

**LOGO**

# Software Requirements

200412358 최상현

# Contents

- 1 THE CONTEXT OF SOFTWARE REQUIREMENTS
- 2 REQUIREMENTS ENGINEERING PROCESS
- 3 REQUIREMENTS ELICITATION
- 4 REQUIREMENTS ANALYSIS

# Contents

- 5 SOFTWARE REQUIREMENTS SPECIFICATION
- 6 REQUIREMENTS VALIDATION
- 7 REQUIREMENTS MANAGEMENT

# 1. THE CONTEXT OF SOFTWARE REQUIREMENTS

- ◆ 소프트웨어 요구사항 항상 사업 문제(business problem)로부터 나옴  
예) 여권 신청 절차, 자동차 안전 시스템 개발
- ◆ 이러한 문제를 해결 할 수 있는 방안들이 통합된 기술이 소프트웨어
- ◆ 고객이 요구한 것을 소프트웨어로 옮기는 작업은 매우 복잡하고 비용이 많이 듦  
EX) 오류 수정 비용
- ◆ 설계와 구현 방법에 따라 비용이 달라짐
- ◆ 따라서, 요구사항 공학 절차를 거침으로써 에러 발생을 최소한으로 줄여 비용을 줄일 수 있음

# 1. THE CONTEXT OF SOFTWARE REQUIREMENTS

## 1.1 Requirements and Constraints

### ◆ 요구사항(Requirements)

#### ▷ 정의

요구사항은 시스템에 의해서 사업 문제를 해결하기 위한 자산 또는 능력

#### ▷ 분류

- 기능적 요구사항 ( Functional Requirement )

: 소프트웨어가 수행 해야 하는 기능

- 비기능적 요구사항 ( Nonfunctional Requirement )

: 시스템의 특징을 묘사

# 1. THE CONTEXT OF SOFTWARE REQUIREMENTS

## 1.1 Requirements and Constraints

### ◆ 요구사항(Requirements)

#### ▷ Emergent properties

- 요구사항과 대조적  
ex) 분산 시스템의 성능

# 1. THE CONTEXT OF SOFTWARE REQUIREMENTS

## 1.1 Requirements and Constraints

### ◆ 제약사항(Constraints)

- ▷ 사업 문제를 해결하기 위한 방법의 범위를 제한

예1) 소음법률로 인한 도난경보 사이렌 울리는 소리의 길이 제한

예2) 메모리, 프로세스 속도, 대역폭 (기술적인 부분)

- ▷ 특정분야의 문제(problem domain)

예) 기차제어 시스템에서의 전자기파

## 2. REQUIREMENTS ENGINEERING PROCESS

- ◆ 요구사항 공학 프로세스(Requirements Engineering Process)는 문제 해결방안의 시스템 속성을 명세서(specification)로 변환 시켜야 함
- ◆ 분석을 통해 나온 요구사항들은 개발자와 유저간의 소통을 위해 문서화 해야 함.
- ◆ IEEE(전기전자기술자협회)에서 발표한 3가지 표준
  - IEEE Std 1362~1998  
Guide for Information Technology - system Definition - Concept of Operation (ConOps) Document [IEEE 98a]
  - IEEE Std 1233-1998  
Guide for Developing System Requirements Specifications [IEEE 98b]
  - IEEE Std 830-1998  
Recommended Practice for Software Requirements Specifications [IEEE 98c]

## 2. REQUIREMENTS ENGINEERING PROCESS

- ◆ 프로젝트 같이 복잡한 것은 요구사항 변경 때문에 개발과정에서 문제가 늦게 발생 할 수 있음
- ◆ 요구사항 공학은 이러한 요구사항 변경으로 인한 예기치 못한 위험을 최소화 하는 과정
- ◆ 즉, 비즈니스 문제를 해결하기 위해 요구사항을 발견하고, 이해하고, 기록하고, 검사하고, 소통하고, 관리

## 3. REQUIREMENTS ELICITATION

- ◆ 요구사항 추출은 요구사항을 발견하는 과정
- ◆ 요구사항 추출을 통해 프로젝트의 규모를 명확히 할 수 있음
- ◆ 요구사항을 추출하기 위해선 먼저 요구사항이 발생하는 근원에 대한 정보 수집이 필요함
- ◆ 그 정보로부터 요구사항을 종합하여 추출함
- ◆ 수집, 수정, 올바른 건지 아닌지, 재 공급 을 반복적으로 수행

## 3. REQUIREMENTS ELICITATION

### 3.1 Requirements Source

- ◆ 큰 시스템에서는 많고 다양한 투자자들 (stakeholders ) 이 있는데 이들이 요구사항의 첫 번째 근원(source)
- ◆ 투자자를 확인하는 것이 중요한 첫걸음
- ◆ 해당 분야 전문가는 요구사항 도출에 중요한 역할을 함.

