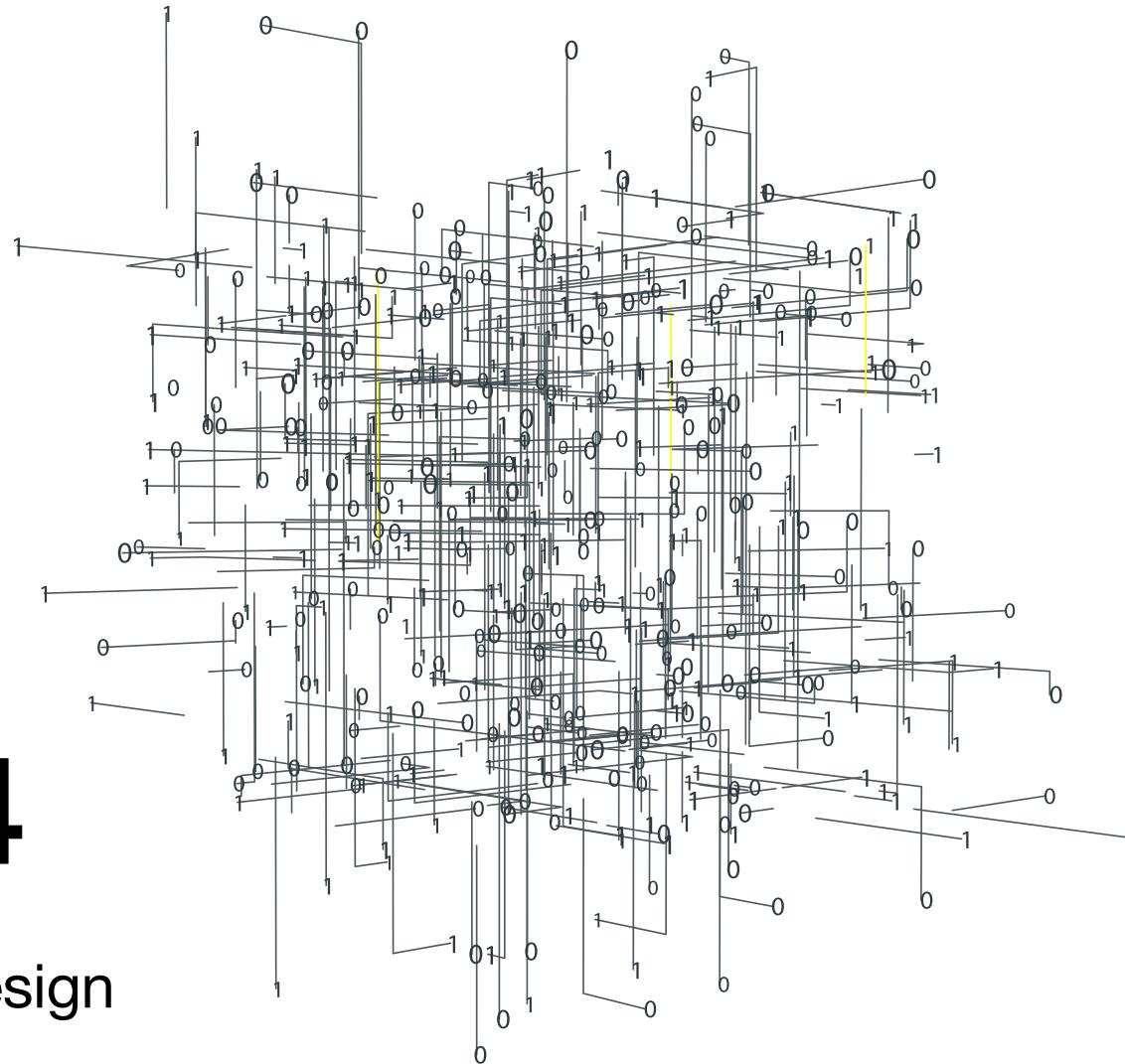
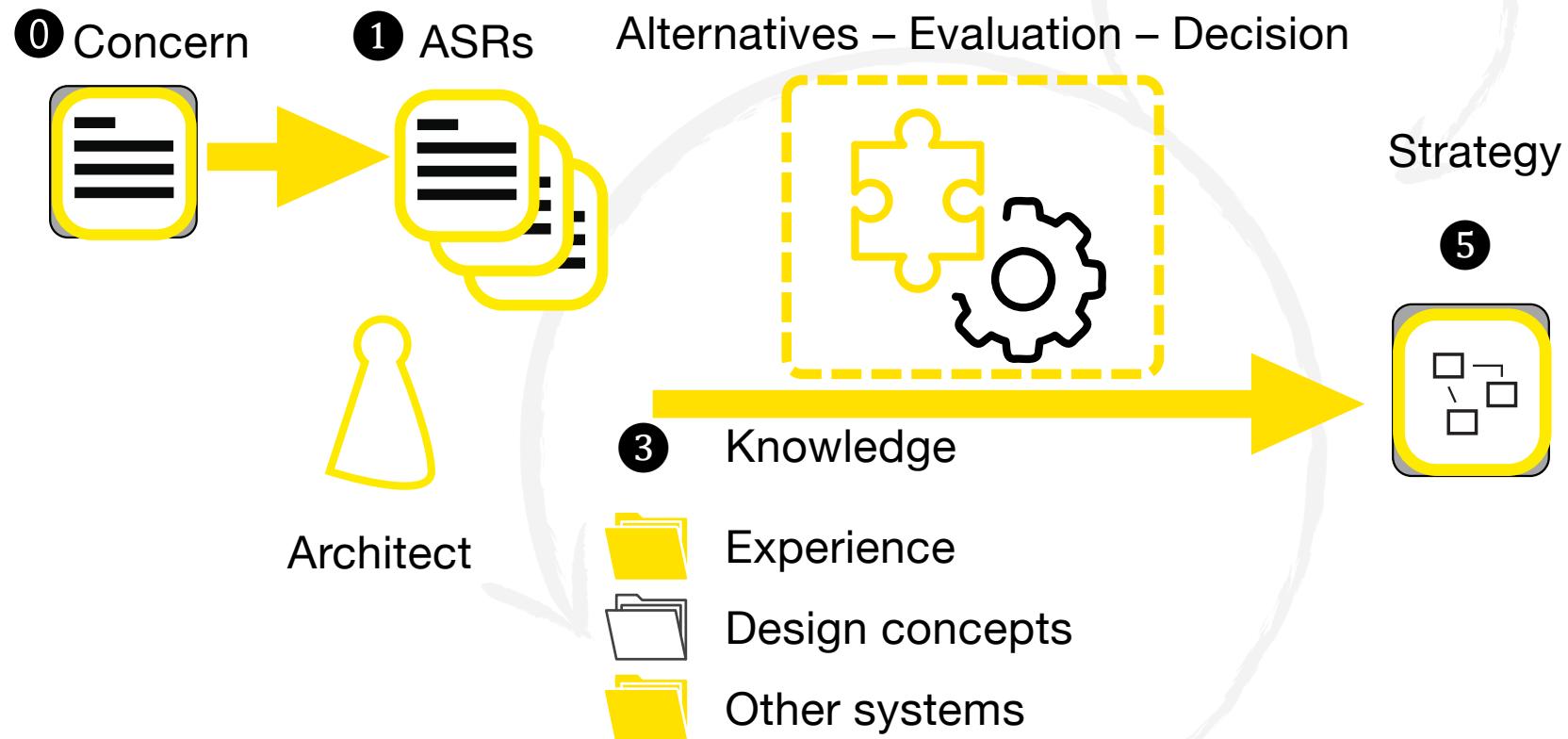


