign-off)

UP에서의 개발과정(4)

Workflow with interest



UP에서의 개발과정(5)



Iteration #1

Iteration #2

Iteration #3

Inception Phase(1)

□ 주요 목표

- 프로젝트의 필요성을 확인하고, business case를 설정
- 프로젝트의 범위와 경계조건을 확정
- 시스템 설계상 중요한 요구를 파악하고, 이를 위한 유즈 케이스를 설정
- 시스템의 구조를 어떻게 할 것인지에 관한 대략적인 구상
- 중요한 위험 요소를 파악하고 순위를 결정
- 프로젝트의 일정계획을 수립하고, 소요 비용을 산정

□ 필수 활동

- 프로젝트 범위의 명확화, 시스템의 배경 파악, 가장 중요한 요구 및 제약사항 포착
- Business case를 준비
 - > 프로젝트 관리를 위한 다양한 대안에 대한 평가 수행
- 시스템 구조의 후보에 대한 장단점 분석
 - > 비용, 일정, 자원에 대한 소요량 산정의 기초자료 확보
 - > 개발/구입/재사용 여부를 결정

Inception Phase(2)

□ 산출물

- Vision Document = 핵심적 요구 + 중요 특성 + 주요 제약사항
- 개괄적인(10%~20%) 유즈 케이스 모델 및 용어 사전
- 초기의 business case 및 위험 평가
- 프로젝트 수행 계획: 개발 단계 및 iteration 계획은 반드시 포함

□ Milestone

- 프로젝트의 범위, 소요 비용, 일정에 대한 이해당사자 간의 합의
- 요구사항에 대한 명확한 이해
- 개발 비용, 투입 공수, 개발 일정, 최종 제품의 품질에 관한 초기 산정을 완료
- 시스템 구조에 대한 프로토타입을 개발
- 예상 비용과 실제 비용이 수용할 수 있는 수준인가?

Elaboration Phase(1)

□ 주요 목표

- 시스템의 구조에 대한 기준선(baseline)을 확정
- 시스템의 비전(vision)에 대한 기준선 확정
- 구축 단계를 위한 계획 수립
- 시스템의 구조를 통하여 비전을 달성할 수 있음을 입증

□ 핵심적인 활동

- 수행 작업시스템에 대한 대부분의 요구를 파악 -> 유즈 케이스로 기술
- 모든 위험 요소를 파악하고, business case에 대한 이들의 영향을 평가
-> 팀원에게 공시
- 시스템의 구조를 수립하고, 컴포넌트의 구현을 통하여 실행 가능성을 검증
- 프로젝트 수행을 위한 정확하고, 신뢰할 수 있는 세부 내역을 결정

Elaboration Phase(2)

□ 산출물

- 유즈 케이스 다이어그램(80% 완성), Supplementary Specifications
- Architecture Document 및 검증을 위한 프로토타입
- 클래스 다이어그램(이름, 책임), 패키지 다이어그램
- Project plan

□ Milestone

- 시스템 구축에 중대한 영향을 미치는 위험요소를 파악하고, 이에 대한 대책을 수립
- 개발 대상 유즈 케이스의 대부분을 기술
- 실행 가능한 architecture baseline을 결정
- 프로젝트 수행계획의 세부내역을 충분하게 준비
- 개발비용의 정확한 산정이 완료
- 프로젝트의 필요성을 입증할만한 business case 의 대부분을 발견

Construction Phase(1)

□ 주요 목표

- 개발 비용의 최소화
- 적절한 품질 기준을 달성
- 가장 빠른 일정으로 고객에게 인도할 시스템을 완성

□ 핵심적인 활동

- 다수의 iteration으로 세분하여 시스템의 기능을 구현
 - 유즈 케이스 기술서를 완성 -> 다음 iteration의 출발점
 - 실행 가능한 increment를 제작: 어느 정도의 결점은 허용
- 안정된 시스템의 구조를 확립
- 고객에게 넘겨 줄 시스템을 완성

Construction Phase(2)

□ 산출물

- 운용 환경에 설치 가능한 시스템
- 사용자 매뉴얼
- 최신 버전에 대한 설명서

□ Milestone

- 고객에게 인도할 수 있는 시스템의 구현
- 모든 이해 당사자가 시스템의 인수에 합의
- 모든 자원의 예상 투입량과 실제 투입량이 수용할 수 있는 수준인가?

Transition Phase(1)

□ 주요 목표

- 시스템(β -버전)을 고객의 기대 수준으로 완성
- 직전 사이클을 통하여 설치, 운용 중인 시스템과 맞물려 가동 가능
- 운용 가능한 DB 구축, 사용자 및 관리자에 대한 교육

□ 핵심적인 활동

- 테스트를 통하여 발견된 오류를 수정
- 시스템의 기능 향상을 위한 제안을 수용하여 보완
- 사용자를 위한 교육 훈련, 도움말 제공
- 시스템 설치 및 유지 보수

Transition Phase(2)

□ 산출물

- 발견된 오류 목록 및 수정 내역
- 사용자 및 시스템 관리자에 대한 교육 기록

□ Milestone

- 고객이 만족하는가?
- 모든 자원의 예상 투입량과 실제 투입량이 수용할 수 있는 수준인가?