## 3. REQUIREMENTS ELICITATION

### 3.2 Elicitation Techniques

- ◆ 다양한 투자자들이 있으면 요구사항이 일치 안될 때 있는데 이때 투자자의 배경 정보를 압으로써 해결 할 수 있음
- ◆ sociotechnical 시스템에서 사용자(user)의 이야기나 시나리오는 가치 있는 도구
- ◆ 시나리오는 요구사항 엔지니어에게 의문에 대한 대답을 제공

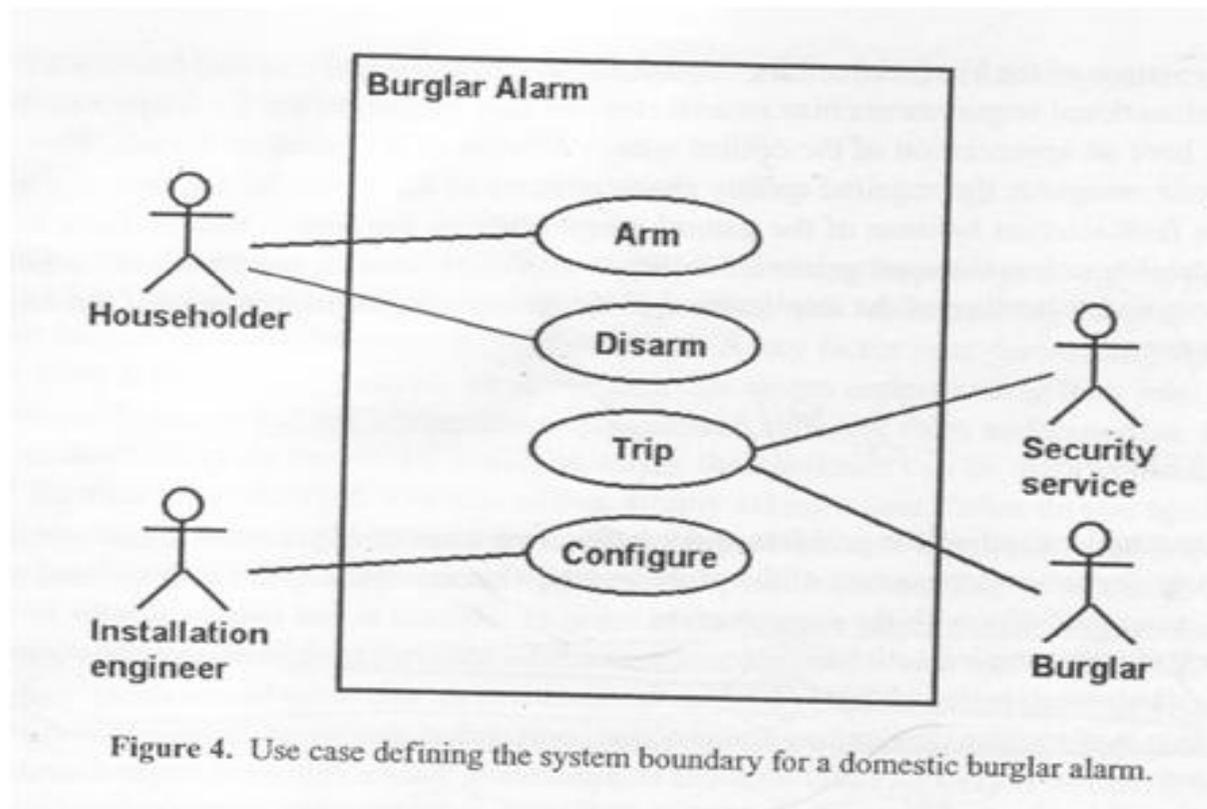
## 4. REQUIREMENTS ANALYSIS

- ◆ 요구사항 분석은 문제를 이해하고 열거된 요구사항들 중 최선의 해결책을 통합 하는것
- ◆ 분석 활동은 요구사항의 기준선을 산출하는 일
  - ▷ 엔지니어와 투자자는 요구사항 기준선을 맞추기 위해 무엇을 포함할지 무엇을 빼야 할지, 일치하지 않은 부분을 협상 해야 함
- ◆ 요구사항의 기준선은 필요하지만 이 기준선이 모든 요구사항을 포함하고 목표달성에 기여하는 것은 아님
- ◆ 하지만 문제를 어떻게 해결할 것인가를 명확히 하기 위해 필요함

