



SW 요구사항 개발

가이드라인 개발 워크숍

Synetics _

한동준 대표

dongjoon.han@synetics.kr

handongjoon@gmail.com

워크숍 목표

- SW 요구사항을 개발하기 위한 가이드라인을 실습 팀 단위로 작성합니다.
 - 이해 관계자 식별
 - 이해 관계자 요구사항 추출
 - 기능 요구사항 정의
 - 비기능 요구사항 정의
 - 제약사항 정의
 - 우선순위 결정
 - 요구사항 검증 체크리스트
- 팀 단위의 작성, 발표, 개선을 반복합니다.
 - 다른 팀의 내용을 확인하고, 우리 팀의 개선 기회를 식별 후 반영

가이드라인이란

○ 지침

- 행동 방식을 결정하기 위한 설명

○ 정해진 프로세스의 Task를 수행하기 위해 작성한 설명서

- 양식(Template)에 따라 산출물을 작성하는 방법을 설명
- “그래서 어떻게 해야하는가?”에 대한 답

○ [참고]

- 일반적으로 “양식”이 우선 정의되어 있고, 이를 작성하기 위한 가이드라인을 개발해야 하나, 과정의 특성 상 “필요 양식 ” 과 “가이드라인 ” 을 함께 개발함

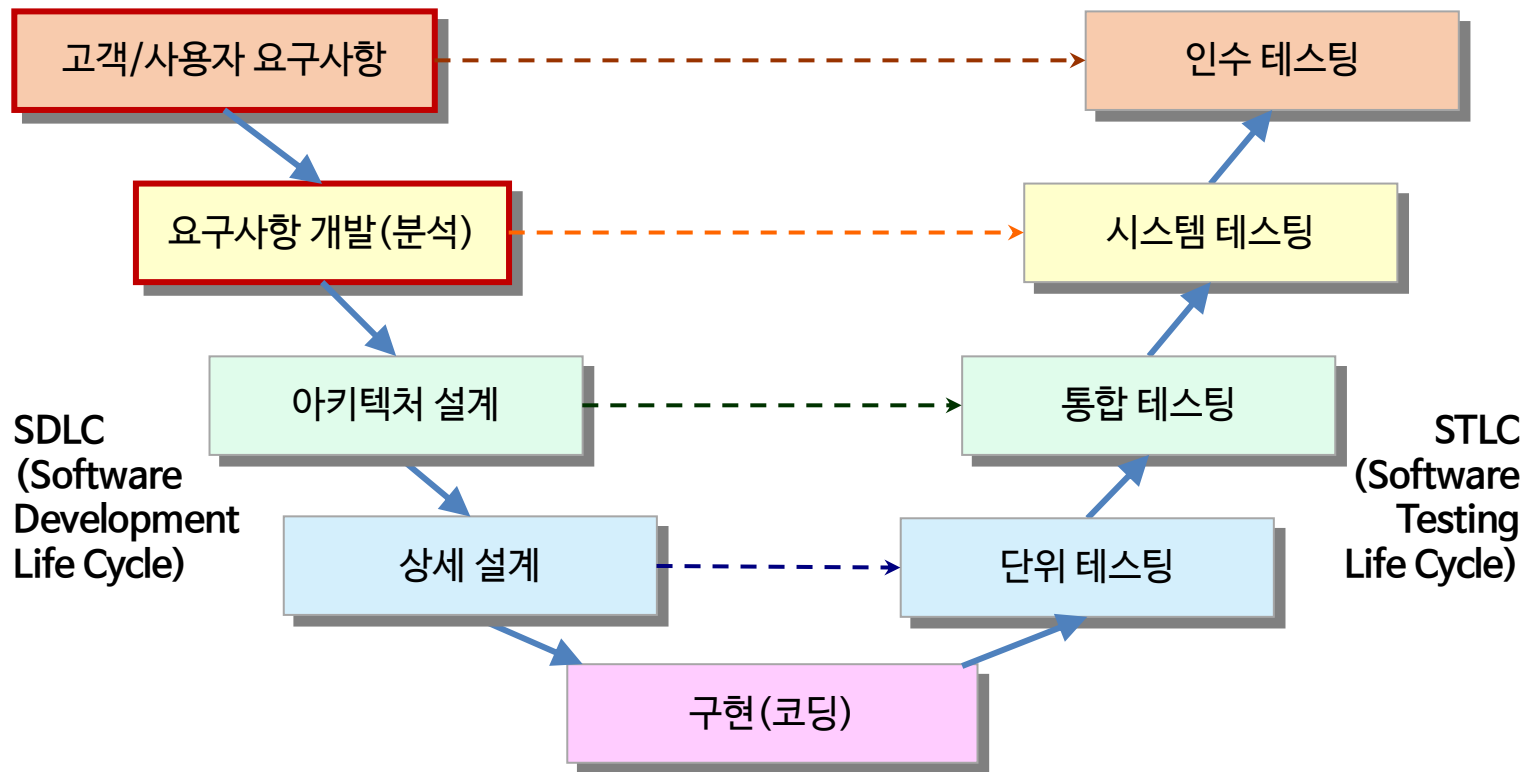


[들어가며] 요구사항 살펴보기

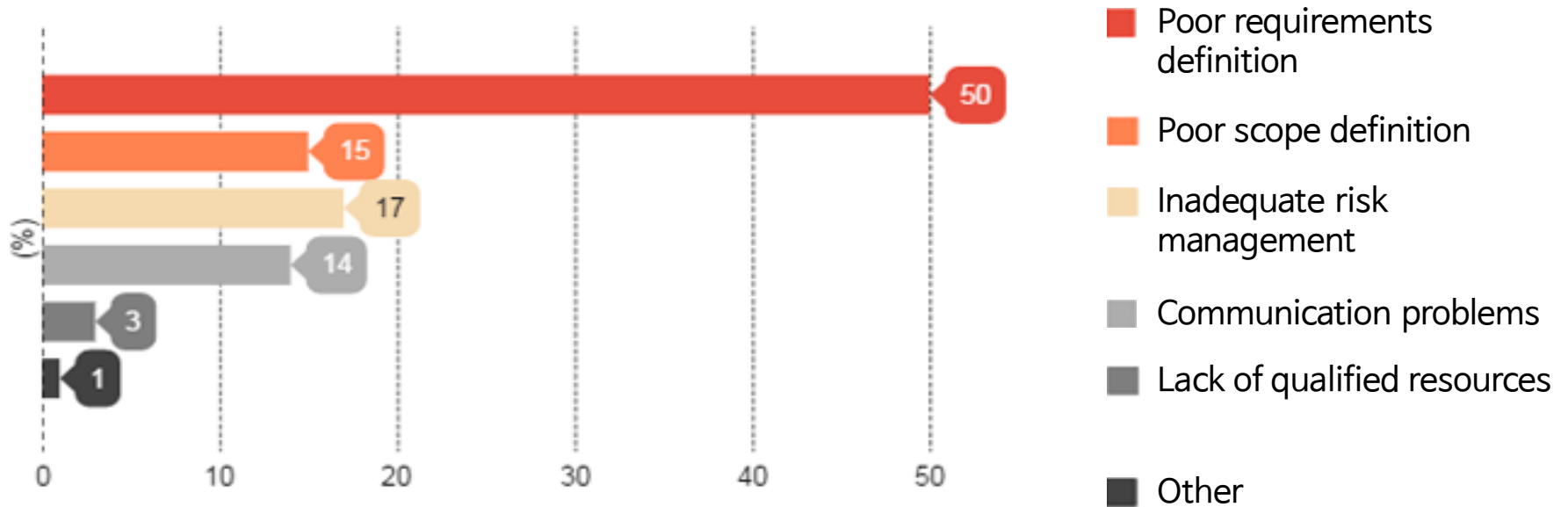
V-Model

- 개발 생명주기의 각 단계와 그에 상응하는 소프트웨어 테스트 단계를 매핑한 모델

- 테스트 설계와 같은 테스트 활동을 코딩 이후가 아닌 프로젝트 시작 시에 함께 시작

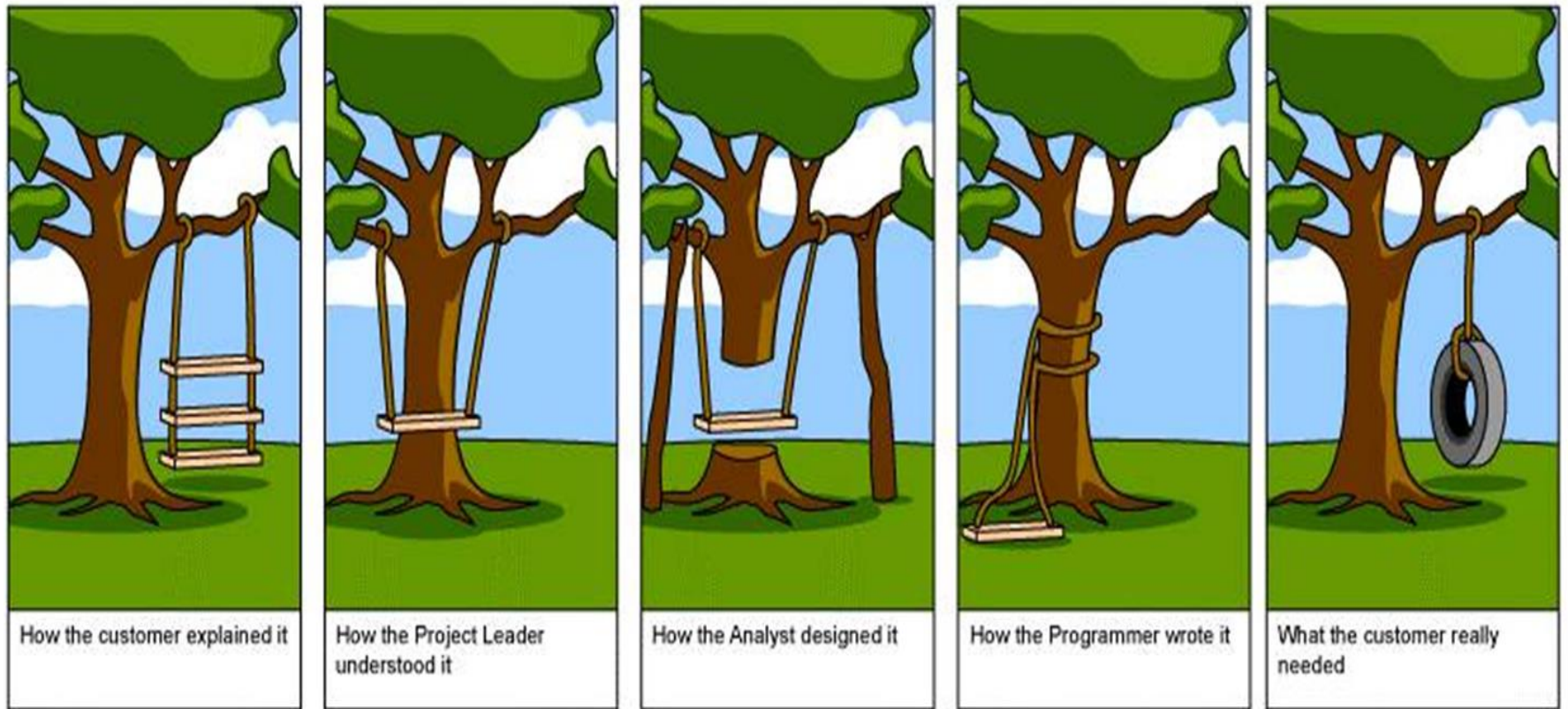


SW 프로젝트의 실패 원인



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

요구사항 정의의 어려움



출처) <https://www.pmchamp.com/how-to-define-scope-of-your-project/project-scope-tyre-cartoon/>

자동차 요구사항은 왜 더 어려운가?

