



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# Software Architecture

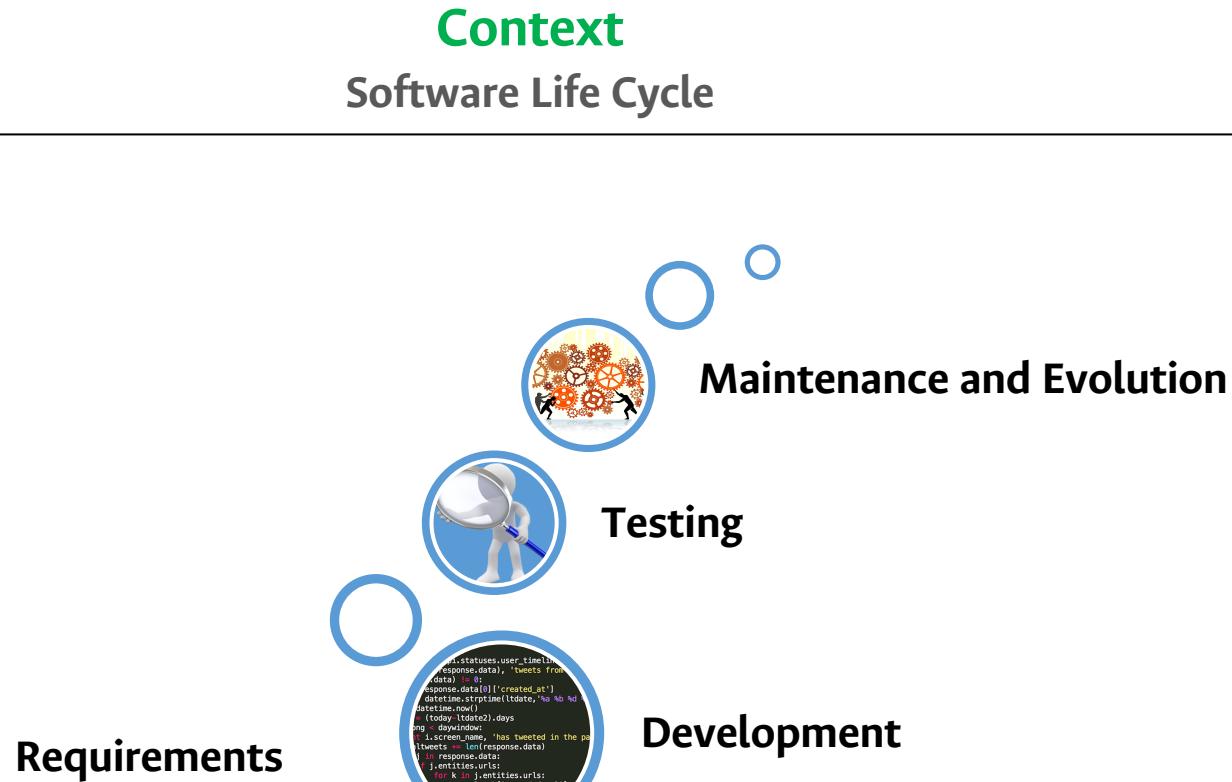
## Software Architecture Fundamentals

**Jeisson Andrés Vergara Vargas**

Departamento de Ingeniería de Sistemas e Industrial  
<http://colswe.unal.edu.co/~javergarav/>  
javergarav@unal.edu.co

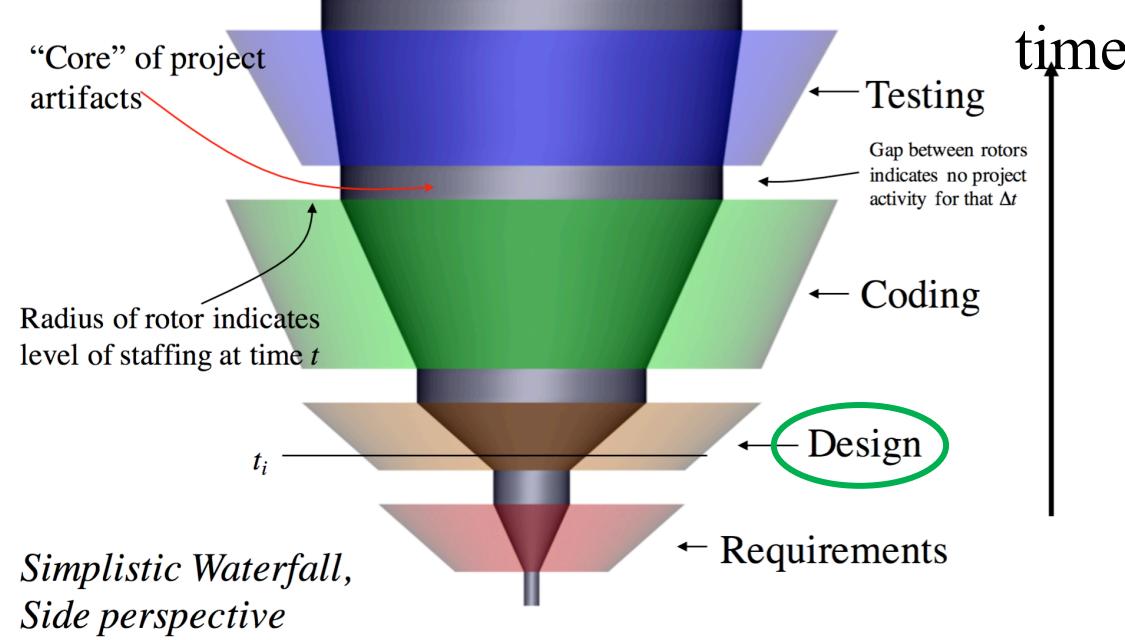
2021-I

©



## Context

### Software Life Cycle



# Software Design

## A Metaphor

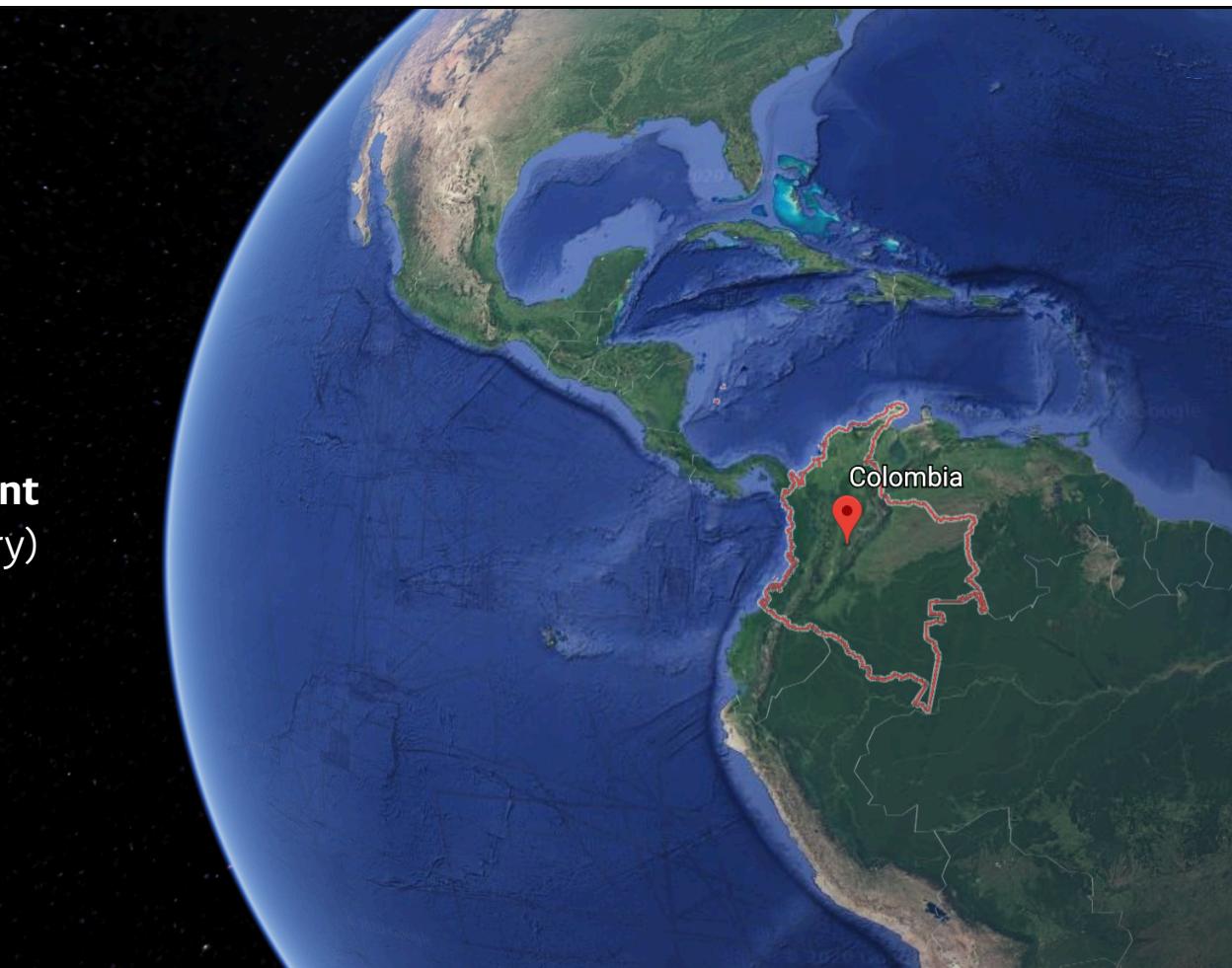
A **software system**  
(the Earth)



