Spring 2023

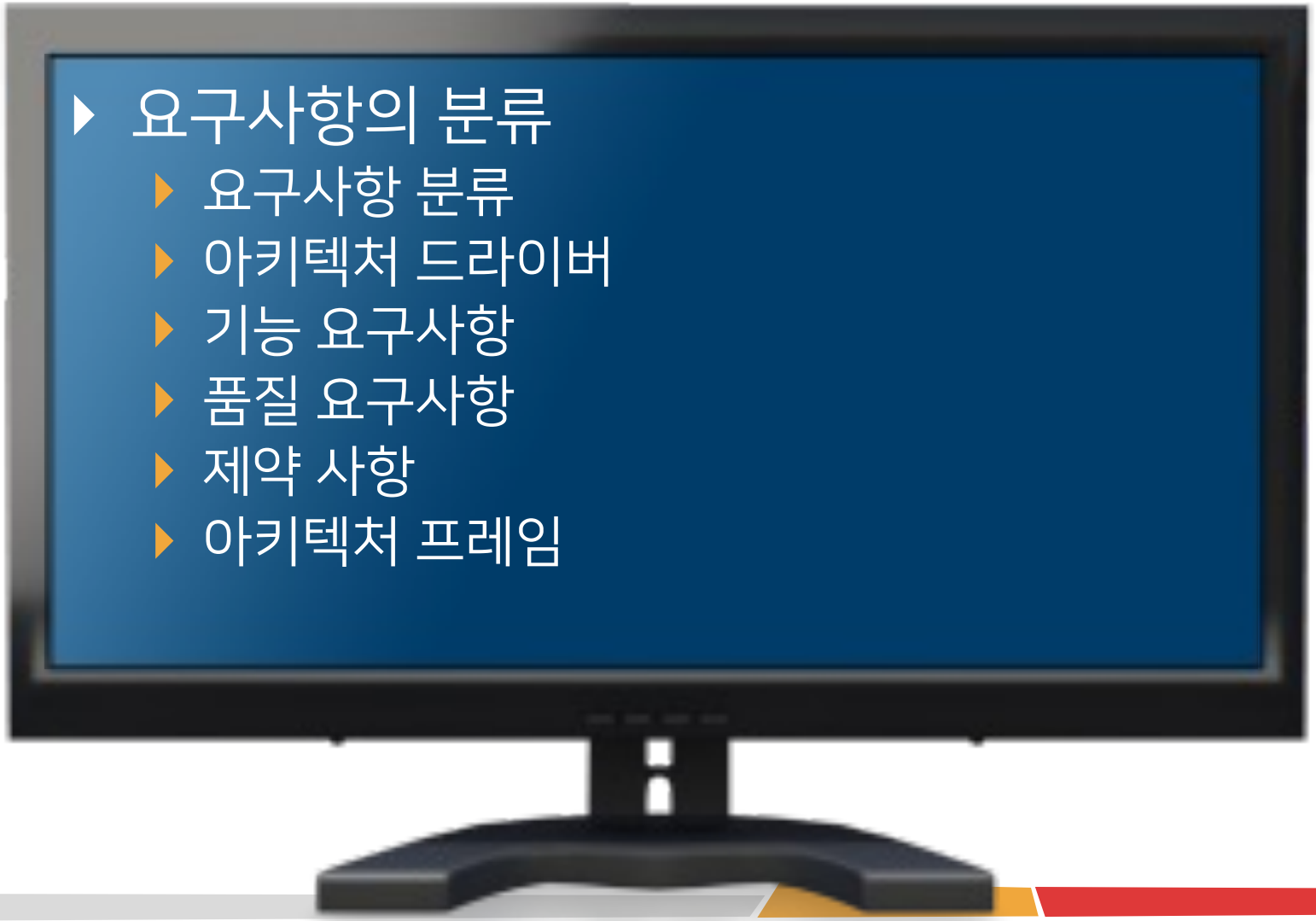
# Quality Attributes and Software Architecture

