Architecture Definition

CSCE 742 - Lecture 4 - 09/04/2018

What is the role of the software architect?

The Role of the Architect

- Some write a lot of code, some hand jobs to the build and testing teams.
- Some specialize in one area (networking, middleware, databases), others have no development background at all.
- What does an architect *do*?

Today’s Class

- Discuss the architecture definition process.
  - How we discover the system architecture.
- Explain the relationship between “architecture”, “design”, and “requirements”.
  - Your “what” is my “how”.
- Define the role of the software architect.

Introducing Architecture Definition

Architecture Definition

“Software architecture is the **fundamental organization** of a system, embodied in its **components**, their **relationships** to one another and the environment, and the **principles** governing its design and evolution.”

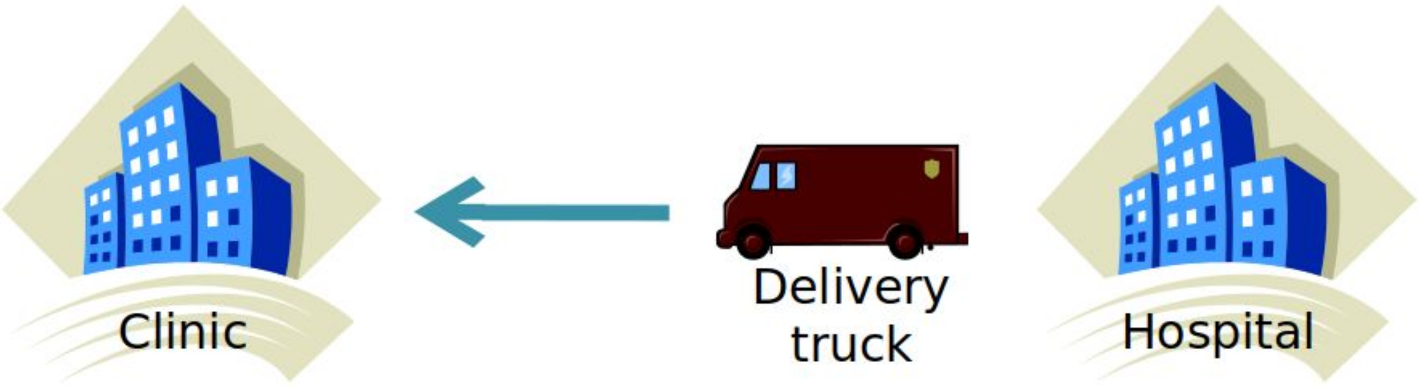
Architecture definition is the process of **discovery** of these components, relationships, and principles.

Architecture Definition

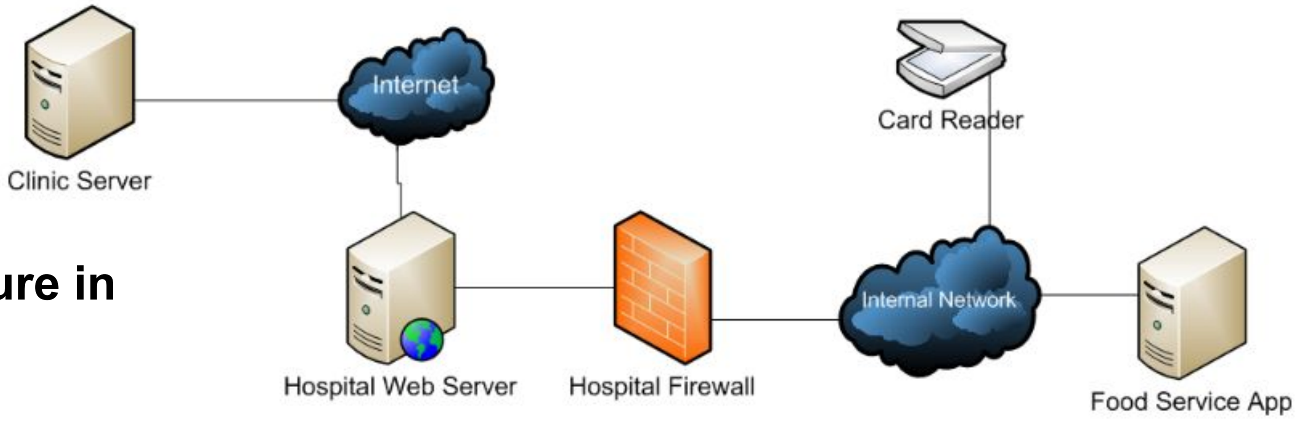
Architecture definition is a process by which stakeholder needs and concerns are captured, an architecture to meet these needs is designed, and the architecture is clearly and unambiguously described via an architectural description.

