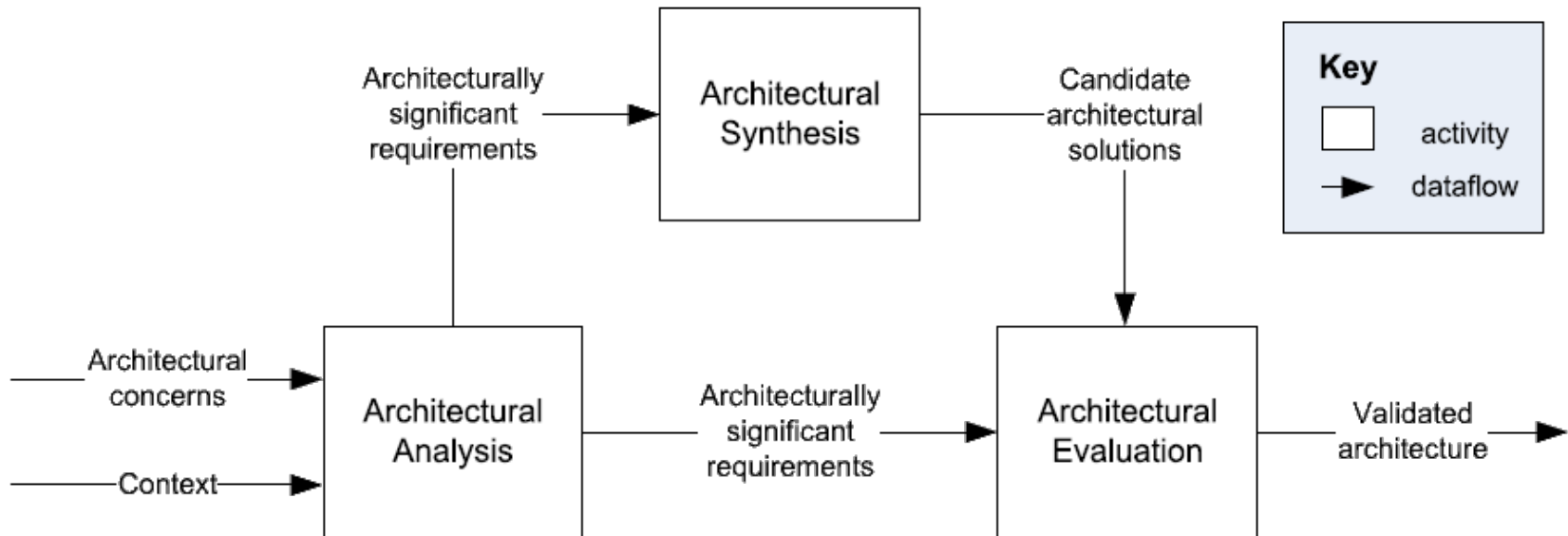


# Software Architecture Process

---

# A General Architecture Process Model



# Architectural Analysis

---

Define the problems the architecture must solve.

## Inputs:

**Architectural Concerns:** Interests which pertain to the system's development, its operation or any other aspects that are critical to one or more shareholders

**Context:** The setting and circumstances of developmental, operational, political and other influences upon the system

## Output:

**Architecturally Significant Requirements:** A requirements upon a software system which influences its architecture

# Architectural Synthesis

---

Proposes architectural solutions to a set of Architecturally Significant Requirements

Input:

**Architecturally Significant Requirements**

Output:

**Candidate Architectural Solutions:** Design decisions about the structure of the software

# Architectural Evaluation

---

Ensure that the architectural designs decisions made are the right ones

Inputs:

**Architecturally Significant Requirements**

**Candidate Architectural Solutions**

Outputs:

**Validated Architecture:** A subset of candidate architectures that are consistent with the Architecturally Significant Requirements

