소프트웨어공학

한성대학교 컴퓨터공학부

소프트웨어 개발 모델

- 소프트웨어 라이프 사이클/소프트웨어 수명(생명) 주기

그림 1-1 소프트웨어 생명주기

소프트웨어 개발 단계

- 요구사항 분석: 개발하고자 하는 소프트웨어에 대한 요구사항을 고객으로 부터 수집하고 분석하며 명세하는 단계
- 설계: 고객의 요구사항을 만족하기 위한 여러 해결책을 제시하고 이 중에서 가장 최적화된 해결책을 선정하는 단계
- 구현: 고객의 요구사항을 실제 서비스의 형태로 제공할 수 있도록 프로그래밍 언어를 사용하여 개발하는 단계
- 테스트: 개발된 프로그램이 고객의 요구대로 동작이 되는지를 시험하는 단계

유지보수

- 수정 유지보수(corrective maintenance): 소프트웨어의 오류가 발견되었을 때 이를 수정하는 작업
- 적응 유지보수(adaptive maintenance): 운영체계나 인프라 환경 등이 변화되었을 때 이 변화를 수용하도록 프로그램을 수정하는 작업
- 완전유지보수(perfective maintenance): 기능이나 성능을 개선하거나 새로운 기능을 추가하기 위하여 프로그램을 수정하는 작업
- 예방 유지보수(preventive maintenance): 수정 유지보수와 달리 소프트웨어의 오류가 발생되기 전에 미연에 방지될 수 있도록 수행하는 작업

소프트웨어의 이해

- 소프트웨어의 정의
  - 프로그램: 프로그래밍한 원시 코드(source code)
  - 소프트웨어
    - 프로그램(코드)을 비롯해 개발 과정에서 생성되는 모든 산출물과 각 단계에서 만들어지는 문서와 사용자 매뉴얼 등
      - (자료 구조, 데이터베이스 구조, 테스트 결과 등)

그림 1-1 소프트웨어가 사용되는 곳

01. 소프트웨어의 이해

- 소프트웨어의 특징
  - 제조가 아닌 개발
    - 제조: 개인 능력에 따라 차이가 크므로 능력에 따른 결과물의 차이는 크지 않음
    - 소프트웨어 개발 과정은 제조와 달리 개인 능력에 따라 차이가 큼
  - 소모가 아닌 품질 저하
    - 하드웨어: 오래 사용하면 부품이 닳고 기능도 떨어짐
    - 소프트웨어: 닳지 않으며 또한 시간이 지나도 고장 빈도가 높지 않음, 사용 시작 단계부터 사용자의 요구가 계속 발생

01. 소프트웨어의 이해

- 소프트웨어 개발의 어려움

①개집 짓기

- 연장과 나무만 있으면 뚝딱뚝딱 빠른 시간에 지을 수 있음
- 도면을 따로 그리지 않고 머릿속으로 구상만 해도 가능
- 지은 집이 균형이 조금 맞지 않아도 사용하는 데 큰 문제가 없음

