

요구 분석

01 요구 사항과 요구 분석 명세서의 이해

02 요구 분석의 이해

03 요구 분석 절차와 요구 사항 종류

04 요구 사항의 표현

05 요구 사항의 문서화

- 요구 분석의 필요성과 어려운 점을 알아본다.
- 요구 사항을 표현하는 방법을 익힌다.
- 요구 분석 명세서를 살펴본다.

Section 01 요구 사항과 요구 분석 명세서의 이해

1. 요구 사항과 요구 분석 명세서의 이해

▪ SW개발의 목적

- 개발된 소프트웨어의 **고객 만족**

▪ 고객 만족을 위한 특성

- **적시성**time to market : **빠른 출시**를 통한 시장의 점유
- **유연성**flexibility : **다양한 환경**에서의 적응성
- **통합**integration : **기존 시스템**과의 쉬운 통합

▪ 고객 만족의 개발 조건

- **고품질**의 제품 개발
- **정해진 기간**내의 개발
- **주어진 예산**내의 개발



그림 4-1 고객 만족

2. 요구 사항/요구 분석 명세서

■ 요구 사항requirement

- 이용자가 어떤 문제를 풀거나 목표를 달성하기 위해 **필요한 조건이나 능력**(사전적 의미)
- 사용자와 개발자 간에 **합의한 개발 범위에서 시스템이 제공해야 하는 기능**(SW개발 의미)

■ 요구 분석 명세서(요구사항 정의서)

- 개발 초기에 사용자의 **요구 사항(비기능 요구 사항 포함)**을 추출하여 정리한 문서
- **what**에만 **초점**, **how**는 고려치 **않음**



그림 4-2 무엇what과 어떻게how

Section 02 요구 분석의 이해

1. 요구 분석의 이해



애매하군.
그렇게 되면
23층의 구조가 달라지는데..

근데 2층이 60평인데
집을 하나로 해서
세를 주기엔 무리일 것 같아.
두 집으로 나눠야 할 것 같은데..



건축 설계 과정을 통한 요구 분석 개념

2. 요구 분석

▪ 요구 분석 과정



▪ 요구 분석

- 소프트웨어 요구 사항 정의를 위해 **사용자의 요구 사항을 조사하고 확인하는 과정**
- 소프트웨어 개발 **생명주기의 첫 단계**
- 소프트웨어 개발 **성패의 열쇠**



2-1 요구 분석 관련자의 역할과 의미

- 발주사 : A 대학에서 종합정보시스템을 외주로 개발하려고 하면 A 대학이 고객, 즉 발주사
- 경영자(총 책임자) : A 대학 종합정보시스템 개발을 결정한 최고 책임자, 즉 A 대학 총장
- 발주 담당자 : A 대학 종합정보시스템 개발 외주를 위해 모든 절차를 준비하는 담당자
- 사용자 : 개발된 시스템을 업무에 실제 사용하는 사람, 예 학사 담당자, 수강 담당자 등
- 수주사 : A 대학교 종합정보시스템을 개발하기로 결정된 업체, 즉 B사
- 분석가 : A 대학의 요구 사항 파악 및 추출하여 정리하는 것까지 담당
- 설계자 : 요구 분석 명세서를 바탕으로 아키텍처 설계, 모듈 설계, DB 설계, UI 설계 등을 담당
- 개발자 : 프로그램뿐만 아니라 분석가, 설계자 등을 말하며, 좁은 의미로는 프로그래머만 해당

3. 요구 분석의 어려움(1)

- 문제 영역에 대한 이해력 부족



그림 4-3 분야별 전문 변호사들의 활동 모습

3. 요구 분석의 어려움(2)

- 의사 소통의 문제

- 견본이 없어 요구 사항 설명이 어려움
- 사용자가 요구 사항에 대한 설명 방법을 잘 모름

⇒ 사용자 의사 전달 능력이 요구 사항 전달 내용에 영향을 미침

⇒ 일관성이 없거나 불안정한 요구 분석 명세서



3. 요구 분석의 어려움(3)

- 계속 변하는 요구 사항

- 초기의 요구 사항은 단순
- 개발될 SW에 대해 이해의 폭이 넓어짐에 따라 새로운 요구 사항 발생 및 증가
- 관련 지식의 증가로 인한 요구 사항 변경 발생



요구 사항간의 충돌, 일관성 결여 및 불일치 발생

- 변경 요구에 대한 대처 방안

- 변경 사항에 대한 날짜별, 기간별 기록
- 변경이 미치는 영향에 대해 사전 분석 필요

3. 요구 분석의 어려움(4)

- **애매모호한 요구 사항**

- 해석을 달리할 수 있는 **애매한 표현**의 요구
- 사용자의 **일관성 없는 요구**
- 부서간의 **상충된 요구**
- 경영진과 실무자간의 **상반된 요구**

- **분석가의 역할**

- 수집된 요구들을 반영하기 전에 **이해 당사자들 간의 주장을 조율**할 수 있어야 함

- **사용자와 개발자의 마찰**

- 최적의 분석명세서에 대한 **사용자의 다른 해석으로 인한 마찰**

4. 분석가에게 필요한 능력(1)

▪ 분석가의 주 업무

- 사용자의 요구를 **정확하고 완전하게 획득**하는 것
- 방법: 다양한 참여자를 만족시킬 수 있는 **공통의 목표**를 추출

▪ 의사 **소통**과 **협상** 능력

- 상반된 의견 → **조율** 능력
- 무리한 요구 → **우선 순위 결정** 및 **실현 가능한 요구 판별** 능력
- 추상적 요구 → **논리적 분할**을 통한 해결 방안 제시

▪ 개발 업무 **영역**에 대한 지식

- 업무 담당자와 **원할 한 대화** 가능
- **주도적으로 추진** 가능

4. 분석가에게 필요한 능력(2)

- 개발 관련 기술에 대한 지식

- 사용자의 요구가 기술적으로 구현 가능한지 판단 능력 필요

- 이해 관계자들의 상반된 요구에 대한 중재 능력

- 경영자: 투자 대비 효과(ROI)에 관심
- 고객(발주자): 개발비가 적고, 개발 기간 엄수할 수 있는 업체에 관심
- 사용자: 업무의 효율적 처리와 편리성 등에 관심

=> 경영자, 고객, 사용자의 상반된 요구에 대해 중재 및 조율 능력 필요

5. 요구 사항 수집 방법

▪ 자료 수집

- 문서 수집(업무 매뉴얼, 업무 흐름도 등) → 전반적인 업무 흐름 파악 및 이해
- 현행 시스템 분석: 입력화면 및 출력물

=> 현행 요구 사항 파악(As-Is)