Hospital Food

Physical World



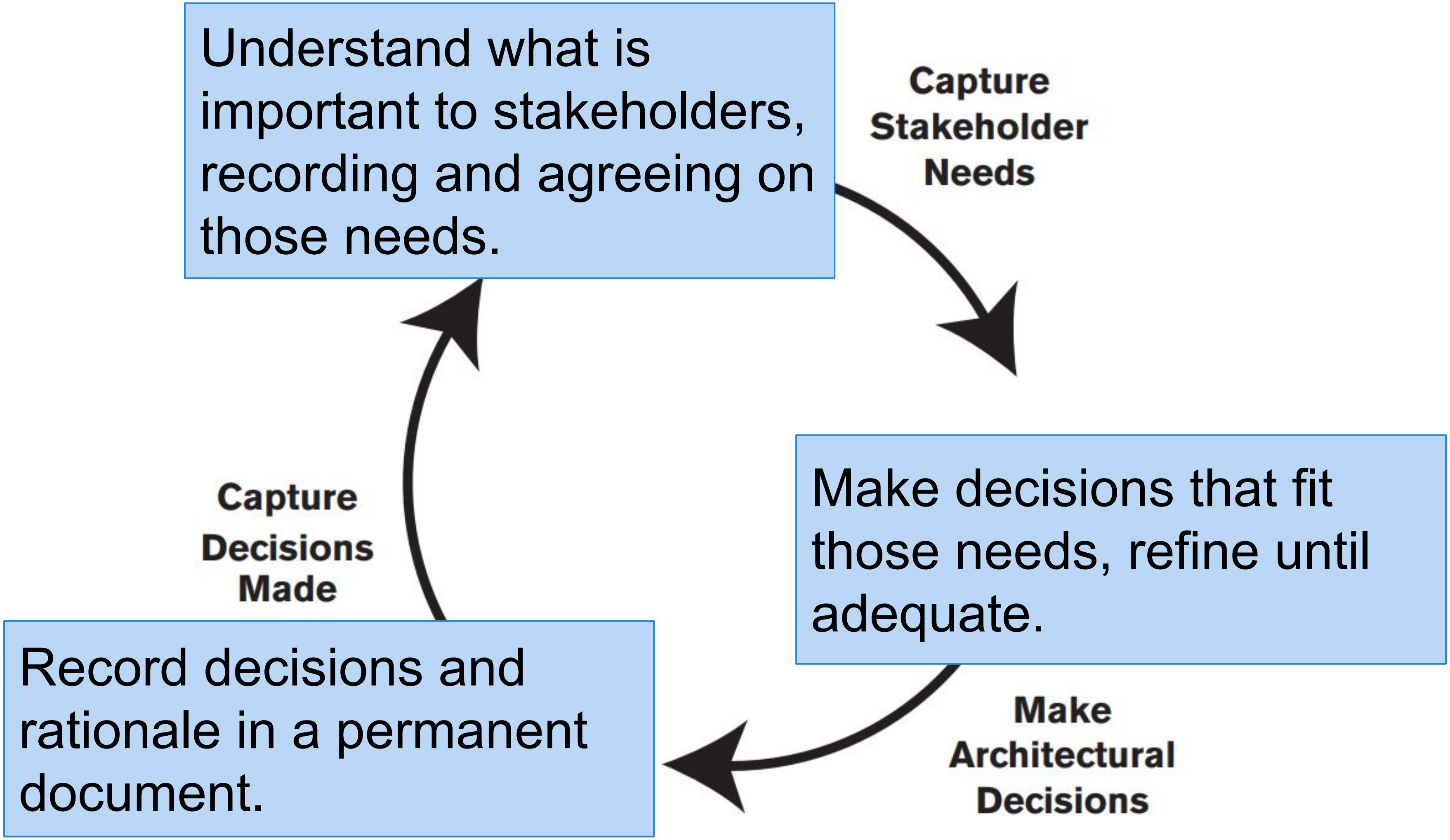
Infrastructure in Place



What Did We Do?

- Examined the stakeholders.
- Asked them about the expected behavior.
- Looked at our options (our infrastructure).
- Looked at the scope of the system.
- Made decisions about how to architect it.
- Repeat the cycle, using the current draft.

The Architecture Definition Process



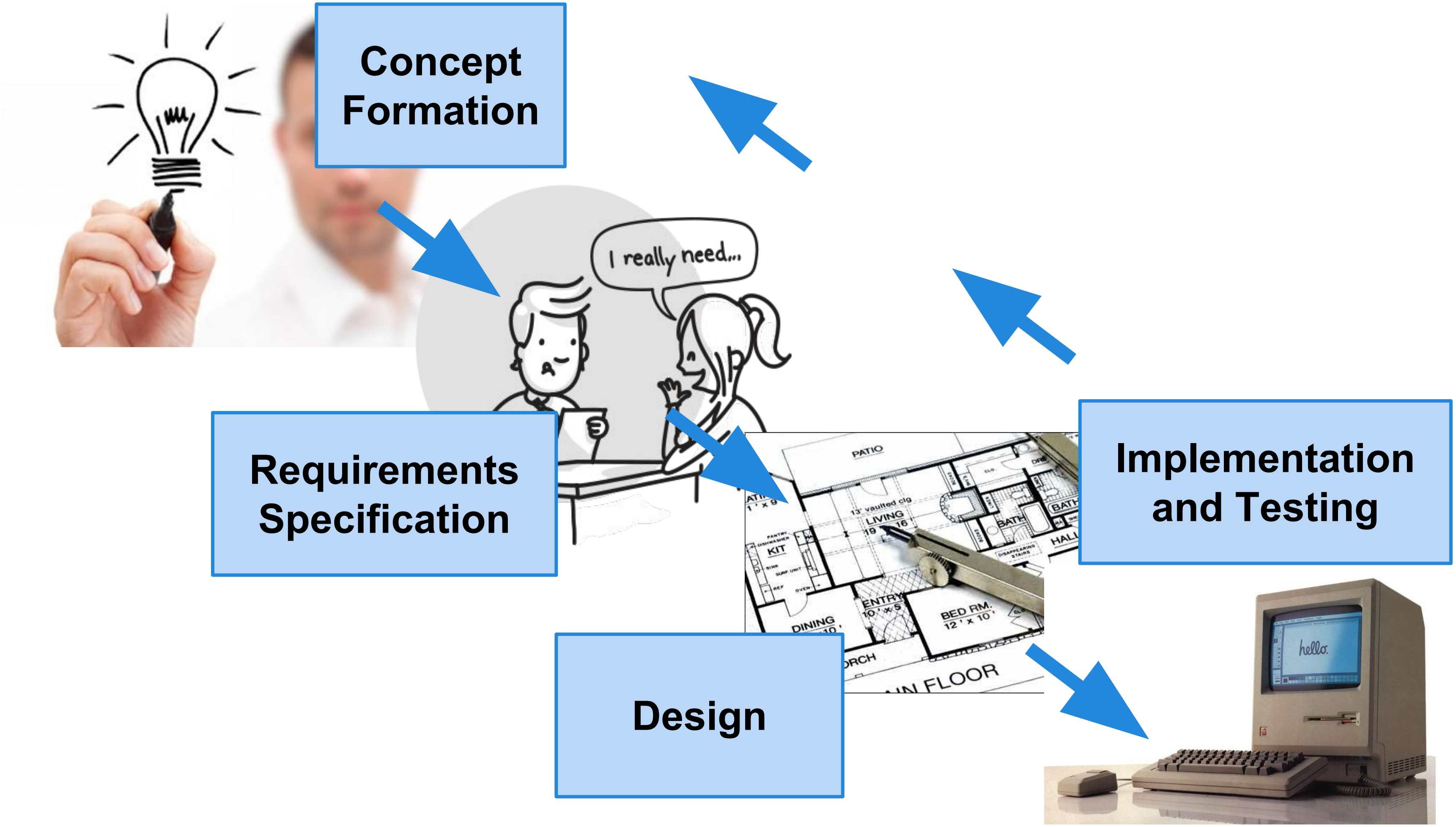
Understand what is important to stakeholders, recording and agreeing on those needs.