# 2DV604

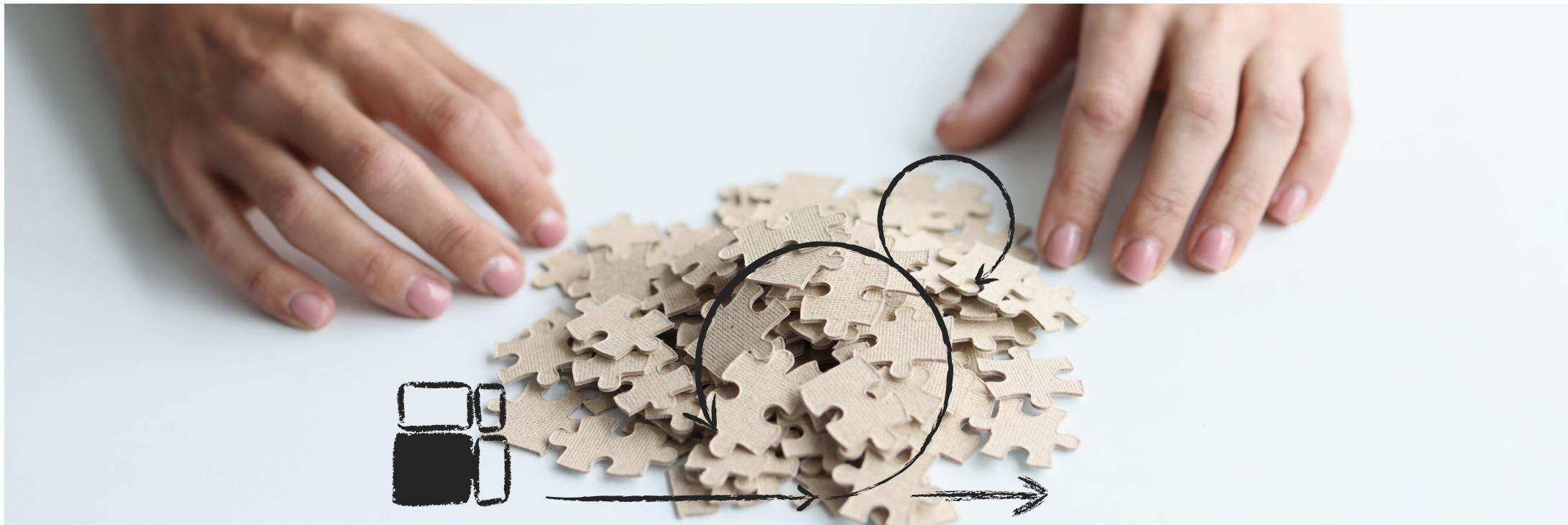
Attribute Driven Design



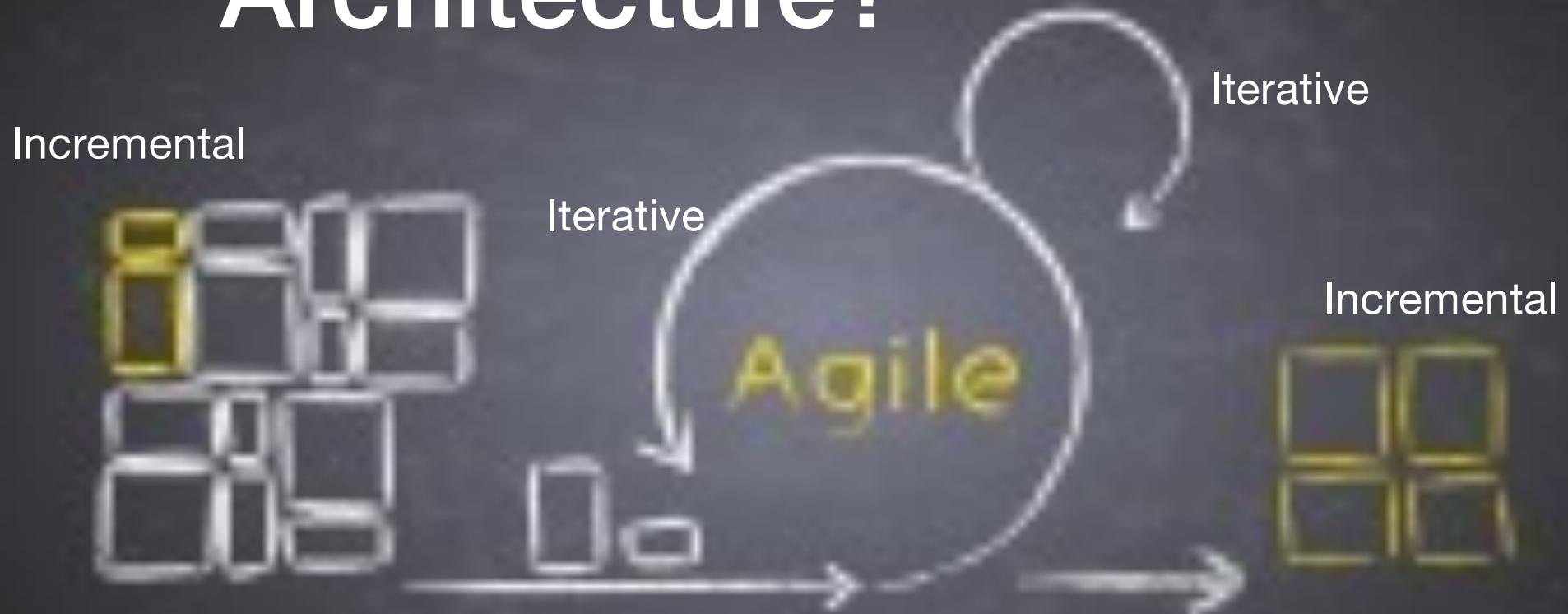
# Today – a Design method



# Iterative & Incremental



# Architecture?



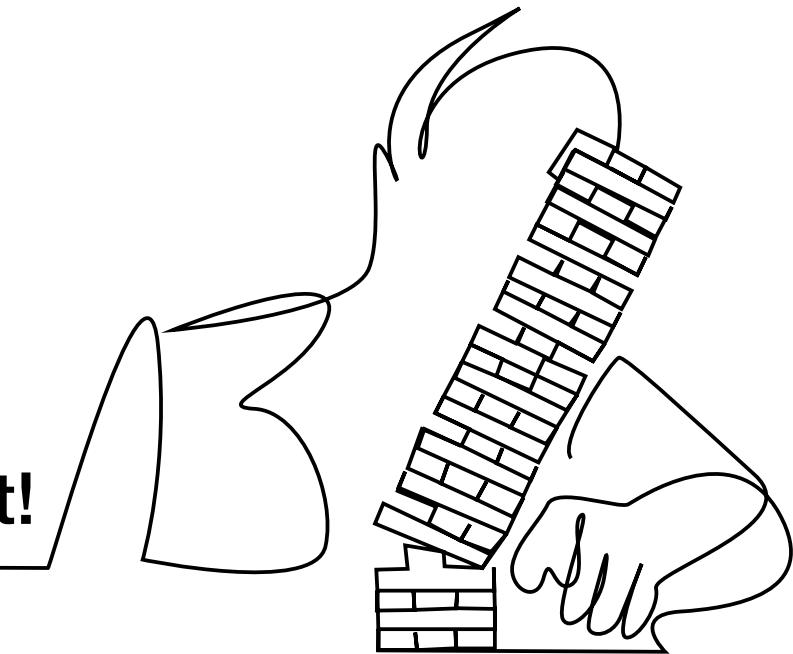
# Architecture is not a Phase!

The principal designs decisions

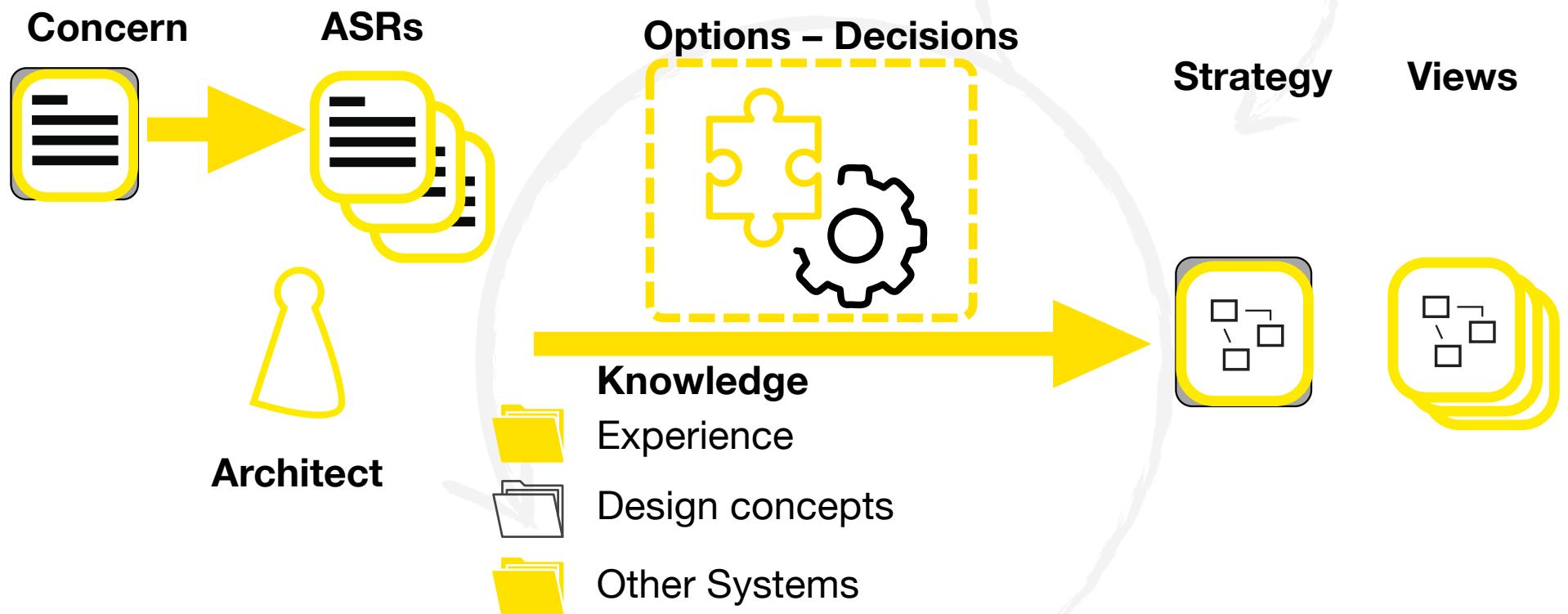
Decisions throughout the life-cycle

It's not a one-shot activity

**Avoid the Big Design Upfront!**



# Agile Architecture Design

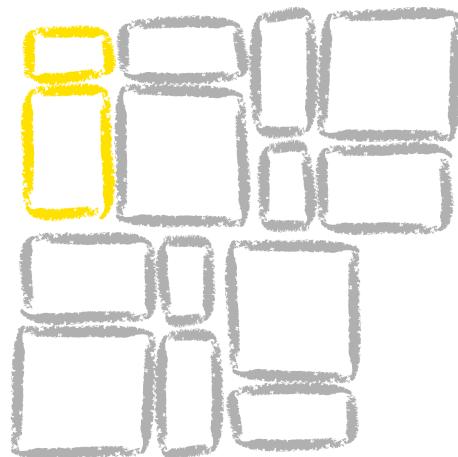


Mapping Architecture Decision steps onto and Agile Process

# Create an Architectural Runway

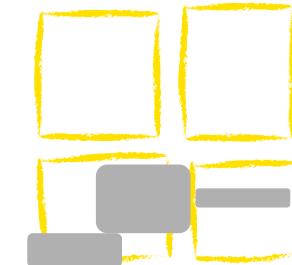
The Architectural Runway consists of the existing code, components, and technical infrastructure needed to implement near-term features without excessive redesign and delay.

Find  
Concerns and ASRs



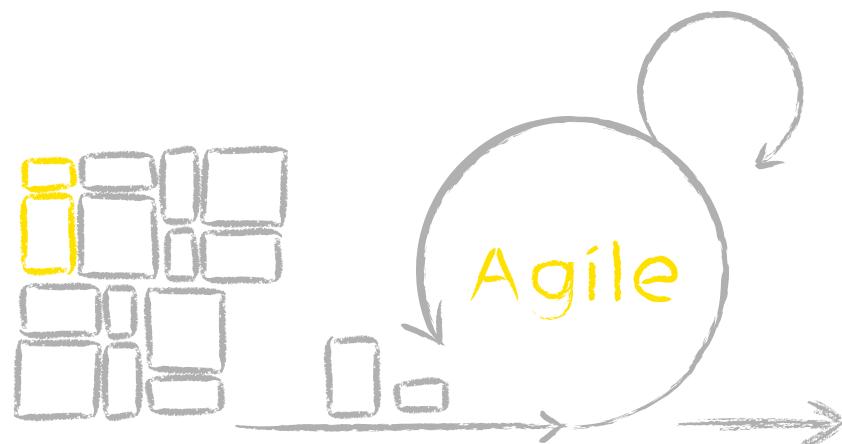
Agile

© Scaled Agile,  
Inc.