① 다양한 기술 특성

- 기계, 전기, 전자, 화학, 생명공학, 소프트웨어 등의 다양한 기술들이 통합되어 완성되는 시스템

② 시스템의 규모

- 자동차가 지능화되어 가면서 시스템의 복잡도가 기하 급수적으로 증가

③ 협업 관계

- 업무의 세분화 및 분업화로 OEM과 Tier 간의 협업 관계 증가

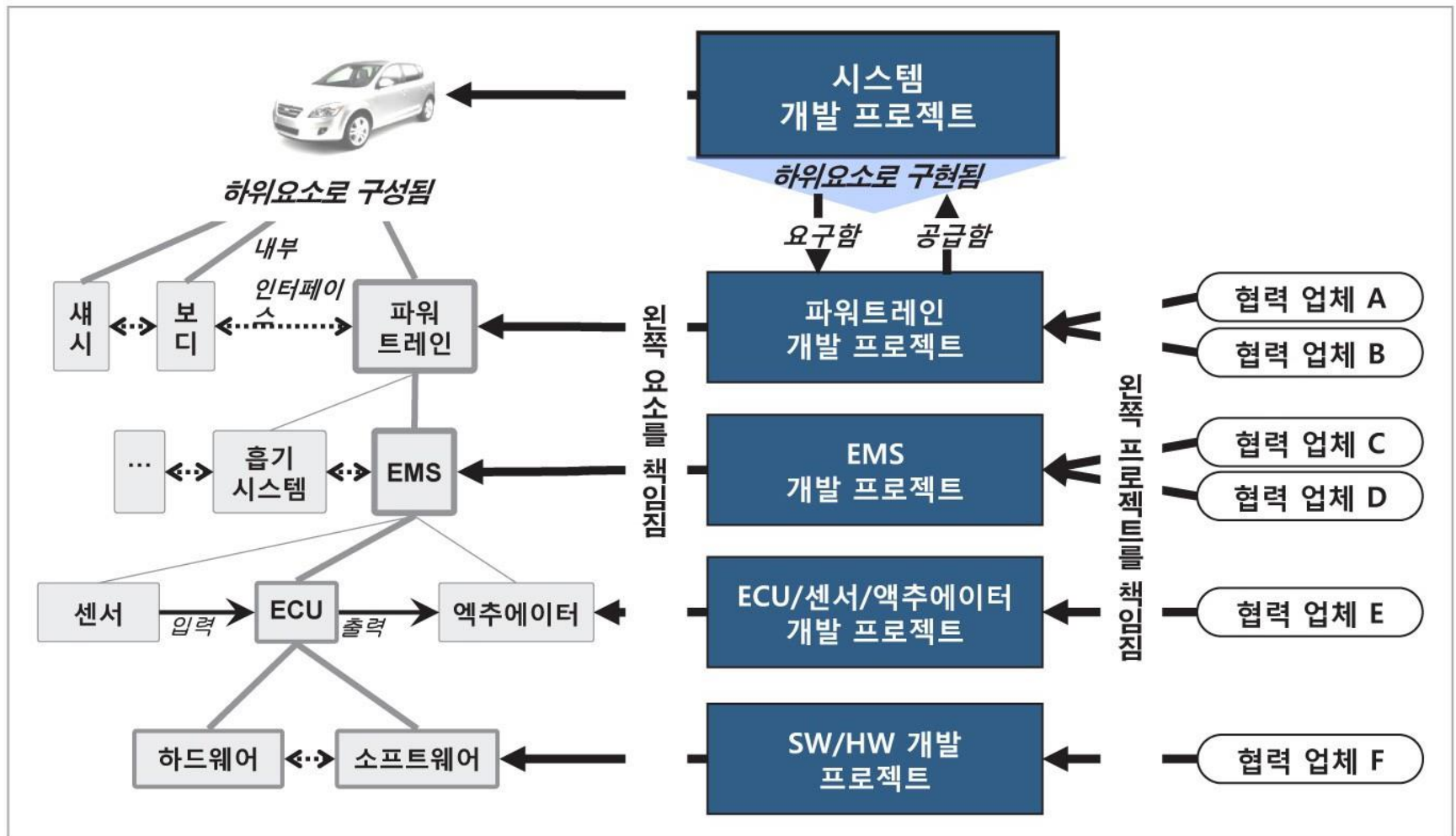
④ 법, 규제 강화

- 자동차 시스템은 인간의 생명, 안전 보장 그리고 환경 보호에 민감한 시스템

⑤ Time-to-Market

- 신기술 개발과 더불어 Time-to-Market의 강화

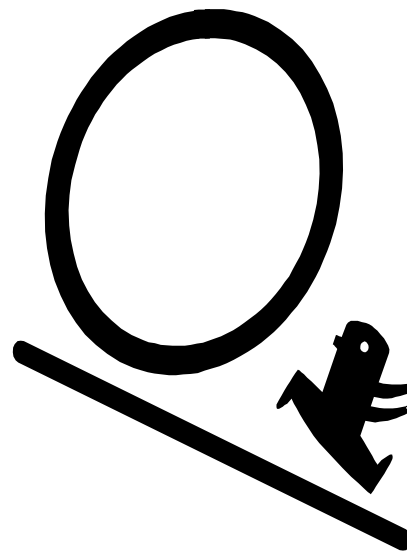
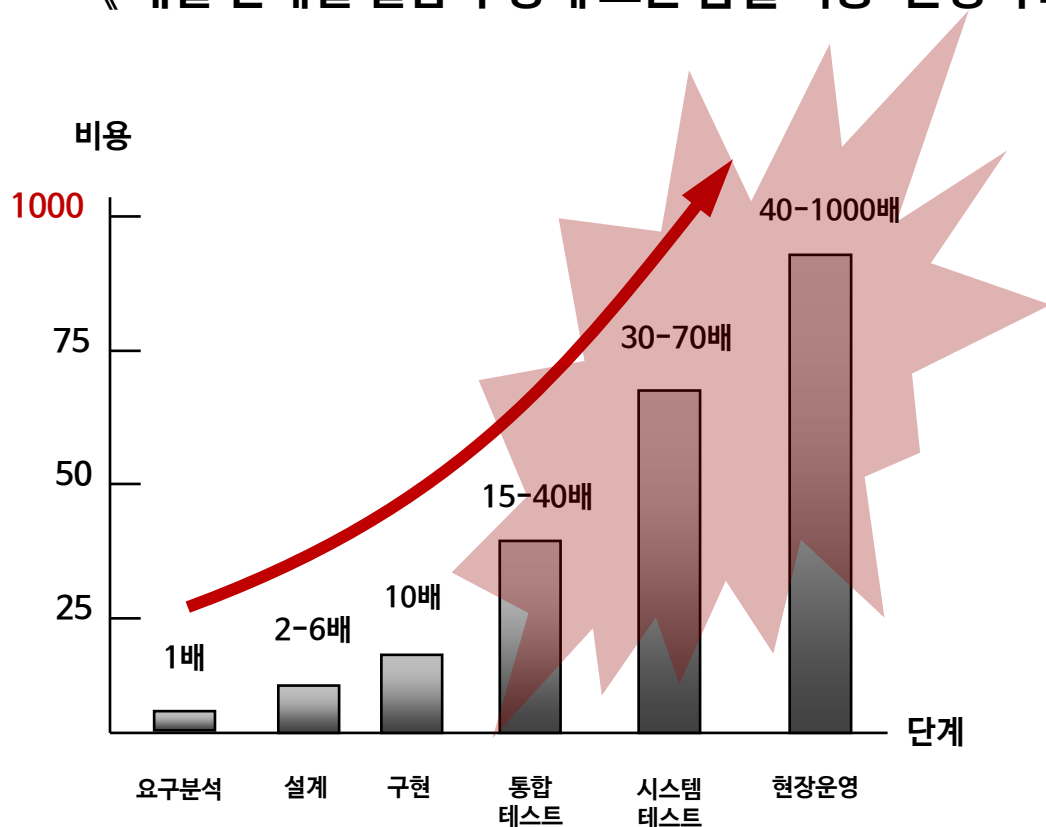
[참고] 자동차 협업 체계



요구사항의 중요성(1/3)

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

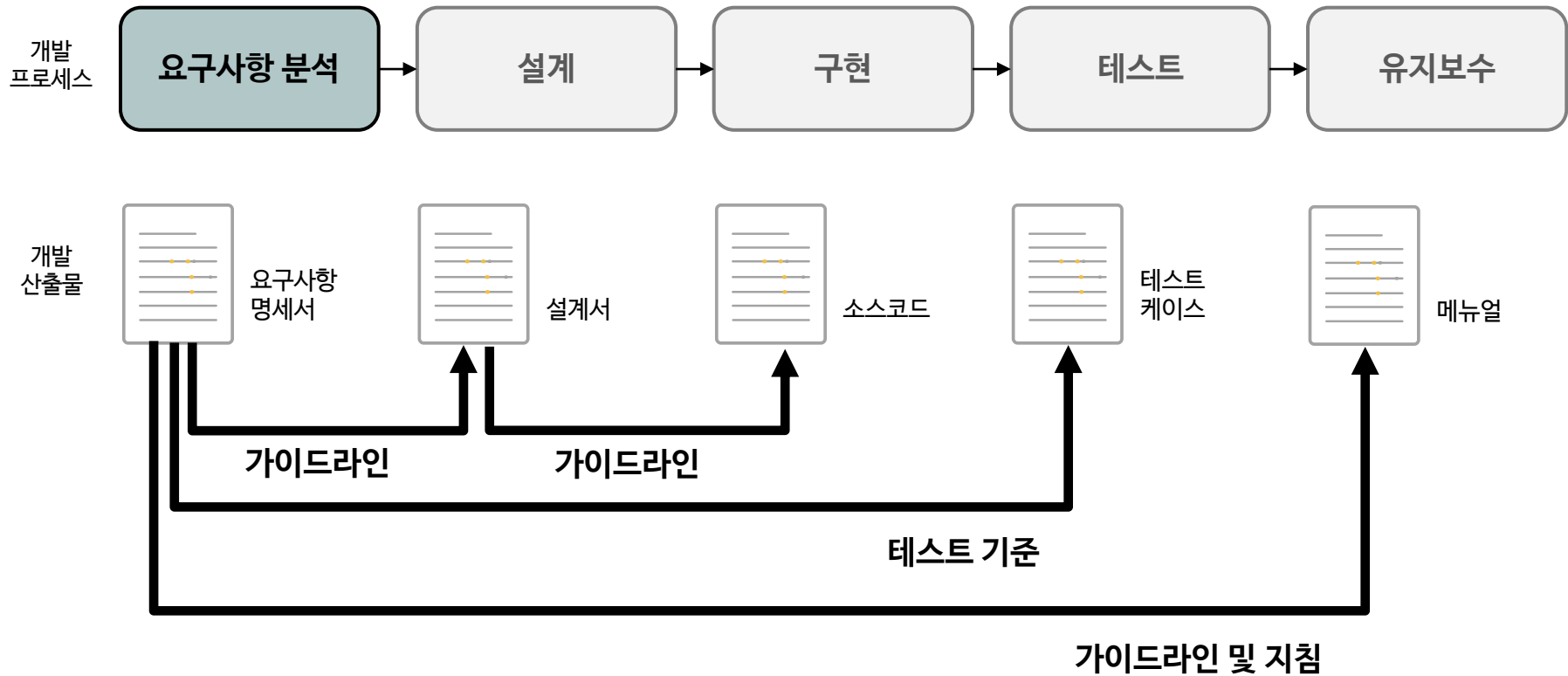
《 개발 단계별 결함 수정에 드는 품질 비용: 눈덩이 효과 》



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

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

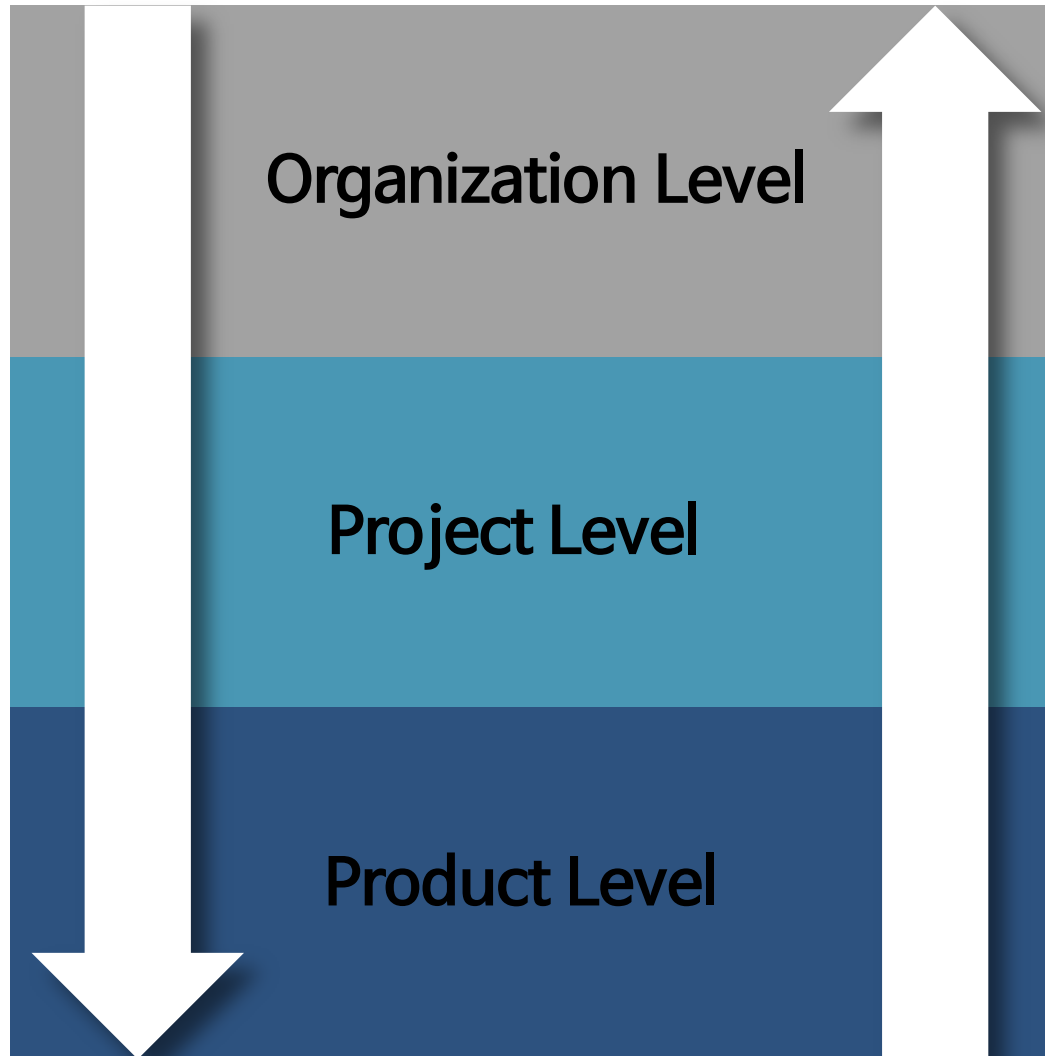
④ 프로젝트 일정/예산 등을 고려하지 않음

- 기술적 난이도, 제약사항 등을 고려하지 않고 정의됨



자동차 개발 프로세스에서의 요구사항

자동차 분야의 프로세스 모델



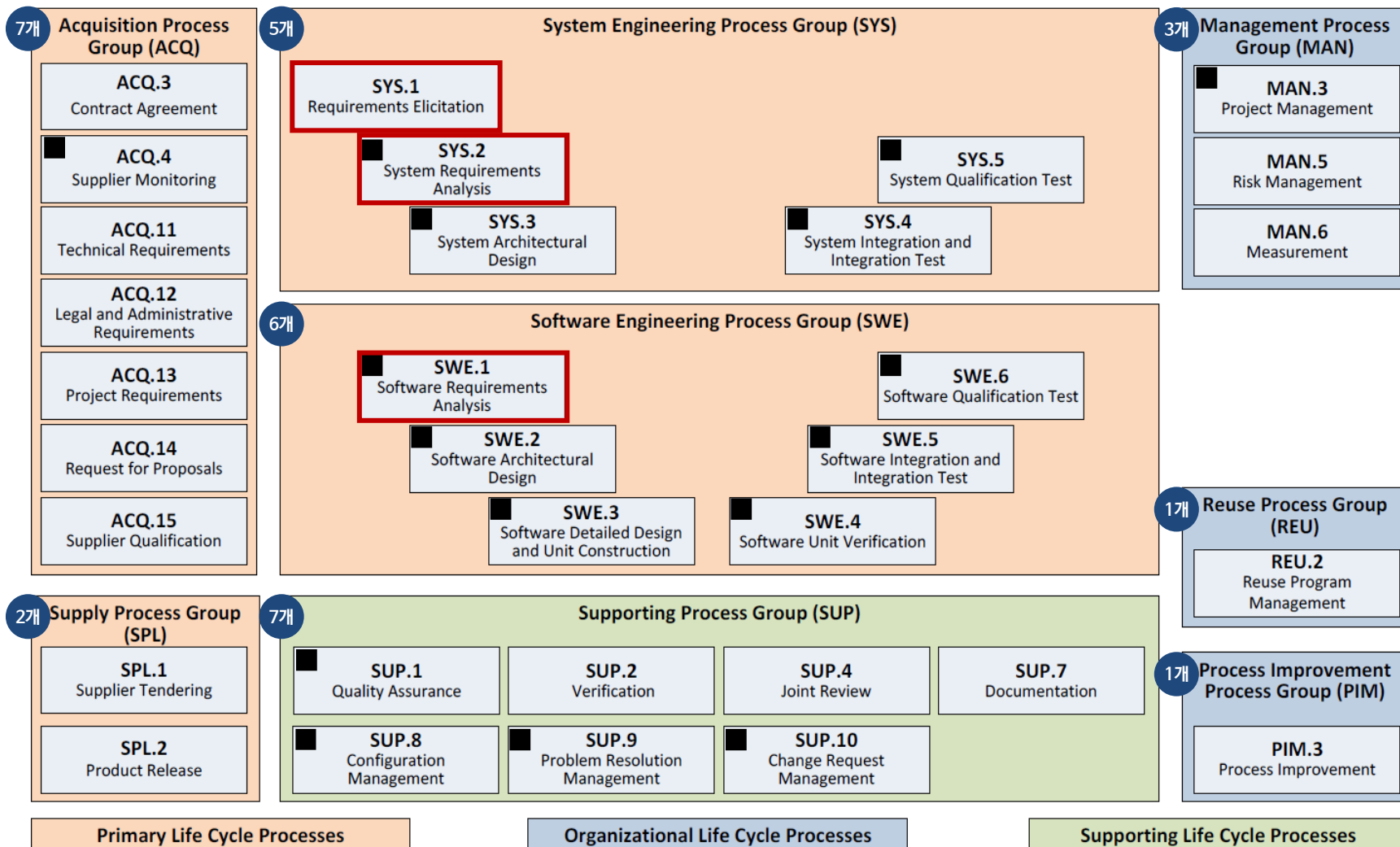
- 자동차 소프트웨어 개발에 특화된 Process Reference Model / Process Assessment Model



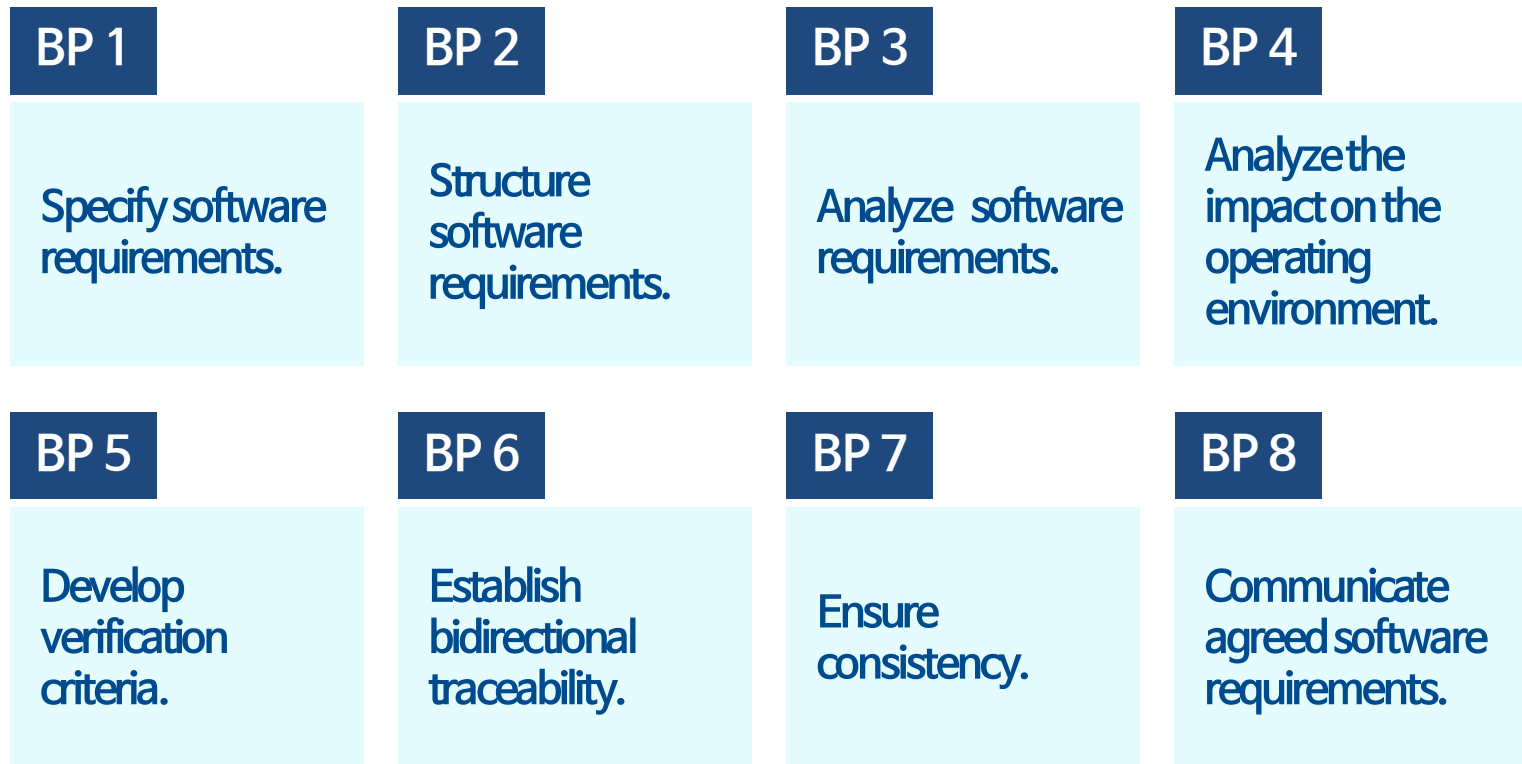
- 자동차 분야의 기능 안전(Functional Safety) 국제 표준