Make decisions that fit those needs, refine until adequate.

Record decisions and rationale in a permanent document.

Architecture, Requirements, and Design

Typical Development Process



Concept Formation

Requirements Specification

Design

Implementation and Testing

What is a requirement?

A **requirement** is a singular documented physical or functional need that a particular product must be able to perform.

“The software shall be able to calculate the sum of a column of integers.”

A statement that identifies a necessary attribute, capability, characteristic, or quality of a system for it to have *value and utility to a stakeholder*.

Requirement Specification Process

- A requirement's **specification** is a comprehensive technical description of how that requirement will be realized.
- The set of specifications will fully describe what the software will do and how it will be expected to perform.
- During the requirement specification process, we define **what the system does.**

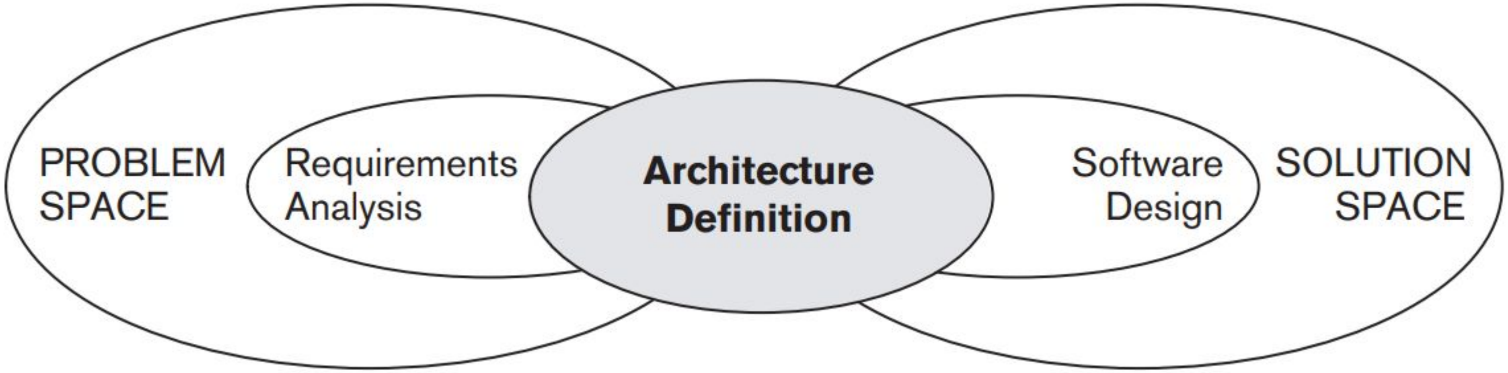
What is Design?

Design is the process of transforming a **problem** into a **solution**.

- In our case, transforming a requirements specification into a detailed description of the software to be implemented.

- Specification - ***what the system does***.
- Design - ***how the system does it***. A description of the structure of the solution.

Where is Architecture Definition?



- A bridge between specification and design.
  - Design focuses on solution space, targets developers. Focused on the classes that make up the codebase.
  - Requirement specification focuses on the problem space, ignores needs of developers.
  - Architecture bridges problem and solution spaces, balancing the needs of the stakeholders.

Requirements v. Architecture: Hospital Food

- Many ideas in solution space
  - Delivery drivers
  - VPNs, Card readers, firewalls
- These ideas **constrain** the problem space
  - (the requirements).
  - If delivery guy picks up food orders, patients have to choose meals earlier
  - If using delivery truck vs. car, we batch meals.
- It is a mistake to consider requirements without also working in the solution space.

Requirements v. Architecture

- Architecture reconciles requirements against the real world.
  - By analyzing, understanding, prioritizing requirements.
  - What are the implementation options?
  - What is the value of a requirement?
  - What is the cost of implementing a requirement?
    - *How* do we implement a requirement given the options in the solution space?

