Arquiteturas de Software

Hugo Paredes

Engenharia de Software ?

"Software engineering aims for the systematic, principled design and deployment of applications that fulfil software's original promise applications that retain full plasticity throughout their life cycle and that are as easy to modify in the field as they are on the drawing board. Software engineers have pursued many techniques for achieving this goal: specification languages, high-level programming languages, and object-oriented analysis and design, to name just a few. However, while each contributes to the goal, the sum total still falls short."

(A. Wolf et al.)

A Engenharia de Software é mais um ramo da Engenharia?

"an engineering must have a scientific theory as foundation and this theory can be applied to build products."

(M. S. Gregory)

Diferente no princício;
Abrangência universal;
Diferente na prática.

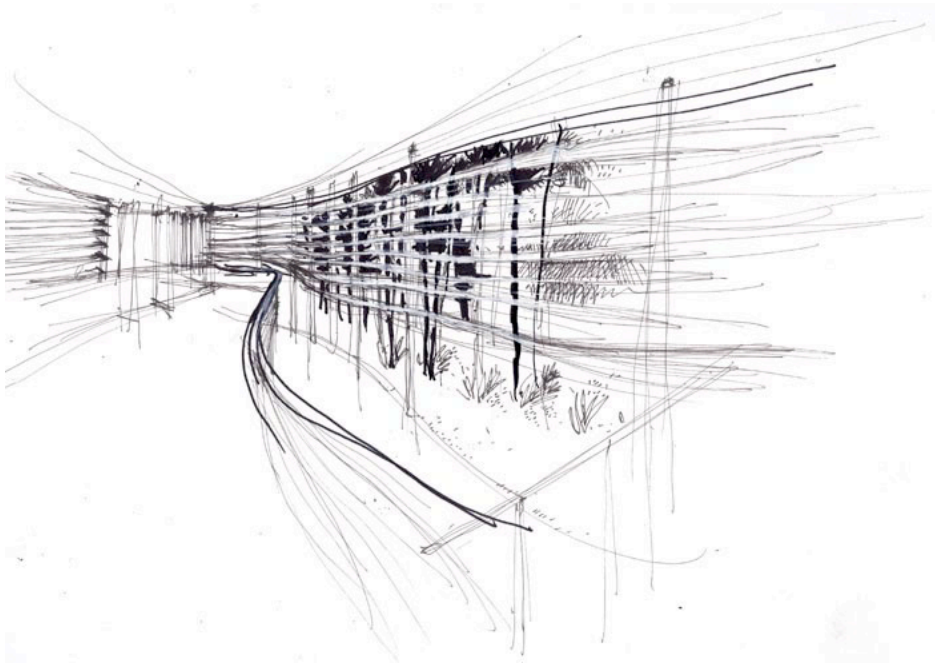
"The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software"

(IEEE definition 610.12)

