



6강. 사용자 요구 분석

컴퓨터과학과 김희천 교수



목차

- ① 요구사항
- ② 요구 공학 프로세스
- ③ 요구사항 모델링





Chapter. 1

요구사항

1. 요구사항

- + 문제 해결이나 목적 달성을 위해 사용자가 필요로 하는 조건이나 능력
 - × 시스템이 갖추고 있고 지켜야 하는 조건이나 능력
 - × 'what'에 관한 것이며 'how'에 관한 것이 아님
- + “요구 분석과 명세” 단계는 요구사항을 찾아 분석하고 문서화하는 과정
 - × 요구 공학이라고도 함

요구사항 명세서

- 시스템이 제공해야 하는 서비스나 제약 조건에 관해 기술한 문서
- 개발과 유지보수 작업의 기초가 되는 문서
- 고객과 개발자 간의 의사소통 수단으로 계약서

2. 요구사항의 종류

기능적 요구사항

- 시스템이 제공해야 하는 기능에 관한 요구사항
- 입출력 양식, 저장 구조, 계산 능력과 같은 기능적 요인을 정의
- 예)
"사용자가 티켓을 구입할 수 있어야 한다"
"사용자가 지방세 정보를 볼 수 있어야 한다"
"제품이 판매된 후 데이터베이스가 수정되어야 한다"

비기능적 요구사항

- 제품의 품질, 서비스나 기능상의 제약, 법률이나 표준의 준수에 관한 내용을 기술한 것
- 사용성, 효율성, 성능, 저장소 용량, 입출력 장치의 성능, 보안, 신뢰도, 이식성 관련 요구사항 등과 프로세스 관련 요구사항 및 하드웨어 사용에 관한 것 등
- 예)
"사용자에게 1초 내로 피드백이 주어져야 한다"
"인터페이스 색상이 회사의 공식 색상과 일관성이 있어야 한다"

3. FURPS+ (1/2)

- + HP에서 개발한 요구사항 분류 모델
- + Functionality는 기능적 요구사항
- + Usability Reliability Performance Supportability와 +는 비기능적 요구사항
 - × 사용성, 신뢰성, 성능, 지원성을 의미하는 품질 요구사항
 - × +는 설계, 구현, 인터페이스 및 물리적 제약 사항에 관한 것
- + 일반적으로 비기능적 요구사항이 기능적 요구사항보다 중요함
 - × 비기능적 요구사항이 만족되지 않으면 시스템을 사용하지 않을 수 있음

3. FURPS+ (2/2)

+ FURPS

×기능성

×사용성

×신뢰성

×성능

×지원성

기능적 요구사항

+ +

×설계 제약사항

×구현 제약사항

×인터페이스 제약사항

×물리적 제약사항

비기능적 요구사항



Chapter. 2

요구 공학 프로세스

1. 요구 공학

요구 공학 프로세스

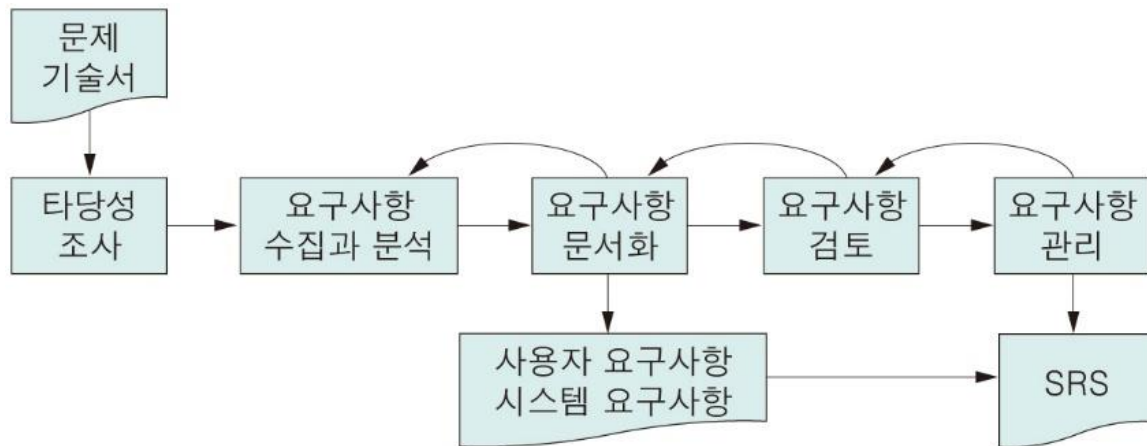


- 시스템의 목표와 기능 및 제약사항을 결정하는 과정
- 반복적이고 협력적인 프로세스를 강조하고
결과의 문서화와 변화하는 요구사항에 대한 이해를 강조한 용어

