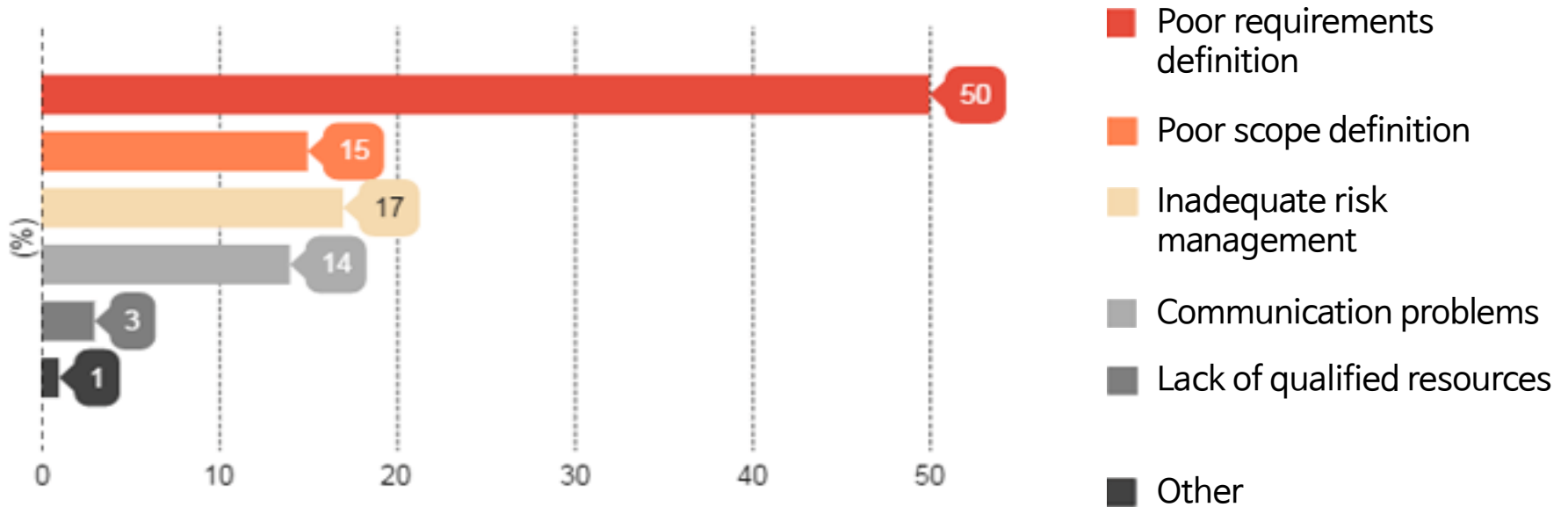




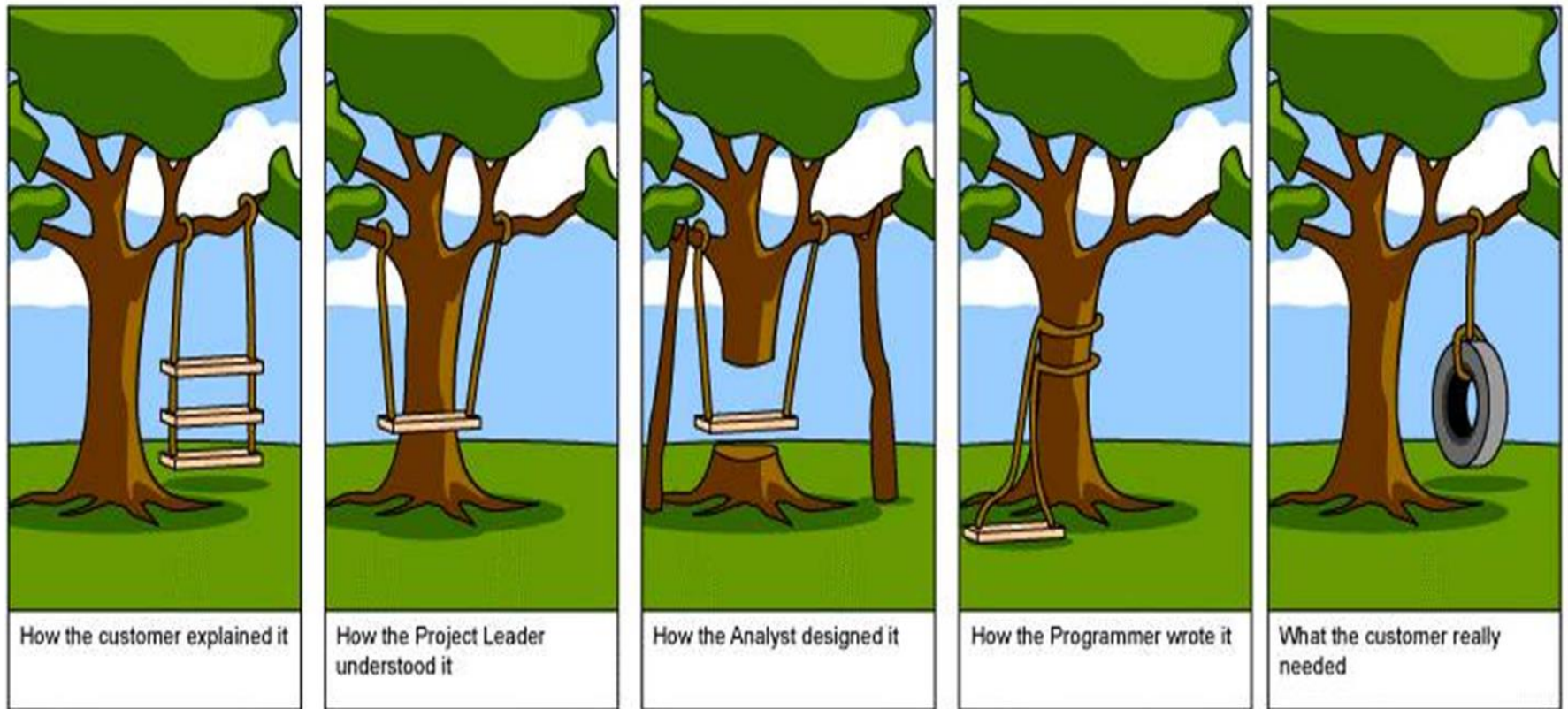
요구사항 개요

SW 프로젝트의 실패 원인



출처) <https://r-stylelab.com/company/blog/web-development