# Software Design

## A Metaphor

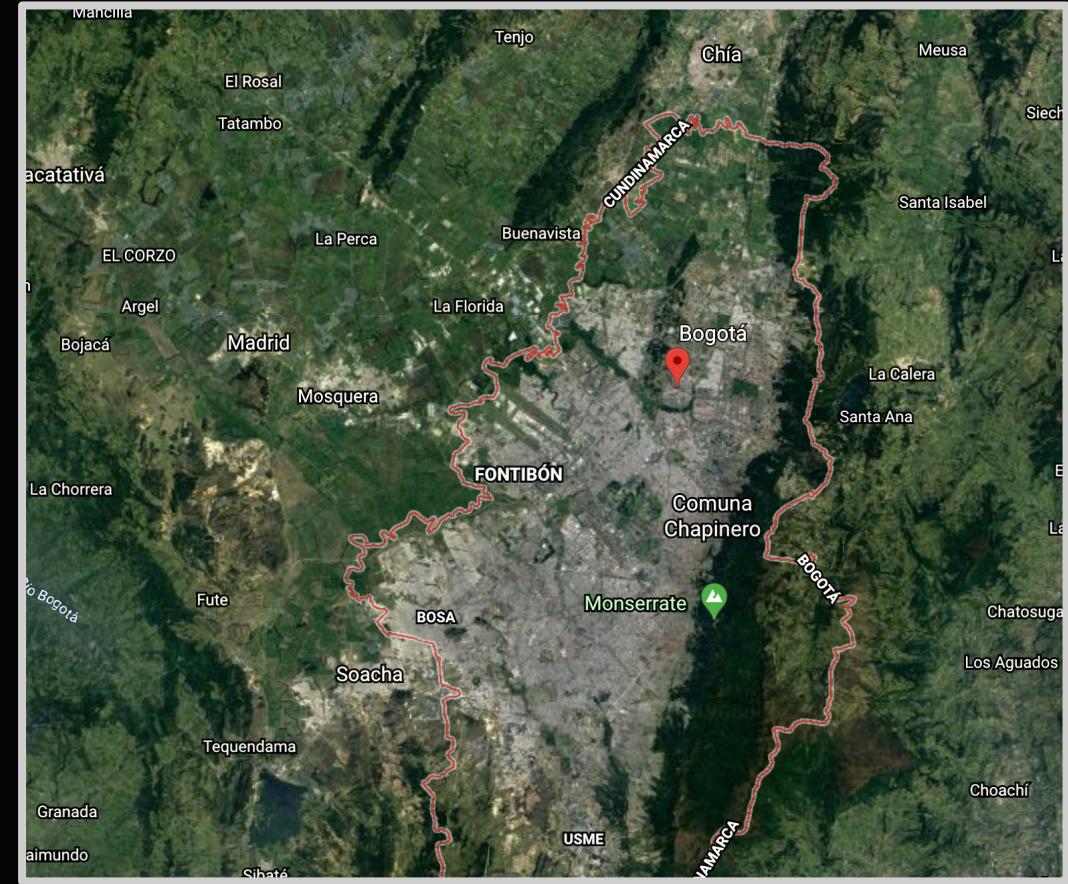
A **component**  
(a country)



# Software Design

## A Metaphor

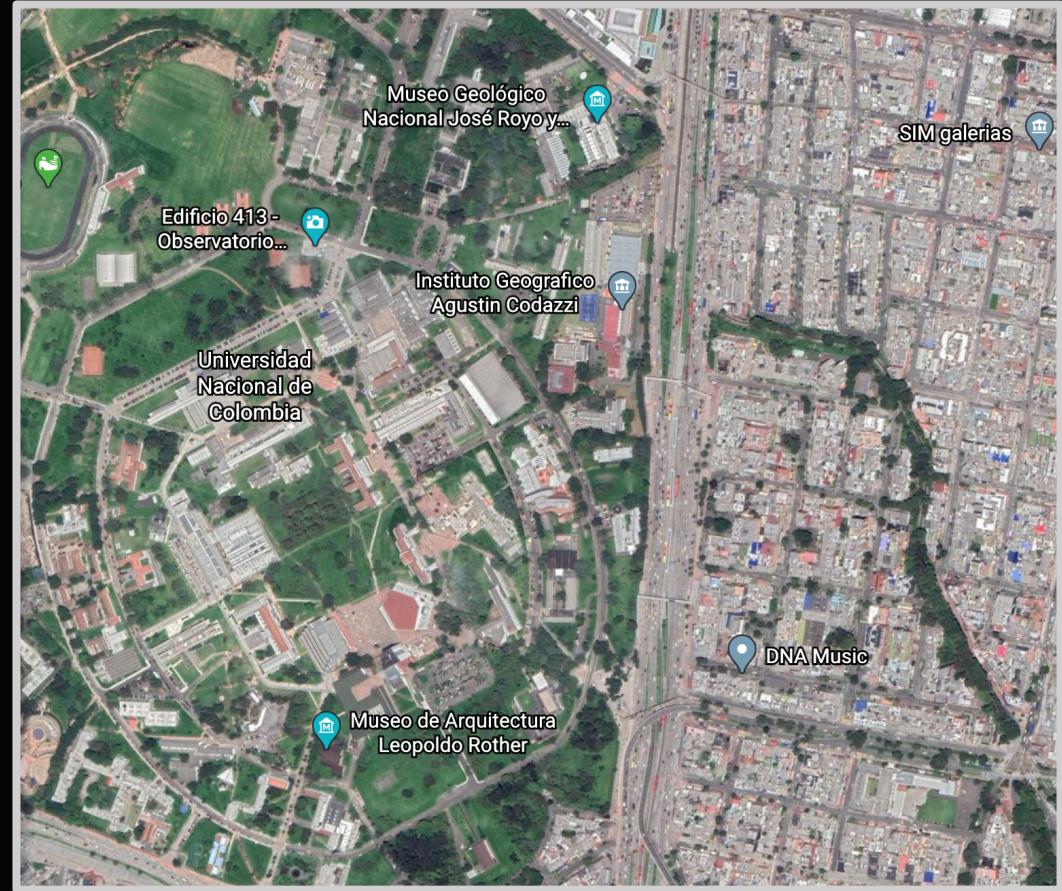
A class  
(a city)