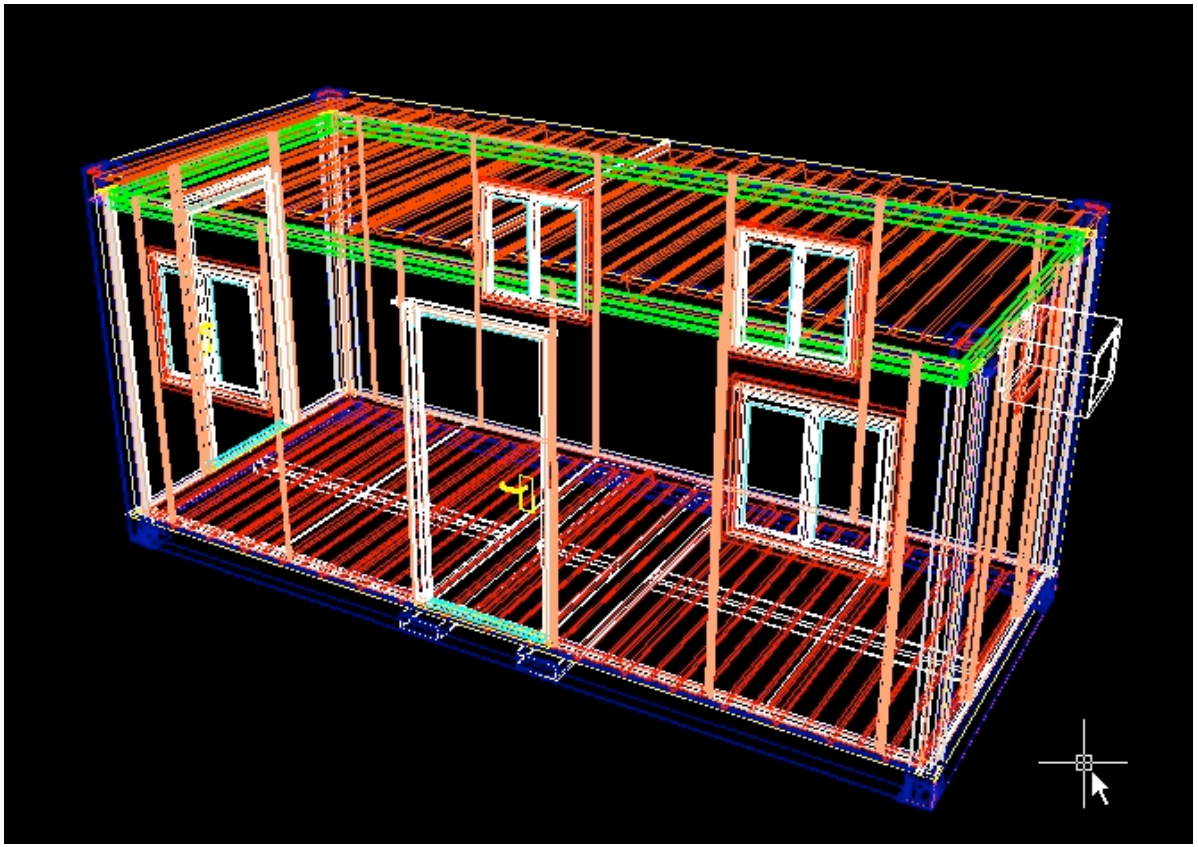
Arquitectura ?

"Arquitectura, do latim architectura, é a arte de dar forma a um meio físico, é a ciência de projectar e construir que caracteriza uma civilização, uma época"

O termo arquitectura está usualmente associado à arte e ao design



No quotidiano o termo Arquitectura é associado à construção de edifícios.



Nesta perspectiva estritamente física uma Arquitectura é caracterizada pela inter-relação do arranjo entre estruturas físicas, na definição de padrões que associam obra e autor.

Porquê uma Arquitectura de Software ?

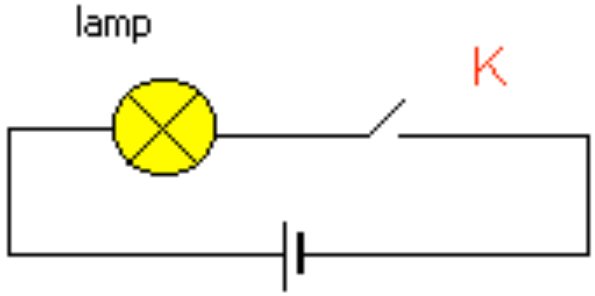
A génese do termo Arquitectura de Software não é somente uma formalidade de nomenclatura.

A utilização do termo Arquitectura associado à disciplina do Software tem uma relação com a ideia intuitiva de Arquitectura usada em outras disciplinas.

