

## 2. 소프트웨어 프로세스와 생명주기

---

# 주요 내용

---

- ❖ 프로세스란?
- ❖ 소프트웨어 프로세스란?
- ❖ 소프트웨어 생명주기란?
- ❖ 소프트웨어 생명주기 모델의 종류
- ❖ 소프트웨어 개발방법론이란?
- ❖ 소프트웨어 개발방법론의 종류

# 목차

---

## ❖ 강의 내용

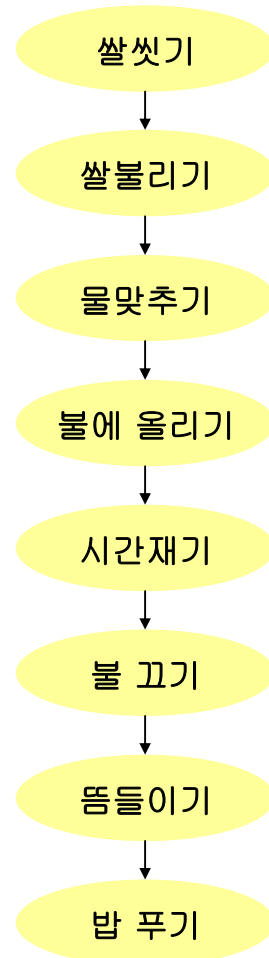
- 프로세스
- 소프트웨어 프로세스
- 소프트웨어 생명주기
- 소프트웨어 생명주기 모델의 종류
- 소프트웨어 개발 방법론
- 소프트웨어 개발 방법론의 종류

## ❖ 팀 프로젝트(3주차)

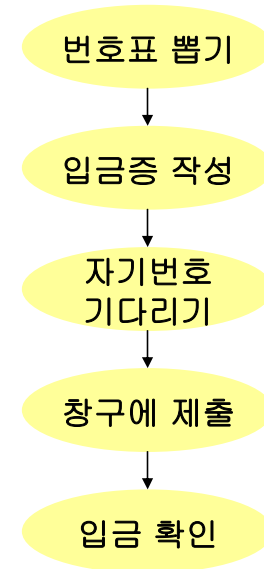
- 팀 구성 확정

# 일상 생활에서의 프로세스

---



재래식 밥 짓기 프로세스



은행 입금 프로세스

# 프로세스

---

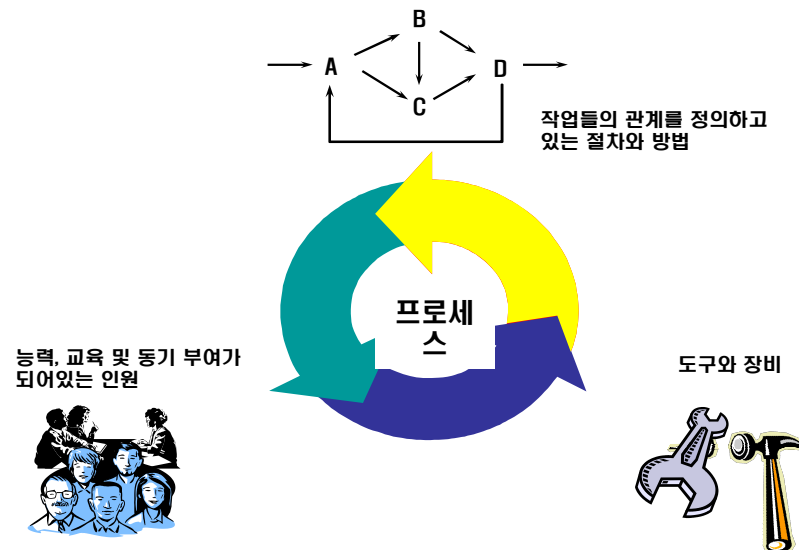
# 프로세스란?

## ❖ 의미

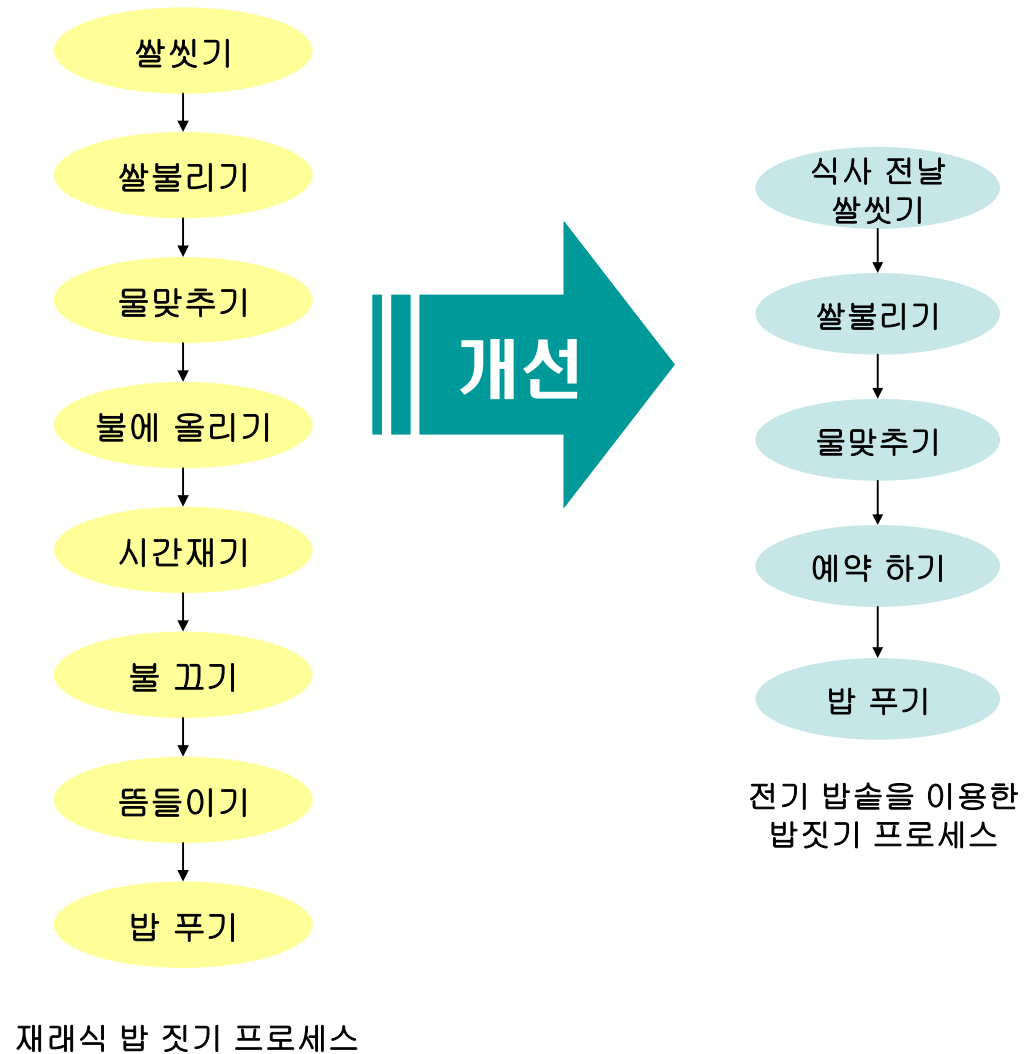
- 주어진 목적을 위해 수행되는 일련의 절차

## ❖ 역할

- 절차, 인력, 기술을 통합
- 각 순서와 활동이 명확하게 정의됨
  - 프로세스를 사용하는 직원들의 공통된 행동 양식을 지정해주는 역할



# IT를 활용한 프로세스 개선 사례



# 소프트웨어 개발 프로세스

---



# 소프트웨어 개발 프로세스의 중요성

---

## ❖ 소프트웨어 개발의 목표

- 정해진 기한 내에, 주어진 예산을 이용해 사용자가 원하는 좋은 품질로 개발하는 것

## ❖ 계속되는 프로젝트 실패

- 소프트웨어의 요구사항이 복잡해지고 규모가 점점 커짐
- 정해진 기간 내에 고품질의 소프트웨어를 개발하는 것이 점점 더 어려워짐

## ❖ 소프트웨어 개발 프로세스의 중요성

- 소프트웨어 제품의 품질은 그 제품을 만들기 위해 사용된 프로세스의 품질에 의해 결정된다 [Watts S. Humphrey]

