

Iterative / Evolutionary / Agile

Unified Process

- 객체지향 시스템을 설계 할 때 가장 인기 있는 Software Development Process
- RUP(Rational Unified Process)

Idea : Iterative Development

- 개발 과정이 작고 길이가 정해진(얼마나 걸리는지) 작은 미니 프로젝트(**Iteration**)으로 정리됨
- 각 Iteration의 결과는 Tested, Integrated, Executable한 시스템임
- Iteration 별로 요구사항 분석, 디자인 구현, 테스팅을 포함함
- 시간이 지날수록 서서히 커짐
- Iterative and Incremental Development

장점

- 빠른 risk 경감
- 초반부터 눈에 보이는 진행상황
- Stakeholder들의 요구에 맞는 시스템 개발이 쉬움(빠른 피드백, 요구, 적응)
- Managed Complexity
- Iteration 과정에서 배운 것을 개발 과정 자체를 발전시킬 수 있음

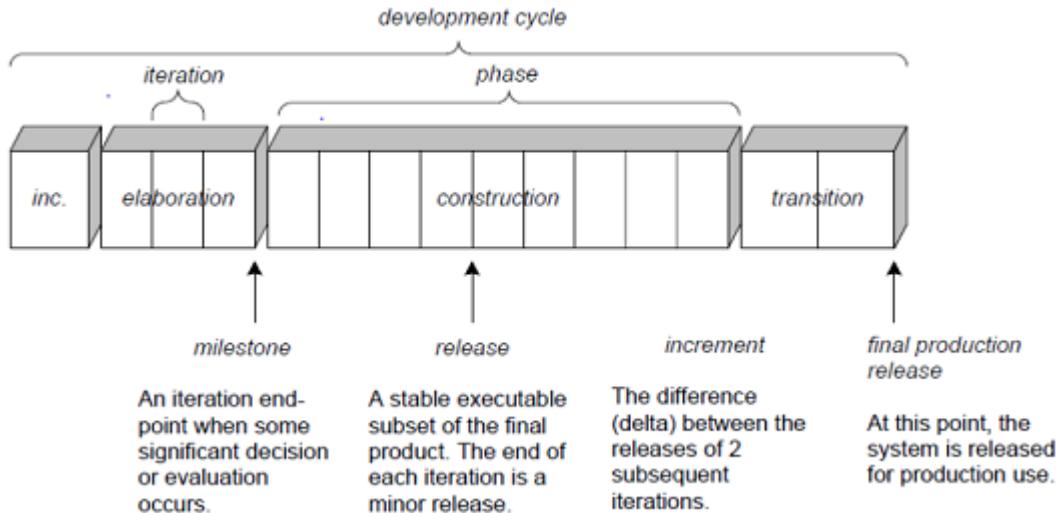
Comparison : Waterfall Model

- 라이프사이클을 세분화함
 - Requirements Specification
 - Design
 - Implementation
 - Integration
 - Testing
 - Installation
 - Maintenance
- 피드백으로 전 스텝으로 돌아 갈 수 있음

Lifecycle Phases

- Inception
 - 대략적인 비전, 비즈니스 케이스, 스코프, 대략적인 추측
- Elaboration
 - 정제된 비전, 코어 아키텍쳐의 Iterative 구현, high risk의 해결, 요구사항/스코프 식별, 현실적인 추측
- Construction
 - 남아있는 lower risk, 쉬운 요소들의 Iterative 구현, Deployment 준비
- Transition

- 베타 테스트, Deployment



Comparison : Agile Model

- 모두 Incremental, iterative
 - 작동하는 소프트웨어를 자주, 그리고 빨리 배포
 - 개발자/소비자 간의 활발한 활동
 - Self-Organizing Teams
 - 변화하는 환경에 적응하기(늦게 온 요구사항 등)
- 팀워크, 적응력, 가까운 커뮤니케이션에 집중
- 전제 : 요구사항은 프로젝트 라이프 사이클 도중 항상 바뀐다

Comparison : Model-Driven Development

- 코드가 자동으로 생성되는 도메인의 모델들을 만든다
- 사람들이 PIM(Platform Independent Model)을 만들어 PDM(Platform Definition Model)과 합쳐 작동하는 코드 생성
- PIM : 기능적 요구사항의 pure realization
- PDM : 플랫폼 세부사항, Quality Attributes

Disciplines(Workflows)

- 유즈케이스 작성 등의 Work Activity 정의
- 하나의 정해진 범위에서의 Set of Activities

Artifacts

- Business Modeling : 하나의 어플리케이션을 만들 때, Domain Object Modeling을 포함함
- Requirements : 어플리케이션의 기능적/비기능적 요구사항 분석
- Design : 설계의 모든 측면