# Repeatable Processes for Software Architecture

---

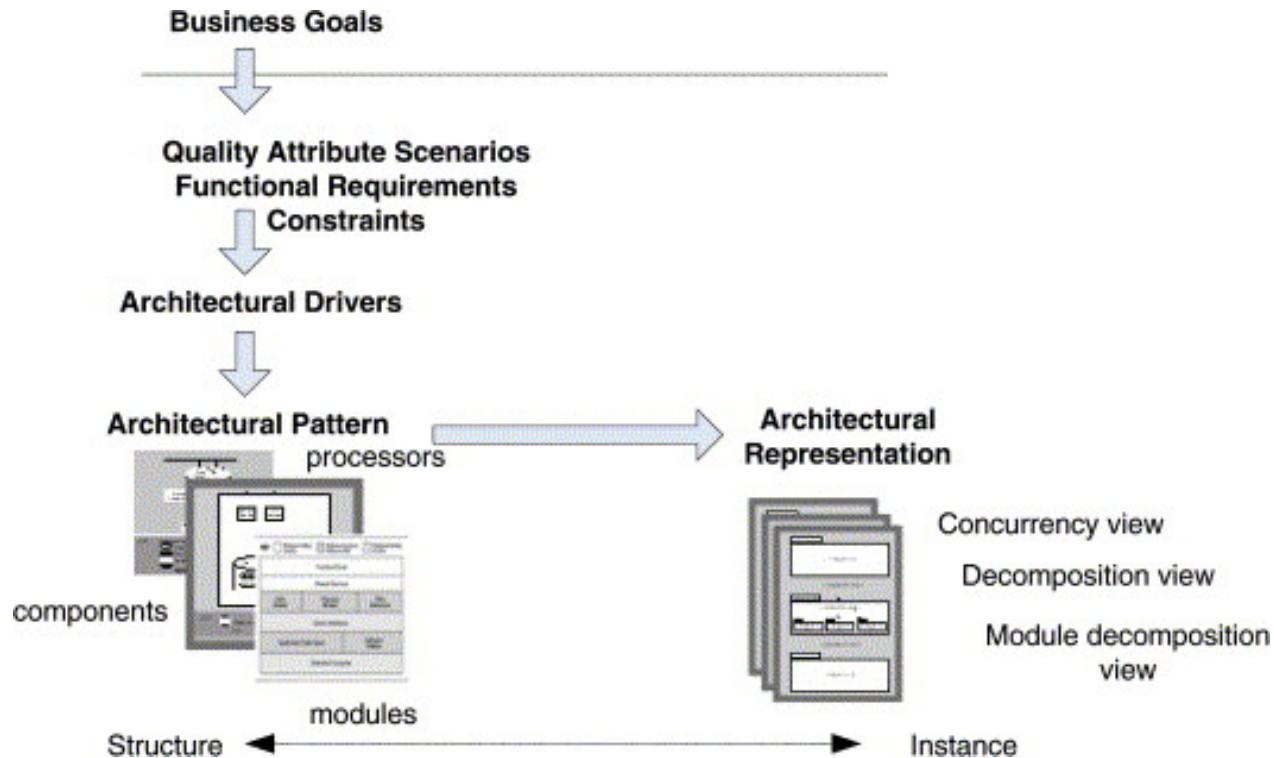
Analysis, Synthesis, and Evaluation are the core concepts of many architectural methodologies.