▪ 인터뷰



▪ 설문 조사

- 설문을 통한 또 한 번의 요구사항 추출

5. 요구 사항 수집 방법

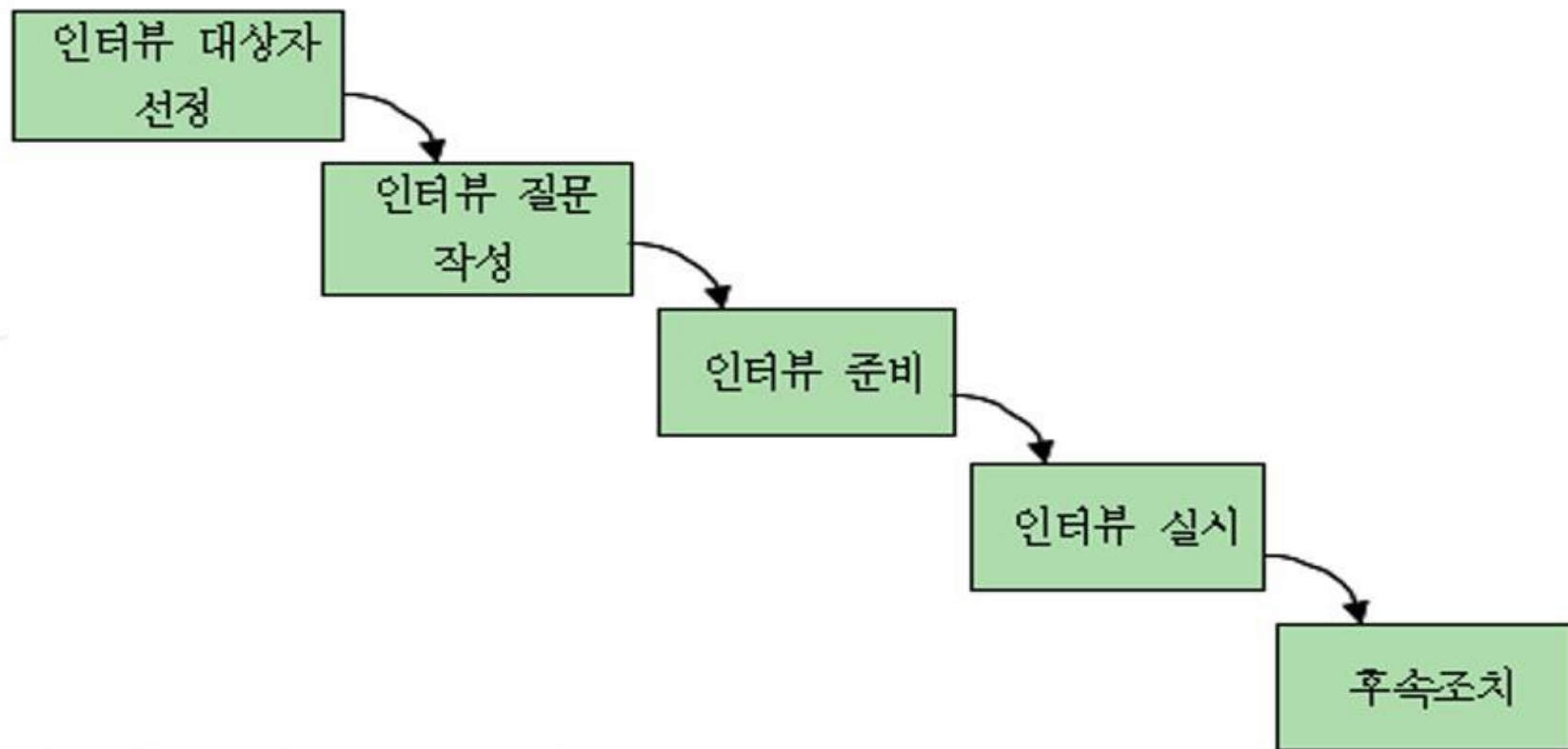
· 자료 수집

OO 공단 요청(1차 자료목록) 접수현황			
2017.01.25			
순위	자료명	사용목적	비고
1	업무지침(편람)	업무분석	접수완료 (법·규제, 운영지침)
2	업무 분장표	업무분석	접수완료(2017.01.25)
3	업무 절차서(프로세스)	업무분석	접수완료(2017.01.19)
4	조직도(최신 버전)	업무분석	접수완료(2017.01.25)
5	정보시스템 구성도	정보자산현황분석	접수완료(소음,실내)
6	네트워크 구성도	정보자산현황분석	접수완료(소음,실내)
7	보유 SW 현황	정보자산현황분석	필요
8	보유 장비 현황	정보자산현황분석	접수완료(소음,실내)
9	선행 프로젝트 보고서	사업현황의 이해	접수완료(소음,실내)
10	2016 사업보고서(계획서)	사업현황의 이해	2016년 사업 종료 시
11	2017 사업보고서(계획서)	사업현황의 이해	필요
12	지침/가이드라인	비정형 사건 등	소음지도 통합관리 가이드라인
13	유지보수업체 리스트	유지보수 현황파악	접수완료
14	소음 관련 자료(중장기 계획)	환경 및 현황 분석	접수완료(소음)
15	실내공기질 관련 자료(중장기 계획)	환경 및 현황 분석	접수완료(실내)
16	2013, 2014, 2015 사업결과보고서	사업현황의 이해	접수완료(실내)
17	시스템 메뉴얼(소음)	정보시스템 현황분석	소음지도관리를위한업그레이드용역 (완료보고서)
18			소음진동측정망통합관리시스템 (사용자설명서, 관리자설명서)
19			국가소음정보시스템 및 WEBGIS (사용자설명서)
20			필요(국가소음정보시스템 메뉴얼)
21	시스템 메뉴얼(실내)	정보시스템 현황분석	실내공기질측정망연계 자료공개서비스 시스템,이중화구축,관제시스템(완료보고서)
22			실내공기질자동측정망관제시스템 (관리자설명서)
23			실내공기질 통합관리시스템 (사용자설명서)-조사
24			필요(실내공기질 통합관리시스템)

5. 요구 사항 수집 방법

▪ 인터뷰

- 사용자와의 **심층 대화로부터 요구**를 끌어내는 방법
- **질문의 준비**가 중요



5. 요구 사항 수집 방법

▪ 인터뷰 질문 작성

- 다른 곳에서 찾을 수 있는 질문은 피해야(뻔한질문)
- 대상자가 **알 수 있다고 예상되는 질문만**
- 세 가지 유형

질문 유형	예
폐쇄형 (특별하고 정확한 정보를 찾을때)	<ul style="list-style-type: none">● 하루에 받는 전화 주문은 몇 개인가?● 고객이 어떻게 주문하는가?● 월별 매출 보고에 빠진 정보는 무엇인가?
자유 대답형 (많은 정보를 모을때)	<ul style="list-style-type: none">● 현재 주문을 처리하고 있는 방식에 대하여 어떻게 생각합니까?● 매일 겪는 문제점은 무엇입니까?● 주문 처리하는 방법에서 개선하고 싶은 것이 있다면 무엇입니까?
유도형 (직전 질문을 더 깊이 알고 싶을때)	<ul style="list-style-type: none">● 왜?● 예를 들어 줄 수 있나요?● 더 자세히 말씀해 주시겠어요?

5. 요구 사항 수집 방법

▪ 설문 조사

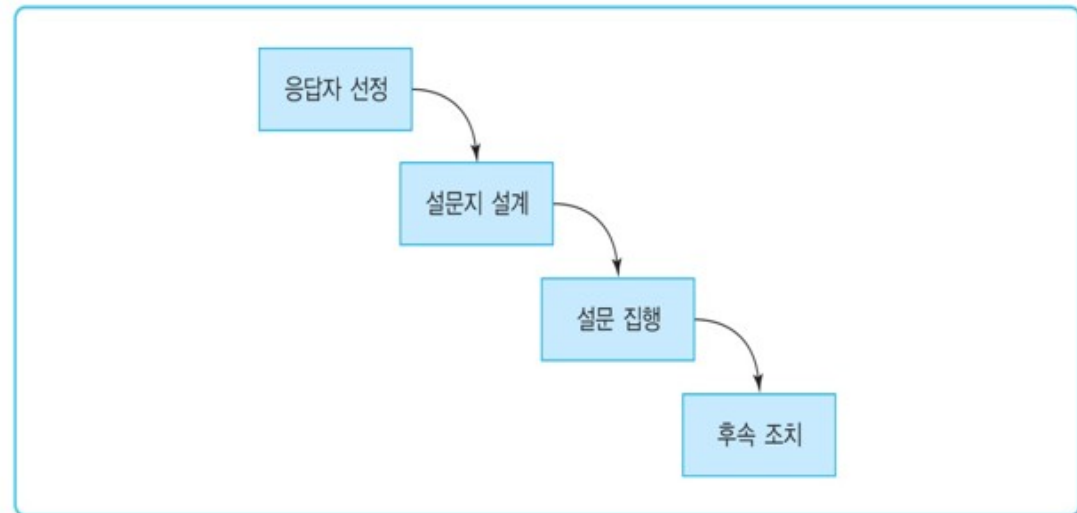
- 개인으로부터 정보를 취하기 위한 서면 질의서

- 광범위한 사용자로부터 의견이 필요할 때

- 지면, 이메일, 웹

- 응답자 선정

- 샘플링하여 대표 선발
- 평균 30~50%의 회수율 감안



5. 요구 사항 수집 방법

▪ 설문 조사

● 설문지 설계

- 일관된 형식
- 사실과 의견을 묻는 질문 구별
- 관심을 끌도록 유도

● 설문 집행

- 목적 및 DUE DATE 명기
- 부서장에게 협조요청

● 후속 조치

- 설문 결과 보고서 작성

- 위협을 주는 것이 아니라 **흥미를 유발시키는 질문**으로 시작한다.
- 각 질문을 논리적인 섹션으로 그룹핑하라.
- 중요한 질문을 설문의 끝에 두지 말 것
- 한 페이지에 너무 많은 내용을 담지 말 것
- 약어를 피할 것
- 치우치거나 제안하는 듯한 질문이나 단어를 피할 것
- 혼돈 피하기 위하여 질문에 번호를 매길 것
- 응답자에게 **익명을 보장**할 것

Section 03 요구 분석 절차와 요구 사항 종류

1. 요구 분석 절차와 요구 사항 종류

· 요구 분석 절차



그림 4-4 요구 분석 절차

- ① 자료 수집: **현행 시스템 파악**, 실무 담당자와 **인터뷰**, 현재 사용하는 **서류 검토**
- ② 요구 사항 도출 : 수집한 자료 정리 및 분류 → 개발에 반영할 **요구 사항 도출**
- ③ 문서화 : **요구 분석 명세서** 작성
- ④ 검증 : **요구 분석 명세서** 검토 → 모순 사항, 빠뜨린 사항 등 점검

2. 요구 사항 분류



그림4-5 요구사항 분류

2. 요구 사항 분류

기능적 요구

- 시스템이 수행하여야 할 처리나 정보 저장 등의 기능

- 비기능적 요구

- 성능, 사용용이성, 보안, 프로젝트 환경

기능적 요구

R1. 상품 검색

R2. 장바구니

R3. 상품 주문

R4. 마이리지

R5. 상품 찜

R6. 채고 탐색

인터넷 쇼핑몰



비기능적 요구

R101. 평균반응시간

R102 가용성

R103 재인ventario
호

R104 처리량

R105 데이터 복구

2. 요구 사항 분류

네비게이션 시스템

기능적 요구

1. GPS 위치

1.1 시스템은 지도의 어떤 부분을 디스플레이 할 것인지를 계산하기 위하여 GPS 정보를 사용하여 접근하고 있는 인근 지역 지도를 디스플레이 한다.

