Spring 2023

Software Architecture Design: Introduction

Seonah Lee

Gyeongsang National University

Design It! From Programmer to Software Architect, The Pragmatic Bookshelf, 2017

목 차

기본적인 설계 생각하기


설계 사고의 4가지 원칙
Human rule
Ambiguity rule
Redesign rule
Tangibility rule

Four Principles of Design Thinking

- 설계 사고(Design Thinking)
  - 문제와 해결책을 영향을 받는 사람의 관점에서 생각하는 방식
- 4가지 원칙
  - **Human rule** – 모든 디자인은 사회적인 영향을 미친다
  - **Ambiguity rule** – 옵션을 열어두기 위한 일부 애매함은 허용한다
  - **Redesign rule** – 기존에 유사하게 설계된 솔루션을 참조할 수 있다
  - **Tangibility rule** – 아키텍처는 명확하게 표현해야 한다

Four Principles of Design Thinking

Design for Humans

- 설계는 기본적으로 사람 중심의 노력이다
- 사람들을 위해, 사람들과 함께 설계한다
- 모든 설계 결정은 개인들을 돕는다
- 설계는 사회적인 활동으로 팀의 중심에 있어야 한다
- 아키텍처와 상호작용하는 사람에게 초점을 두는 것은, 더 나은 설계자, 소통자, 리더가 되게 한다

Four Principles of Design Thinking

Preserve Ambiguity

- 공학에서 애매함은 위험하다.
  - 일단 설계에 대한 결정이 되면, 명확하고 명료해야 한다.
  - 요구사항, 설계 결정 등은 애매하게 두어서는 안된다.
- 그러나, 옵션을 열어두기 위한 일부 애매함을 사용할 수 있다
  - 아키텍처는 기대 품질 속성을 만족시키기 위해 구조를 배열하는 것이다
  - 이를 위한 최소한의 아키텍처 설계와 위험 감소에만 초점을 맞출 수 있다
  - 나머지 설계 결정은 그 다음 작업의 설계자에게 남겨둘 수 있다