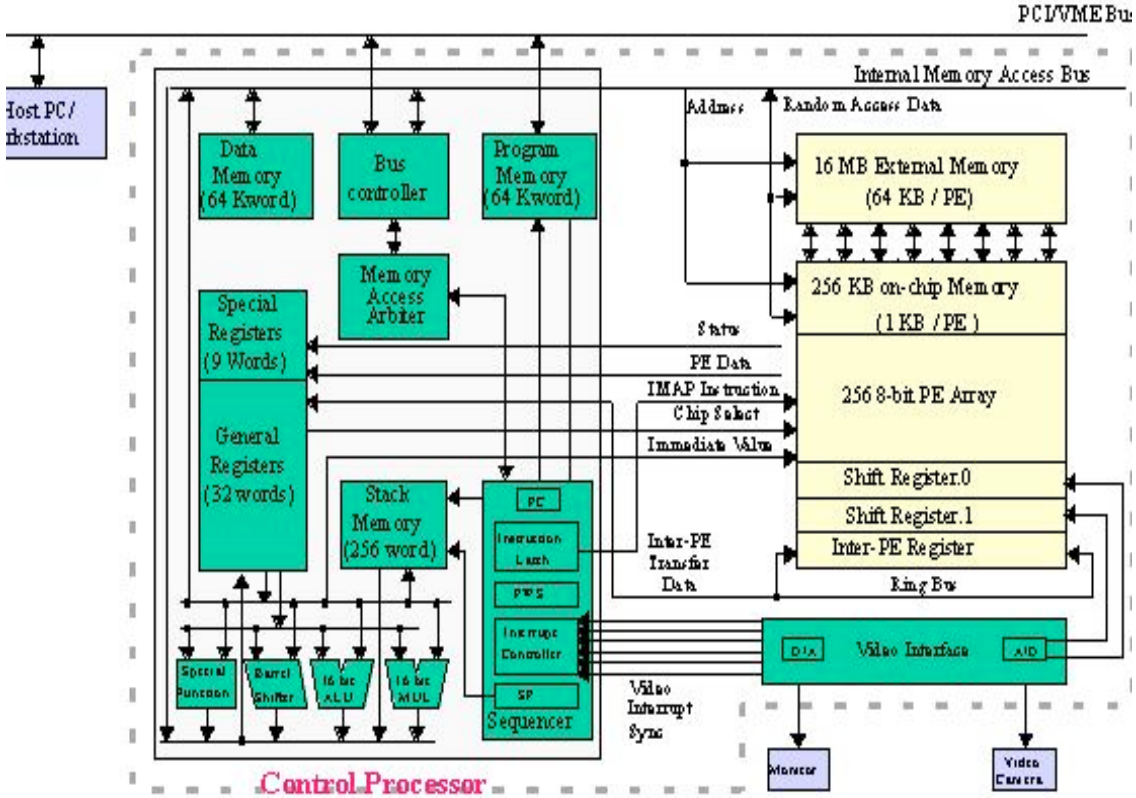
Arquitectura Clássica / Arquitectura de Edifícios