Seonah Lee

Gyeongsang National University

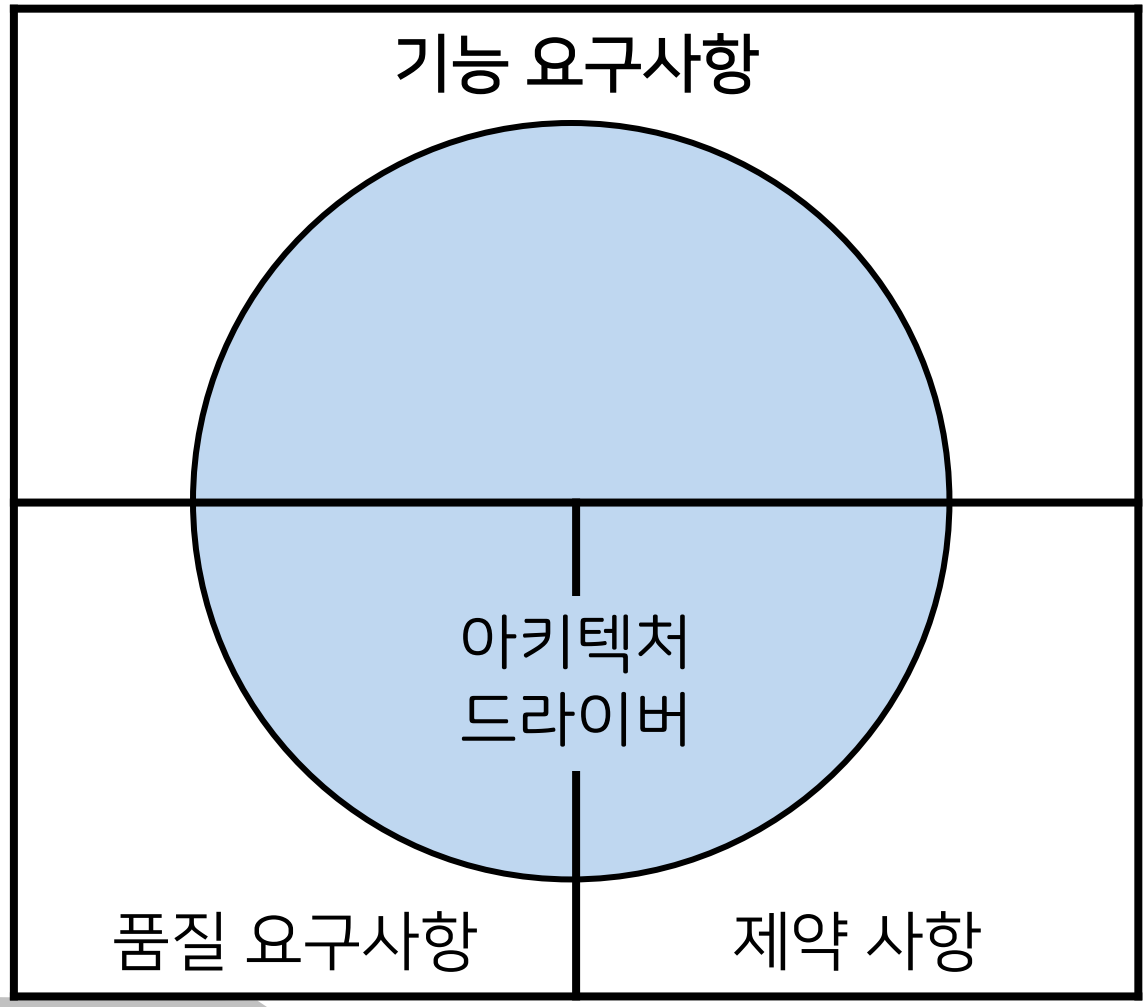
# 목 차

# 품질 속성



# Software Requirements

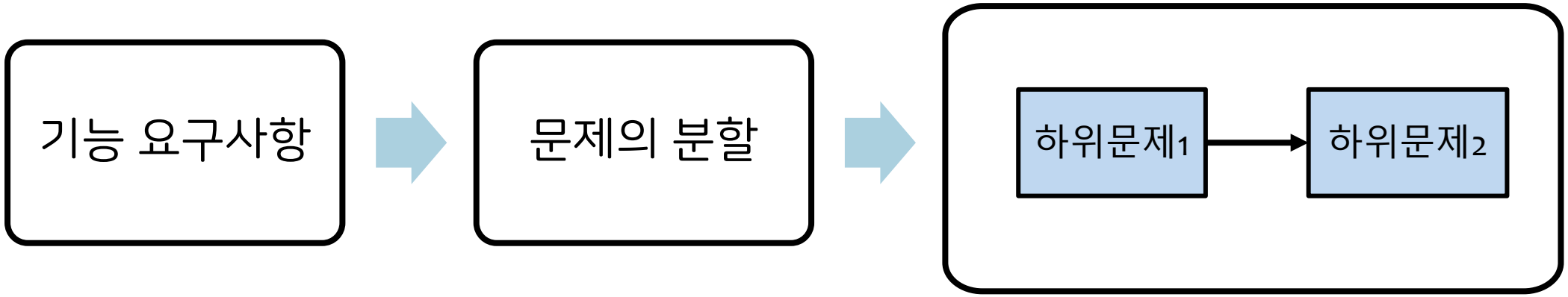
# Software Drivers




- 정의. 시스템 아키텍처의 결정에 큰 영향을 미치는 요구사항을 아키텍처 드라이버라고 부른다.
- 기능요구사항은 간접적으로 품질요구사항을 발생시킴으로써 아키텍처에 영향을 준다.

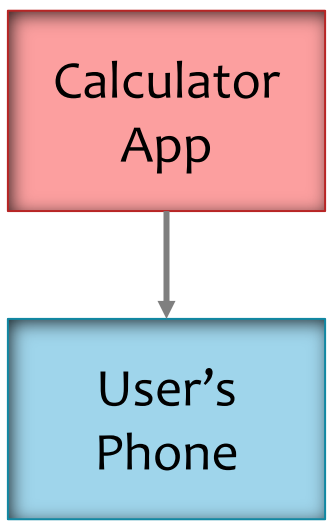
# Functional Requirements

- 시스템의 행위를 정의하며, 유스케이스 혹은 사용자 스토리 등으로 기술
- 소프트웨어 아키텍처와의 관계
  - 주로 기능적 요구 사항에 대한 분석으로 문제를 분할함
  - 기능적 요구사항이 아키텍처 설계와 직접적으로 연관되지는 않음
    - 비유 예: 나는 방 **3**개와 화장실 **2**개가 있는 집이면 좋겠습니다.

