

요구사항 개발 및 관리

황두현

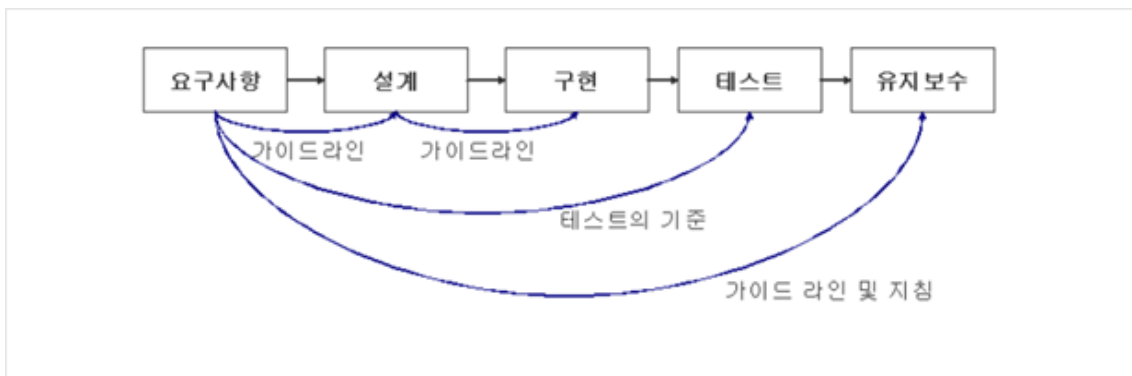
1. 요구사항

A. 의미

- i. 문제의 해결이나 목적을 달성하기 위해 고객이 요구하거나, 어떤 조건을 만족하기 위해 가져야하는 서비스

B. 중요성

- i. 개발되는 제품을 전체적으로 파악하게 하여 이를 기반으로 계획을 세워 의사소통 시간을 절약하게 해준다.



2. 요구사항 분류

A. 기능적 요구사항

- i. 수행될 기능과 관련된 입력과 출력 및 이들 사이의 처리 과정
- ii. 제품의 구현을 위해 소프트웨어가 가져야하는 기능적 속성

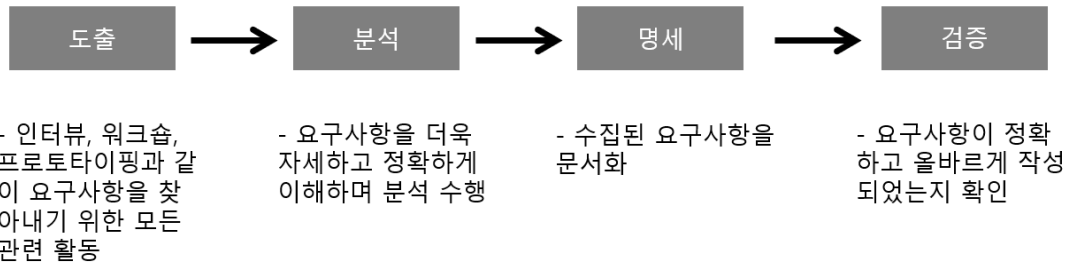
B. 비기능적 요구사항

- i. 품질 기준 등을 만족시키기 위해 소프트웨어가 가져야 하는 성능.
- ii. 사용의 용이성, 안전성과 같은 행위적 특성
- iii. 시스템의 기능에 관련되지 않은 사항

3. 요구사항 개발

A. 의미

- i. 고객으로부터 구현될 제품의 사양을 정확히 도출하여 요구사항을 구체화하고, 이를 분석한 결과를 개발자들이 이해할 수 있는 형식으로 기술하는 작업



4. 요구사항 추출

A. 의미

- i. 고객의 요구사항을 수집하고 개발하는 시스템에 대한 요구와 기능 및 제약사항을 식별하고 이해하는 단계

B. 추출 기법의 종류

- i. 인터뷰
 - ① 프로젝트의 참여자들과 직접적인 대화를 통해 정보를 추출하는 기법
- ii. 시나리오
 - ① 시스템과 사용자간에 상호 작용을 시나리오로 작성하여 추출하는 기법

5. 요구사항 분석

A. 의미

- i. 추출된 요구사항을 분석 기법을 이용해 문제점들을 도출하고 요구사항을 이해하는 과정
- ii. 추상적 요구사항들을 명세서 작성 전에 일관성 있게 정리하는 활동

6. 요구사항 분석 기법의 종류

A. 구조적 분석

- i. 시스템의 기능을 중심으로 구조적 분석을 진행
- ii. 시스템의 기능을 정의하기 위해서 프로세스들을 도출하고, 도출된 프로세스 간의 데이터 흐름을 정리

B. 객체지향 분석

- i. 요구사항을 사용자 중심의 시나리오 분석을 통해 유스케이스 모델로 구축하는 것
- ii. 요구사항을 추출하고, 유스케이스의 실체화 과정을 통해 추출된 요구사항을 분석