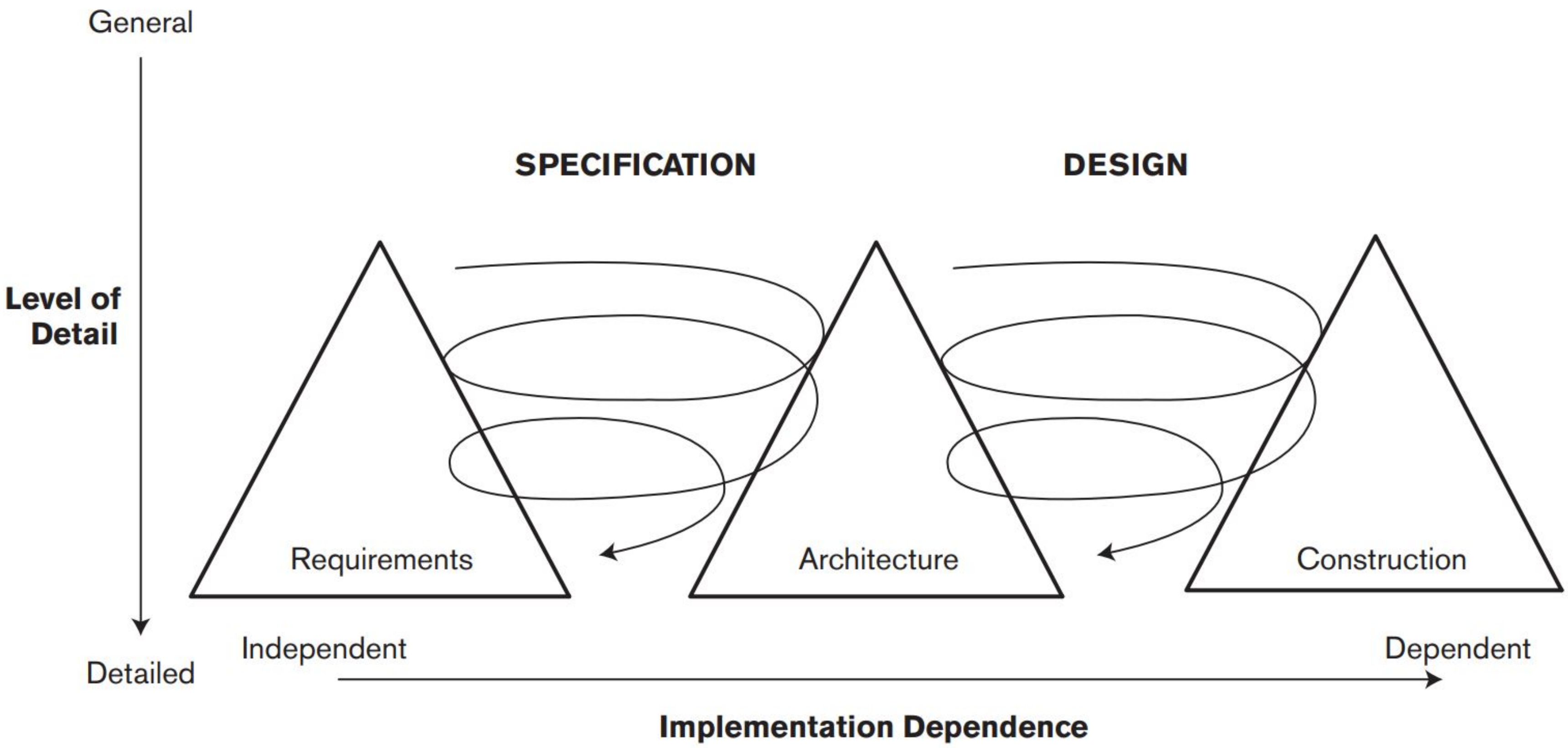
Design v. Architecture: Hospital Food

- Careful examination of infrastructure is key
    - Physical items (delivery trucks), existing software base, available language/library/OS support, 3rd party solutions
- These are all part of the solution space.
    - In this example, we could implement a system meeting the requirements with no new software.
- Architecture defines the interfaces and constrains the range of solutions.
    - It is not about writing code.
    - We are defining the architecture, not the class list.

Design v. Architecture

- Is an element *architecturally significant*?
  - A concern, problem, or element is architecturally significant if it has a wide impact on the structure of the system or on its quality properties.
- An architect must balance a high-level view and exploring the detail.
- As architecture evolves, review the scope and add detail when appropriate.
  - Do not be afraid of constraining the design.

The Three Peaks Model



The Three Peaks

- Requirements provide context for architecture definition by defining scope, functionality, and quality properties.
  - Architecture definition reveals requirement problems, helps stakeholders understand cost of meeting requirements. Allows refinement of requirements.
- Architecture definition constricts design.
  - Design provides refinement to architecture.
  - Alterations to architecture further shape design.

Requirements or Design Information?

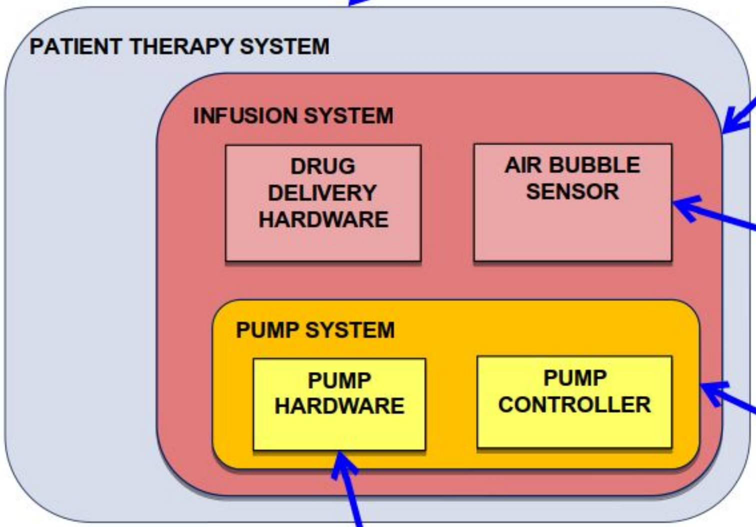
- The patient shall never be infused with a single air bubble more than 0.5ml volume.
- When a single air bubble more than 0.5ml volume is detected, the system shall stop infusion within 0.2 seconds.
- When a single air bubble more than 0.5ml volume is detected, the system shall issue an air-embolism command.
- The system shall always stop the piston at the bottom or top of the cylinder.
- When air-embolism command is received, the system shall stop piston movement within 0.1 second.

Both!

