

1

## 소프트웨어 개발방법론의 개요

## 1 소프트웨어 개발방법론이란?

### 🔍 의미

- 소프트웨어 개발 생명주기 내의 각 단계에서의 수행 방법과 활동들을 구체적으로 정의

### 🔍 종류

- UP (Unified Process)
- XP (eXtreme Programming)
- 마르미 (MaRMI: Magic and Robust Methodology Integrated)

2

## UP 방법론

## 02 UP 방법론

### 2 UP (Unified Process)

#### 🔍 개요

- Jacobson, Booch, Rumbaugh에 의해 개발된 객체지향 소프트웨어 개발 방법론
- 소프트웨어 개발 단계를 시간의 순서에 따라 네 개의 범주(Initiation, Elaboration, Construction, Transition)로 나누고, 각 범주에는 요구사항 도출부터 평가까지 개발 생명주기가 포함되어 있음

## 02 UP 방법론

### 2 UP (Unified Process)



- 반복적(Iterative)이고,  
점진적(Incremental)으로 개발
- ✓ 요구사항 분석, 설계, 구현  
그리고 평가의 한 사이클이 여러 번 반복되어 개발
- ✓ 반복되는 과정을 통해서 실행 가능한 Release가  
산출되어, 결국 최종 시스템으로 발전

## 02 UP 방법론

### 2 UP (Unified Process)



- 유스케이스(Usecase)를 기반으로 함
  - ✓ 요구사항을 식별하고, 정의하는데 있어서 UML의 유스케이스 사용
  - ✓ 유스케이스에 대한 자세한 설명은 4장 참고

## 02 UP 방법론

### 2 UP (Unified Process)



- 아키텍쳐 (Architecture) 중심의 개발을 지향
  - ✓ 시스템 전체를 표현한 아키텍쳐는 프로젝트 참여자들에게 최종 산출물의 모습을 인지하게 하고, 구성원들을 공통된 시각을 갖도록 함