7. 요구사항 명세

A. 의미

- i. 분석된 요구사항을 명확하게 기록하는 것
- ii. 시스템이 수행해야하는 모든 기능과, 구현 상의 제약 조건 및 성능에 관한 사항 등을 작성하는 것

항목	설명
외부 인터페이스 요구사항 (External Interface Requirement)	<p>모든 소프트웨어 시스템으로의 입력과 출력에 대한 요구사항을 상세히 기술한다. 사용자 인터페이스, 하드웨어 인터페이스, 소프트웨어 인터페이스, 통신 인터페이스 등으로 분류될 수 있으며, 다음과 같은 형식의 내용이 포함되어야 한다.</p> <ul style="list-style-type: none"> - 항목의 이름 (Name of item) - 목적 상세 설명 (Description of purpose) - 입력의 시작점 및 출력의 도착점 (Source of input or destination of output) - 유효 범위, 정확도, 오차 (Valid range, accuracy, and/or tolerance) - 측정 단위 (Units of measure) - 시간 (Timing) - 다른 입력 및 출력과의 관계 (Relationships to other inputs/outputs) - Screen formats/organization; - Window formats/organization; - 데이터 형식 (Data formats) - 명령 형식 (Command formats) - 종료 메시지 (End message)
기능 요구사항 (Functional Requirement)	<p>소프트웨어의 입력 처리와 출력을 생성하는 처리 과정에서 발생할 수 있는 기본적인 동작에 대하여 기술하며, '시스템은 ~해야 한다' 형식으로 작성되어야 하며 아래와 같은 항목들이 포함된다.</p> <ul style="list-style-type: none"> - 입력의 유효성 확인 (Validity checks on the inputs) - 동작의 정확한 흐름 (Exact sequence of operations) - 비정상 상황에 대한 동작 (Responses to abnormal situations, including, Overflow, Communication facilities, Error handling and recovery) - 파라미터의 영향 (Effect of parameters) - 출력에서 입력까지의 관계 (Relationship of outputs to inputs, including Input/output sequences, Formulas for input to output conversion)
성능 요구사항 (Performance Requirement)	<p>소프트웨어 전체적으로 사람과의 상호작용 혹은 소프트웨어에서 확인할 수 있는 정적이고 동적인 수치적 요구사항을 기술한다.</p> <ul style="list-style-type: none"> - 정적인 수치적 요구사항(static numerical requirements): 소프트웨어에서 다루어지고 있는 정보의 양이나 라임과 같은 고정적인 수치적 요구사항 - 동적인 수치적 요구사항(dynamic numerical requirements): 정해진 시간이 아니라 일정한 시간 안에 수행해야 하는 것들을 정해진 수치가 없는 요구사항
논리적 데이터베이스 요구사항 (Logical Database Requirement)	<p>데이터 베이스에서 사용될 정보를 위한 논리적 요구사항에 대하여 기술하며, 다음과 같은 항목을 포함한다.</p> <ul style="list-style-type: none"> - 여러 기능에서 사용될 정보의 타입 (Types of information used by various functions) - 사용 빈도 (Frequency of use) - 접근 가능성 (Accessing capabilities) - 데이터 엔티티 및 관계 (Data entities and their relationships) - 무결성 제약조건 (Integrity constraints) - 데이터 유지 요구사항 (Data retention requirements)
설계 제약사항 (Design Constraints)	<p>다른 표준이나 하드웨어적 제한으로 인해 적용되는 설계적 제한사항에 대하여 기술한다.</p>
소프트웨어 시스템 속성 (Software System Attribute)	<p>신뢰도(Reliability), 사용가능성(Availability), 보안(Security), 유지보수(Maintainability), 이식성(Portability) 등의 소프트웨어 속성을 요구사항으로서 기술한다.</p>

8. 요구사항 검증

A. 의미

- i. 요구사항이 명세서에 올바르게 기술되었는지 검토하는 활동
- ii. 구현되는 시스템이 사용자의 요구와 목표를 만족하는가에 대해 검토하는 활동

B. 검증 내용

- i. 요구사항이 시스템 사용자의 실제 요구를 반영했는가
- ii. 요구사항 명세가 문서 표준을 따르고 설계 단계의 기초로 적합한가
- iii. 문서 상의 요구 사항이 서로 상충되지 않았는가
- iv. 의도한 모든 기능과 제약이 정의되었고, 요구사항이 문서에 모두 포함되었는가
- v. 실제로 구현 가능한가
- vi. 시스템 요구사항과 시스템 설계 문서를 추적할 수 있는가