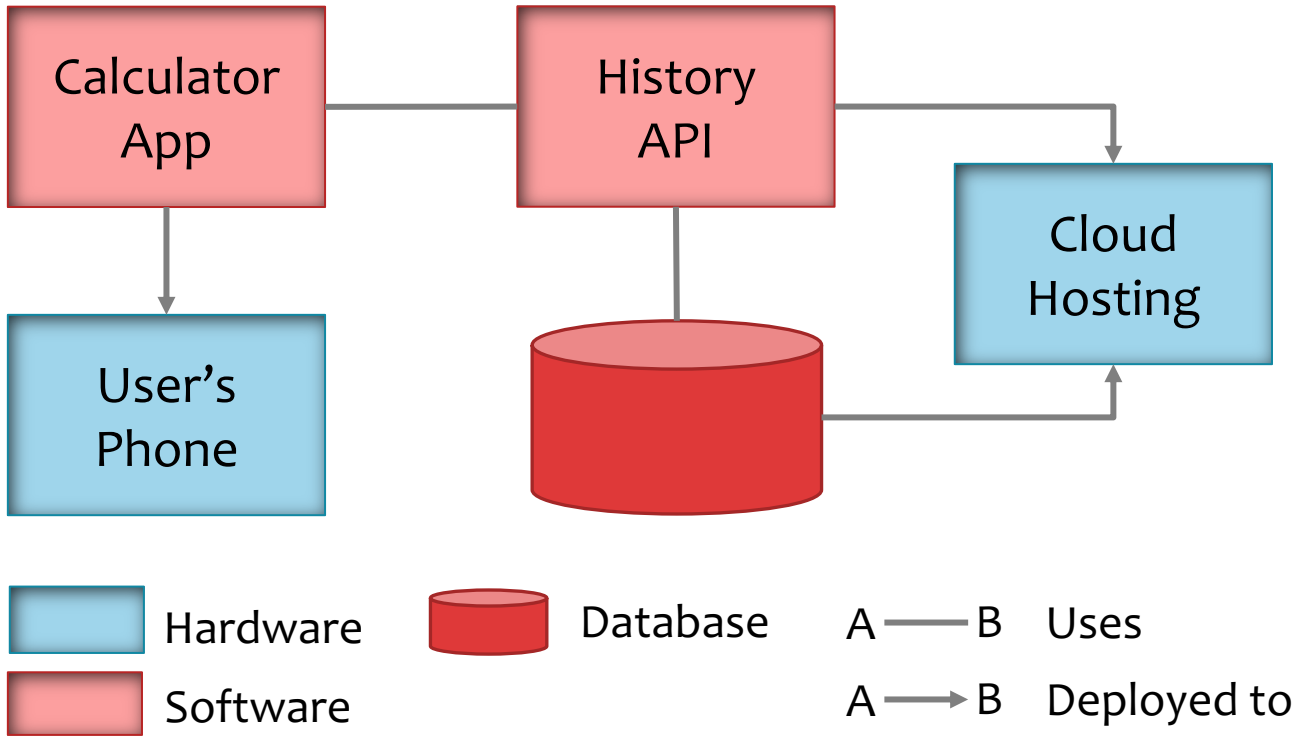


# Functional Requirements

- 문제의 분할에 주요 영향을 미치는 기능 요구사항이 있다
  - 계산기 앱: “폰을 분실해도 계산 이력 조회가 가능해야 한다”




이력 조회 없음



# Quality Requirements

- 알람 시스템이 울리거나 센서가 작동할때, 사용자는 알람이 울리지 않도록 숫자로 된 패스워드를 입력할수 있도록 한다.


Q) 어느 정도로 요구사항을 상세하게 해야 하는지?

테스트가 가능한 각각의 요구사항을 작성하는 것. 요구사항이 적절하게 구현되었는지를 확인하기 위해 몇 개의 관련된 테스트 사례를 생각하면 그 정도가 적절한 상세 수준이 될 것이다.
만약 생각하고 있는 테스트가 너무 많고 다양하다면?

# Quality Requirements

**품질 속성이란 특정 문맥에서 어떻게 시스템이 동작하는지를 특성화한 시스템의 특성. 양이나 질로 관찰하여 수치로 측정**

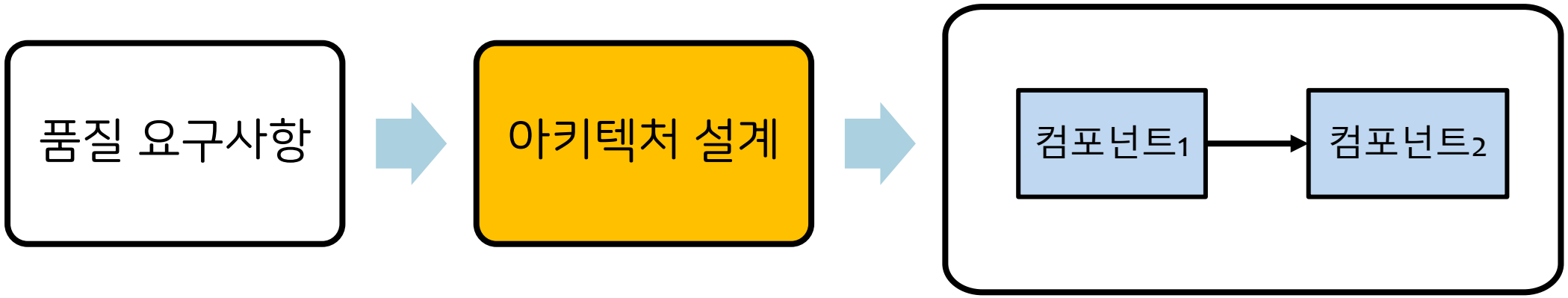
- 품질속성은 이해관계자들의 관심사와 요구사항을 그대로 반영한다.
- 아키텍처는 이해관계자들이 원하는 수준으로 품질 속성을 달성해야 한다.