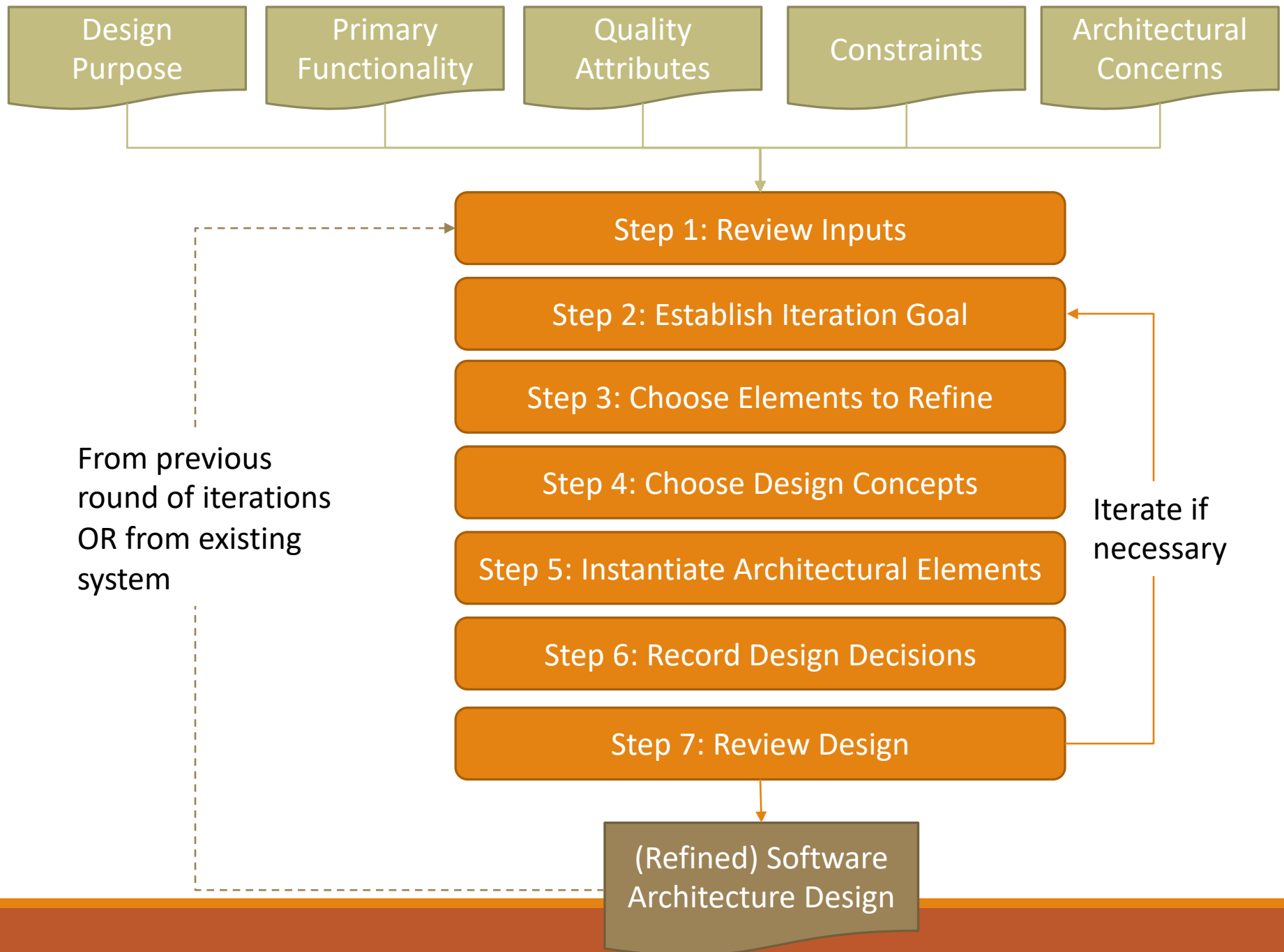
Examples:

- Attribute-Driven Design
- RUP 4+1 Views
- Business Architecture Process and Organization (BAPO/CAFCR)

No one process is best – each has its strengths and weaknesses.