A-SPICE 에서의 요구사항

■ : VDA Scope

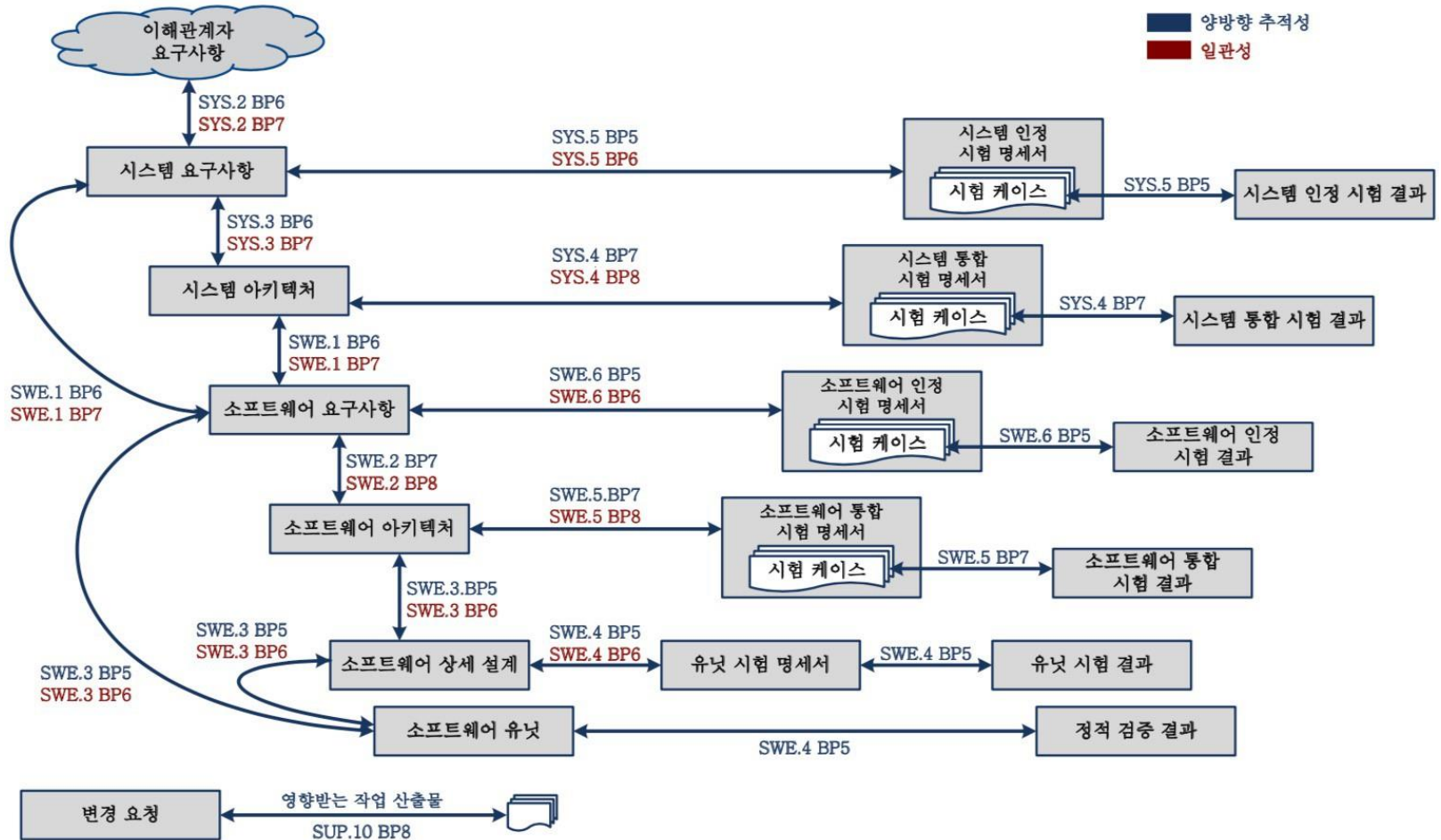


[참고] A-SPICE : Software Requirements Analysis Process (SWE.1)



[참고] 요구사항 추적관리

- 프로젝트 Control과 QA를 제공, 요구사항 변경에 의한 재결함 유입방지



요구사항 정의



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)

Needs와 Requirement의 차이



Problem



Solution



Needs



Requirement



요구사항(Requirement)이란? (2/3)

고객의 기대 : Expectation



Problem description

- 고객 언어로 표현된 해결할 문제
- 고객의 SW에 대한 승인 조건

현실 : Constraints



Solution specification

- 제약 조건하에 고객 Needs 충족 방법
- 제품/서비스 만족을 위한 특성



요구사항(Requirement)이란? (3/3)



- 해당 시스템과 연관된 이해관계자(Stakeholder)는?
- 모든 이해관계자가 요구사항을 동일하게 이해할 수 있을까?



다양한 이해관계자의 문제/요구를 만족시키기 위한 해결책(Solution)
→ 요구사항(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]



단계별 요구사항 및 요구사항 종류

단계별 요구사항

추상화

비즈니스
요구사항

프로젝트가 **왜(Why)** 시작되었는가?

비즈니스 목표, 목적, 범위 등 시스템이 개발되어야 하는 궁극적인 비전을 서술

이해관계자
요구사항

사용자는 **무엇을(What)** 원하는가?

시스템과 이해관계가 있는 이해관계자(Stakeholder) 관점에서 무엇을 원하는지에 대한 기술적 요구사항

시스템
요구사항

시스템은 **왜(Why)** 필요한가?

왜 시스템이 구축되어야 하는지, 어떻게(How) 동작할 것인지에 대한 시스템 관점의 기술적인 요구사항

서브시스템/
컴포넌트
요구사항

상세화

어떤 **컴포넌트**(하드웨어/소프트웨어)가 **개발**되어야 하는가?

왜 컴포넌트가 구축되어야 하는지, 어떻게(How) 동작할 것인지에 대한 컴포넌트 관점의 기술적인 요구사항

시스템 요구사항과 소프트웨어 요구사항

	시스템 요구사항	소프트웨어 요구사항
Definition	<ul style="list-style-type: none"> 왜 시스템이 구축되어야 하는지, 어떻게 동작할 것인지에 대한 시스템 관점의 기술적인 요구사항 	<ul style="list-style-type: none"> 왜 소프트웨어가 구축되어야 하는지, 어떻게 동작할 것인지에 대한 소프트웨어 관점의 기술적인 요구사항
Functionality/ Quality Attribute	<ul style="list-style-type: none"> 비즈니스 요구사항이나 이해관계자 요구사항을 만족시키기 위한 시스템의 기술적인 요구사항 (기능&비기능)을 기술 	<ul style="list-style-type: none"> 시스템 요구사항을 만족시키기 위해 구현되어야 할 컴포넌트(하드웨어, 소프트웨어, 센서 등)가 만족시켜야 할 모든 기술적인 요구사항(기능&비기능)을 기술
User & Environment	<ul style="list-style-type: none"> 구현될 시스템을 운영할 환경 등의 Interfaced Factors 구현될 시스템을 사용할 사용자 	<ul style="list-style-type: none"> 구현될 소프트웨어를 운영할 환경 등의 Interfaced Factors 구현될 소프트웨어를 사용할 사용자
Integration plan for Sub-system/COTS, etc.	<ul style="list-style-type: none"> 시스템을 구성하는 하위 시스템이나 컴포넌트들의 역할과 Allocation Information & Integration Information 	<ul style="list-style-type: none"> 소프트웨어를 구성하는 하위 시스템이나 컴포넌트들의 역할과 Allocation Information & Integration Information
Constraints	<ul style="list-style-type: none"> 시스템을 개발하는 과정에서 발생하는 제약사항 	<ul style="list-style-type: none"> 임베디드 소프트웨어의 경우에는 탑재되는 시스템의 아키텍처나 하드웨어 제약 등의 제약사항을 요구사항으로 기술

요구사항 종류

○ 기능 요구사항 (Functional Requirement)

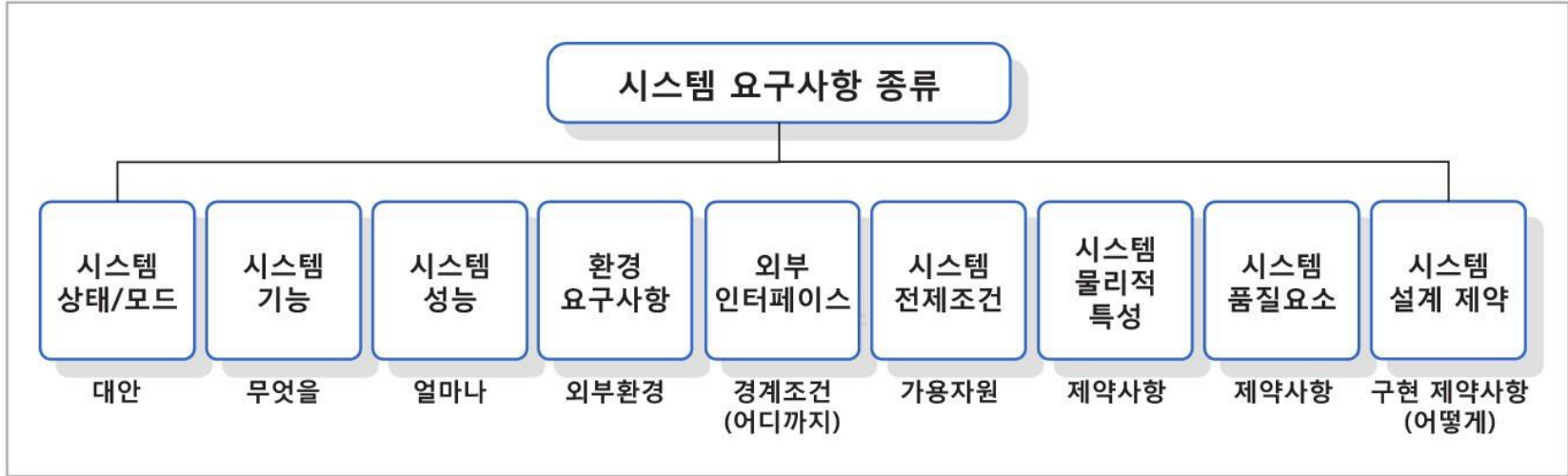
- *What the software must do?*
- 제품이 목표를 달성하기 위해 사용자에게 제공해야 하는 **행위적(기능적)** 속성
- 시스템이나 소프트웨어가 **무엇(What)**을 해야 하는지에 대해 정의한 요구사항

○ 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

시스템 요구사항의 기능 & 비기능 요구사항

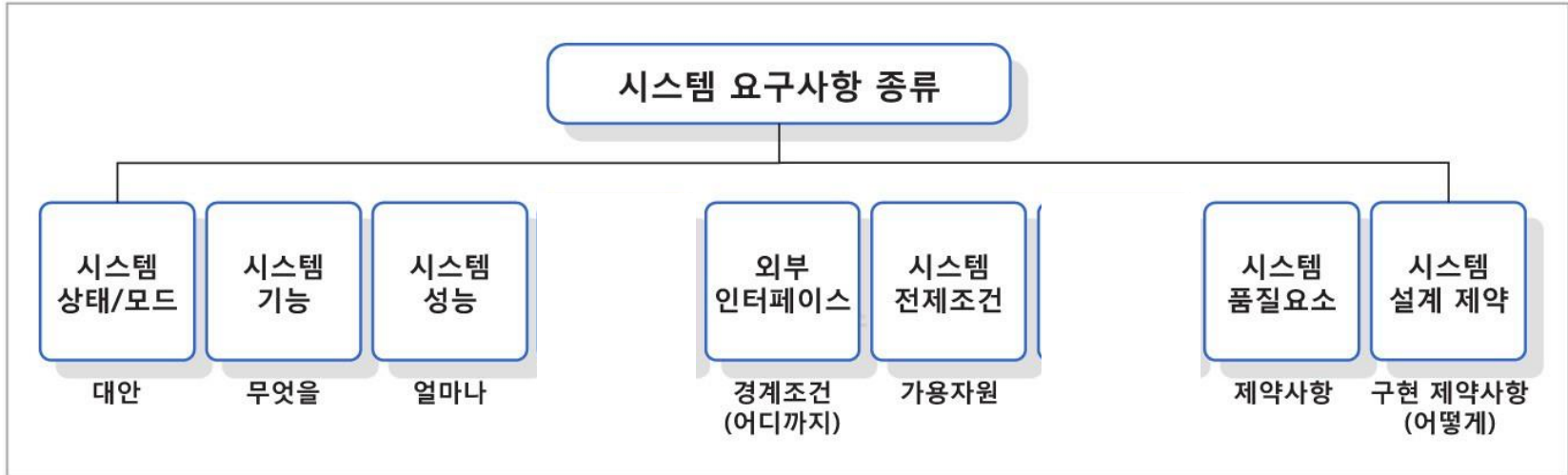
출처) 스마트카 소프트웨어 엔지니어링



- 상태/모드: 상태는 시스템의 요구조건, 모드는 상태 유지를 위한 기능
- 기능: 시스템이 수행해야 하는 기능
- 성능 요구사항: 시스템이 기능을 얼마나 잘 수행해야 하는지에 대한 목표
- 환경 요구사항: 시스템에 영향을 미치는 온도, 습도, 충격, 진동, 풍속, 먼지 등과 같은 환경 요소
- 외부 인터페이스: 시스템과 인터페이스 되어 상호 작용하는 외부 경계
- 전제조건: 시스템을 위해 전력 같이 필요한 소비 및 사용될 수 있는 가용한 자원
- 물리적 특성: 시스템의 크기, 무게, 용적, 재질 등과 같은 물리적 특성
- 품질 요소: 시스템이 만족시켜야 할 신뢰성, 안전성, 사용성, 호환성, 유지보수성과 같은 품질
- 설계 제약: 시스템의 설계 방법을 미리 제시하거나 기존 플랫폼을 재사용하는 등의 설계 제약 조건