/how-to-write-software-project-requirements>

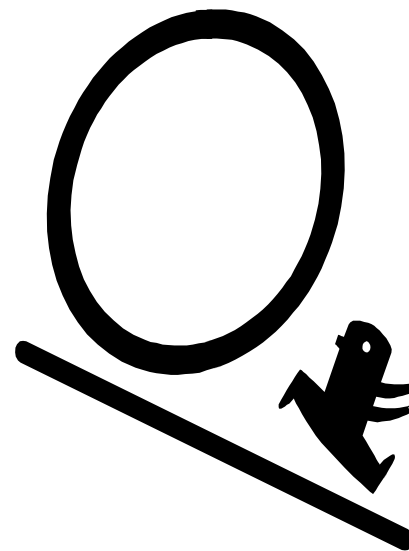
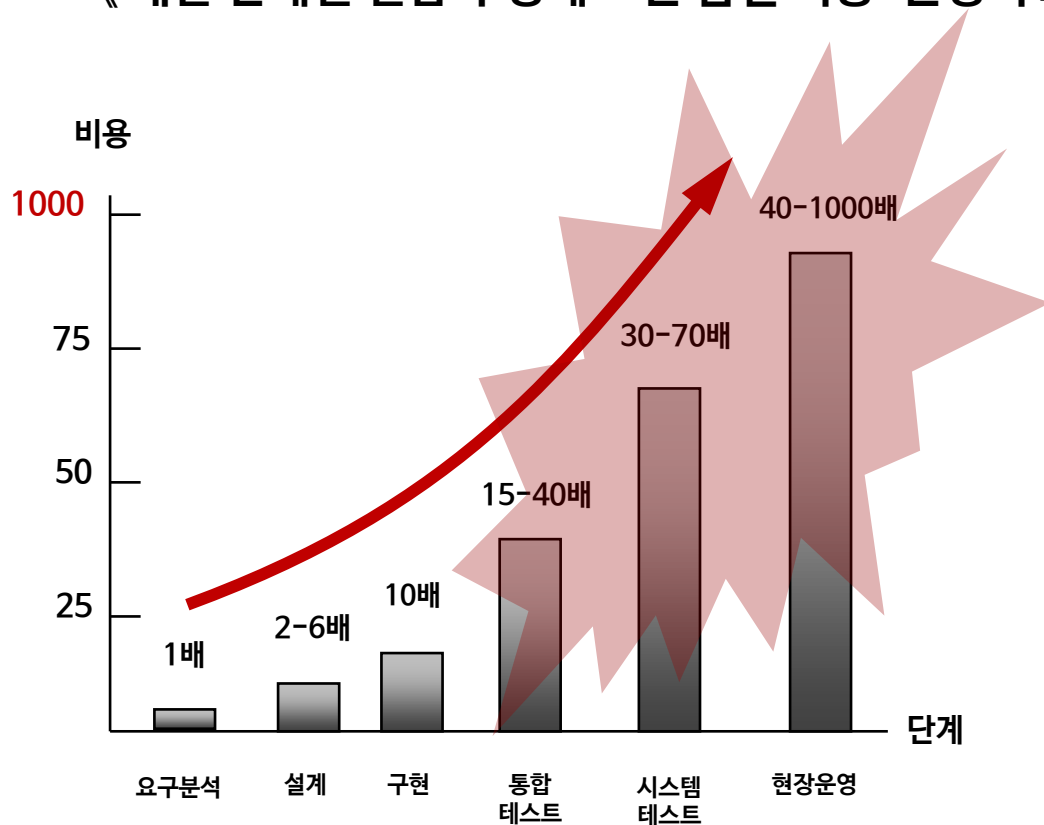
요구사항 정의의 어려움



요구사항의 중요성(1/3)