# 소프트웨어 개발 프로세스

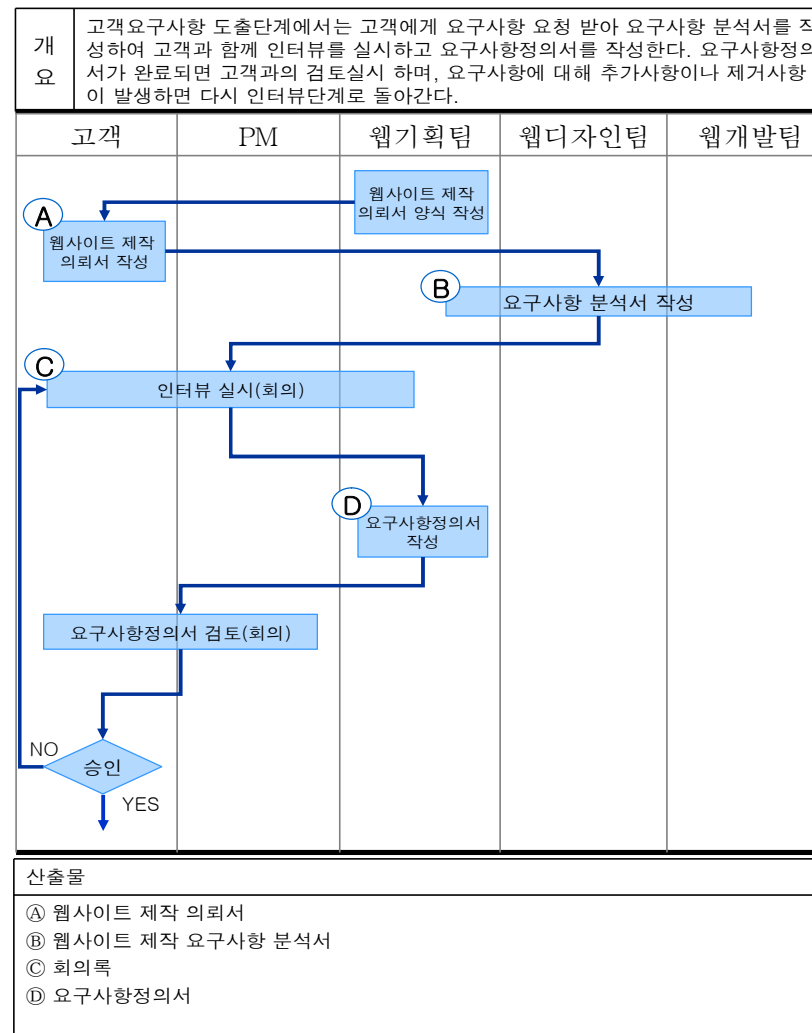
## ❖ 정의

- 소프트웨어 개발에 필요한 절차만이 아니라, 그와 관련된 인력, 방법, 도구 등이 통합되는 수단
- 소프트웨어와 이에 관련된 산출물을 개발, 유지하기 위해 사용하는 활동, 방법, 절차의 집합

자료원	소프트웨어 프로세스 정의
IEEE-STD-610	주어진 목적을 달성하기 위한 순서적인 절차 틀
Olson et al.(1989)	특정한 목표나 목적을 달성하기 위한 활동, 작업 및 절차들의 집합
SEI CMM (Humphrey, 1989: Paulk et al., 1993)	소프트웨어의 생산 및 진화에 사용되는 활동, 방법 및 실무 활동 들의 집합 인력, 절차, 방법, 장치 및 도구들이 원하는 산출물을 생산할 수 있도록 통합하는 수단

# 웹사이트 제작 '고객 요구사항 도출 프로세스' 사례 [1/3]

## ❖ 고객 요구사항 도출 프로세스의 Swim Lane 예



# 웹사이트 제작 ‘고객 요구사항 도출 프로세스’ 사례 (2/3)

---

## ❖ 고객 요구사항 도출 프로세스의 수행 활동

- 웹 사이트 제작 의뢰서 작성
  - 고객 입장에서 요구하는 웹 사이트의 기능을 의뢰서에 작성
- 인터뷰 실시
  - 개발자와 고객이 질문 · 답변 형식으로 인터뷰를 진행
  - 고객의 정확하고, 상세한 요구사항을 도출하기 위한 목적으로 시행
- 요구사항 정의서 작성
  - 요구사항 분석서와 인터뷰 결과 등을 바탕으로 고객의 요구사항을 정의하여 문서화
- 요구사항 정의서 검토 및 승인 여부 결정
  - 고객과 개발자는 요구사항 정의서를 검토하고, 요구사항에 대해 합의할 지를 결정

# 웹사이트 제작 '고객 요구사항 도출 프로세스' 사례 (3/3)

## ❖ 요구사항 정의서의 예

요구사항 명	상세요구사항	우선순위	비고
관리자모드	사이트 수정 및 업데이트, 게시판 관리	1	관리자 사전교육 실시 요망
공지 게시판	글 등록, 수정, 삭제, 조회	2	관리자만 가능
회원 문의 게시판	글 등록, 수정, 삭제, 조회	2	로그인 회원만 가능
게시판 답글	회원문의 게시판에 관리자 답글 가능	2	공지 게시판 답글 불가
회원가입	회원 정보로서 이름, 전화번호, 직업, 관심분야, 회사, E-Mail 필요	1	중복회원가입 불가
메일링	메일링 체크 회원에게 교육, 컨퍼런스 정보 발송	1	
메뉴	회사 소개, 교육 소개, 컨퍼런스, 공지사항이 있어야 함	1	
최신 공지사항	메인 화면에는 최신 글 표시해야 함	2	
교육 및 컨퍼런스 공지	새로운 교육과 컨퍼런스 개최 시 메인 화면에 공지	1	

# 소프트웨어 개발 생명주기

---

# 소프트웨어 개발 생명주기 [Software Development Life Cycle]

---

## ❖ 의미

- 소프트웨어를 어떻게 개발할 것인가에 대한 추상적 표현
- 순차적 또는 병렬적 단계로 구성됨
- 개발 모델 또는 소프트웨어 공학 패러다임이라고도 함

## ❖ 특징

- 개발 생명주기의 각 단계에 관련된 활동들이 정의되어 있음
- 단계별 활동들을 통해 다음 단계에 활용될 수 있는 산출물이 작성됨
- 전체 프로젝트의 비용 산정과 개발 계획을 수립할 수 있는 기본 골격 제시
- 참여자들 간에 의사소통의 기준과 용어의 표준화를 가능하게 함
- 문서화가 충실한 프로젝트 관리를 가능하게 함