1. The patient shall never be infused with a single air bubble more than 0.5ml volume.

2. When a single air bubble more than 5ml volume is detected, the **system** shall stop infusion within 0. 2 seconds.

3. When a single air bubble more than 0.5ml volume is detected, the **system** shall issue an air-embolism command



4. The **system** shall always stop the piston at the bottom or top of the cylinder.

5. When piston stop is received, the **system** shall stop piston movement within 0.1 seconds.

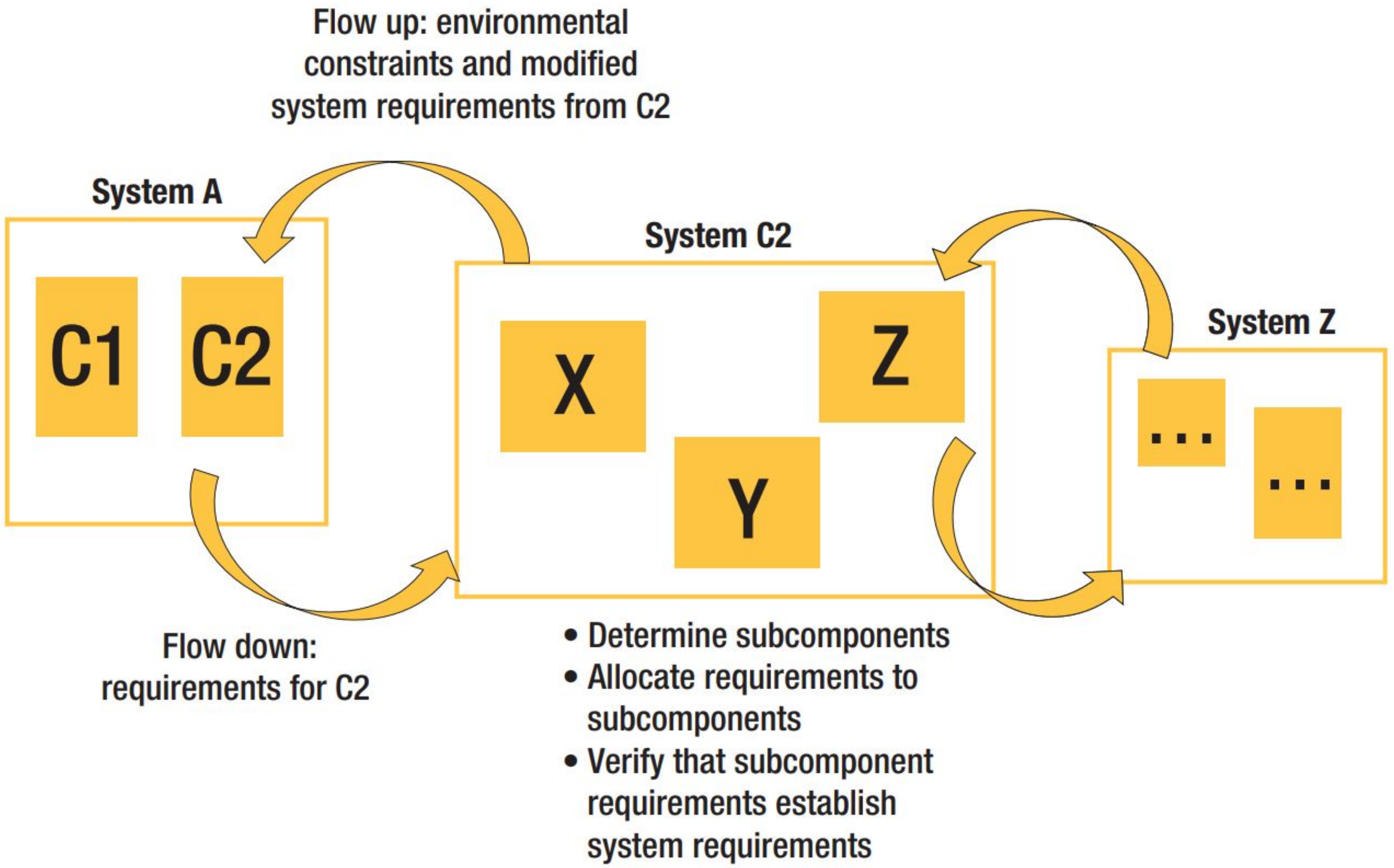
Your “What” is my “How”

- Systems are hierarchically architected as interacting subsystems
- Design considerations at one level determine what happens at the next level in the hierarchy.
- Requirements at one level are implemented as design decisions (an architecture), which imposes requirements on that architecture’s elements.
- Requirements vs. architectural design is a matter of perspective

Architecture Often Comes First

- May have candidate architectures from previous systems
  - Designer familiarity
  - Cost amortization
- Architecture may guide requirements discovery.
  - Program families
  - Certification or criticality requirements
- Architectural choices often restrict set of achievable system requirements.

Flow is Bidirectional


Flow up: environmental constraints and modified system requirements from C2
System A
C1
C2
System C2
X
Y
Z
System Z
...
...
Flow down: requirements for C2

- Determine subcomponents
- Allocate requirements to subcomponents
- Verify that subcomponent requirements establish system requirements

The Architecture Definition Process

Architecture Design Principles

- Must be driven by stakeholder concerns.
- Must encourage communication of decisions, principles, and the solution to stakeholders.
- Must ensure decisions are adhered to throughout the product lifecycle.
- Must be structured into a series of steps, with clear objectives, inputs, and outputs.