# Software Design

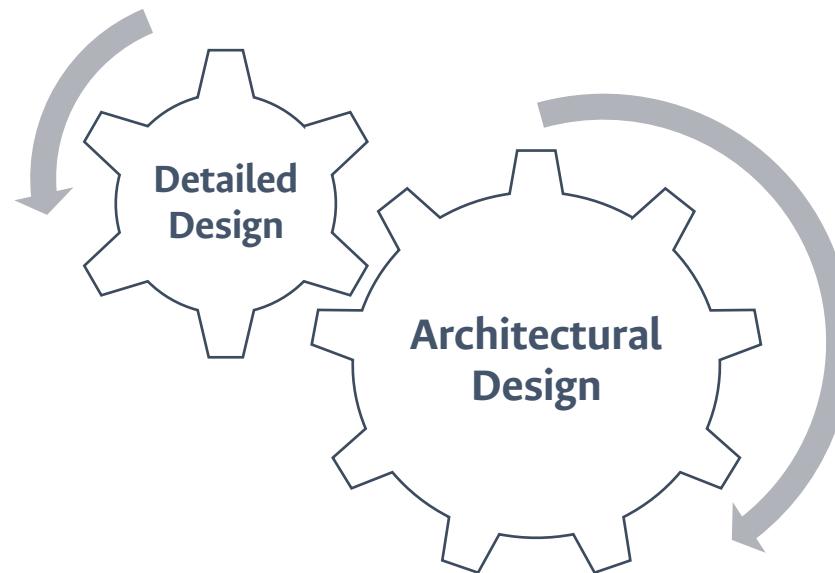
## A Metaphor

The **source code**  
(streets and buildings)



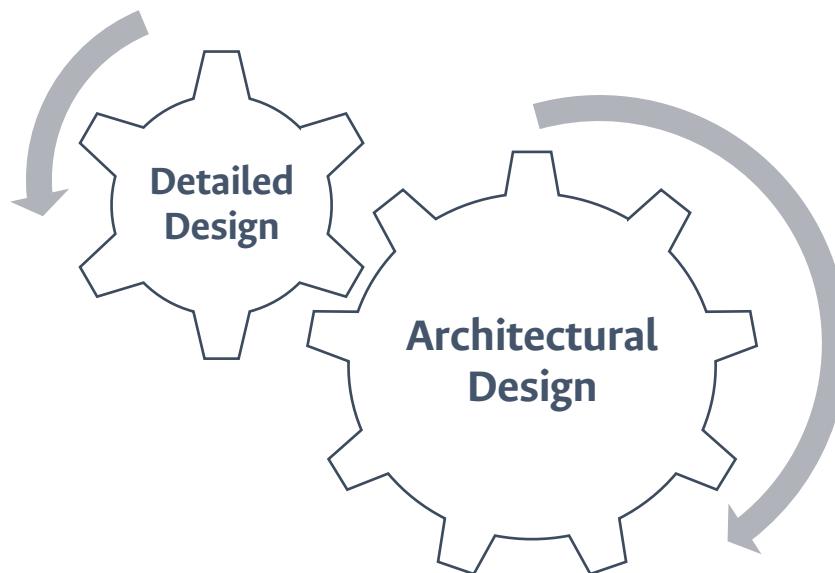
# Software Design

## Approaches



# Software Design

## Approaches



### Architectural Design

It specifies the **fundamental structure** and **patterns** of the system under development.

### Detailed Design

It focuses on all of the **implementation details** necessary to implement the **architecture** that is specified.

# Software Design

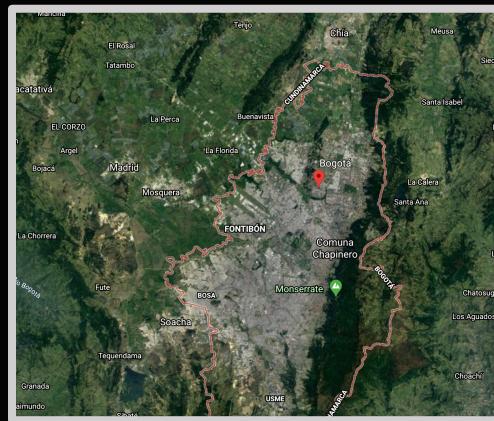
## Approaches



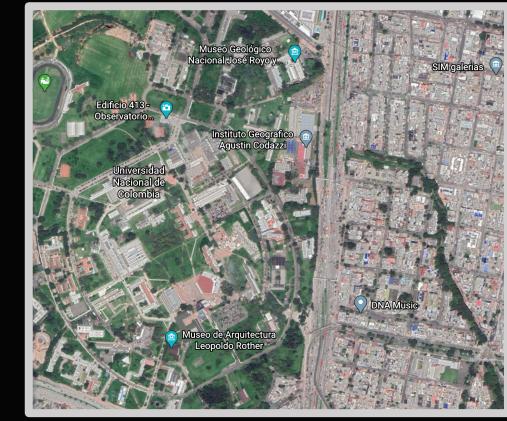
Architectural Design

# Software Design

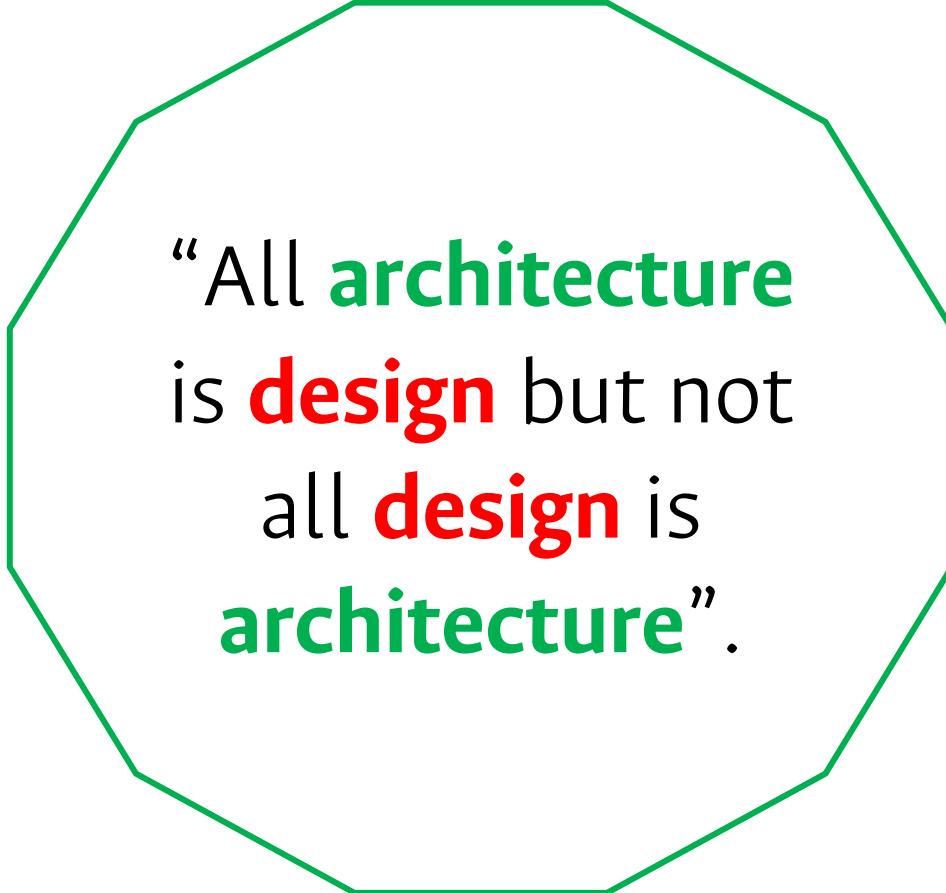
## Approaches



### Detailed Design



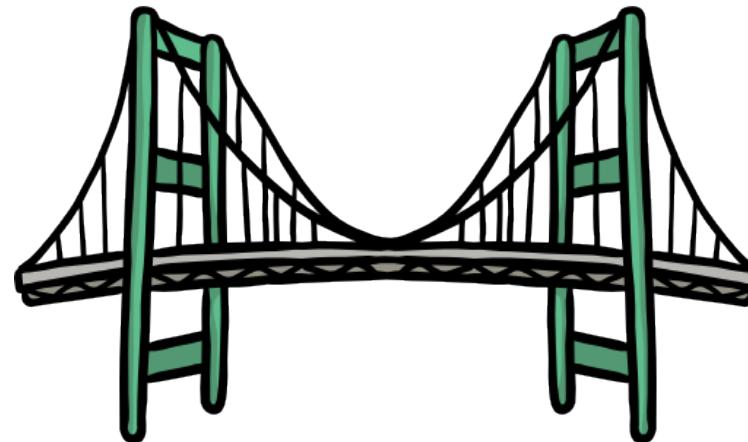
## Software Design



“All **architecture**  
is **design** but not  
all **design** is  
**architecture**”.