# 소프트웨어 개발 생명주기 모델의 종류

---

- ❖ 주먹구구식 개발 모델(Build-Fix Model)
- ❖ 폭포수 모델(Waterfall Model)
- ❖ 원형 모델(Prototyping Model)
- ❖ 나선형 모델(Spiral Model)



# 주먹구구식 개발 모델(Build-Fix Model) [1/2]

---

## ❖ 개요

- 요구사항 분석, 설계 단계 없이 일단 개발에 들어간 후 만족할 때까지 수정작업 수행

## ❖ 적용 가능한 경우

- 크기가 매우 작은 규모의 소프트웨어 개발

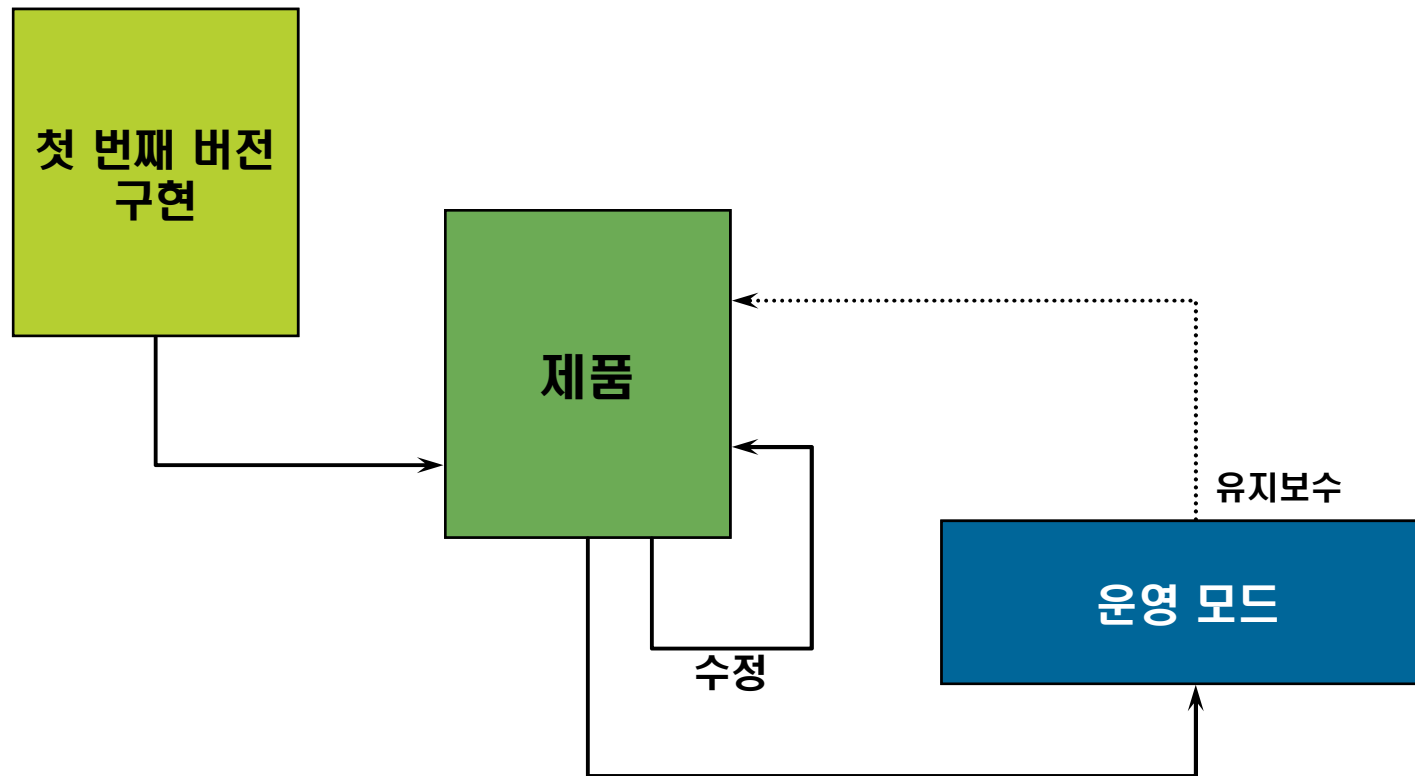
## ❖ 단점

- 정해진 개발 순서가 없기 때문에
  - 계획이 정확하지 않음
  - 관리자는 프로젝트 진행 상황 파악에 어려움
  - 개발 문서가 없기 때문에 개발 및 유지보수에 어려움