그림 1-4 개집 짓기

01. 소프트웨어의 이해

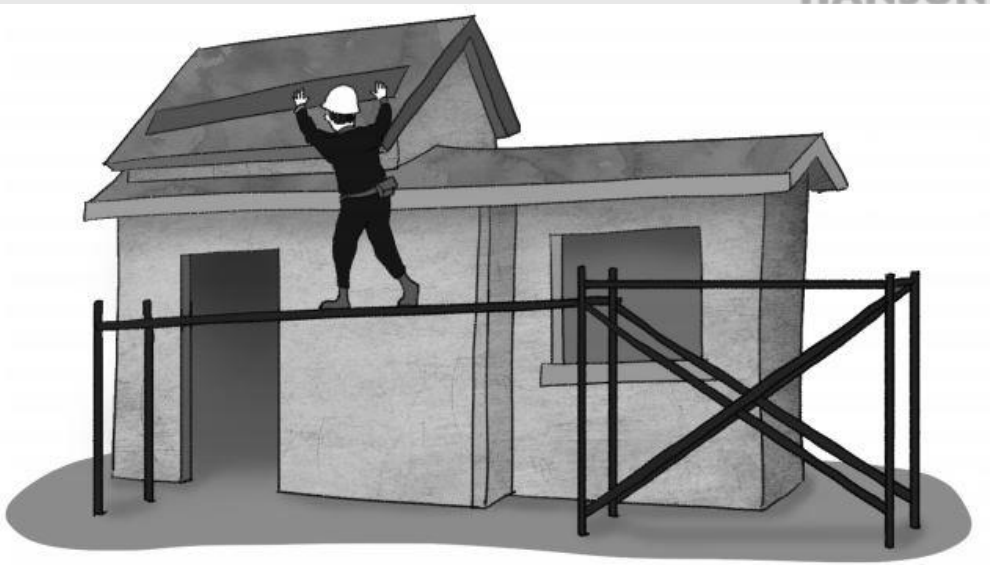

그림 1-5 단독주택 짓기

- 소프트웨어 개발의 어려움

② 단독주택 짓기

  - 개집과 달리 여러 사람이 참여해야 하며 설계와 공정 과정도 필요
  - 단순한 연장만으로는 안 되고 전문적인 도구들이 필요

01. 소프트웨어의 이해

그림 1-6 대형 빌딩 짓기

- 소프트웨어 개발의 어려움

③ 대형 빌딩 짓기

- 지반이 안전한지부터 확인한 후에 토목공사가 이루어짐
- 지진에 대비한 내진 설계, 하중 문제 등 고려해야 할 사항도 훨씬 많고 복잡
- 그래서 수많은 사람과 여러 작업을 중앙에서 조정하고 통제하는 팀이나 부서가 필요