vs

Arquitectura Hardware

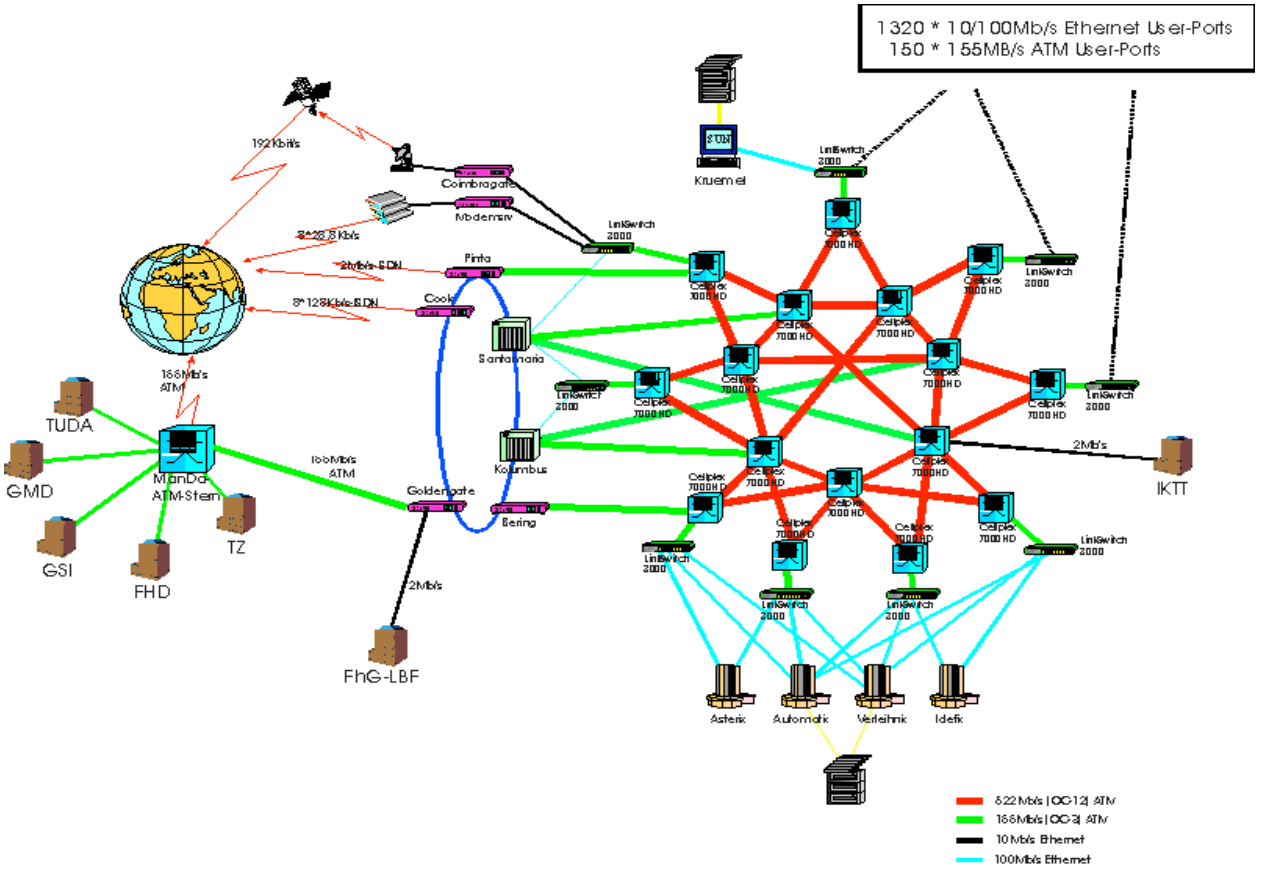


vs



Arquitectura de redes

VS



Qual o problema em comum ?

COMPLEXIDADE do PROBLEMA

Quais as características em comum ?

Componentes

Conectores

Lógica de interligação

Documentação universalmente compreendida

Vistas múltiplas adaptadas aos vários perfis

Quais os objectivos em comum ?

Compreensão do sistema

Reutilização

Evolução

Análise

Manutenção

Separação de desenho e engenharia

Arquitectura de software

vs

Aplicação das analogias à disciplina de Software ?

SIM

Mas tais analogias entre as disciplinas enunciadase sistemas de software não devem ser tomadas literalmente, uma vez que elas se tornam rapidamente inconsistentes, embora ajudem a compreender que a perspectiva é importante e que a estrutura pode ter significados diferentes dependendo da motivação para a sua análise.

Paul Clements

Definições de Arquitectura de Software

"the structure of components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time"

(SEI, 1994)

"a set of architectural elements that have a particular form."

(Perry and Wolf)

a collection of computational components - or simply components - together with a description of the interactions between these components - the connectors.