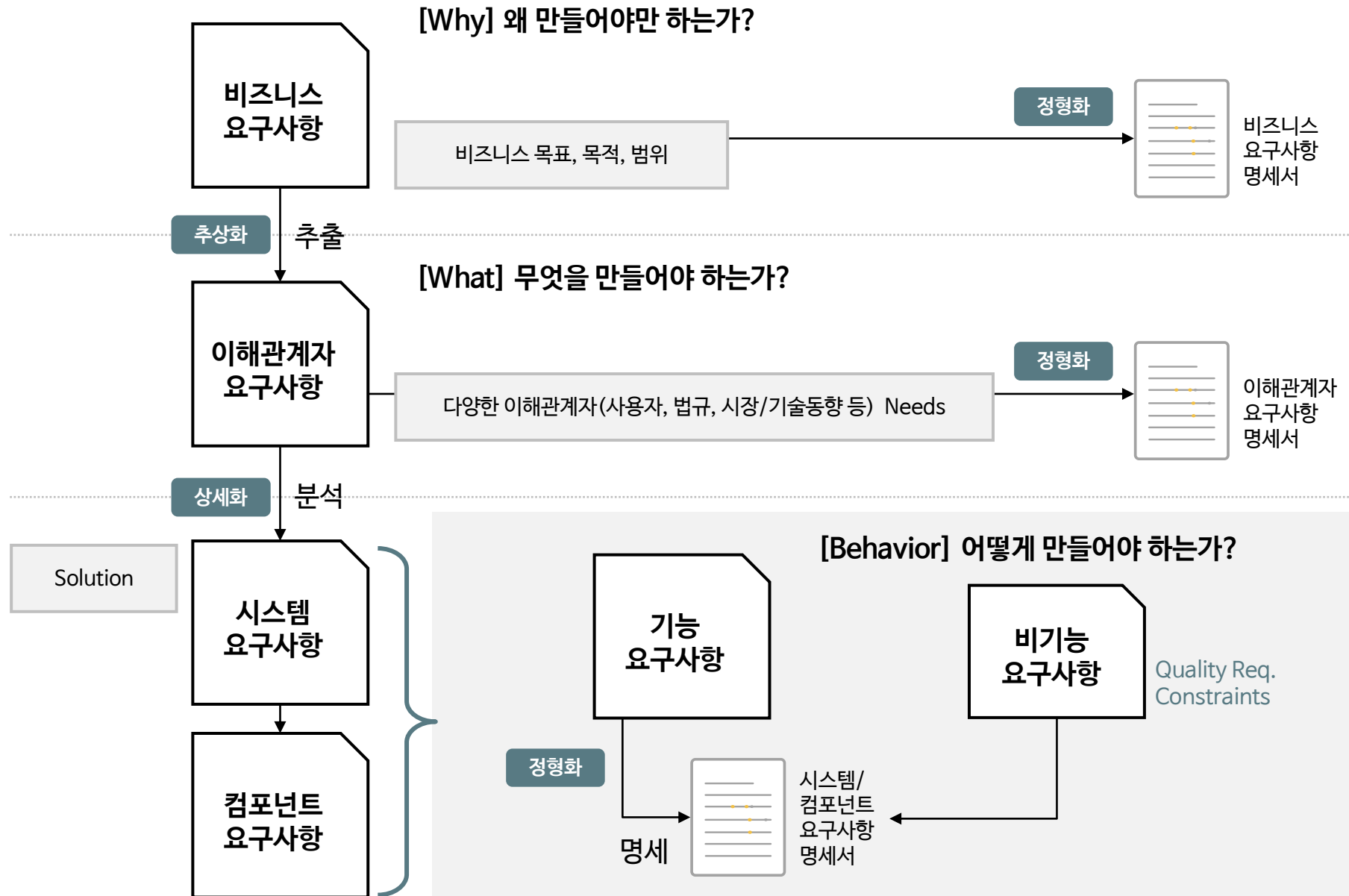
시스템 & 소프트웨어 요구사항 차이

출처) 스마트카 소프트웨어 엔지니어링



- 상태/모드: 상태는 시스템의 요구조건, 모드는 상태 유지를 위한 기능
- 기능: 시스템이 수행해야 하는 기능
- 성능 요구사항: 시스템이 기능을 얼마나 잘 수행해야 하는지에 대한 목표
- ~~환경 요구사항: 시스템에 영향을 미치는 온도, 습도, 충격, 진동, 풍속, 먼지 등과 같은 환경 요소~~
- 외부 인터페이스: 시스템과 인터페이스 되어 상호 작용하는 외부 경계
- 전제조건: 시스템을 위해 전력 같이 필요한 소비 및 사용될 수 있는 가용한 자원
- ~~물리적 특성: 시스템의 크기, 무게, 용적, 재질 등과 같은 물리적 특성~~
- 품질 요소: 시스템이 만족시켜야 할 신뢰성, 안전성, 사용성, 호환성, 유지보수성과 같은 품질
- 설계 제약: 시스템의 설계 방법을 미리 제시하거나 기존 플랫폼을 재사용하는 등의 설계 제약 조건

[참고] 단계별 요구사항 및 요구사항 종류



요구공학

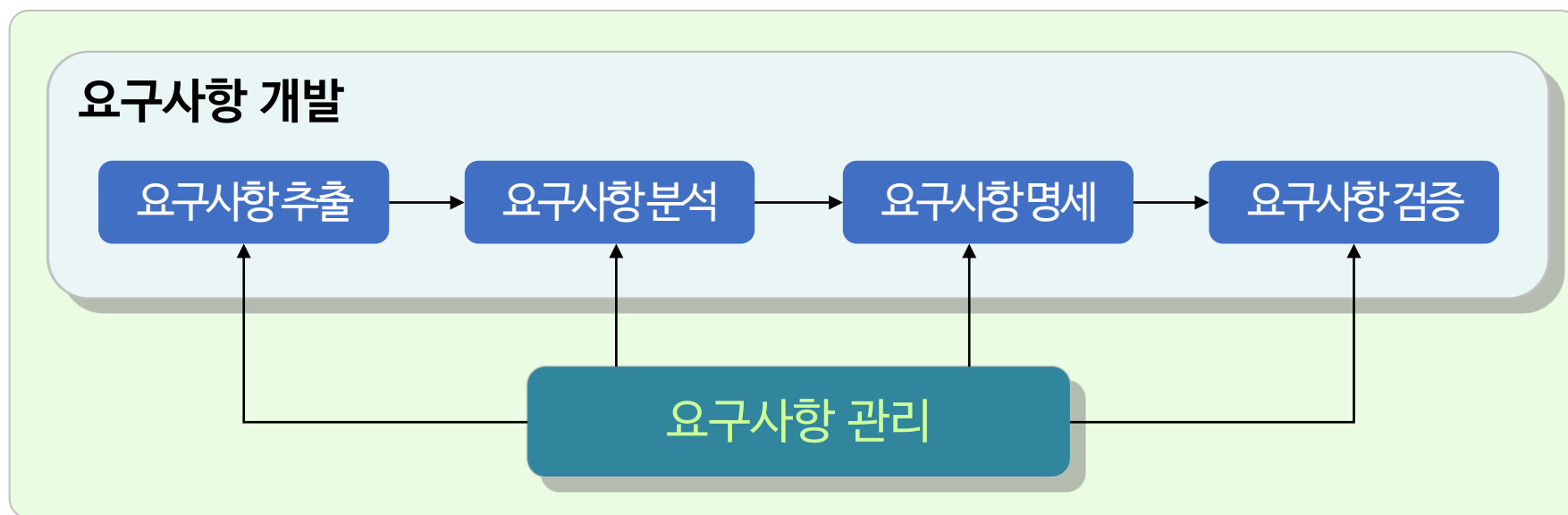


요구공학 (Requirement Engineering)

- 고객의 비공식적이고 종종 모호한 요구 사항에서 정확한 정형화된 명세를 얻는 방법.

[Sannella 1988]

소프트웨어를 개발하기 위한 **요구사항을 추출, 분석, 명세, 검증**하며,
개발된 **요구사항의 변경 및 추적을 관리**하는 공학적 접근 방법



요구공학의 다양한 관점

○ 비즈니스 관점(Faster) : Traceability

Make business profit through SW Business-Oriented Management

- 수/발주자간 공통 기대/목적 제공 비즈니스 요구의 가시성 확보

○ 개발 관점(Cheaper) : Qualitative

Minimize dev. effort and defect Process-Oriented Management

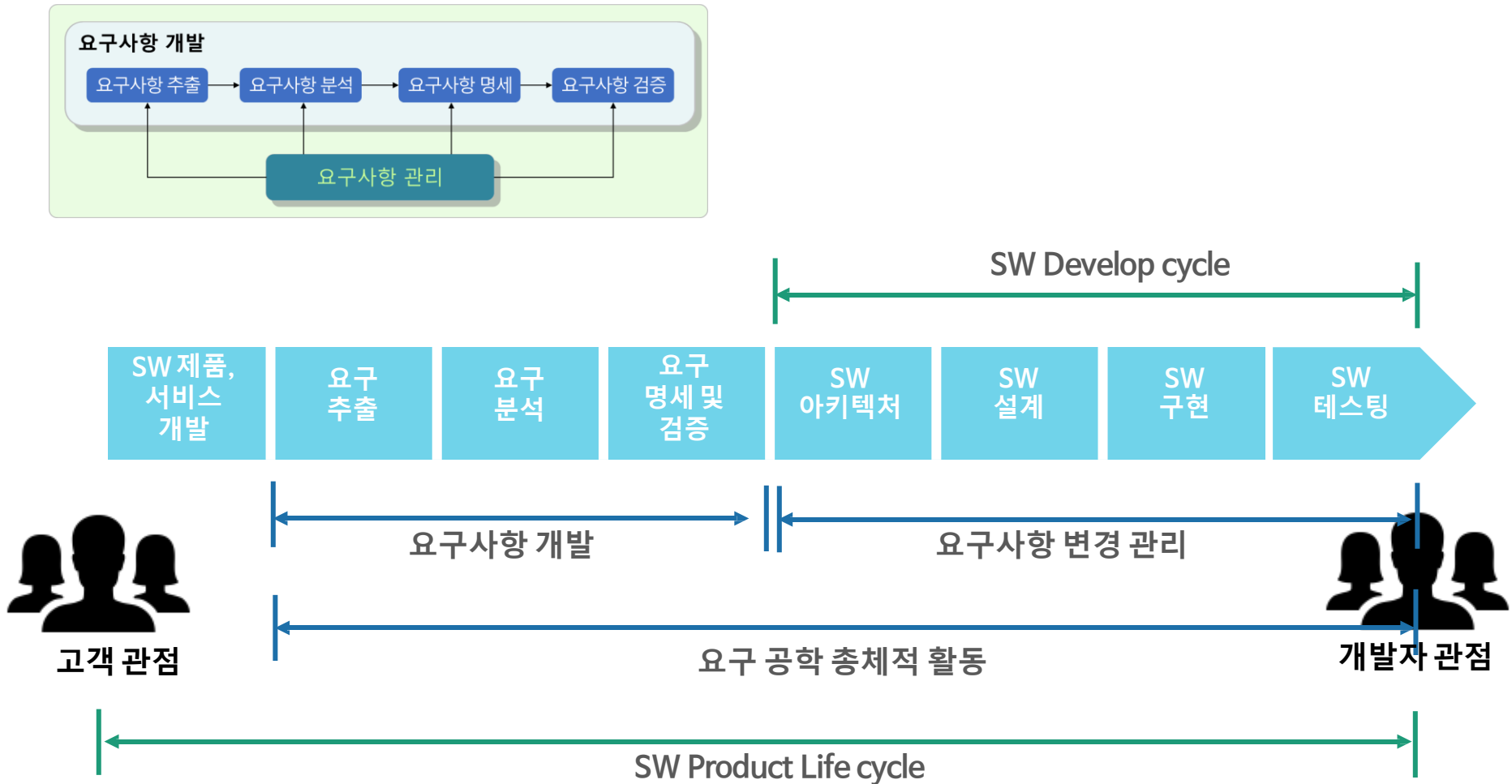
- 수/발주자간의 의사소통 제공 SW개발 정확성 및 변경최소화

○ 고객 관점(Better) : Quantitative

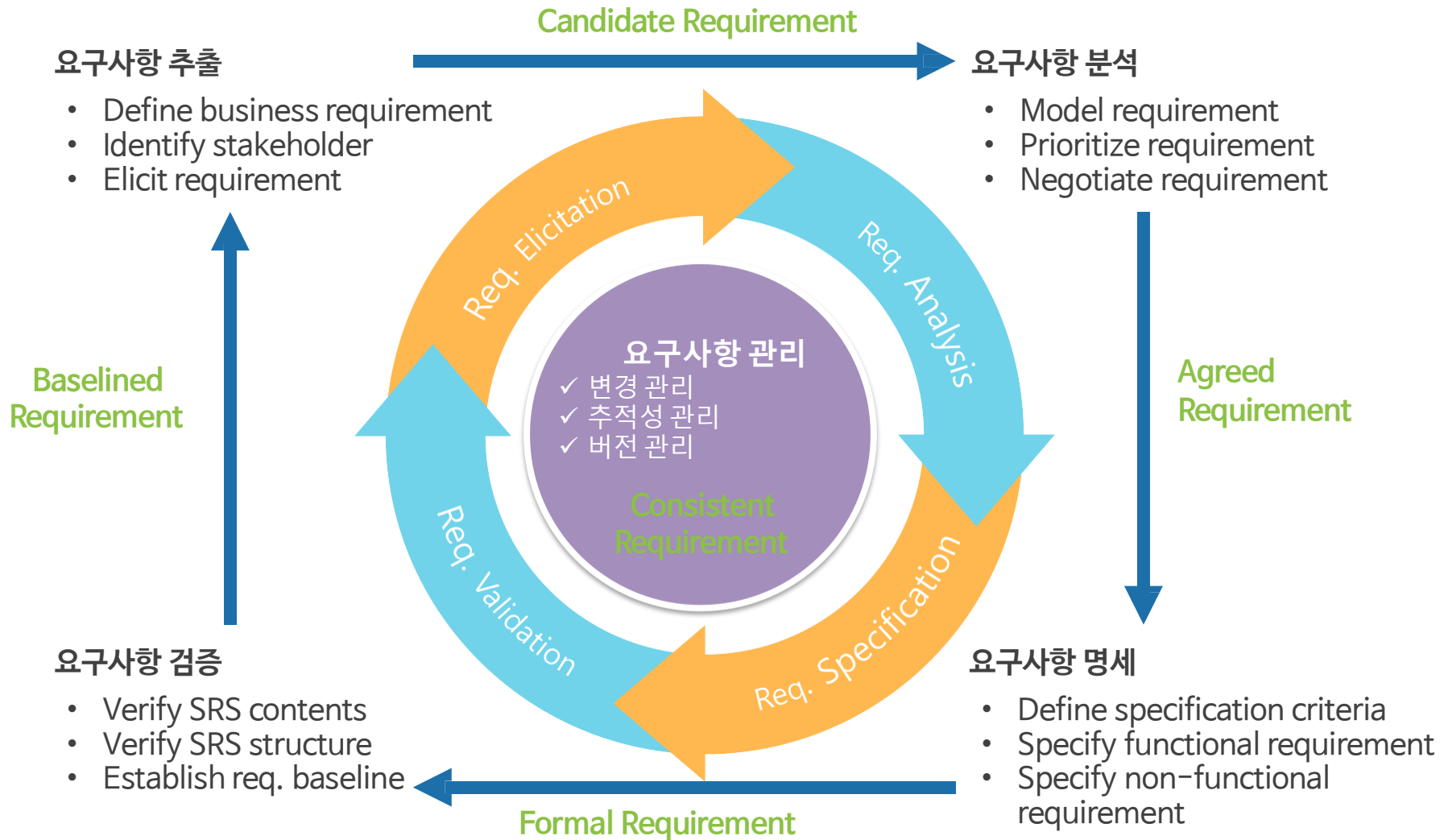
Maximize customer satisfaction Customer-Oriented Management

- 고객 만족의 기준 제공 SW 품질 및 신뢰성 확보

요구공학의 프로세스



요구사항 개발/관리 프로세스





[실습 #1]

가이드라인 없이 요구사항 개발하기

[실습] 가이드라인 없이 다음을 명세 (40분)

○ 예제 기능

- 전자식 차일드락 시스템

○ 양식

- 제공 양식 없음 (PPT 활용)

○ 명세 항목

- 이해관계자
- 이해관계자의 Needs
- 요구사항
 - 기능 요구사항
 - 비기능 요구사항
 - 반드시 만족해야 하는 요건, 반드시 수행하지 말아야 하는 요건
- 각 요구사항의 우선 순위



[실습 #2]

이해관계자 식별 가이드라인

요구사항 추출

○ 무엇을 개발할 것인지 개발 대상을 이해하고, 이해 관계자들이 기대하는 것(Needs)이 무엇인지 파악

- 정확한 요구를 파악하여 개발 대상(What)을 추출하는 단계
- 고객과 개발자간의 요구사항 이해에 대한 Gap을 줄이는 단계
- 개발에 대한 범위를 확정하는 단계
- 고객의 모호한 요구사항을 구체적으로 진화시키는 단계



Input	과거차 문제 분석서 필드 문제 분석서 대상 차종 컨셉 제어기 주요 기능 컨셉	차량/시스템 레벨 요구사항 개발 범위 식별	이해관계자 리스트	비즈니스 요구사항 이해관계자 리스트 요구사항 추출 방법
Activity	개발해야 할 시스템이 만족시킬 범위를 정의	개발 대상의 이해관계자들을 식별 이해관계자들의 우선 순위를 부여	이해관계자들의 특징이나 프로젝트 상황에 따라 요구사항을 추출하기 위한 방법 선택	이해관계자들의 기대사항(Needs)을 추출하여, 기대(Needs)의 근본 원인을 파악하여 이해관계자 요구사항으로 정의
R&R	사양개발 담당자	사양개발 담당자	사양개발 담당자	사양개발 담당자
Output	차량/시스템 레벨 요구사항 개발 범위 식별	이해관계자 분석 결과	요구사항 추출 방법	이해관계자 요구사항