01. 소프트웨어의 이해

•소프트웨어 개발의 어려움

- 대규모 소프트웨어를 개발하는 것은 대형 빌딩 짓기와 유사

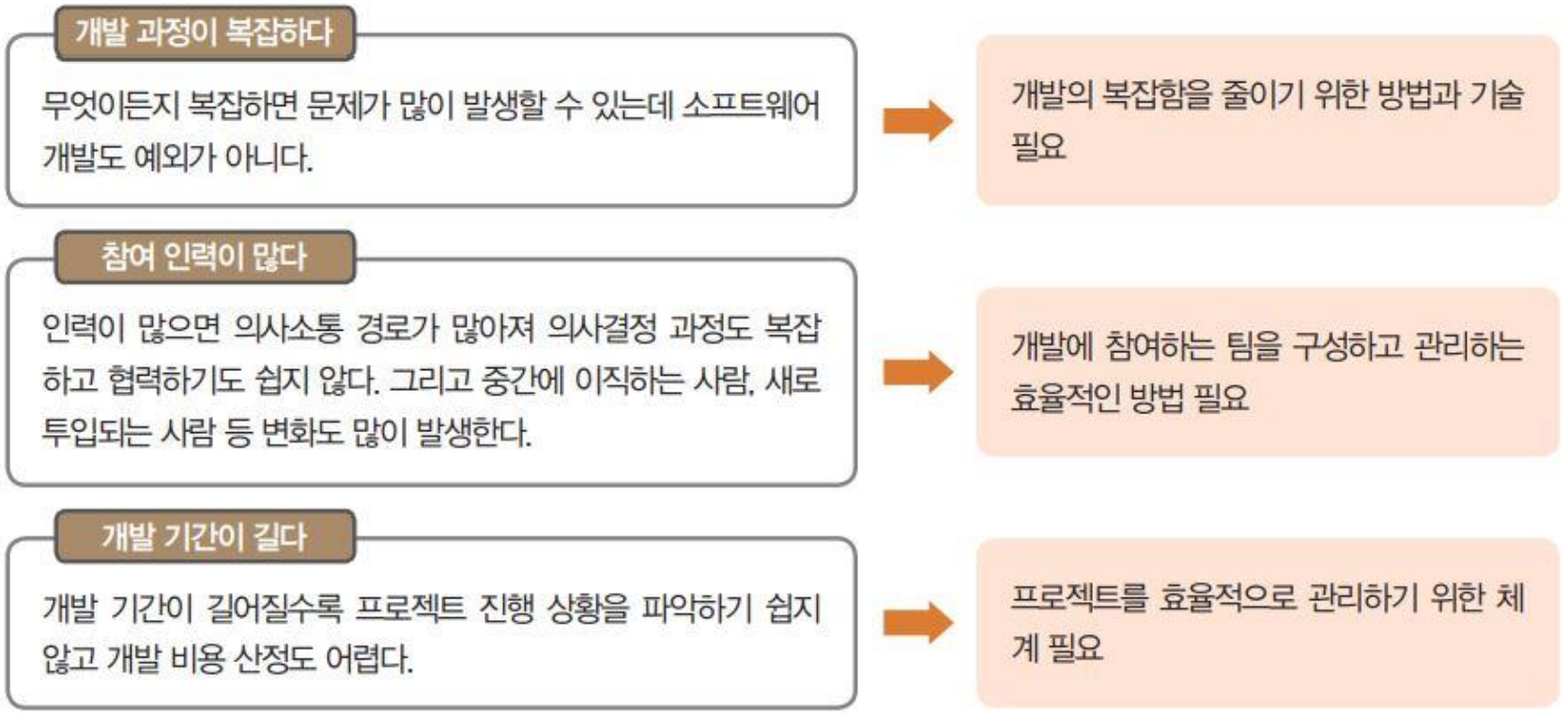

그림 1-7 대규모 소프트웨어 개발의 어려움

01. 소프트웨어의 이해

- 단점만 있는 것이 아니다. 소프트웨어 개발 프로세스는 일반적인 다른 시스템의 개발 프로세스와는 차이가 있다. 소프트웨어는 눈에 보이지 않으므로.. 소프트웨어는 만들어 놓고 잘못 만들었을 때, 다시 만드는 데 비용이 적게 든다.
- 잘못되더라도 돌아오는데 비용이 다른 시스템에 비해 비용이 적게 든다.

SW는 반복적으로 하는 것이 중요

01. 소프트웨어의 이해

- 소프트웨어 공학
  - 소프트웨어 공학의 학문적 정의
    - 품질 좋은 소프트웨어를 경제적으로 개발하기 위해 계획을 세우고, 개발하며, 유지 및 관리 하는 전 과정에서
      공학, 과학 및 수학적 원리와 방법을 적용해 필요한 이론과 기술 및 도구 들에 관해 연구하는 학문

  - 소프트웨어 개발 생명주기
    - 소프트웨어를 만들기 위해 계획 단계에서 유지보수 단계에 이르기까지 일어나는 일련의 과정
    - (계획, 분석, 설계, 구현, 테스트, 유지보수)

02. 소프트웨어 개발 프로세스

그림 1-9 요리 프로세스

- 프로세스
  - 프로세스: 흔히 일을 처리하는 과정 또는 순서
  - 주어진 일을 해결하기 위한 목적으로 그 순서가 정해져 수행되는 일련의 절차
  - 어떤 일을 해결하고자 할 때는 해당 프로세스만 잘 따르면 목적을 달성할 수 있음
    - 예) 요리, 빨래, 소프트웨어 설치 등…

02. 소프트웨어 개발 프로세스

- 소프트웨어 개발 프로세스
  - 소프트웨어 개발 프로세스의 의미
    - 좁은 의미: 소프트웨어 제품을 개발할 때 필요한 절차나 과정
    - 넓은 의미: 작업을 수행하는 데 필요한 방법과 도구를 비롯해 개발과 관련된 절차를 따라 작업을 수행하는 참여자 포함