➡ 이후 체계적인 소프트웨어 개발 생명주기 모델의 연구를 가져옴

## 주먹구구식 개발 모델(Build-Fix Model) (2/2)

---



# 폭포수 모델(Waterfall Model) (1/4)

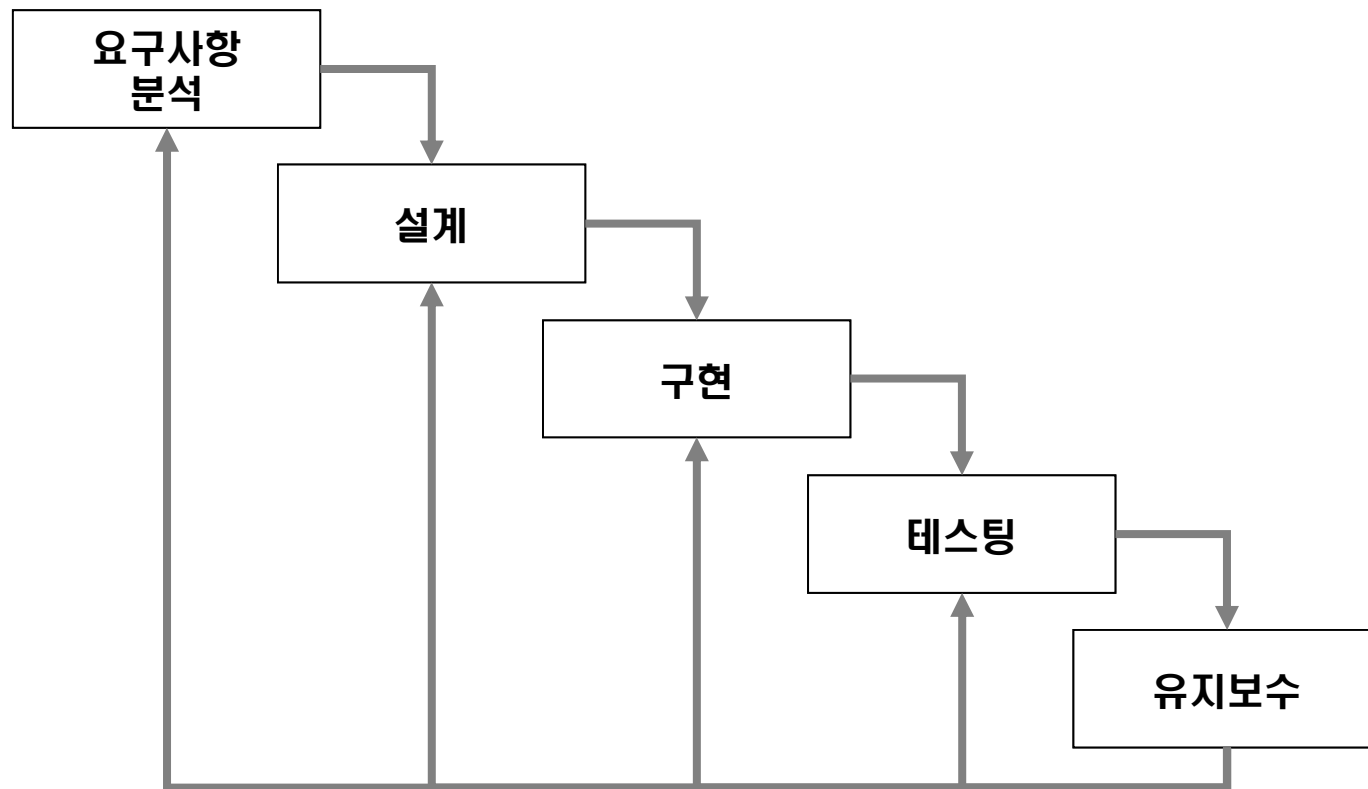
---

## ❖ 개요

- 순차적으로 소프트웨어를 개발하는 전형적인 개발 모델
- 대부분의 소프트웨어 개발 프로젝트의 기본적 모델이며 가장 많이 사용되는 모델
- 소프트웨어 개발의 전 과정을 나누어 체계적이고 순차적으로 접근하는 방법
  - 개발 과정: 요구사항 분석, 설계, 구현, 테스트, 유지보수

## 폭포수 모델(Waterfall Model) (2/4)

---



# 폭포수 모델(Waterfall Model) (3/4)

---

## ❖ 단계별 활동

### - 요구사항 분석

- 개발하고자 하는 소프트웨어에 대한 요구사항 수집, 문제 이해 및 분석 단계
- 소프트웨어 엔지니어 또는 분석가: 고객의 요구사항을 기능, 성능, 인터페이스 등으로 파악하고 문서화
- 산출물: 요구사항 명세서(Requirement Specification)

### - 설계

- 프로그램의 데이터 구조, 소프트웨어 구조, 인터페이스 구조, 알고리즘 등 모든 시스템의 구조 결정
- 산출물: 설계 명세서

### - 구현

- 설계 명세서를 시스템의 실제 모습으로 변환 시키는 것
- 산출물: 소스 코드 및 프로그램

### - 테스트

- 프로그램이 입력에 따라 요구되는 결과대로 작동하는지, 내부적 이상 여부 및 오류 발견을 위해 수행
- 테스트 계획을 세운 후 문서화

### - 유지보수

- 개발된 소프트웨어의 변경사항을 수정하는 것
- 수정 유지보수, 적응 유지보수, 기능 추가 유지보수 등이 있음

# 폭포수 모델(Waterfall Model) (4/4)

---

## ❖ 장점

- 각 단계별로 정형화된 접근 방법 가능
- 체계적인 문서화가 가능하여 프로젝트 진행을 명확하게 할 수 있음

## ❖ 단점

- 앞 단계가 완료될 때까지 다음 단계들은 대기 상태여야 함
- 실제 작동되는 시스템을 개발 후반부에 확인 가능하기 때문에 고객이 요구사항 확인하는데 많은 시간이 걸림

# 원형 모델(Prototyping Model) (1/3)

---

## ❖ 개요

- 폭포수 모델의 단점을 보완한 모델
- 점진적으로 시스템을 개발해 나가는 접근 방법
- 원형(Prototype)을 만들어 고객과 사용자가 함께 평가한 후 개발될 소프트웨어의 요구사항을 정제하여 보다 완전한 요구사항 명세서를 완성함

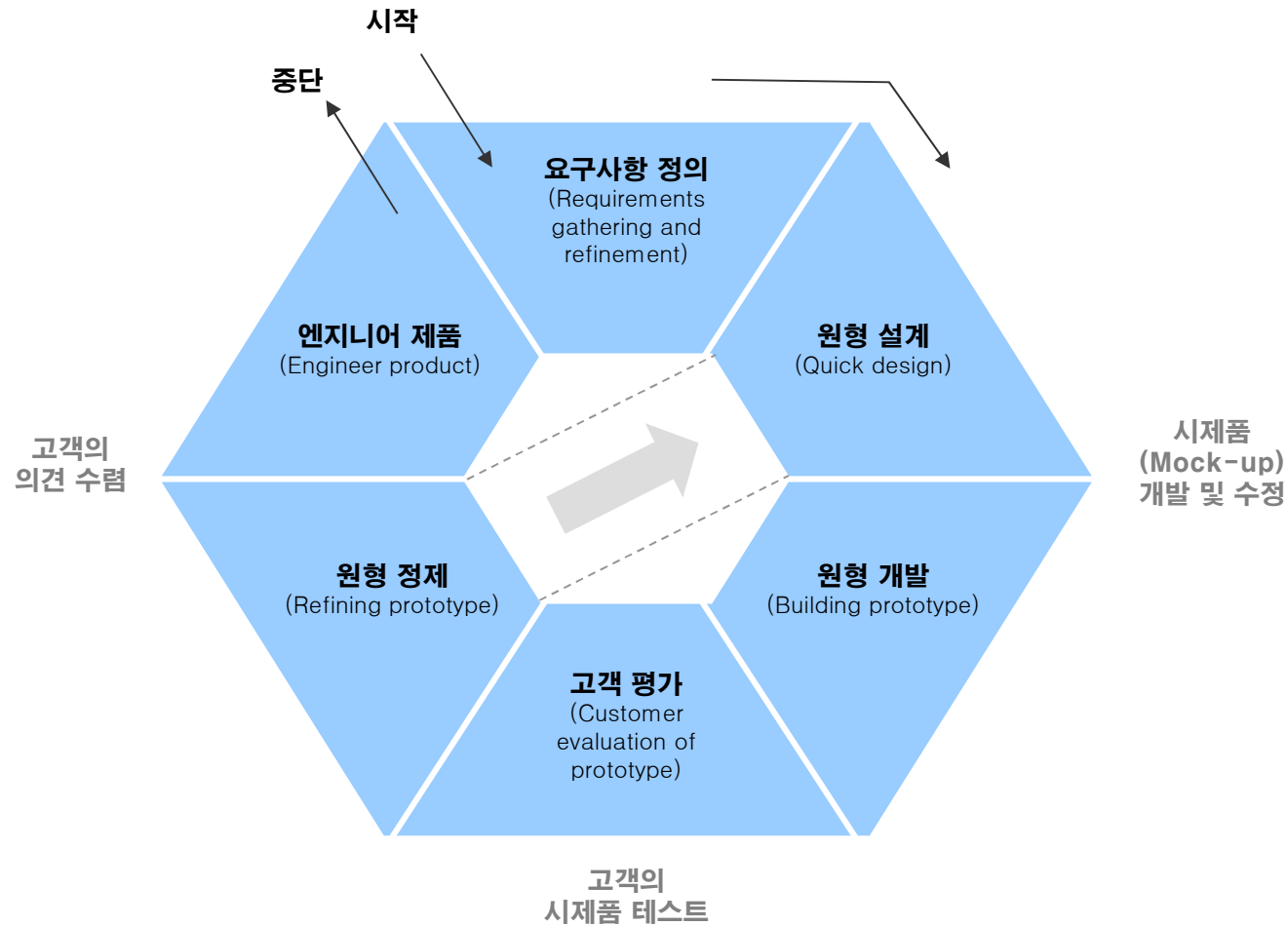
## ❖ 목적

- 원형을 가능한 빨리 개발하여 고객과 검증하는 것
- 방법
  - 고객으로부터 피드백을 받은 후 원형을 폐기
  - 시스템 기능 중 중요한 부분만 구현하여 피드백을 얻은 후 지속적으로 발전시켜 완제품을 제작