**품질 요구사항이란 시스템이 제공해야 할 품질 속성의 수준**

- 품질 속성은 관찰할 수 있고 측정할 수 있어야 하기 때문에 품질 요구사항도 가능하면 정확한 수치로 제시되어야 한다.

# Quality Requirements

- 소프트웨어 시스템의 가시적인 특성을 기술; 시스템 동작의 기대를 표현
- 소프트웨어 아키텍처와의 관계
  - 품질 요구사항을 바탕으로 우선 순위를 수립함
  - 설계한 컴포넌트를 대상으로 조합하여 기대 품질이 가능한지를 확인



# Constraints

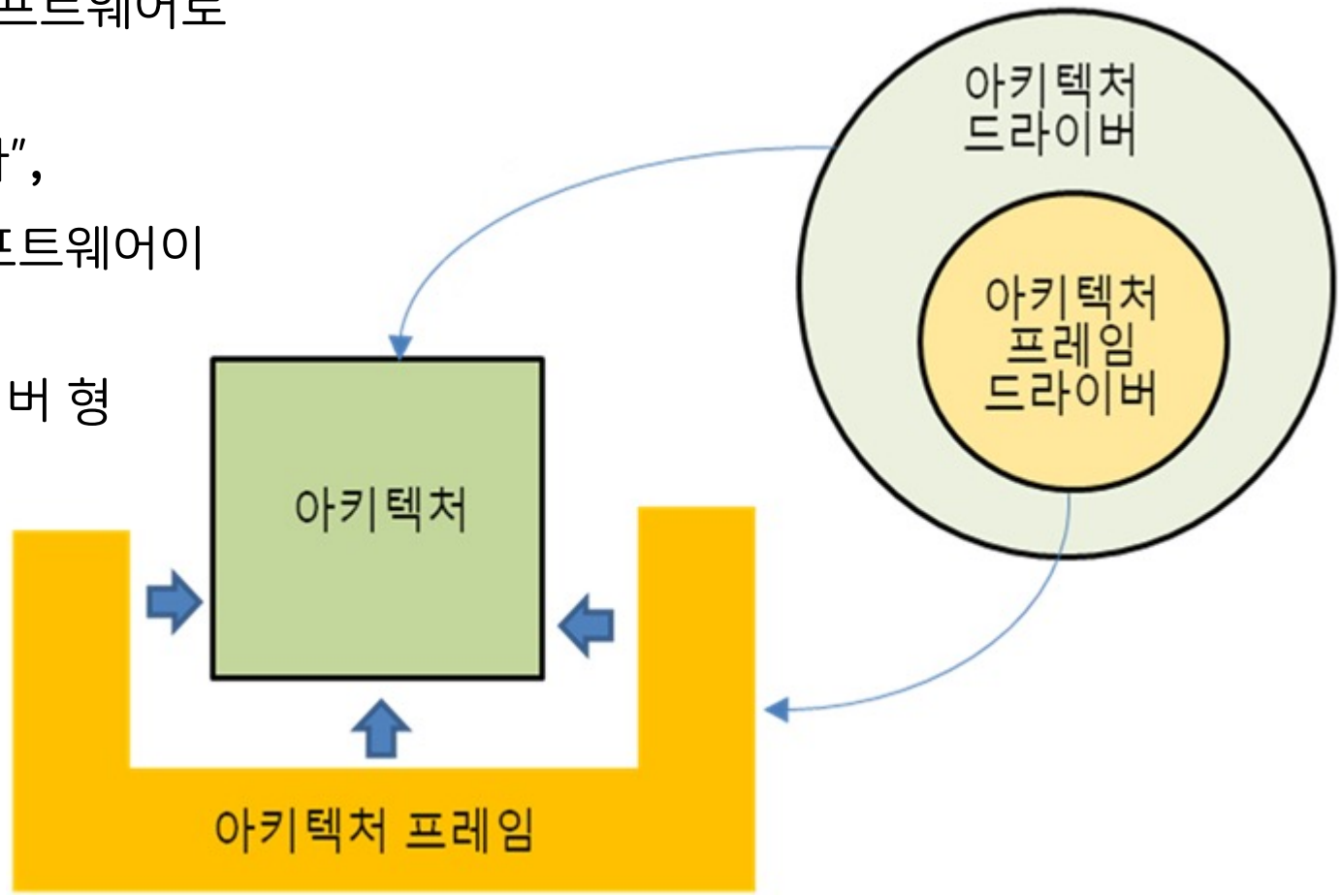
- 변경할 수 없는 설계 결정, 대부분 프로젝트를 시작하는 시점에 이미 결정되어 있음. 가끔씩 선택할 수 있음

| Technical Constraints | Business Constraints |
|---|---|
| **프로그래밍 언어**: 모든 것이 JVM에서 실행될 수 있도록 Java를 선택 | **팀 구성**: 팀 X에서는 XYZ 컴포넌트를 개발하여 빌드한다 |
| **운영 체제, 플랫폼**: Wihdow와 Linux에서 실행되어야 함 | **일정 및 예산**: Big Trade Show까지는 오픈을 해야 하며 비용은 $800,000이하여야 한다 |
| **컴포넌트 및 기술**: 현재 보유하고 있는 DB2를 데이터 베이스로 사용 | **법적 제약**: 우리의 라이센스에 하루 5GB까지만 사용할 수 있도록 되어 있다 |