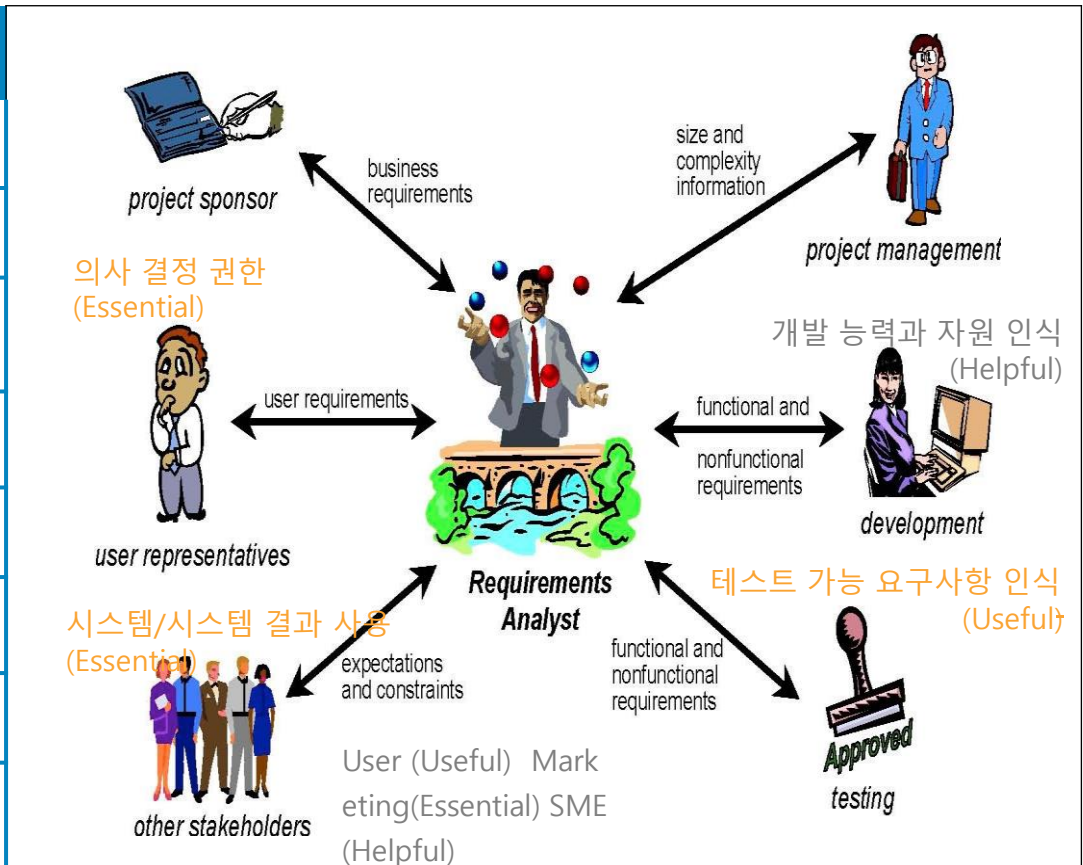
이해관계자 식별

○ 개발 대상의 이해관계자를 식별하고, 비즈니스 목표에 따라 순위를 부여

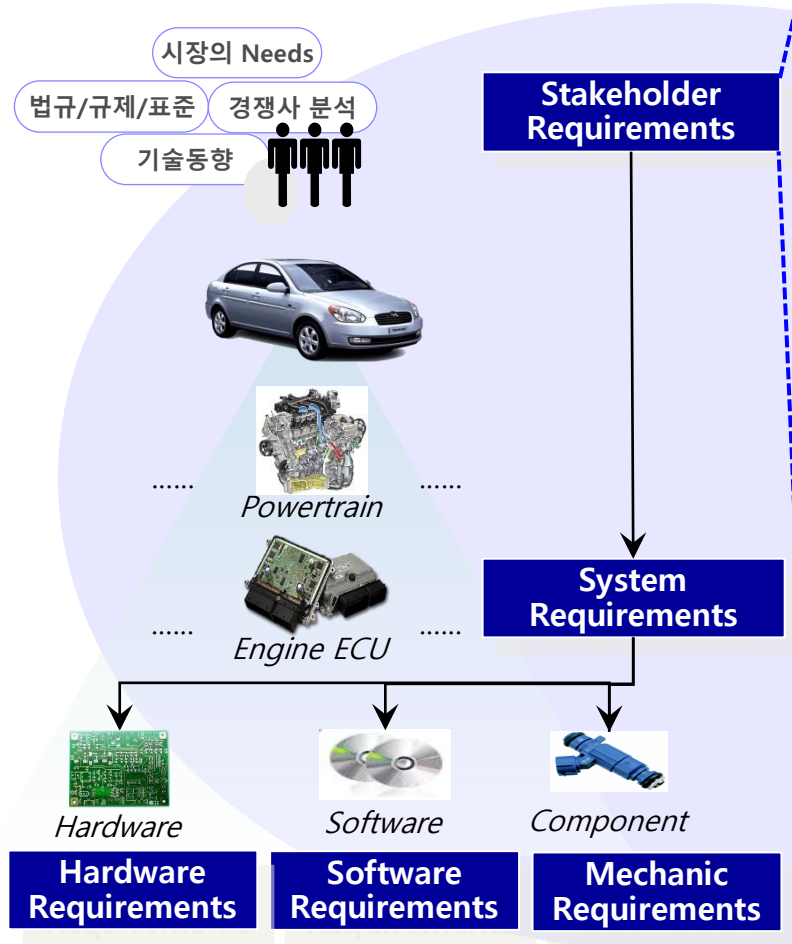
- 시스템 구축을 의뢰한 고객은 진정한 고객이 아니다. 고객의 고객이 진정한 고객이다.

이해관계자 식별을 위한 체크리스트

- 누가 시스템을 사용할 것인가?
- 누가 시스템을 구매하고 판매할 것인가?
- 시스템에 안전이나 품질 등의 영향을 받는 것들은 무엇인가?
- 시스템의 기능이나 성능에 제한을 주는 것은 무엇인가?
- 시스템의 경쟁상대는 무엇인가?
- 누가 시스템을 개발하고 검증하고 유지보수 할 것인가?
- 시스템은 어떤 환경에서 운영될 것인가?
- 시스템을 개발하기 위해서 어떤 기술이 필요 한가?



[예시] 이해관계자 식별



Type	Stakeholder	Needs of stakeholder
Global	Economic	Low Cost
	Market	High Technical etc...
	Technology	Technical Promotion, Technologies of standard, Generates of frame conditions
	Legal & Politics	Technical Characteristics
	Social & Cultural	Generates boundary conditions of product
Competition	Competitor	Strengthen market position, Technological superiority
	Potential Customer	Wants to buy
	Internal Supplier	Customer satisfaction
	External Supplier	Customer satisfaction
Operation	User & Customer	Safety satisfaction, Usability & Reliability, Quality & Functionality
	Pedestrians	Safety satisfaction
	Other Vehicles	Safety satisfaction, Wants to travel freely

[실습] 이해관계자 식별 가이드라인 (40분)

- 이해관계자란 무엇인가?
- 이해관계자 예시 제공
- 이해관계자를 어떻게 명세할 것인가?
- [양식]
 - 1. 이해관계자 식별
 - 이해관계자 식별 가이드라인 작성

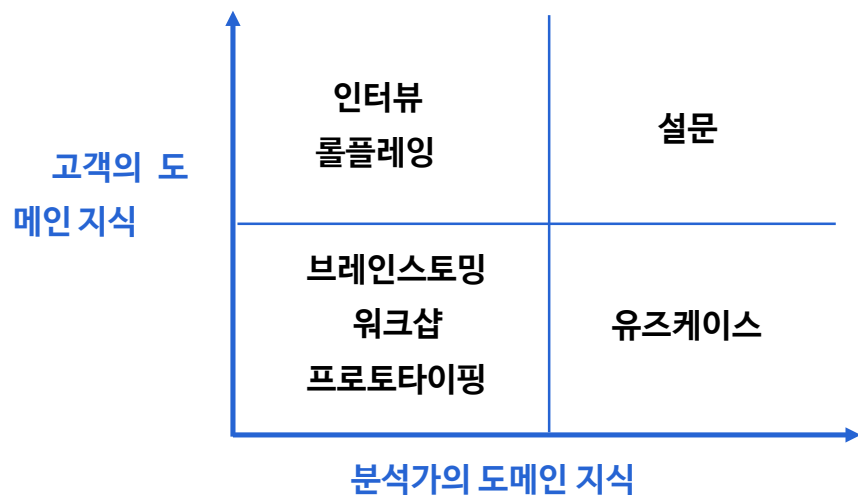


[실습 #3]

이해관계자 요구사항 추출

이해관계자 추출 기법 선택

- 이해관계자의 특징을 고려하여 효과적인 요구사항 추출 방법을 선택



[도메인 지식 정도에 따라]

이해관계자	요구사항 추출 방법
사용자	피드백, 설문, 클레임
경쟁 소프트웨어	벤치마킹, 특허 분석
잠재적 고객	협상
기술 지원, 세일즈 등 협력자	변경 요청, 문제(버그) 보고서, 피드백
외부 협력자	변경 요청, 설문
법규, 정책	전문가 설문, 분석
문화	전문가 설문, 분석
기술	기술 조사, 특허 분석, 표준 위원회

[이해관계자의 특징에 따라]

이해관계자 요구사항 추출

- 이해관계자들의 기대사항(Needs)을 추출하여, 기대(Needs)의 근본 원인을 파악하여 이해관계자 요구사항으로 정의

이해관계자	추출 방법	ID	이해당사자의 니즈	해당 니즈의 실질적 인 문제는 무엇인가?	이해당사자 요구사항은 무엇인가?	요구사항의 영향	우선 순위
서울, 2,30대 주부	Inter view	Needs 02	소형 건물의 경우 근처 까지만 안내가 가능하 니 주차장 입구를 찾느 라해맨 다	목표한 곳의 정확한 위치를 찾아주지 못한다.	가고자하는 곳의도 로 및 건물의 형태를 고려 하여 안내하여야 한다.	개전시, 효과는 높으나 이를 위해 투입 되어야 하는 오프라인 조사 인력의 투자가 필요하 고 주기적인 업데이트가 필요하다. 소프트웨 어 개발이 아닌 정책적인 문제가 함께 고려되 어야 한다.	+
		Needs 05	음성 인식 기능은 메뉴 단 계마다 음성으로 하고, 실 패하면 다시 처음부터 해 야 하니 그냥 터치 를 이용 한다.	네비게이션 음성 인 식 기능의 사용성이 떨어진다.	네비게이션 음성 인식 기능의 사용 편의성이 개선되어야 한다.	개발 비용 대비 사용자 만족도가 높은 항목 이다.	++

- 이전 버전 시스템의 문제를 수집/분석하여 중요한 문제점들을 식별하고, 식별된 문제의 근본 원인 을 분석하여 문제의 심각도를 정량화해서 신규 개발될 시스템에 반영되어야 할 내용을 이해관계자 요구 사항으로 정의함

이해관계자	니즈	근본 원인	요구사항의 영향	요구사항의 종류	요구사항	중요도	우선순위
Proble ms	주행중 엔진 이 정지되는 현상발 생	Camshaft Sensor 의 고장으로 제 기능 을 수행하지 못함		Functionality	Camshaft 고장이 발생할 경우엔, RAM 에 기억된 표준 값을 이용하여 엔진을 작동 시킴	Critical	

[실습] 이해관계자 요구사항 추출 가이드라인 (40분)

- 어떻게 이해관계자 요구사항을 추출할 것인가?
- 요구사항 추출 방법 예시
- 이해관계자 요구사항을 어떻게 명세할 것인가?
- 우선순위를 어떻게 선정할 것인가?
- [양식]
 - 2. 이해관계자 요구사항 추출
 - 이해관계자 요구사항 추출 가이드라인



[실습 #4]

전제사항 분석/명세

[참고] 이해관계자 요구사항 VS 시스템 요구사항

○ 이해관계자 요구사항

- 이해관계자가 제시한 문제(Problem) 혹은 요구(Needs)

○ 시스템 요구사항

- 이해관계자 요구사항을 개발자언어로 구체화하여 작성한 요구사항
- 명시적 요구사항과 묵시적 요구사항을 모두 포괄하여 요구사항을 작성

이해관계자 요구



“자동차 외부에서 원격으로 주차를 할 수 있으면 좋겠어”



스마트폰 앱을 통해 원격으로 주차를 수행할 수 있음

명시적
요구사항

승인된 운전자만 접근할 수 있어야 함(회원가입 혹은 스마트키와 연동)

주차지원시스템(PAS)과 연동하여 자동 주차를 수행할 수 있어야 함

묵시적
요구사항

실패한 프로젝트의 상당수가 **묵시적 요구사항에 대한 이해 정도가 달라서 발생**

요구사항 분석

○ 전제 조건이나 운용 관점의 요구사항, 기능 및 비기능 요구사항, 기술 관점의 요구사항 등을 분석하고 정의

- 개발 대상을 완성하기 위해 개발 대상의 범위와 기능 등의 요구사항을 정의
- 전제조건, 운용관점, 개발대상 관점, 기술관점 등 다각도에서 대상 분석
- 추출된 기능이 어떠한 흐름으로 동작하는지 파악하여 개발할 시스템을 명확히 하는 활동



Input	이해관계자 요구사항 차 량/시스템 레벨 요구사 항	시스템 범위	시스템 기능 & 비기능 요구 사항	시스템 메커니즘
Activity	이해관계자 요구사항과 상 위시스템 요구사항을 분석 하여 개발해야 할 시스템이 만족시켜야 할 시스템의 범 위를 정의	시스템의 Scope을 기반 으로 시스템의 기능 및 비기능 요구사항을 정의	시스템의 기능 및 비기능 요 구사항을 만족시키기 위한 기능적 & 동적 메커니즘을 정의	시스템 물리적 특성 및 기술적 제약사항 등 구현 기술 관점의 요구사항
R&R	요구사항 분석가	요구사항 분석가	System Engineer	System Engineer
Method	GAP Analysis & I mpact Analysis & Priority Analysis	Mind Mapping & Question Analysis	State/Sequence Diagram & Impact Analysis	Impact Analysis & System Context Diagram
Output	시스템 범위	시스템 기능 & 비기능 요구사항	시스템 메커니즘	제약 사항 및 구현 기술 요 구사항

전제조건 식별

- 상위 시스템(제품)이나 고객으로부터 전달받은 요구사항이나 제약사항 등이 개발 대상의 전제조건으로 정의되며, 반드시 만족시켜야 할 목표가 된다.

- ✓ 비즈니스 목표 및. 시스템에 탑재될 제품 목표
- ✓ 시장 동향 및. 경쟁업체 분석 내용
- ✓ 시스템 범위 및. 요구되는 주요 기능
- ✓ 국내/양산 지역 법규 및 규제, 사규 분석 내용
- ✓ 요구되는 기술 및. 특허 회피 방안
- ✓ 사용자 및 환경 요구사항
- ✓ 개발/양산/판매/유지보수/운영/폐기 관련 요구사항
- ✓ 과거 문제점 및. 개발 관련 요구사항
- ✓ 내/외부 협업 관계 및. 프로젝트 조직에 따른 요구사항
- ✓ 개발 프로세스 및 리소스, 인프라 관련 요구사항
- ✓ 개발 일정 및 주요 마일스톤
- ✓ 재사용 또는 상업용 컴포넌트(COTS)
- ✓ 프로젝트 통제 방법
- ✓ 교육 및 훈련
- ✓ 문서화 요구사항

전제조건	내용
적용 차종	'OO' 후속 차종
목표 연비	기존 차종 대비 OO% 향상
양산 목표	북미 20XX 내수 20XY
탑재 사양	중량 및 부피: 경쟁차 □□ 대비 우세 진동 소음: 00 DB 이하
제한 사항	'XX' 차종에 적용된 유사 시스템 활용
운영 방침	시스템 Fail시 안전성에 문제 없을 것 (ISO26262 기준 준수)
개발 비용	...
특허 회피 방안	...