+ 입력은 고객이 준비한 문제 기술서이며
출력은 요구사항 명세서(SRS)

2. 타당성 조사

- 시스템이 기관의 목적에 부합하는지, 현재의 기술과 비용으로 일정에 맞춰 개발될 수 있는지, 다른 시스템과 통합될 수 있는지 판단



3. 요구사항 수집과 분석 (1/3)

✦ 요구사항 수집(또는 요구사항 추출)

- 고객이나 사용자와 의사소통하여 시스템이 제공해야 하는 서비스, 요구되는 시스템의 성능 및 제약사항 등을 찾아내는 것
- **분석가**가 도메인에 관한 지식을 가지고 있어야 함
- 다양한 사람이 관련되어 있음. 시스템에 관심을 가지고 있으며 시스템에 직간접적으로 영향을 받는 사람을 이해관계자(stakeholder)라 함

✧ 요구사항 분석

- 요구사항 추출 후의 작업
- 요구사항의 분류 → 요구사항들 간의 충돌 해결 → 우선 순위 매기기

3. 요구사항 수집과 분석 (2/3)

✦ 요구사항 추출과 분석의 어려운 점

- 정확히 표현하지 못하거나 비현실적인 요구를 함
- 각자의 환경에 익숙한 표현을 사용함
- 다양한 관점에서 서로 다른 요구를 하기도 함
- 주변 상황이 바뀌면 요구사항 분석 중에 요구사항이 바뀔 수 있음

JAD(Joint Application Design)

- 애플리케이션의 설계와 개발 과정에 고객과 사용자를 참여시키는 방법
- 모든 관련자들이 참여하는 JAD 세션(워크숍)에서 요구사항을 추출함
- 3~5일 정도의 회의를 진행함
- 최종 JAD 문서는 합의된 요구사항 명세서임

3. 요구사항 수집과 분석 (3/3)

+ JAD 절차

- 조사 단계: JAD 진행자가 인터뷰를 통해 프로젝트의 목적과 범위를 정함
- 준비 단계: 고객, 관리자, 시스템 분석가, 응용 분야 전문가 등으로 구성되는 JAD 팀을 정함
- 회의: 회의를 통해 시나리오, 유스케이스, 사용자 인터페이스를 정의
- 최종 문서 작업: 회의 중에 합의된 내용을 명세화하여 최종 문서를 만들고, 배포하여 검토함

× JAD 고려 사항

- 협력 작업을 통해 참석자들 간의 의사소통을 증진시키고 빠른 합의를 이끄는 방법
- 개발자는 사용자를, 사용자는 개발 작업의 문제를 이해하게 됨
- JAD 진행자의 역량이 중요

4. 요구사항 문서화

02

01

사용자 요구사항

- + 고객이나 사용자를 위해 작성됨
- + 자연어, 간단한 테이블이나 다이어그램을 사용
- + **고수준의 추상적 요구사항**으로 기술적 전문 용어의 사용을 피할 것
- + 제안 요청서(RFP)의 형태가 어려함

시스템 요구사항

- + **개발자를 대상으로 작성되는 것**
- + 서비스와 제약조건을 상세히 기술한 것
- + 설계와 개발 작업의 기초가 되며 계약서의 역할
- + 시스템 모델을 사용하여 표현함

5. 요구사항 검토 (1/2)

- + 개발 팀이 요구사항의 의미를 설명하고
고객, 계약자, 사용자 등이 오류가 있는지 시스템 요구사항을 검토하는 것
- + 요구사항 명세서는 계약서로의 기능을 하므로 고객과 개발자는 모두 상세히 검토해야 함