Four Principles of Design Thinking

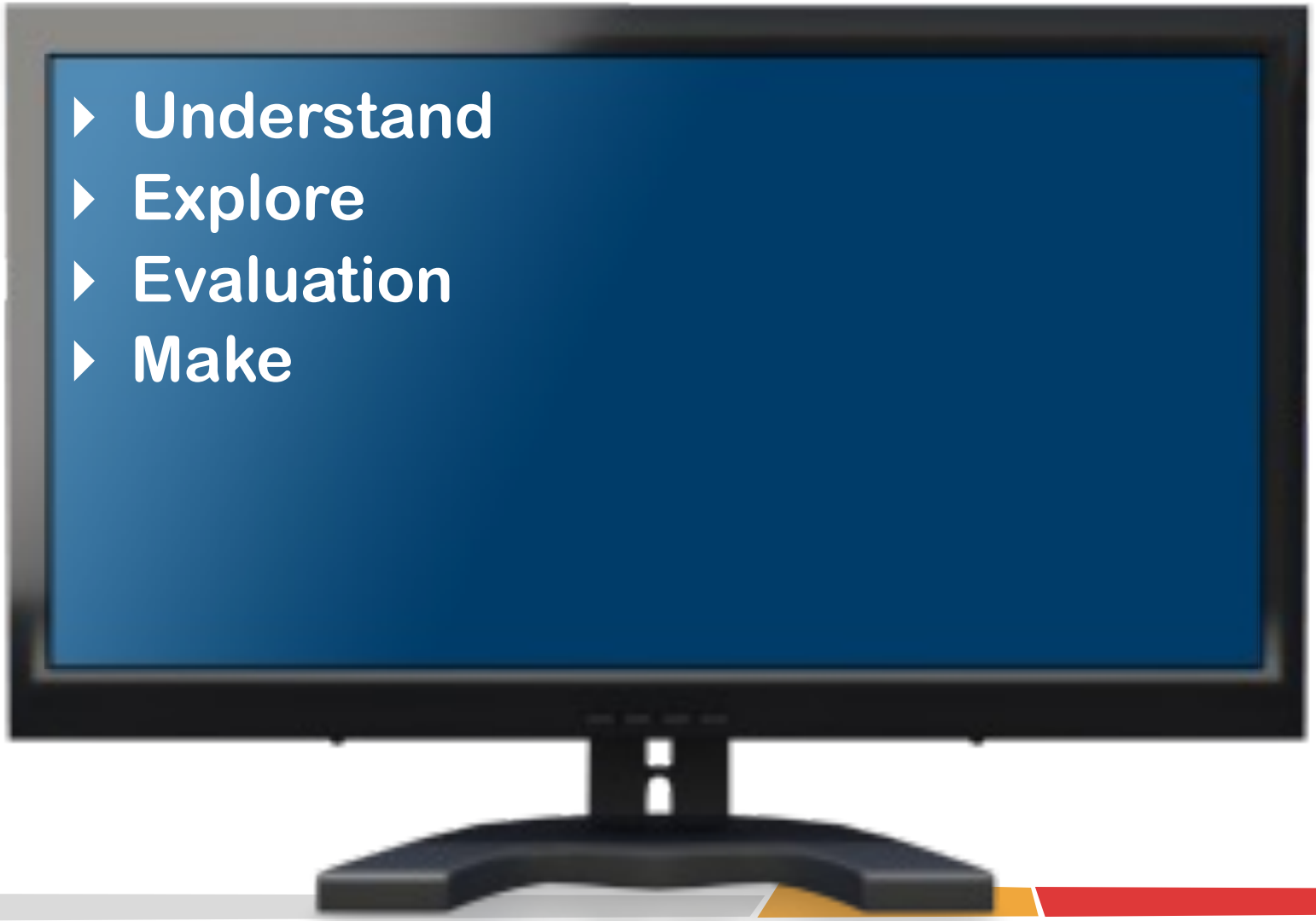
Design is Redesign

- 기존의 문제에 대한 솔루션 패턴을 집대성하면, 유사한 문제에 대해서 유사한 솔루션을 적용할 수 있다
- 다른 팀 또한 우리 팀과 유사한 문제를 겪었을 가능성이 있다
- 우리가 풀어야 할 문제에 대해 누군가 패턴을 만들어 놓았을 수 있다
- 혹은 유사한 문제에 대한 프레임워크를 만들어 놓았을 수 있다
- 기존에 유사하게 설계된 솔루션을 참조하거나 고도화할 수 있다

설계 사고의 4가지 원칙

- **Make the Architecture Tangible**
  - 아키텍처 구조가 코드에만 있다면, 아키텍처가 잘 보이지 않는다
  - 코드는 읽기 힘들고, 품질 속성, 설계 이유 등을 설명하지 않는다
  - 다른 사람과 아키텍처를 공유하려면, 더 구체화해야 한다.
  - 아키텍처는 명확하게 표현하기 위한 여러 방법이 있다.
    - 그리거나, 프로토타이핑하여 해당 구조에서의 품질을 경험하게 할 수 있다
    - 모델을 만들어 어떤 식으로 작동하는지 보여줄 수 있다

설계 마인드 셋



Four Design Mindsets

Understand

Explore

Evaluate

Make

Four Design Mindsets

Understand the Problem

- 문제를 이해하기 위하여 관련자들로부터 정보를 수집한다
  - 이를 위하여 시스템 관련자들의 필요를 이해해야 한다
  - 관련자들이 중시하는 비즈니스 목표와 품질 속성을 명시한다
  - 설계 결정의 우선순위와 트레이드 오프를 이해한다

Four Design Mindsets

Explore Ideas

- 여러 설계 개념을 생성하고 공학적인 접근을 명시한다
  - 다양한 패턴 탐색
  - 기술 탐색
  - 개발 프랙틱스 탐색
- 주의! 관련자들과 함께 작업할 때 의미가 있다