Architectural increments

# Architectural Runway



Spikes

Stories from the backlog

Chores



Architect



Team

Assess the runway

Design strategy

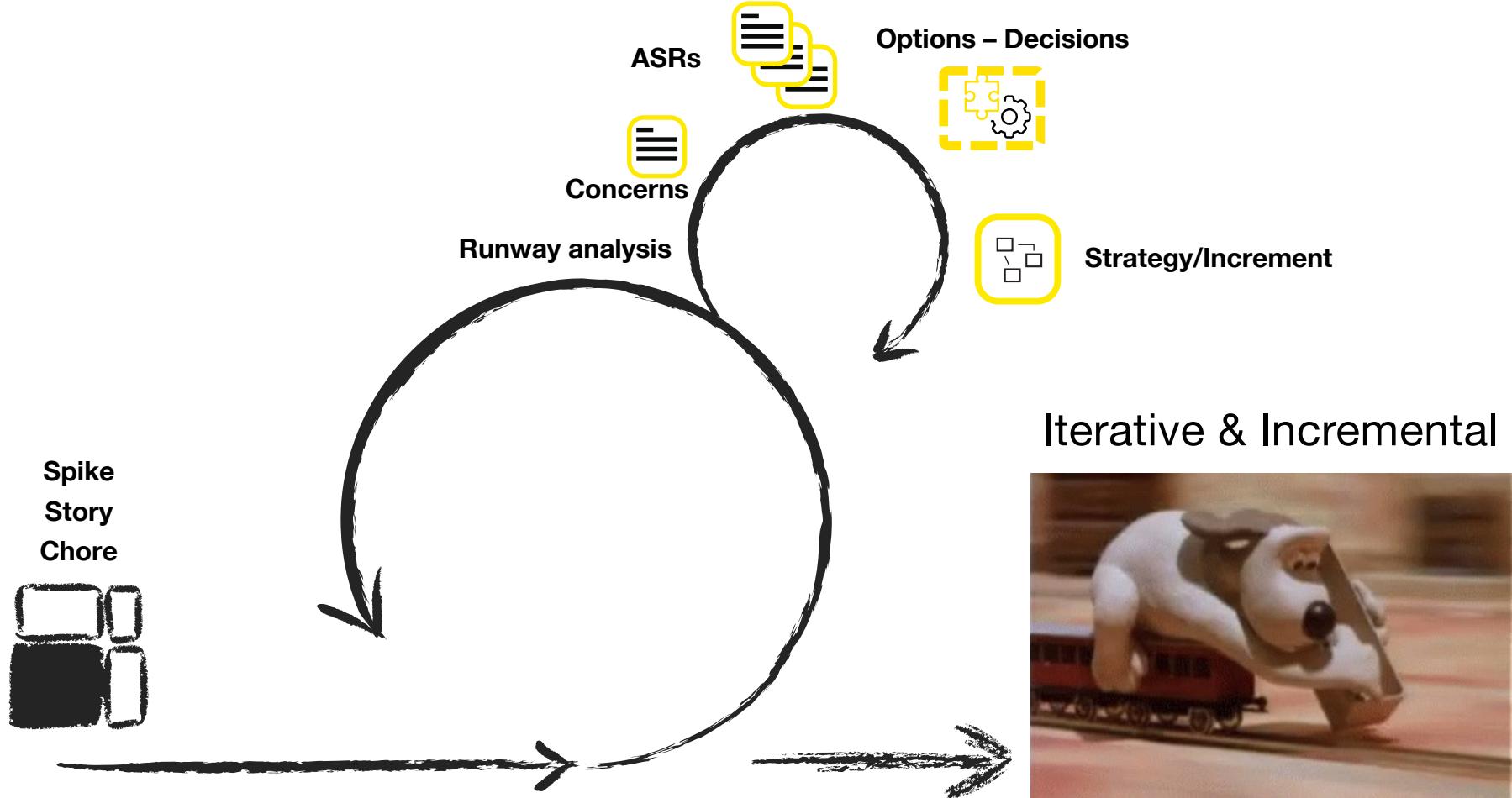
Architectural increment

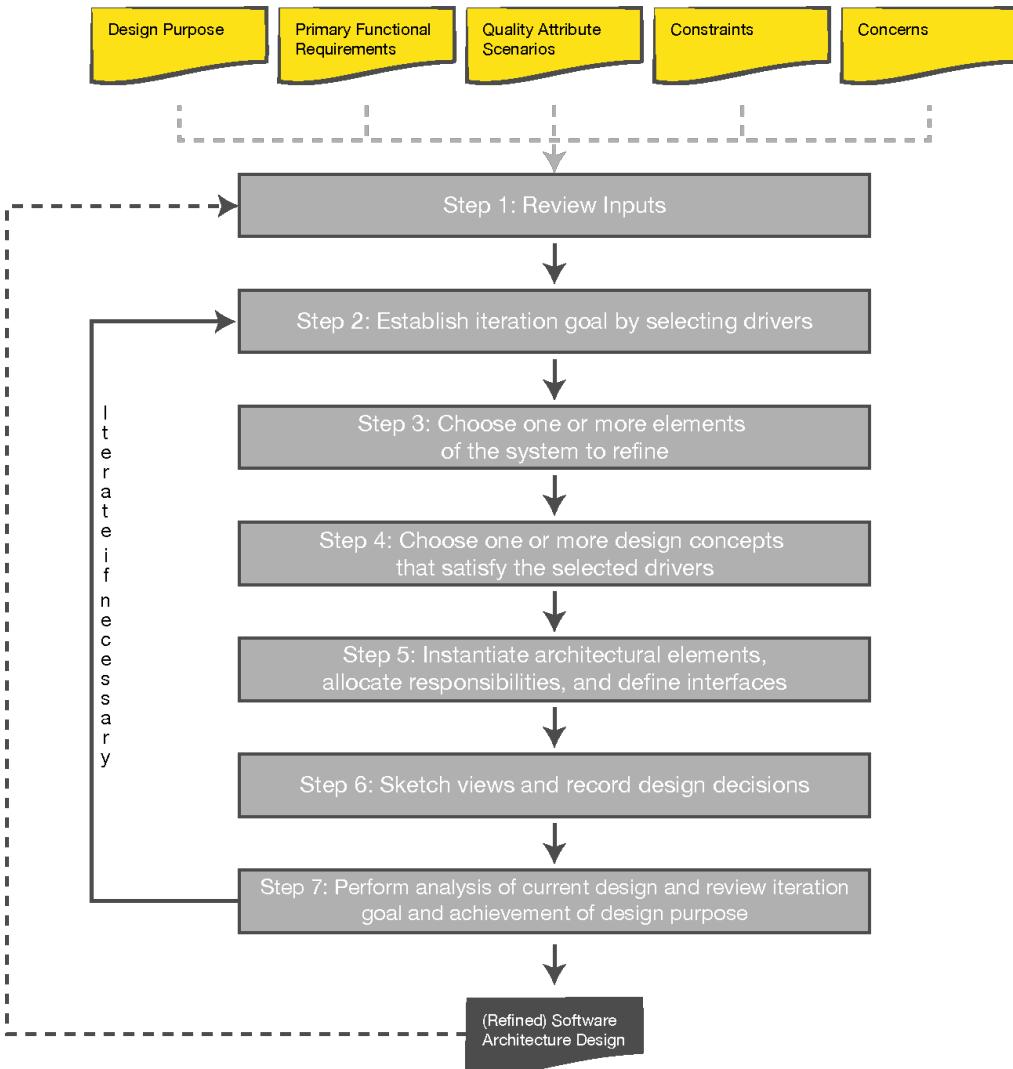
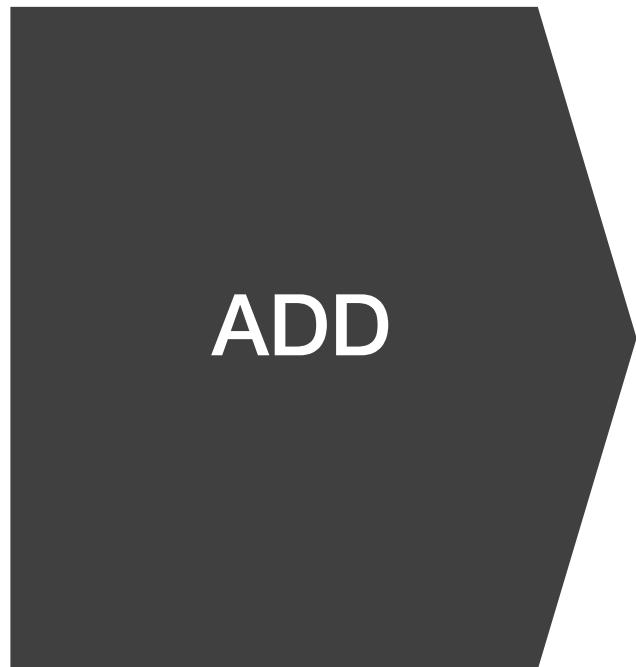
Runway realization



Runway ready for  
Feature stories

# Architectural Increments







Nice if users could interact with our stores  
and get support!

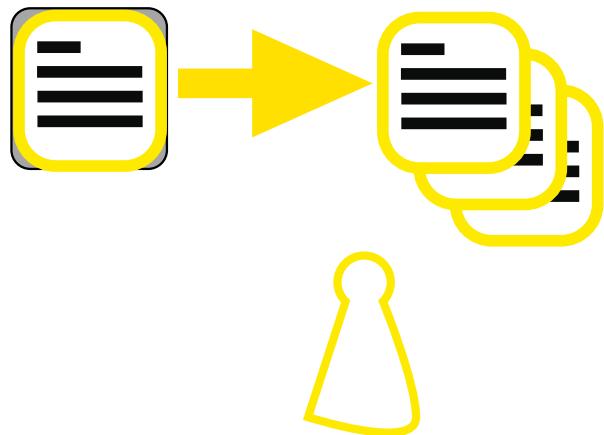
It would be great if I could keep track of the  
business in real time!



Jed - Owner

# Let's start here!

Concern



ASRs

It would be great if I could keep track of the business in real time!



Nice if users could interact with our stores and get support!

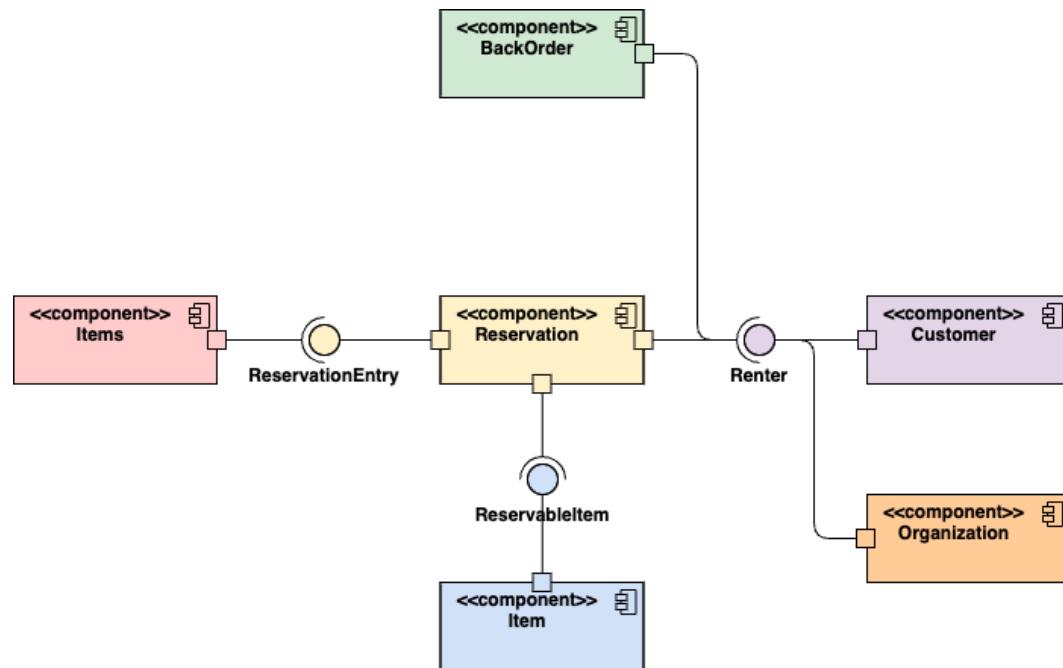
Functionality?

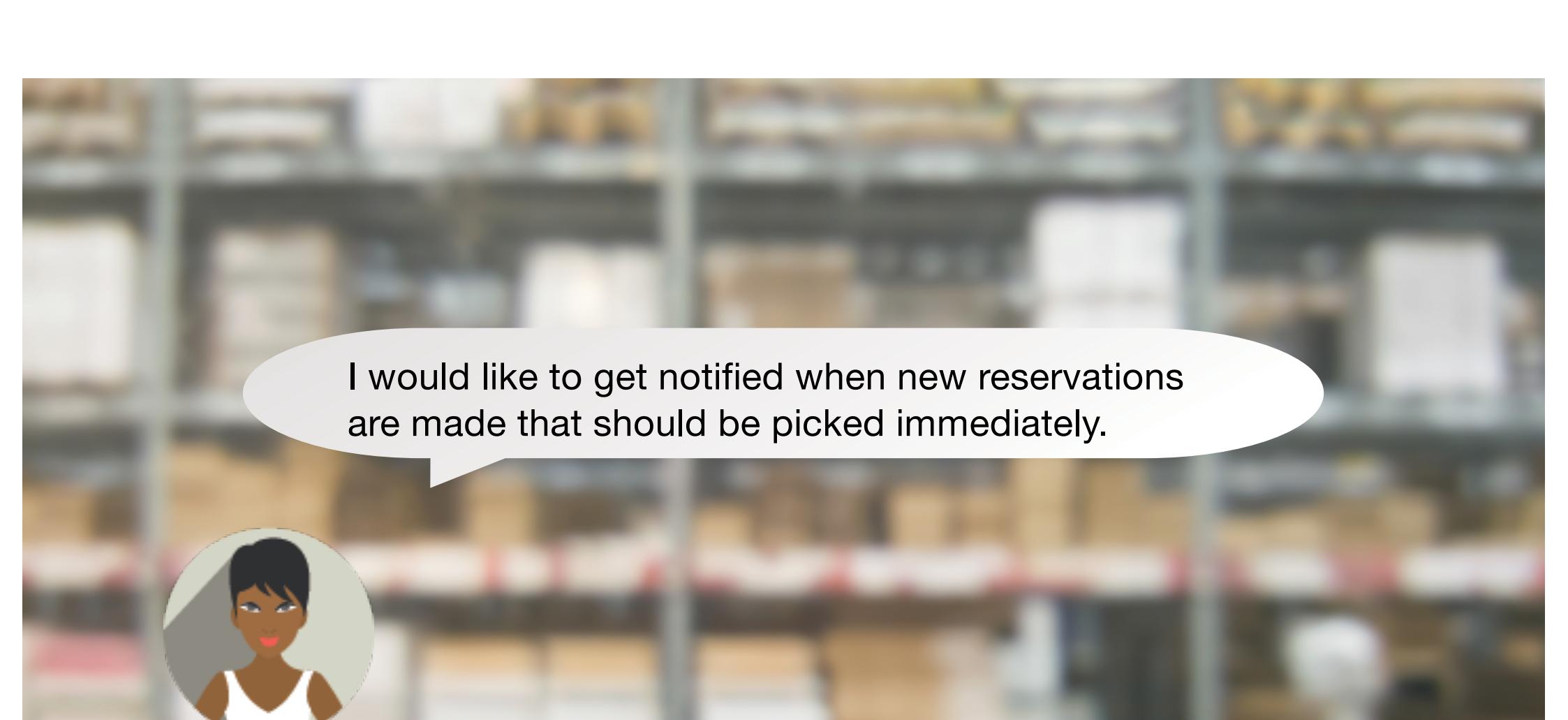
Architecture significant?

# What is going on?



It would be great if I could keep track of the business in real time!





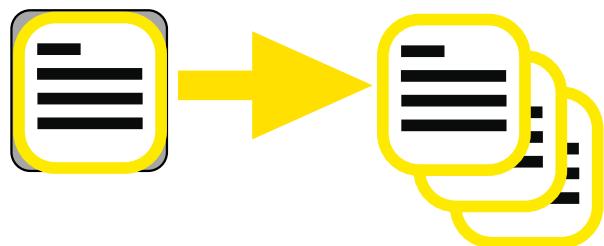
I would like to get notified when new reservations  
are made that should be picked immediately.



# Judy – Head of Operations

# and Here?

Concern



Architect



Nice if users could interact with our stores and get support!

It would be great if I could keep track of the business in real time!



I would like to get notified when new reservations are made that should be picked immediately.



Functionality?

Architecture significant?