출시 이후에 오류를 수정하는 것은 개발 초기단계에 수정하는 것보다
40-1000배 이상의 비용 소요

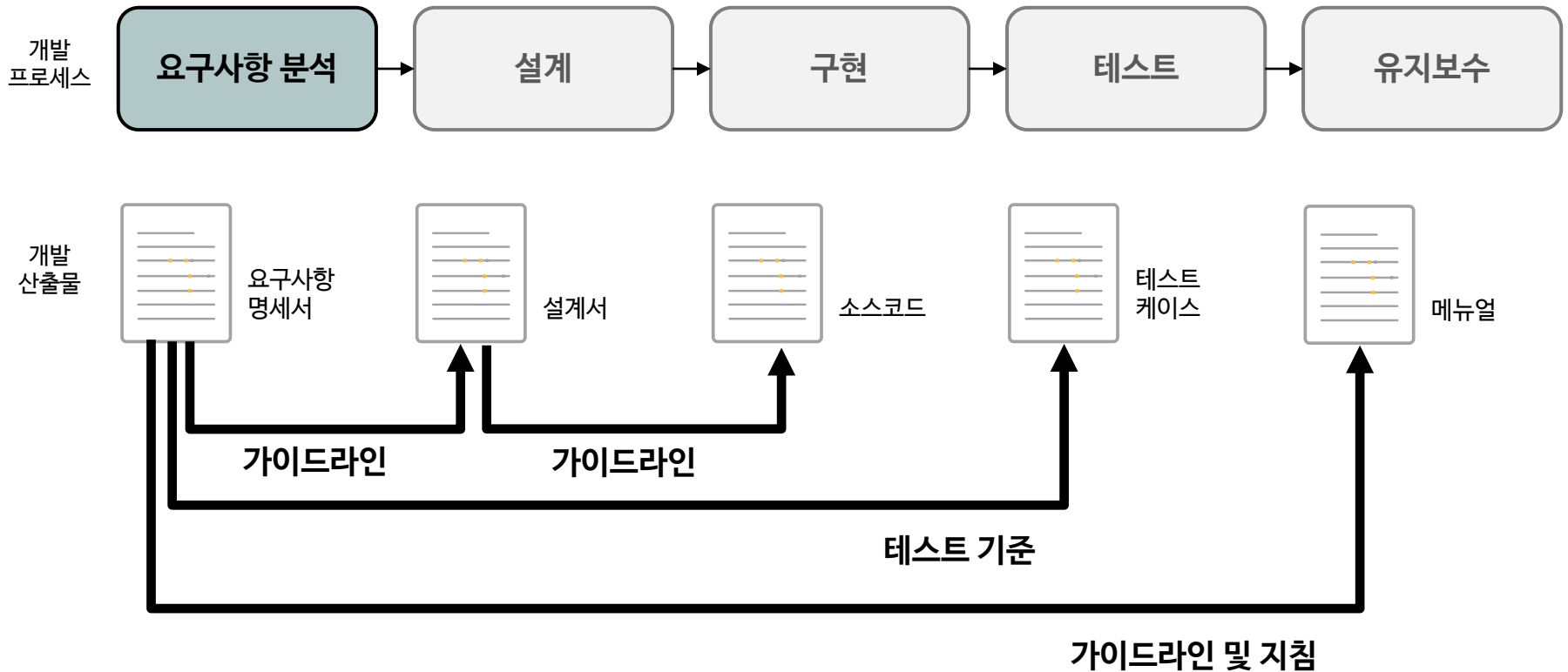
《 개발 단계별 결함 수정에 드는 품질 비용: 눈덩이 효과 》



출처) Kaplan, C., R. Clark, and V. Tang, *Secrets of Software Quality: 40 Innovations from IBM*, Mc-Graw-Hill, 1995

요구사항의 중요성(2/3)

요구사항은 소프트웨어 제품을 전체적으로 파악하도록 하며,
전체 개발 단계의 기준 및 가이드라인으로 활용됨



요구사항의 중요성(3/3)



요구사항

- 개발의 시작
- 프로젝트의 기준
- 개발의 진척도를 측정하는 기준
- 제품 개발 완료 승인의 기준
- 변경 발생의 근본 원인

요구사항은 가정해서는 안되며, 정확한 내용을 기술하여야 함

[참고] 요구사항의 실체

요구사항에 대한 기대치



- 완벽한 요구사항이 제공될 거야
- 시스템 운영 환경에 대한 완벽한 정보가 제공되겠지
- 요구사항을 검증할 완벽한 검증 기준을 제시해 줄 거야

요구사항의 실체

- ① **고객으로부터 전달되는 요구사항의 미흡함**
 - 위시 리스트 수준의 요구사항이 전달됨
 - 완벽하지 않으며, 이해하기 어렵고 애매모호하게 작성됨
- ② **개발이 진행되면서 기술적 내용이 구체화됨**
 - 요구사항도 진화하고 변경됨
- ③ **개발이 진행되면서 커뮤니케이션의 문제가 해소됨**
 - 요구사항이 변경되고, 명확해짐
- ④ **프로젝트 일정/예산 등을 고려하지 않음**
 - 기술적 난이도, 제약사항 등을 고려하지 않고 정의됨

요구사항(Requirement)이란? (1/3)

- 사용자의 Needs를 어떻게 해결(Solution)해 줄 것인지에 대한 명세
 - Needs란 그저 사용자가 원하는 것
 - 요구사항은 Needs를 어떻게 실현시켜줄 수 있느냐에 대한 구체적인 명세



사용자가 원하는 것(Needs)은?

원하는 것이 곧 요구사항인가?

요구사항(Requirement)은 무엇인가?

해결해야할 문제(Problem) 혹은 요구(Needs)에 대한 해결책(Solution)
→ 요구사항(Requirement)

[토론] Needs와 Requirement의 차이(1/2)



Problem



Solution



Needs



Requirement



요구사항 정의

현실 세계의 문제를 해결하기 위하여 **고객에 의해 요구되거나 표준** 등을 만족하기 위해
제품이 가져야 하는 서비스 또는 제약사항

- 요구사항은 사용자가 문제를 해결하거나 목표를 달성하는데 필요한 조건 또는 기능

[IEEE 729 Std. Glossary]

- 요구사항이란 무엇이 구현되어야 하는가에 대한 명세
- 요구사항은 시스템이 어떻게 동작하여야 하는지 또는 시스템 특징이나 속성들에 대한 설명
- 요구사항은 시스템 개발 프로세스 상의 제한 사항

[Sommerville & Sawyer 1997]

- 요구사항은 설계를 선택하게 하는 모든 것

[Brain Lawrence]

- 요구사항은 원하는 시스템의 외부에서 관찰 가능한 특성

[Alan Davis]

