

# 소프트웨어공학

## 소프트웨어 개발 방법론 2

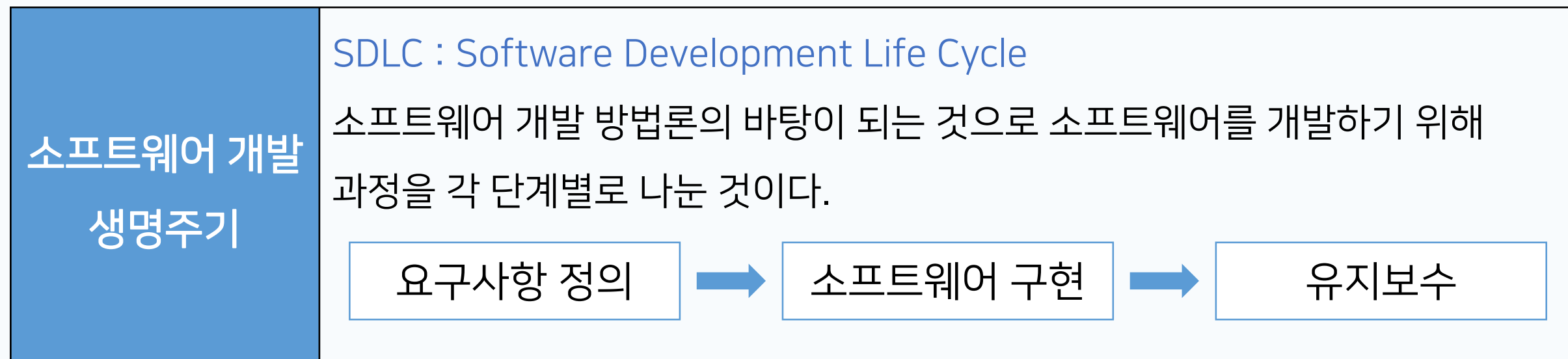
1 생명주기의 정의와 역할

2 생명주기 각 과정과 내용

# 생명주기의 정의와 역할

## 1. SDLC(Software Development Life Cycle)

- SDLC(Software Development Life Cycle)의 정의
  - S/W 개발 방법론의 바탕이 되는 것으로 S/W를 개발하기 위해 과정을 각 단계별로 나눈 것



# 생명주기의 정의와 역할

## 1. SDLC(Software Development Life Cycle)

- S/W 개발 생명주기(Software Development Life Cycle)의 역할
  - 프로젝트의 비용 산정과 개발 계획을 수립할 수 있는 기본 골격이 된다.
  - 용어 및 기술의 표준화를 가능하게 한다.
  - 문서화가 충실한 프로젝트 관리를 가능하게 한다.
  - 여러 소프트웨어 간에 상호 일관성을 유지하게 한다.
  - 프로젝트 진행 방향을 명확하게 파악하게 한다.

# 생명주기의 정의와 역할

## 1. SDLC(Software Development Life Cycle)

- S/W 개발 생명주기(Software Development Life Cycle)의 일반적 공정 과정

|                        |               |                   |
|------------------------|---------------|-------------------|
| 1.정의 단계<br>( what )    | 1) 타당성 검토 단계  | - RFP             |
|                        | 2) 개발 계획 단계   | - 수행계획서           |
|                        | 3) 요구사항 분석 단계 | - 요구사항 명세서        |
| 2. 개발 단계<br>( how )    | 4) 설계 단계      | - 설계 문서           |
|                        | 5) 구현 단계      | - 실행 코드           |
|                        | 6) 테스트 단계     | - 테스트 계획서, 결과서    |
| 3. 유지보수 단계<br>(change) | 7) 유지보수 단계    | - 유지보수 계획서, 폐기승인서 |