1.2 속도와 방향회전 기록에 대한 정보를 통합하여 자동차의 위치 정보를 표현한다.

2. 목적지 셋업

2.1 목적지 셋업을 위하여 시스템은 지도를 보여준다. 디폴트 지도는

1:25000 척도이며 중앙이 현재 자동차의 위치다.

2.2 목적지를 선택하기 위하여 주소나 신호를 입력받는다.

2.3 선택된 목적지를 찾아 위치를 지도에 디스플레이 하고 사용자에 게 확인 시킨다.

3. 네비게이션

3.1 사용자의 현재 위치가 지도의 가시범위 안에 있을 때는 빨간 화살표로 표시한다. 화살표는 사용자가 가야할 방향을 나타낸다.

3.2 목적지까지 주행로를 다음의 두 가지 방법으로 안내한다.

3.2.1 디스플레이 화살표 - 왼쪽, 오른쪽, U턴 심볼 등

3.2.2 음성 - 운전자에 어디로 운전하여야 하는지, <건물>에서 <방향>쪽으로 회전하십시오.' <도로번호>에 <건물>까지

<방향>쪽으로 직진하십시오'라고 안내한다.

비기능적 요구

1. 운용

1.1 시스템은 GPS 신호를 수신하는데 실패하더라도 서비스를 유지하도록 강인하여야 한다.

1.2 GPS 신호가 없는 경우 대신할 수 있는 관성항법 장치로부터의 입력을 받을 수 있도록 설계하여야 한다.

2. 성능

2.1 GPS에 의한 위치는 5미터 이내의 정확도를 유지하여야 한다.

2.2 시스템은 주행시 회전이나 길 안내는 적어도 100미터 전에 이루어져야 한다.

2.3 목적지의 탐색은 10초 이내에 이루어져야 한다.

3. 융통성

3.1 시스템은 무선 인터넷이나 GPS 기술의 변경이 미래의 릴리스에 잘 통합되도록 설계되어야 한다.

3. 비기능적 요구 사항

- 비기능적 요구 사항

- 수행 가능한 환경, 품질, 제약 사항

- 제약 사항의 예

- 자바 언어를 사용해 개발하고, CBD 개발 방법론을 적용해야 한다.
- 레드햇 리눅스 엔터프라이즈 버전에서 실행해야 한다.
- 웹로직 서버 WebLogic Server를 미들웨어로 사용해야 한다.
- 윈도우즈 운영체제와 리눅스 운영체제에서 모두 실행할 수 있어야 한다.

- 품질

- 신뢰성, 성능, 보안성, 안전성, 사용성

3. 비기능적 요구 사항

- 비기능적 요구

- 운용, 성능, 보안. 문화적. 정책적 프로젝트 환경

비기능적 요구 리얼	설명	사례
운용	시스템이 실행되는 물리적 기술적 환경	<ul style="list-style-type: none"> 시스템은 휴대용 장치 위에 실행된다. 시스템은 현재 창고 관리 시스템과 통합될 수 있어야 한다. 시스템이 어떤 웹 브라우저에도 작동되어야 한다.
성능	속도, 용량, 신뢰도	<ul style="list-style-type: none"> 사용자와 시스템 사이의 상호작용이 2초를 넘어서는 안 된다. 시스템은 백업 후 5분 내에 새로운 상태 파라미터를 다운받아야 한다.
보안	시스템에 누가 어떤 조건에 접근이 허용될 것인가?	<ul style="list-style-type: none"> 직속 상관만이 개인 기록을 열람할 수 있음 고객은 영업 시간 내에만 주문 내역을 확인할 수 있음
문화적 정책적	시스템에 영향을 주는 문화적 정책적 법적 요구	<ul style="list-style-type: none"> 시스템이 웰화와 외화를 구별할 수 있어야 한다. 회사의 규정에 특정한 하드웨어만을 구매했어야 한다. 채인 정보는 법률적으로 보호되어야 한다.

4. 품질(1)

① 신뢰성(reliability)

- 신뢰: 소프트웨어를 믿고 사용할 수 있는 것

사용자가 주어진 시간과 환경에서 고장 없이 사용할 수 있어야 한다는 것

- 신뢰도: 장애 없이 동작하는 시간의 비율

(예) 신뢰도 98%(0.98): 100번 수행했을 때 오류 없이 동작하는 회수가 98번

· 신뢰도 측정

- 고장 간 평균 시간(MTBF)과 이용 가능성(가용성)을 척도로 사용

$$MTBF = MTTF + MTTR$$

- MTBF 고장 간 평균 시간, Mean Time Between Failure : 고장에서 다음 고장까지의 평균 시간
- MTTF 평균 실패 시간, Mean Time To Failure : 수리한 후 다음 고장까지의 평균 시간
- MTTR 평균 수리 시간, Mean Time To Repair : 고장 발생 시점에서 수리 시까지의 평균 시간

$$\text{이용 가능성(availability)} = MTTF / (MTTF + MTTR) \times 100\%$$

- 이용 가능성(가용성): 주어진 시점에서 프로그램이 요구에 따라 작동되고 있을 가능성

4. 품질(1)

- 신뢰성(Reliability)의 평가

- ✓ 시스템 전체의 **가동률**
 - ✓ 시스템을 구성하고 있는 **각 요소의 신뢰도**
 - ✓ 신뢰성 향상을 위해 시행한 처리의 **경제적 효과**
 - ✓ 시스템의 **안정성을 평가하는 것으로 시스템이 고장 없이 많은 작업을 정확하게 처리하여 효율과 이익을 주는가를 평가하는 것**
-
- ✓ 평균 고장 시간(Mean Time Between Failure : MTBF)
 - ✓ 평균 수리 시간(Mean Time To Repair : MTTR)

4. 품질(1)

▪ MTBF/MTTR의 예제

가동 중

고장 중

가동 중

고장 중

가동 중

고장 중

a_1

b_1

a_2

b_2

a_3

b_3

✓ **평균 고장 시간(MTBF)=평균 가동 시간**

: 고장 완료 시점으로부터 다음 고장 발생 시점까지 작업이
가동 중인 시간의 평균값

$$\text{MTBF} = \frac{\text{작업한 시간의 총합}}{\text{작업한 횟수}} = \frac{(a_1 + a_2 + a_3)}{3}$$

4. 품질(1)

▪ MTBF/MTTR의 예제

✓ 평균 수리 시간(MTTR)

: 시스템 고장으로 인해 **가동하지 못한 시간의 평균값**

$$\text{MTTR} = \frac{\text{수리한 시간의 총합}}{\text{수리한 횟수}} = \frac{(b_1 + b_2 + b_3)}{3}$$

✓ 신뢰도 ★★★★★☆

: 전체 시간 중에서 **가동 중인 시간의 비율(가용도)**

$$\text{신뢰도} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} = \frac{(a_1 + a_2 + a_3)}{(a_1 + a_2 + a_3 + b_1 + b_2 + b_3)} = \frac{\text{가동 시간}}{\text{운용 시간}}$$

4. 품질(1)

▪ MTBF/MTTR의 예제

기출1] 시스템 운용 기간이 다음과 같을 때 평균 고장 시간(MTBF) 및 신뢰도는?

가동	고장	가동	고장	가동	고장
11	1	19	3	15	1

$$* \text{MTBF} = (11+19+15) / 3 = 15$$

$$* \text{신뢰도} = \text{MTBF} / \text{MTBF} + \text{MTTR}$$

$$= (11+19+15) / (11+19+15) + (1+3+1) = 45 / 45+5 = 45 / 50 = 0.9(90\%)$$

4. 품질(1)

▪ MTBF/MTTR의 예제

기출2) 시스템 운용 기간이 다음과 같을 때 **평균 가동 시간(MTBF)** 및 **신뢰도**는?

