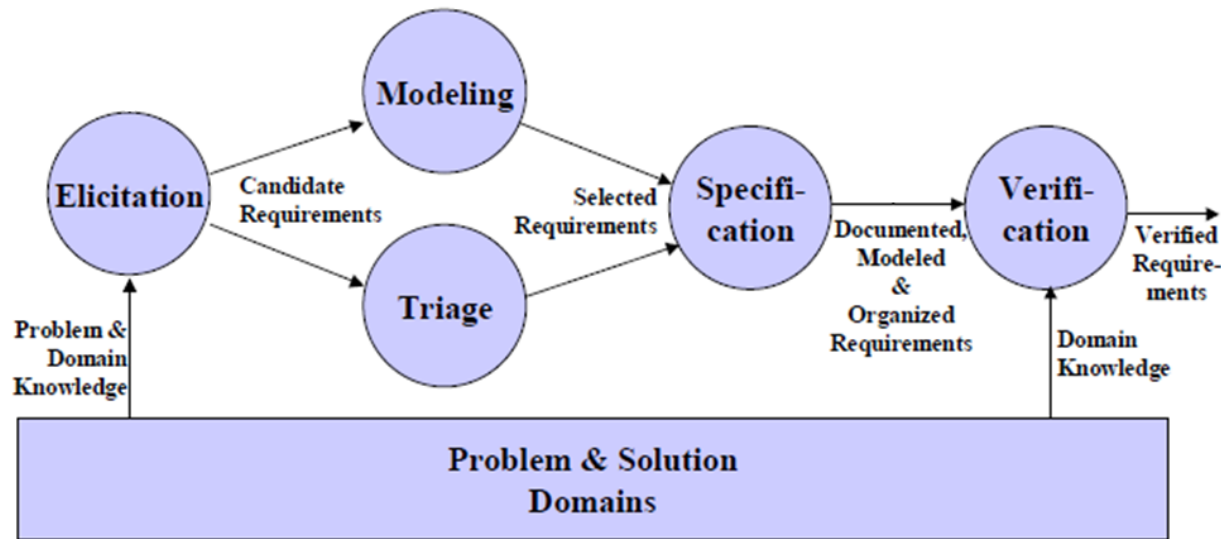


Chaper 1

요구공학 개요

강좌 구성

SW 요구분석 전문가가 갖추어야 할 핵심이론과 실무기법을 전달하는 세션으로 다음 사항을 이론과 실습을 통해 학습한다



- 요구사항 추출과 명세: Interview, Use case, User story 등
- 요구모델링: OO modeling, State modeling, Formal modeling 등
- 요구 우선순위화 : Value-oriented prioritization 기법
- 품질 요구: Quality Attribute Scenario, ATAM 기법
- 요구 명세 및 검증: Specification criteria, Review & Verification

강좌 구성

	주제	시간	세부주제		주제	시간	세부주제
Day 1	Introduction	2	Requirement Overview Practice #1	Day 2	Requirement Modeling	2	Usecase Modeling & Evaluation
	Requirement Elicitation	2	Requirement Elicitation		Requirement Prioritization	1	Requirement Prioritization
		1	Practice #2 Team Practice: Elicitation			1	Requirement Prioritization Practice #4
	Requirement Analysis	2	Requirement Analysis		Quality Requirement & Specification	3	Quality Requirement Individual Practice #5
		1	Team Practice #3: Modeling & Analysis			1	Requirement Specification & Verification

목 차

Chapter 1. 요구공학 개요

- 프로젝트의 성공과 실패
- 오류와 요구
- 요구의 구분
- 요구개발의 단계

1. 프로젝트의 실패와 성공

Project의 성공기준

TRADITIONAL RESOLUTION

- OnTime
- OnBudget
- OnTarget (scope)

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

MODERN RESOLUTION

- OnTime
- OnBudget
- On Target (scope)
- On Goal
- Value
- User Satisfaction

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

1. 프로젝트의 실패와 성공

Project Failure Profiles

Project Challenged Factors	% of Responses
1.	12.8%
2.	12.3%
3.	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

Project Impaired Factors	% of Responses
1.	13.1%
2.	12.4%
3. Lack of Resources	10.6%
4.	9.9%
5. Lack of Executive Support	9.3%
6.	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%
Other	9.9%

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%



2. 요구오류와 에러

Error/Mistake, Fault, Failure의 예

Error/Mistake

- Incorrect usage of software **by users**
- Bad design **by architects**
- Bad programming **by developers**
- Inadequate testing **by testers**
- Incorrect configuration **by build members**

Fault

- An incorrect **statement**
- Wrong **data type**
- Wrong formula **in design document**
- Missing functionality **in the system**

유입의 순서

Failure

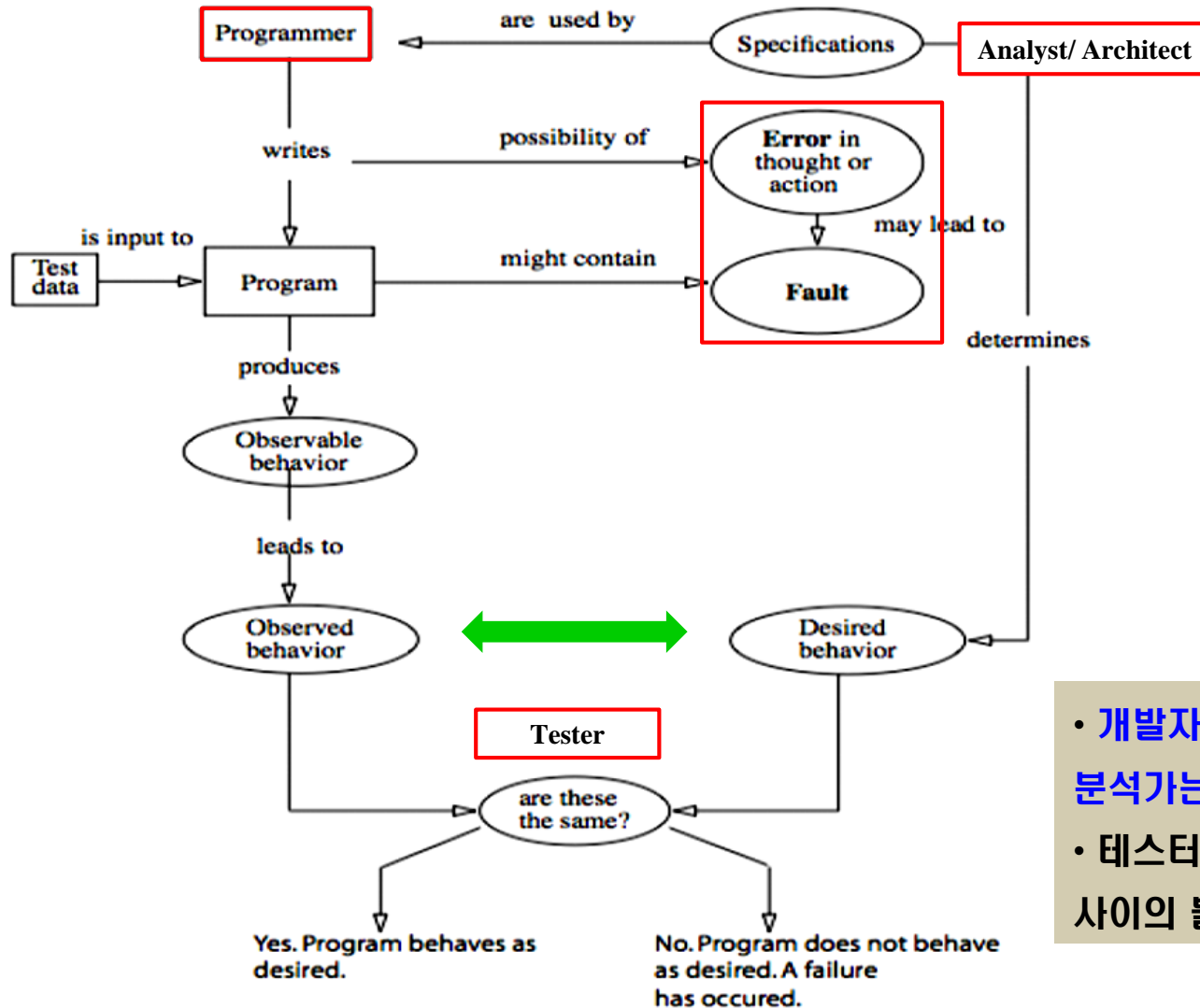
- Faulty piece of code is executed leading to **incorrect state** that propagates to the program's output.