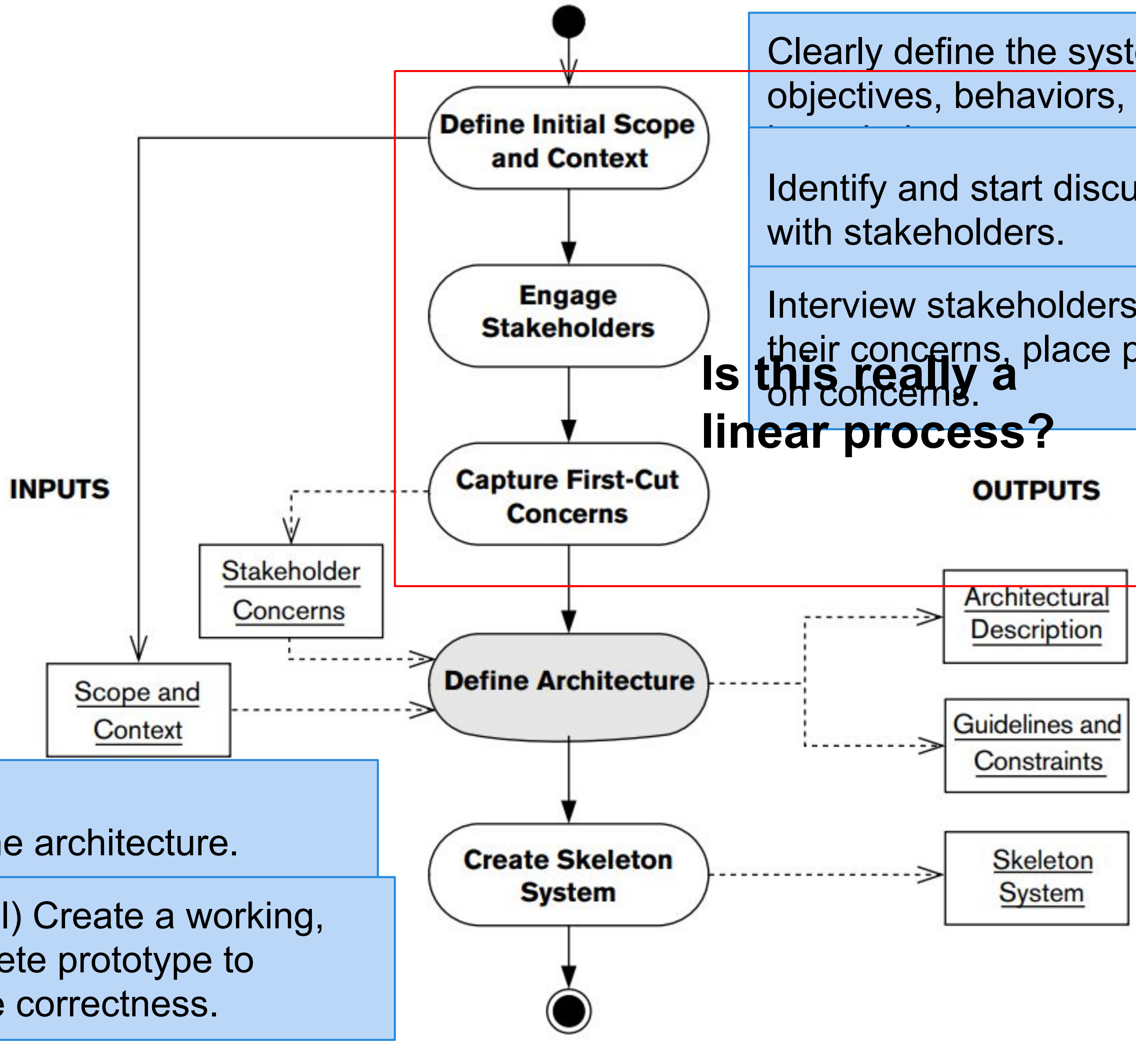
Architecture Design Principles

- Must be pragmatic, considering time and budget.
- Must be flexible, tailorable to your context.
- Must be technology-neutral, not dependent on one single technology or tool.
- Must integrate with the chosen development lifecycle.
- Must align with software engineering best practices to integrate with existing approaches.

Outcomes of the Process

- A sound architecture, also:
  - Clarification to requirements and other inputs.
    - Requirements and concerns may be revised.
  - Management of stakeholders’ expectations.
    - Explanation for compromises made.
  - Identification and evaluation of architectural options.
    - Rationale for the choices made.
  - Description of the architecture acceptance criteria.
    - What do stakeholders require to accept the proposed architecture?
  - Creation of a set of design inputs.
    - Ensures detailed design conforms to architecture

Outline of the Process



Clearly define the system objectives, behaviors, and

Identify and start discussions with stakeholders.

Interview stakeholders, identify their concerns, place priorities on concerns.

Is this really a linear process?

Create the architecture.

(optional) Create a working, incomplete prototype to illustrate correctness.

Define Initial Scope and Context

- **Aims:** Define boundaries of behavior and responsibilities and the system context.
- **Inputs:** Client needs, organizational strategy, IT architecture.
- **Outputs:** List of goals, list of exclusions, system context definition.

Engage the Stakeholders

- **Aims:** Identify the key stakeholders, begin initial discussions of their concerns.
- **Inputs:** Initial scope and context, organizational structure.
- **Outputs:** Defined list of stakeholder groups, people who will represent each group.

Capture First-Cut Concerns

- **Aims:** Clearly understand and prioritize the concerns that each stakeholder group has.
- **Inputs:** Initial scope and context, stakeholder list.
- **Outputs:** Initial definition of the prioritized concerns for each stakeholder group.

Architecture Definition

Assess, validate, and refine initial inputs.