가동	고장	가동	고장	가동	고장
14	3	16	2	12	1

기출2) MTBF = $(14+16+12) / 3 = 14$

$$\text{신뢰도} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} = \frac{(14+16+12)}{(14+16+12)+(3+2+1)} = \frac{42}{48} = 87.5\%$$

4. 품질(2)

② 성능performance

- 사용자가 시스템에 어떤 요구를 했을 때 해당 기능을 **정상적으로 수행**하는 것은 물론, 사용자가 원하는 **조건(응답 시간, 데이터의 처리량 등)**을 만족시키는 것

(예 1) 대학 종합정보시스템은 수강 신청 시 **동시 접속자 수 10,000명은 가능**해야 한다.

(예 2) 도서 관리 시스템에서 사용자가 책을 검색한 **결과를 2초 이내**로 보여주어야 한다.

③ 보안성security

- **인증을 받지 않은 사람**이 시스템에 접근하는 것을 처음부터 막아 **시스템과 데이터를 보호**

④ 안전성safety

- 작동하는 모든 시스템이 소프트웨어 오류로 인해 **인명 피해가 발생**하지 않도록

(예 1) 공장의 생산 라인에 사람 손이나 물체가 인식되면 **절단 작업이 중단**되어야 한다.

(예 2) 전동차 출발 시 문에 가방 같은 물체가 끼면 **출발을 멈춰야** 한다.

4. 품질(3)

⑤ 사용성 usability

- 소프트웨어를 사용할 때 혼란스러워하거나 사용하는 순간에 고민하지 않게 하는 편의성
- 의미가 같은 단어 두 개를 혼용하여 사용자를 혼란스럽게 하는 경우
- 주민번호를 입력할 때 '-'를 넣어야 할지 말아야 할지 명확하게 제시해주지 못하는 경우

5. 사용자 요구 사항과 시스템 요구 사항

- **사용자를 위한 도면**

- 건축주와 일반인에게 설계 의도를 정확히 전달할 목적으로 작성된 도면
- 사용자가 알고 싶어하는 내용을 중심으로 작성
- 사용자 요구 분석 명세서 또는 요구 사항 정의서

- **시공자를 위한 도면**

- 전기 배선 기술자가 전기선을 포설하기 위해 필요한 도면
- 각 분야의 기술자가 작업하는 데 필요한 도면
- 실시 설계도: 사용자용 기본 설계도를 바탕으로 실제로 집을 짓는 데 필요해 작성한 도면
- 시스템 요구 분석 명세서에 해당

5-1 사용자 요구 사항

▪ 사용자 요구 분석 명세서

- 사용자 요구 사항을 정리하여 작성한 문서
- 건축의 사용자를 위한 도면과 유사
- 목적: 사용자와 대화 시 거부감을 줄이고 충분히 이해할 수 있도록 쉽게 작성
- 방법: 사용자가 이해할 수 있도록 전문 용어보다는 쉬운 용어 사용
- 도구: 다이어그램 사용
- 사용자와 분석가가 서로 충분한 대화를 나누며 함께 작성

▪ 사용자 요구 분석 명세서 작성 방법

- 유사한 프로젝트 경험을 가진 분석가 선정
- 표준 양식 사용
- 수집한 요구 사항에 대한 근거(출처) 마련

5-2 시스템 요구 사항

- **시스템 분석 명세서**

- 시스템 요구 사항을 설계하는데 도움이 되도록 **기술적 용어나 전문적 표현** 사용하여 작성
- 건축에서 **시공자를 위한 도면**과 유사
- 설계를 위해 사용되므로 **완전하고 일관성** 있게 작성

- **사용 도구**

- **구조적** 방법론의 **구조적 언어**
- **객체지향** 방법론의 **유스케이스 다이어그램**
- 검증에 강한 **Z 명세**와 같은 **정형화된 수학적 명세 언어**

Section 04 요구 사항의 표현

1. 요구 사항의 표현

- **표현**

- 음악(작곡, 작사, 노래), 미술(그림, 조각), 아이들의 그림, 수학

- **모델**

- 정의: 어떤 복잡한 대상의 핵심 특징만 선별하여 일정한 관점으로 단순화시켜 기호나 그림 등을 사용해 체계적으로 표현한 것
- 악보, 수학 공식, 모델 하우스, 플라스틱 자동차나 로봇, DNA분자 모델 등

- **모델의 필요성**

- 모델을 통해 실제 모습을 생각하고 확인해볼 수 있기 때문
(예) 모델하우스, 장난감 자동차, DNA 분자 모델

2. 모델

▪ 건축의 모델

- 하나의 사물을 여러 관점에서 바라보는 **건축의 도면**
- **조감도**: 건물 위에서 바라본 모습
- **평면도**: 건물을 수평으로 절단했을 때 바라본 모습
- **배치도**: 건물과 부지의 위치 관계를 나타낸 모습
- **배선도**: 전기 공사를 위해 전기선의 연결 관계를 나타낸 모습

▪ 소프트웨어 개발에서의 모델

- 여러 설계 도면을 보고 건물을 시공하는 것처럼 **SW 개발 시에도 여러 관점의 모델** 사용
- **UML의 다양한 다이어그램**을 통해 **소프트웨어의 범위나 개략적인 구조와 기능**을 이해

2-1 SW에서 모델 사용의 장단점

- **장점**

- 개발될 소프트웨어에 대한 **이해도 향상**, 이해 당사자 간의 **의사소통 향상**
- **유지보수 용이**

- **단점**

- 과도한 문서 작업으로 인한 **일정 지연** 가능성
- **형식적인 산출물**로 전락할 가능성

3. 모델링

- 모델링

- 모델을 제작하는 과정 또는 작업

- 현실 세계를 단순화하여 표현하는 기법

(예 1) 연필로 종이에 간단한 콘티를 작성하는 것

(예 2) 스토리보드같이 자기가 표현하고 싶은 것을 문장으로 서술하는 것

(예 3) 오션지에 음표를 그리는 것

- 소프트웨어 개발에서의 모델링

- UML 다이어그램을 이용하여 표현

(예 1) 요구 사항 표현 시 UML의 유스케이스 다이어그램 사용

(예 2) 자연어 사용

(예 3) 형식 언어의 형식적 표기법 사용

4. 모델링 언어

- 모델링 언어

- 애매모호한 표현 등의 문제점 해결하기 위해 모델링을 할 때 사용하는 기호, 표기법, 도구
(예) 악보 기호, 수학 기호, UML 다이어그램, Z 언어

- 개발 방법론에 따른 모델링 언어

- 구조적 방법론: DFDData Flow Diagram, DDData Dictionary , 프로세스 명세
- 정보공학 방법론: DB 설계 시 표현은 ERDEntity-Relationship Diagram
- 객체지향 방법론: UML 표기법