발견의 순서

← 개선의 포인트는?

2. 요구오류와 에러

Observed behavior와 Desired behavior



- 개발자는 observed behavior를, 분석가는 desired behavior를 만듦
- 테스터는 테스트를 통해 두 결과물 사이의 불일치를 발견함

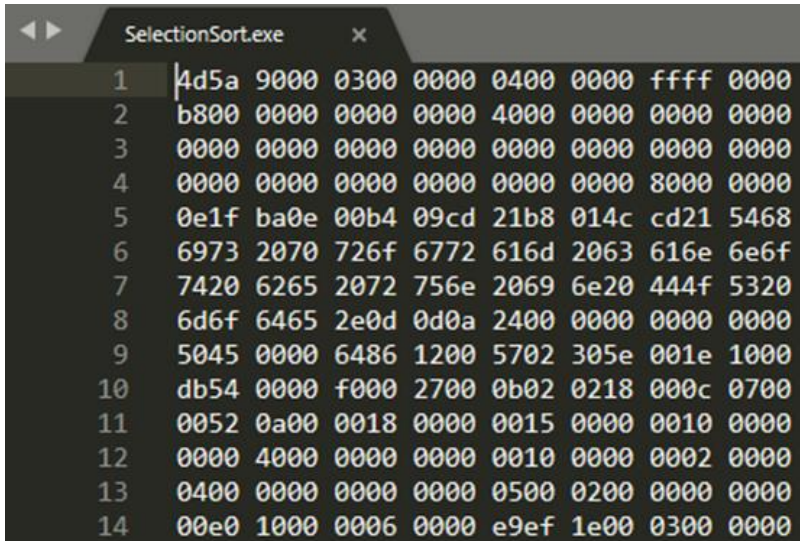
2. 요구오류와 에러

Observed behavior와 Desired behavior

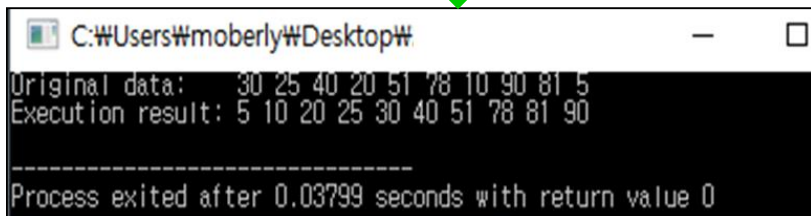
specification

Find the list[o..n-1], where
 $\text{list}[0] \leq \text{list}[1] \leq \dots \leq \text{list}[n-2] \leq \text{list}[n-1]$

observed behavior



1	4d5a	9000	0300	0000	0400	0000	ffff	0000
2	b800	0000	0000	0000	4000	0000	0000	0000
3	0000	0000	0000	0000	0000	0000	0000	0000
4	0000	0000	0000	0000	0000	0000	8000	0000
5	0e1f	ba0e	00b4	09cd	21b8	014c	cd21	5468
6	6973	2070	726f	6772	616d	2063	616e	6e6f
7	7420	6265	2072	756e	2069	6e20	444f	5320
8	6d6f	6465	2e0d	0d0a	2400	0000	0000	0000
9	5045	0000	6486	1200	5702	305e	001e	1000
10	db54	0000	f000	2700	0b02	0218	000c	0700
11	0052	0a00	0018	0000	0015	0000	0010	0000
12	0000	4000	0000	0000	0010	0000	0002	0000
13	0400	0000	0000	0000	0500	0200	0000	0000
14	00e0	1000	0006	0000	e9ef	1e00	0300	0000



```
C:\Users\moberly\Desktop#\>
Original data: 30 25 40 20 51 78 10 90 81 5
Execution result: 5 10 20 25 30 40 51 78 81 90
-----
Process exited after 0.03799 seconds with return value 0
```

observable behavior

```
void foo(double list[], int arraySize)
{
    for (int i = arraySize - 1; i >= 1; i--) {
        double currentMax = list[0];
        int currentMaxIndex = 0;
        for (int j = 1; j <= i; j++) {
            if (currentMax < list[j]) {
                currentMax = list[j];
                currentMaxIndex = j; }
        }
        if (currentMaxIndex != i) {
            list[currentMaxIndex] = list[i];
            list[i] = currentMax; }
    }
}
```

2. 요구오류와 에러

Error/Mistake, Fault, Failure의 예

- ❑ Sample requirement: find speed (s) and print the result, when distance (t) and time (t) are given

```
distance = float(input("Enter the distance (in units): "))  
time = float(input("Enter the time taken (in units of time): "))  
speed = distance / time  
print("The speed is:", speed)
```



mistake	programmer missed out the case time is 0
fault	there is no code for handling the case : time = 0
failure	System stopped with exception when time is 0

2. 요구오류와 에러

Error/Mistake, Fault, Failure의 예

- ❑ better requirement: find speed (s) and print the result, when distance (t) and time (t) are given **with exception handling for divide by zero case**

```
try:
    distance = float(input("Enter the distance (in units): "))
    time = float(input("Enter the time taken (in units of time): "))

    if time == 0:
        raise ZeroDivisionError("Time cannot be zero.")
    speed = distance / time
    print("The speed is:", speed)

except ValueError:
    print("Invalid input. Please enter a valid number.")

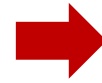
except ZeroDivisionError as err:
    print("Error:", err)
```

2. 요구오류와 에러

정교한 요구명세의 사례

일반적인 (오름차순) 정렬 명세의 예

Find the list[0..n-1], where
 $list[0] \leq list[1] \leq \dots \leq list[n-2] \leq list[n-1]$



```
void sort(double list[], int arraySize)
{
    for (int i = 0; i <= arraySize - 1; i++)
        list[i] = 0;
}
```

(X)



완전한 명세의 형태

Find the list[0..n-1], where

- 1) $list[0] \leq list[1] \leq \dots \leq list[n-2] \leq list[n-1]$
- 2) result list is a permutation of actual (original) list

2. 요구오류와 에러

정교한 요구명세의 사례 – ChatGPT의 예

정확한 프로그램은 정교한 명세로부터 시작이 됩니다.

Python code를 짜 주세요.