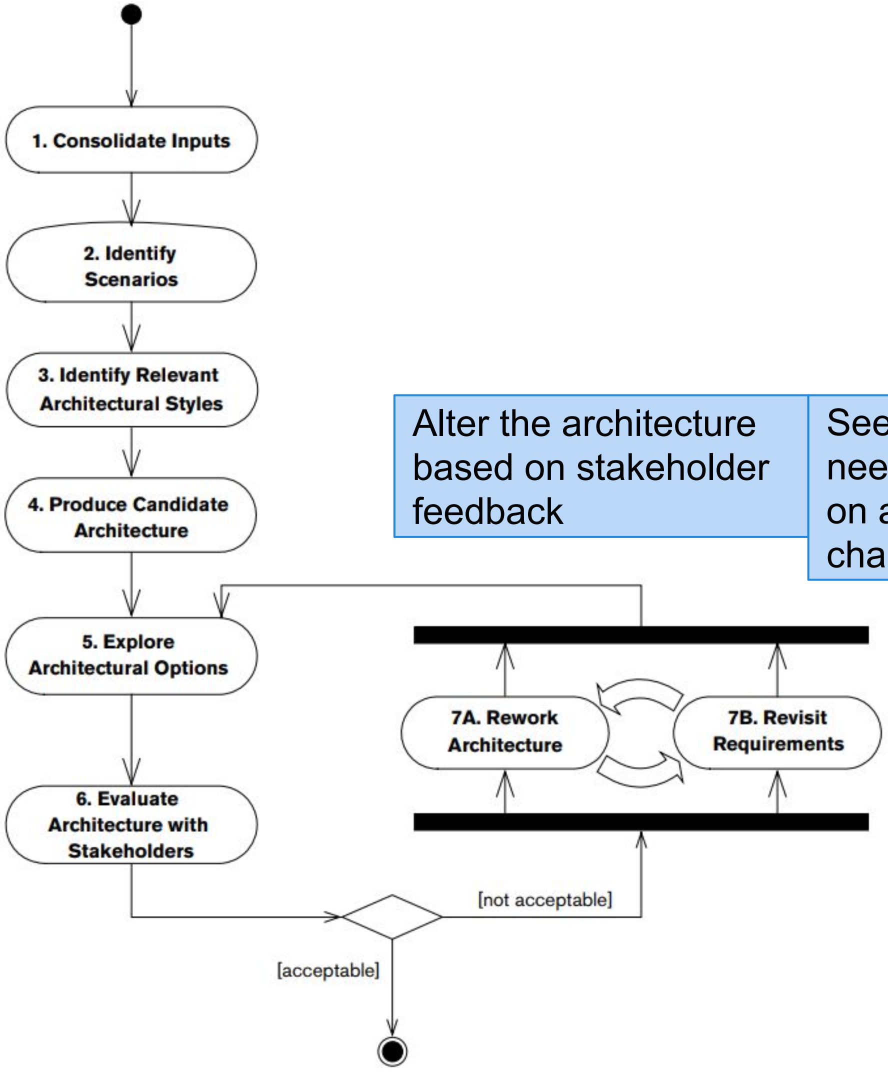
Identify scenarios that illustrate the requirements.

Identify proven architectural styles that fit the problem.

Create a draft architecture that can be refined.

Explore possibilities and make key decisions.

Work with stakeholders to assess the draft.



1. Consolidate Inputs
2. Identify Scenarios
3. Identify Relevant Architectural Styles
4. Produce Candidate Architecture
5. Explore Architectural Options
6. Evaluate Architecture with Stakeholders
7A. Rework Architecture
7B. Revisit Requirements
[not acceptable]
[acceptable]

Alter the architecture based on stakeholder feedback

See if requirements need to change based on architecture changes

Consolidate the Inputs

- **Aims:** To understand, validate, and refine the initial inputs.
- **Inputs:** Scope and context, stakeholder concerns
- **Activities:** Take the inputs, resolve inconsistencies, answer open questions, and produce a baseline.
- **Outputs:** Refined inputs

Identify Scenarios

- **Aims:** To identify a set of scenarios that illustrate the system's requirements.
- **Inputs:** Consolidated inputs.
- **Activities:** Produce scenarios that characterize the most important attributes required of the architecture, that can be used to evaluate a proposed architecture.
- **Outputs:** Architectural scenarios.

Identify Relevant Styles

- **Aims:** To identify proven architectural styles that could be used as a basis for the system.
- **Inputs:** Consolidated inputs, scenarios.
- **Activities:** Review existing catalogs of architectural styles, and consider system organizations that have worked before. Identify those that appear to be relevant.
- **Outputs:** Architectural styles to consider.

Produce Candidate Architecture

- **Aims:** To create a first-cut architecture for the system that reflects its primary architectural concerns.
- **Inputs:** Consolidated inputs, relevant architectural styles, viewpoints, and perspectives.
- **Activities:** Produce architectural views, define ideas, guided by best practice.
- **Outputs:** Draft architectural views

Explore Architectural Options

- **Aims:** Explore possibilities for the system and make the key architectural decisions.
- **Inputs:** Draft architectural views, scenarios, viewpoints, and perspectives.
- **Activities:** Apply scenarios to the draft models. Take risk, concern, or uncertainty and further explore the requirements, problems, and issues.
- **Outputs:** Refined architectural views

Evaluate Architecture

- **Aims:** Evaluate with stakeholders, capture problems, and gain acceptance.
- **Inputs:** Consolidated inputs, architectural views, and perspective outputs.
- **Activities:** Evaluate architecture with representative stakeholders. Capture and agree on improvements or comments.
- **Outputs:** Architectural review comments.

Rework the Architecture

- **Aims:** Address any concerns that have emerged during the evaluation task.
- **Inputs:** Views; review comments; relevant styles, viewpoints, and perspectives.
- **Activities:** Take the results of the evaluation and address them. This involves functional analysis, the use of viewpoints and perspectives, and prototyping.
- **Outputs:** Reworked architectural views.

Revisit the Requirements

- **Aims:** Consider changes to requirements in light of architectural evaluation.
- **Inputs:** Architectural views, architectural review comments.
- **Activities:** Work may reveal inadequate, inconsistent, infeasible requirements. Revisit requirements with stakeholders and obtain agreement to revisions.
- **Outputs:** Revised requirements.

Create a Skeleton System

- **Aims:** Create a working (but incomplete) implementation of the architecture that can evolve into a full system.
- **Inputs:** Architectural definition, guidelines, constraints.
- **Outputs:** A limited working system that illustrates that the architecture can address at least one of your scenarios.

What is the role of the software architect?