# Architectural Design

## A Metaphor



A bridge



Architect

Architectural Design

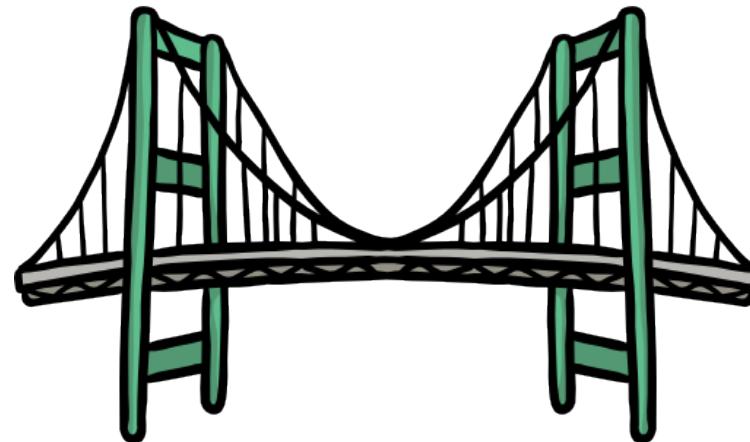


Civil Engineer

Structural Design

# Architectural Design

## A Metaphor



A software system



Software Architect

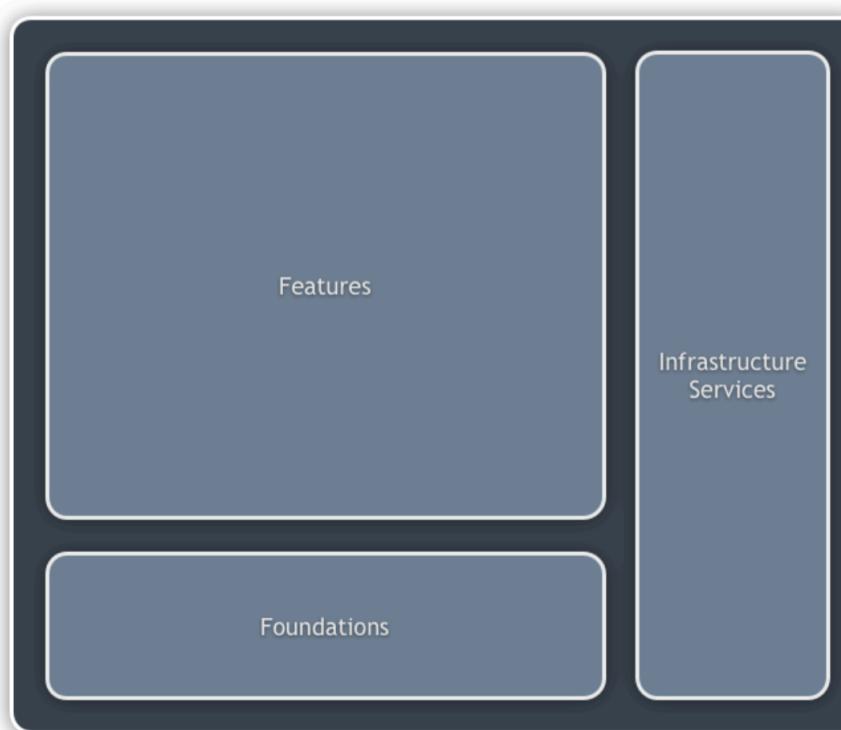
Architectural Design

Structural Design



# Software Architecture

## Application Architecture



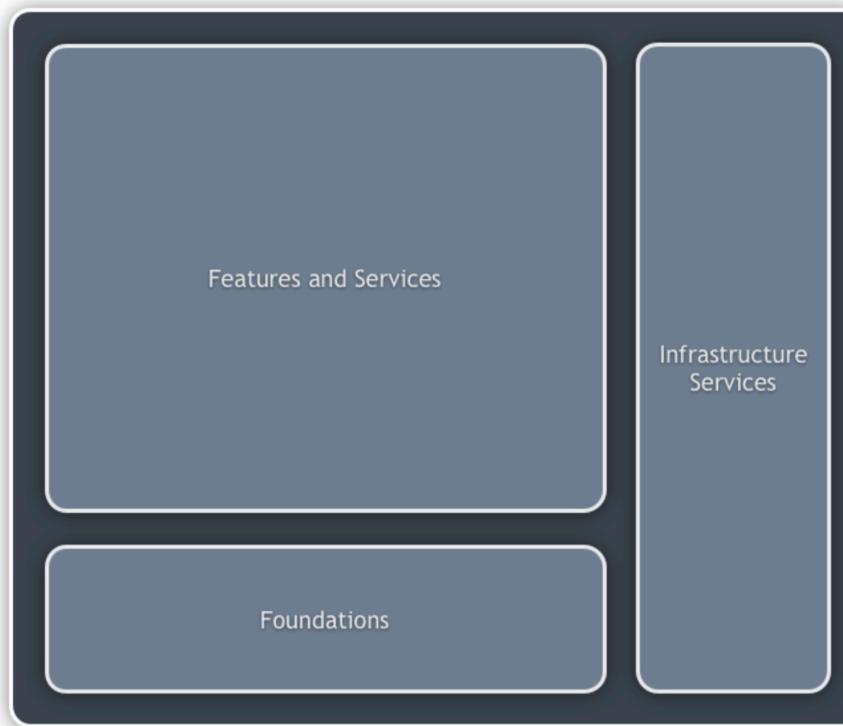
*The building blocks are  
software ... programming  
languages, libraries,  
frameworks, code, etc*



*Described in terms of  
classes, components, modules,  
functions, interactions,  
patterns, etc*

# Software Architecture

## System Architecture



*The building blocks are software and hardware ... technologies, libraries, frameworks, infrastructure, servers, etc*

*Described in terms of components, services, subsystems, systems, interactions, integration, interoperability, patterns, etc*

# Software Architecture

## Definition

**Software architecture** is simply the combination of **application** and **system** architecture.

[BROWN]

# Software Architecture

## Definition

A **software** system's **architecture** is the set of **principal design decisions** made about system.

[TAYLOR]

# Software Architecture

## Definition

**Software architecture** of a computing system is the set of **structures** needed to **reason** about the system, which comprises software **elements**, **relations** among them, and **properties** of both.

[CLEMENTS]

# Software Architecture

## Definition

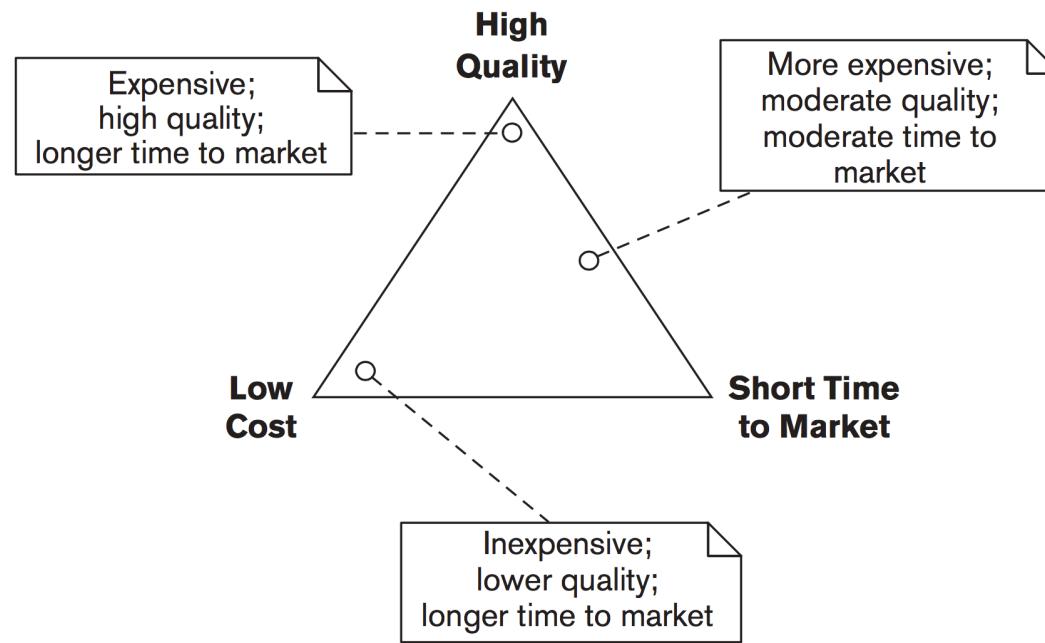
The **software architecture** of a program or computing system is the **structure** or **structures** of the system, which comprise software **elements**, the externally visible **properties** of those elements, and the **relationships** among them.