5. 요구사항 검토 (2/2)

+ 좋은 요구사항이 갖추어야 할 특성

완전성 모든 가능한 시나리오를 기술

일관성 요구사항들 간에 모순이 없어야 함

명확성 요구사항 표현에서 모호한 부분이 없어야 함

정확성 고객이 필요로 하고 개발자가 만들어야 하는 것을 표현해야 함

실행성 제약 조건을 만족하면서 시스템을 구현할 수 있어야 함

검증가능성 언급된 요구사항이 만족되는지 시스템을 테스트할 수 있어야 함

이해성 시스템의 구입자나 최종 사용자가 이해할 수 있어야 함

추적성 제안자, 요구사항, 설계 문서, 코드 사이의 관계를 추적할 수 있어야 함

적응성 다른 요구사항을 대규모로 수정하지 않고 변경될 수 있어야 함

6. 요구사항 관리 (1/3)

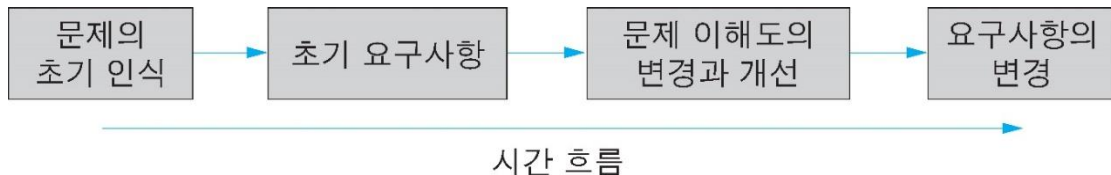
요구사항 관리



- 요구사항의 변화를 이해하고 통제하는 프로세스
- 요구사항을 조직화하고 문서화하고, **요구사항 변경을 관리**

✦ 초기 요구사항은 불완전하고 불일치가 존재하므로 요구사항은 변경되거나 진화함

– 환경의 변화, 시스템에 대한 이해도 상승, 새로운 요구 등으로 인해 요구사항이 변경됨



6. 요구사항 관리 (2/3)

- + 요구사항이 변경될 때 영향을 받는 모든 정보를 추적하여 함께 수정해 주어야 함

×추적성 정보

- 요구사항의 변경을 관리하기 위한 정보
- 요구사항들 간에, 그리고 요구사항, 시스템 설계, 코드 사이에는 다양한 의존 관계가 존재함
- 요구사항이 변경될 경우, 그 것에 영향을 받는 관련 요구사항(또는 설계 요소)을 파악할 수 있어야 함

6. 요구사항 관리 (3/3)

+ 추적성 정보의 활용

- 개발자는 모든 요구사항을 개발했는지 확인할 수 있음
- 테스터는 시스템이 요구사항과 일치하는지 알 수 있음
- 유지보수자는 변경의 파급 효과를 평가할 수 있음

× 추적성 정보의 유형

- 소스 추적성: 요구사항들을 그것의 제안자 및 제안 이유와 연결하는 것
- 요구사항 추적성: 특정 요구사항과 관련 있는 다른 요구사항들에 관한 정보를 유지하는 것
- 설계 추적성: 요구사항과 관련된 설계 문서를 파악하고 유지하는 것



Chapter. 3

요구사항 모델링

1. 요구사항 모델링

- + 시스템을 명확히 이해하거나 명세화하기 위해 모델을 사용함
 - × 사용자 요구사항은 자연어로 작성할 수 있음
- + 자연언어를 이용한 명세의 문제점
 - × 모호성, 요구사항의 혼합, 과도한 유연성, 모듈화의 어려움
- + 상세한 시스템 요구사항의 명세화 작업을 위해서 수학적 명세, 다이어그램이나 테이블, 구조화된 언어, 설계 기술 언어를 사용함