Four Design Mindsets

Make it Real

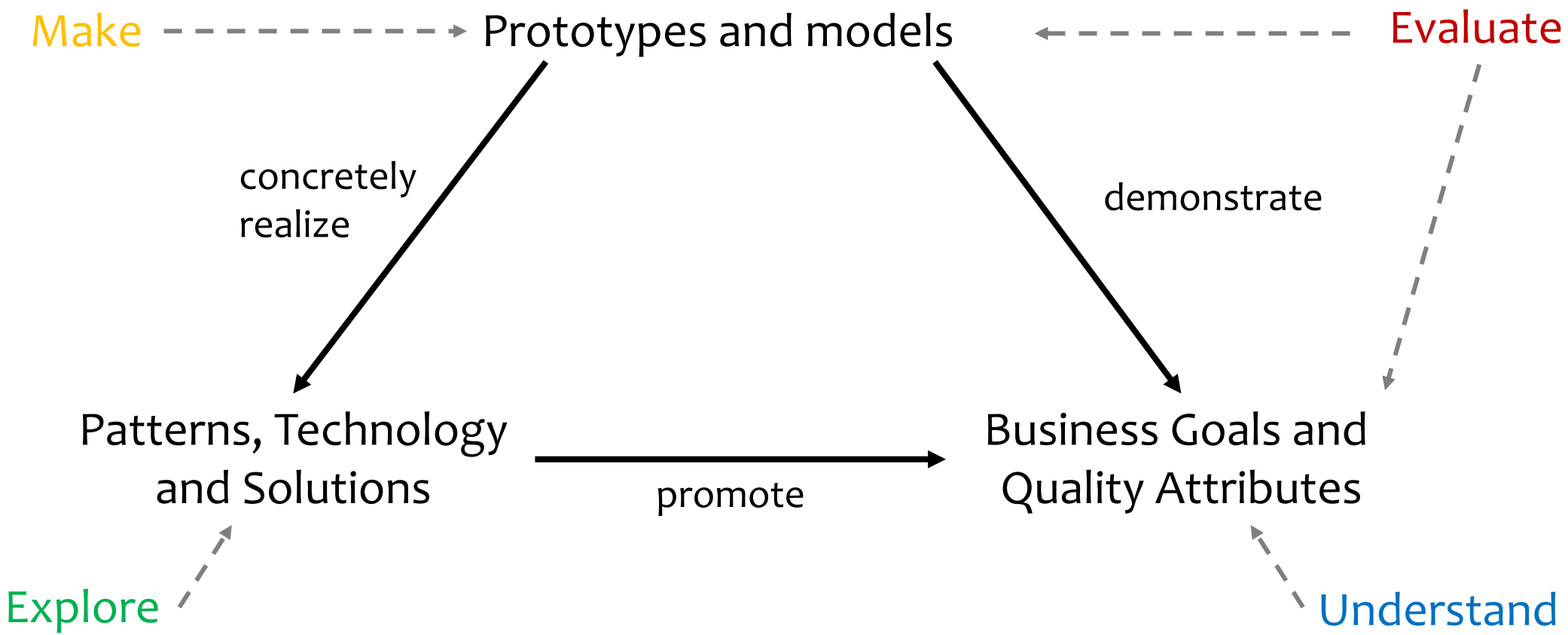
- 아이디어를 구체적인 산출물로 표현하여 공유해야 한다
  - 모델을 만들고, 프로토타이핑을 하고, 문서를 작성하고, 수치를 계산하며 다양한 접근 방법으로 구체화한다
  - 계획에 대한 소통을 위해서 아키텍처를 구체화하여야 한다