[BASS]

# Software Architecture

## Concerns, Stakeholders and Architecture

A **concern** about an architecture is a **requirement**, an **objective**, an **intention**, or an **aspiration** that a **stakeholder** has for that architecture.



## Software Architecture

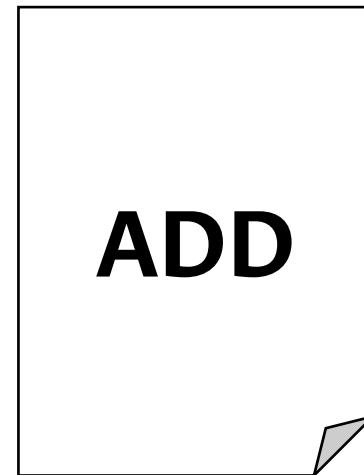


“A **good architecture** is one that successfully meets the objectives, goals, and needs of its **stakeholders**”.

# Software Architecture

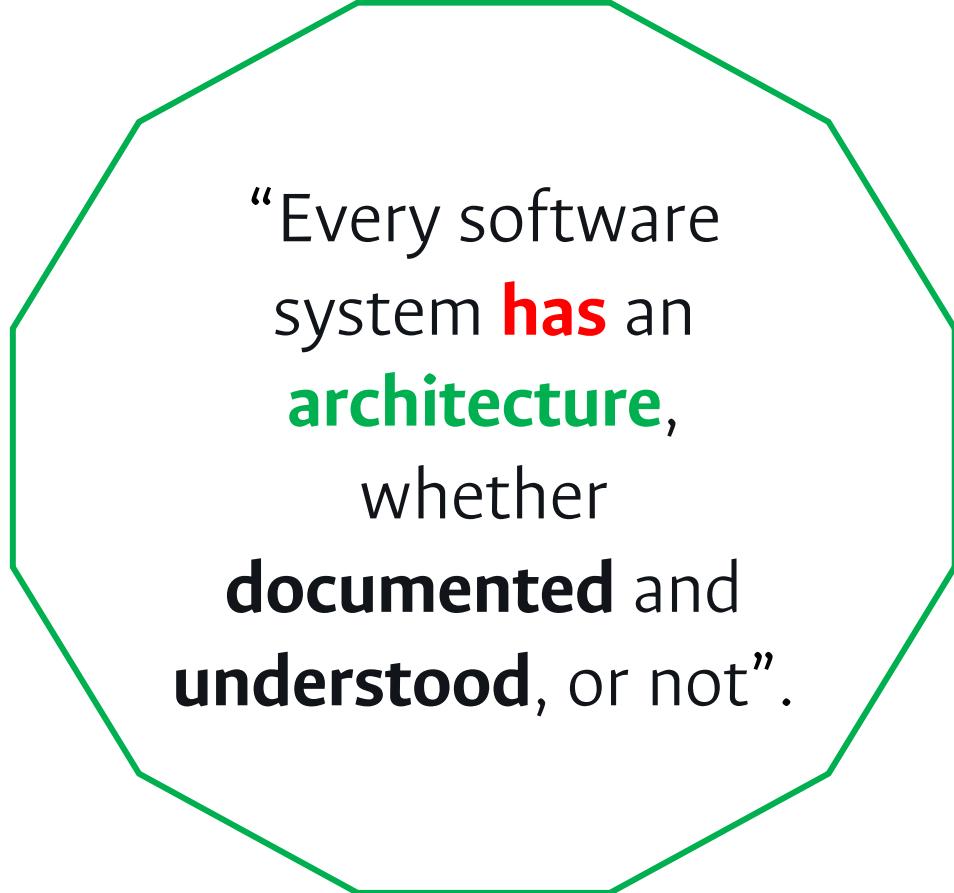
## Architectural Description

An architectural description (AD) is a **set of products that documents** an architecture in a way its **stakeholders** can **understand** and **demonstrates** that the architecture has met their **concerns**.



**Architecture Description Document**

## Software Architecture



“Every software system **has** an **architecture**, whether **documented** and **understood**, or not”.

# Software Architecture

## Architectural Elements

An **architectural element** (or just **element**) is a **fundamental piece** from which a system can be considered to be constructed.

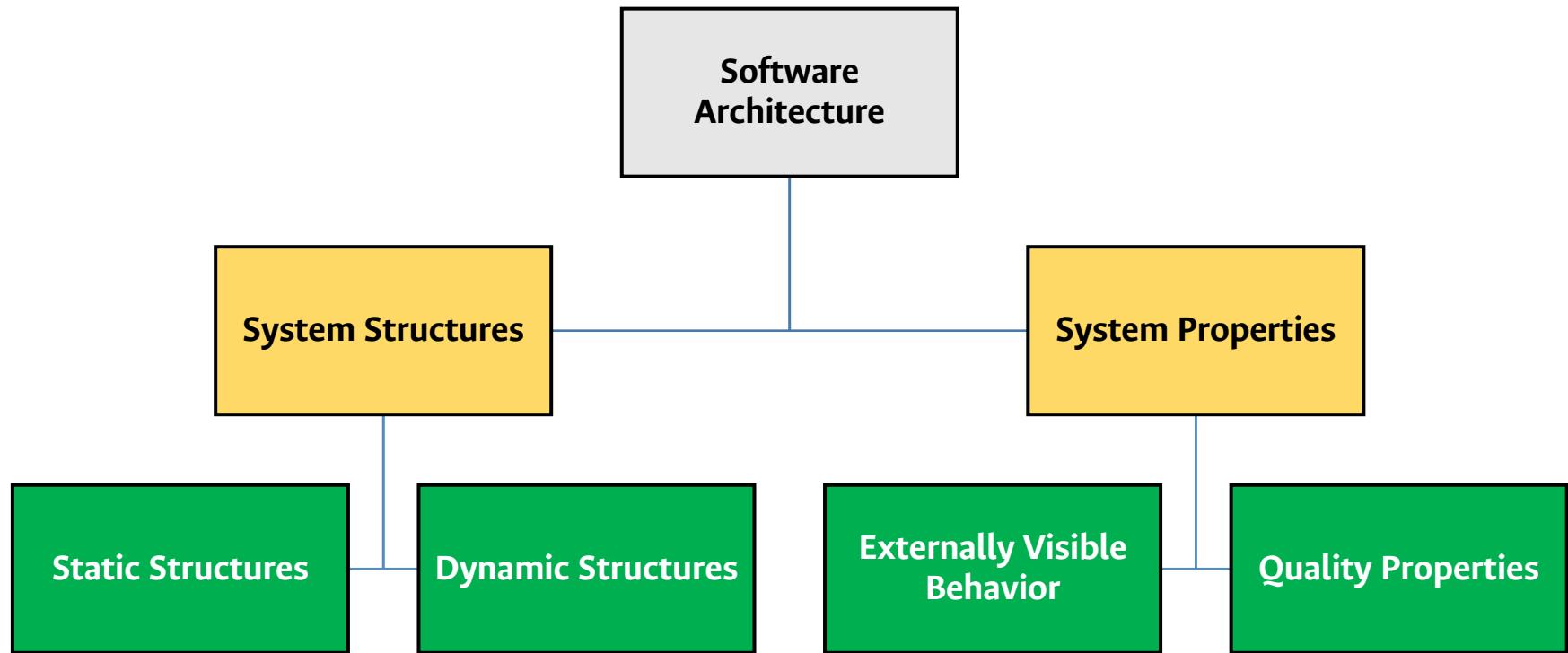
A clearly defined set of **responsibilities**.