## 02 UP 방법론

### 2 UP (Unified Process)

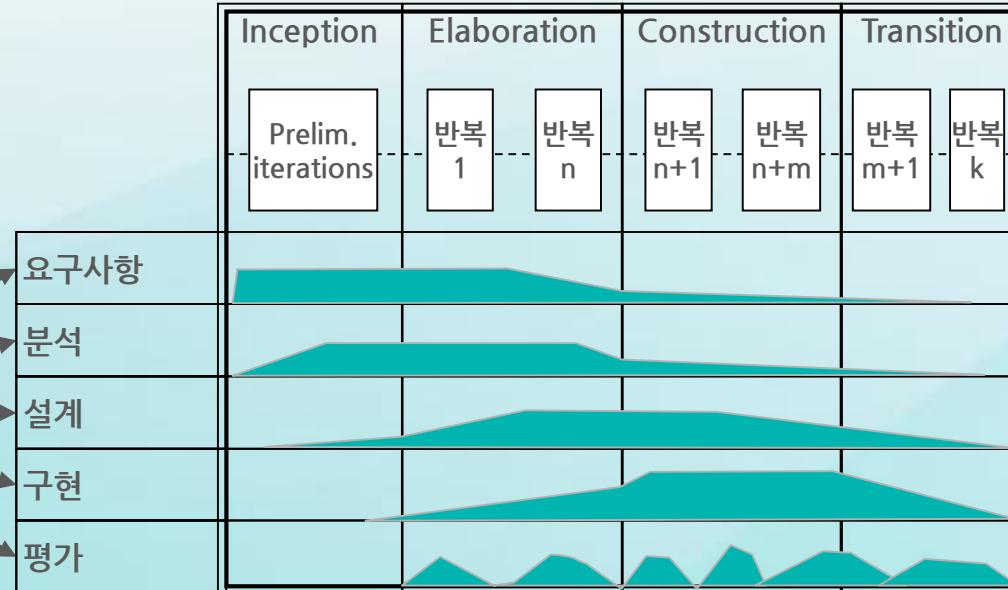


#### ■ 위험 관리를 중시

- ✓ 프로젝트 성공에 장애가 될 수 있는 위험요소들을 파악하고, 위험도가 높은 것일수록 프로젝트 초기에 처리 방안을 찾아 해결

## 02 UP 방법론

### 2 UP (Unified Process)



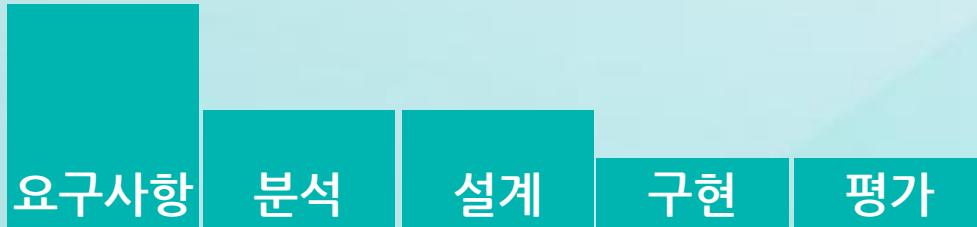
※ 출처 : 소프트웨어공학의 소개, 한혁수저, 홍릉과학출판사

## 02 UP 방법론

### 2 UP (Unified Process)

#### 특징

- 전체 요구사항을 대략적으로 이해하는데 중점을 둠
- 구현 및 평가에 대한 비중은 상대적으로 낮음
- 프로젝트 목표와 실현 가능성 그리고 대략적인 비용 평가를 통해 프로젝트 개발 여부를 결정하는 단계



[도입 단계의 반복 워크 플로우]

## 02 UP 방법론

### 2 UP (Unified Process)

#### 🔍 상세(Elaboration)

- 요구사항 분석 및 아키텍처를 확정하고, 위험 요소를 해결하는데 중점을 둠
- 아키텍처를 실행 가능한 수준으로 확장하며, 구축 단계에 대한 계획을 수립
- 구현 및 평가에 대한 비중은 상대적으로 낮으나, 시스템의 중요한 기능에 대한 구현 및 평가는 이뤄짐



[상세 단계의 반복 워크 플로우]

## 02 UP 방법론

### 2 UP (Unified Process)

#### 구축(Construction)

- 사용자의 환경에서 실행 가능한 시스템을 구축하고 평가하는 데 중점을 둠
- ✓ 시스템에 필요한 모든 컴포넌트 및 기능 등이 개발되고 평가됨
- 요구사항 분석 및 설계에 대한 비중은 상대적으로 낮음



[구축 단계의 반복 워크 플로우]

## 02 UP 방법론

### 2 UP (Unified Process)

#### 이행(Transition)

- 제품 릴리즈 완성 단계로서 시스템 개발을 완료하고 그에 따른 품질을 보장하여 사용자에게 인도하는데 중점을 둠
- 사용자 환경에서 인수 테스트가 수행되고 시스템에 대한 사용자 교육 및 훈련이 수행



[이행 단계의 반복 워크 플로우]

## 02 UP 방법론

### 2 UP (Unified Process)

#### 🔍 이점

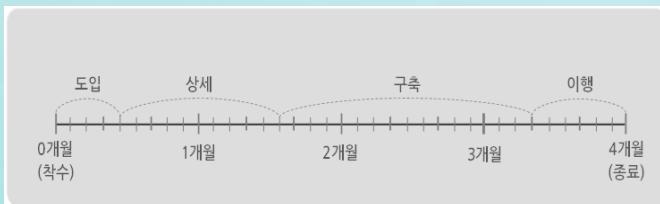
- 기술적 또는 요구사항 변경 등에 관한 위험요소를 초기에 완화 시킬 수 있음
- 진척 사항을 가시화할 수 있음
- 발주자의 실제 요구사항에 근접한 시스템을 만들 수 있음
- 이전 반복을 통해 얻은 교훈은 다음 반복의 피드백으로 작용하여 반복이 거듭될 수록 개선된 소프트웨어 개발이 가능