## 생명주기의 정의와 역할

### 2. 타당성 검토 및 개발 계획

- 타당성 검토 단계 : 개발할 S/W가 법적·경제적·기술적으로 실현 가능성이 있는지 조사
- 개발 계획 단계 : S/W 개발에 사용될 일정, 인력과 비용을 측정하고 개발 문서를 만드는 과정

|         |  |
|---------|--|
| 프로젝트 개요 | 프로젝트의 목표와 배경, 범위 등을 기술하며 시스템 구성 및 시스템 배경도가 포함될 수 있다.     |
| 추진 전략   | 프로젝트 목적 달성을 위한 추진 전략 및 구축 방안을 제시한다.                      |
| 위험 분석   | 프로젝트를 수행하며 예상되는 위험과 해결 방법을 기술한다.                         |
| 자원 요구사항 | 시스템 구성과 개발에 요구되는 하드웨어, 소프트웨어 및 네트워크 구축장비 등의 자원 기술을 기술한다. |
| 조직      | 프로젝트에 참여할 사람 및 역할 정의, 개발팀 구성과 업무 분할에 대해 기술한다.            |
| 관리      | 개발관리 방법, 추진 일정 계획, 품질보증 및 시험, 교육훈련계획 등을 기술한다.            |

### 2. 타당성 검토 및 개발 계획

- 참고 1
  - 공급자 기준
    - ✓ 주문생산 제품(customized products) : 구매할 고객이 확정, 일반적으로 고객이 많은 비용 부담
    - ✓ 상업화될 제품(commercial products) : 구매할 고객이 미정, 개발회사가 개발 비용을 부담

# 생명주기의 정의와 역할

## 2. 타당성 검토 및 개발 계획

### 참고 1

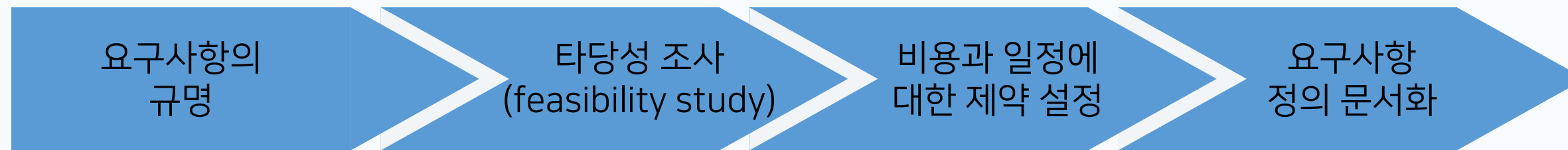
- 수요자 기준 : 정보요청서 → 제안요청서 → 제안서 → 입찰 순으로 진행

| 구 분                                   | 내 용   |
|---------------------------------------|---|
| 정보요청서<br>RFI(Request For Information) | <ul style="list-style-type: none"> <li>- S/W 개발사들의 회사소개, 시장동향, 경쟁사 정보 등을 받아 진행하고자 하는 프로젝트에 대한 정보를 미리 수집,분석하는 준비단계</li> <li>- 요청: 발주사 → 개발사(RFI제공) → 발주사</li> </ul>  |
| 제안요청서<br>RFP(Request For Proposal)    | <ul style="list-style-type: none"> <li>- S/W를 필요로 하는 회사에서 해결해야 할 문제들과 해결책에 대한 요구사항들을 설명하여 S/W 개발 회사에 보내는 문서.</li> <li>- 필요로 하는 S/W를 개발할 회사를 선택하지 않았을 때 사용.</li> <li>- 원하는 S/W를 개발할 능력이 있다고 판단되는 회사에 보내지는 문서.</li> <li>- 또는 인터넷 등을 이용하여 공지.</li> <li>- 발주사(RFP 작성) → 개발사들</li> </ul> |
| 제안서<br>Propasal                       | <ul style="list-style-type: none"> <li>- S/W를 개발할 회사들은 제안요청서를 기초로 S/W 개발계획을 담은 제안서를 작성.</li> <li>- 제안에 참여하는 목적과 배경, 프로젝트 추진 전략, 사업 수행 범위, 제안의 특징, 장점 및 차별성, 프로젝트를 수행하기 위한 조직, 기술, 관리, 결과물 등을 기술.</li> <li>- 개발사 → 발주사</li> </ul>  |
| 입찰                                    | <ul style="list-style-type: none"> <li>- 제안서를 받아 심사하고, 선정 후 계약하는 과정.</li> </ul>   |