[실습] 전제조건 분석/명세 가이드라인 (40분)

- 무엇이 전제조건이 되는가?
- 전제조건을 어떻게 명세할 것인가?
- 개발할 시스템의 범위는 어디까지인가?
- 전제조건 예시

- [양식]

- 3. 전제조건
- 전제조건 분석/명세 가이드라인



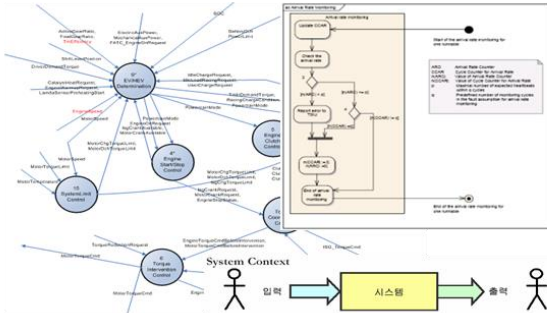
[실습 #5]

기능 요구사항 분석/명세

다양한 관점에서의 요구사항 분석 (1/2)

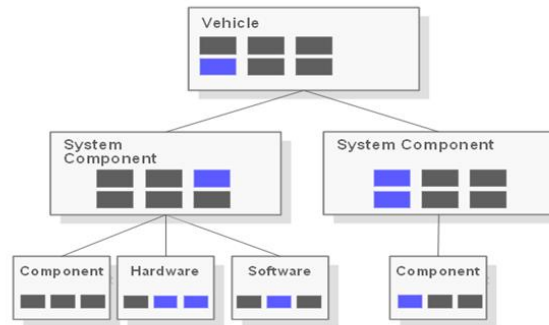
기능적인 관점

- 어떠한 기능이 필요한가?
 - Use-Case Diagram
 - DFD (Data Flow Diagram)
 - Process Diagram
- 어떠한 세부 기능들이 필요한가?
 - Sequence Diagram
 - Activity Diagram
 - Flow Chart
 - DFD (Data Flow Diagram)
 - Process Diagram



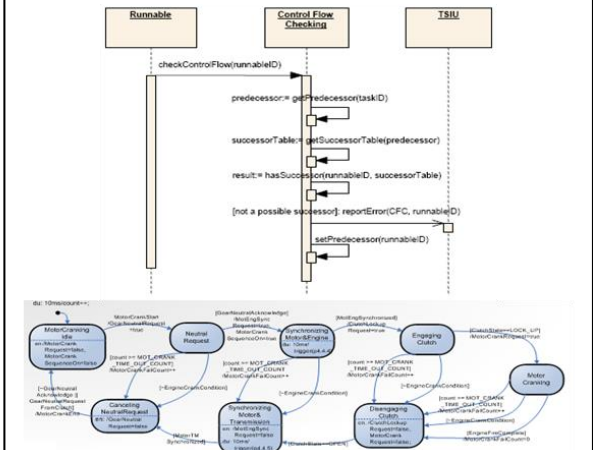
구조적인 관점

- 시스템의 구조는?
 - Class Diagram
 - Component Diagram
 - Deployment Diagram
 - ER-Diagram
 - Structure Chart



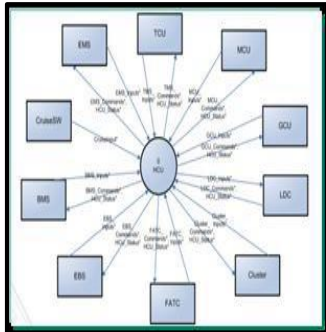
동적인 관점

- 기능은 어떻게 흘러가는가?
 - Sequence Diagram
 - State-Transition Diagram
 - Flow Chart



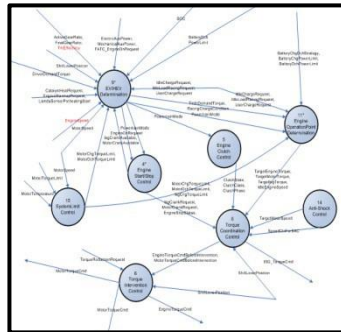
다양한 관점에서의 요구사항 분석 (2/2)

- 정의된 시스템 범위를 기반으로 시스템의 기능 및 비기능 요구사항을 정의함



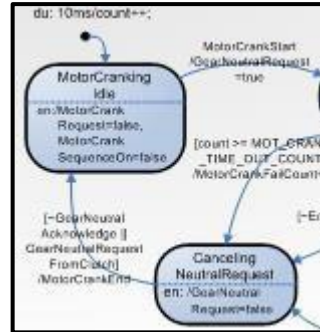
인터페이스 분석

- 시스템과 상호작용하는 외부 인터페이스 정의



구조적 기능분석

- 시스템 입력과 출력을 근거로 한 시스템의 기능을 구조적으로 분석



동적 기능 분석

- 시스템의 기능 간의 상호작용이나 시스템 상태 모드 등 시스템 동작 흐름 분석

ID	Name	Description	Status
1001	Vehicle Speed	Vehicle speed must be maintained within 10% of the setpoint.	Pass
1002	Engine RPM	Engine RPM must be maintained within 5% of the setpoint.	Pass
1003	Gear Shift	Gear shift must be completed within 10 seconds.	Pass
1004	Engine Torque	Engine torque must be maintained within 10% of the setpoint.	Pass
1005	Vehicle Acceleration	Vehicle acceleration must be maintained within 10% of the setpoint.	Pass

요구사항을 보조해주는 속성을 포함한 요구사항 정의

- 요구사항 타입 (Function, Reliability, ...)
- 요구사항 분류 (공통/특화 기능 등)
- 속성, 속성값, 속성의 단위
- 제약 사항
- 관련 자료나 정보
- 수행 시간

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?

Formal한 정도에 따른 명세 방법의 구분

Informal Notation	Semi-formal Notation	Formal Notation
<p>자연어(Natural Language)를 이용하여 서술형으로 기술된 요구사항</p> <ul style="list-style-type: none"> ✓ 자연어 기반의 서술, 작업흐름도 등의 그림 중심 작성 ✓ 작성 및 이해 용이, 사용자와 개발 팀간의 의사전달이 용이 ✓ 불충분한 명세, 일관성 결여, 내용의 모호성, 완전성 검증 곤란 	<p>SysML이나 UML등의 모델링 언어를 이용하여 기술된 요구사항</p> <ul style="list-style-type: none"> • 기능적 관점, 구조적 관점, 동적 관점으로 시스템을 표현 • 이해하기 용이하고 커뮤니케이션 오류 감소 • 누락되거나 중복되는 사항을 파악하기 용이함 • 요구사항 변경 영향을 파악하기 용이함 	<p>수학적 기호나 수식으로 기술된 요구사항</p> <ul style="list-style-type: none"> ✓ 모델기반 언어 : Z, VDM (Vienna Development Method) ✓ 대수처리 기반 언어 : CSP, CCS, LOTOS ✓ 명세의 정확성, 불완전성, 불일치 검토 입증, 모호성을 방지 ✓ Not everybody can read formal specs

ISO26262-8 : Specifying Safety Requirements

Methods		ASIL			
		A	B	C	D
1a	Informal notations for requirements specification	++	++	+	+
1b	Semi-formal notations for requirements specification	+	+	++	++
1c	Formal notations for requirements specification	+	+	+	+

Automotive Safety Integrity Level (ASIL)

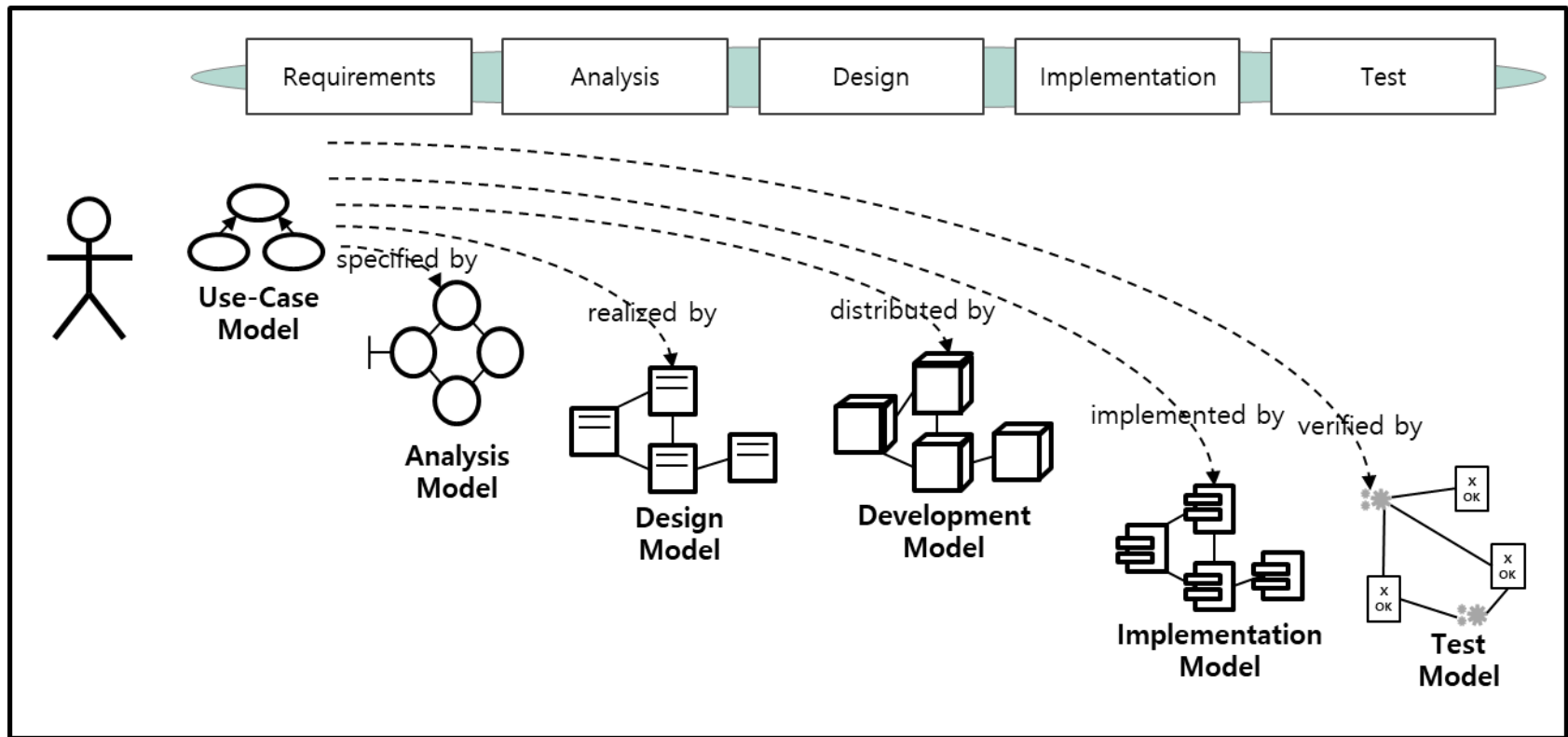


D등급이 안전이 중요한 프로세스와 엄격한 테스트 규제를 가지는 등급

UML(Unified Modeling Language) (1/2)

○ 소프트웨어 공학에서 시스템 설계를 표현하는데 사용하는 통합된 모델링 언어

- 개발하는 소프트웨어의 설계를 그래픽 모델링 언어를 이용해 작성하여 참여자인 설계자, 개발자, 분석가, 사용자의 의사소통 향상
- 다이어그램을 통해 개발 산출물을 가시화/명세화/문서화 함으로써, 요구사항의 누락사항이나 불일치를 줄임



UML(Unified Modeling Language) (2/2)

모델

- 추상화
- 실제 모습의 공통 특징을 표현한 것

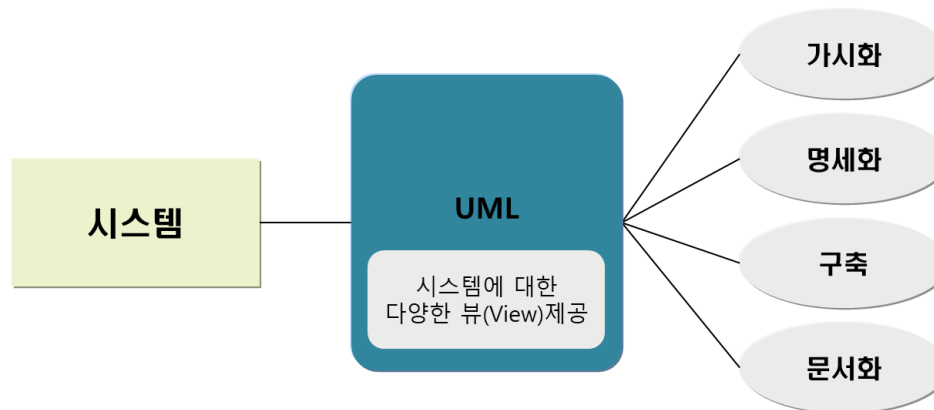
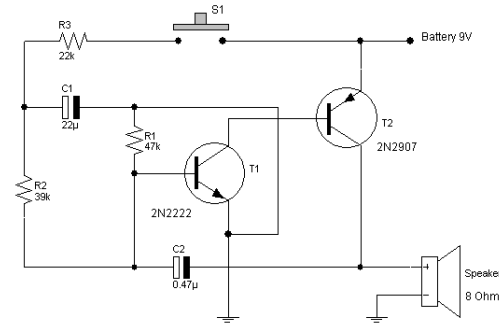
모델 사용 이유

- 복잡한 문제와 시나리오의 명확화
- 구현하기 전 설계를 현실에 더 가깝게 만들

다이어그램(Diagram)

- 모델을 표현하는 방법의 하나
- 기호, 선, 점 등을 사용해 각종 사상의 상호관계나 과정, 구조 등을 이해시키는 설명적인 그림

[전형적인 하드웨어의 모델]



[시스템 구축 시 UML의 역할]

UML 다이어그램 종류

- UML 다이어그램은 구조(Structure)와 행위(Behavior) 다이어그램으로 구분하며, 총 14개가 존재

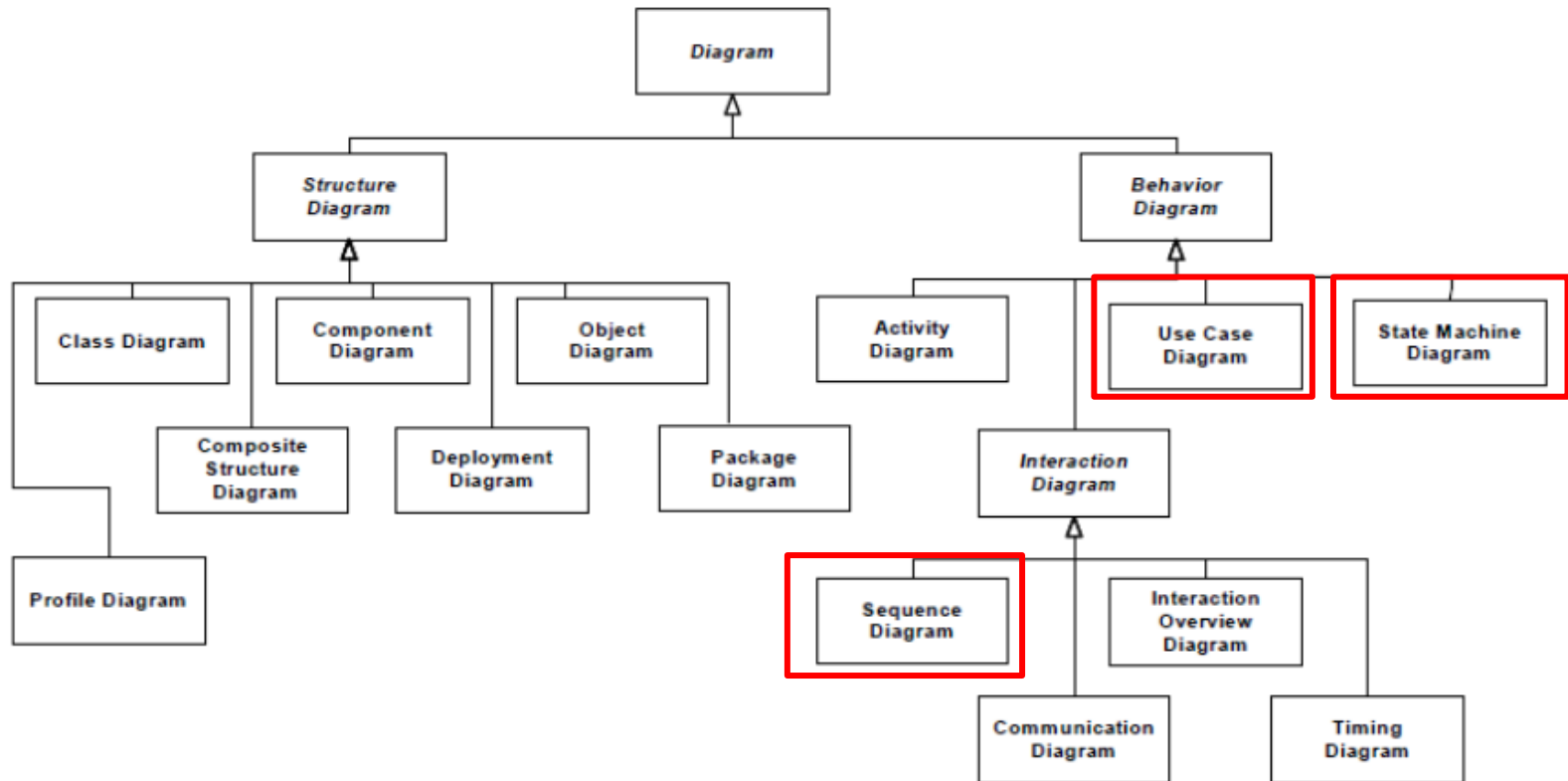


Figure A.5 The taxonomy of structure and behavior diagrams

Use Case 다이어그램

○ 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)