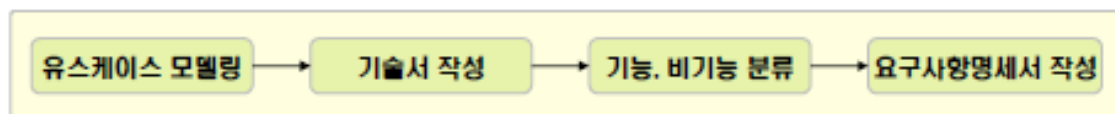
9. 요구사항 분석

A. 의미

- i. 요구사항 명세서 작성의 기반을 다지는 작업

B. 분석 방법

- i. 객체지향 방법인 유스케이스 기반 분석



10. 유스케이스

A. 의미

- i. 시스템이 액터에게 제공해야하는 기능의 집합
- ii. 시스템의 요구사항을 보여줌

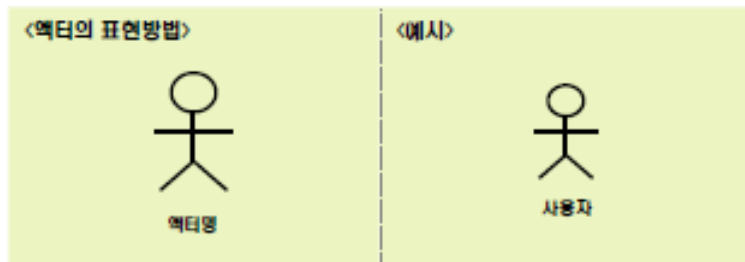
B. 표기법

- i. 타원형으로 표시하고 그 안쪽이나 아래쪽에 유스케이스 명을 기술
- ii. 이름은 "~한다" 와 같이 동사로 표현
- iii. 각 유스케이스가 개발될 기능 하나와 연결될 수 있도록 함

11. 액터

A. 의미

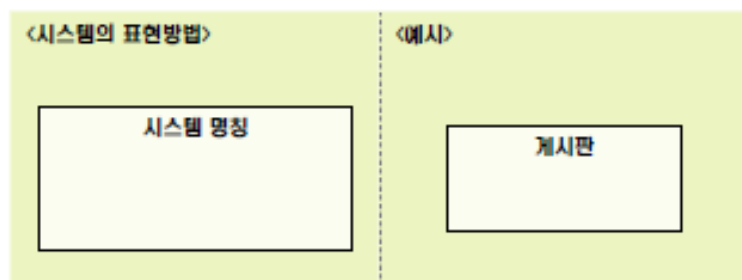
- i. 시스템의 외부에 있으면서 시스템과 상호 작용을 하는 사람 또는 다른 시스템
- ii. 표기법



12. 시스템

A. 의미

- i. 만들고자 하는 어플리케이션
- ii. 표기법



13.관계

A. 의미

- i. 액터와 유스케이스 사이의 의미 있는 관계

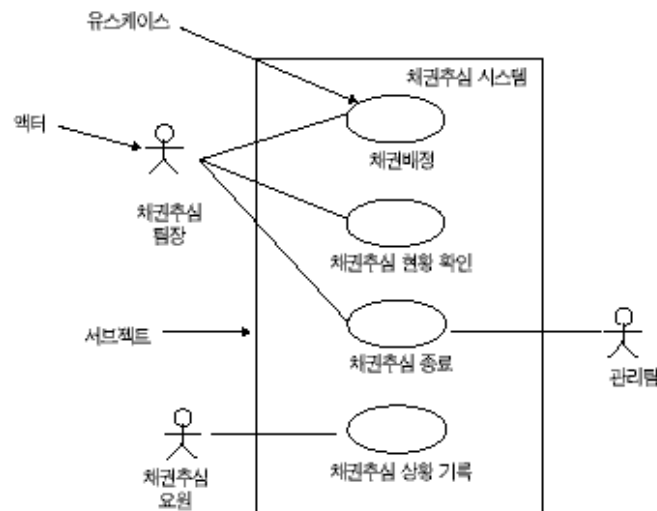
B. 종류

- i. 연관 관계
- ii. 의존 관계
- iii. 일반화 관계

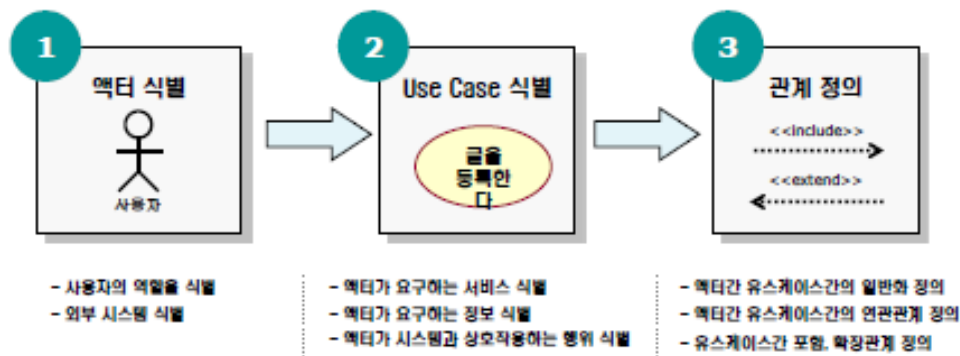
14.유스케이스 다이어그램

- A. 사용자의 관점에서 시스템의 서비스 혹은 기능 및 그와 관련한 외부 요소를 보여주는 다이어그램

- B. 고객과 개발자가 함께 보며 요구사항에 대한 의견을 조율할 수 있음



- C. 유스케이스 다이어그램 작성 순서



15. 완성된 유스케이스 다이어그램