## 생명주기 각 과정과 내용

### 1. 요구사항 분석 단계

- 요구사항 분석 단계
  - 품질과 밀접한 관계이므로 사용자의 요구 사항을 보다 상세하고 정확하게 분석하는 단계



< 요구사항 분석 과정에서 수행되는 내용 >



## 1. 요구사항 분석 단계

### ① 요구사항의 규명

- 사용자의 관점에서 시스템의 요구사항을 모으는 것
- 기능(function) 요구, 성능(performance) 요구, 인터페이스(interface) 등의 예
- 업무 분석(요구 원인, 배경, 환경 등에 대한 분석 / 시스템이 필요하게 된 고객의 환경 분석)
- 환경 분석
  - ✓ 내적 요소(인력규모축소, 생산성증대, 기술력향상, 서비스향상)
  - ✓ 외적 요소(경쟁력 강화, 시장여건의변화, 법규나제도의변화)

## 생명주기 각 과정과 내용

### 1. 요구사항 분석 단계

#### ② 타당성조사

- 계획 단계나 분석 단계에서 수행
- 시스템 개발에 요구되는 시간, 비용, 인력 등의 자원 → 시스템의 타당성에 직접적인 영향
  - ✓ 경제적 타당성, 기술적 타당성, 법적 타당성, 대체 방안 등에 집중

## 생명주기 각 과정과 내용

### 1. 요구사항 분석 단계

#### ③ 비용과 일정에 대한 제약 설정

- 요구사항을 분석하는 분석가는 응용 분야에 대한 해박한 지식이 요구
- 개발 비용, 개발 일정, 시스템 성능 등에 대해서도 정확한 예측
- 프로젝트 관리가 매우 중요

|       |   |
|-------|---|
| 관리 활동 | <ul style="list-style-type: none"><li>- 요구되는 자원과 성취해야 할 임무</li><li>- 소요 기간</li><li>- 추적해야 할 이정표 등</li></ul> |
|-------|---|

## 1. 요구사항 분석 단계

### ④ 요구사항 정의 문서화

- 사용자의 요구사항과 시스템의 기능 문서화
- 고객과 개발 회사의 약속문서 (계약서)
- 분석가가 만드는 경우가 대부분이나 고객과 함께 만드는 경우도 있음
- 프로젝트와 관계된 모든 사람이 읽고, 이해하기 쉽도록 작성
- 추후에 발생하는 문제와 변화에 대한 책임이 명확히 규명 가능
- 시스템에 연관된 당사자들이 동의하여 서명
- 주요 산출물 : 요구사항 명세서 ( = 기능 명세서, 기능 요구서 )

#### 중요한 산출물

요구사항 명세서(requirements specification)  
- 추적해야 할 이정표 등

## 2. 설계단계

설계  
(design)

요구사항 분석 과정에서 모아진 요구사항을 설계 도면에 옮기는 것

- 설계 단계 : S/W의 구조, 알고리즘, 자료구조 등을 작성하는 단계  
(에러가 가장 많이 발생)
  - 무엇(what), 어떻게(how), 서브시스템(subsystem)들로 이루어지는 시스템 구조 결정
  - 서브시스템들을 하드웨어 및 소프트웨어 등의 구성요소에 할당
  - 품질에 직접적인 영향을 주게 되며, 설계가 제대로 되지 않으면 안정감이 없는 시스템이 제공
  - 안정감이 없는 시스템은 유지보수가 어렵다.
  - 변경 사항 발생시 재개발 가능성이 높다.