요구사항 종류

○ 기능 요구사항 (Functional Requirement)

- *What the software must do?*
- 제품이 목표를 달성하기 위해 사용자에게 제공해야 하는 **행위적(기능적)** 속성
- 시스템이나 소프트웨어가 **무엇(What)**을 해야 하는지에 대해 정의한 요구사항

○ 비기능 요구사항 (Non-Functional Requirement)

- *How well (much) the software does?*
- 제품의 기능이 **성능, 안전성, 사용성 등의 품질 기준**을 만족시키기 위해 가져야 하는 속성
- 품질 요구사항
 - 시스템이나 소프트웨어가 무엇(What)을 **얼마나(How much)** 해야 하는지에 대해 정의한 요구사항
- 제한 사항 (Limitations)
 - 시스템이나 소프트웨어의 개발에 반영되어야 하지만 우리가 변경할 수 없는 요구사항
- 제약 사항 (Constraints)
 - 시스템이나 소프트웨어의 개발에 고려해야 하고, 조율이 가능하지만 약속되어 있는 요구사항

[예시] 요구사항 종류

What the software must do?

Functional

Function

- Behavior Functions
- Interface Functions
- Control Functions
- Accuracy

How well (much) the software does?

Non-Functional

Quality

- Reliability
- Fault Tolerance
- Maintainability
- Serviceability
- Usability
- Reusability
- Security
- Adaptability

Safety

- Fault Detection
- Fault Control
- Fault Avoidance

Performance

- Response Time
- Workload Capacity

Limitation

External Environments

- Standard & Legal
- Cultural & Political

Internal Environments

- Company Vision
- Product Goal
- Resources (Cost, Schedule, People Efforts...)

기능 요구사항 (Functional Requirement)

- *What the software must do?*
- 제품이 목표를 달성하기 위해 사용자에게 제공해야 하는 **행위적(기능적)** 속성
- 시스템이나 소프트웨어가 **무엇(What)**을 해야 하는지에 대해 정의한 요구사항

Control Functions	What the system must do?
Interface Functions	How the system can perform those features in interaction with its environments?
Decomposed Functions	How the functions should be supported?
Operation Functions	How the system should be statically formed?
Behavior Functions	How the vehicle should be executed?

[참고] 자연어 기반으로 명세 시 고려사항

- 요구사항은 의도대로 명확한 해석이 가능할 수 있도록 기술되어야 함
 - Key Requirement를 선별하여, 표준 용어나 표준 요구사항 작성 기술 이용
 - Mandatory : “Must” or “Shall”
 - Interdiction : “Must not”
 - Desired : “Should”
 - Optional : “Can”
 - Performance : “Sufficient”, “Most”, “Close to”, “Fast”
- 하나의 요구사항은 하나의 객체로 간결, 명료하게 기술되어야 함
- 하나의 요구사항은 검증이 가능하게 정량화하여 기술되어야 함
- 기술적으로 구현 가능한 내용이 요구사항으로 정의되어야 함
- 개발될 제품과 관련된 모든 종류(기능, 성능, 제약사항 등)의 요구사항이 기술되어야 함
- 기존의 구조와 양식을 유지하면서 쉽고, 완전하고, 일관성 있게 요구사항을 변경할 수 있도록 기술되어야 함
- 개발될 제품에 반드시 필요한 요구사항들만 기술되어야 함

예제를 통한 요구사항 명세

무료배송

- 무료 배송은 VIP 고객이 책을 특정 수량만큼 구입할 때 제공된다. 일반 고객이나 VIP 고객이 책 외의 다른 것을 구입할 때는 무료 배송을 제공하지 않는다.
- 무료 배송을 받기 위한 책의 최소량이 5권일 경우, 다음과 같은 사항을 기대할 수 있다.

고객 유형	장바구니 내용	배송
VIP	책 5권	무료배송
VIP	책 4권	일반배송
일반	책 10권	일반배송
VIP	노트 5권	일반배송
VIP	책 5권, 전자기기 1대	일반배송

[출처: Specification by Example]

- 개발과 테스트에 일관되게 사용되는 공통의 예제 사용
- 명확하고 완전하게, 현실적이고 이해하기 쉽게 작성

Context & Services Model :

- 소프트웨어가 어떤 기능을 제공해줄 수 있는지를 설명할 수 있는 모델링
- Use Case 다이어그램
 - 개발 대상 시스템에 대해 원하는 동작을 캡처하기 위한 수단
 - 시스템의 동작을 전달하는 방법
 - 시스템과 상호 작용하는 사람 또는 대상 식별, 그들 간에 수행해야 하는 작업 식별
 - 모든 요구사항이 식별되었는지 검증
- 장점
 - 요구사항에 대한 컨텍스트 제공
 - 이해하기 쉬움
 - 고객과의 합의 촉진
 - 시스템이 필요한 이유를 설명
 - Use cases: 시스템이 사용되는 이유
 - Actors: 누가/무엇이 시스템과 상호 작용하고 싶어 하는지
- 문법
 - Actor/Service(Use-case)/Relation

Usecase Diagram 구성요소

시스템(System)

- 개발하고자 하는 시스템의 범위

액터(Actor)