01. 소프트웨어의 이해

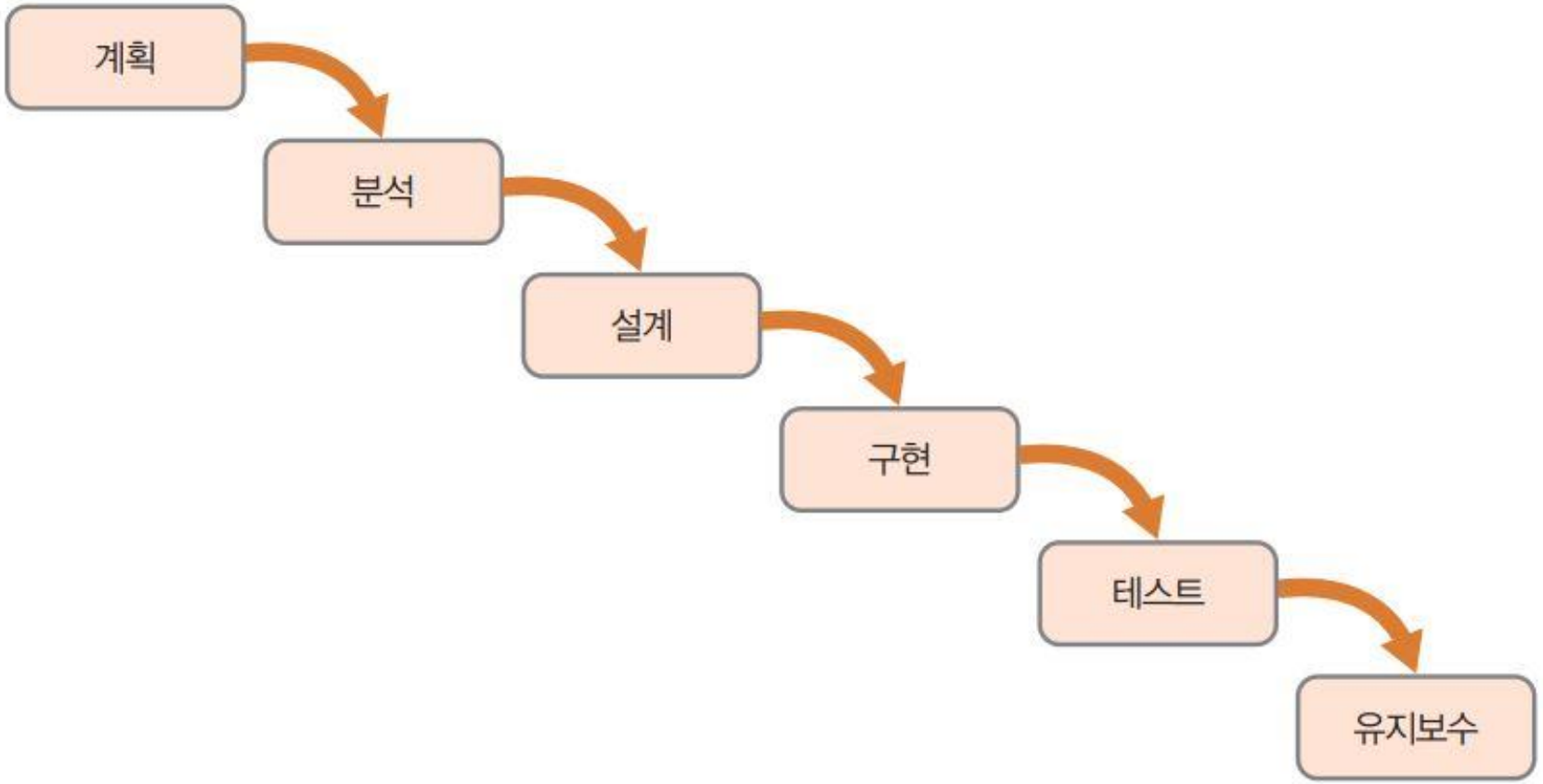

그림 1-8 소프트웨어 개발 생명주기

요구사항 분석

01. 요구사항

- 요구사항 분석
  - 시스템의 목표를 확립하는 과정
  - 시스템이 만족시켜야 할 요구사항의 발견, 정제, 모델링, 명세화하는 과정
  - 새롭게 만들고자하는 시스템의 명세를 만들어내는 과정
    - ▶ 시스템이 만족시켜야 할 기능, 성능, 다른 시스템과 인터페이스 규명
  - 요구사항 분석의 최종 산출물: 요구사항 명세서 = 기능 명세서 = 목표 문서