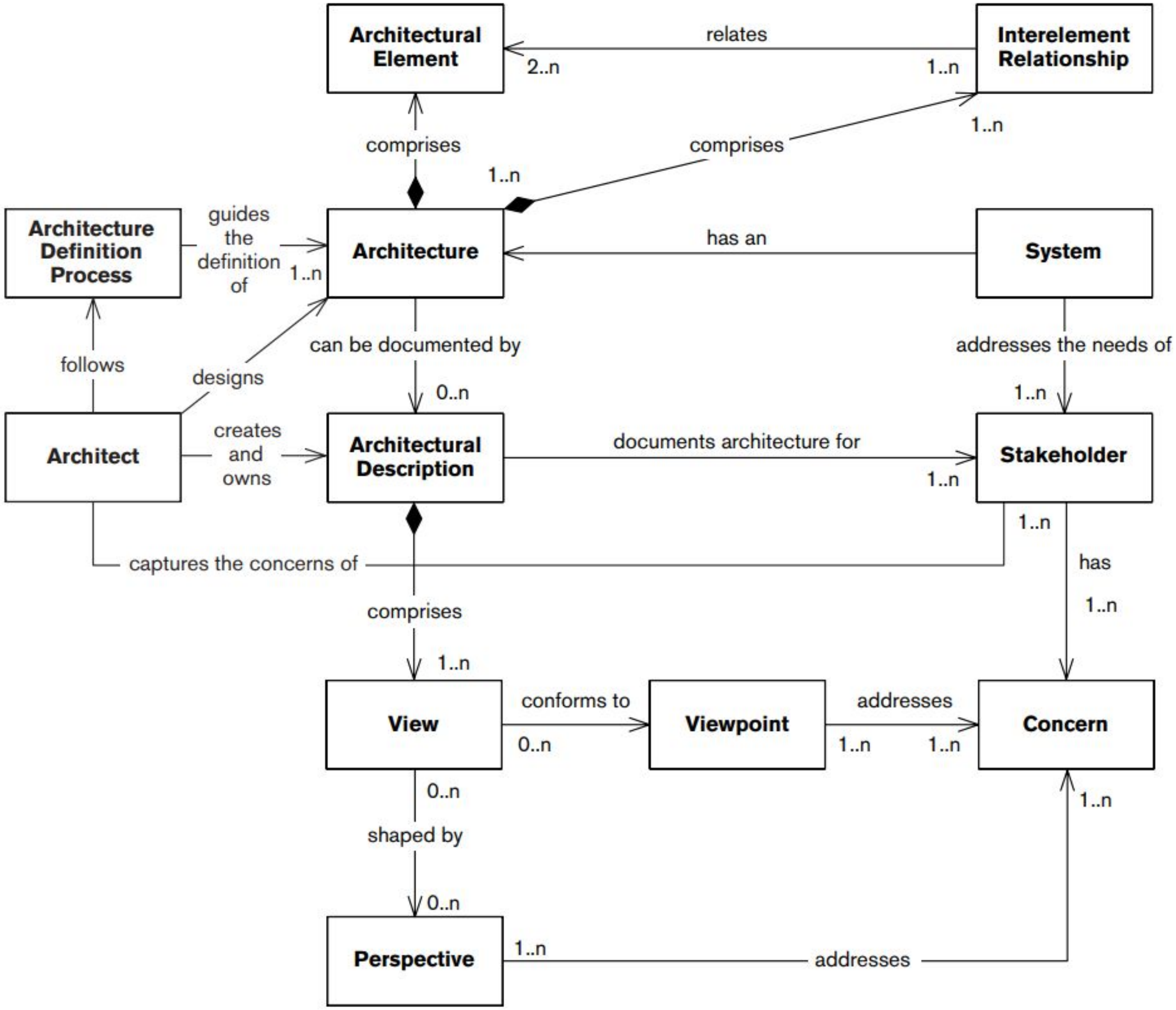
The Role of the Architect

- The **architect** designs, documents, and leads the construction of a system that meets the needs of all its stakeholders.
  - An architect identifies and engages stakeholders.
  - An architect understands and captures the stakeholders’ concerns.
  - An architect creates and owns the definition of an architecture that addresses these concerns.
  - An architect takes a leading role in realizing the architecture as a final product.

The Role of the Architect

- The **architect** owns the big picture.
    - You develop and maintain a high-level view of the elements in a system.
    - This is translated into detailed design, coded, tested, and deployed.
    - You ensure the “big picture” is the right one.
        - All problems have multiple possible solutions. You identify the correct one.
- Architects are involved at all stages of development, ensuring the product conforms to the architecture.

Bringing it Together



Key Points

- Process of architecture definition includes:
  - Defining scope and context, engaging stakeholders, capturing first-cut concerns, defining the architecture, and creating a skeleton system.
- The activities involved in creating a architecture definition include:
  - Consolidating inputs, identifying scenarios, identifying relevant styles, producing a candidate, exploring options, evaluating the architecture, reworking the architecture, and revisiting the requirements.

Key Points

- Architecture bridges requirements analysis and detailed class design.
- Architects “own the big picture”.
    - They design, document, and lead the construction of a system that meets the needs of all its stakeholders.

Next Time

- **No class or office hours Thursday!**
- Next Tuesday: Setting Context and Identifying Stakeholders
    - Sources: Rozanski & Woods: ch. 9, 16

- Homework: Reading Assignment 1 - Due Thursday
- Project Part 1 is up! Due September 18.

Project

- Semester-long project intended to enable you to critique a software architecture.
    - What are the major components of the architecture?
    - What patterns or styles are used?
    - What architectural views are relevant for stakeholders?
    - How well does the architecture align with stakeholder interests?
    - How amenable is the architecture to change?
- Enable you to extend or refactor architecture
    - Justify why this better meets stakeholder needs.

Phase 1: Choosing a System

- You will understand, document, and extend an existing architecture.
- Phase 1 - choose a system
    - Must be a real system
    - Source code must be available
        - Open-source, or at least, available to your team and me.
    - Must be at least 10,000 source LOC.

Phase 1: Choosing a System

- Include a description of the system
- Explain why it is architecturally interesting.
- List the quality attributes relevant to the system and its stakeholders.
  - For each, explain why this attribute is important.
- No minimum length. However, it is expected to be detailed and provide a complete treatment of the subject.
  - (Probably at least 5 pages)