- 1) array `stock_price[i]`에 매일의 주가가 있습니다. `i`의 범위는 0에서 `N-1` 로 가정을 합니다.
- 2) 이 `stock_price[i]`에서 20일 이동평균을 구하여 `mv_20[i]`에 저장을 하고, 또 5일 이동평균을 구하여 `mv_5[i]`에 저장을 합니다.
- 3) 20일 이동평균이 5일 이동평균의 밑에 있다가 위로 올라가는 때에 알려주는 코드를 부탁드립니다.
- 4) 또 볼린저 밴드 상한값은 `bol_high[i]`, 볼린저 밴드 하한값은 `bol_low[i]`에 저장을 해 주세요.
- 5) 끝으로 `stock_price[i]`, `mv_5`, `mv_20`, `bol_high`, `bol_low` 를 중첩해서 그림을 그려주세요.

```
import numpy as np
import matplotlib.pyplot as plt

# Define the window sizes for the moving averages and Bollinger Bands
window_size_5 = 5
window_size_20 = 20
num_std_dev = 2

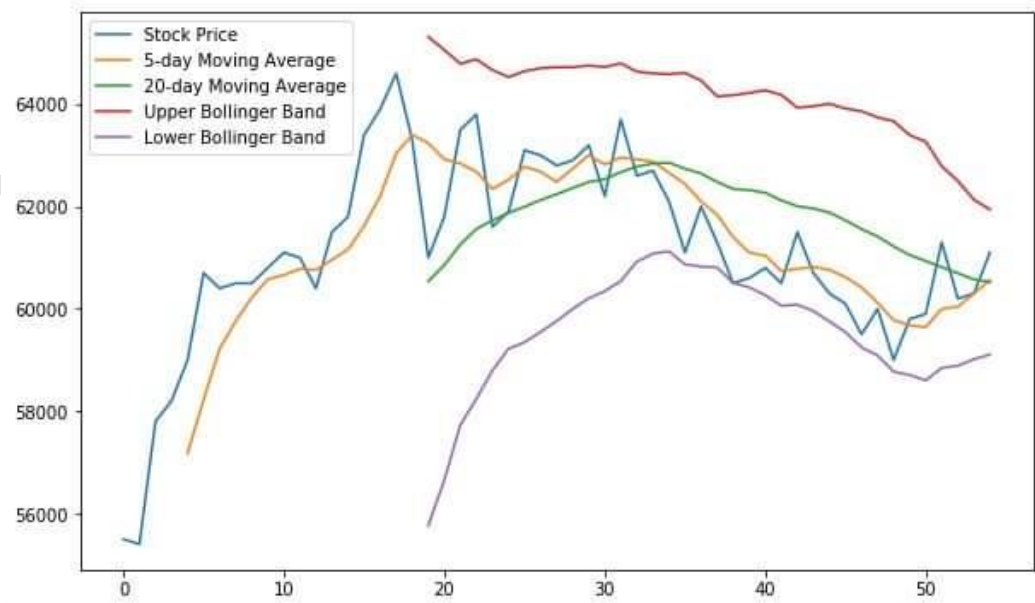
# Calculate the moving averages
mv_5 = np.convolve(stock_price, np.ones(window_size_5)/window_size_5, mode='valid')
mv_20 = np.convolve(stock_price, np.ones(window_size_20)/window_size_20, mode='valid')

# Calculate the standard deviation and the Bollinger Bands
rolling_std_dev = np.std(stock_price)
bol_high = mv_20 + num_std_dev * rolling_std_dev
bol_low = mv_20 - num_std_dev * rolling_std_dev

# Initialize variables for tracking when the moving averages cross
crossed_below = False
crossed_above = False
crossing_indices = []

# Loop through each day and check if the moving averages have crossed
for i in range(window_size_20-1, len(stock_price)):
    if mv_20[i] < mv_5[i] and not crossed_below:
        crossed_below = True
    elif mv_20[i] > mv_5[i] and crossed_below:
        crossed_below = False
        crossing_indices.append(i)
    elif mv_20[i] > mv_5[i] and not crossed_above:
        crossed_above = True
    elif mv_20[i] < mv_5[i] and crossed_above:
        crossed_above = False
```

... (중략)



3. 요구의 구분

Requirements의 역할

❑ 올바른 명세의 역할



- ❑ Domain property: things that users wish or need
- ❑ Requirements : things that we wish to be made true by delivering the proposed system
- ❑ A specification : a description of behaviours that program must have in order to meet the requirements

3. 요구의 구분

Needs와 Requirements



요구(needs)란 고객이 실제로 원하는 것이다

Something required; something wanted or needed.

웹스터 사전

A requirement is a complete statement of **what system will do** without referring to how it will do it.

일반적인 SE 정의

An **Externally-Observable Characteristic** of a Desired System.

Alan Davis

A requirement describes a condition or capability to which a system must conform; either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document

IBM*

3. 요구의 구분

Requirements- 대상과 목적에 따른 구분



대상과 목적

**Business
Requirements**

User Requirements

System Requirements

클래스 구분

**Functional
Requirements**

**Non-functional
Requirements**

**Domain
Requirements**

3. 요구의 구분

Requirements- 대상과 목적에 따른 구분

❑ Business Requirements

- high-level statements of the goals, objectives, or needs of an organization. Explain 'why' the project is needed.

❑ User requirements

- Statements of the services and operational constraints the user can access. Explain 'what' services the user may have.
- Written for the users.

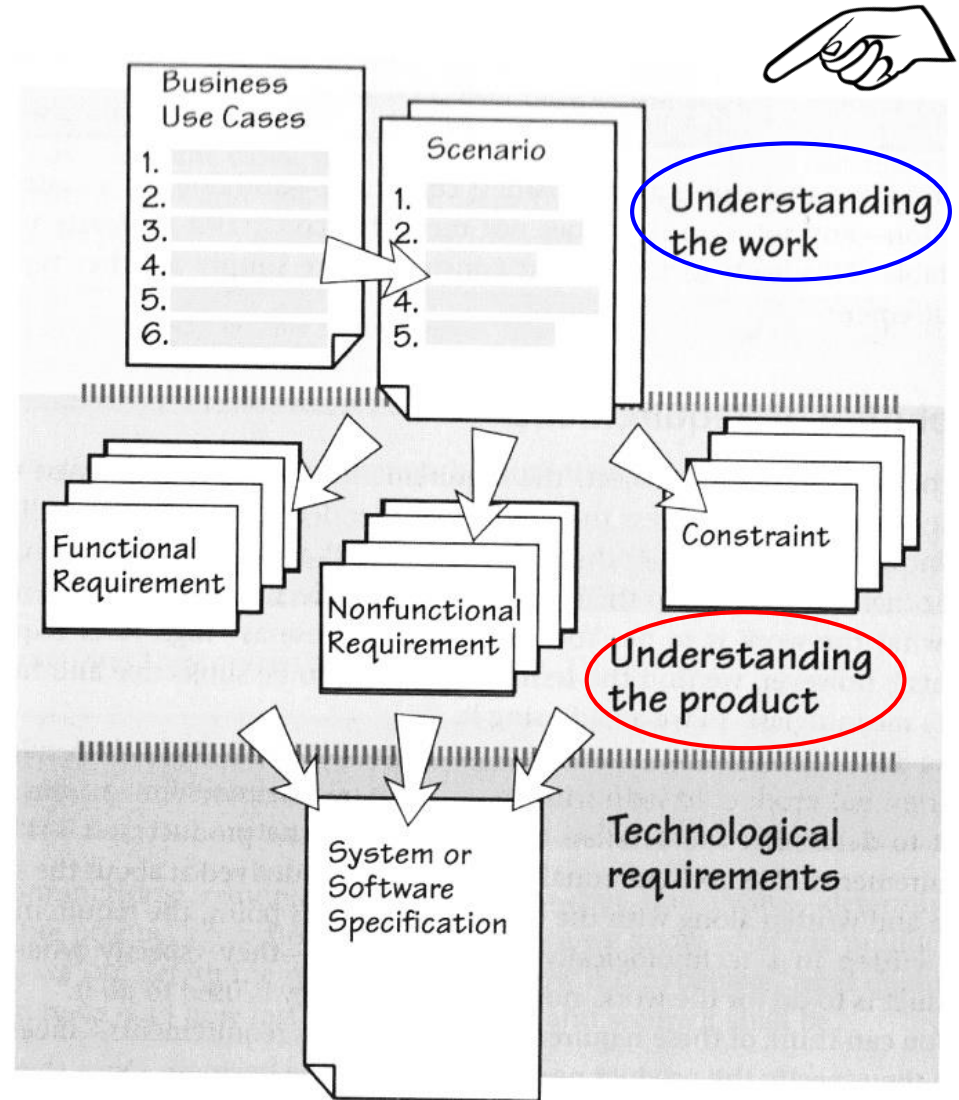
❑ System requirements

- A structured document setting out detailed descriptions of the system's functions, services and operational constraints.
- Defines what should be implemented so may be part of a contract between client and developers.
- Used for subsystems or components' high-level specifications.