특징

- "How" 대신 "What"에 초점이 맞춰져 있음
- 고객들과 협상하여 공동의 목표를 끌어내야 함
- 실패나 실수를 방지하고 프로젝트 초기 단계의 중대한 실수를 최소화

소프트웨어는 눈에 보이지도 않고 만져지지도 않는다.

그래서 요구사항 분석이 어렵다.

소프트웨어의 모호성을 없애주는 것이 중요, 모호성이 없어져야 프로그램을 짤 수 있다.

요구사항 분석은 한 번에 딱하고 끝나는 게 아니다. 목적에 맞게 끔만 하고 넘어간 다음에 또 하고, 또 하고, 애자일로 가면 그게 극단적으로 일어나는 것

•요구사항은 변할 수 밖에 없다? 왜?

- 요구 사항의 잘못된 정의
- 고객과 개발자간의 잘못된 의사소통이나 부재

01. 요구사항

- 요구사항
  - 요구사항의 정의
    - 사전적 정의: '이용자가 어떤 문제를 풀거나 목표를 달성하는 데 필요한 조건이나 능력'
    - 소프트웨어 개발에서의 정의: '사용자와 개발자가 합의한 범위 내에서 사용자가 필요로 하는 기능'

      시스템이 제공하는 기능 요구와 품질과 같은 비기능 요구로 나뉨
    - 요구사항이 정확히 무엇인지 파악하는 작업은 요구분석 단계에서 이루어짐

02. 요구분석의 이해

그림 4-2 건축 설계 과정

- 요구분석의 정의와 목적
  - 집짓는 과정의 예시
    - 집을 지으려는 사람은 본인이 수집한 자료와 구상한 내용을 토대로 건축 설계사에게 어떤 집을 짓고 싶은지 설명
    - 본인이 가지고 있는 예산과 언제 입주하고 싶은지, 건축에 필요한 제반 사항 등에 대해서도 이야기를 나눔
    - 이 과정에서 고객은 어떤 건물을 원하는지 충분히 설명할 수 있고 건축 설계사는 고객이 원하는 것을 설계 도면에 빠짐없이 반영할 수 있음

02. 요구분석의 이해

- 요구분석의 정의와 목적
  - 요구분석의 정의
    - 컴퓨터 용어사전: '시스템이나 소프트웨어의 요구사항을 정의하기 위해 사용자 요구사항을 조사하고 확인하는 과정
    - 요구분석은 소프트웨어 개발 성패의 열쇠
  - 요구분석의 목적
    - 목적사용자에게서 필요한 요구사항을 추출해 목표하는 시스템의 모델을 만들고 '요구분석 명세서'를 작성하기 위해서
    - 요구분석명세서
      - 요구분석 단계에서 생성 되는 최종 산출물로 시스템의 기능이 무엇인지(what)에만 초점을 두고 정리
      - 요구분석단계 후 설계 단계에서는 '설계서'가 만들어지는데 이 문서는 어떻게(how) 구현할지 기술

02. 요구분석의 이해

- 요구분석 관련자(대학 학사관리시스템 개발의 예시)
  - 발주사: 외주 방식으로 학사관리시스템 개발을 의뢰한 A대학교
  - 경영자(총 책임자): 학사관리시스템 개발을 결정한 A대학교 총장
  - 발주 담당자: 학사관리시스템 개발을 외주로 진행하는 모든 절차를 준비하는 담당자
  - 사용자: 외주 방식으로 개발된 학사관리시스템을 실제 사용하는 사람 (학생, 교수, 조교, 학사담당직원 등)
  - 수주사: A대학교의 학사관리시스템 개발을 위한 응찰 경쟁에서 이겨 개발이 결정된 업체 (B개발사)
  - 분석가: 사용자 요구사항이 무엇인지 파악하고 추출 및 정리하는 것까지 담당