Four Design Mindsets

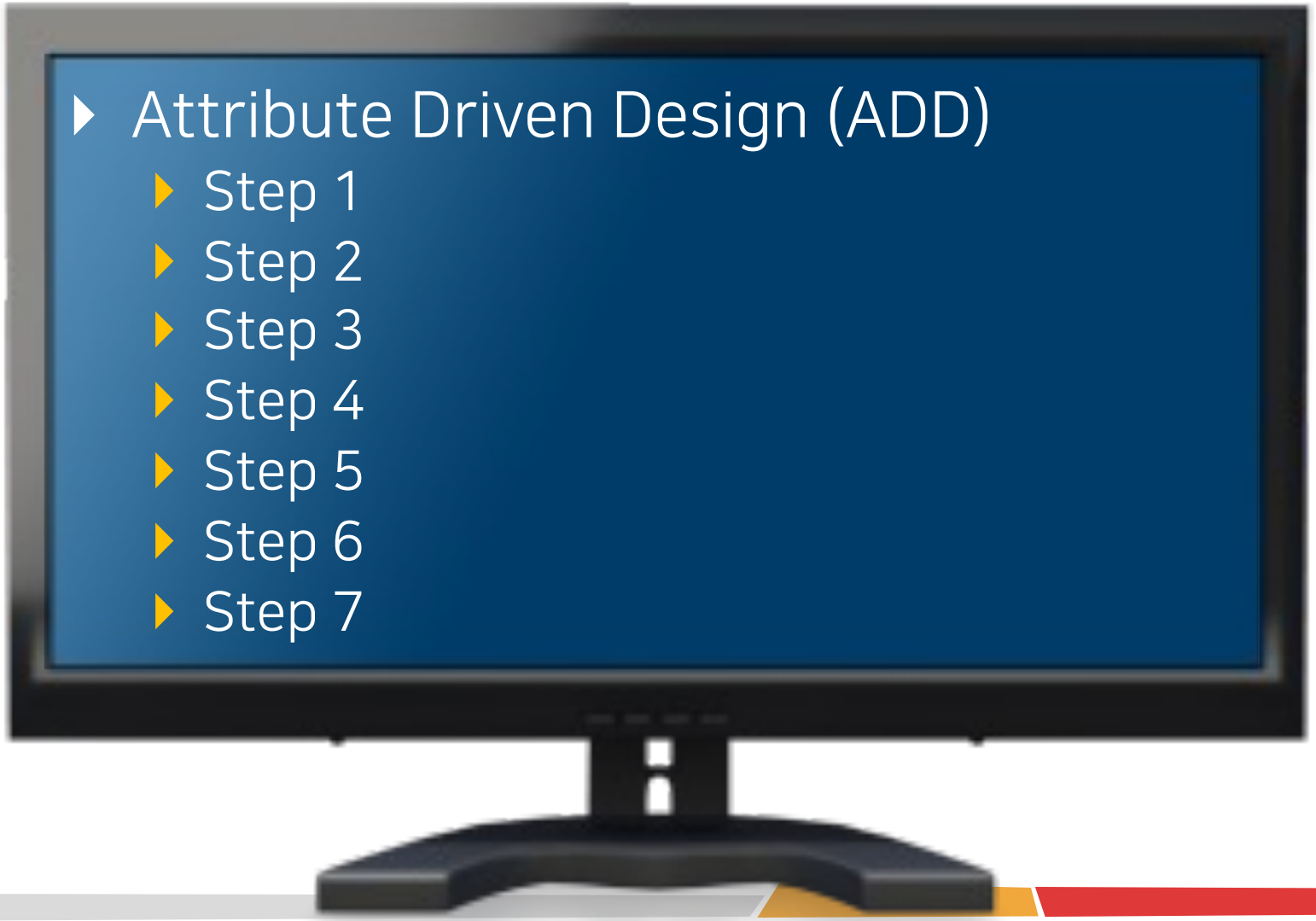
Evaluate Fit

- 설계 결정이 현재의 문제를 풀 수 있는지를 검증해야 한다
  - 일반적인 방법은 여러 시나리오를 활용하여 아키텍처를 돌려봄으로써 검증하는 것이다
  - 혹은 실험이나 결정과 관련된 위험을 조사하여 설계 결정을 시험한다

설계의 마인드 셋


Make
Prototypes and models
Evaluate
concretely
realize
demonstrate
Patterns, Technology
and Solutions
promote
Business Goals and
Quality Attributes
Explore
Understand