3. 요구의 구분

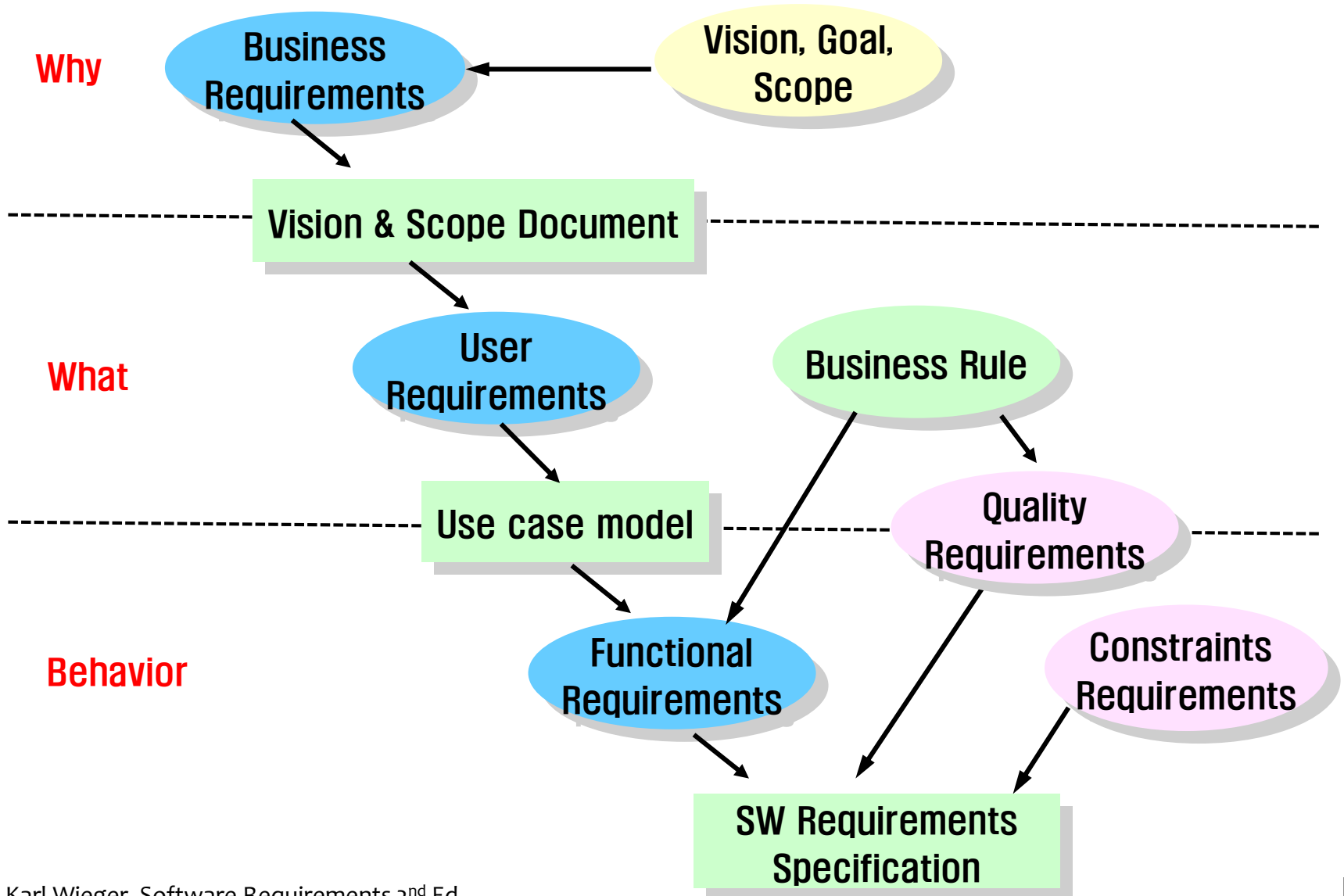
Requirements- 대상과 목적에 따른 구분

- ❑ Business requirements
 - Business Use Cases
- ❑ User requirements
 - Functional Requirements
 - Non-functional Requirements
- ❑ System requirements
 - System Specification
 - Sub-system specification
 - Software Requirements Specification (SRS)



3. 요구의 구분

Requirements- 계층에 따른 구분 (K. Wieger)



3. 요구의 구분

Requirements- Business Requirements

비즈니스 요구의 특징

- SW 납기와 예산 추정은 요구사항 정의 이전에 경영진이나 마케팅에 의해 결정
- 비현실적인 납기에 따른 형편없는 추정과 그에 따른 자원의 압박은 위험요소임
- 프로젝트 진행 중 거의 조정되지 않음

궁극적인 목적에 대한 전략 수립 : 요구사항의 정의 이전에 조직의 문제를 이해

Need – “왜 해당 조직에서 프로젝트를 수행하는가?”

- Establishes the focus for the project team
- The Need should not change

Goals – “프로젝트를 통해 성취하고자 하는 것은 무엇인가?”

- “고객이 제품 개발로 얻을 수 있는 이득이 무엇인가?”

Objectives – “요구를 만족시키기 위하여 무엇을 수행하여야 하는가?”

3. 요구의 구분

Requirements- Business Requirements

비즈니스 요구사항을 조사하는 이유는 새롭게 창출될 비즈니스의 기회, 목적, 성공조건, 제품 비전, 프로젝트의 범위에 대한 이해를 공유하기 위함이다

솔루션의 목적



- 해결하고자 하는 비즈니스 문제점이 무엇인가? (방향성)
- 고객은 성공적인 솔루션을 통해 무엇을 얻는가?
- 솔루션의 성공여부를 판단하는 기준은 무엇인가? (목표)

솔루션의 범위



- 솔루션에 포함될 및 포함되지 말아야 할 비즈니스 활동과 항목은 무엇인가? (범위, 경계의 정의)
- 이 프로젝트에 영향을 미치거나 받게될 프로젝트는?

솔루션의 영향



- 이 프로젝트에 영향을 미치거나, 받는 사람이나 그룹은? (이해당사자 식별)
- 어떤 프로젝트가 이 프로젝트에 영향을 미치거나 받는가?
- 예측하지 못한 결과나 부작용을 발생할 수 있는가?

6. 요구 추출 기법

비즈니스 요구사항을 위한 질문



화학관리 시스템에 관심을 갖게 된 계기는 무엇이지요?

수작업의 화학약품 재고관리는 비용이 많이 들고 비효율적이어서 고려하게 되었습니다.

약품관리에 대한 비용을 얼마나 줄이고자 하나요?

1년 안에 25% 줄이고자 합니다.

높은 비용과 비효율을 초래하는 원인은 무엇인가요?

재고에 어떤 화학제품이 남았는지 모르다 보니 불필요한 제품을 구입하게 됩니다. 결국 유통기한이 지난 너무 많은 화학약품을 폐기하게 됩니다.

제가 알아야 할 것이 또 있나요?

주문이 복잡하다 보니 오래 걸립니다. 그리고 정부 보고서를 수작업으로 만드는 것도 시간이 너무 오래 걸립니다.



[for whom] 화학제품주문이 필요한 실험자에게 [product] 화학제품 관리시스템은 화학약품 창고나 공급업체와 단일접점을 제공하는 정보시스템이다. [what] 본 시스템은 사내의 모든 화학용기 위치, 재고수량, 사용내역에 대한 모든 기록을 저장한다. [why] 이 시스템은 사용빈도가 떨어지거나 유통기한이 지난 용기를 폐기하며, 표준화학 약품 구매 프로세스 사용함으로써 회사지출의 25%를 절약한다. [Unlike] 기존의 수동 주문 프로세스와는 달리 우리제품은 화학약품 사용및 보관, 폐기현황보고를 의무화하는 정부규정을 준수하는데 필요한 보고서를 생성한다.