2. 시스템 모델

+ 가장 널리 사용되는 것은 **시스템 모델**을 이용하여 표현하는 것

- × 업무 프로세스, 해결할 문제, 시스템 자체를 그래픽을 사용해 표현
- × 다양한 관점에서 시스템의 모습을 표현
- × 정확한 기술 방법으로 표현하므로 검증이 가능함
- × 그림은 자연언어보다 이해하기 쉬움
- × 분석과 설계 과정의 다리 역할을 함

3. 객체지향 분석 (1/2)

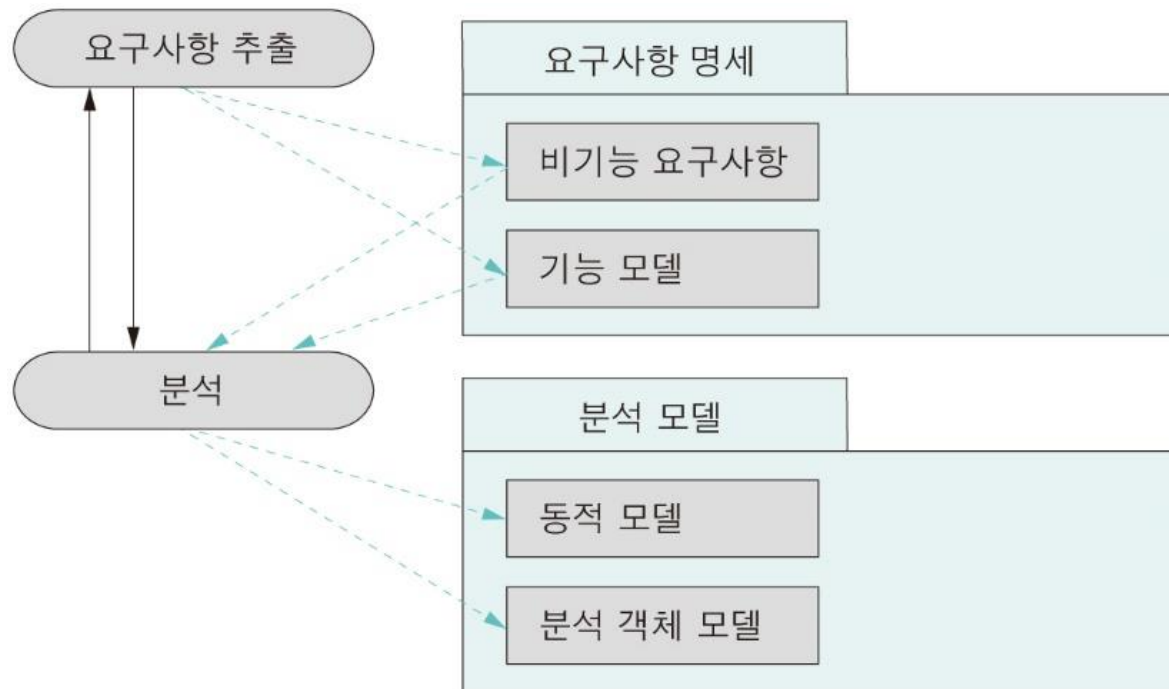
✦ 요구사항 분석 작업에 객체지향 방법을 적용하는 것

- 객체를 찾고 객체 간의 관계를 바탕으로 요구사항을 구조화하고 정형화하여 시스템의 모델을 만드는 일
- **요구사항의 정형화 과정**으로 어려운 일
- 도메인 지식과 기술적 지식의 부족 및 의견 불일치로 인해 미뤄지는 경향이 있음

✦ 분석 모델

- **객체지향 분석의 결과물**로 시스템을 사용자 관점에서 표현
- 기능 모델(유스케이스와 시나리오)은 분석 과정의 입력물이자 결과물
- 결과물은 분석 객체 모델(클래스 다이어그램)과 동적 모델(상태 다이어그램, 시퀀스 다이어그램)

3. 객체지향 분석 (2/2)



4. 구조적 분석 (1/2)

✦ 시스템 요구사항을 정의하기 위해 분할 정복과 하향식 기능 분해 방법을 사용한 요구사항 분석 방법

- 결과를 그래픽 표현에 기초한 시스템 모델로 표현
- 시스템 모델을 사용하여 분석가와 사용자가 의사 소통함
- 기능적 분해에 의해 시스템을 다루기 쉬운 조각들로 나눔