4-1 DFD

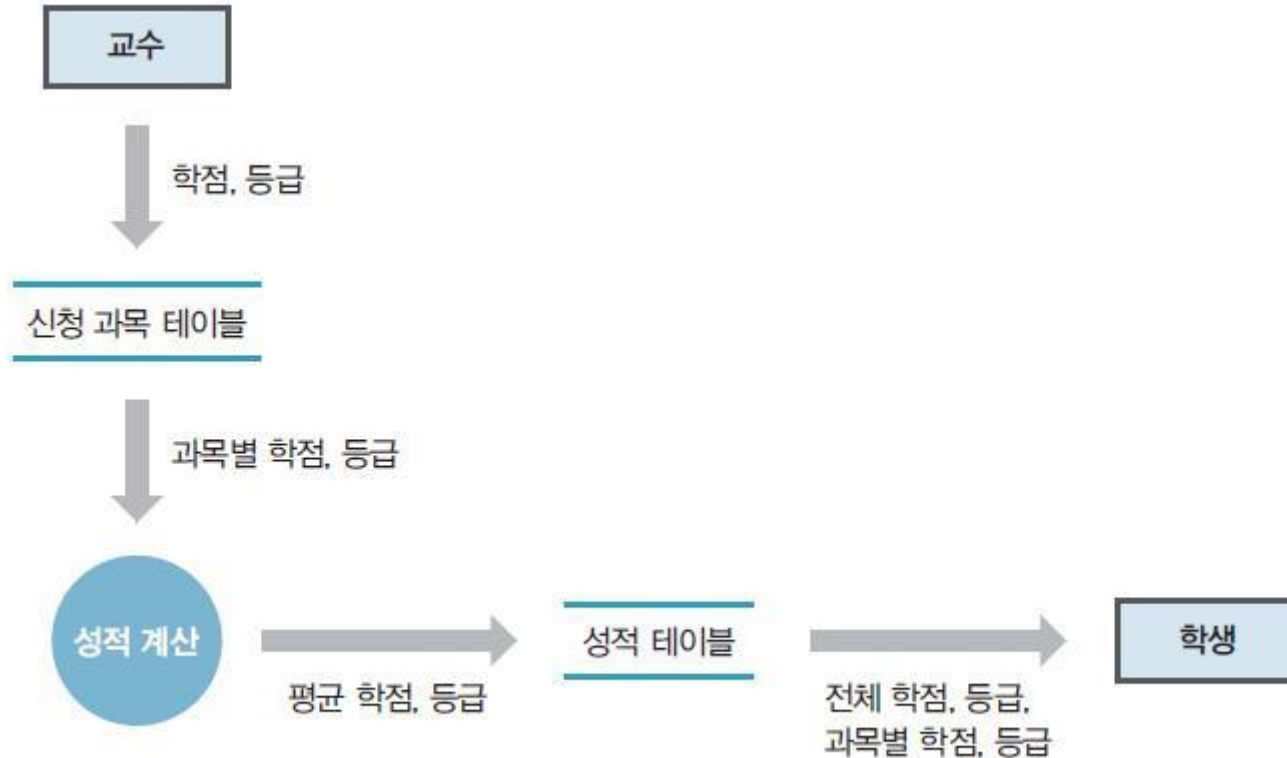
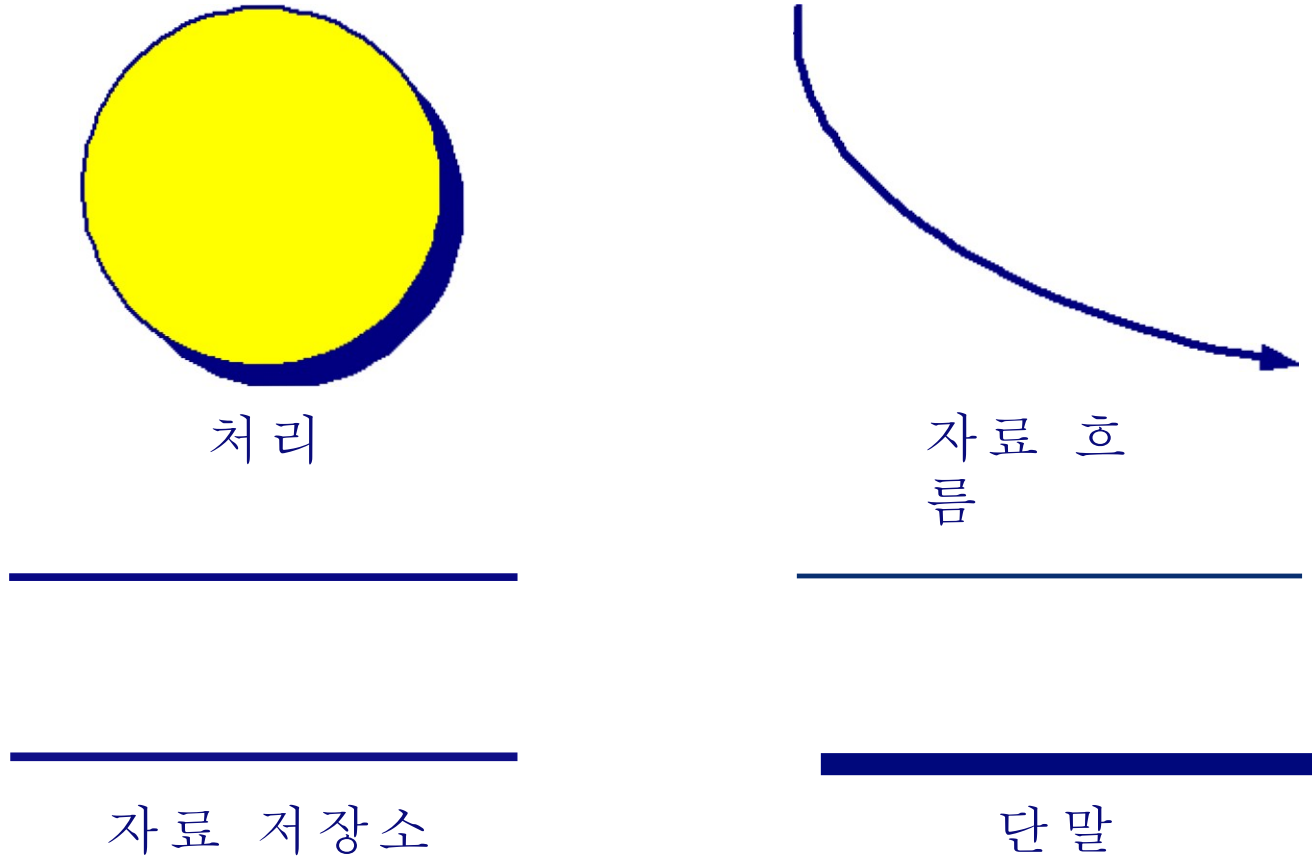


그림 4-6 DFD의 예

- **Terminator:** 출원지, 목적지를 나타냄(학생, 교수)
- **data flow:** 자료의 흐름을 나타냄(화살표)
- **data store:** 자료가 저장되는 곳을 나타냄(데이터베이스(테이블))
- **Process:** 자료를 입력 받아 처리하는 알고리즘(성적 계산)