3. 요구의 구분

Requirements: Functional Requirements

- ❑ Functional requirements
 - Statements of services the system should provide, **how the system should react to particular inputs** and **how the system should behave in particular situations**.
 - Describe functionality or system services.
 - Depend on the type of software, expected users and the type of system where the software is used.
 - **Functional user requirements** may be high-level statements of what the system should do.
 - **Functional system requirements** should describe the system services in detail.



기능요구는 사용자와 개발자 시각에서 모두 서술이 가능하다
따라서 요구의 확인과 검증이 중요하다

3. 요구의 구분

User and System Requirements

User Requirements

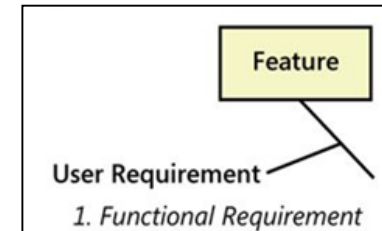
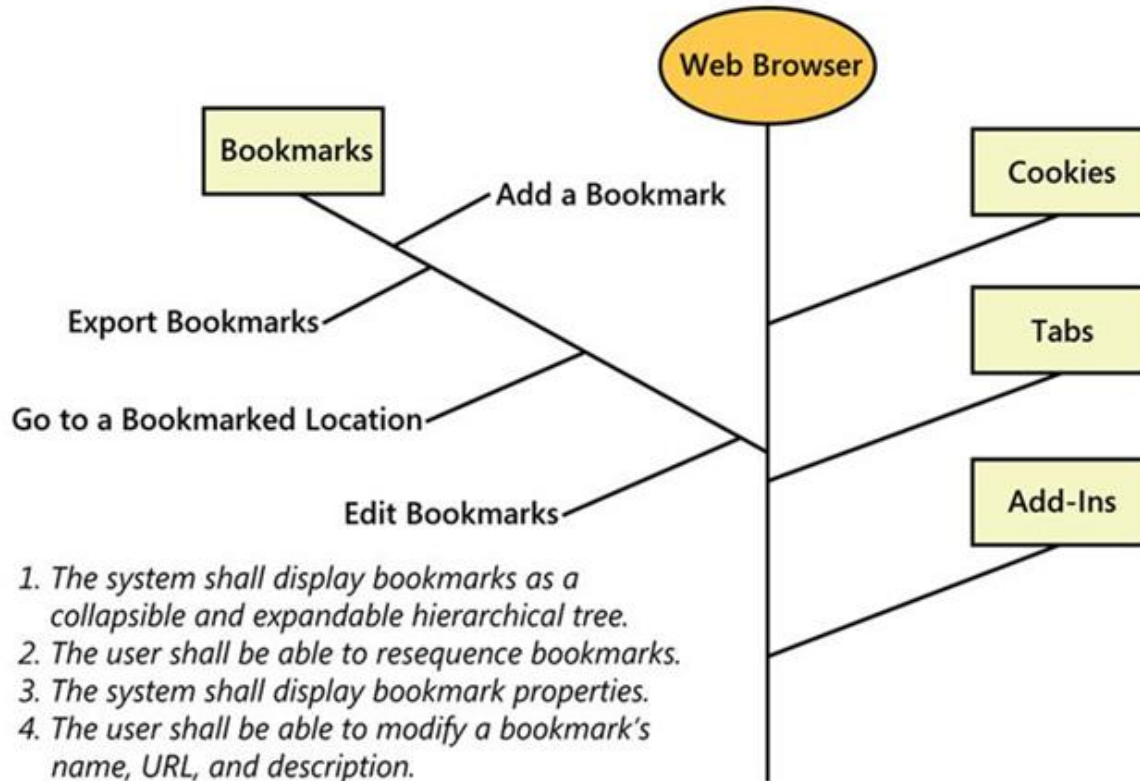
1. The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

Functional (System) Requirements

- 1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- 1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
- 1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
- 1.4 If drugs are available in different dose units (e.g. 10mg, 20 mg, etc.) separate reports shall be created for each dose unit.
- 1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

3. 요구의 구분

Requirements: Functional Requirements– function, feature



특징(feature)이란 논리적으로 관계를 가지는 기능 요구사항들의 집합이며, 사용자에게 하나의 기능을 제공하고, 하나의 비즈니스 목표를 만족시키는 역할을 한다

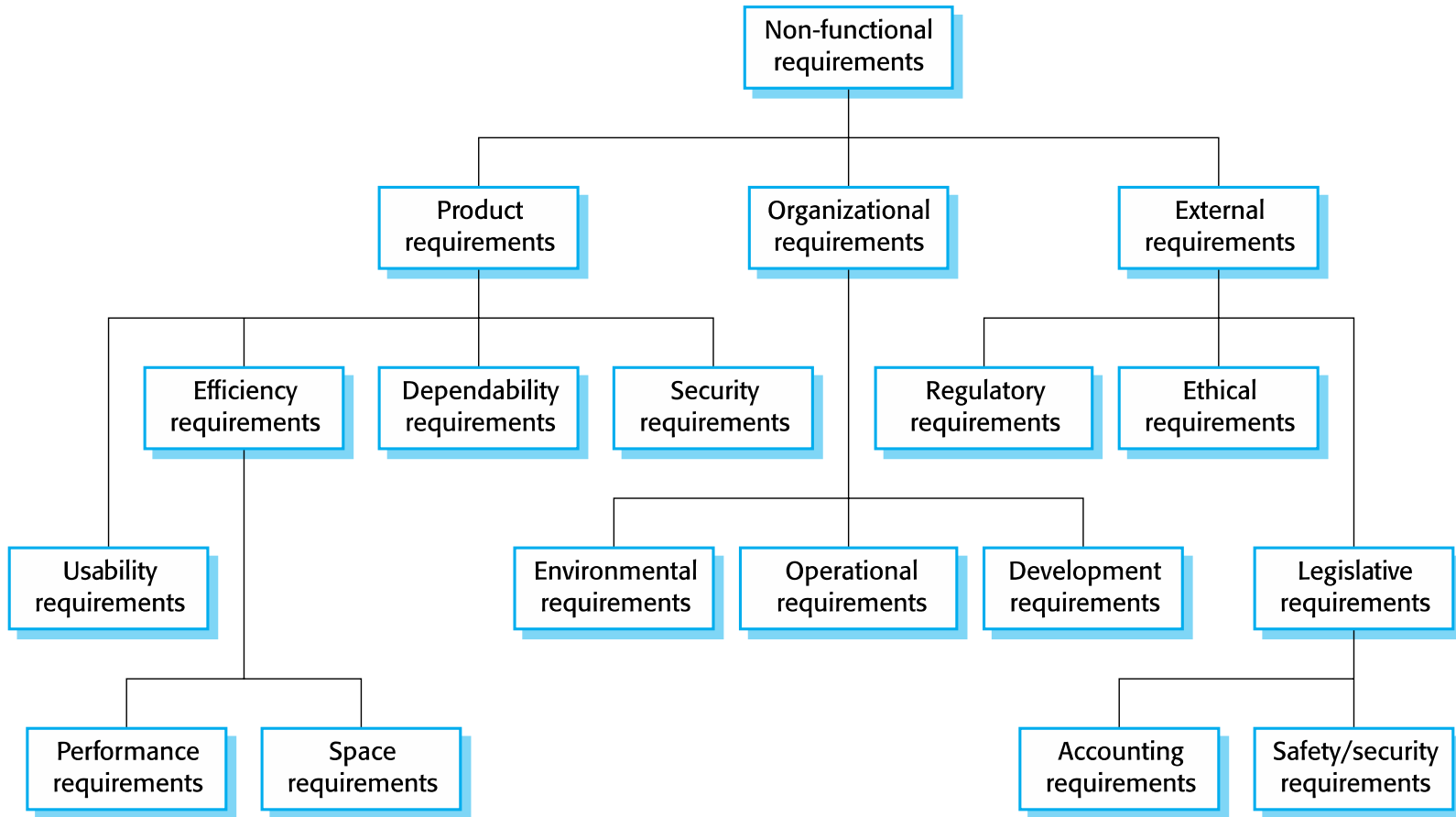
3. 요구의 구분

Requirements : Non-functional / Domain Requirements

- ❑ Non-functional requirements
 - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
 - Often apply to the system as a whole rather than individual features or services.
 - These define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
- ❑ (Operational) Domain requirements
 - Constraints on the system from the domain of operation
 - Domain requirements reflect the environment in which the system operates
 - Example :The system safety shall be assured according to standard IEC 60601-1:Medical Electrical Equipment.