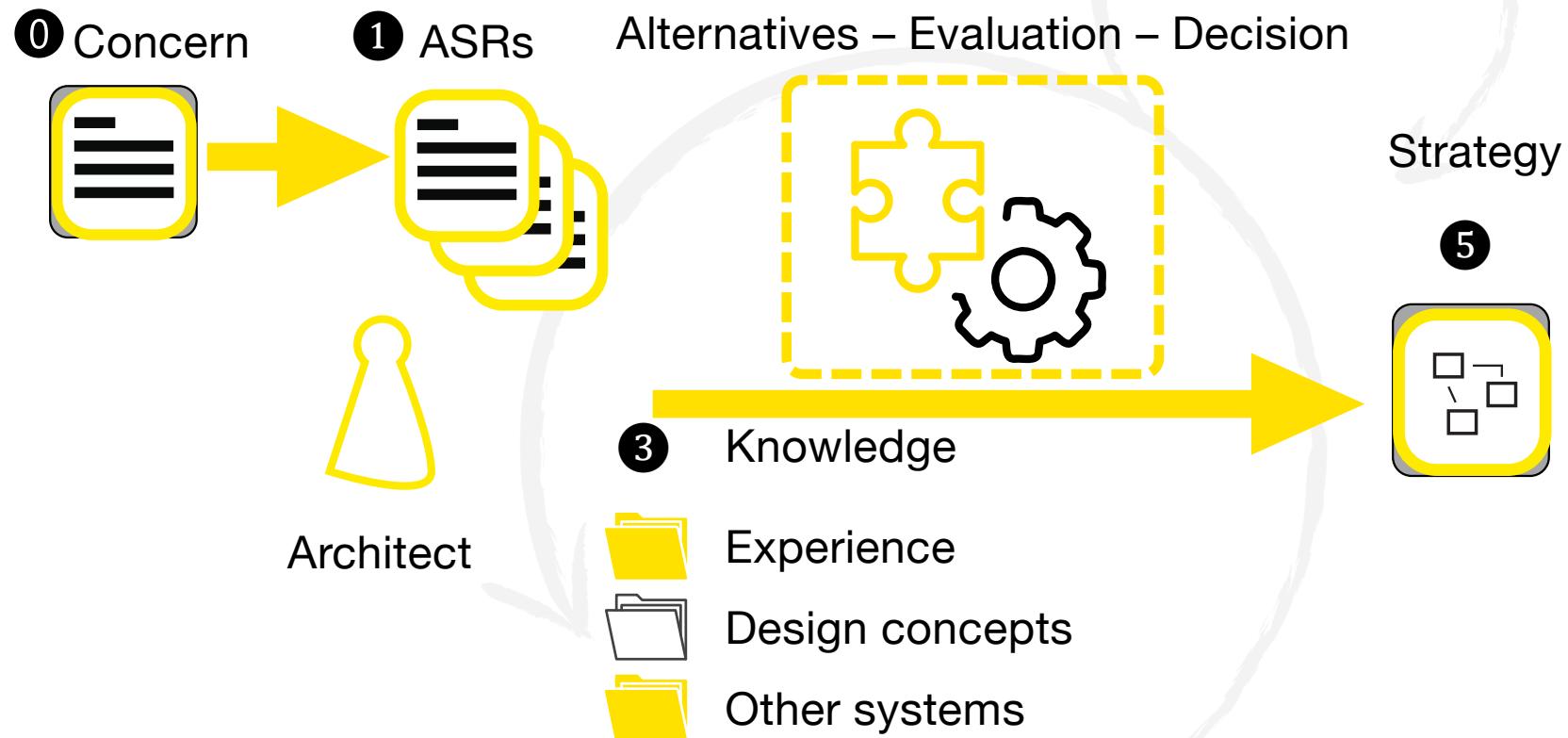
Wonderful If I get a message when an order  
is ready for pickup!

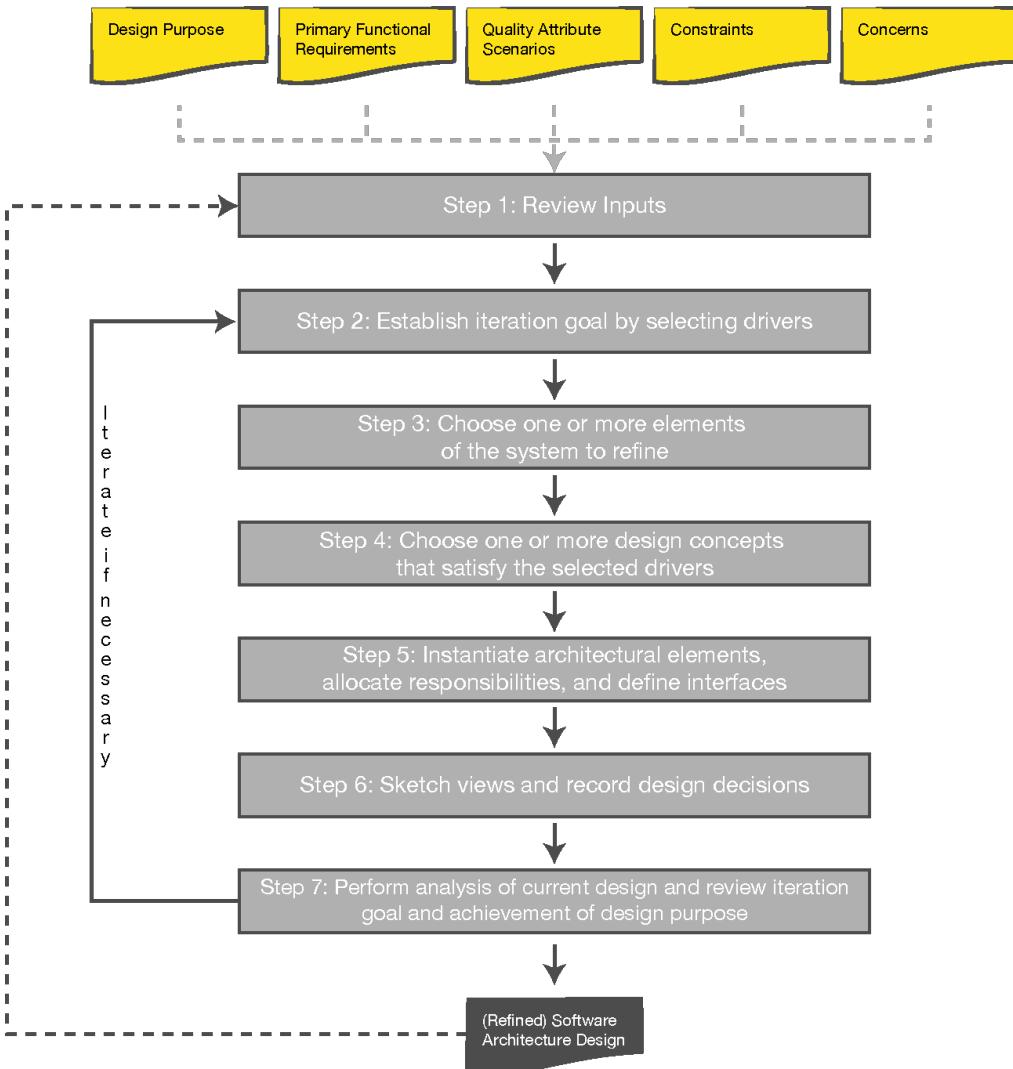
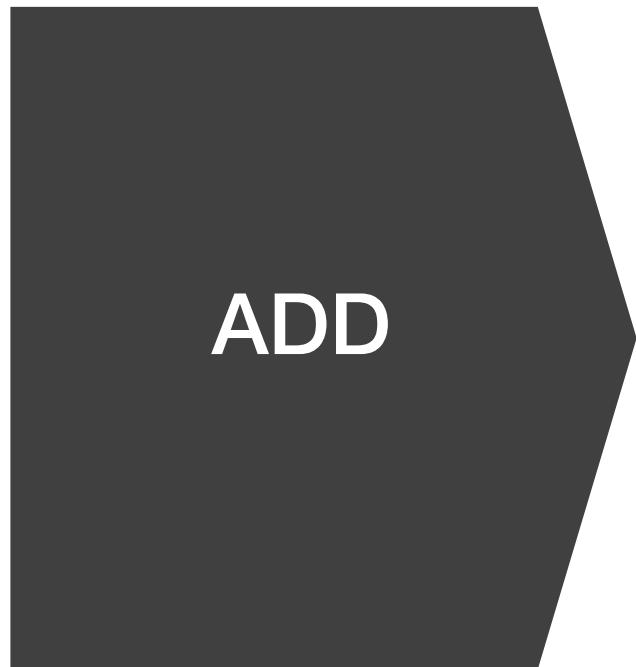
I would like to see the status of my orders all the time in the app.

Great if I could get an email when a returned order is  
accepted!