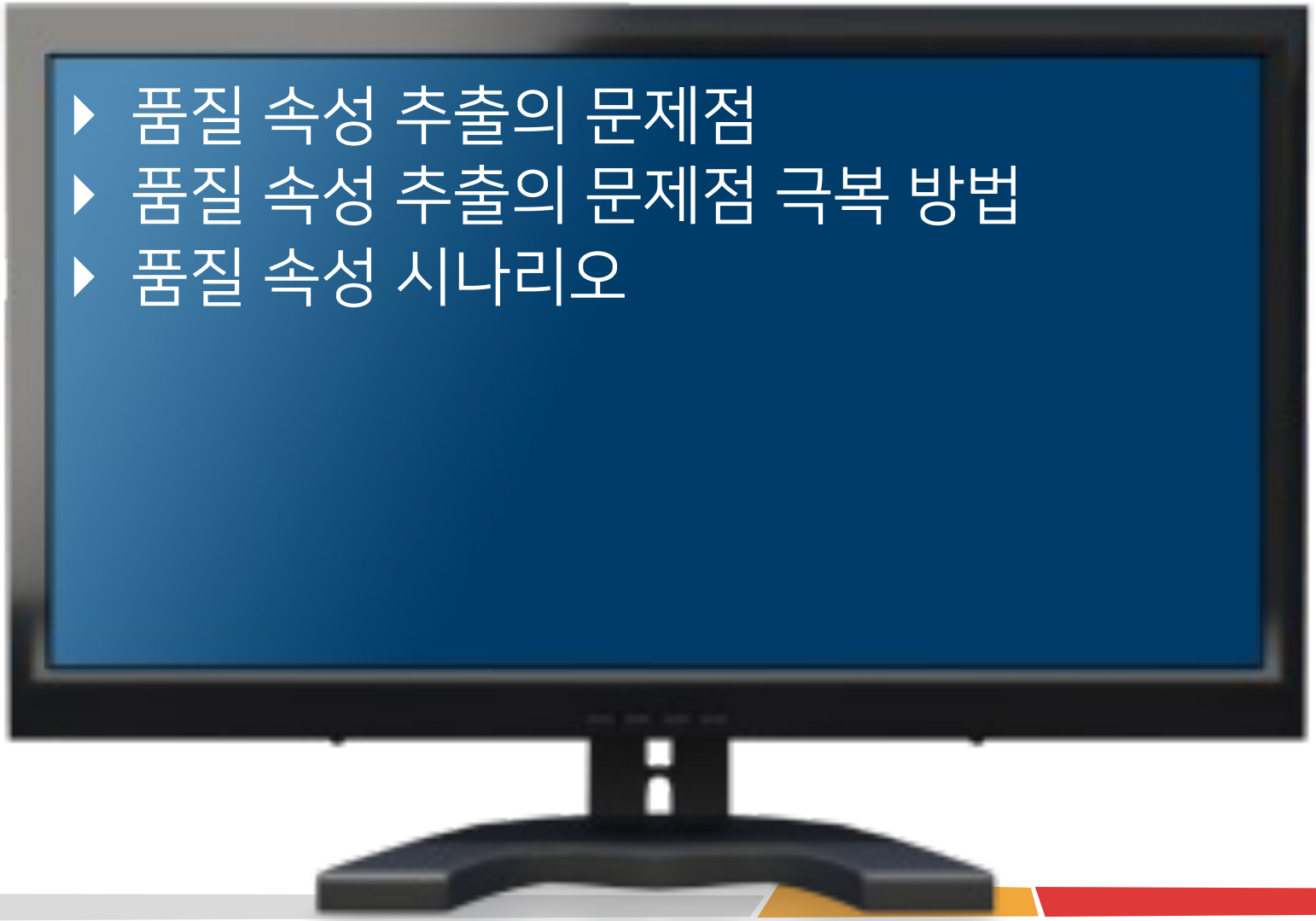
- 잘 선택된 제약 사항은 문제를 단순화할 수도 있음
- 혹은 아키텍트의 선택의 폭을 좁혀 요구사항 만족을 힘들게 함

# Architecture Frame

- "개발대상소프트웨어가 임베디드 소프트웨어로서 **CORBA** 미들웨어를 사용한다",
- **"**개발대상 소프트웨어가 웹 응용이다",
- **"BPM**엔진을 사용하는 기업업무소프트웨어이다"
- "**J2EE**를 기반으로 한 클라이언트 서버 형 분산소프트웨어이다"