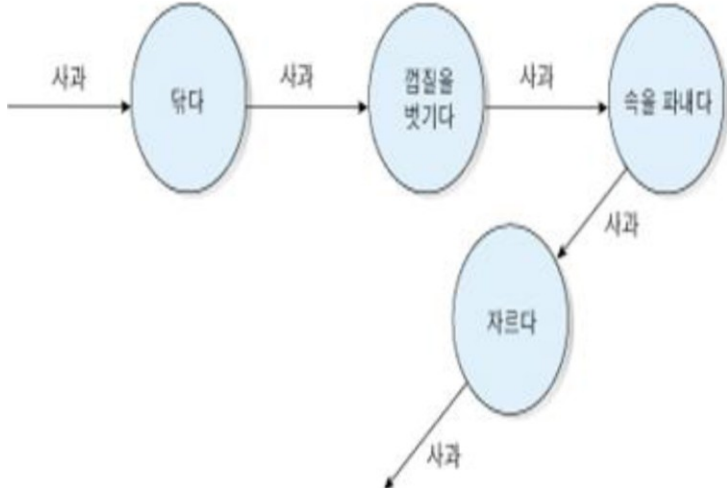
4-1 DFD

∴ X ~~흐름도~~의 구성 5요소

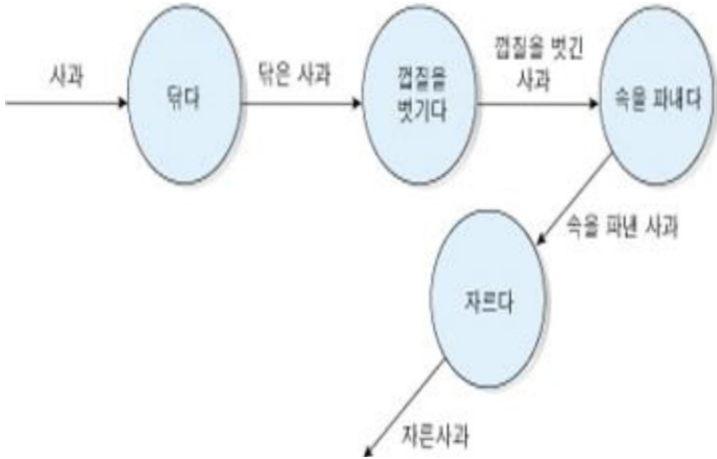


되 자토초 흐름(Data flow) {꺼 AI

지표호60이합했에



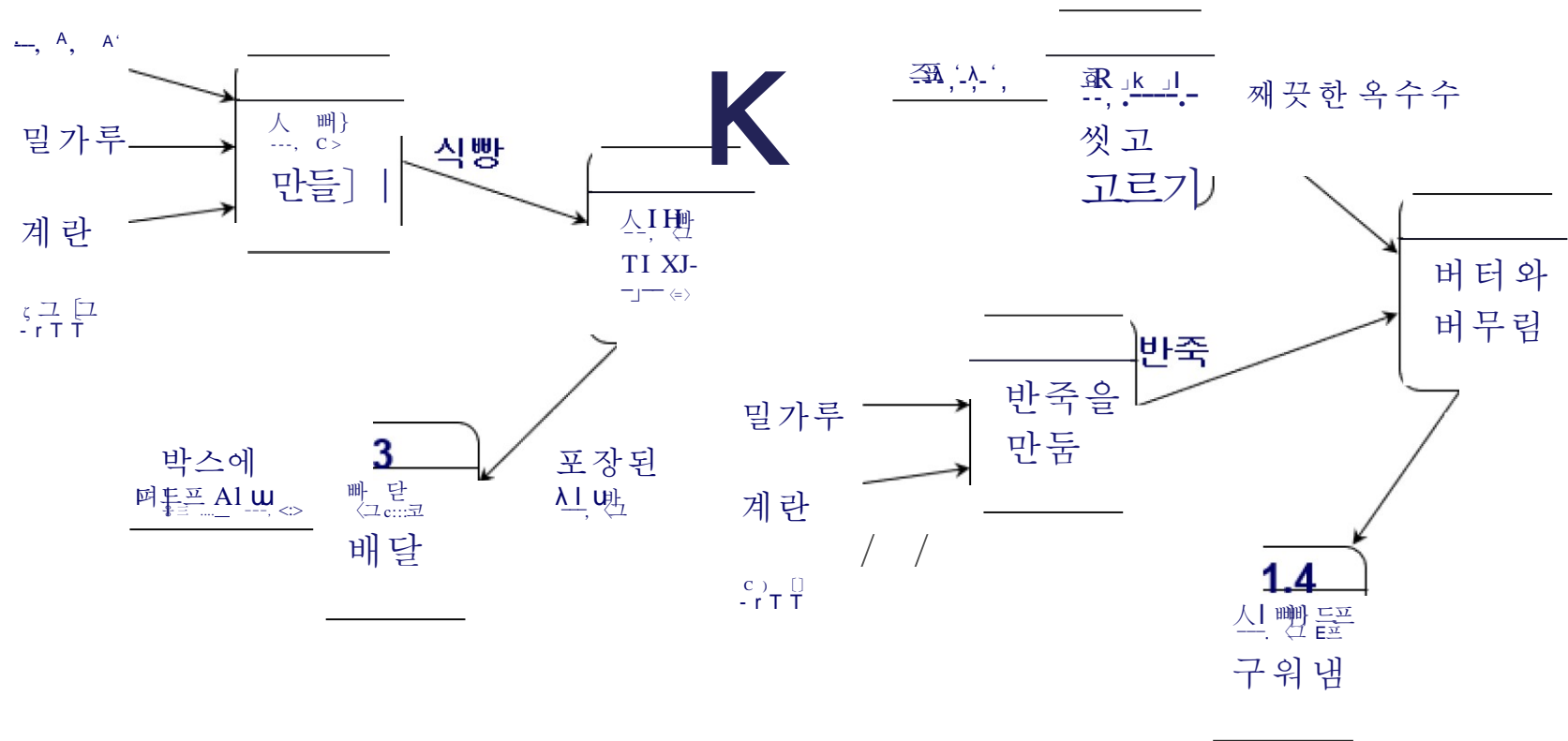
씨알 힘 겹 더



따림4-9]자알흐름

4-1 DFD

- 덩석빵공장 A 활동도 예



4-2 ERD

ER 다이어그램

- 데이터베이스에 저장할 데이터를 **개체(entity)**와 **관계(relationship)**를 중심으로 작성

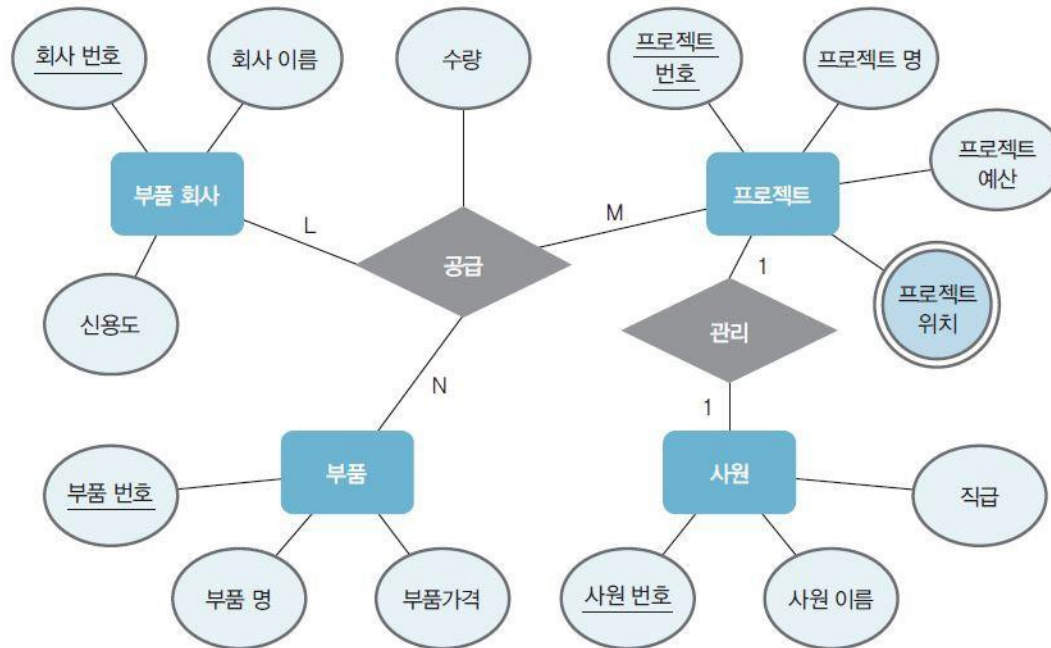



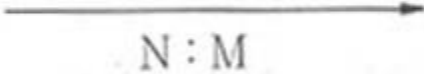


그림 4-7 ER 다이어그램의 예

4-2 ERD

▪ ER 다이어그램

- 데이터베이스에 저장할 데이터를 **개체**entity와 **관계**relationship를 중심으로 작성

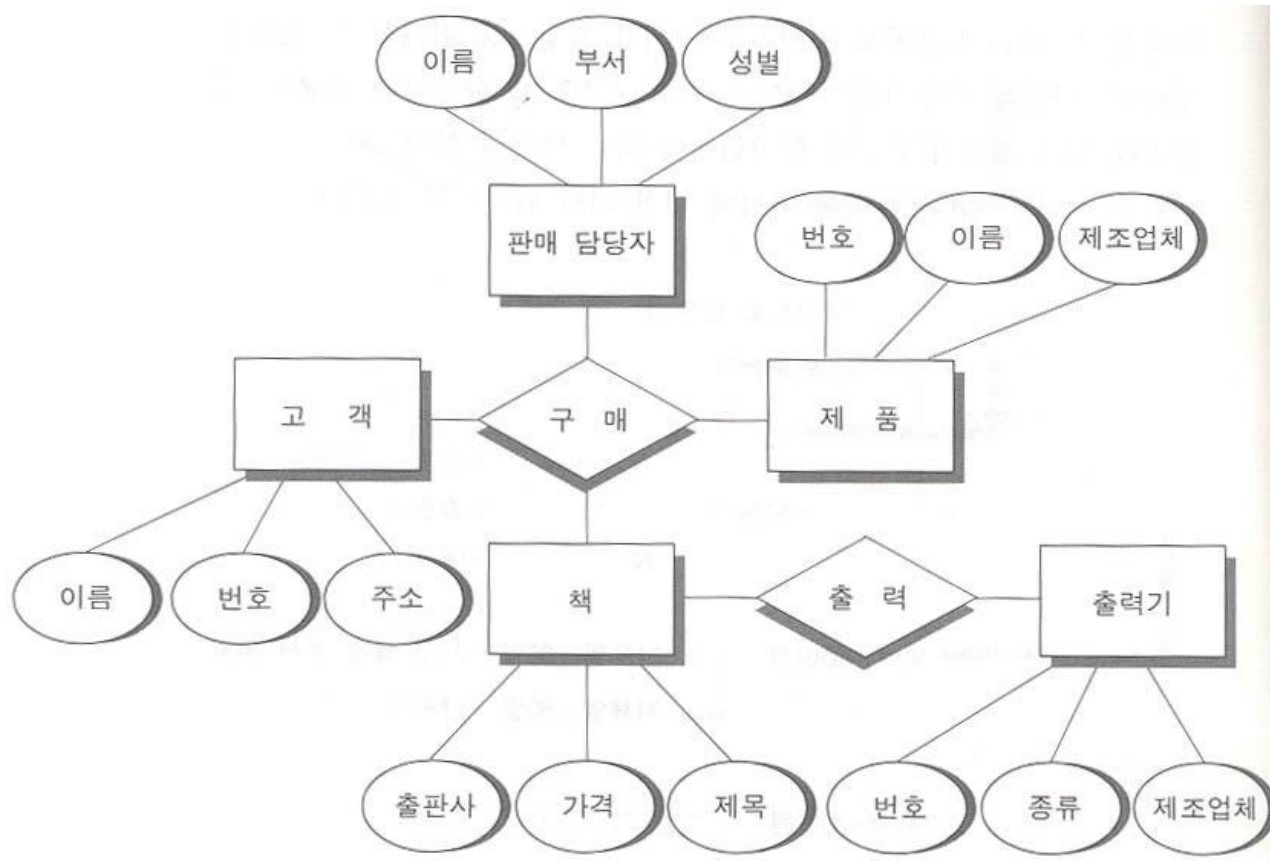
[표 8.2] ER 다이어그램의 사용 기호

기 호	의 미	기 호	의 미
	개 체 (entity)		링 크 (rink)
	관 계 (relationship)		속 성 (attribute)