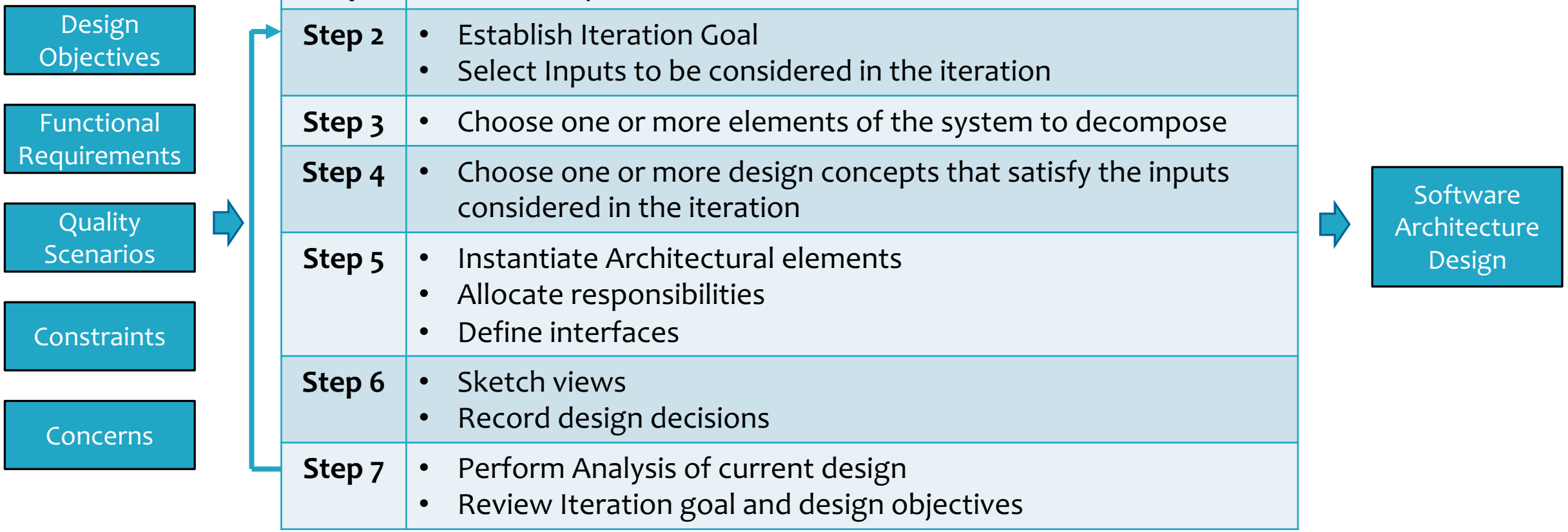
아키텍처 설계


Attribute Driven Design (ADD)
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 7

Attribute Driven Design

- An architecture design method "driven" by quality attribute concerns
- The method promotes an iterative approach to design
- It provides a detailed set of steps for architecture design
  - It enables design to be performed in a systematic, repeatable way
  - It leads to predictable outcomes.

Attribute Driven Design (3.0)



Attribute Driven Design (3.0)

- Step 1
  - Review Inputs and ensure that there is clarity on the overall design problem that needs to be solved
    - Design objectives
    - Functional requirements
    - Quality Attribute Scenarios
    - Constraints
    - Concerns
- Step 2
  - Establish Iteration Goal
    - The design problem is divided into several sub problem
  - Select Inputs to be considered in the iteration
    - An iteration starts by deciding which sub problem to address

Step 3-5

- 3 types of decisions are made to address the subproblem:
  - Selection of the parts that need to be decomposed
    - Step 3
  - Identification and selection of existing solutions that support the decomposition
    - Step 4
  - Creation of elements from the existing solution and establishment of their responsibilities and interfaces
    - Step 5

Step 3-4

- Step 3
  - Choose one or more elements of the system to decompose
  - 3 types of decisions are made to address the subproblem:
    - Selection of the parts that need to be decomposed
- Step 4
  - Choose one or more design concepts that satisfy the inputs considered in the iteration
  - 3 types of decisions are made to address the subproblem:
    - Identification and selection of existing solutions that support the decomposition

Step 4***

- Most sub-problems that are addressed during an iteration can be solved using existing solutions, i.e. design concepts
  - We want to avoid re-inventing the wheel
  - It is better (and faster) to use a proven solution to a problem for which we may not be experts
  - Creativity in design involves identifying, adapting and combining them

Step 4***

- There are several categories of design concepts, some are abstract and some more concrete.
- Here we consider:
  - Reference Architectures
  - Deployment Patterns
  - Architectural / Design Patterns
  - Tactics
  - Externally developed components (e.g. Frameworks)

vs.