## 2. 설계단계

설계  
(design)

요구사항 분석 과정에서 모아진 요구사항을 설계 도면에 옮기는 것

- 설계 원칙

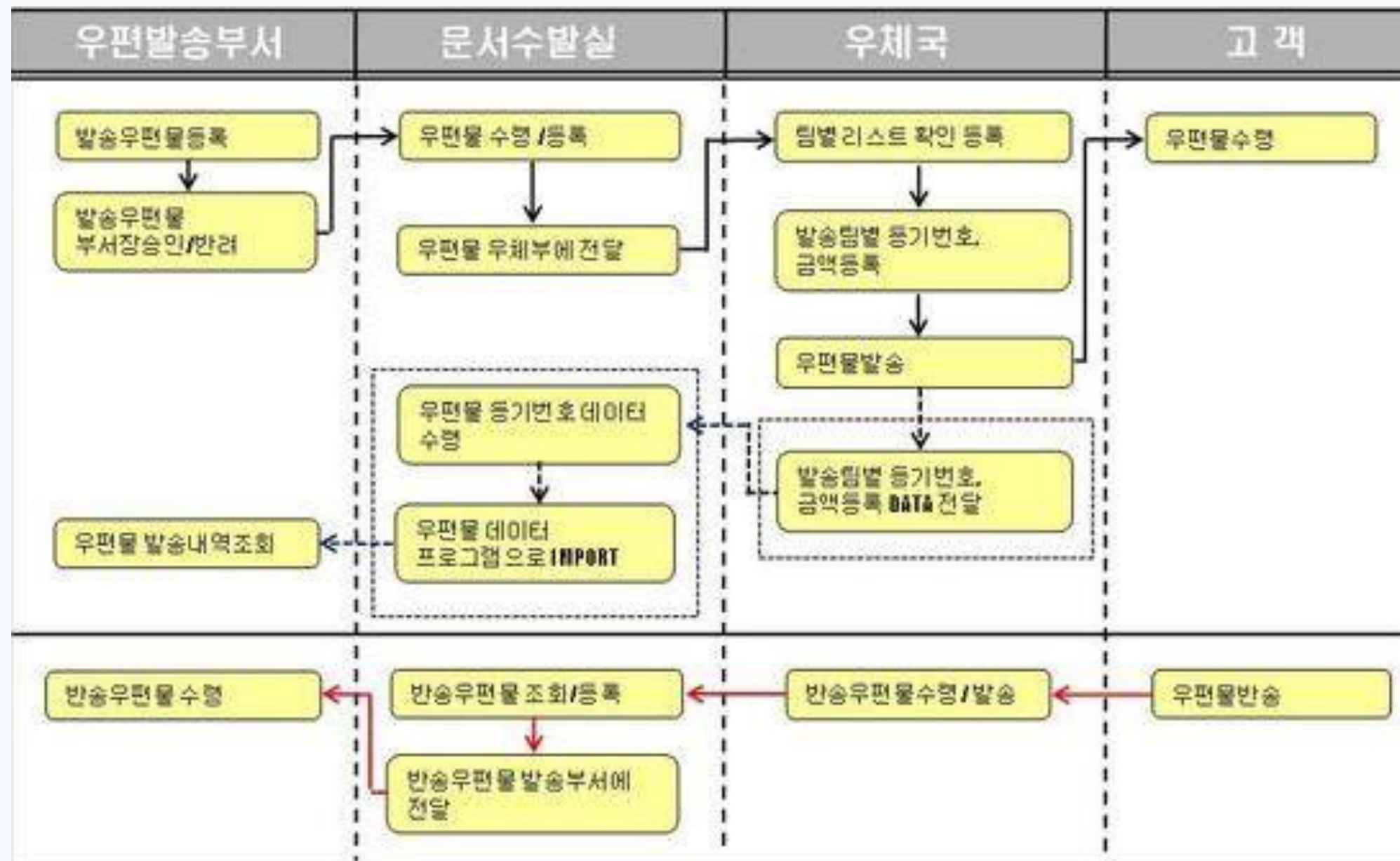
- ① 시스템을 구성 요소로 적절하게 분할
- ② 시스템의 구성요소들 사이에 주고받는 정보의 소통이 최소화되고,  
각 구성요소의 독립성이 유지될 수 있도록 시스템을 분할
- ③ 요구되는 성능과 자원에 대한 예측 가능해야 함