4-2 ERD

ER 다이어그램

- 데이터베이스에 저장할 데이터를 **개체**entity와 **관계**relationship를 중심으로 작성



4-3 유스케이스 다이어그램(1)

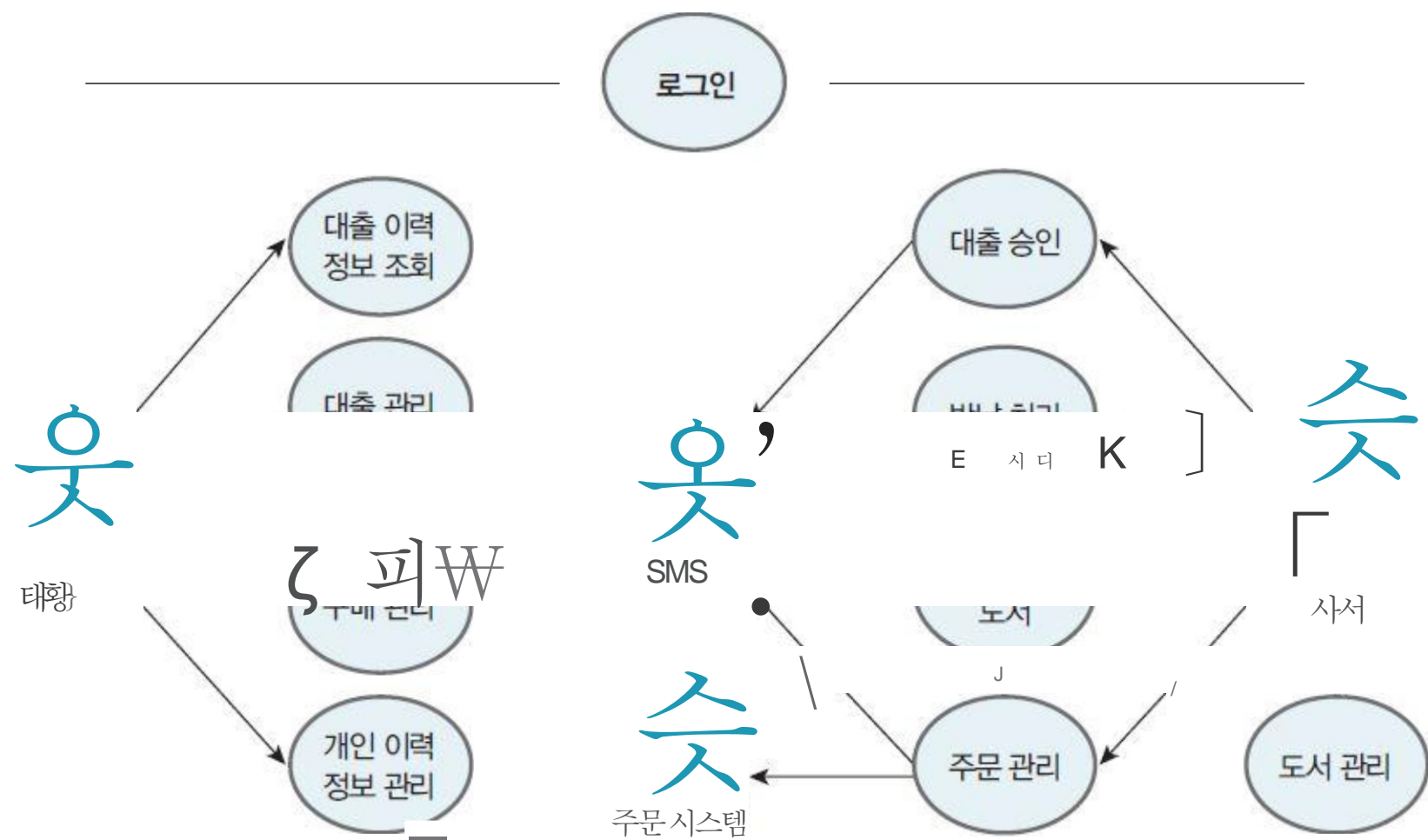


그림 4-8 유스케이스 다이어그램의 예

4-3 유스케이스 다이어그램(2)

- 유스케이스

- 사용자의 **요구**를 나타내는 기능
- 실제로 코딩할 수 있을 만큼의 **가장 작은 단위의 기능**

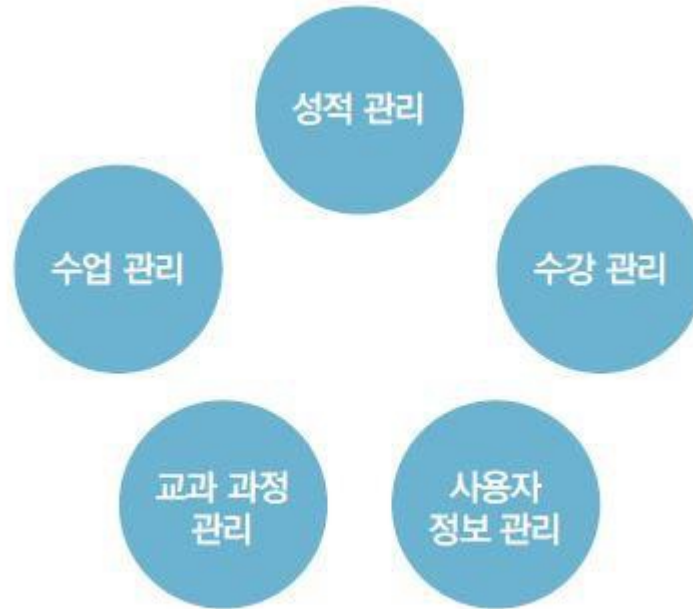


그림 4-9 유스케이스의 예

4-3 유스케이스 다이어그램(3)

▪ 사용자 액터

- 시스템을 사용하는 사람(역할)

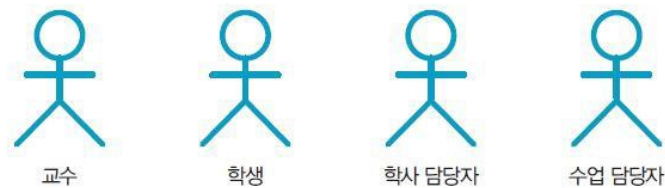


그림 4-10 사용자 액터의 예

▪ 시스템 액터

- 본 시스템과 데이터를 주고받는 등 서로 연동되는 또 다른 시스템
- 해당 프로젝트의 개발 범위에 속하지 않고 이미 다른 프로젝트에서 개발된 시스템
(예) 대학 종합정보시스템에서 등록납부 현황 자료를 받아오는 은행 시스템
(예) 입학생들의 기초 자료를 교육부로부터 받아오는 교육부시스템

Section 05 요구 사항의 문서화

1. 요구 사항의 문서화

▪ 요구 분석 명세서의 정의

- 요구 분석 과정의 최종 산출물로 사용자와 개발자를 연결시키는 중요한 문서
- 설계 및 구현에서 참조할 사항, 전반적으로 알아야 할 사항을 포함하는 문서
- 사용자와 개발자 간의 계약서

1-1 이해당사자 관점의 요구 분석 명세서

▪ 사용자 입장

- 사용자와 의사 소통하는 도구로 사용되면서 동시에 계약서로도 사용
- 개발이 완료 시 이 문서의 내용이 판단 기준으로 사용
- 개발된 소프트웨어의 수용(인수 테스트) 여부를 결정하는 데 사용

▪ 개발자 입장

- 요구 분석 명세서를 읽고 어떤 시스템이 개발될 것인지 이해하는데 사용
- 요구 분석 명세서에 기술된 기능적/비기능적 요구 사항을 기반으로 분석, 설계, 코딩
- 개발이 완료 후 요구 분석 명세서 대로 구현되었는지 점검 항목으로 사용
- 사용자 지침서 초안 작성용으로 사용

▪ 테스터 입장

- 테스트 케이스 생성 및 오류에 대한 판단과 동작에 대한 기준으로 사용

2. 요구 분석 명세서 작성 시 주의 사항(1)

- 사용자가 읽기 쉽고, 이해할 수 있도록 작성



그림 4-12 깨알 같은 작은 글씨의 약관

- 개발자가 설계와 코딩에 효과적으로 사용할 수 있도록 작성
- 비기능적 요구를 명확히 작성

2. 요구 분석 명세서 작성 시 주의 사항(2)

- 테스트 기준 용도로 사용할 수 있도록 **정량적으로 작성**
- **품질에 대한 우선순위를 명시**
 - 상충 관계^{trade-off} 시 **품질 우선 순위**를 정할 것



그림 4-13 상충 관계^{trade-off}의 예

3. 잘 만든 요구 분석 명세서의 특성(1)

■ **완전성** completeness

- '완전하다'는 말은 빠진 부분 없이 **모두 있다**는 의미이다.
- **기능적 요구사항**뿐 아니라 **성능, 제약 사항** 등 누락되지 않고 모두 서술되어야 한다.