## ❖ 적용 가능한 경우

- 소프트웨어 개발 초기에 고객 요구사항을 완전히 파악하기 어려울 때

# 원형 모델(Prototyping Model) (2/3)





# 원형 모델(Prototyping Model) (3/3)

---

## ❖ 단계별 활동

### - 요구사항 정의

- 고객의 일부 요구사항 또는 불완전한 요구 사항으로부터 제품의 윤곽을 잡음

### - 원형 설계

- 주어진 요구사항을 기반으로 빠른 설계를 함
- 주로 제품의 사용자 인터페이스에 초점을 맞춤

### - 원형 개발

- 설계된 원형을 RAD(Rapid Application Development) 도구 등을 사용하여 빠르게 구현함
- 고객이 요구하는 기능을 구현하고 필요한 요소를 파악하는데 중점을 둠
- 프로그램의 신뢰도나 품질이 아니라 가능한 빨리 원형을 구현하는 것이 목적

### - 고객 평가

- 고객과 개발자가 함께하는 가장 중요한 단계
- 고객 요구사항을 정확하게 규명하기 위해 원형에 대한 사용 및 평가 시간을 충분히 제공
- 개발될 소프트웨어의 요구사항 정제에 중요한 정보로 활용

### - 원형 정제

- 원형이 어떻게 수정되어야 할지를 결정함
- 원형 개발과 검증, 요구사항 정제의 순환을 반복하여 추가적인 정보를 통해 요구사항을 완성해 나감

# 나선형 모델(Spiral Model) (1/4)

---

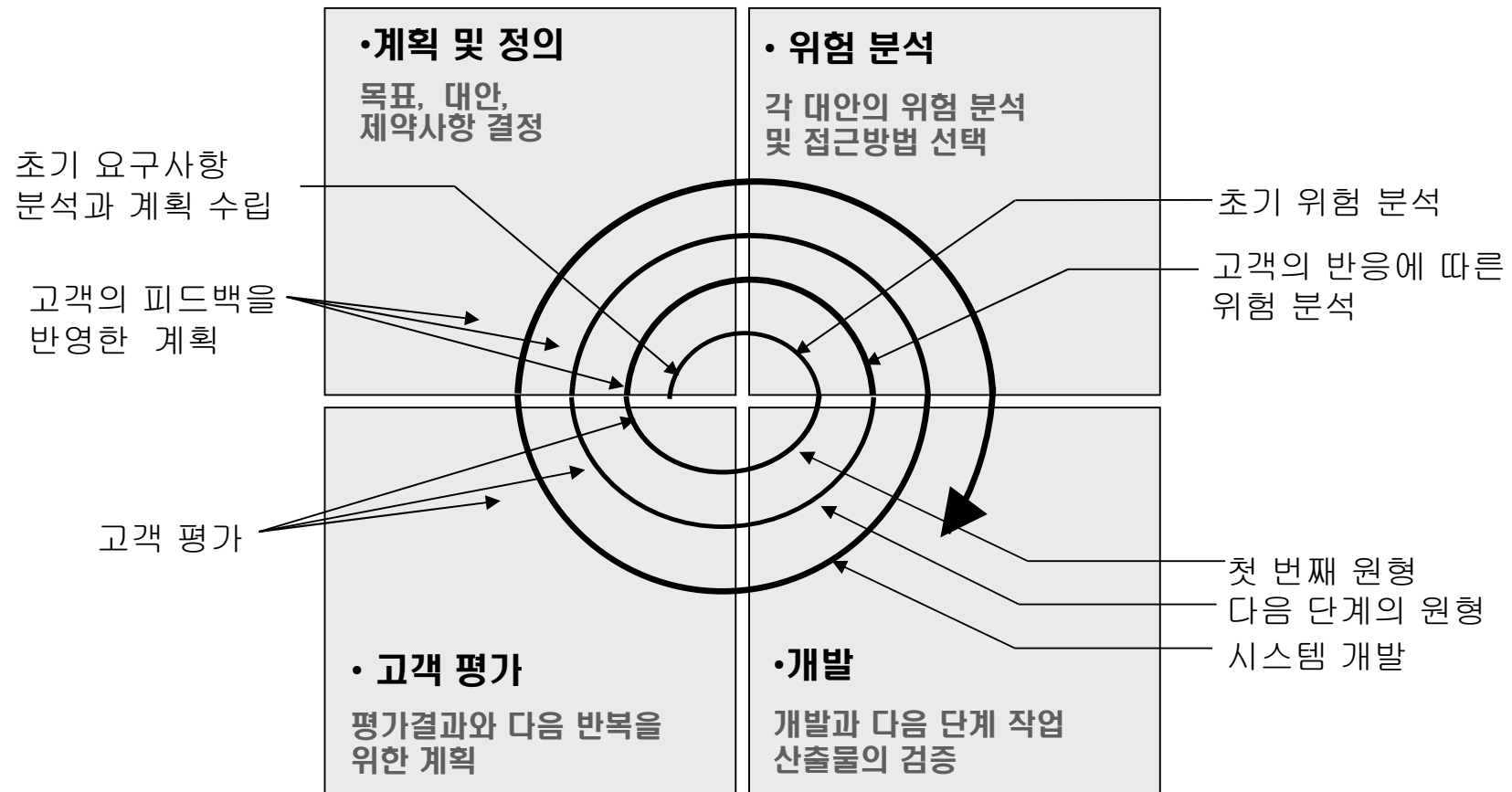
## ❖ 개요

- 폭포수 모형과 원형 모형의 장점을 수용하고 위험 분석(Risk analysis)을 추가한 점증적 개발 모델
- 프로젝트 수행 시 발생하는 위험을 관리하고 최소화 하려는 것이 목적

## ❖ 특징

- 여러 개의 작업 영역으로 구분
- 나선상의 각 원은 소프트웨어 개발의 점증적 주기 표현
  - 가장 안쪽 타원부터 개념적 개발 프로젝트, 실제 제품 개발 프로젝트, 제품 향상 프로젝트, 유지보수 프로젝트
- 단계가 명확히 구분되지 않고, 엔지니어가 프로젝트 성격이나 진행 상황에 따라 단계 구분

# 나선형 모델(Spiral Model) (2/4)



# 나선형 모델(Spiral Model) (3/4)

---

## ❖ 단계별 활동

### - 계획 및 정의 단계

- 개발자는 고객으로부터 요구사항을 수집
- 개발자는 시스템의 성능, 기능을 비롯한 시스템의 목표를 규명하고 제약 조건을 파악
- 목표와 제약 조건에 대한 여러 대안들을 고려하고 평가함으로써 프로젝트 위험의 원인을 규명 가능

### - 위험 분석 단계

- 초기의 요구 사항을 토대로 위험 규명
- 위험에 대한 평가가 이루어지면 프로젝트를 계속 진행할 것인지 아니면 중단할 것인지를 결정

### - 개발 단계

- 시스템에 대한 생명주기 모델을 선택하거나 원형 또는 최종적인 제품을 만드는 단계

### - 고객 평가 단계

- 구현된 소프트웨어(시뮬레이션 모형, 원형 또는 실제 시스템)를 고객이나 사용자가 평가함
- 고객의 피드백을 얻는데 필요한 작업이 포함
- 다음 단계에서 고객의 평가를 반영할 수 있는 자료 획득 가능