# Lisa – Owner Events by Lisa

# Agile Architecture Design





# ADD



1<sup>st</sup> level

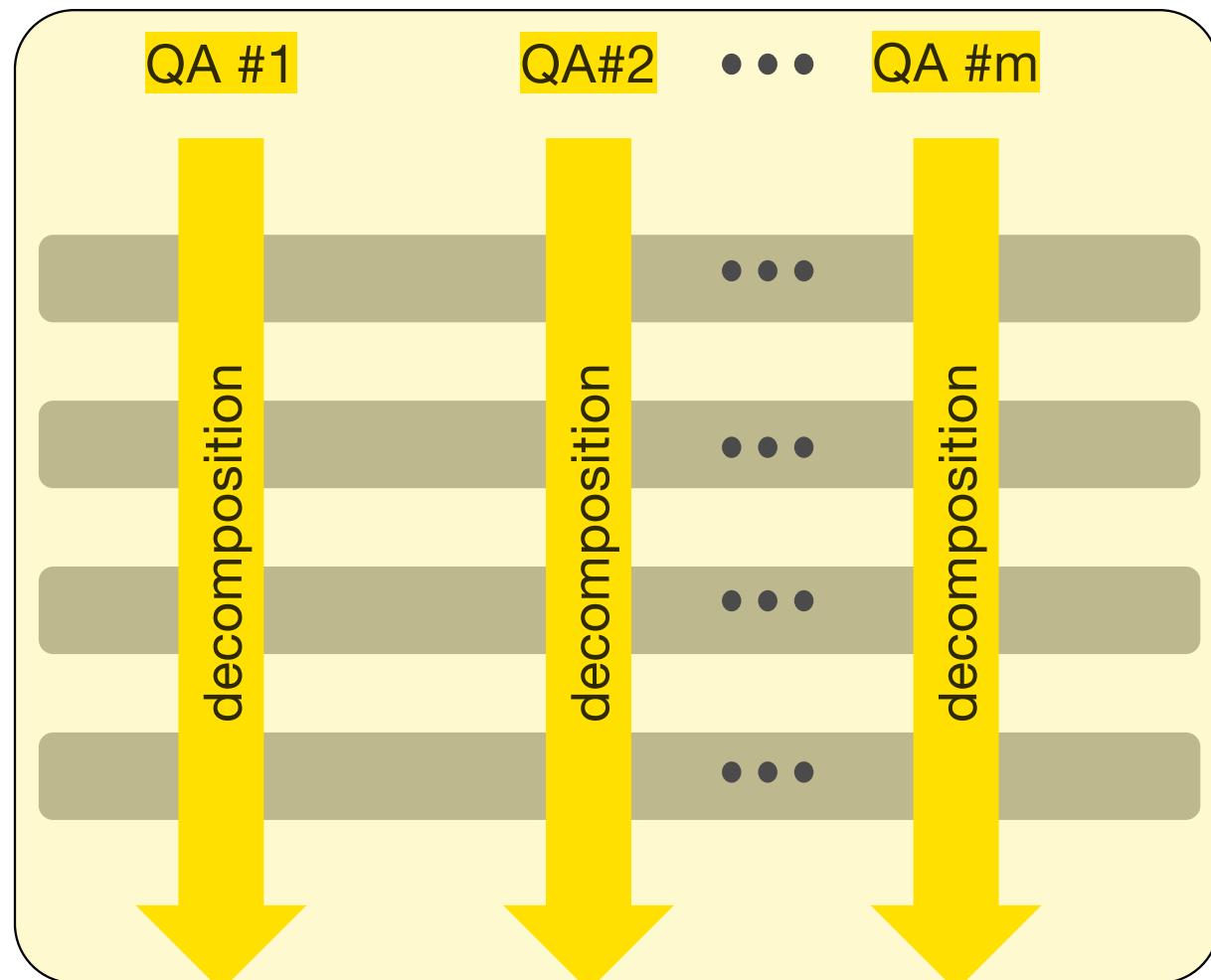
2<sup>nd</sup> level

3<sup>rd</sup> level

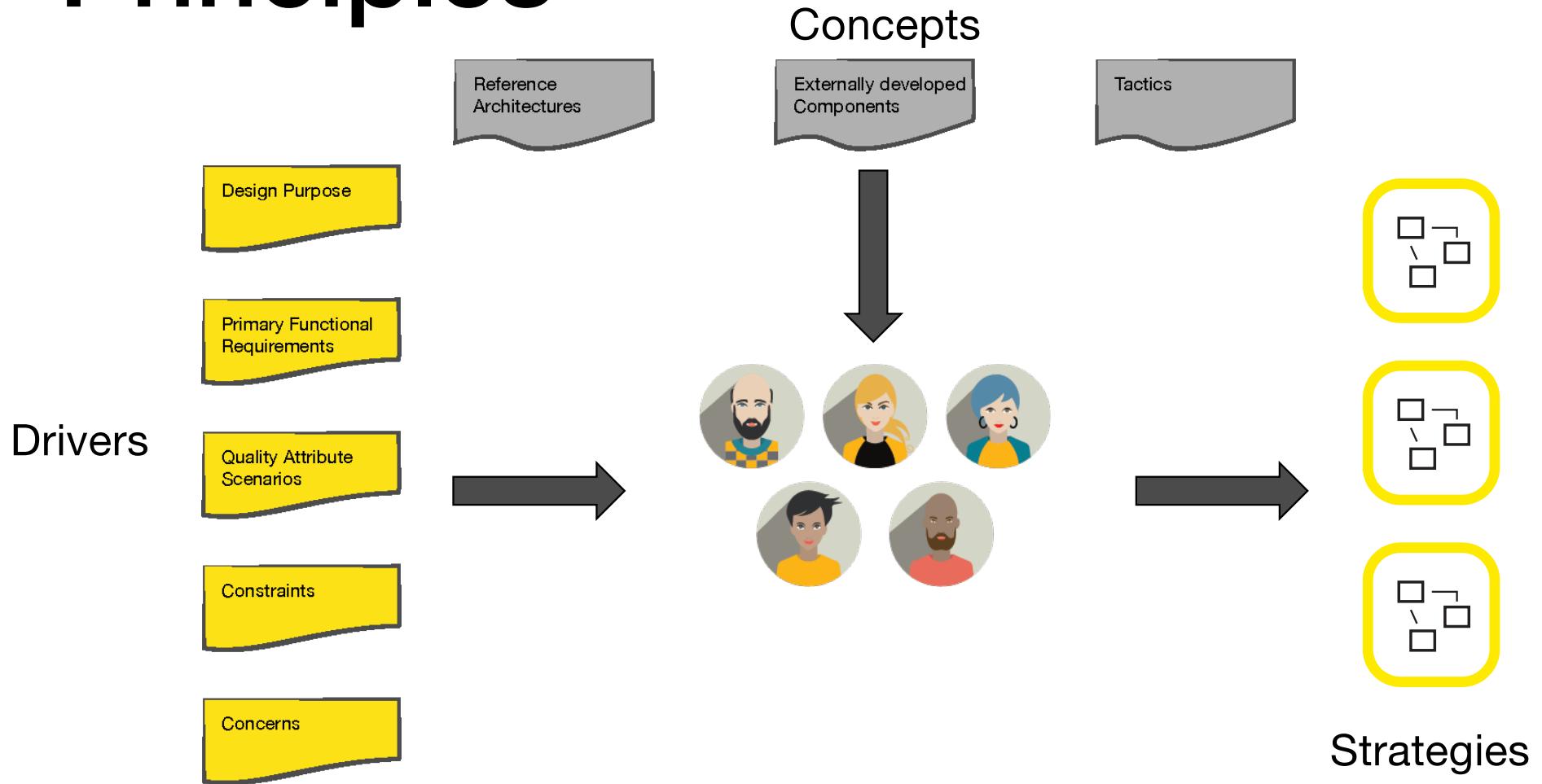
• • •

$n^{\text{th}}$  level

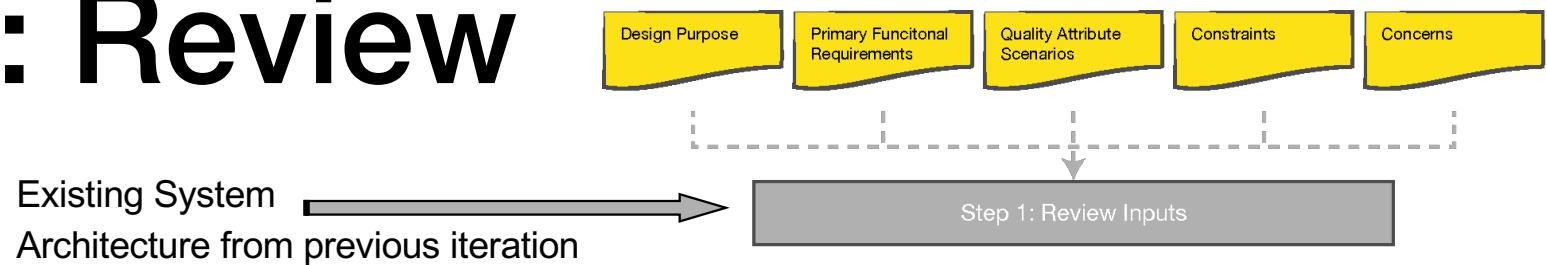
System Integrity



# Principles

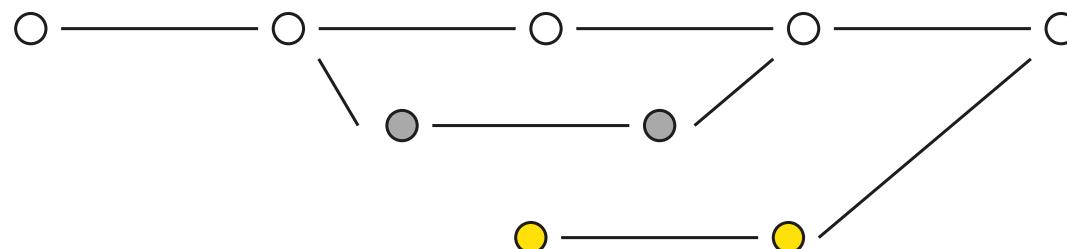


# Step 1: Review



Check that architectural drivers are available and useful

- The **purpose** of the design round
- The primary functional requirements
- The primary quality attribute (QA) scenarios
- Any constraints
- Any concerns

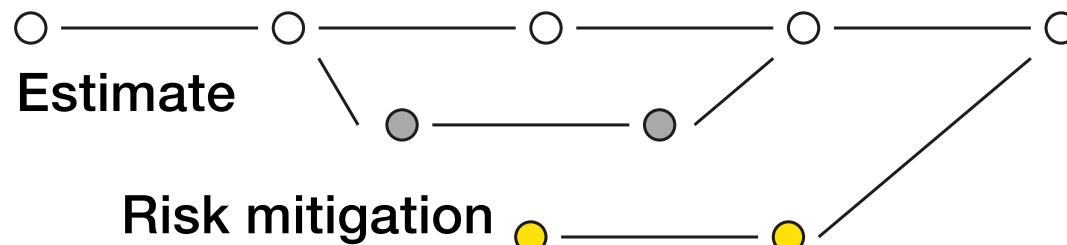


# Design Purpose

Some examples



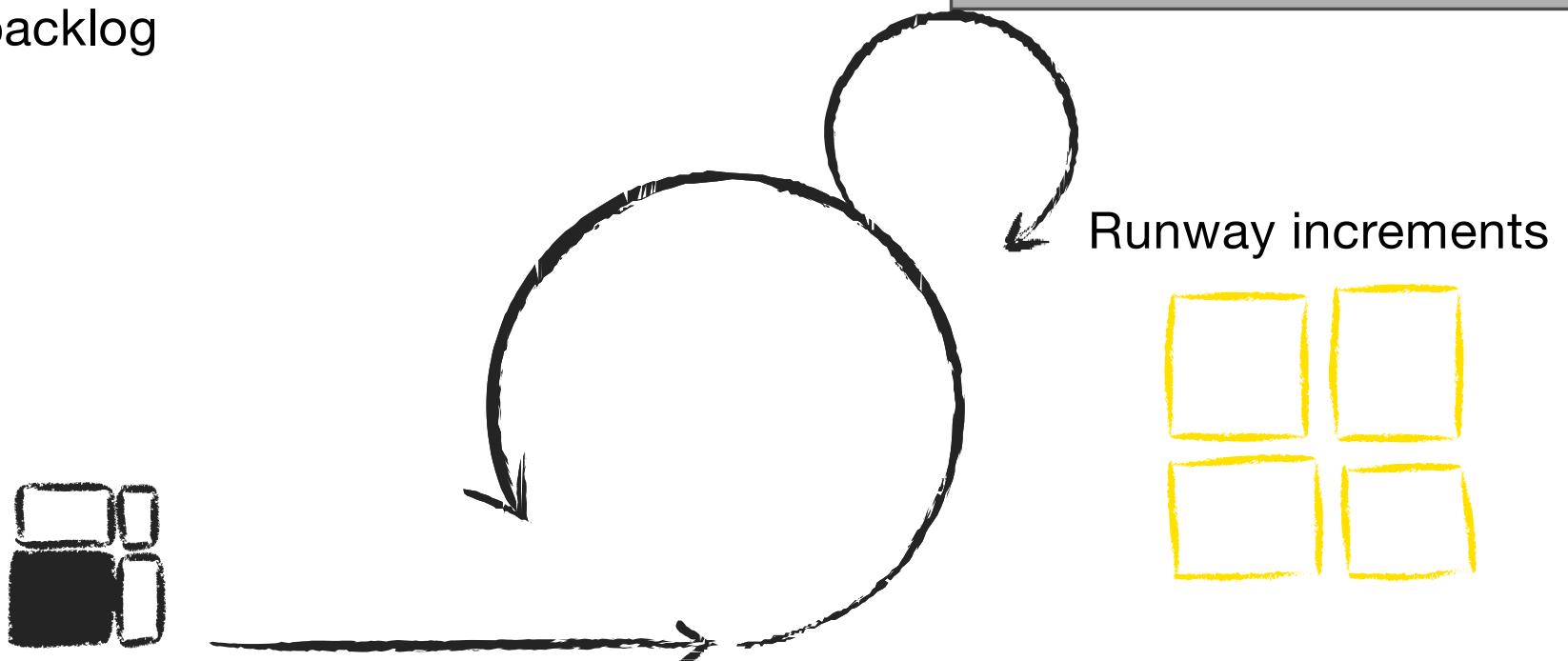
Runway increment



# Step 2: Iteration Goal

≈ Sprint backlog

Step 2: Establish iteration goal by selecting drivers



e.g., select scenarios or strategies to address

# Iteration Goals

Nice if users could interact with our stores and get support!

It would be great if I could keep track of the business in real time!



Wonderful If I get a message when an order is ready for pickup!

I would like to see the status of my orders all the time in the app.

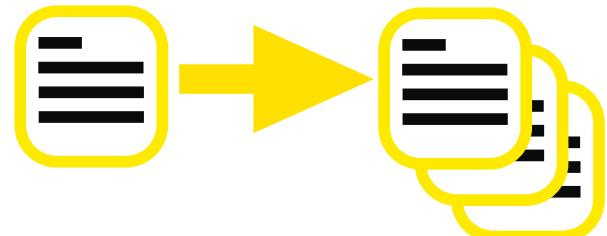
I would like to get notified when new reservations are made that should be picked immediately.



Great if I could get an email when a returned order is accepted!