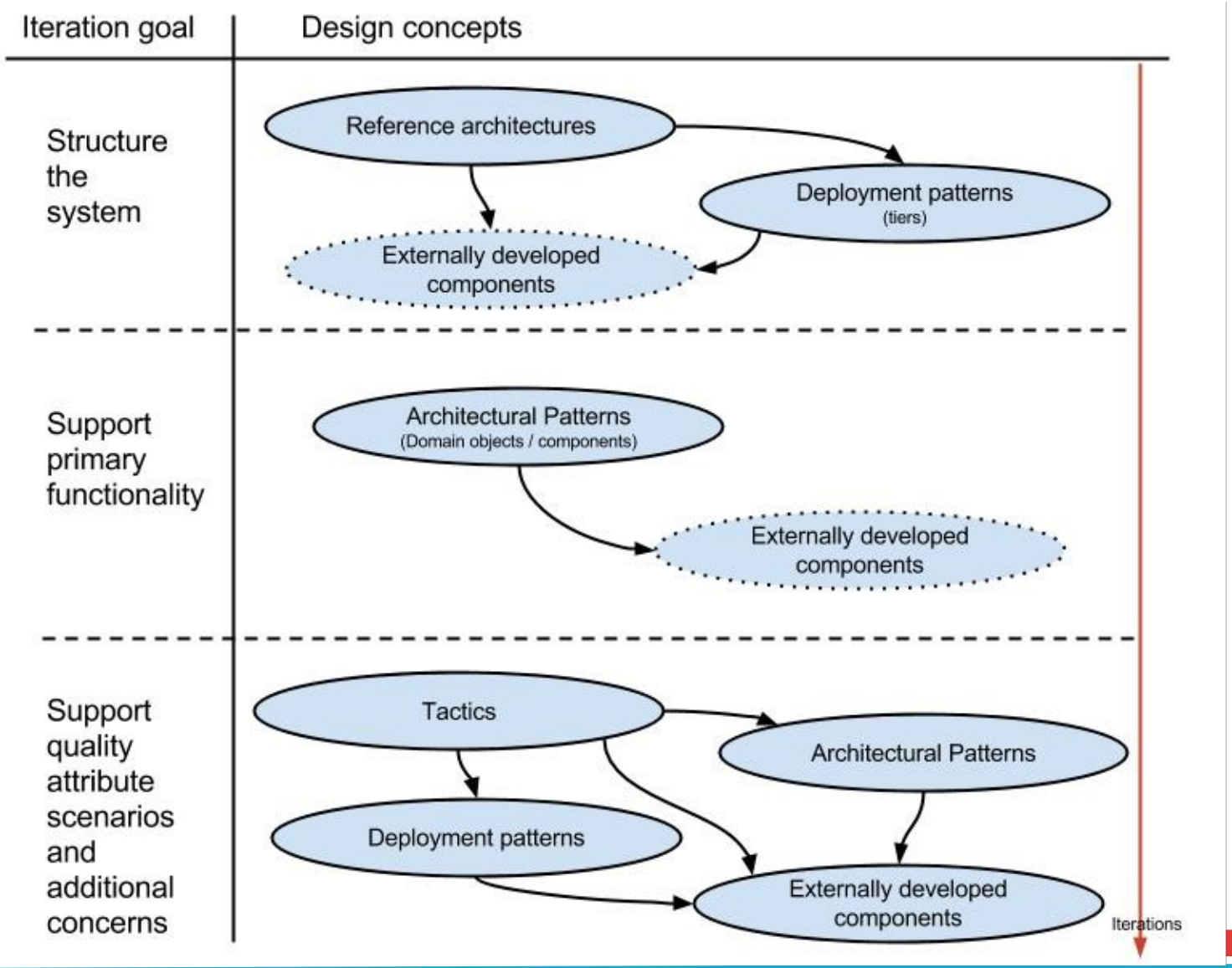
Step 5-6

- Step 5
  - Instantiate Architectural elements
  - Allocate responsibilities
  - Define interfaces
  - 3 types of decisions are made to address the subproblem:
    - Creation of elements from the existing solution and establishment of their responsibilities and interfaces
- Step 6
  - Sketch views
    - The “blueprint” is refined.
    - This may be done in parallel with step 5.
      - Note: This is not full blown documentation but rather sketches.
  - Record design decisions