BUT!

- 가끔씩 빠트리는 **요구 사항 존재**
→ 일반적이고 정상적인 요구 사항이 아닌 **예외처리**처럼 아주 드물게 발생하는 요구 사항

■ **명확성** unambiguity

- 요구 분석 명세서: 계약서와 같은 효력 발생 → 문제 발생 시 근거 자료로 활용
- ∴ 애매모호하지 않은 **명확한 표현**으로 작성 → 관점에 따라 **다른 해석 불가**하도록 작성

3. 잘 만든 요구 분석 명세서의 특성(2)

- **일관성**consistency, **무모순성**

- 일관성 없는 명세서: 서로 상반된 요구, 불일치한 요구, 중복된 요구가 존재

- **변경 용이성**modifiability

- 변경하기 쉽게 요구 분석 명세서를 작성하는 것
- 방법: 요구 사항이 서로 의존적이지 않고 독립적으로 서술되어야 함

- **검증 가능성**verifiability

- 방법: 시스템이 요구 사항을 만족하는지에 대해 체계적으로 검사할 수 있게 작성
- 잘못된 예: 많은 학생이 동시에 수강 신청을 해도 문제가 없어야 한다.
- 잘된 예: 10,000명의 학생이 동시에 수강 신청을 해도 문제가 없어야 한다.

3. 잘 만든 요구 분석 명세서의 특성(3)

■ 추적 가능성 traceability

- 추적이 가능하도록 요구 분석 명세서를 작성하는 것

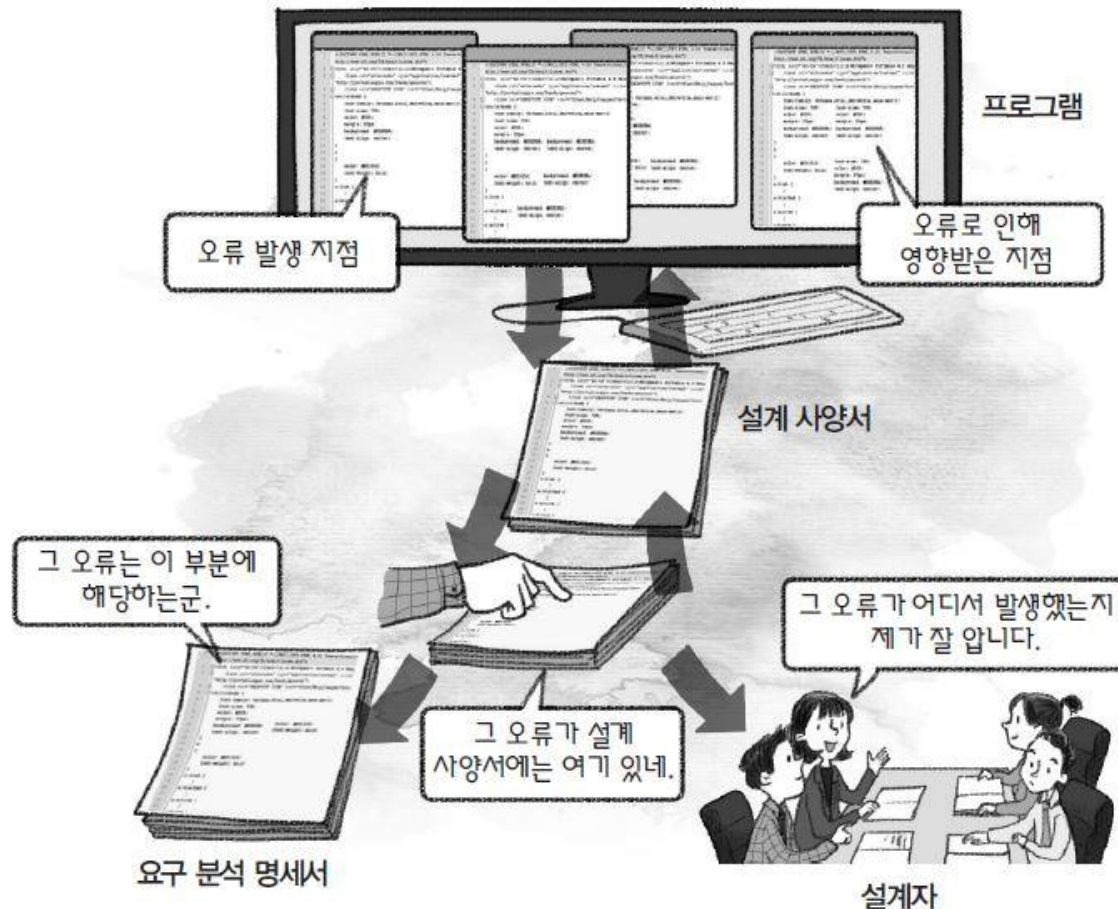


그림 4-14 추적 가능성

4. 요구 명세 기법

▪ 비정형 명세 기법

- 자연어, 다이어그램 사용

(장점 1) 특별한 기술이 필요 없어 작성하기 쉽다.

(장점 2) 쉬운 이해 → 용이한 의사 전달 → 사용자의 적극적 참여 유도

(단점 1) 자연어 사용 → 애매모호한 표현 → 다른 해석 가능 → 일관성 떨어짐

▪ 정형 명세 기법

- Z 정형 명세 언어(수학적 원리와 기법 사용)

(장점 1) 정확하고 간결한 표현 → 증명 기술을 이용한 일관성/완전성 검증

(장점 2) 정형화된 형태의 명세 → 테스트케이스 생성 용이

(단점 1) 수학적 표기법 공부 → 표기법을 이용한 정확한 표현

5. 요구 사항 검증

▪ 요구 사항 검증

- 정리한 사용자의 요구 분석 명세서가 정확하고 완전하게 서술되었는지 검토하는 활동
 - **완전성**completeness : 모든 요구 사항이 **누락되지 않고** 완전하게 반영되고 있는가?
 - **일관성**consistency : 요구 사항 간에 **모순되거나 충돌되지** 않고 일관성을 유지하는가?
 - **명확성**unambiguity : **표현이 애매모호하지** 않고 참여자가 명확히 이해할 수 있는가?
 - **기능성**functionality : '**어떻게**'보다 '**무엇을**'에 관점을 두고 서술되었는가?
 - **검증 가능성**verifiability : 사용자가 **요구하는 내용과 일치**하는지를 검증할 수 있는가?
 - **추적 가능성**traceability : 사용자 **요구 분석 명세서와 설계 사양서를 추적**할 수 있는가?
 - **변경 용이성**easily changeable : 변경 시 **쉽게 찾아 변경**할 수 있도록 작성되었는가?

6. IEEE Std. 830-1998에서 권고하는 요구 분석 명세서의 항목

1. 소개

- 1.1 목적
- 1.2 범위
- 1.3 정의약어
- 1.4 창조
- 1.5 7빠

2. 헨 적 서 술

- 2.1 제품 관련 기능
 - 사용자 인터페이스
 - 하드웨어 인터페이스
 - 소프트웨어 인터페이스
 - 통신 인터페이스
 - 메모리
 - 운영
- 2.2 제품 기능
- 2.3 사용자 특성
- 2.4 제약 사항
 - 규제 정책 하드웨어 제약 사항, 다른 응용 프로그램과의 인터페이스, 병렬 수행 감사 기능, 제어 가능, 신뢰성 요구 사항, 안전 및 보안 요구 사항
- 2.5 가정 및 의존성
- 2.6 요구 사항 할당

3. 구체적인 요구 사항

- 3.1 외부 인터페이스
 - 사용자 인터페이스 하드웨어 인터페이스, 소프트웨어 인터페이스 통신 인터페이스 가능
- 3.2 성능 요구 사항
- 3.3 로컬 DB 요구 사항
- 3.4 설계 제약 사항
- 3.5 소프트웨어 시스템 속성
 - 신뢰성, 가용성, 보안성, 유지 보수 용이성

인덱스
부록