## 02 UP 방법론

### 3 [예] UP를 활용한 쇼핑몰 웹 사이트 개발

#### 개요

- 4개월 동안 수행되는 쇼핑몰 웹 사이트 개발 프로젝트



#### 도입(Inception)

- 목표, 실현가능성 및 비용 평가
- 간단한 유스케이스 작성
- 관리자모드, 판매 요구사항에 대한 프로토타입 구축 및 평가

#### 상세(Elaboration)

- 유스케이스 작성을 통한 전체 요구사항 정의
- 요구사항 분석
- 아키텍처 확정
- 위험요소 식별 및 해결
- 주요 기능 구현 및 평가
- 구축 단계의 계획 작성

#### 구축(Construction)

- 대부분의 쇼핑몰 웹사이트 구축 및 평가

#### 이행(Transition)

- 인수테스트 수행
- 웹사이트 관리자 교육 및 훈련

※ 출처 : 소프트웨어공학의 소개, 한혁수저, 홍릉과학출판사

## 02 UP 방법론

### 3 [예] UP를 활용한 쇼핑몰 웹 사이트 개발

#### 🔍 단계 별 활동

- 도입 단계 (기간: 2주)
  - ✓ 쇼핑몰 웹사이트의 목표인 온라인 판매 및 개발하고자 하는 관리자 모드, 판매, 결제, 배송 등의 요구사항이 실현 가능한지를 평가
  - ✓ 요구사항에 대한 전반적인 이해를 위해 유스케이스 및 이에 대한 프로토 타입을 작성하고 평가

## 02 UP 방법론

### 3 [예] UP를 활용한 쇼핑몰 웹 사이트 개발

#### 단계 별 활동

- 상세 단계 (기간: 4주)
  - ✓ 전체 요구사항을 정의 및 분석하고 아키텍처를 정의하여 쇼핑몰 웹사이트에 대한 구조를 확정하고 평가하고 위험 요소를 식별하여 완화시키는 노력을 함

## 02 UP 방법론

### 3 [예] UP를 활용한 쇼핑몰 웹 사이트 개발

#### 🔍 단계 별 활동

- 구축 단계 (기간: 8주)
  - ✓ 구성된 쇼핑몰 웹사이트 아키텍처를 구현하는 단계로 각각의 기능이 짧은 반복을 통해 구축되고, 각 반복이 종료될 때마다 실행 가능한 기능이 결과물로 나타남

## 02 UP 방법론

### 3 [예] UP를 활용한 쇼핑몰 웹 사이트 개발

#### 단계 별 활동

- 이행 단계 (기간: 3주)
  - ✓ 구축된 쇼핑몰 웹 사이트를 고객의 사용 환경에서 평가하고 담당자에 대한 교육 및 훈련을 실시

3

## XP 방법론

## 03 XP 방법론

### 1 XP(eXtreme Programming)

#### 🔍 개요

- 1990년대 초,  
Kent Beck에 의해 고안된 개발 방법론
- 요구사항 변경으로 인한 비용이 개발 기간에  
상관없이 일정하게 유지되도록 하는 것을  
주목적으로 함

## 03 XP 방법론

### 1 XP(eXtreme Programming)

#### 특징

- 요구사항이 변경된다는 것을 가정하고, 고객의 피드백을 수용하기 위해 고객과 개발 팀이 함께 상주
- 동료 프로그래머와의 의사소통을 중요시 함
- 단순하고 명확한 설계 유지
- 가장 우선순위가 높은 것을 먼저 개발함
- 되도록 초기에 고객에게 시스템을 전달하여 피드백을 받음
- 프로그래머는 요구사항과 기술의 변경에 용감하게 대응할 수 있음