    B개발사의 분석가는 A대학교를 방문해 현행 시스템 의 현황과 문제점을 파악하고 새로운 요구사항을 수집해 최종 산출물인 요구분석명세서를 작성
  - 설계자: 요구분석명세서를 바탕으로 아키텍처 설계, 모듈 설계, 데이터베이스DB 설계, 사용자 인터페이스 설계 등을 담당
  - 개발자: 넓은 의미에서 개발자는 학사관리시스템 개발에 참여하는 B개발사의 분석가, 설계자, 프로그래머 등을 포함

    좁은 의미에서는 프로그래머

02. 요구분석의 이해

- **요구분석의 어려움**
  - **문제 영역에 대한 분석가의 이해력 부족**
    - 법적인 문제를 해결할 경우 해당 분야에 전문적인 변호사를 선임
    - 요구분석가는 IP 분야의 전문가라 문제 영역(회계 분야, 금융 분야, 기업의 통합관리 등)에 대한 이해력이 부족할 수 있음
    - 사용자 요구를 잘못 이해한 채 분석해 필수 요구사항을 빠뜨릴 수도 있음
    - 이런 문제를 최소화하려면 문제 영역과 관련된 프로젝트를 개발한 경험이 많은 분석가를 투입하는 것이 바람직

그림 4-4 분야별 전문 변호사의 활동 모습

그림 4-5 경험 많은 분석가들

02. 요구분석의 이해

그림 4-6 요구사항을 계속 추가하는 사용자

- **요구분석의 어려움**
  - **사용자와 분석가의 의사소통 문제**
    - 소프트웨어 개발에서는 견본이 없는 경우가 대부분이므로 사용자가 원하는 것이 무엇인지 분석가에게 사용자가 설명하기가 어려움
    - 원하는 바를 분석가에게 어떻게 설명해야 할지 잘 모름
    - 필요한 요구사항은 반영하지 못하고 불필요한 사항만 포함되기도 함
    - 의사소통 문제는 개발자 간에도 있을 수 있음
  - **사용자의 계속되는 요구사항 추가**
    - 사용자의 처음 요구는 단순하고 간단할 수 있지만 분석가와 대화 후 점점 새로운 생각이 늘어나거나 관련 지식이 늘어남
    - 그에 따라 새로운 요구가 생기고 요구사항이 계속 추가될 수 있음
    - 소프트웨어 개발 과정 중에 목표 환경이 달라지면서 요구사항이 변경되기도 함

02. 요구분석의 이해

그림 4-7 조율에 능숙해야 하는 분석가

- 요구분석의 어려움
  - 사용자의 모호한 요구사항
    - 사용자는 분석가에게 모호하게 요구사항을 전달하면 분석가가 해석할 수 있는 여지가 많으므로 오해로 인한 문제가 발생
    - 세부적인 내용을 전달하지 않는다면 분석가는 세부적인 요구사항을 본인의 경험에 비추어 해석해 사용자의 의도와 달라짐
    - 부서 간의 요구가 서로 어긋나고 경영진, 실무자의 요구가 상반되어 일관성이 없고 모순되는 요구사항이 발견될 수 있음
    - 분석가는 이러한 요구를 정리해서 반영 하기 위해 이해 당사자 간의 주장을 조율해야 함

02. 요구분석의 이해

- 요구사항 수집
  - 자료 수집
    - 가장 먼저 할 일은 A대학교의 현행 시스템에서 모든 업무의 입력 화면과 출력물을 받아 기본 기능을 분석하는 것
    - 관련 직원으로부터 현행 시스템의 문제점과 개선점, 새로 추가되어야 할 요구사항을 파악
  - 인터뷰
    - 가능하면 먼저 자료를 수집한 후 이를 정리해 분석한 결과를 바 탕으로 각 부서 담당자를 인터뷰하는 것이 효율적
    - 미리 받은 자료를 토대로 대화를 해야 큰 줄기를 흩트리지 않고 계획대로 진행할 수 있음
    - 오랜 시간 동안 대화하기보다는 핵심 요구사항을 빨리 습득하는 것도 능력
  - 설문 조사
    - 설문 조사를 통해 또 한 번의 요구사항을 도출할 수 있음
    - 중요한 것은 효율적인 설문 조사를 위해 설문 문항을 잘 만들어야 한다는 것