3. 요구의 구분

비기능요구의 분류(taxonomy)



4. 요구와 요구공학

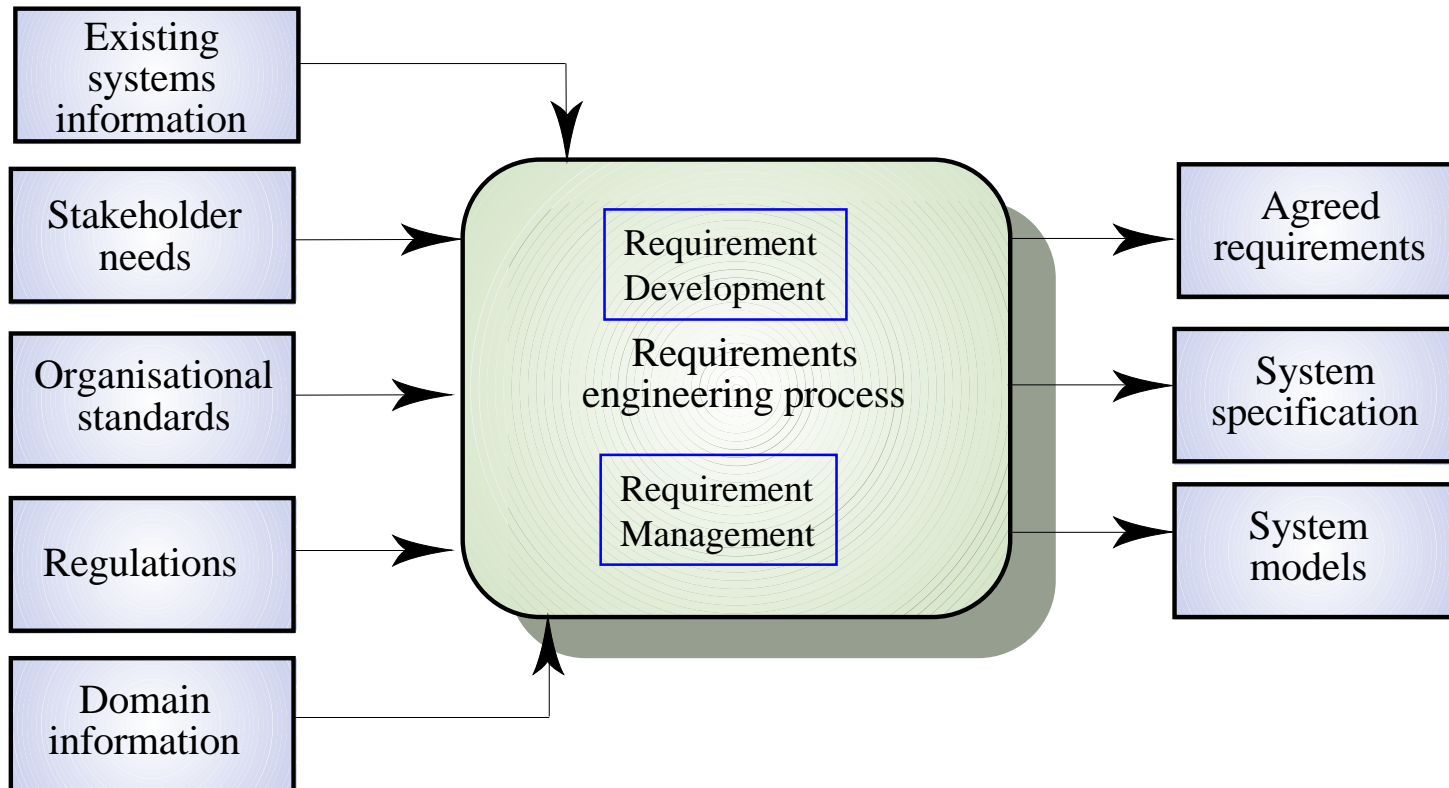
요구공학과 요구분석

- 요구공학 (Requirements Engineering)
 - 컴퓨터 기반 솔루션과 관련된 요구 집합의 파악, 문서화, 관리를 위한 모든 공학적 활동을 지칭함 (Sommerville)
- 요구분석 (Requirements Analysis)
 - 개발될 시스템이 수행해야 할 기능과 제약사항을 명확히 정의하고 기록하며, 나아가서는 검증하는 일을 수행함
 - 요구공학은 요구분석에 대한 올바른 기법과 도구, 방법을 체계적으로 제공하는 목표를 갖는다
- 공학적 (engineering)이란 의미는 요구에 대한 완전성, 일관성, 정확성을 확보할 수 있도록 하는 시스템적이고 반복적인 (systematic and repeatable) 접근을 말한다