A clearly defined **boundary**.

A set of clearly defined **interfaces**.

# Software Architecture

## Structures and Properties



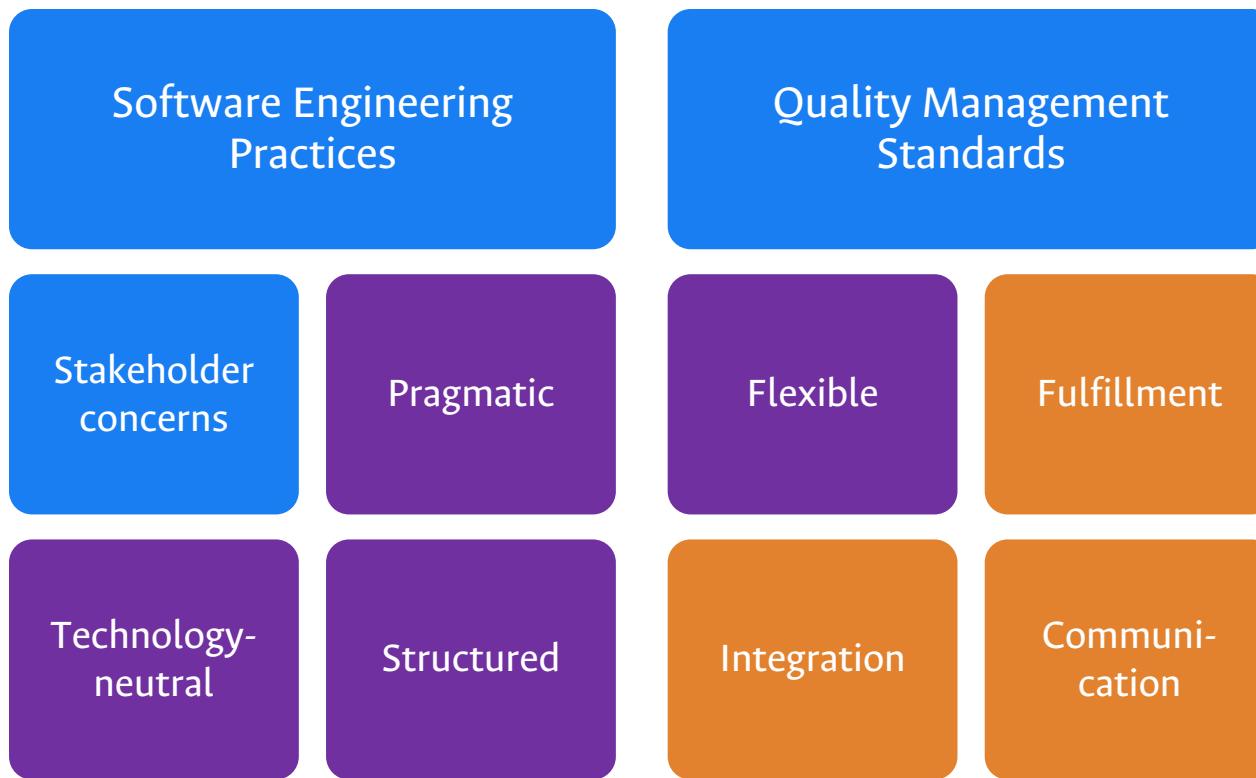
# Software Architecture

## Structures and Properties

<b>Structures</b>	<p>The <b>static structures</b> of a software system define its internal <b>design-time elements</b> and their arrangement.</p>
	<p>The <b>dynamic structures</b> of a software system define its <b>runtime elements</b> and their interactions.</p>
<b>Properties</b>	<p>The <b>externally visible behavior</b> of a software system defines the <b>functional interactions</b> between the system and its environment.</p>
	<p>A <b>quality property/attribute</b> is an externally visible, <b>nonfunctional property</b> of a system such as: security, interoperability, performance, scalability, resilience, high availability and usability.</p>