# 나선형 모델(Spiral Model) (4/4)

---

## ❖ 적용 가능한 경우

- 개발에 따른 위험을 잘 파악하여 대처할 수 있기 때문에
  - 고비용의 시스템 개발
  - 시간이 많이 소요되는 큰 시스템 구축 시 유용

## ❖ 장점

- 프로젝트의 모든 단계에서 기술적인 위험을 직접 고려할 수 있어 사전에 위험 감소 가능
- 테스트 비용이나 제품 개발 지연 등의 문제 해결 가능

## ❖ 단점

- 개발자가 정확하지 않은 위험 분석을 했을 경우 심각한 문제 발생 가능
- 폭포수, 원형 모델에 비해 상대적으로 복잡하여 프로젝트 관리 자체가 어려울 수 있음

# 소프트웨어 개발 방법론

---

# 소프트웨어 개발 방법론

## [Software Development Methodology]

---

### ❖ 의미

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

### ❖ 소프트웨어 개발 방법론의 종류

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

# UP (Unified Process) (1/7)

---

## ❖ 개요

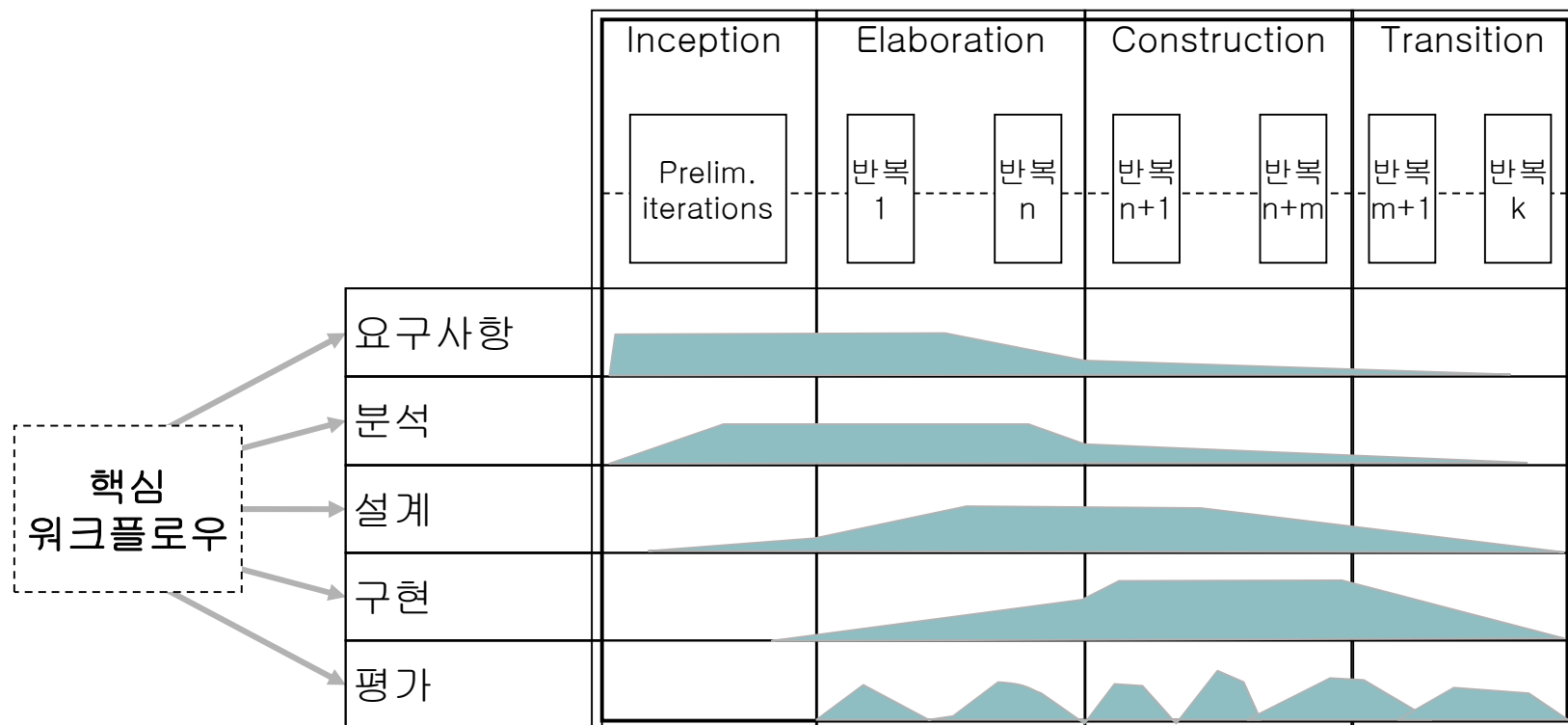
- Jacobson, Booch, Rumbaugh에 의해 개발된 객체지향 소프트웨어 개발 방법론
- 소프트웨어 개발 단계를 시간의 순서에 따라 네 개의 범주(Inception, Elaboration, Construction, Transition)로 나누고, 각 범주에는 요구사항 도출부터 평가까지 개발 생명주기가 포함되어 있음

## ❖ 특징

- 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
- 유스케이스(Usecase)를 기반으로 함
  - 요구사항을 식별하고, 정의하는데 있어서 UML의 유스케이스 사용
  - 유스케이스에 대한 자세한 설명은 4장 참고
- 아키텍처 (Architecture) 중심의 개발을 지향
  - 시스템 전체를 표현한 아키텍처는 프로젝트 참여자들에게 최종 산출물의 모습을 인지하게 하고, 구성원들을 공통된 시각을 갖도록 함
- 위험 관리를 중시
  - 프로젝트 성공에 장애가 될 수 있는 위험요소들을 파악하고, 위험도가 높은 것일수록 프로젝트 초기에 처리 방안을 찾아 해결



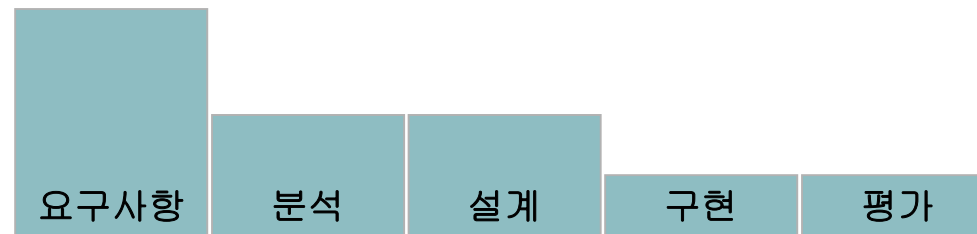
# UP (Unified Process) (2/7)