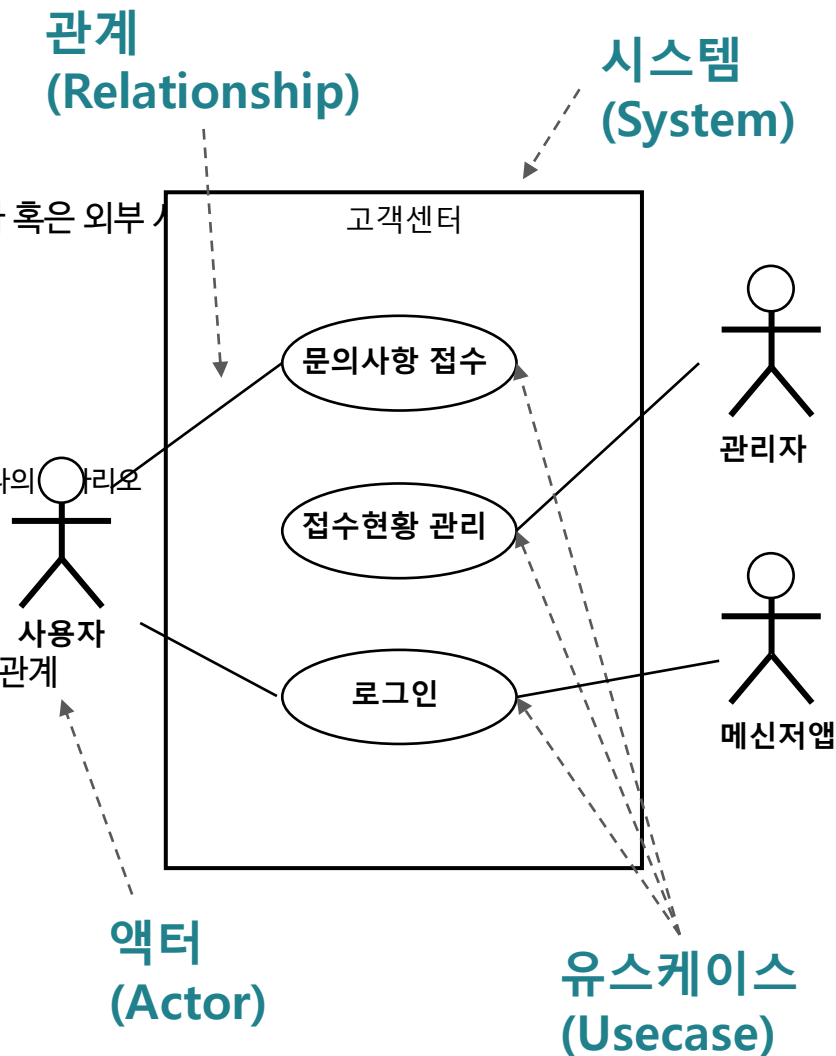
- 시스템의 외부에 있으면서 시스템과 상호작용을 하는 사용자 혹은 외부 시스템

○ 유스케이스(Usecase)

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

○ 관계(Relationship)

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



Usecase 다이어그램 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 유스케이스 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 다이어그램 : Actor 식별

○ Actor

- 시스템과 상호 작용하는 시스템 외부의 시스템/사용자

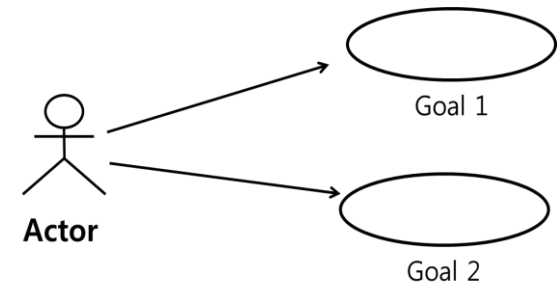
○ Actor를 식별하기 위한 질문

- 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 다이어그램 : Use-case 식별

○ Use-case

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



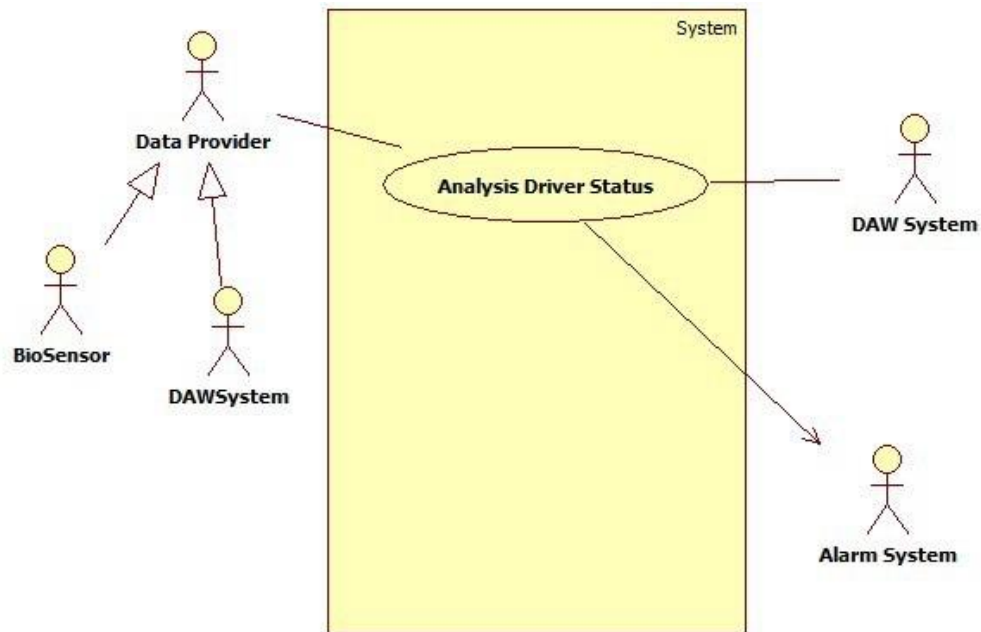
○ Use-case 식별

- 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 다이어그램



Use Case 명세서

- 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

시나리오(Scenario)

- 결과를 달성하기 위한 사건의 흐름

- 정상 흐름(Normal Flow) - Success

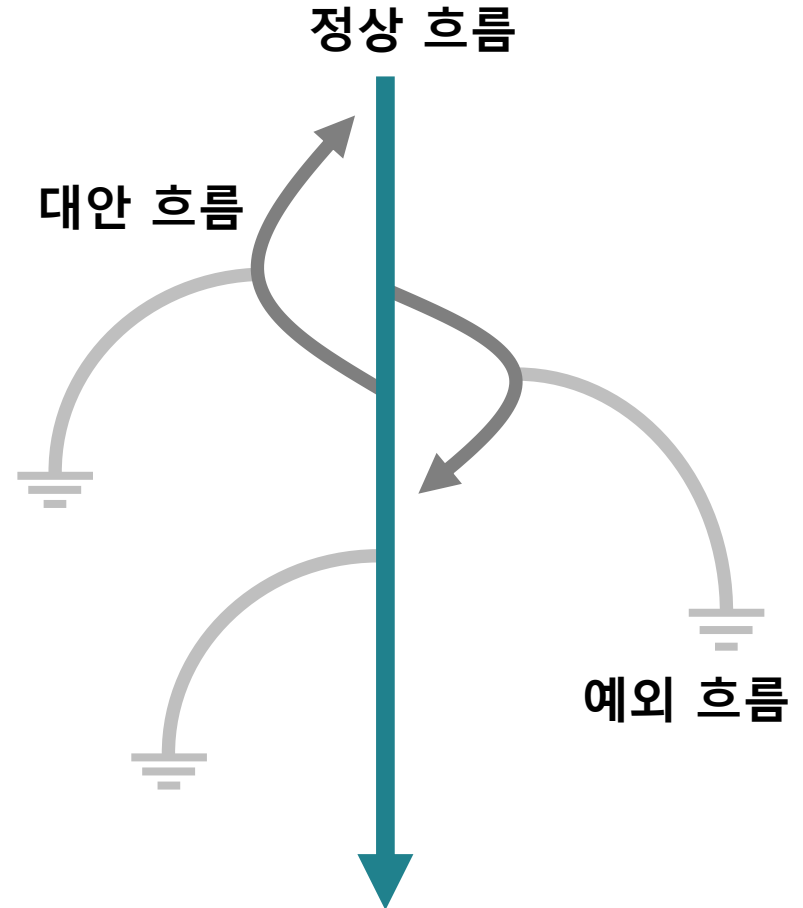
- 정상적으로 수행되는 흐름

- 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



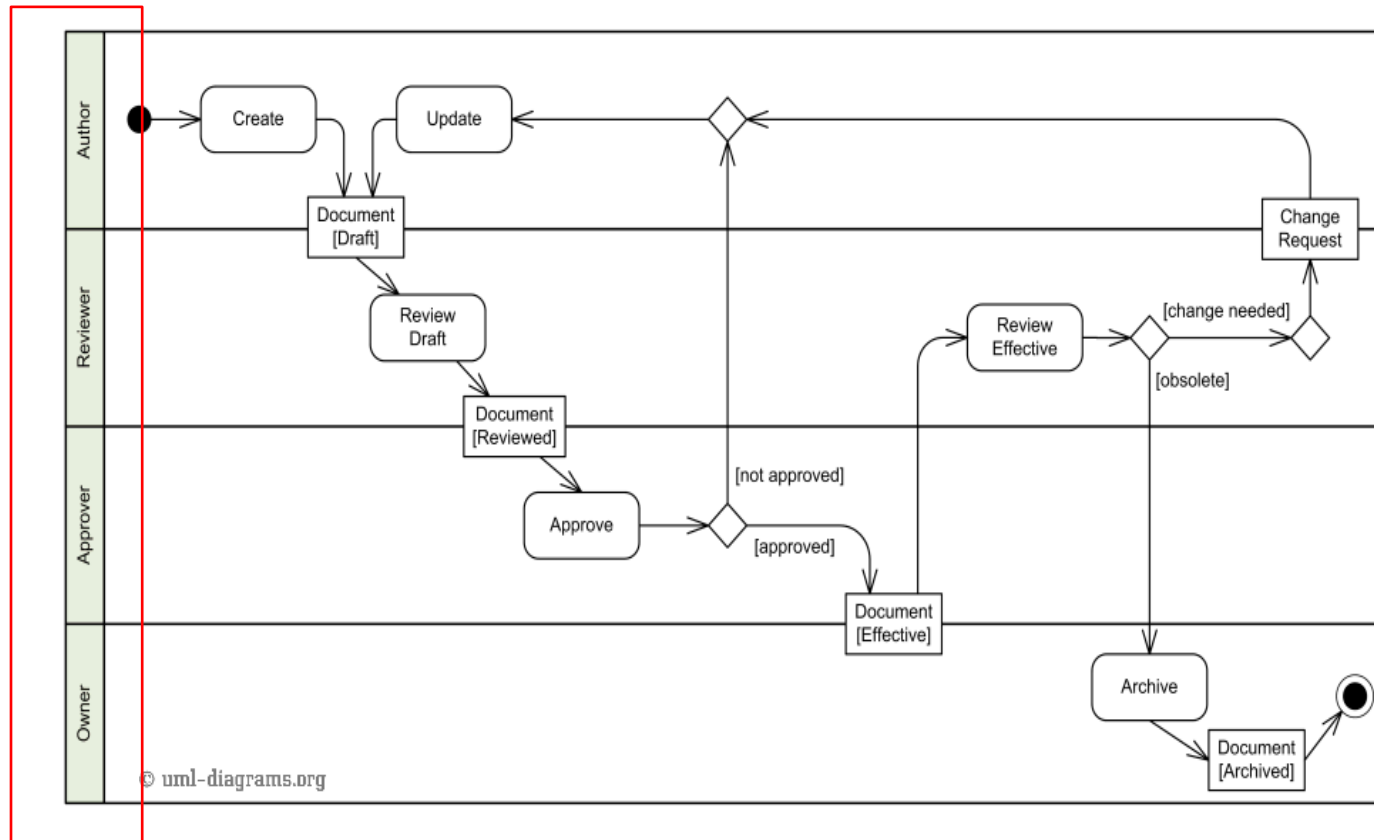
[예제] Use Case 명세서

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>

Activity 다이어그램

○ 사용 목적

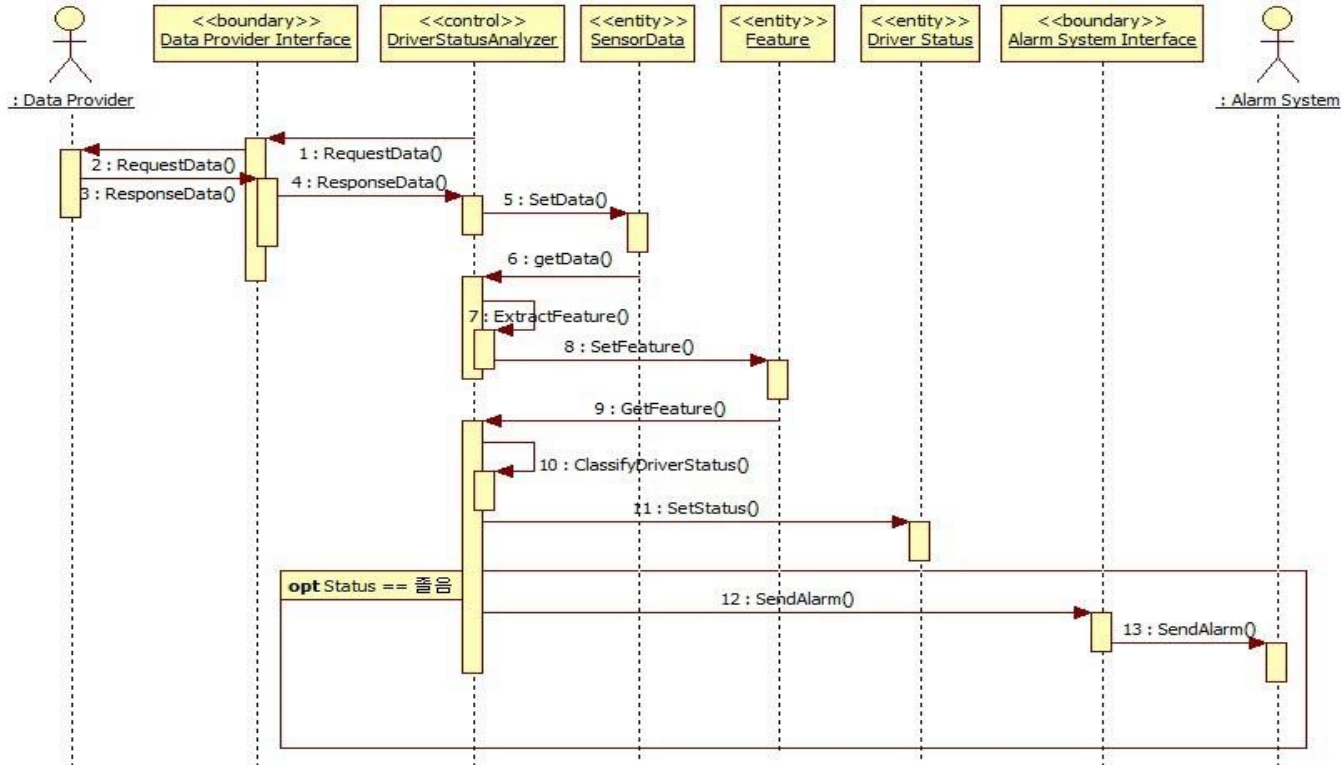
- 시간에 따라 발생하는 객체 간의 활동의 흐름을 강조
- 문장으로 작성하기에 복잡한 흐름을 시각적으로 표현
 - 예. 복잡한 Usecase나 업무 프로세스
 - 로직 검토에 유용



Sequence 다이어그램

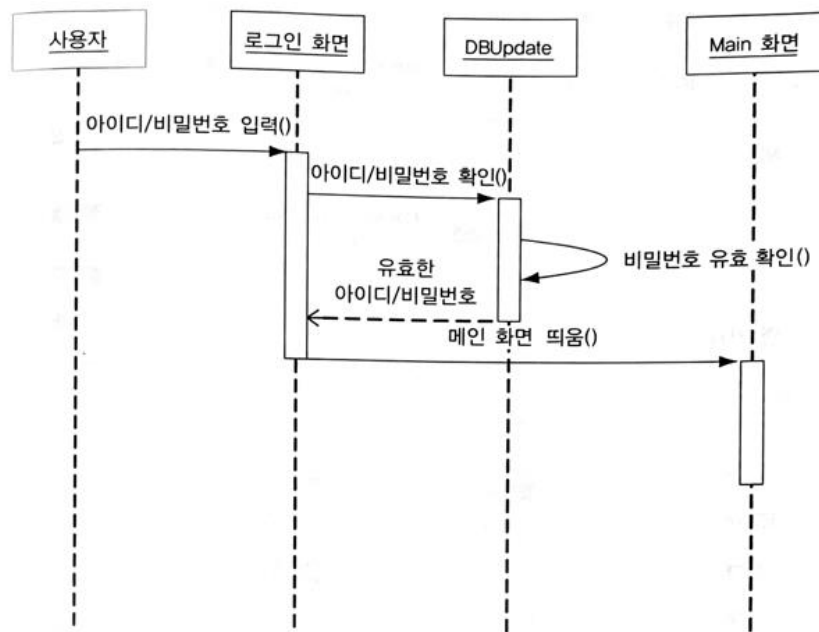
○ 사용 목적

- 시간 진행에 따른 UML 요소(Actor, 객체, 서브 시스템 등) 간 메시지 순서를 강조
- 같은 Context에서 요소들이 만들어내는 하나의 시나리오 실행 순서를 추적하는데 사용
 - **Usecase에서 시스템의 구성 요소들이 각 Usecase의 목표를 달성하기 위해 사용**
 - 구성 요소의 인터페이스로부터 받은 요청을 처리하기 위한 내부 구성 요소 간 상호 작용 방식을 이해하기 위해 사용



[예제] Sequence Diagram

UC-1	로그인 및 로그아웃 - 메시지를 사용하기 위해 로그인을 하거나, 메시지 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 ...