# The Architecture Definition Process

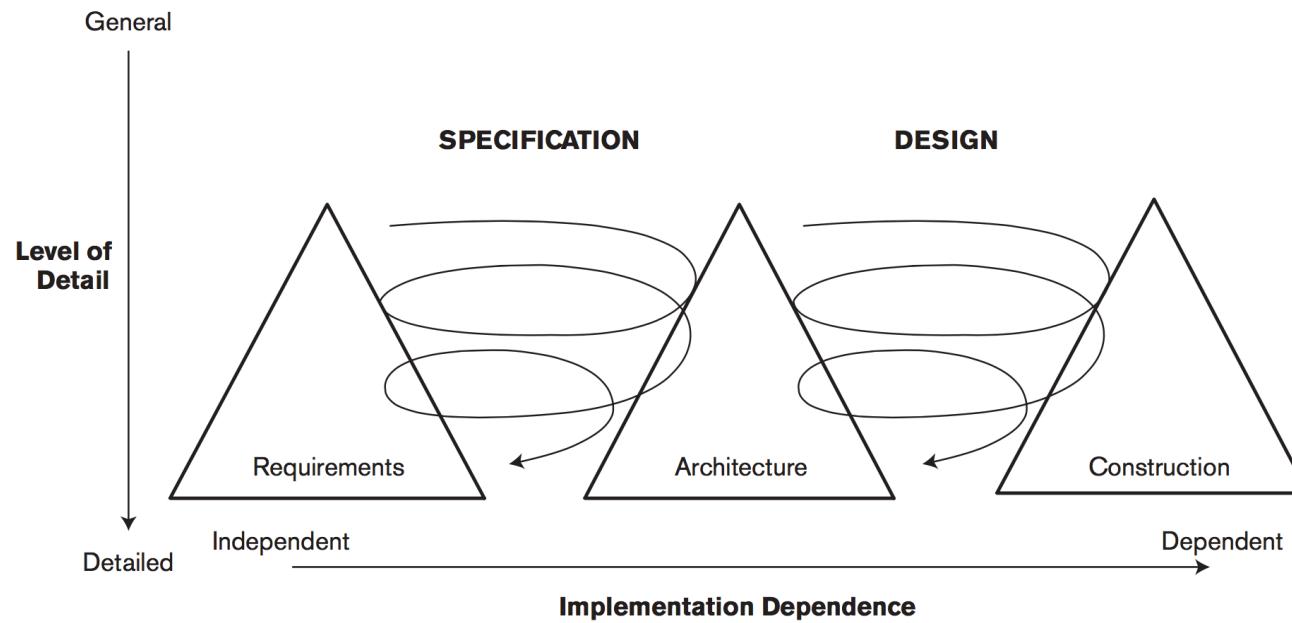
## Guiding Principles



# The Architecture Definition Process

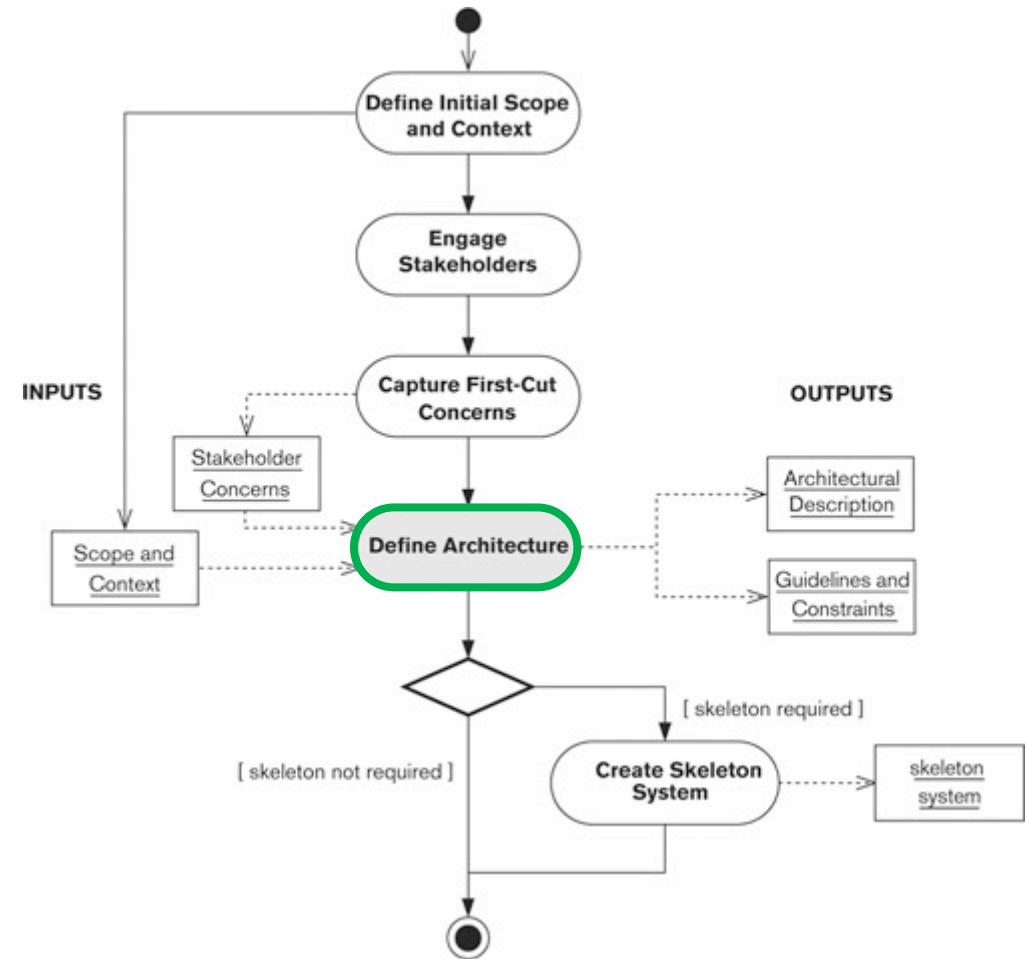
## Process Outcomes

1. Clarification of **requirements** and of **other inputs** to the process.
2. Management of **stakeholders' expectations**.
3. Identification and evaluation of **architectural options**.
4. Description of **architectural acceptance criteria**.
5. Creation of a set of **design inputs**.