## 03 XP 방법론

### 2 XP에서 사용하는 용어

용 어	설 명
스토리(Story)	고객이 직접 작성하는 요구사항
스토리 추정	개발자가 고객이 제시한 스토리가 어느 정도의 난이도 또는 기간인지를 결정하는 것
릴리즈(Release)	고객에게 구현된 제품을 배포하는 것
반복(Iteration)	하나의 릴리즈 안에 반복되는 작업

## 03 XP 방법론

### 2 XP에서 사용하는 용어

용 어	설 명
드라이버(Driver)	Pair Programming에서 키보드를 치면서 코드를 작성하는 사람
파트너(Partner)	Pair Programming에서 드라이버를 도와 코드의 구조 및 결함에 대한 조언을 하는 사람

## 03 XP 방법론

### 2 XP에서 사용하는 용어

역 할	활동 설명
개발자	<ul style="list-style-type: none"><li>스토리 난이도 추정</li><li>소프트웨어 설계, 구현, 테스트 실시</li></ul>
관리자	<ul style="list-style-type: none"><li>개발자 앞에 놓인 개발 이외의 장애물 제거</li><li>개발 활동에 직접 관여하지 않고 개발 활동을 지시, 정리 및 결과 보고</li></ul>

## 03 XP 방법론

### 2 XP에서 사용하는 용어

역 할	활동 설명
고객	<ul style="list-style-type: none"><li>• 스토리를 제출하고, 스토리 구현 순서를 결정</li><li>• 구현된 스토리의 인수 테스트 실시</li><li>• 개발팀과 함께 상주</li></ul>

## 03 XP 방법론

### 3 XP의 개발 활동

- 🔍 고객이 작성한 스토리를 기반으로 개발자와 고객은 스토리 추정을 통하여 전체 프로젝트가 몇 번의 릴리즈로 구성될 것인지 계획
- 🔍 하나의 릴리즈 시작 시에는 해당 릴리즈에서 개발될 스토리가 결정되고, 해당 릴리즈가 몇 번의 반복을 거칠 것인가를 결정
  - 하나의 반복은 계획, 구현, 통합, 테스트의 Task로 구성
  - 전체 태스크가 종료되면 해당 반복의 하나의 스토리가 종료되는 것이고, 전체 반복이 종료되면 해당 릴리즈 하나가 종료