# Attribute Driven Design





# ADD Step 1

**Review Inputs: Ensure the architecture drivers are available and correct**



- **Design Purpose** – The goal of this round of Architectural Design
  - Evaluate opportunities
  - Design the foundation of the system
  - Augment the current architecture
- **Primary Functionality** – Functionality that is critical to achieve the business goals that motivate the development of the system
- **Quality Attributes** – The subset of QAs this round is focused on
  - Derived from ASRs, focus on measurable or testable qualities
- **Constraints** – Non-negotiable concerns
  - Existing systems
- **Architectural Concerns** – Things that need to be considered as part of the architectural design, often not captured by traditional requirements
  - Overall System Structure
  - Allocation of modules to teams
  - Issues that come from design reviews

Design  
Purpose

Primary  
Functionality

Quality  
Attributes

Constraints

Architectural  
Concerns

# ADD Step 2

## Establish Iteration Goal by Selecting Drivers

Waterfall development model consists of one round, therefore the goal is to satisfy all architectural activities

Iterative development (Agile) model consist of multiple rounds; each round should focus on a specific goal

- Round 1 of a Greenfield System in a mature field may be to select the overall Reference Architecture
- Round 1 of a Greenfield System in a new, evolving field may be to develop a prototype architectural model based on the key Drivers
- Round 1 of a Brownfield System may focus on how the existing architecture extends to support the new functionality
- Later rounds may focus on a specific area – Security, Performance, etc. Where the specific area is based on the Drivers



# ADD Step 3

## Choose One or More Elements of the System to Refine

Satisfying drivers requires making architectural design decisions

Based on the Drivers selected during Step 2, identify the elements that require additional refinement. Potential types of refinement:

- **Decomposition**
- **Composition**
- **Improve / extend existing elements**

For a Greenfield system the first iteration will start with the system context diagram and refining / defining it

For Brownfield systems the first iteration may focus on understanding the existing system and the elements you will need to be working with



# ADD Step 4

---

## Choose One or More Design Concepts that Satisfy the Selected Drivers

Leverage existing solutions, combine and adapt them as necessary



- **Reference Architectures** – A model architecture to solve a common problem, for instance a Client-Server model for providing a Web Service or AUTOSAR a reference model for software in automobiles
- **Architectural Design Patterns** – Conceptual solutions to reoccurring design problems, the Layer pattern is a commonly used example
- **Deployment Patterns** – How to physically structure a system for deployment, Load Balanced Cluster
- **Tactics** – Design decisions that influence the control of a quality attribute

# ADD Step 5

## Instantiate Architectural Elements, Allocate Responsibilities and Define Interfaces

Apply the design concept to the element(s) under consideration

For instance if we're focused on the enabling user interactions with the system and we've chosen the architectural pattern Model-View-Controller

- Where does each component reside?
- What is the model we need to convey to address Usability?
- What parts of the system does the controller need to interact with?

We may not know the answers to all these questions and future iterations may be required to further refine them



# ADD Step 6

---

## Sketch Views and Record Design Decisions

Document it

Design decisions have been made, the results are likely diagrams and supporting rationale

Now is the time to capture that information, before its lost



# ADD Step 7

---

## Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Goal

Review the architecture against the drivers and overall system goals

If the drivers are met and the design purpose is satisfied, declare victory and move forward