## 02

# 생명주기 각 과정과 내용

## 2. 설계단계 예시

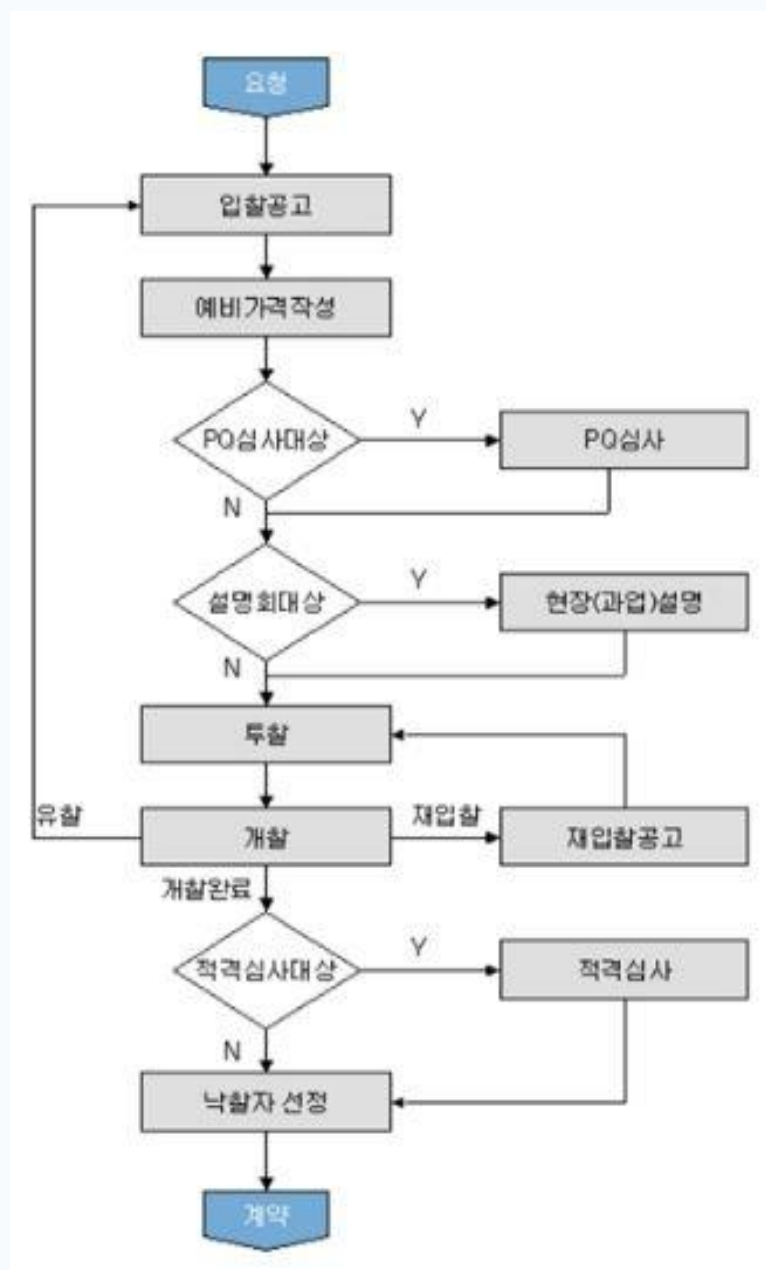
- 업무 설계



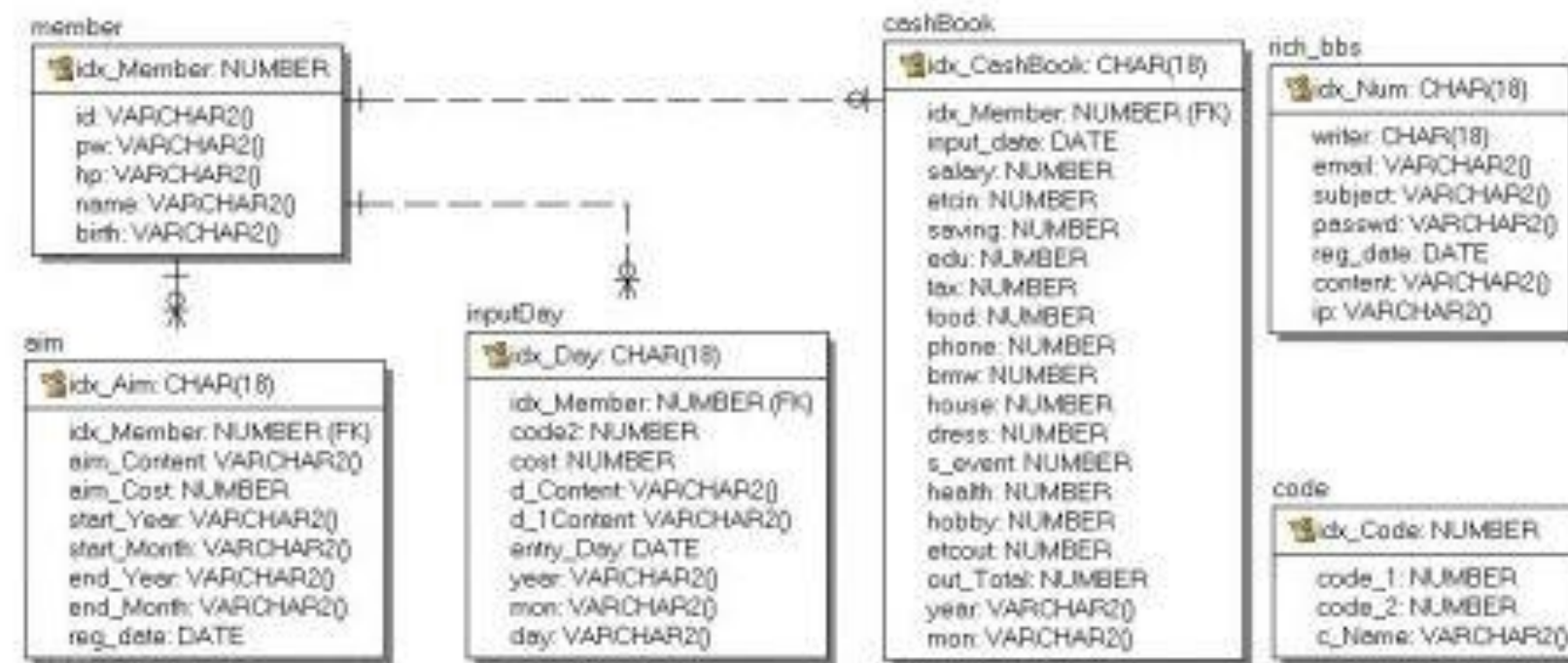
# 02

## 생명주기 각 과정과 내용

### 2. 설계단계 예시



< Flow Chart >



< 데이터 설계 예시[ERD] >



## 3. 분석단계 VS 설계단계

| 분석 단계  | 설계 단계   |
|--|---|
| <ul style="list-style-type: none"><li>- 개념적(conceptual) 단계</li><li>- 무엇(what)을 제공할 것인가에 초점.</li><li>- 무엇(what)은 사용자나 시스템의 기능을 사용자의 관점.</li></ul> | <ul style="list-style-type: none"><li>- 물리적(physical) 실현의 첫 단계</li><li>- 어떻게(how to) 그 문제를 해결할 것인가를 결정.</li><li>- 어떻게(how to)는 기능의 수행 방법을 엔지니어의 관점.</li></ul> |