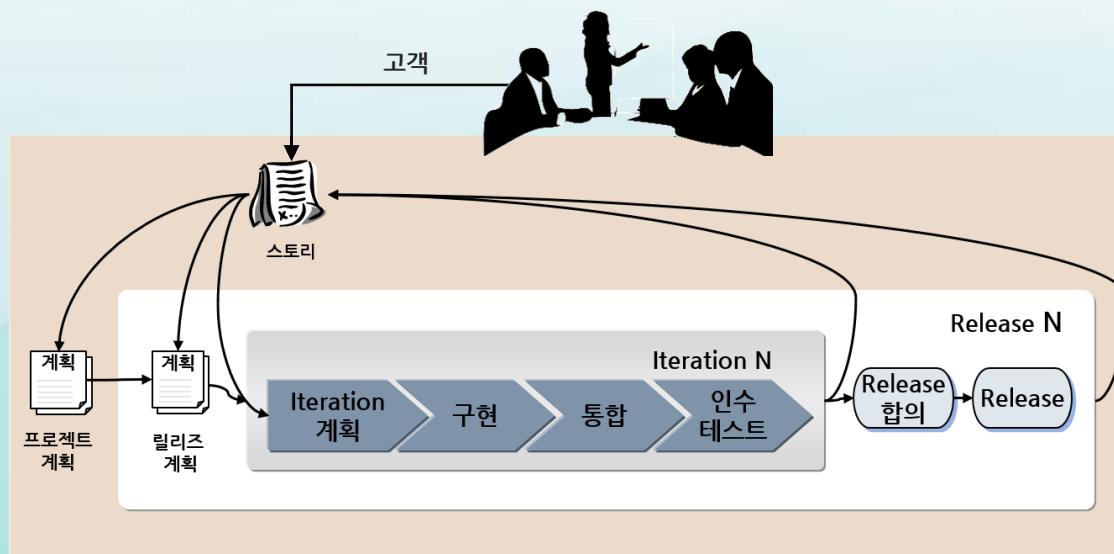
## 03 XP 방법론

### 3 XP의 개발 활동

- 🔍 구현 시에는 Pair Programming을 통한 구현과 단위 테스트가 수행
- 🔍 구현된 부분을 전체 시스템에 통합하는 과정을 거쳐 전체 시스템의 테스트를 수행하고 마지막으로 고객의 인수테스트를 수행

# 03 XP 방법론

## 3 XP의 개발 활동



※ 출처 : 소프트웨어공학의 소개, 한혁수저, 홍릉과학출판사

## 03 XP 방법론

### 4 XP의 가치

가치	설명
단순성 (Simplicity)	많은 양의 문서를 작성하는 것이 아닌 시스템 구조에 대한 큰 그림에 대해 개발팀이 합의한 후 구현하여 설계의 단순함을 실현
의사소통 (Communication)	고객 및 동료 개발자와의 원활한 의사소통을 위해 고객이 항상 개발팀에 상주하여야 하며 문서보다는 구두에 의한 의사소통을 중요
피드백 (Feedback)	개발자가 수행하는 지속적인 단위테스트와 릴리즈를 위한 고객의 인수테스트 시 발견된 부적합 사항에 대해, 개발팀이 피드백을 받음

## 03 XP 방법론

### 4 XP의 가치

가치	설명
용기 (Courage)	위의 세 가지 가치들을 꾸준히 이행할 수 있도록 개발팀이 가져야 하는 가치
존중 (Respect)	프로젝트에 포함된 모든 사람이 프로젝트에 기여함을 인정하고 존중함

## 03 XP 방법론

### 5 XP의 13가지 실천 사항

- 🔍 계획 게임(Planning Game)
- 🔍 짧은 릴리즈(Small Release)
- 🔍 메타포(Metaphor)
- 🔍 단순 설계(Simple Design)
- 🔍 테스트 우선 개발(Test-First Development)
- 🔍 리팩토링(Refactoring)
- 🔍 짹 프로그래밍(Pair Programming)

## 03 XP 방법론

### 5 XP의 13가지 실천 사항

- 🔍 공동 코드 소유(Collective Code Ownership)
- 🔍 지속적인 통합(Continuous Integration)
- 🔍 주당 40시간 업무(40 hour Week)
- 🔍 고객의 참여(On-site Customer)
- 🔍 코딩 표준(Coding Standard)
- 🔍 전체 팀(Whole Team)