소프트웨어 시스템은 눈에 보이지 않기 때문에 요구사항을 찾아내기도 굉장히 어렵다.

먼저 이해관계자를 찾아야 한다 : 소프트웨어 시스템의 이해관계자는 굉장히 많다.

보통 이해관계자를 사용자라고만 생각한다.

=> 사용자, 고객, 팀장, 임원, 사장, 개발자, 마케팅, 기획, 영업, 설치, 유지보수자 모두 이해관계자

이해 관계자 --관심사항--> 시스템

(모호성 존재)

주어진 데이터를 나누어 원하는 목적을 찾아내는 행위(목적을 이뤄내는 것)
여기서 주어진 데이터가 concerns / 나눠 / 목적 : 모호성을 없애버리는 것이 목적

말은 하지만 모호한게 있다.

관심 사항과 요구사항은 다르다. 많이 착각, 모호한 문구가 너무 많다.

모호성이 있으면 관심 사항, 모호성이 없으면 요구사항

예) 불편해.. 편리하게 해줘

03. 요구분석 절차와 요구사항 종류

- 요구분석 절차와 요구사항 분류
  - 요구 분석 절차

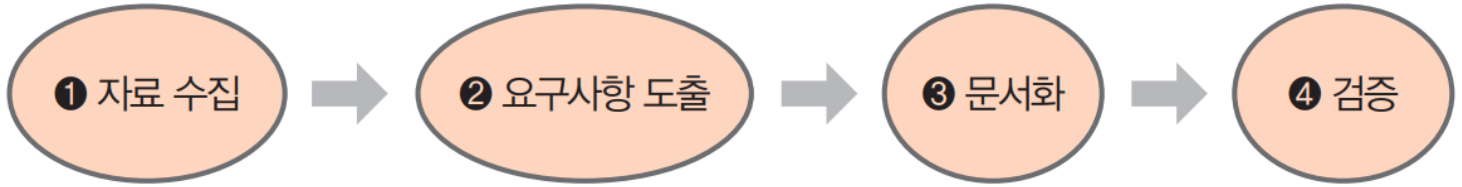

그림 4-8 요구분석 절차

❶ 자료 수집
현행 시스템의 문제점 도출, 실무 담당자 인터뷰, 현재 사용하는 문서 검토 등을 통해 가능한 모든 자료를 수집

❷ 요구사항 도출
수집한 자료를 정리해 적절히 분류하고 개발에 반영할 요구사항을 도출

❸ 문서화
도출한 요구사항을 요구분석명세서로 작성

❹ 검증
요구분석명세서에 사용자 요구가 정확히 기록되어 서로 모순되는 사항은 없는지, 빠뜨리지 않고 전부 기록했는지 등을 점검

03. 요구분석 절차와 요구사항 종류

- 요구분석 절차와 요구사항 분류
  - 요구 분석 분류
    - 사용자 요구사항은 개발될 소프트웨어가 제공할 기능 요구사항과 성능, 제약 사항, 품질과 같은 비기능 요구사항으로 분류