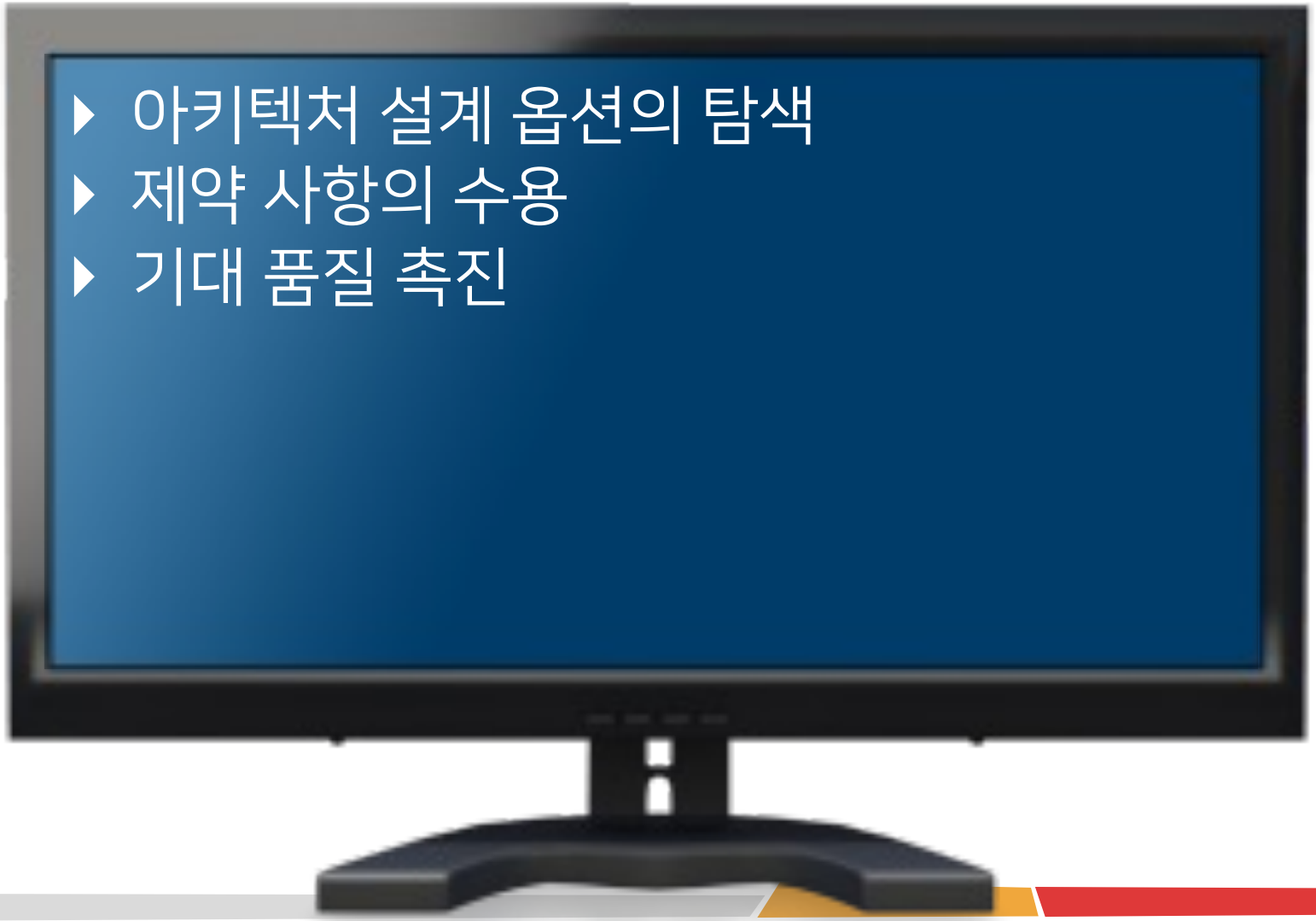
Step 7

- Perform Analysis of current design
  - Decisions made at this point are analyzed along with the advances in the overall design process
  - It is to decide if more iterations are necessary
- Review Iteration goal and design objectives
  - The design is produced.
  - Note: This may be only a partial architecture design and is not Big Design Up Front (BDUF)!