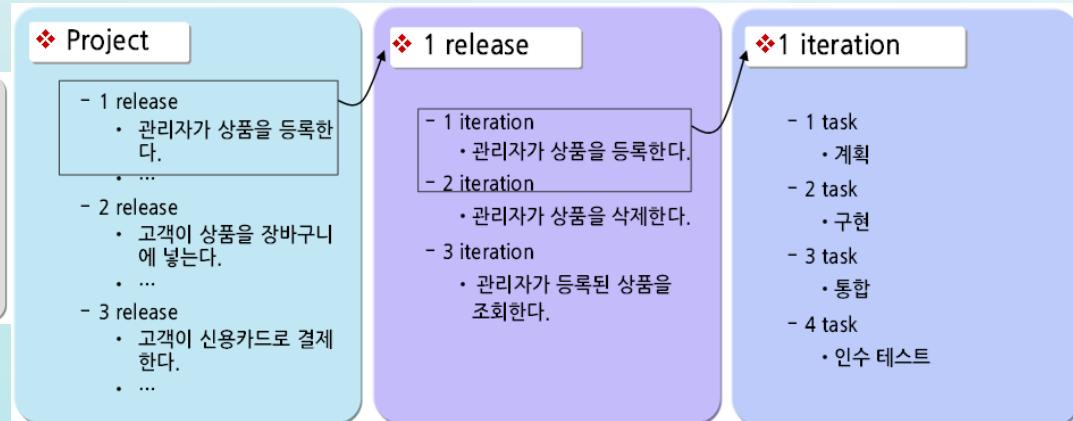
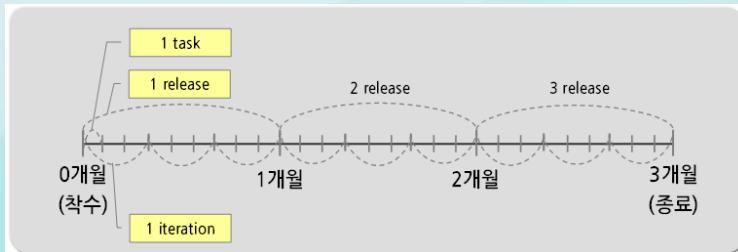
## 03 XP 방법론

### 6 [예] XP를 활용한 쇼핑몰 웹 사이트 개발



개요

- 3개월 동안 수행되는 쇼핑몰 웹 사이트 개발 프로젝트



[고객이 정한 우선순위에 따라, 우선순위가 높은 요구사항부터 릴리즈를 구성]

※ 출처 : 소프트웨어공학의 소개, 한혁수저, 홍릉과학출판사

4

## MarMI 방법론

## 1

## 마르미 (MaRMI)



- 마르미(MaRMI: Magic and Robust Methodology Integrated)
- 한국전자통신연구원(ETRI: Electronics and Telecommunications Research Institute)의 소프트웨어 공학 연구팀에서 국내 여건을 반영하여 개발한 한국형 소프트웨어 개발 방법론

## 1

## 마르미 (MaRMI)

- 구조적 방법의 소프트웨어 개발을 지원하는 방법론
- 관련 국제 표준인 ISO12207을 수용하여 개발 프로세스를 계층화하고 상세화 한 정보시스템 구축 방법론
- 개발과 관리를 결합하고 산출물을 간소화하여 개발 조직이 체계적으로 소프트웨어를 개발 및 관리할 수 있도록 함

## 04 MarMI 방법론

### 2 마르미-II

- UML 기반 객체지향 시스템 개발을 지원하는 방법론
- 반복적이고 점진적인 개발 프로세스 그리고 위험 관리 등에 대한 방안을 실제 소프트웨어 개발 현장에서 활용할 수 있는 실용적인 방법으로 제시함

## 04 MarMI 방법론

### 3 마르미-III

- 🔍 컴포넌트 기반의 소프트웨어 개발(CBD: Component Based Development)을 지원하는 방법론
  - CBD: 전체 소프트웨어를 이루는 여러 컴포넌트를 먼저 개발하고 이들을 조립하여 하나의 소프트웨어를 개발하는 방법론
- 🔍 버전 1.0 : 처음으로 발표한 공개 버전
- 🔍 버전 2.0 : J2EE 지원
- 🔍 버전 3.0 : 닷넷(.NET) 지원
- 🔍 버전 4.0 : 웹 서비스 기반 개발 방법의 지침을 보강

## 4

## 마르미-RE

- ‘재공학(Re-Engineering)’에 필요한 지침을 제공하는 방법론
  - 코볼 등의 프로그래밍 언어로 개발된 소프트웨어를 재활용하여 컴퓨터 기반의 소프트웨어로 변환함으로써, 신규 시스템 개발 시 기존의 주요 부분들을 체계적으로 활용하도록 함