Concerns



ASRs

# Step 3: Select Elements

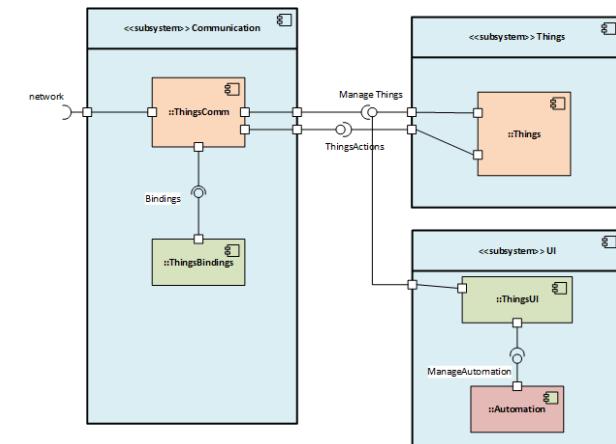
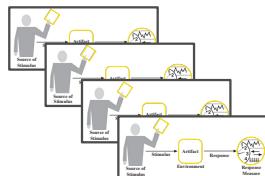


Step 3: Choose one or more elements of the system to refine

Identify which structures and elements that are affected

Two scenarios

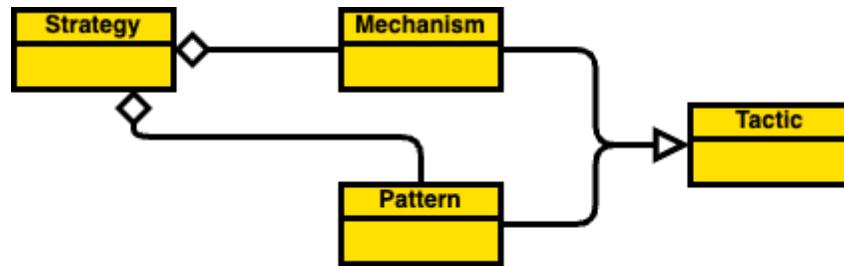
- Greenfield
- Refinement



# Step 4: Choose Concepts

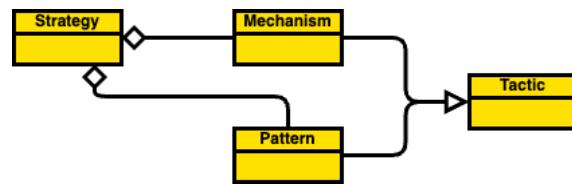


Step 4: Choose one or more design concepts  
that satisfy the selected drivers



Options and alternatives ➔ Reasoning

# Step 4: Design Concepts



Step 4: Choose one or more design concepts  
that satisfy the selected drivers

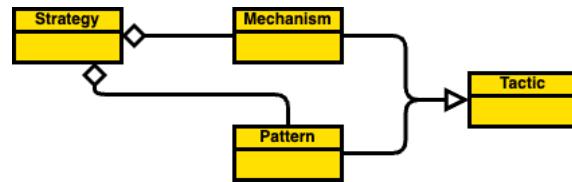
Options and alternatives → Reasoning

Examples of design concepts

- tactics,
- patterns,
- reference architectures,
- externally developed components and services

For each type, many options may exist!