[실습] 기능 요구사항 분석/명세 가이드라인 (40분)

- 기능 요구사항은 무엇인가?
- 기능 요구사항을 어떻게 명세할 것인가?
 - 어떤 다이어그램을 사용할 것인가?
 - 어떤 양식의 표를 사용할 것인가?
- 기능 요구사항 예시
- [양식]
 - 4. 기능 요구사항
 - 기능 요구사항 분석/명세 가이드라인



[실습 #6]

비기능 요구사항 분석/명세

비기능 요구사항 (Non-Functional Requirement)

- *How well (much) the software does?*
- 제품의 기능이 성능, 안전성, 사용성 등의 품질 기준을 만족시키기 위해 가져야 하는 속성
 - 시스템이나 소프트웨어가 얼마나 사용자에게 만족을 느끼게 하는가와 연관이 있음
- 품질 요구사항
 - 시스템이나 소프트웨어가 무엇(What) 을 얼마나(How much) 해야 하는지에 대해 정의한 요구사항

소프트웨어 제품 품질 특성

- 소프트웨어 제품 품질이 가져야하는 세부 속성
- ISO 25010 (과거 ISO 9126)에 소프트웨어 품질 특성 정의

▪ 기 능성	요구되는 기능을 만족 시키는 능력
▪ 신 뢰성	규정된 환경에서 결함 없이 의도된 기능 및 작업을 수행하는 능력
▪ 사 용성	사용자가 이해하고 배우기 쉬운 정도
▪ 효 율성	적절한 자원의 사용 및 적절한 반응시간 정도
▪ 유 지보수성	소프트웨어의 수정 및 변경의 용이성
▪ 이 식성	지원하는 다양한 환경에서 운영될 수 있는 능력
▪ 상 호운영성	다른 시스템과의 상호 연동 능력
▪ 보 안성	정보 및 데이터를 보호하는 능력

소프트웨어 제품 품질 특성 - 부특성 (1)

주특성	부특성	내용
기능성	기능 성숙도 Functional Completeness	명시된 요구사항 구현 정도
	기능 정확도 Functional Correctness	정의된 정밀도에 따라 정확하게 결과를 제공하는 정도
	기능 타당성 Functional Appropriateness	사용자의 목적 달성에 소프트웨어가 도움을 주는 정도
효율성	시간 반응성 Time-behavior	기능 수행 시 응답, 처리 시간과 처리율이 요구사항을 충족시키는 정도
	요소 활용 Resource Utilization	기능 수행 시, 사용되는 자원의 유형 및 양이 요구사항을 만족 시키는 정도
	기억 용량 Capacity	제품 혹은 시스템 파라미터(최근 사용자 수, 통신 대역폭, 데이터베이스가 저장할 수 있는 데이터 양 등)의 최대 한계가 요구사항을 만족시키는 정도
상호 운영성	공존성 Co-existence	다른 소프트웨어에 유해한 영향을 주지 않고 환경 및 자원을 공유하면서 요구된 기능을 효과적으로 수행하는 정도
	상호 운영성 Interoperability	둘 혹은 그 이상의 시스템, 제품 혹은 구성요소가 정보를 교환하거나 교환된 정보를 이상 없이 사용할 수 있는 정도

소프트웨어 제품 품질 특성 - 부특성 (2)


주특성	부특성	내용
사용성	타당성 식별력 Appropriateness recognisability	사용자의 요구에 적절한 기능인지 식별할 수 있는 정도
	학습성 Learnability	사용자가 소프트웨어의 사용법을 배워 명시된 목적을 달성할 수 있는 정도
	운용성 Operability	제품 혹은 시스템이 작동 및 제어를 쉽게 할 수 있는 정도
	사용자 오류 보호 User error protection	소프트웨어가 발생한 오류로부터 사용자를 보호하는 정도 (버튼 비활성화, 알림 창 등)
	사용자 인터페이스 미학 User interface aesthetics	사용자 인터페이스가 사용자에게 만족스러운 정도
	접근성 Accessibility	연령과 장애에 관계없이 사용될 수 있는 정도
신뢰성	성숙성 Maturity	소프트웨어 구성요소가 표준적 환경에서 신뢰도 요구를 충족 시키는 정도
	가용성 Availability	사용자가 원하는 시간에 사용 및 접근이 가능한 정도
	결점 완화 Fault tolerance	시스템, 제품 및 구성요소가 하드웨어 혹은 소프트웨어에 결함이 존재하더라도 이를 극복하고 의도한대로 작동해야 함
	회복 가능성 Recoverability	중단 및 실패 발생 시, 제품 혹은 시스템이 데이터를 복구할 수 있는 정도

소프트웨어 제품 품질 특성 - 부특성 (3)

주특성	부특성	내용
보안성	기밀성 Confidentiality	제품 혹은 시스템은 반드시 권한이 있는 데이터에만 접근 가능하도록 해야 함
	무결성 Integrity	시스템, 제품 혹은 구성요소가 컴퓨터 프로그램 혹은 데이터에 대해 무단으로 접근 혹은 변경되는 것을 방지하는 정도
	부인 방지 Non-repudiation	사건 및 행위 후에 부인하지 못하도록 행동 및 사건에 대해 입증되는 정도
	책임성 Accountability	시스템 내의 각 개인을 유일하게 식별하여 언제 어떠한 행동을 하였는지 기록하여 필요 시 그 행위자를 추적할 수 있는 능력
	진본성(인증성) Authenticity	사건 및 행동에 대해 행위자임을 증명할 수 있는 능력
유지 보수성	모듈성 Modularity	최소의 영향을 가진 개별 구성요소로 구성된 정도
	재사용성 Reusability	자신이 하나 이상의 시스템에서 사용될 수 있고, 기타 자신을 구축할 수 있는 정도
	분석성 Analyzability	시스템 변화에 대해 어떠한 영향을 받는지 평가 할 수 있는 보고서를 제공하는 정도
	수정 가능성 Modifiability	제품 혹은 시스템이 장애 없이 효과적이고 효율적으로 수정될 수 있는 정도
	시험 가능성 Testability	제품 사용전, 사용에 필요한 검증 기능 제공 여부
이식성	적용성 Adaptability	제품 혹은 시스템이 다른 하드웨어, 소프트웨어 혹은 기타 사용 환경에 효과적이고 효율적으로 적용될 수 있는 정도
	설치성 Installability	제품 또는 시스템이 성공적으로 설치 및 제거될 수 있는 정도
	대치성 Replaceability	제품이 동일한 환경에서 동일한 목적을 위해 다른 지정 소프트웨어 제품으로 대체될 수 있는 정도

[실습] 비기능 요구사항 분석/명세 가이드라인 (40분)

- 비기능 요구사항은 무엇인가?
 - 기능 요구사항과 어떻게 구분할 것인가?
- 비기능 요구사항을 어떻게 명세할 것인가?
 - 필수 비기능 요구사항은 무엇인가?
 - 비기능 요구사항 예시
- [양식]
 - 5. 비기능 요구사항
 - 비기능 요구사항 분석/명세 가이드라인



[실습 #7] 제약사항 분석/명세

제약 사항

- 제한사항과 제약사항을 합쳐, 일반적으로 제약사항으로 정의
- 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

제약 사항(Constraints)

○ Solution constraints

- 구현 기술에 대한 제약
- How에 대한 제약, Architecture mechanism, design decision, implementation technique 등의 engineering decision

○ Boundary constraints

- 시스템 경계에 대한 제약
- 개발되는 시스템의 외부 요건이나 내부 인터페이스에 의해 시스템의 기능이나 비기능에 영향

○ Safety constraints

- Safety(안전성)에 대한 제약
- 안정과 관련한 solution constraints. Requirement interlocks, Physical barriers around moving parts, Safeguards concerning electricity, Toxic chemicals 등
- 기능 안전 전장품의 경우, 기능 안전 요구사항으로 별도 정의

제한 사항(Limitations)

- Company & System Vision
 - 회사 및 시스템의 비전
- Market
 - 출시하고자 하는 마켓의 제약사항(Time-to-Market, etc.)
- Competitor
 - 경쟁 업체의 동향 및 특허 등에 의한 제약사항
- Resource
 - 비용, 예산, 일정 등에 의한 제약 사항
- Standards
 - 산업 표준에 의한 제약 사항
- Cultural & Political
 - 문화나 정치에 의한 제약사항
- Regulations
 - 법률이나 사규에 의한 제약사항

[실습] 제약사항 분석/명세 가이드라인 (40분)

- 제약사항은 무엇인가?
 - 우리가 반드시 지켜야 하는 제한사항은 무엇인가?
 - 제약사항은 무엇인가?
- 제약사항을 어떻게 명세할 것인가?
 - 제약사항 예시
- [양식]
 - 6. 제약사항
 - 제약사항 분석/명세 가이드라인



[실습 #8]

요구사항 우선순위 선정

우선순위(1/4)

- 엔지니어나 프로젝트 관리자는 일정, 예산, 리소스, 품질 목표 등의 제약사항들을 고려하여 프로젝트 범위를 정의
- 엔지니어나 프로젝트 관리자는 우선 순위가 높은 요구사항을 정의하여, 우선 순위에 따라서 업무를 진행

최저비용으로 최대 제품가치 제공을 위한 구현 순서 부여

- 구현 일정 : “기능이 어떤 increment에서 구현될 것인가”
- 비즈니스 니즈와 가장 중요한 요구사항에 초점

- 이해관계자들 간에 중요한 요구사항이 다를 경우, 프로젝트 관리자는 Trade-off decision을 정의

요구사항의 상대적 필요성 결정 : 필수/선택 요구사항

- 불필요한 “잠재성 요구사항(nice-to-haves)” 제거
- 개인 선호도, 비즈니스 가치, 구현 비용, 위험 등에 따라

우선순위(2/4)

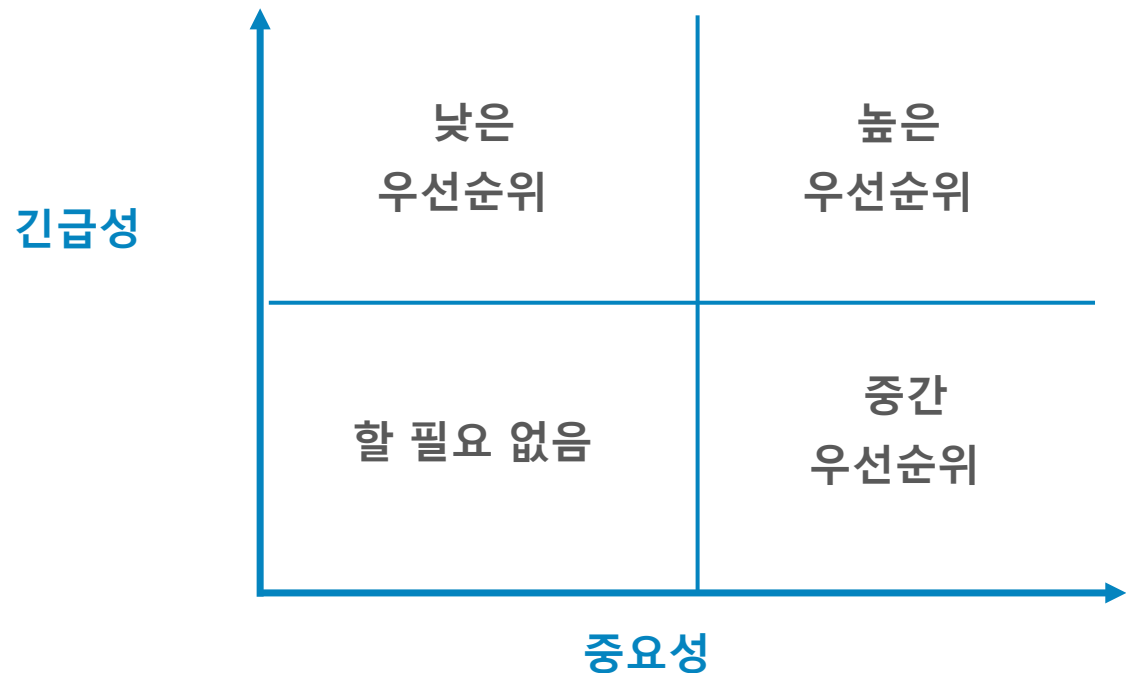
○ 우선 순위 결정의 어려움

- Mandatory nature of requirements : 모든 요구사항이 요구 (필수적: 85-90%)
- Large number of requirements : 그룹화 (주요 기능, 다음 릴리스 요구사항 한정)
- Limited resources : 자원 사용량의 결정 어려움
- Quality requirements : 품질 요구사항의 낮은 우선순위, 미기술, 모호성
- Goals vs. requirements : 비즈니스 목표 중요성과 우선순위의 직접적 관련 어려움
- Changing priorities : 비즈니스 환경, 이해관계자, 비용과 일정, 증진적 개발, 추가
- Incompatible priorities : 경험/훈련 정도에 따라 이해관계자들이 다른 순위 부여
- Incompatible requirements : 한 요구사항 응락이 다른 요구사항 구현 문제 발생
- Lack of trust : 즉각적으로 구현을 요구하는 고객의 잘못된 가정
- Non-requirements : 초기 요구사항-실제 요구사항이 아닌 wish-list
- Subjective prioritization : 주관적, 편향적, 정책 영향 - 우선순위 이유 무시
- Consequences of poor prioritization : 이해관계자 불만, 비용 문제

우선순위(3/4)

○ 우선 순위 결정 방법

- Yes/No 방법
- 100 달러 방법
- MoSCow 방법 : Must(필수) / Should / Could(시간/자원) / Won't(향후)
- **중요성과 긴급성**
- VOP (Value-Oriented Prioritization)
- AHP (Analytic Hierarchy Process)



우선순위(4/4)

○ 품질 요구사항 우선순위

- 품질 요구사항에 의한 충돌은, 요구사항 단계가 아닌 아키텍처 수립 시 발생 여부를 정확히 판단할 수 있으며,
- 그 때, 요구사항 분석 시 결정된 우선 순위에 의해서 어떤 품질 요소를 우선 만족 시킬 것인지를 결정할 수 있음

	Availability	Efficiency	Flexibility	Integrity	Interoperability	Maintainability	Portability	Reliability	Reusability	Robustness	Testability	Usability
Availability								+		+		
Efficiency			-		-	-	-	-		-	-	-
Flexibility		-		-		+	+	+		+		
Integrity		-			-				-		-	-
Interoperability		-	+	-			+					
Maintainability	+	-	+					+			+	
Portability		-	+		+	-			+		+	-
Reliability	+	-	+			+				+	+	+
Reusability		-	+	-				-			+	
Robustness	+	-						+				+
Testability	+	-	+			+		+				+
Usability		-								+	-	

[실습] 우선순위 선정 가이드라인 (40분)

○ 우선순위는 무엇인가?

- 우선 순위를 선정하는 기준은?
- 반드시 상위 등급에 포함되어야 하는 요구사항이 있는가?
- Nice to have에 해당하는 요구사항이 있는가?
- 우선 순위를 표현하는 방법은? (3점 척도 등)

○ 우선순위를 어떻게 명세할 것인가?

○ [양식]

- 4~6. 항목에 작성
- 우선순위 선정 가이드라인



[실습 #9]

검증 체크리스트 개발

좋은 체크리스트는?

- **한 페이지를 넘지 않는 체크리스트**
 - 10~12개 문항이 적절
- **한 질문에 한 항목을 평가**
 - “~을 작성하고, ~을 하였는가?” 와 같이 여러 가지의 질문 자체
- **명확한 문장으로 질문**
 - 적절히, 적당히, 많이, 빠르게 등의 사용 자체
- **체크리스트 문항을 명령문으로 바꾸었을 때, 자연스러워야 함**
 - 체크리스트 문항이 애매한 경우, 명령문이 자연스럽지 않음
 - 예) 수행했을때 체크인가, 수행하지 않으면 체크인가?

[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?

실현 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?

[실습] 요구사항 검증 체크리스트 (40분)

- 무엇을 검증할 것인가?
- [양식]
 - 7. 체크리스트 항목에 작성



감사합니다.