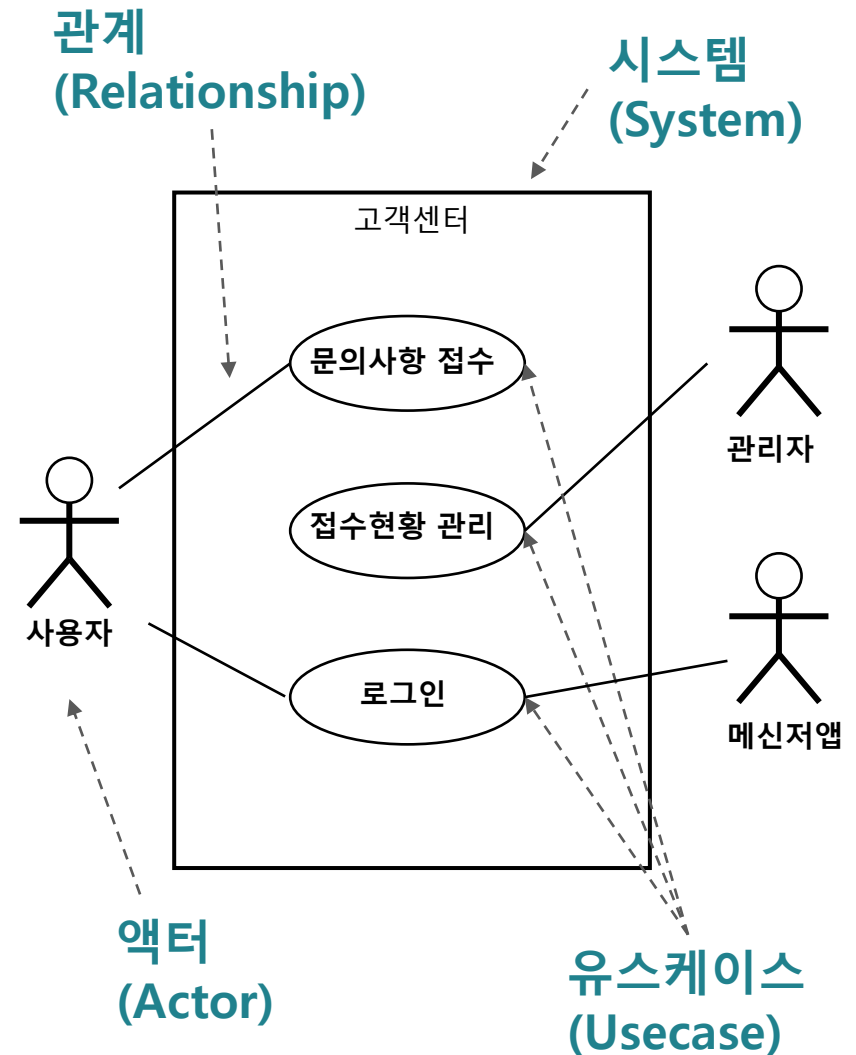
- 시스템의 외부에 있으면서 시스템과 상호작용을 하는 사용자 혹은 외부 시스템

유스케이스(Usecase)

- 시스템이 액터에게 제공해야하는 기능의 집합
 - 사용자가 어떤 목적으로 시스템과 상호작용해야 하는지 보여주는 하나의 시나리오

관계(Relationship)

- 액터와 유스케이스 / 유스케이스와 유스케이스 사이의 연관 관계
 - 일반화 관계
 - 포함 관계
 - 확장 관계



Usecase 모델 작성 방법

1. 시스템 범위 식별

2. 시스템과 연관된 액터 식별

3. 유스케이스 식별

1. 액터가 요구하는 서비스 식별
2. 액터가 요구하는 정보 식별
3. 액터가 시스템과 상호작용하는 행위 식별

4. 관계 식별

1. 액터와 유스케이 간의 연관 관계 정의
2. 유스케이스 간 포함/확장 관계 정의
3. 액터 간, 유스케이스 간의 일반화 정의

5. 유스케이스 기술서 작성

6. 유스케이스 모델 검증

Use Case Diagram : Actor 식별

○ Actor

- 시스템과 상호 작용하는 시스템 외부의 시스템/사용자

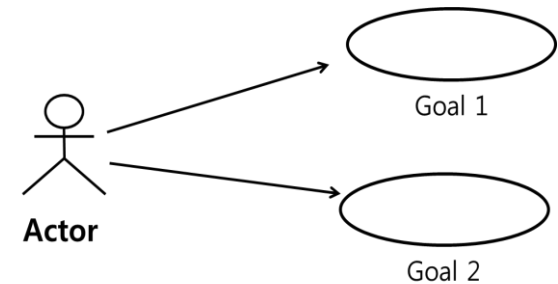
○ Actor를 식별하기 위한 질문

- 누가/무엇이 시스템을 사용하나요?
- 누가/무엇이 이 시스템으로부터 정보를 얻나요?
- 누가/무엇이 이 시스템에게 정보를 제공하나요?
- 회사내의 어디서 이 시스템을 사용하나요?
- 누가/무엇이 이 시스템을 지원하거나 유지보수하나요?
- 이 시스템이 어떤 다른 시스템을 사용하나요?