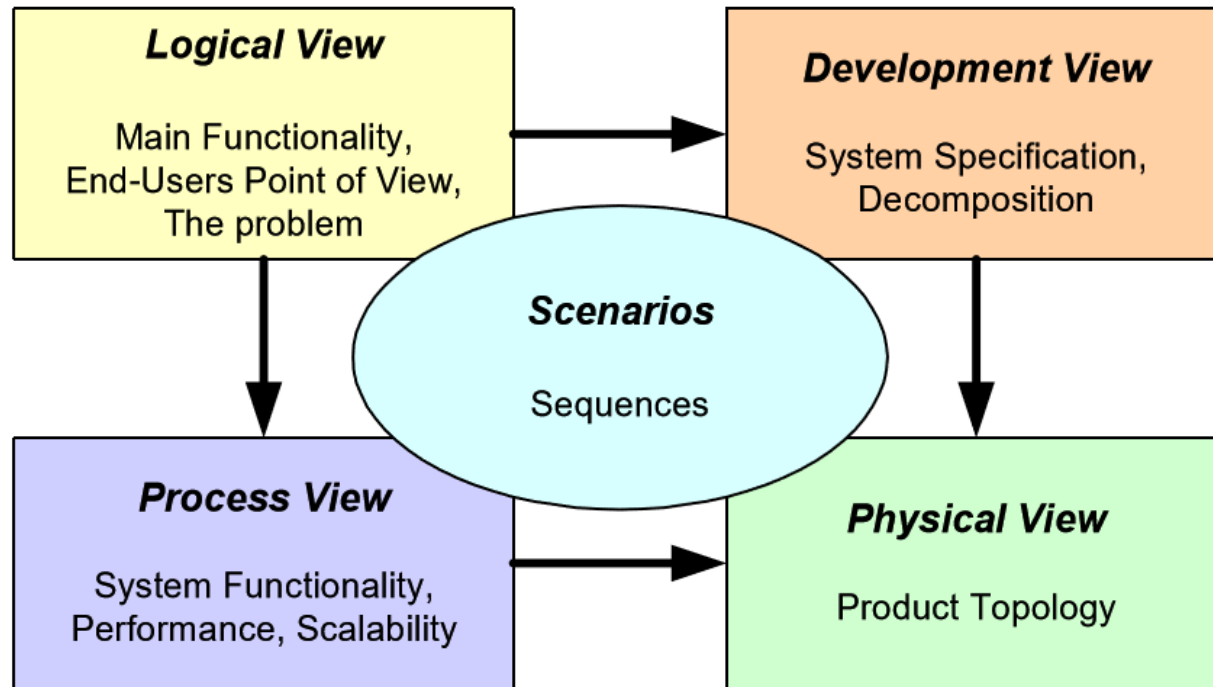
If there is still work to complete to meet the design purpose, continue on to the next set of drivers

If the drivers are not met, determine what needs to change and revise the architecture