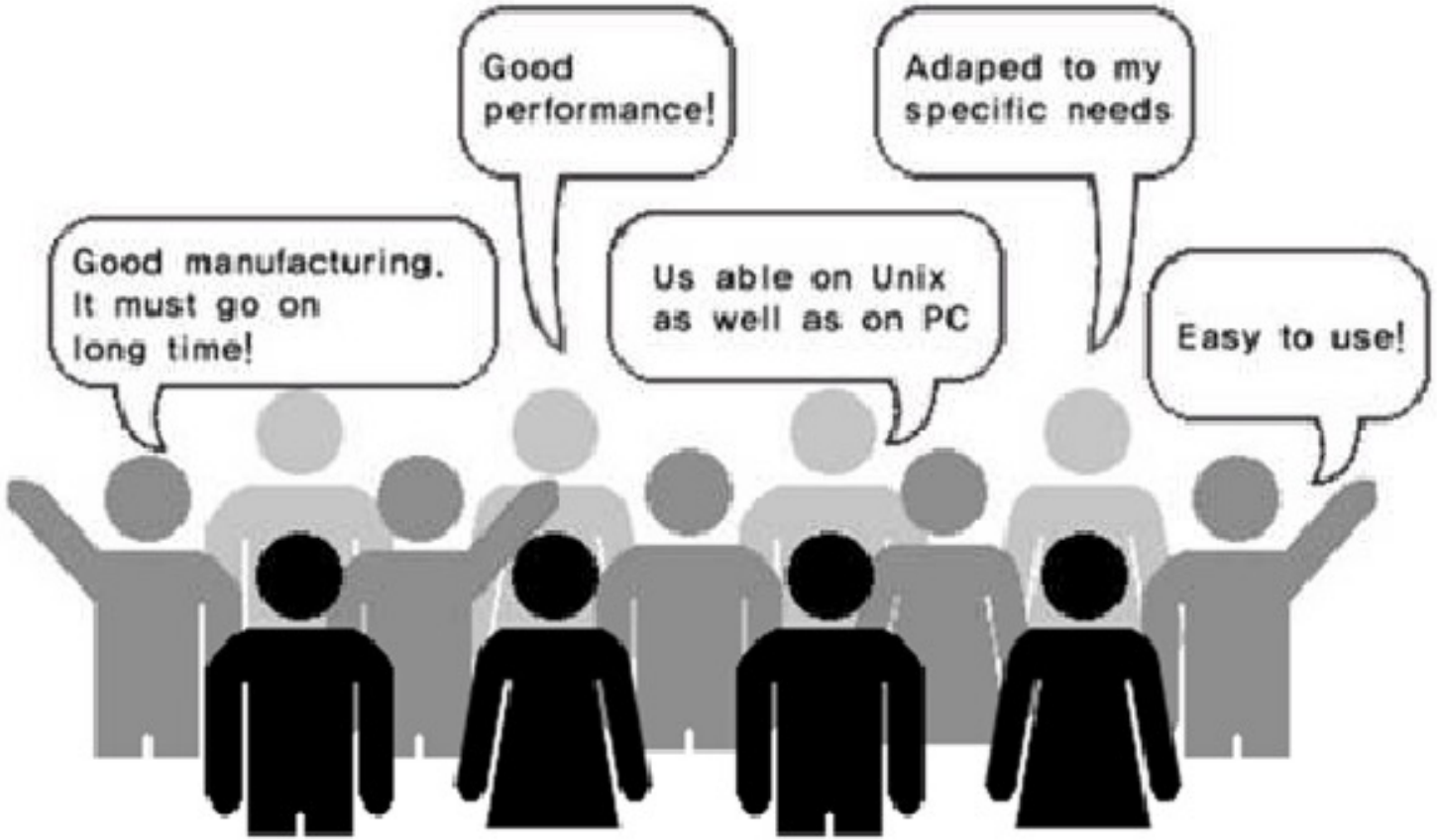


# 품질 시나리오



# Quality Attributes: Problem

- 사람들은 저마다 품질(**Quality**)을 다르게 생각한다



# Quality Attributes: Problem

- 품질 속성의 개념이 모호하여 아키텍처에 반영하기에는 부족하다.
  - 예: 우리 시스템은 유지보수성이 좋아야 한다.
- 품질 속성을 표현하는 용어가 저마다 다른 용도로 사용된다.

- 검증 가능성
  - 검증할 수 없는 주장은 의미 있는 주장이 아니다.
  - 의미 있는 요구 사항은 측정 가능하여야 한다.