Design Selection Roadmap

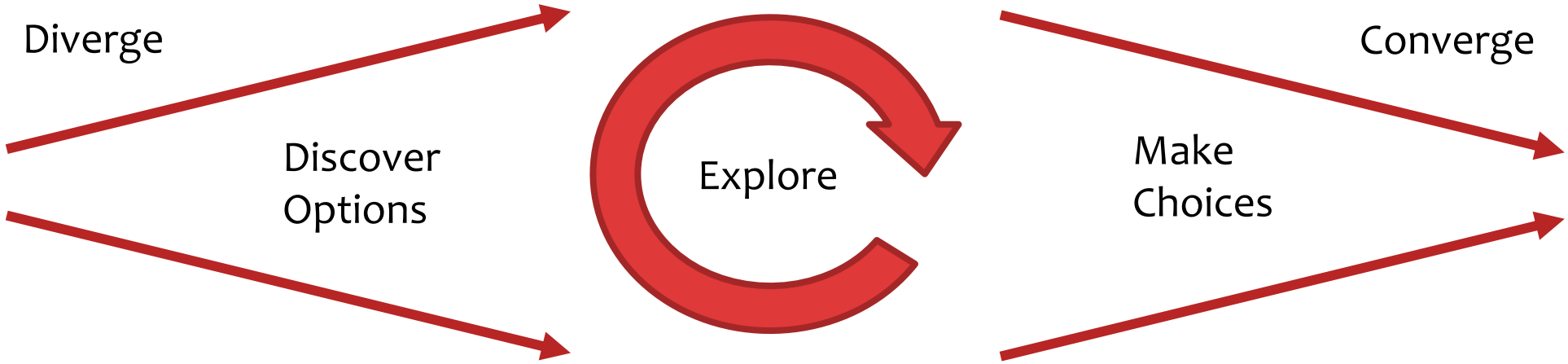


아키텍처 선택하기



아키텍처 설계 옵션의 탐색

- **Explore the design space**
- **Eliminate options that are a poor fit for the current problem**



아키텍처 설계 옵션의 탐색

- 아키텍처의 일반 형태를 구성하기 위한 요소들과 책임의 탐색
- 요소들의 상호작용을 결정짓기 위한 관계와 인터페이스 탐색
- 도메인 모델의 이해와 탐색
- 기대 품질을 촉진할 기술과 프레임워크의 탐색
- 구축과 배포 방식의 탐색
- 기존 설계의 탐색

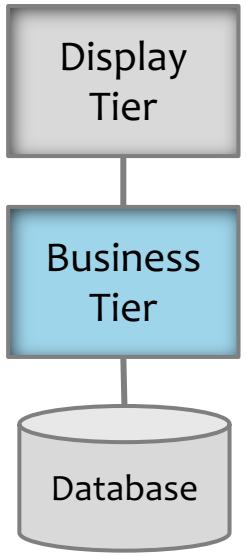
제약 사항의 수용

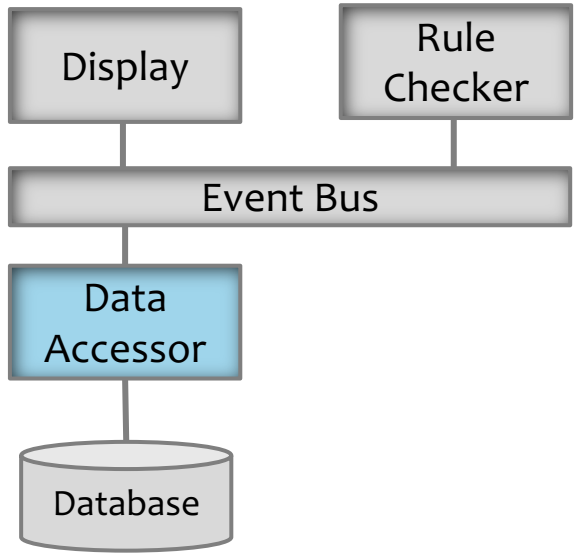
- 기술적 제약사항, 인력, 비용, 일정, 프로세스 등의 수용
  - 동시 개발을 수행할 수 있도록 하는 방식 선택
  - 증분의 배포를 가능하게 하는 패턴 선택
  - 위험을 줄일 수 있는 기술 선택
  - 자동화와 개발 가속화를 할 수 있는 기술 선택
  - 기술적 부채를 승인하거나 일부 계획을 미루는 등의 선택 등

기대 품질 촉진

	Standard Blender	Hand Blender
Cleanability	Neutral	Positive
Quietness	Neutral	Positive
Power	Neutral	Negative
Portability	Negative	Positive

기대 품질 촉진





	3-Tier	Publish-Subscribe
Availability	+	O
Performance	O	-
Security	O	-
Scalability	O	O

Summary: SW Architecture Design?

- 아키텍처 설계는 사람 중심의 관점이 시작점
  - **Design for humans is important**
- 아키텍처 설계에서 문제와 솔루션이 일관되고 구체적인 것이 중요
  - **In SW architecture design, it is important to be consistent between problems and solutions**
- 아키텍처 설계는 점진적이고 점증적인 방법임
  - **SW architecture design is iterative and incremental**
- 아키텍처 설계에서의 선택은 기대 품질에 기반함
  - **Architecture design is determined based on the desired qualities**

Question?

Seonah Lee

saleese@gmail.com