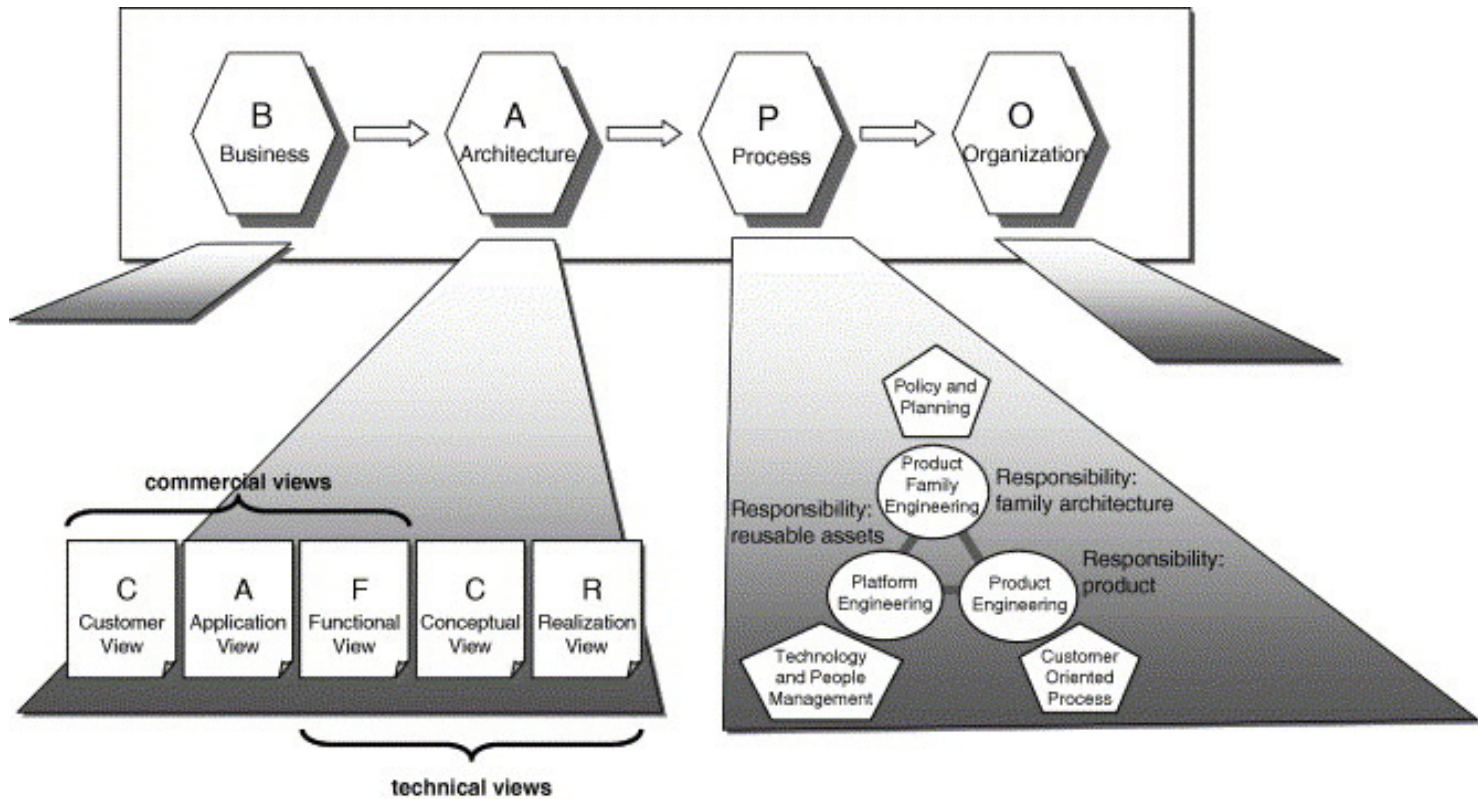
# RUP 4+1 View

---



*Figure 2. Software architecture description with 4+1 views.*

# BAPO/ CAFCR



# PRODUCT DELIVERY PROCESS

## PROCESS PHASES



## PHASE TRANSFER REVIEWS

### PRE-CONCEPT INITIATION

- Business Need Identified
- Management Process Defined

### CONCEPT INITIATION

- Technology Readiness Demonstrated
- QCD/Customer Satisfaction Targets Established
- Product Intent Design Layout Completed

### PRODUCT INITIATION

### DEVELOPMENT INITIATION

### PRODUCTION/ LAUNCH READINESS

- Design Stability Confirmed
- Product Manufacturability Verified via Pilot Production Build and Test
- Worldwide Quality Launch Plan/Operating Unit Engagement Plans Completed
- Launch Readiness Assessed

### LAUNCH GO/WAIT

### PRODUCT VALIDATION

- Product QCD/Customer Satisfaction Targets Assessed
- Market/Field/Service Performance Audit Completed
- Product Improvements and Enhancements Planned
- Fleet Upgrade (End of Life) Strategy Defined

### FLEET UPGRADE

## REVIEW OBJECTIVES

### Concept Initiation

- Business Need Confirmed
- Architecture/Technology Set Selected
- Business Case Scoped

### Development Initiation

- Production Intent Design Demonstrated
- QCD/Customer Satisfaction Targets Committed
- Worldwide Quality Launch Strategy Defined

### Launch Go/Wait

- Final Customer Testing Completed
- Product/Field Readiness Confirmed
- Product Announcement Completed and Worldwide Launch Initiated

### Fleet Upgrade (End of Life)

- Product Withdrawal/Service Discontinuance Plans Implemented
- Final Product Disposition Completed

# Scenarios

---

Web application for banking

Describe a:

- Performance Attribute
- Security Attribute
- Integrity Attribute
- Availability Attribute

Control system for driverless car

Describe a:

- Performance Attribute
- Security Attribute
- Safety Attribute
- Reliability Attribute

What would be a good example of a Portability Attribute?

# Examples

---

If you are building a desktop application

- Connects to online brokerage system

List some assumptions

List some dependencies