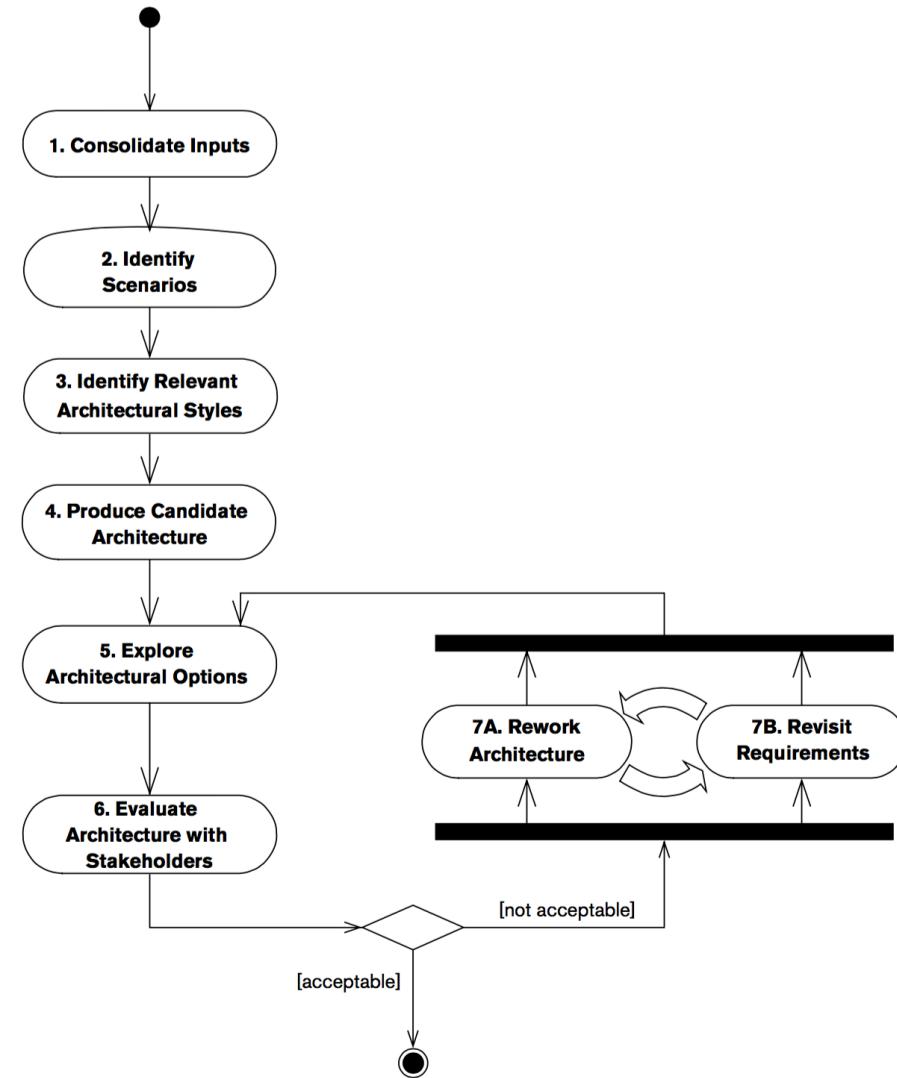
# The Architecture Definition Process

## Supporting Activities



# The Architecture Definition Process

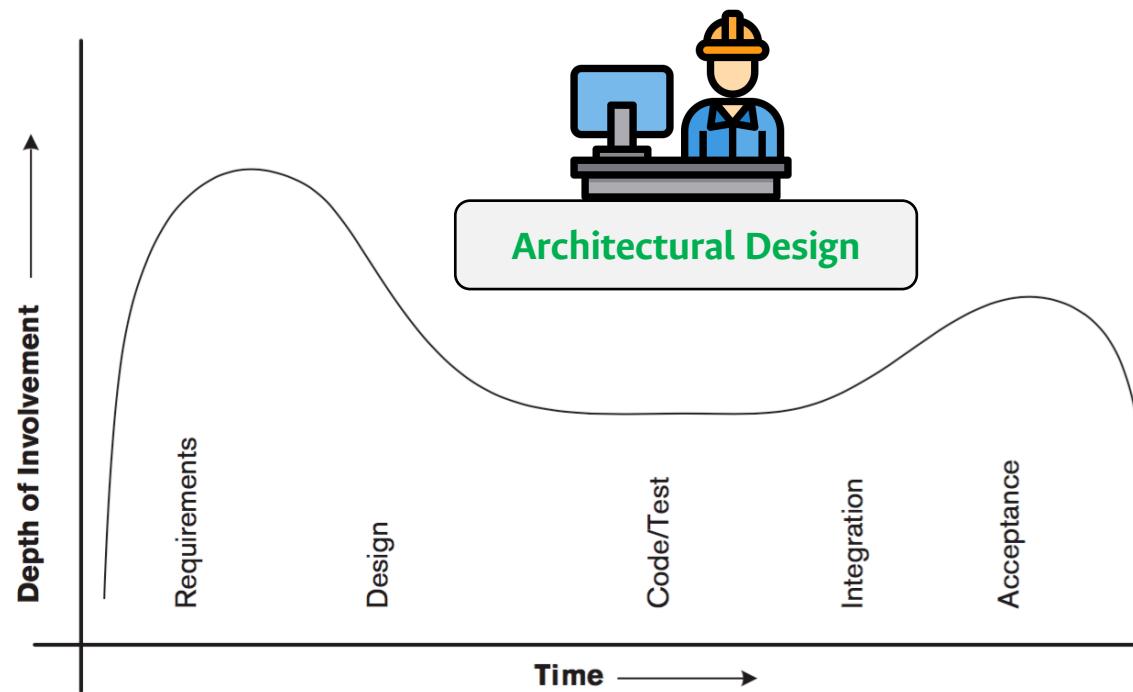
## Architecture Definition Activities



# Software Architect

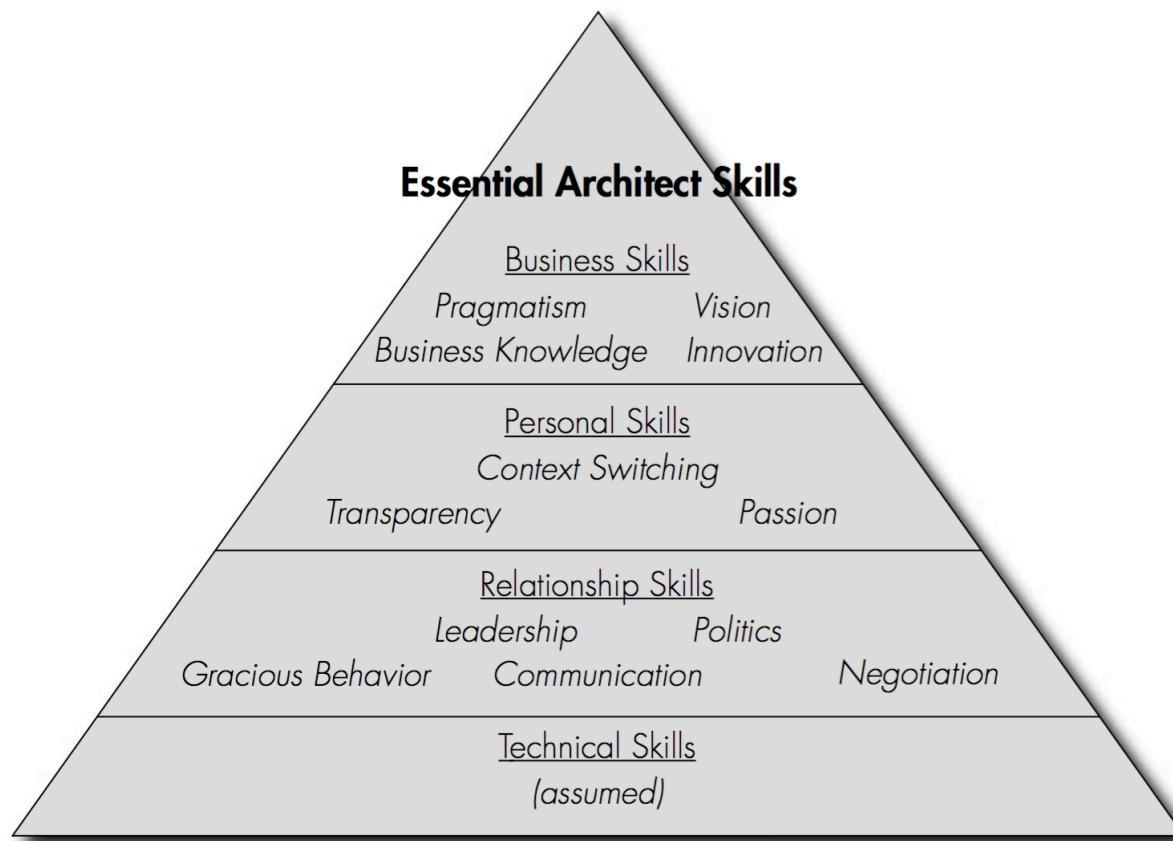
## The Role

The **architect** is responsible for **designing**, **documenting**, and **leading** the **construction** of a system that meets the needs of all its stakeholders.

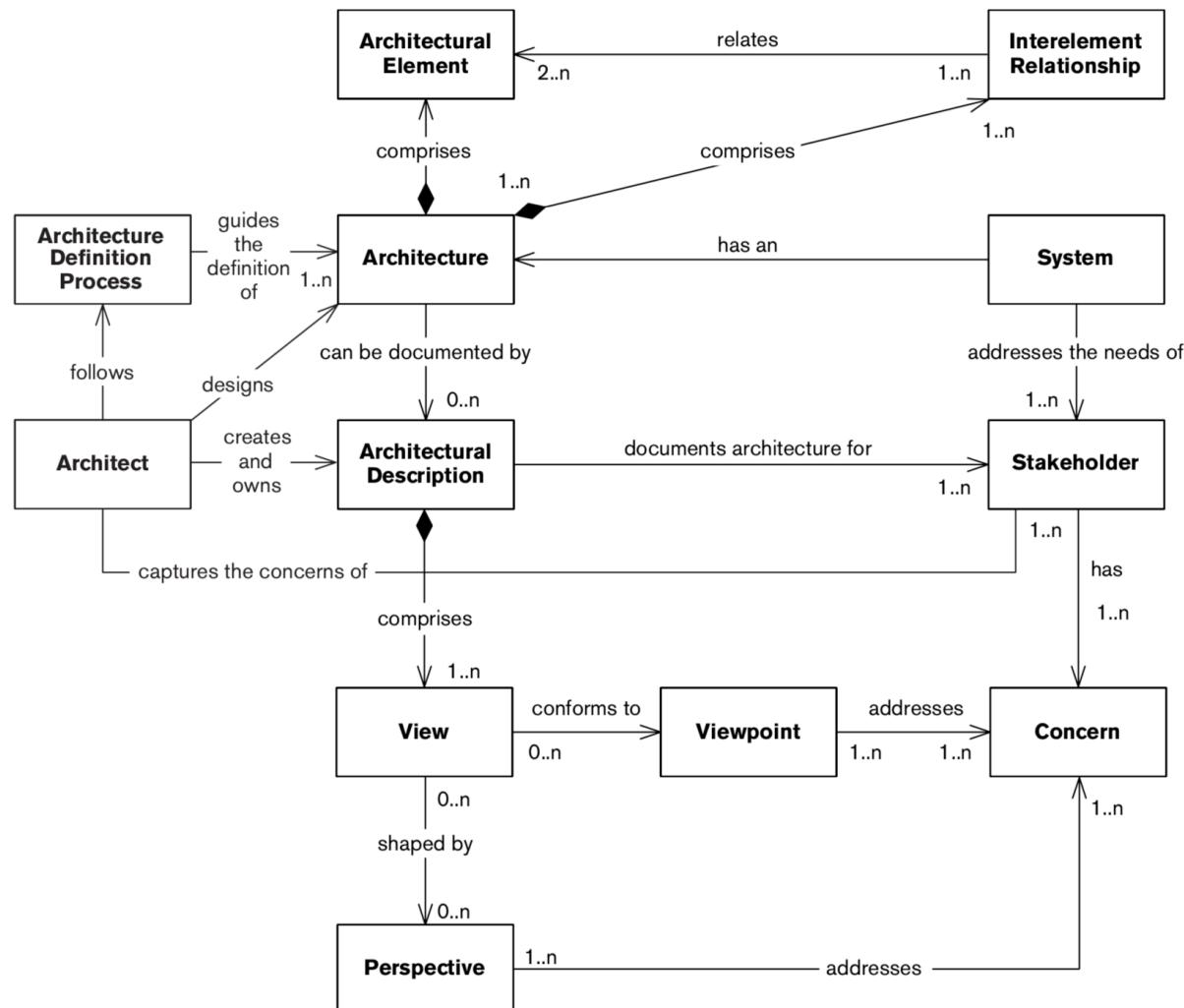


# Software Architect

## Essential Skills



# Software Architecture



## References

- **[BASS]** L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. 2013.
- **[BROWN]** S. Brown, Software Architecture for Developers: A Practical and Pragmatic Guide to 21st Century Software Architecture. 2012.
- **[CERVANTES]** H. Cervantes and R. Kazman, Designing Software Architectures - A Practical Approach. 2016.
- **[CLEMENTS]** P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford, Documenting Software Architectures: Views and Beyond. 2011.
- **[HENDRICKSEN]** D. Hendricksen, 12 Essential Skills for Software Architects. 2012.
- **[ROZANSKI]** N. Rozanski and E. Woods, Software Systems Architecture, 2nd ed. 2011.
- **[STEPHENS]** R. Stephens, Beginning Software Engineering. 2015.
- **[TAYLOR]** R. N. Taylor, N. Medvidovic, and E. M. Dashofy, Software Architecture - Foundations, Theory, and Practice. 2010.