# Step 4: Choose Concepts



Step 4: Choose one or more design concepts  
that satisfy the selected drivers

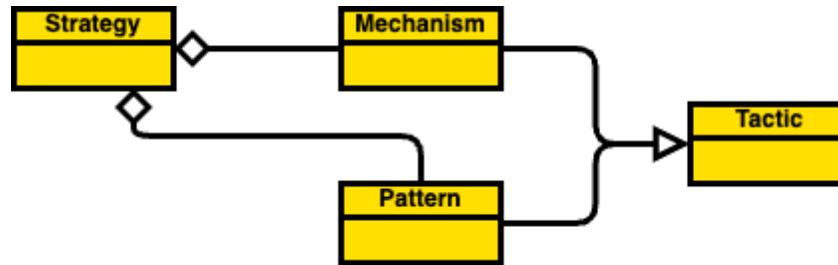
Options and alternatives → Reasoning

# Steg 4: Choose Concepts

Välj designkoncept

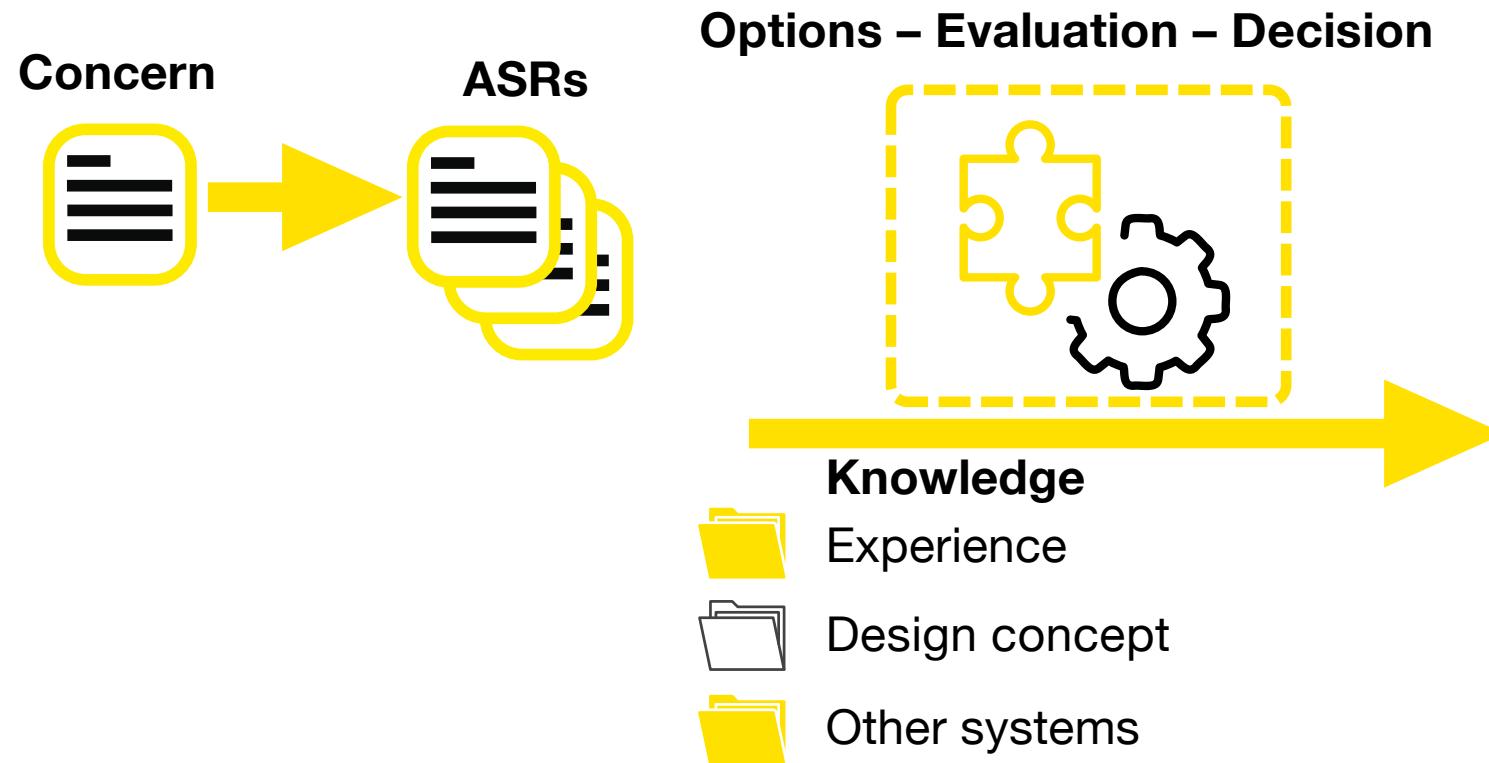


Step 4: Choose one or more design concepts  
that satisfy the selected drivers



Val och alternativ ➔ Resonemang

# Architectural Design



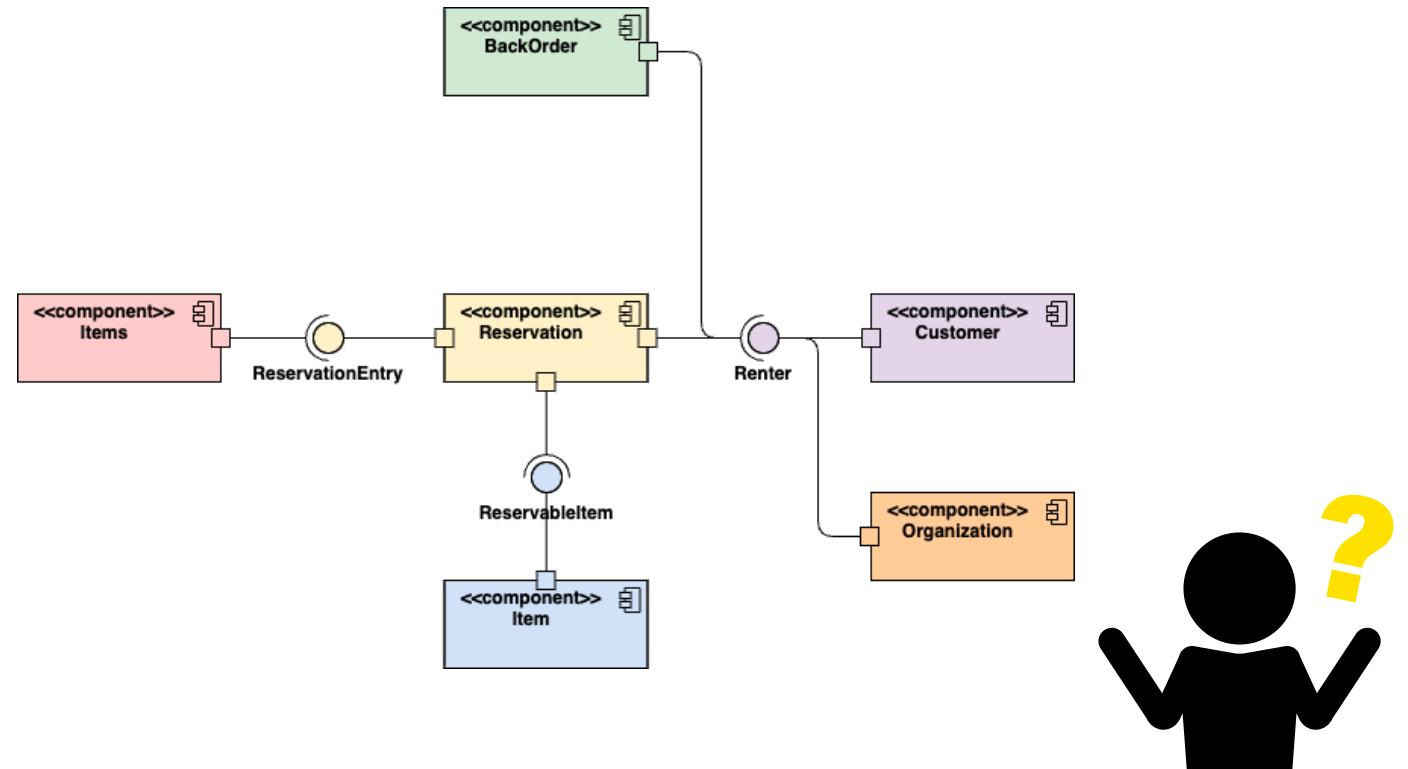
# Knowledge

# Concerns

Communication

Configuration

Data collection

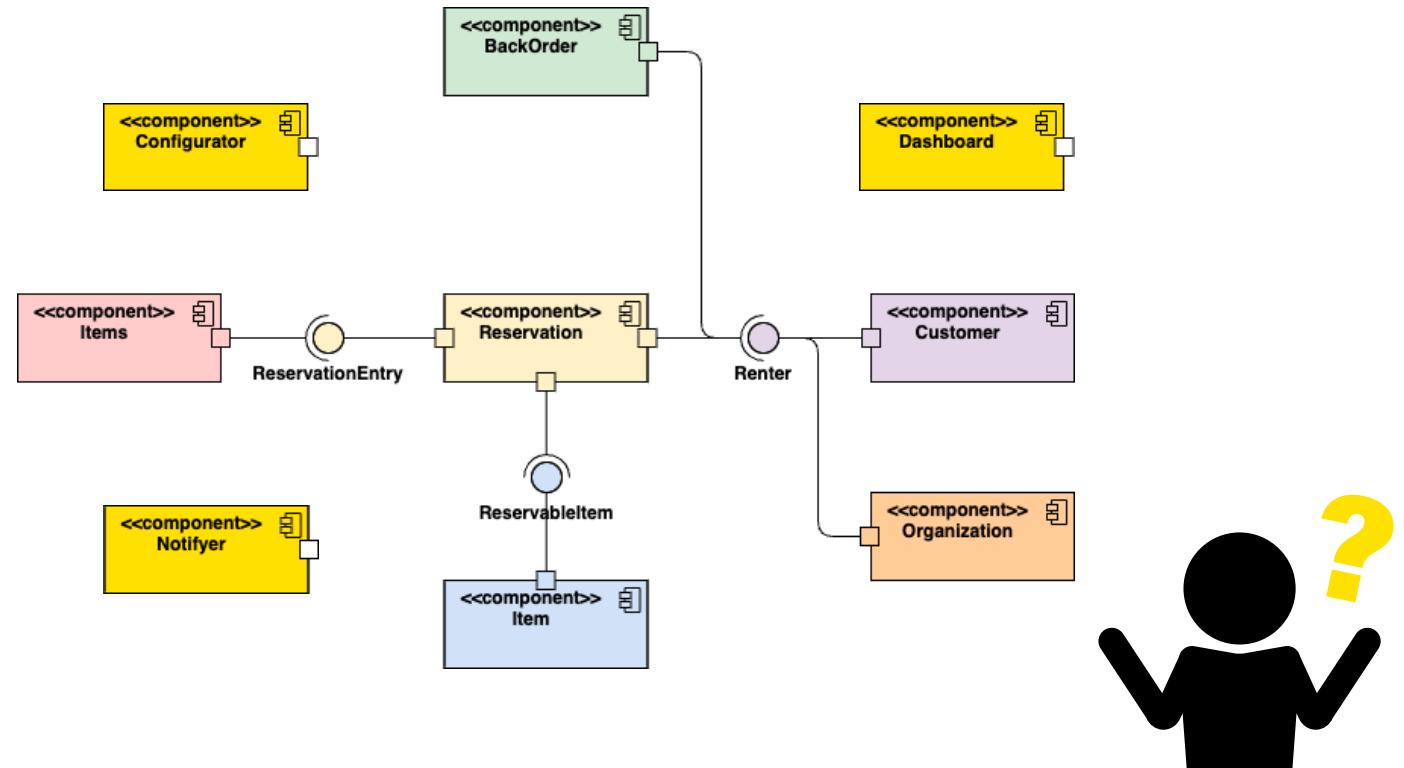


# Concerns

Communication

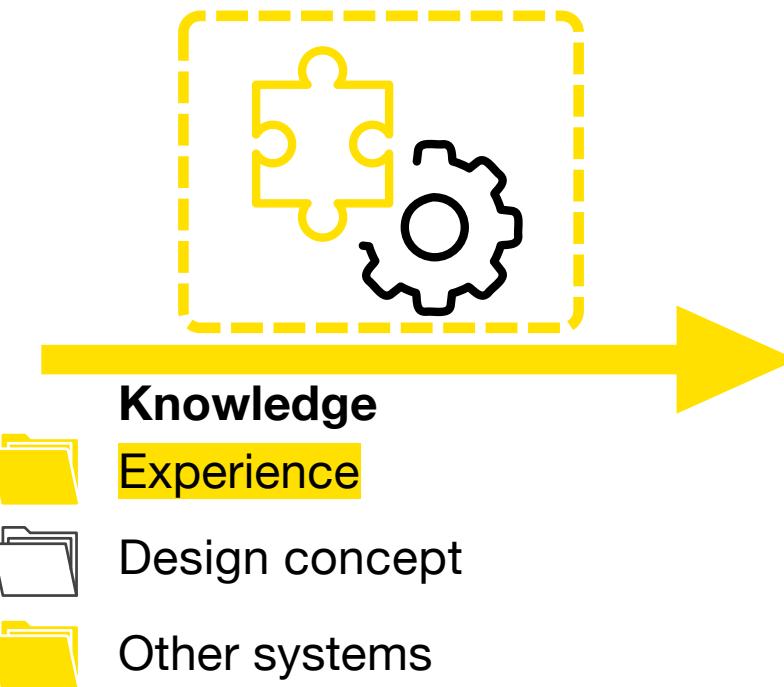
Configuration

Data collection



# Architectural Design

Options – Evaluation – Decision



# Experience

# Experience

Tactics – Reference Architectures

# Experience

Tactics – Reference Architectures

# Other systems

# Developing concepts for



Communication

Configuration

Data collection

## Knowledge



Experience



Design concepts



Other systems

# Options and Alternatives

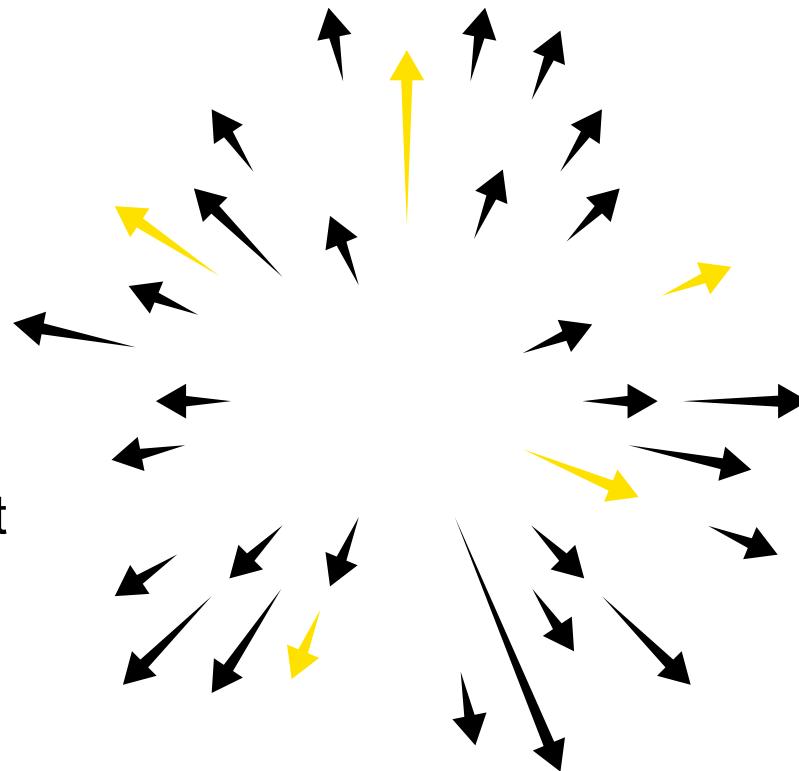


New concept

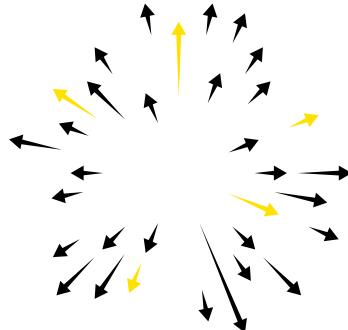
Reuse your own concept

External – customized – concept

External concept



# Options and Alternatives



New concept

Reuse your own concept

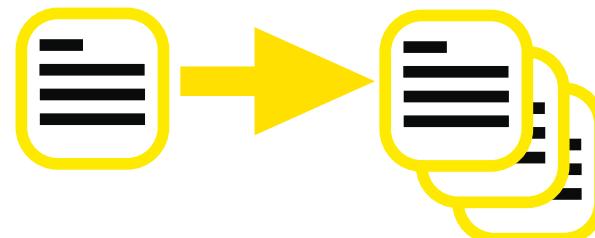
External – customized – concept

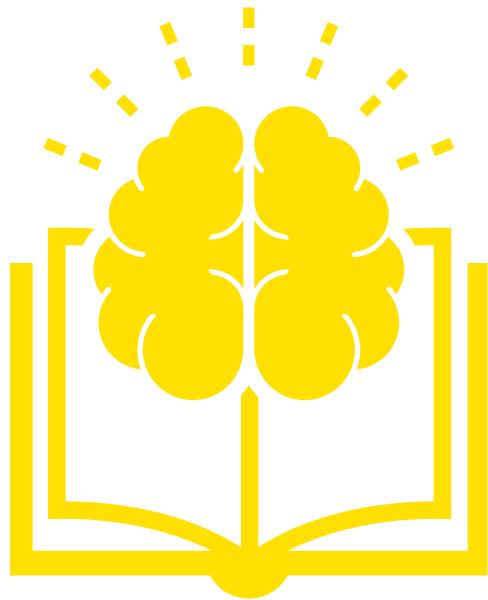
External concept

The needs are specified by

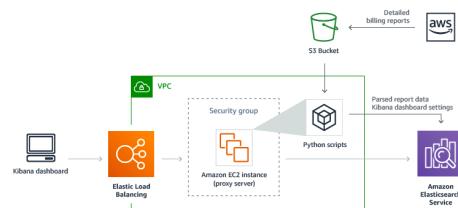
Concern

ASRs

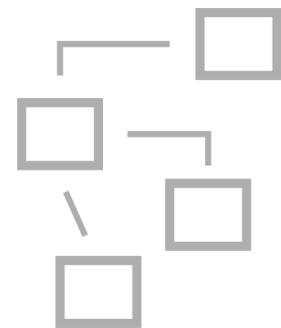




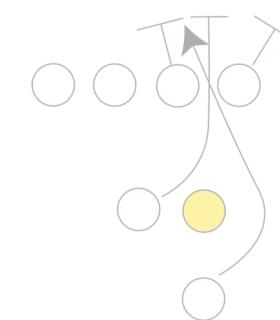
# Design Concepts



Reference architectures



Patterns



Tactics

# Developing concepts for



Communication

Configuration

Data collection

## Knowledge



Experience



Design Concepts



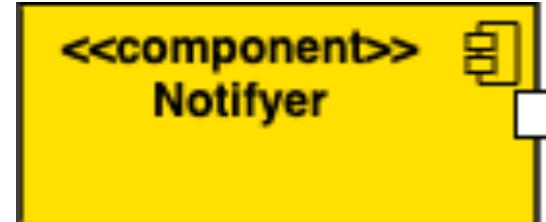
Other systems

# Think about a concept for

sms

in-app

mail



???



# Options and Alternatives

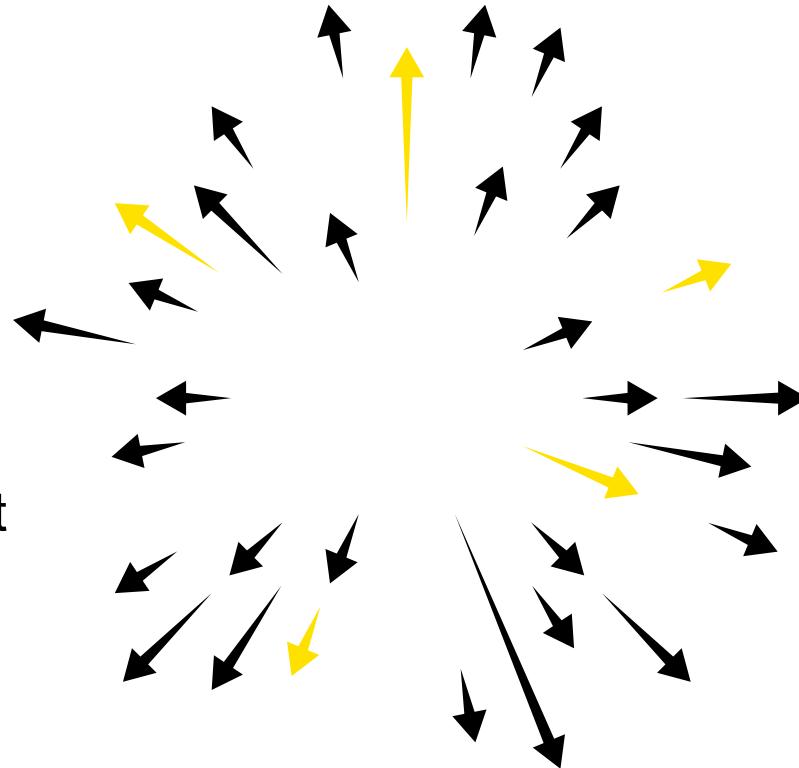


New concept

Reuse your own concept

External – customized – concept

External concept



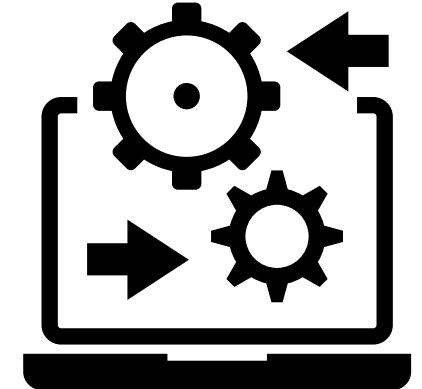
Function – Quality – Cost – Time – Life cycle

# Examples of concepts

Enterprise Architecture Integration framework/platform

Enterprise application integration is an integration framework composed of a collection of technologies and services which form a middleware or "middleware framework" to enable integration of systems and applications across an enterprise.