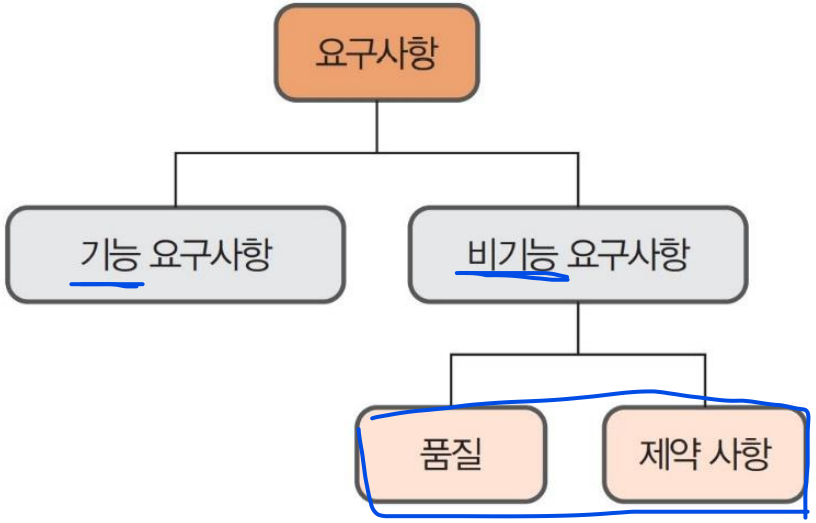

그림 4-9 요구사항 분류

03. 요구분석 절차와 요구사항 종류

- **기능 요구사항**
  - 사용자가 원하는 기능
  - 사용자는 시스템을 통해 기능을 제공받기 바라며 시스템은 사용자에게 필요한 기능을 제공
  - 사용자가 원하는 기능은 요구분석명세서에 완전하고 일관성 있게 표현해야 하며 시스템에도 전부 반영
  - 완전성: 사용자가 원하는 모든 기능이 포함 / 일관성: 요구사항 간에 모순이 있어서는 안 됨

영어로 function, 수학에서는 함수, 본질은 똑같다.
함수의 정의 : input 대비 output 값의 변화다. y=f(x)

인풋이 들어가서 아웃풋이 나오는 것, 그 사이에서 변화가 뭔지를 기술하면 됩니다.
쉽지는 않다.

03. 요구분석 절차와 요구사항 종류

품질(품질의 요구사항이 분석 대상), 제약사항(분석대상이 아니다.)
제약사항은 언제까지 해. 비용이 얼마야? 이런건데 분석대상이 아니다.
명확하기 때문에 모호성이 없다.

- 비기능 요구사항
  - 비기능 요구사항의 개요
    - 수행 가능한 환경, 품질, 제약 사항 등을 말함
    - 비기능 요구사항이 매우 중요한 소프트웨어에는 군사 무기나 의료 장비 등이 해당

품질 : 시스템이 어느정도 맞춰줘야하는 정도
안전성, 신뢰성, 성능, 유지보수, 사용성, 보안 => 대표적인 소프트웨어 품질
품질을 높이기 위해서 추가적으로 도입되는 기능 => 기능성
이것도 기능, 즉 품질에 정의된 것을 높이기 위해서도 기능적으로 해결한다
가끔 안되면 품질(안정성)이 떨어지는 것(n번 테스트)

- 제약 사항
  - 개발 소프트웨어가 수행될 환경에 의한 조건
  - 예시
    - 자바 언어를 사용해 개발하고, CBD 개발 방법론을 적용해야 함
    - 레드햇 리눅스 엔터프라이즈 버전에서 실행해야 함
    - 웹 로직 서버를 미들웨어로 사용해야 함
    - 윈도 운영체제와 리눅스 운영체제에서 모두 실행할 수 있어야 함

퀴즈/과제

- 배포 자료 참고

참고문헌

- 오픈 소스 소프트웨어로 실습하는 소프트웨어 공학, 정인상 교수님, 생능출판사
- 쉽게 배우는 소프트웨어 공학, 김치수, 한빛아카데미
- 소프트웨어 공학 이론과 실제, 홍장의, 한빛아카데미
- 소프트웨어공학의 모든 것, 최은만, 생능출판사
- 소프트웨어 공학, 조민호, 인피니트북스

T h a n k y o u

TECHNOLOGY

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Velit ex
ullicabo ipsum, labore sed tempora ratione asperiores des
cipserat bore sed tempora rati jgert one bore sed tem!