Use Case Diagram : Use-case 식별

○ Use-case

- Actor가 이 시스템을 사용하여 수행하려는 작업
- 시스템에서 수행하는 일련의 작업을 정의



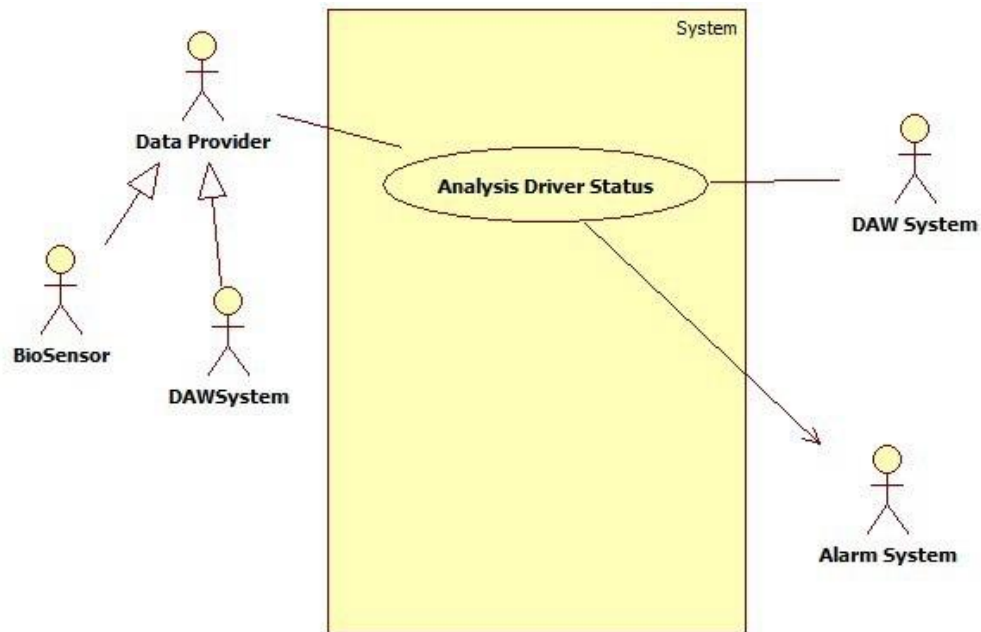
○ Use-case 식별

- 각 actor의 목표는 무엇인가요?
- Actor가 이 시스템을 사용하는 이유는 무엇인가요?
- Actor가 시스템에서 데이터를 생성/저장/변경/삭제 또는 조회하나요? 그렇다면 그 이유는 무엇인가요?
- Actor가 외부 이벤트 또는 변경 사항에 대해 이 시스템에 알려야 하나요?
- Actor에게 이 시스템의 특정 사건(이벤트)에 대해 알려야 하나요?

Usecase는 시스템의 요구사항으로, 하나의 유스케이스는 하나의 기능(서비스)!

- 사용자에게 어떤 가치를 제공할 수 있는 크기의 기능이어야 함. (시스템의 단위 기능 X)
 - 사용자의 입력에 대한 시스템의 응답
 - 사용자의 입력으로부터 시작하여, 시스템의 응답으로 종료될 때 까지의 일련의 과정 내포.

[예제] Use Case Diagram



[실습] Use Case Diagram 작성

- Use Case Diagram 표기법을 참고하여, 앞서 실습한 시스템/소프트웨어 요구사항의 Use case Diagram을 작성해보자.

[예시]

- 이해관계자 요구사항
 - 근본 원인 : DB에 입력되어 있는 아이디와 비밀번호를 사용하여 로그인
 - 요구사항 : 회원가입을 통해 정해진 회원만 시스템에 접근
- 시스템 요구사항
 - XX 시스템은 정해진 회원만 접근 가능하다.
 - XX 시스템은 “회원가입” 페이지를 통해 회원가입이 가능하며, XX 시스템의 회원 유형은 다음과 같다.
 - 관리자 : ... / DB 관리자 : ... / 일반 회원 : ...
 - XX 시스템은 “로그인” 페이지를 통해 로그인/로그아웃을 할 수 있다.
- Usecase Diagram

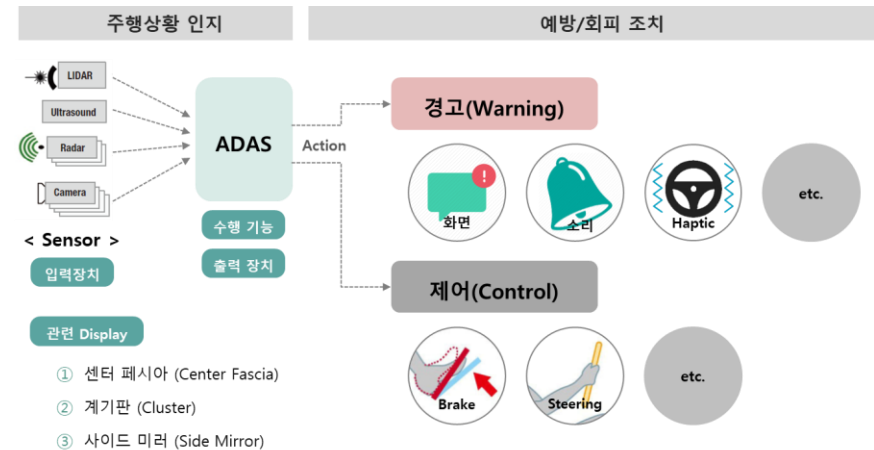
[실습] Use Case Diagram 작성

- Use Case Diagram 표기법을 참고하여, 앞서 실습한 시스템/소프트웨어 요구사항의 Use case Diagram을 작성해보자.