4. 요구와 요구공학

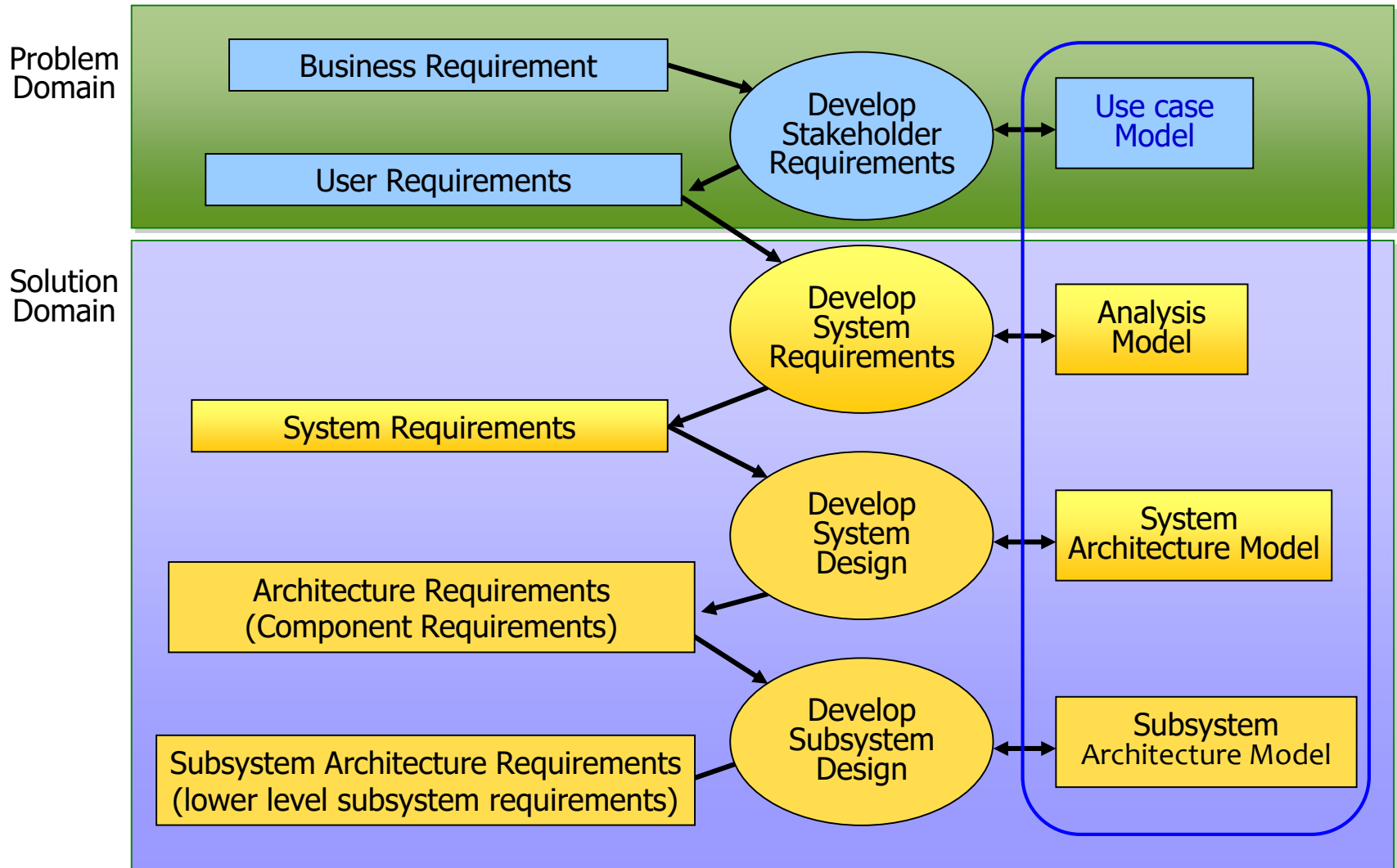
요구공학 프로세스 – Sommerville

- 요구분석을 구성하는 두 개의 프로세스: 요구개발과 요구관리



4. 요구와 요구공학

SDLC 상의 Model



4. 요구와 요구공학

SW요구 명세서의 구성

1. Introduction

1.1 Purpose of the system

1.2 Scope of the system

1.3 Objectives and success criteria of the project

1.4 Definitions, acronyms, abbreviations

1.5 References

2. Current system

3. Proposed system

3.1 Overview

3.2 Functional requirements

3.3 Nonfunctional requirements

3.3.1 Usability

3.3.2 Reliability

3.3.3 Performance, ...

3.4 System models

3.4.1 Scenarios

3.4.2 Use case model

3.4.3 Object model

3.4.4 Dynamic model

3.4.5 User interface, navigation path, screen mock-ups

4. Glossary

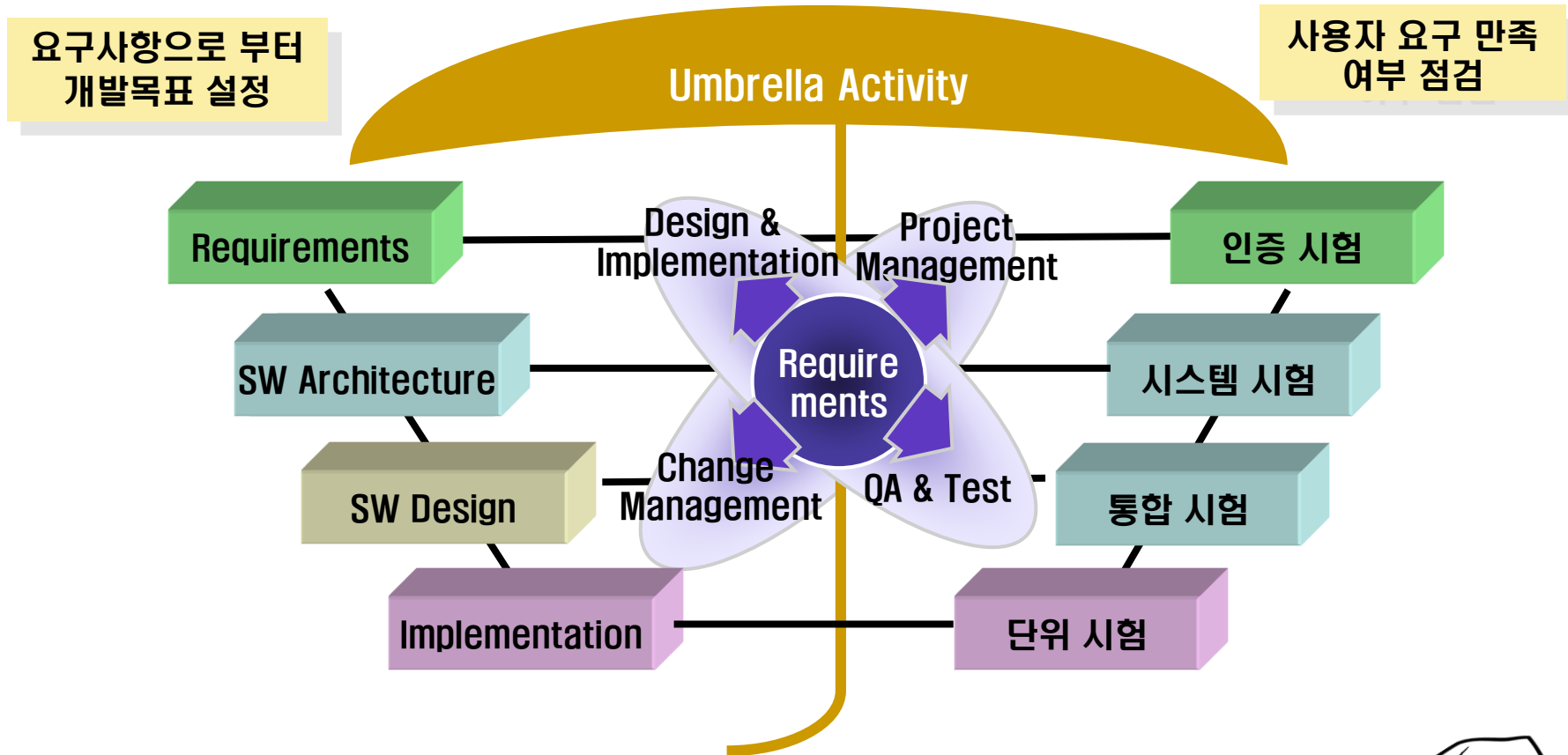
4. 요구와 요구공학

요구관리

- ❑ 요구관리 활동은 획득된 요구사항을 문서화 한 이후 발생하는 변경에 관한 관리가 주요 내용이다.
- ❑ 요구관리 활동은 다음 세부 내용을 포함한다.
 1. 요구사항 기준선 설정
 2. 요구사항 변경에 대한 검토 및 위험 관리
 3. 변경 승인
 4. 프로젝트 계획과 요구사항 일관성 유지
 5. 추정된 변경 요구사항에 대한 협상
 6. 요구사항 후속 공정에 대한 산출물 관리와 추적
 7. 요구사항 상태 기록 및 변경 활동의 추적