# Quality Attribute Scenarios: Solution

- 개념을 명확하게 정의하지 않아서 용어의 의미가 모호함
- 어떤 문제가 어떤 품질 속성에 속하는 것인지가 불명확

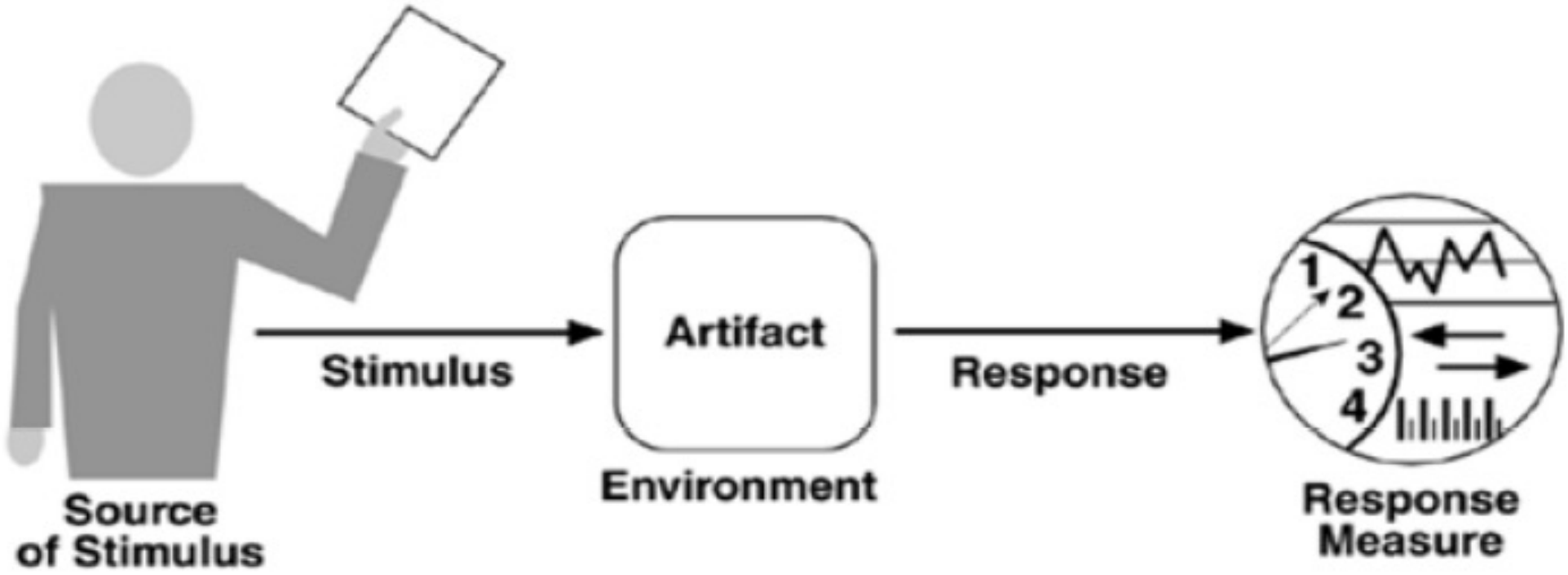
→ 품질 속성 시나리오로 극복

- 각 품질 속성은 저마다 다른 용어로 표현됨

→ 각 품질 속성마다 부연 설명

# Quality Attribute Scenarios

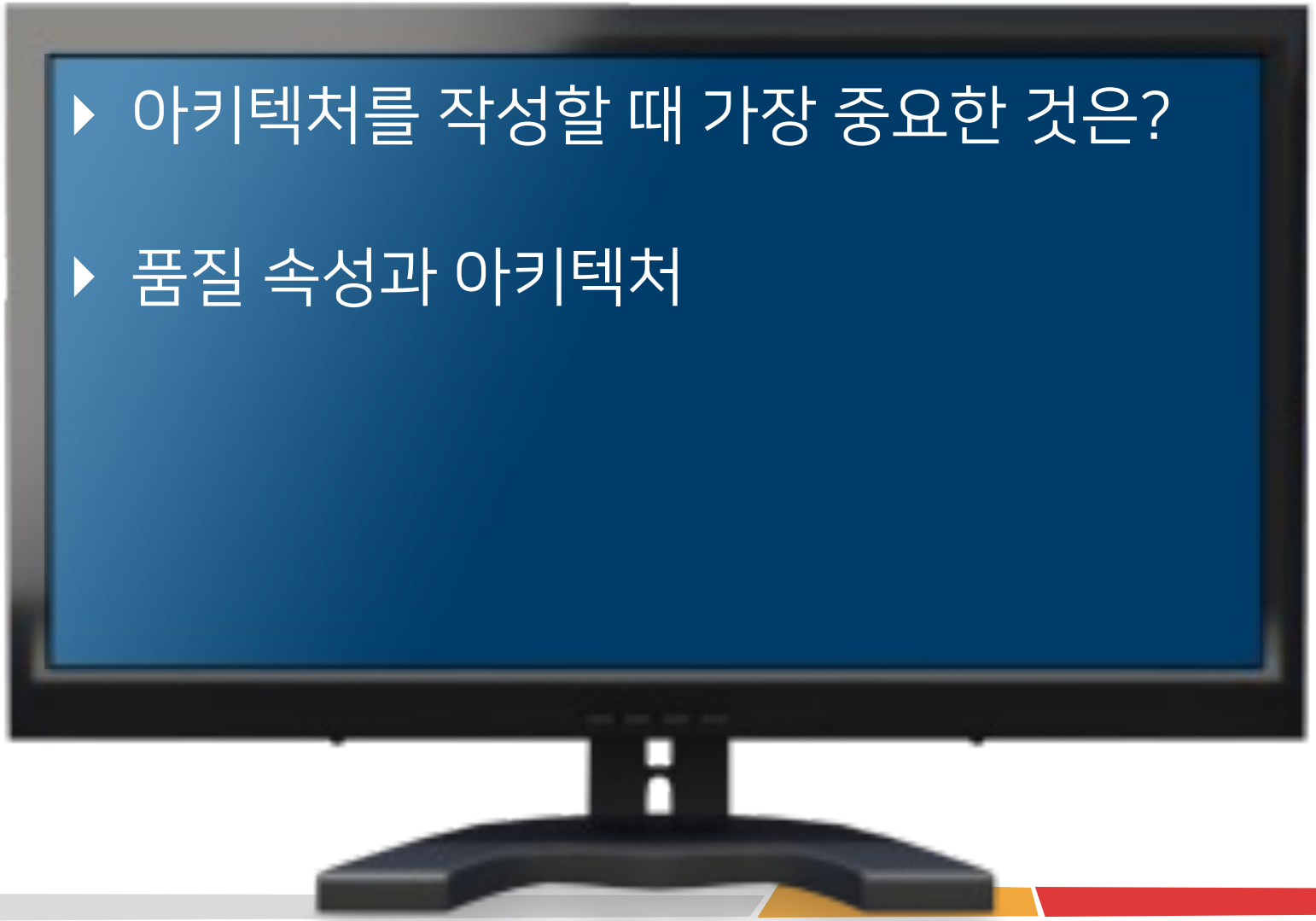
- 소프트웨어 시스템이 일정한 환경 문맥 하에서 어떻게 작동할지에 대한 기대 시나리오