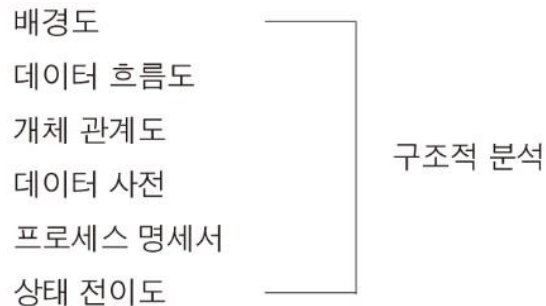
✦ 구조적 분석에 사용된 소프트웨어공학 원리

- 추상화의 원리: 복잡한 사물로부터 핵심적인 부분만을 간추려 내는 것
- 형식화의 원리: 엄격한 접근방법을 사용하고 단계별로 결과물을 문서화하는 것
- 분할과 정복: 큰 문제를 작고 독립적인 문제들로 나누어 푸는 것
- 계층화: 분할된 모듈을 트리 구조로 구성하는 것

4. 구조적 분석 (2/2)

+ 구조적 분석에서 사용되는 모델

- × 배경도, 데이터 흐름도, 개체 관계도, 데이터 사전, 프로세스 명세서, 상태 전이도



+ 구조적 설계에서 사용되는 모델

- × 구조도, 모듈 명세서, 데이터 사전



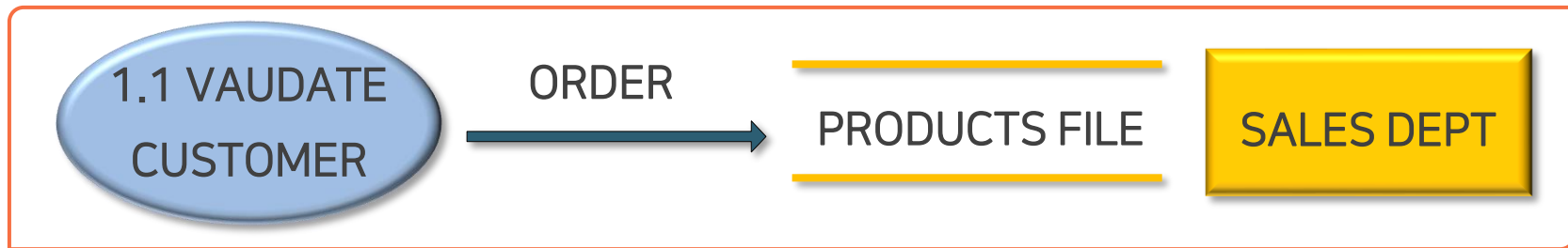
5. 데이터 흐름도 (1/3)

DFD

- + 구조적 분석에서 사용되는 기능 관점의 시스템 모델
- + 시스템에서 데이터 흐름과 변환을 보여주는 네트워크 형태의 다이어그램
- + 입력 데이터에 대하여 어떤 계산을 통해서 결과가 나오는지에 관한 단계적 처리를 보여줌
- + 정보 처리 시스템의 기능을 명세할 때 널리 사용됨
- + 데이터 저장소에 저장되거나 프로세스 사이에서 전달되는 데이터는 데이터 사전에 정의됨