# UP (Unified Process) (3/7)

---

## ❖ 도입(Inception)



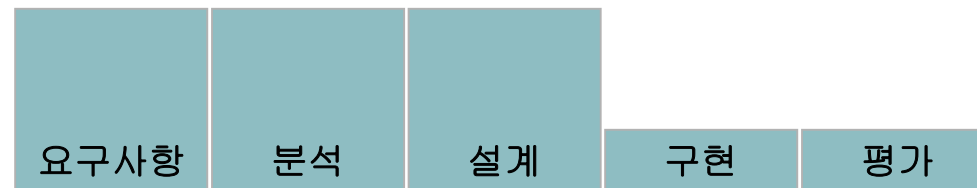
<도입 단계의 반복 워크 플로우>

- 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

# UP (Unified Process) (4/7)

---

## ❖ 상세(Elaboration)



<상세 단계의 반복 워크 플로우>

- 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 이루어짐

# UP (Unified Process) (5/7)

---

## ❖ 구축(Construction)



<구축 단계의 반복 워크 플로우>

- **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**

# UP (Unified Process) (6/7)

---

## ❖ 이행(Transition)



<이행 단계의 반복 워크 플로우>

- 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

# UP (Unified Process) (7/7)

---

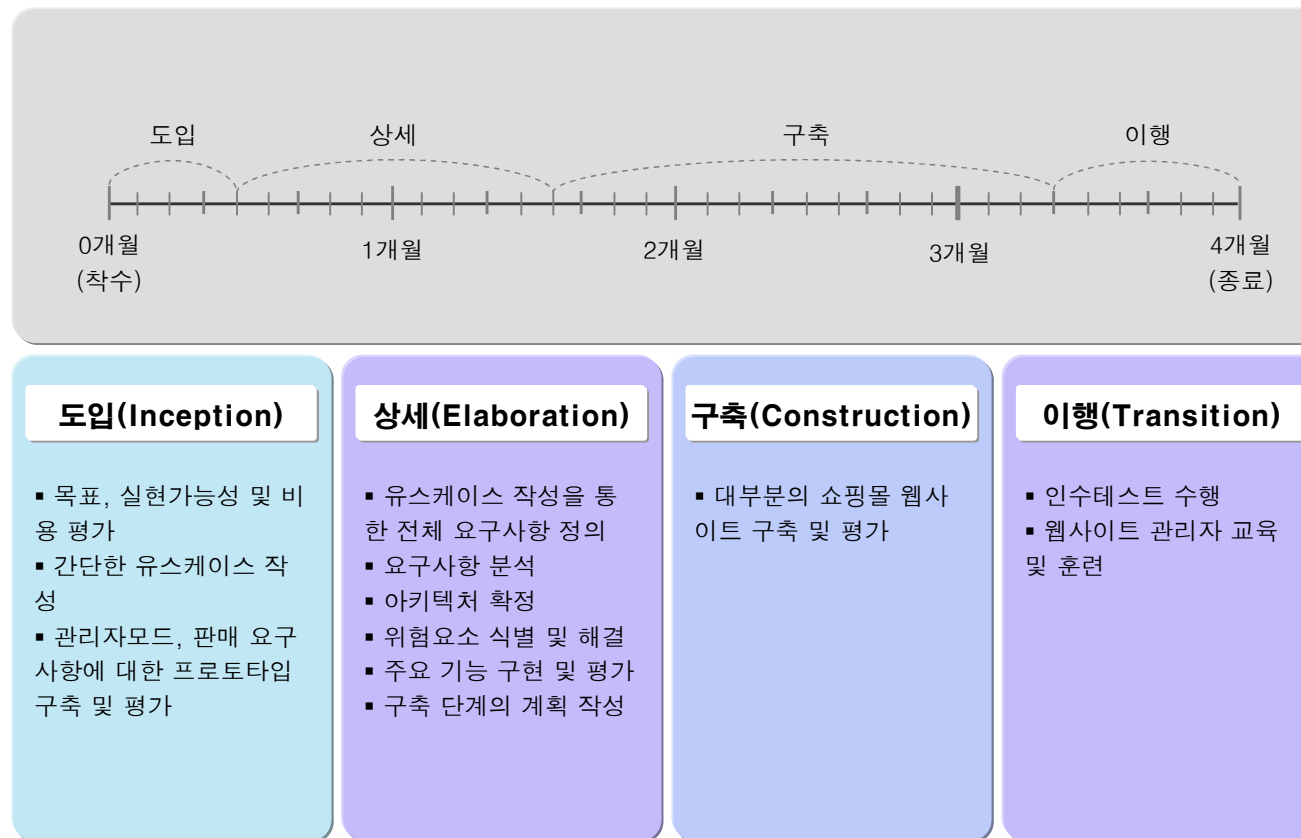
## ❖ 이점

- 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 거듭될수록 개선된 소프트웨어 개발이 가능

# [예] UP를 활용한 쇼핑몰 웹 사이트 개발 [1/2]

## ❖ 개요

- 4개월 동안 수행되는 쇼핑몰 웹 사이트 개발 프로젝트



# [예] UP를 활용한 쇼핑몰 웹 사이트 개발 (2/2)

---

## ❖ 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 프로토타입을 작성하고 평가

### - 상세 단계 (기간: 4주)

- 전체 요구사항을 정의 및 분석하고 아키텍처를 정의하여 쇼핑몰 웹사이트에 대한 구조를 확정하고 평가하고 위험 요소를 식별하여 완화시키는 노력을 함

### - 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

### - 이행 단계 (기간: 3주)

- 구축된 쇼핑몰 웹 사이트를 고객의 사용 환경에서 평가하고 담당자에 대한 교육 및 훈련을 실시



# 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

## ❖ 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

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

# XP에서의 역할

---

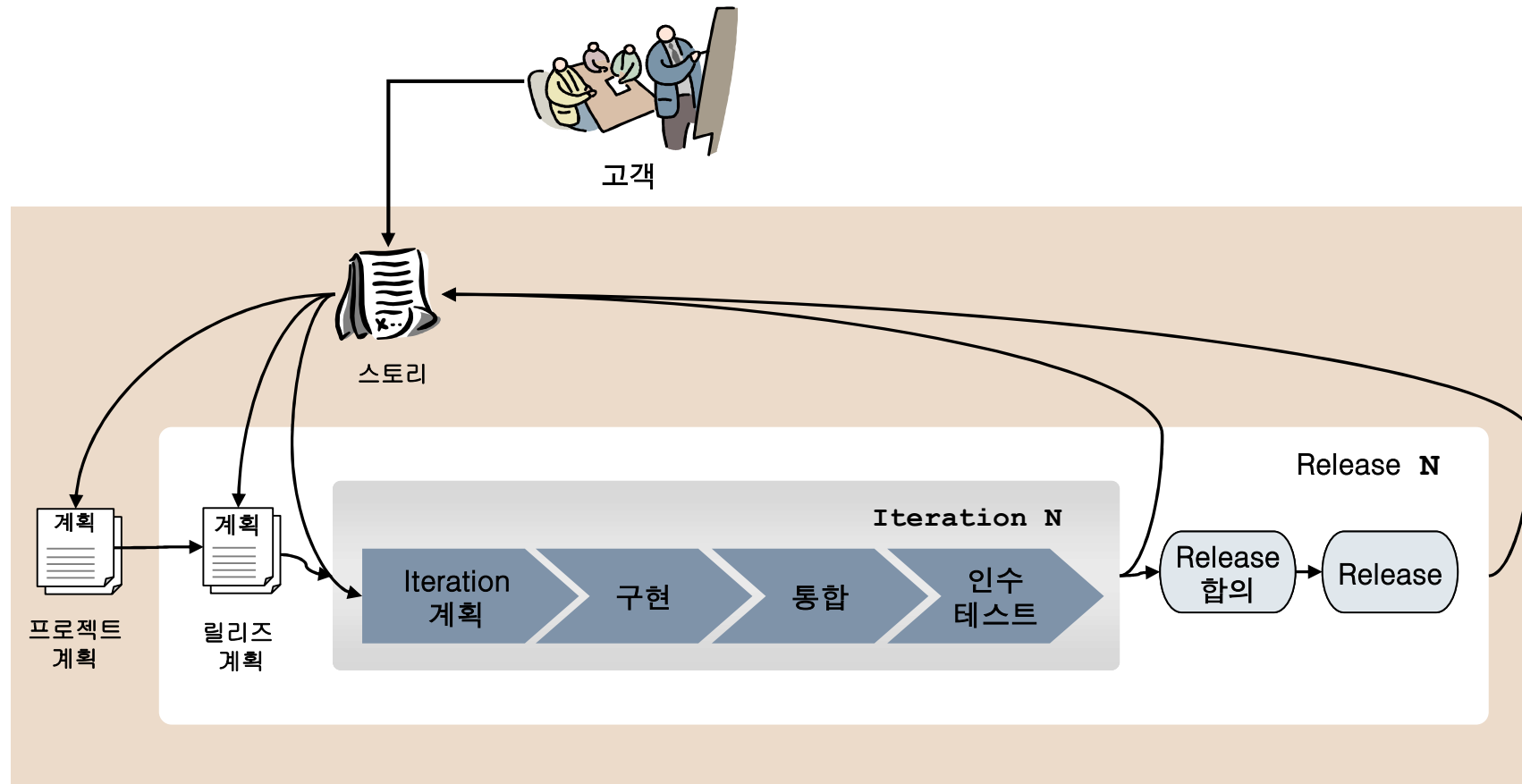
역 할	활동 설명
개발자	<ul style="list-style-type: none"><li>• 스토리 난이도 추정</li><li>• 소프트웨어 설계, 구현, 테스트 실시</li></ul>
관리자	<ul style="list-style-type: none"><li>• 개발자 앞에 놓인 개발 이외의 장애물 제거</li><li>• 개발 활동에 직접 관여하지 않고 개발 활동을 지시, 정리 및 결과 보고</li></ul>
고객	<ul style="list-style-type: none"><li>• 스토리를 제출하고, 스토리 구현 순서를 결정</li><li>• 구현된 스토리의 인수 테스트 실시</li><li>• 개발팀과 함께 상주</li></ul>