# Quality Attribute Scenarios

| Component | Description |
| --- | --- |
| 자극의 근원 (Source) | 자극을 일으키는 사람 혹은 시스템<br>예: 사람, 시스템 컴포넌트, 외부 시스템 |
| 자극 (Stimulus) | 시스템이 어떤 방식으로 반응할 때 필요한 이벤트<br>품질 시나리오를 시작하는 역할. 품질 속성에 따라 다양함 |
| 환경 (Environment) | 시나리오 진행 동안 시스템이 놓여지는 운영 환경을 의미<br>예: 정상, 비정상 피크 로드, 실패 발생 상황 |
| 대상 (Artifact) | 시스템의 일부로 행위가 시나리오를 통해 특성화되는 대상<br>예: 전체 시스템이거나 특정 컴포넌트 |
| 응답 (Response) | 자극의 결과로 발생하는 대상에서의 외부적으로 보이는 행위 |
| 응답 측정 (Response Measure) | 성공적인 반응이 무엇인지를 정의하는 시나리오의 성공 기준<br>해당 기준은 구체적이고 측정 가능해야 함 |

# 품질 속성과 아키텍처



# The Most Important in Designing Software Architecture

- 아키텍처를 작성할 때 가장 중요한 것은 ?

## 품질 (Quality)

사용성 이식성 재사용성

성능 유지보수성

- Technology Trends
- Architect's Ability, Skill, Knowledge
- Team's Ability, Skill, Knowledge
- Team Organization
- …

# The Most Important in Designing Software Architecture

- 품질은 소프트웨어  아키텍처만 관련이 있는가?
  - 소프트웨어  아키텍처
  - **System(H/W, N/W)** 아키텍처
  - **Data** 아키텍처
  - **Business** 아키텍처
  - 코드
  - 알고리즘 …
- 품질과 가장 관련이 많은 것은?
  - 가장 변수가 많은 것

소프트웨어  아키텍처

# Quality Attributes and Software Architecture

- 기능과 품질은 상호 독립적**(orthogonal)**이다.

  *Orthogonal: 어떤 두 개념이 상호간에 관계없이 독립적으로 존재하는 것.*

- 품질 속성은 반드시 초기 설계에 반영이 되어야 한다.

- 아키텍처 구성에 있어 품질 속성 간 **Trade-off**가 있을 수 있다.

  *Trade-off: (서로 대립되는 요소 사이의) 균형*

# Quality Attributes and Software Architecture

- 기능과 품질은 상호 독립적(**orthogonal**)이다. **Functionality is largely orthogonal to the structure of the architecture and quality attribute requirements [Kazman 2004]**

- 두 개의 다른 시스템에 같은 기능을 구현할 수 있다.
- 두 개의 시스템이 서로 다른 구조를 가지고 있는 것은 품질 속성이 다르기 때문이다.
  - 포탈의 **communication channel**
  - 우주 왕복선의 **communication channel**

# Quality Attributes and Software Architecture

- 품질 속성은 반드시 초기 설계에 반영이 되어야 한다. **The degree to which a system meets its quality attribute requirements is dependent upon architectural decisions [Kazman 2004]**

- 이해하고 구현하기 힘든 소프트웨어 품질 속성에 대한 요구사항으로 인해 아키텍처 개념이 필요함
- 소프트웨어 제품의 품질을 향상시킬 수 있는 초기 설계 원칙 제공

- 기능 요구사항을 만족시키는 개발을 한 후 품질을 더한다?

# Quality Attributes and Software Architecture

- 아키텍처 구성에 있어 품질 속성 간 **Trade-off**가 있을 수 있다.

- 아키텍처 구성은 특정한 품질을 높이지만 특정 품질에는 안 좋을 수 있다.
- 미들웨어를 두는 것은 유지보수에 도움은 되지만 성능을 오히려 저해한다.

| | Availability | Efficiency | Installability | Integrity | Interoperability | Modifiability | Performance | Portability | Reliability | Reusability | Robustness | Safety | Scalability | Security | Usability | Verifiability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Availability | | | | | | | | | + | | + | | | | | |
| Efficiency | + | | | | – | – | + | – | | | – | | + | | – | |
| Installability | + | | | | | | | | + | | | | | + | | |
| Integrity | | | – | | – | | – | | | – | | + | | + | – | – |
| Interoperability | + | | – | – | | | – | + | + | | + | – | | – | | |
| Modifiability | + | | – | | | | – | | + | + | | | + | | | + |
| Performance | | + | | | – | – | | – | | | – | | – | | – | |
| Portability | | – | | | + | – | – | | | + | | | | – | – | + |
| Reliability | + | – | | + | | + | – | | | | + | + | | + | + | + |
| Reusability | | – | | – | + | + | – | + | | | | | | – | | + |
| Robustness | + | – | + | + | + | | – | | + | | | + | + | + | + | |
| Safety | | – | | + | + | | – | | | | + | | | + | – | – |
| Scalability | + | + | | + | | | + | + | + | | + | | | | | |
| Security | + | | | + | + | | – | – | + | | + | + | | | – | – |
| Usability | | – | + | | | | – | – | + | | + | + | | | | – |
| Verifiability | + | | + | + | | + | | | + | + | + | + | | + | + | |

https://medium.com/analysts-corner/those-other-requirements-quality-attributes-and-their-inescapable-tradeoffs-31dc0691974d

# Questions

- 기능과 품질은 [ ]이다.
- 품질 속성은 반드시 [ ]에 반영이 되어야 한다.
- 아키텍처 구성에 있어 품질 속성 간 [ ]가 있을 수 있다.

# Question?

Seonah Lee
saleese@gmail.com