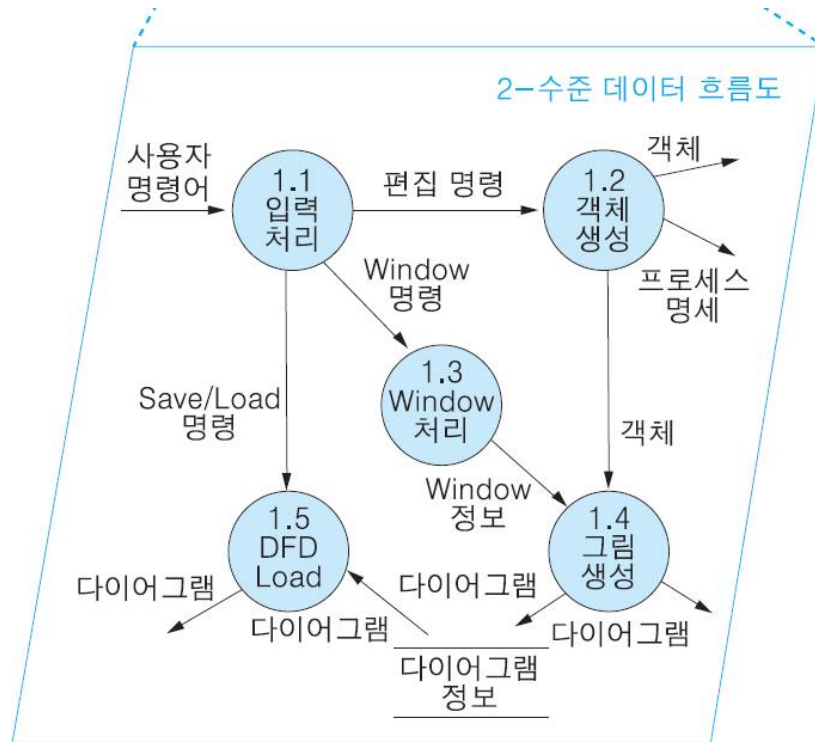
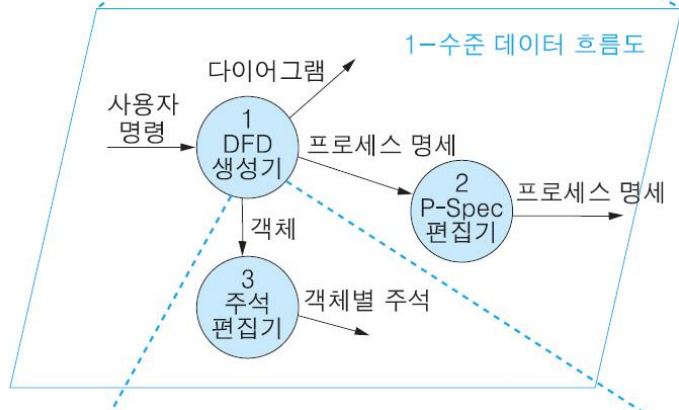
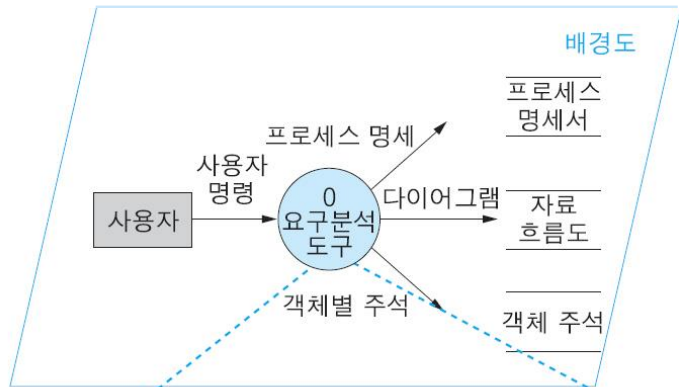
5. 데이터 흐름도 (2/3)

+ 표기법



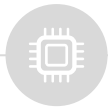
- × 프로세스는 데이터의 변환 작업을 의미함
- × 최상위 수준의 데이터 흐름도(배경도 또는 0-수준 DFD)에서 시작하여 확장됨
- × 최하위 수준의 데이터 흐름도에 존재하는 프로세스를 기능 단위라 함
- × 최하위 프로세스에 대해 프로세스 명세서를 작성
- × 프로세스 명세서는 의사 코드 등을 사용하여 프로그램의 논리를 보여줌

5. 데이터 흐름도 (3/3)



6. 구조적 분석과 객체지향 분석의 비교

- + 구조적 분석은 기능 관점에서 시스템을 표현함
- + 객체지향 분석은 기능과 데이터를 함께 캡슐화한 객체를 가지고 시스템을 표현함
- + 구조적 분석은 기능 분해 또는 합성에 의한 표현 방법을 사용하며, 객체지향 분석은 상속 개념도 사용할 수 있음



다음강의

7강. 소프트웨어 설계