# XP의 개발 활동 [1/2]

---

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

## XP의 개발 활동 (2/2)



출처: CMMI기반의 XP를 위한 형상 관리 프로세스 프레임워크 개발, 한혁수, 한동준, 2007

# XP의 가치

---

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

# XP의 13가지 실천 사항

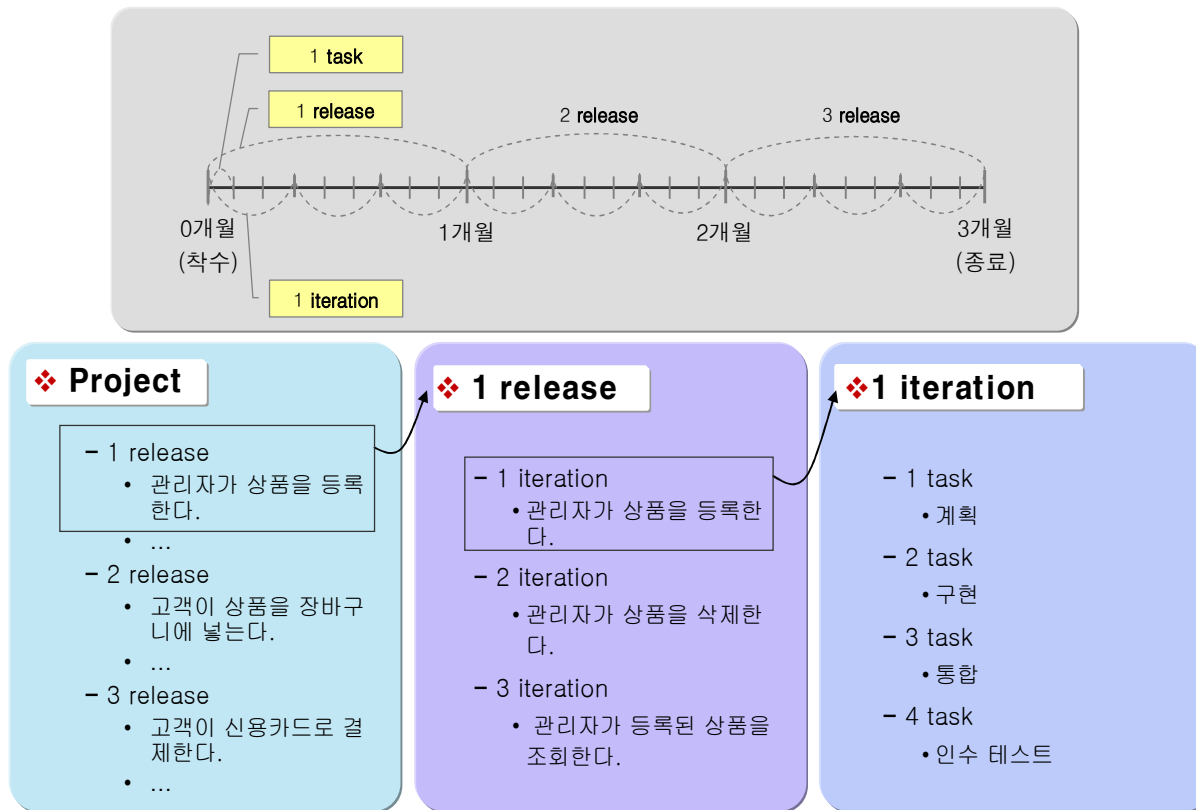
---

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

# [예] XP를 활용한 쇼핑몰 웹 사이트 개발

## ❖ 개요

- 3개월 동안 수행되는 쇼핑몰 웹 사이트 개발 프로젝트



- 고객이 정한 우선순위에 따라, 우선순위가 높은 요구사항부터 릴리즈를 구성한다.



# 마르미 (MaRMI) (1/3)

---

## ❖ 개요

- 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

# 마르미 (MaRMI) (2/3)

---

## ❖ 마르미

- 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

## ❖ 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

# 마르미 (MaRMI) (3/3)

---

## ❖ 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

## ❖ 마르미-RE

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

## 연습문제 (1/2)

---

1. 소프트웨어 생명 주기의 역할은 무엇인가?
2. 소프트웨어 개발 생명주기 단계 중 가장 오랜 시간이 걸리며, 대부분의 비용을 차지하는 단계는 무엇인가?
3. 소프트웨어 개발 생명주기 중 폭포수 모델(Waterfall Model)의 개발 단계를 표현하라.
4. 폭포수 모델에 대하여 설명하라.
5. 실제 상황이 나오기 전에 가상으로 시뮬레이션을 통해 최종 결과물에 대한 예측을 할 수 있는 소프트웨어 개발 생명주기는 무엇인가?
6. 원형 모델의 가장 큰 장점은 무엇인가?
7. 원형 모델의 개발에 필요한 작업을 순서대로 표현하라.

## 연습문제 (2/2)

---

8. 나선형 모델에 대하여 설명하라.
9. 나선형 모델의 태스크(task)를 나열하라.
10. Jacobson, Booch, Rumbaugh에 의해 1999년에 개발된 객체지향 소프트웨어 개발 방법론인 UP(Unified Process)의 특징을 나열하라.
11. XP(eXtreme Programming)는 어떠한 방법론인가?
12. XP에서 고객의 요구사항을 무엇이라 하는가?
13. 한국전자통신연구원의 소프트웨어 공학 연구팀에서 국내 여건을 반영하여 개발한 한국형 소프트웨어 개발 방법론은 무엇인가?