## 4. REQUIREMENTS ANALYSIS

### 4.1 The System Boundary

- ◆ System Boundary는 proposed system의 모든 문제의 해결방안을 제공하고 환경에서 어떻게 시스템이 작동하는지도 포함 됨



## 4. REQUIREMENTS ANALYSIS

### 4.2 Requirements Modeling

- ◆ 엔지니어는 요구사항에 대한 적절한 해결책을 찾기 위해 문제를 모델로 구성
- ◆ 모델은 요구사항에 대해 개발자와 proposed system사이 의사소통에 도움을 줌
- ◆ 모델링 예 : UML

## 4. REQUIREMENTS ANALYSIS

### 4.3 Derived Requirements

- ◆ 요구사항을 추출하는 과정
- ◆ 분석과정에서 새롭게 추출된 기록으로부터 요구사항을 유도

### 4.4 Requirements Attributes

- ◆ 요구사항을 편리하게 관리하기 위한 것
- ◆ 종류 : Identifier, Source, Date, Rationale, Type, Priority, Stability, Verification Procedure, Status

### 4.5 Requirement Trade-offs

- ◆ 요구사항에 필요한 것은 포함 시키고 불필요한 것은 빼는 작업

## 5. SOFTWARE REQUIREMENTS SPECIFICATION

- ◆ 소프트웨어 요구사항의 문서 ( SRS :소프트웨어 요구사항 명세서 )
- ◆ 알기 쉽고 일관되게 작성해야하고 전체가 긴밀하게 연결되어야 함
- ◆ 시스템 모델과 연결된 요구사항 뿐만 아니라 전후 관계, 제약사항, 협약(약조), 가설과 같은 정보를 제공

## 6. REQUIREMENTS VALIDATION

- ◆ 요구사항은 사업 문제를 해결하기 위해 필요한 것이 무엇인지 정확히 반영되어야 함.
- ◆ 검증(Validation)은 항상 요구사항 명세서에 적용되며 이것이 최종 도안
- ◆ 한번 검증되면 그 명세서는 변경 가능
- ◆ 즉, 검증된 문서가 형식과 버전을 변경 해야만 할 때 다시 설계 또는 실행의 기초가 됨.

예외) 투자자의 동의가 필요할 경우, 프로세스 요구사항 변동 없을 경우

- ◆ 요구 수용했는가에 대해 입증 해야 함.
- ◆ 요구사항에 대한 가능한 해결방안을 계획하는데 너무 과도하게 어려운 것이 입증된다면 요구사항이 잘못임
- ◆ 요구사항이 유용해야만 적법함

## 7. REQUIREMENTS MANAGEMENT

- ◆ 요구사항 관리(management)는 4가지 업무를 포함
  - change control
  - version control
  - requirements tracing
  - status tracking
- ◆ 요구사항이 유일할 때 4가지 업무 가능
- ◆ 요구사항 관리는 중요한 업무이나 등한시 되기도 함
- ◆ 1990년대 초 SW-CMM(Capability Maturity Model for Software)에 의해 부각

# 7. REQUIREMENTS MANAGEMENT

## 7.1 Change Control

- ◆ 개발단계에서 요구사항은 언제라도 변화(change)이 발생 할 수 있음
- ◆ 허락되지 않은 변화는 문제를 야기 시킴  
예) 예산초과, 기한 초과
- ◆ 변경의 평가 기준 : 비용, 이로움, 평가위원단(assessment panel)에서  
변화 요청 확인
- ◆ 평가위원단은 변화가 프로젝트에 기간과 비용에 어떤 영향을 끼칠지 고려해야 함
- ◆ 진보적인 변화는 항상 우선순위를 정하고 개발 계획을 변화에 따라  
조절해야 함

# 7. REQUIREMENTS MANAGEMENT

## 7.2 Version Control

- ◆ 요구사항의 변화는 저장되어야 하는데 Version Control에서는 이 변화를 자세한 부분까지 포함
- ◆ 변화 할 때마다 새로운 요구사항 명세서가 나옴
- ◆ 문서 버전을 위해 번호를 붙이는 작업이 본질적인 일

## 7.3 Requirements Tracing

- ◆ Change Control 과 Status Tracking을 하기 위해서는 요구사항을 반드시 추적해야 함
- ◆ 개념적으로 Requirements Tracing은 그래프 형태는 acyclic 그래프 임

## 7.4 Status Tracking

- ◆ 프로세싱과 요구사항의 실행에 대한 정보를 유지하는 것

**LOGO**

**Thank You !**

[www.themegallery.com](http://www.themegallery.com)