*Linthicum, David S. (2000). Enterprise Application Integration*



External – customized – concept

Framework

External concept

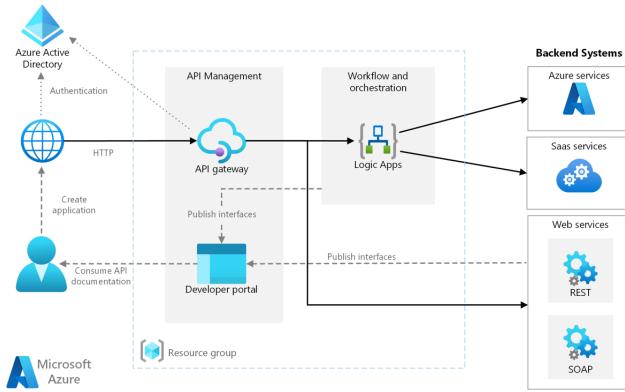
Platform

Service

---

## Function – Quality – Cost – Time – Life cycle

# EAI in the cloud



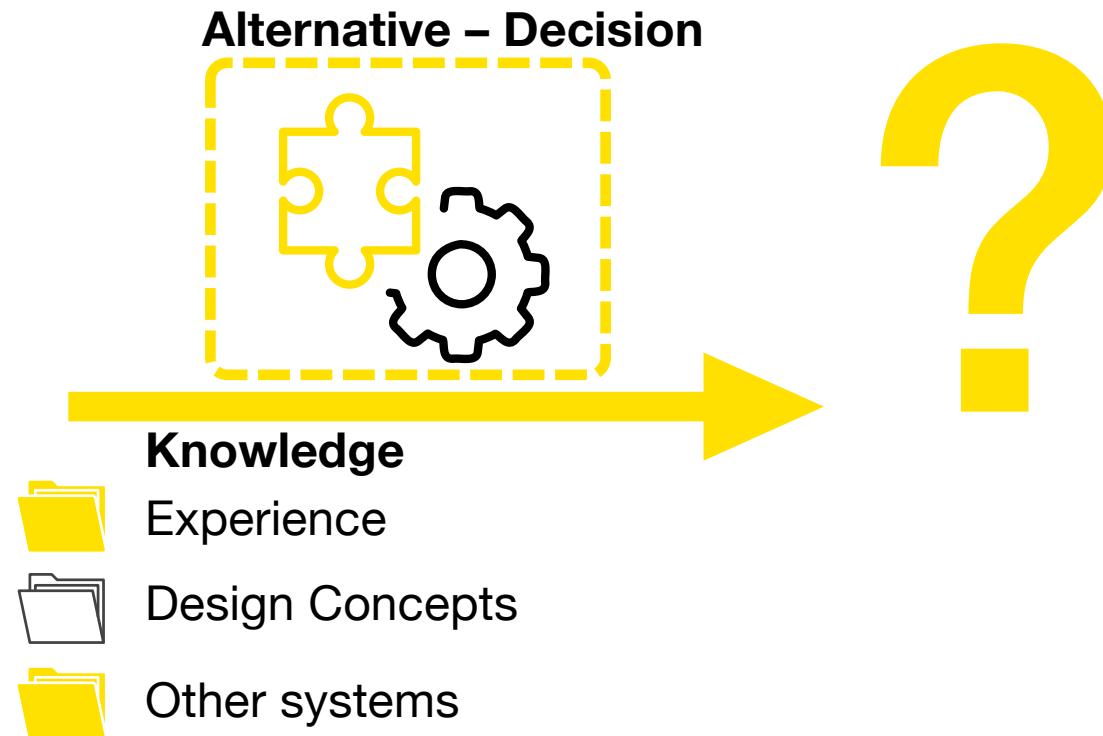
## AWS Application Integration services

Category	What is it	AWS service
API Management	Create, publish, maintain, monitor, and secure APIs at any scale for serverless workloads and web applications	<a href="#">Amazon API Gateway</a>
	Create a flexible API to securely access, manipulate, and combine data from one or more data sources	<a href="#">AWS App Sync</a>
Event Bus	Build an event-driven architecture that connects application data from your own apps, SaaS, and AWS services	<a href="#">Amazon EventBridge</a>
Messaging	Reliable high throughput pub/sub, SMS, email, and mobile push notifications	<a href="#">Amazon Simple Notification Service (SNS)</a>
	Message queue that sends, stores, and receives messages between application components at any volume	<a href="#">Amazon Simple Queue Service (SQS)</a>
	Message broker for Apache ActiveMQ and RabbitMQ that makes migration easy and enables hybrid architectures	<a href="#">Amazon MQ</a>
No-code API Integration	Automate the flow of data between SaaS applications and AWS services at nearly any scale, without code.	<a href="#">Amazon AppFlow</a>
Workflows	Coordinate multiple AWS services into serverless workflows so you can build and update apps quickly	<a href="#">AWS Step Functions</a>
	Run Apache Airflow at scale without provisioning or managing infrastructure	<a href="#">Amazon Managed Workflows for Apache Airflow (MWAA)</a>



**Function – Quality – Cost – Time – Life cycle**

# Architectural Design



# Step 5: Elements



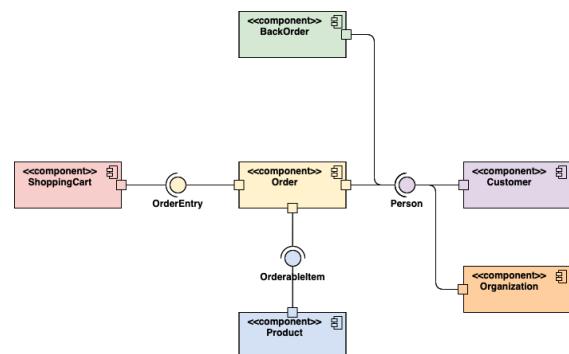
Step 5: Instantiate architectural elements, allocate responsibilities, and define interfaces

Create the elements for the (refined) strategy

Allocate responsibilities

Define interfaces

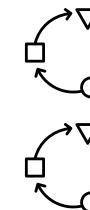
Integrate!



## 2<sup>nd</sup> level and 3<sup>rd</sup> and ... n<sup>th</sup>?

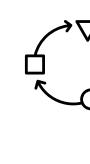
From Black-box to White-box

FIRST LEVEL DECOMPOSITION



- Add and refine components
- Add and refine responsibilities
- Add and refine interfaces
- Add and refine connections

SECOND LEVEL DECOMPOSITION



Linnæus University

# And now we're almost done

## Step 6: Outline Views



**Purpose:**  
Show that the system addresses their concerns

Descriptive  
UX mockups  
Scenarios

↓  
Step 6: Sketch views and record design decisions



**Purpose:** Show how the system addresses the concerns  
**Models**

Structure   Behavior   Argumentation

---

Linnæus University

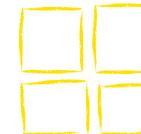
## Step 7: Evaluation

↓  
Step 7: Perform analysis of current design and review iteration goal and achievement of design purpose

Lightweight review

≈ sprint review

Runway Increments



Step 1: Assess iteration

- Design Purpose
- Primary Functional Requirements
- Quality Attribute Scenarios
- Constraints
- Concerns

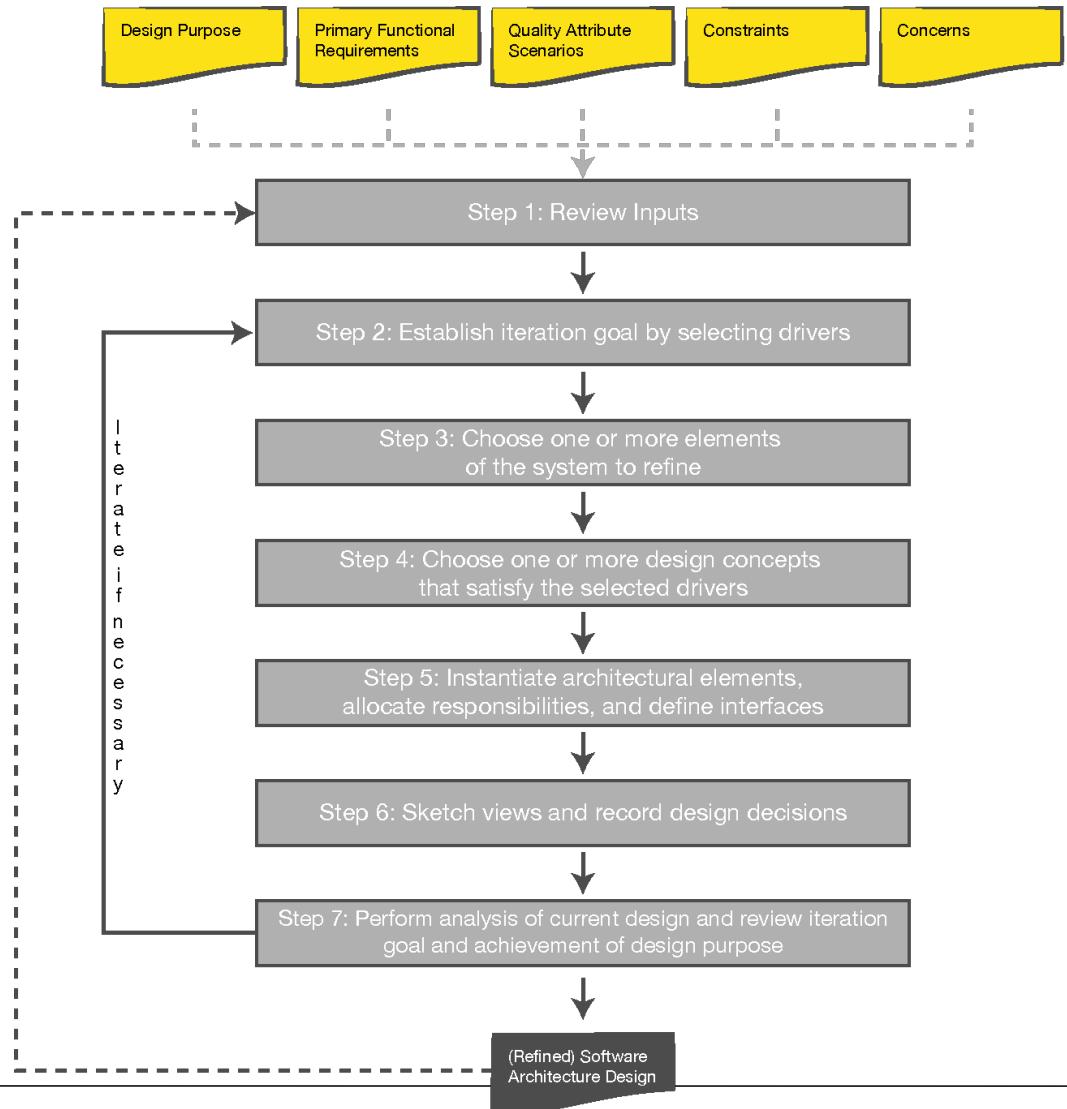
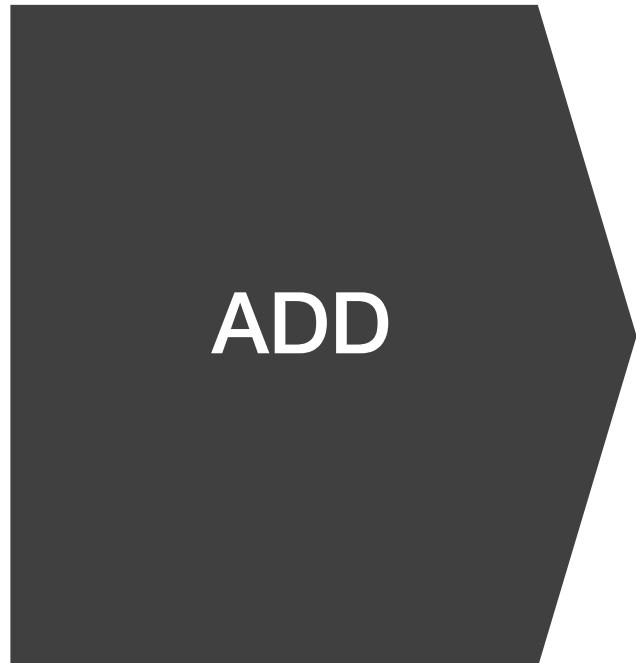
Step 2: Assess Architecture

(Refined) Software  
Architecture Design

---

Linnæus University

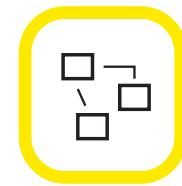
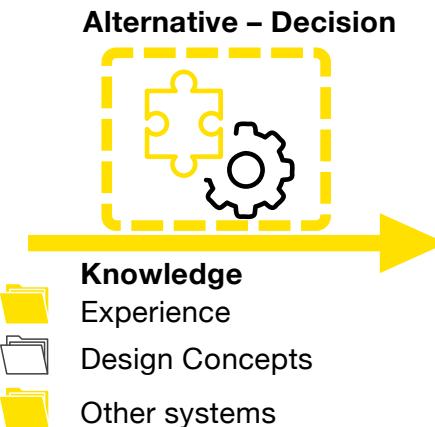
# Iiterate!



# Take-aways for today!



Think broadly!  
Own or not?  
Pros and Cons



Cost/Time



Function/Quality

In-house/External



## 2DV604 Software Architecture