5. 요구관리

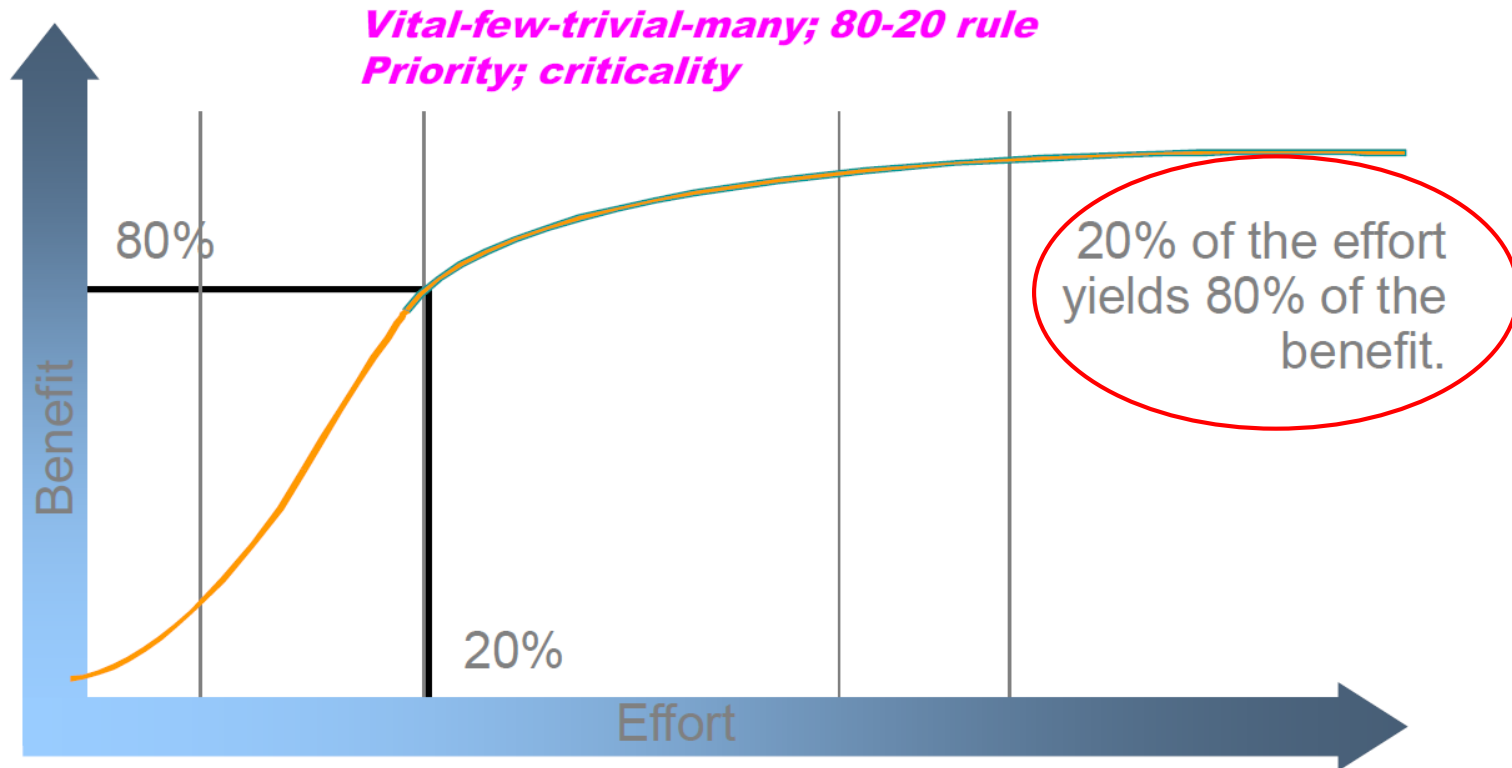
Umbrella 활동으로서의 요구관리



- communication 수단을 제공 : Stakeholders 사이에 공통의 이해 설정
- Umbrella Activity : SE 전반에 적용되는 관리활동

5. 요구관리

요구관리와 Pareto's Law



What are in the 20% of the problem of your project?

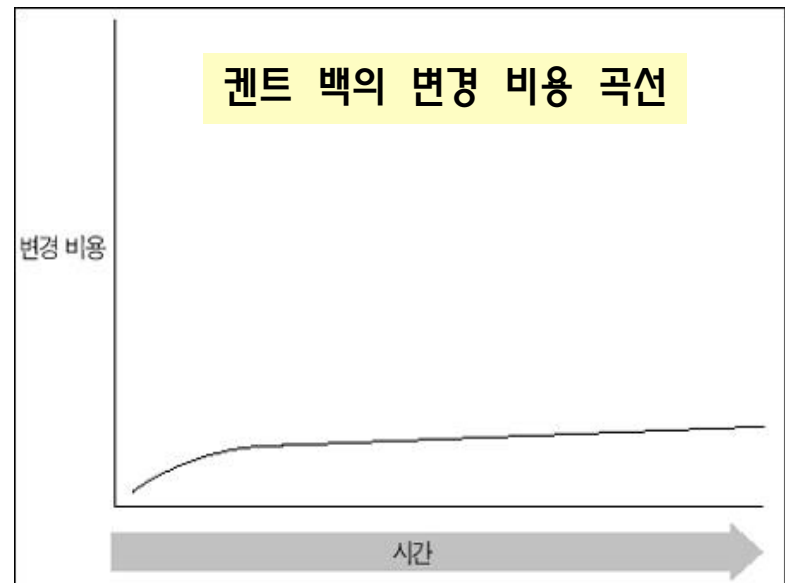
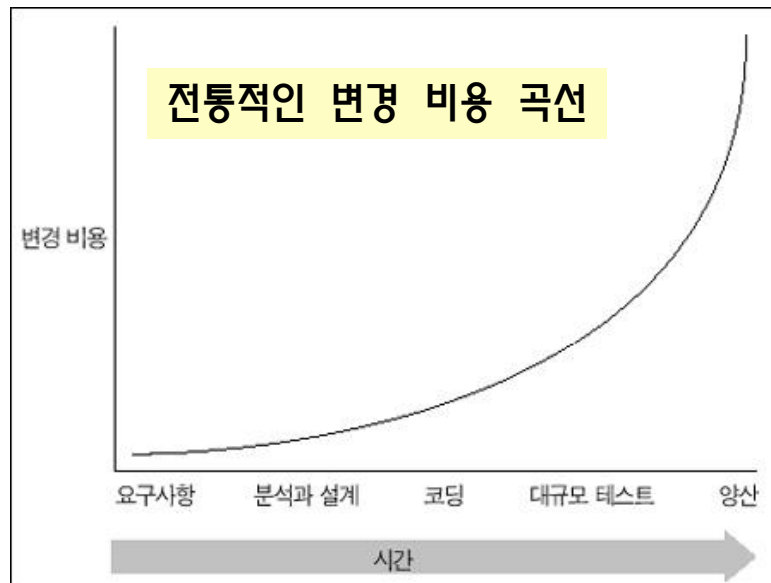
5. 요구관리

Agile 기반의 변경관리

XP (eXtreme Programming)

- 문서를 강조하지 않고 변경 장려, 개발 초기부터 테스트 병행을 권고
- Pair programming, informative workspace 유지
- 고객이 가장 중요하게 생각하는 것을 먼저 구현
- 1 ~ 3주 내에 구현할 수 있는 요구사항(유저 스토리)

XP 5 values: : Communication, Simplicity, Feedback, Courage, Respect



Workshop #1 – Business Requirement

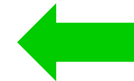
비즈니스 요구사항 작성

- 요구사항 : 각 팀별로 확정된 case study에 대해 다음 사항이 표현될 수 있도록 비즈니스 요구사항을 작성합니다.
- 비즈니스 요구사항을 중심으로 프로젝트의 성공기준의 내용이 반영될 수 있도록 서술합니다 (비즈니스 요구사항 템플릿 참조)
 - Time, Budget, Target user
 - Goal, Value
 - Quality Issues
- 시간: 60 분 (A4 1매 분량)
- 제출파일명: **팀명_비즈니스 요구사항.***** 혹은 docs링크 (Naver 카페에 업로드함)

Workshop #1 – Business Requirement

비즈니스 요구사항 템플릿

1. **비즈니스 요구사항**
 - 1.1 배경
 - 1.2 비즈니스 목표
 - 1.3 성공지표 및 품질수준
 - 1.4 비즈니스 위험
 - 1.5 비즈니스 가정 및 의존성
2. **범위 및 한계**
 - 2.1 주요기능
 - 2.2 주요 품질 요구
 - 2.3 출시범위 및 방법
3. **비즈니스 컨텍스트**
 - 3.1 이해관계자 프로파일
 - 3.2 프로젝트의 우선순위



팀 토론을 통해 필요한 부분을
선정하여 정의해봅니다