[예시]

이해관계자 요구사항

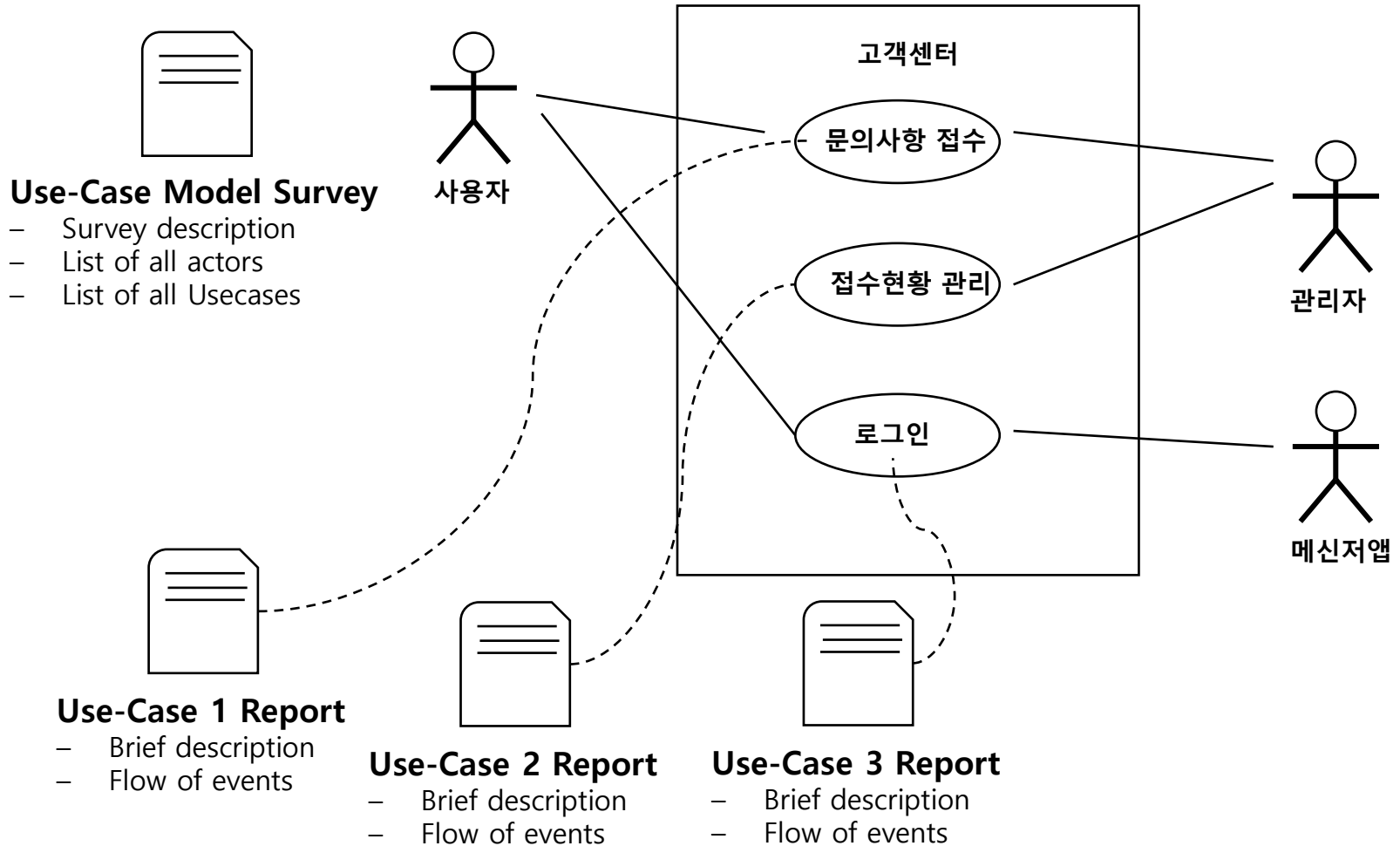
- 졸음 운전을 방지하기 위한 알람 시스템



시스템 요구사항: 헬스케어 시스템 - 운전자의 건강상태를 분석해서, 졸음 상태이면 알람 시스템이 동작한다.

- 헬스케어 시스템은 생체정보를 수집하여 운전자의 건강 상태를 파악할 수 있다.
 - 운전자의 건강 상태는 다음과 같이 정의한다. ...
 - 비접촉신 센서를 통해 운전자의 움직임과, 스티어링 휠에 내장된 접촉신 센서를 통해 심전도, 심박 등을 측정하여 생체정보를 수집한다.
- 운전자의 건강 상태에 따라 HMI를 통해 안내/경고를 표시할 수 있다.
 - 일반 상태 :
 - 졸음 상태 :
 - ...
- 졸음 감지 시, XXX 시스템과 연계하여 긴급 연락을 취하거나 안전 운전을 유도할 수 있다.

Use Case Scenario



Use Case Scenario

- 하나의 기능이 어떻게 동작하는지를 설명, 그러나 기술적인 요소를 제외
- Use-Case Specification (each use-case)
 - Pre-conditions
 - Post-conditions
 - Trigger
 - Main Success Scenario
 - Alternative (success) scenarios
 - Exception (failure) scenarios

[참고] 시나리오(Scenario)