## 4. 구현 및 테스트 단계

- 구현 단계
  - 설계 단계에서 작성된 문서를 기초로 프로그래밍(코딩) 하여 사용자가 이용할 수 있게 변환하는 것
  - 시스템의 기능이 수행 가능한 모습으로 표현
  - S/W의 경우 설계가 제대로 이루어지면 시스템 구현은 상대적으로 단순하고, 기계적인 과정으로 효율적

## 4. 구현 및 테스트 단계

- 테스트 단계
  - 구현된 S/W 제품의 내제된 오류를 발견하고, 수정하는 단계
  - 우수한 품질의 제품을 얻기 위해서는 제품이 개발되는 공정 과정마다 품질보증을 위한 절차 준수
  - 공식적인 검토를 통해 잘못된 것을 걸러내는(filtering) 작업 수행
  - 품질보증 활동의 중요한 일부분으로 사용자 요구사항, 설계, 구현의 전 과정에 대한 최종 점검을 포함
  - 체계적인 S/W시스템 테스트를 하기 위해서는 테스트 계획(test plan) 수립
  - 테스트 진행 단계, 테스트에 사용되는 데이터 및 테스트의 한계점 등이 기술
  - 최소한의 시간과 비용을 투자해서 최대한의 확률로 오류 검출

## 5. 유지보수 단계

- 유지보수 단계
  - S/W를 직접 사용하여 '변화'에 초점을 두고 S/W를 적용 및 유지시키는 단계
  - 시간과 비용이 가장 많이 투입되는 단계
  - 시스템 변경에 의한 재요구분석, 재설계, 재구현, 재테스트가 필요하게 되고, 관련된 문서의 수정까지도 수반하기 때문에 체계적인 관리 기능이 필요
  - 소프트웨어 시스템은 개발할 때부터 유지보수에 대비한 제작

## 5. 유지보수 단계

- 유지보수의 4가지 활동

- ① 잘못된 것을 수정하는 유지보수(corrective maintenance) - 수정
- ② 시스템을 새 환경에 적응시키는 유지보수(adaptive maintenance) - 적응
- ③ 새로운 기능을 추가하는 유지보수(perfective maintenance) - 완전
- ④ 미래의 시스템 관리를 위한 유지보수(preventive maintenance) - 예방

## 6. SW품질 보증을 위한 조건

- SW일반 공정 과정인 계획, 분석, 설계, 구현, 테스트 및 유지보수 단계에 적용될 수 있는 기법과 도구들이 확립되어야 하며, 각 공정 과정의 임무, 입력물, 산출물, 사용도구 등도 정의
- SW품질보증 활동 충족 조건
  - ① 사용자가 요구하는 기능적인 요구사항들이 만족
  - ② 요구사항에 대한 부합여부가 시스템 구축 전 과정에 걸쳐 확인
  - ③ 시스템의 성능에 관한 사양이 만족
  - ④ 정해진 비용과 기간의 목표가 만족

## 7. SDLC 장, 단점

- 장점

- SW 개발 과정을 쉽게 이해하기 위한 효과적인 도구
- SW 유형, 관점, 개발 방침, 요구 사항, 표준 정책에 따라 다양한 방법이 존재
- 요구사항 정의, 설계, 프로그래밍 등에 대한 기법들을 활용한 템플릿이 제공될 수 있음
- 체계적인 문서화, 단계별 산출물 체크를 통한 프로젝트 진행의 명확화

- 단점

- 문서 중심의 개발 접근 방식으로 개발자가 문서화에 대한 부담 가중
- 개발 초기에 사용자의 요구사항을 명확하게 찾아내기 어려움
- 완벽한 분석이 요구되며 피드백과정이 없어 변경이 어려움