○ 결과를 달성하기 위한 사건의 흐름

○ 정상 흐름(Normal Flow) - Success

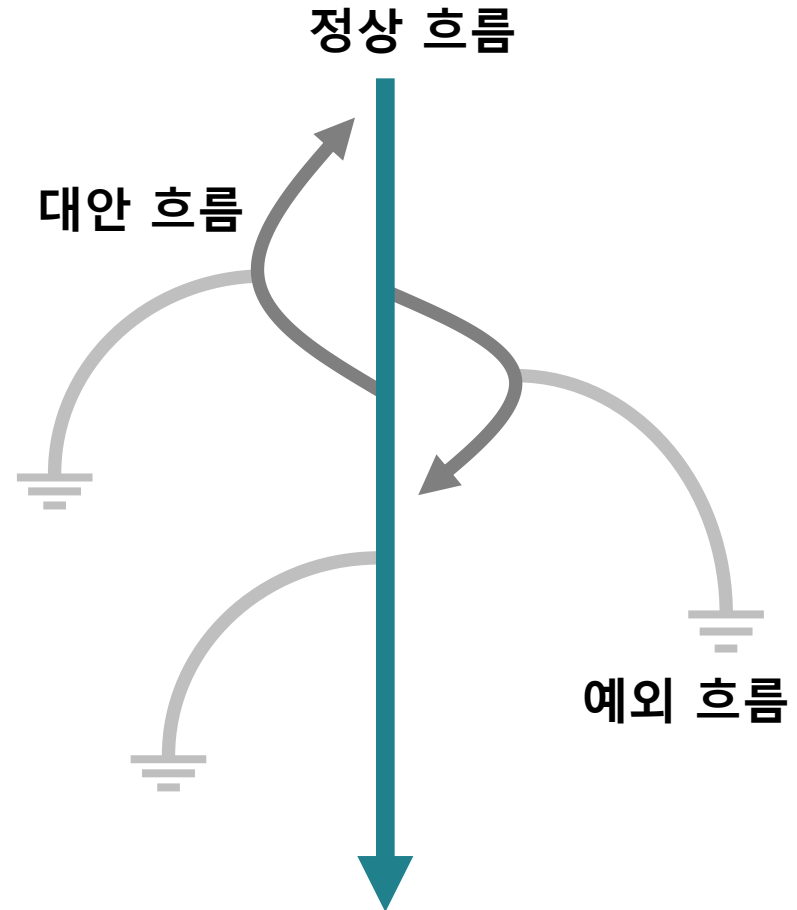
- 정상적으로 수행되는 흐름

○ 대안 흐름(Alternative Flow) - Success

- 작업 흐름이 수행되는 중에 특정 시점에서 여러가지 선택적인 흐름으로 나뉘어질 경우에 수행되는 흐름
 - 선택 흐름 별 사후 조건이 다를 경우 구분하여 작성

○ 예외 흐름(Exceptional Flow) - Failure

- 작업 흐름이 수행되는 중에 발생할 수 있는 예외 상황이나 오류를 표현하는 흐름
 - 사후 조건으로 연결되지 않음
 - 예외 상황에 대한 처리 방안 서술



[예제] Use Case Scenario

UC-1	로그인 및 로그아웃 - 메신저를 사용하기 위해 로그인을 하거나, 메신저 사용 종료를 위해 로그아웃을 한다.
Pre-condition	프로그램을 실행시킨다.
Post-condition	메인 화면이 실행된다.
Trigger	로그인 버튼 클릭
Main Success Scenario (MSS)	<ol style="list-style-type: none"> 1. 사용자가 아이디와 비밀번호를 입력하고 로그인 버튼을 클릭한다. 2. 시스템은 아이디 존재 여부를 확인한다. 3. 시스템은 아이디에 맞는 비밀번호인지 확인한다. 4. 아이디와 비밀번호가 맞으면 사용자임을 인증하고 메인 프로그램을 실행시킨다.
Alternative Scenario 1	1. 카카오 인증으로 로그인하는 경우
Exception Scenario	<ol style="list-style-type: none"> 1. 사용자가 아이디를 잘못 입력한 경우 2. 사용자가 비밀번호를 잘못 입력한 경우 3. 사용자가 아이디와 비밀번호를 5회 이상 잘못 입력한 경우 <p>...</p>

[예제] Use Case Scenario

UC-1	Analysis Driver Status
Pre-condition	Bio sensor and DAW system are operational
Post-condition	Alarm system displays the status on the screen and makes alarm sound.
Trigger	The data providers operate.
Main Success Scenario (MSS)	<ol style="list-style-type: none"> 1. The system requests bio data to the bio sensor. 2. The bio sensor response with bio data to the system. 3. The system requests DAW data to the DAW system. 4. The DAW system response with DAW data to the system. 5. The system extracts features using bio data, DAW data. 6. The system classifies the driver status using the both features. 7. The system sends the driver's status, alarm type, alarm info. to Alarm system.
Alternative Scenario 1	Alternative in step 2 of MSS: The system fails to gather bio data Go to step 2. & skip step 4 & classify status using one feature.
Alternative Scenario 2	Alternative in step 3 of MSS: The system fails to gather DAW data. Go to step 4 & skip step 5 & classify status using one feature
Exception Scenario	Exception in step 2 & 3 of MSS: The system fails to gather all data. Report the problems to the alarm system and take action according to the failover policy

Q1. 시나리오를 보고 테스트케이스를 만들 수 있을까요?

[실습] Use Case Scenario 작성

UC-1	
Pre-condition	
Post-condition	
Trigger	
Main Success Scenario (MSS)	
Alternative Scenario 1	
Alternative Scenario 2	
Exception Scenario	

[예제] 요구사항 검토 체크리스트

완전성과 관련된 검토 체크리스트 예시

- ☑ 기능과 운영 모드/운영 환경 등을 모두 도출하였는가?
- ☑ 누락된 정보는 TBD(To Be Determined)로 표시하였는가?
- ☑ 기능 이외의 비기능 요구사항을 모두 반영하였는가?

명확성과 관련된 검토 체크리스트 예시

- ☑ 요구사항이 중의적인 의미를 포함하지는 않는가?
- ☑ 요구사항이 상충되거나 이해하기 어려운가?
- ☑ 다이어그램의 표기법이 정확한가?

실행 가능성과 관련된 검토 체크리스트 예시

- ☑ 프로젝트 일정, 기간 내에 구현 가능한 범위인가?
- ☑ 조직의 기술 수준을 고려할 때 구현 가능한가?
- ☑ 적절한 우선순위가 정해졌는가?

추적 가능성과 관련된 검토 체크리스트 예시

- ☑ 고객 요구사항과 추적 가능한가?
- ☑ 시스템 요구사항과 추적 가능한가?
- ☑ 요구사항 별로 고유 식별자가 존재하는가?