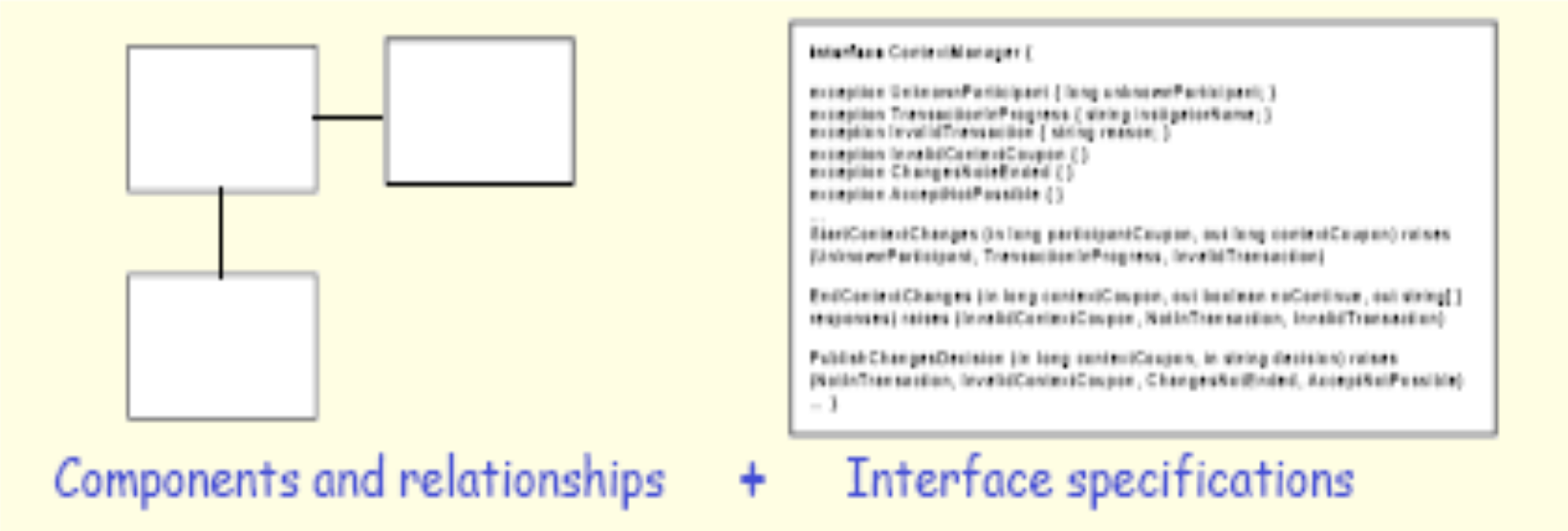
(Garlan and Shaw)

"an architecture should be viewed and described from different perspectives, and should identify its components, their static inter-relationship, their dynamic interactions, properties and characteristics, and constraints on these items"

(Penedo and Roddle)

“Arquitectura” é a estrutura do sistema compreendida de:

Componentes, ou blocos de construção
Propriedades dos componentes visíveis externamente, e
As relações entre eles



Componentes:

Blocos do sistema de alto nível que fornecem:

Modularidade

Separação de preocupações

Colaboração dos Componentes:

Para fornecer serviços (funcionalidades)

Em níveis de serviço (qualidades)

Interface dos componentes:

Fornecem encapsulamento, com pontos de acesso restrito

Especificação dos componentes:

Definem as propriedades visíveis

Fornecem o modo de utilização

Problemas do software

Complexidade

Mas também:

Imprevisível
Pouca qualidade
Crescente dificuldade de alteração
Difícil reutilização
Problemas morais
Perda de mercado

Principais preocupações das arquitecturas de software ?

Decomposição do sistema

como dividir o sistema em peças ?

temos todas as peças necessárias ?

as peças encaixam ?

Integridade do Sistema

Decomposição: Bottom-Up vs Top-Down ?

Decomposição

"dividir para reinar"

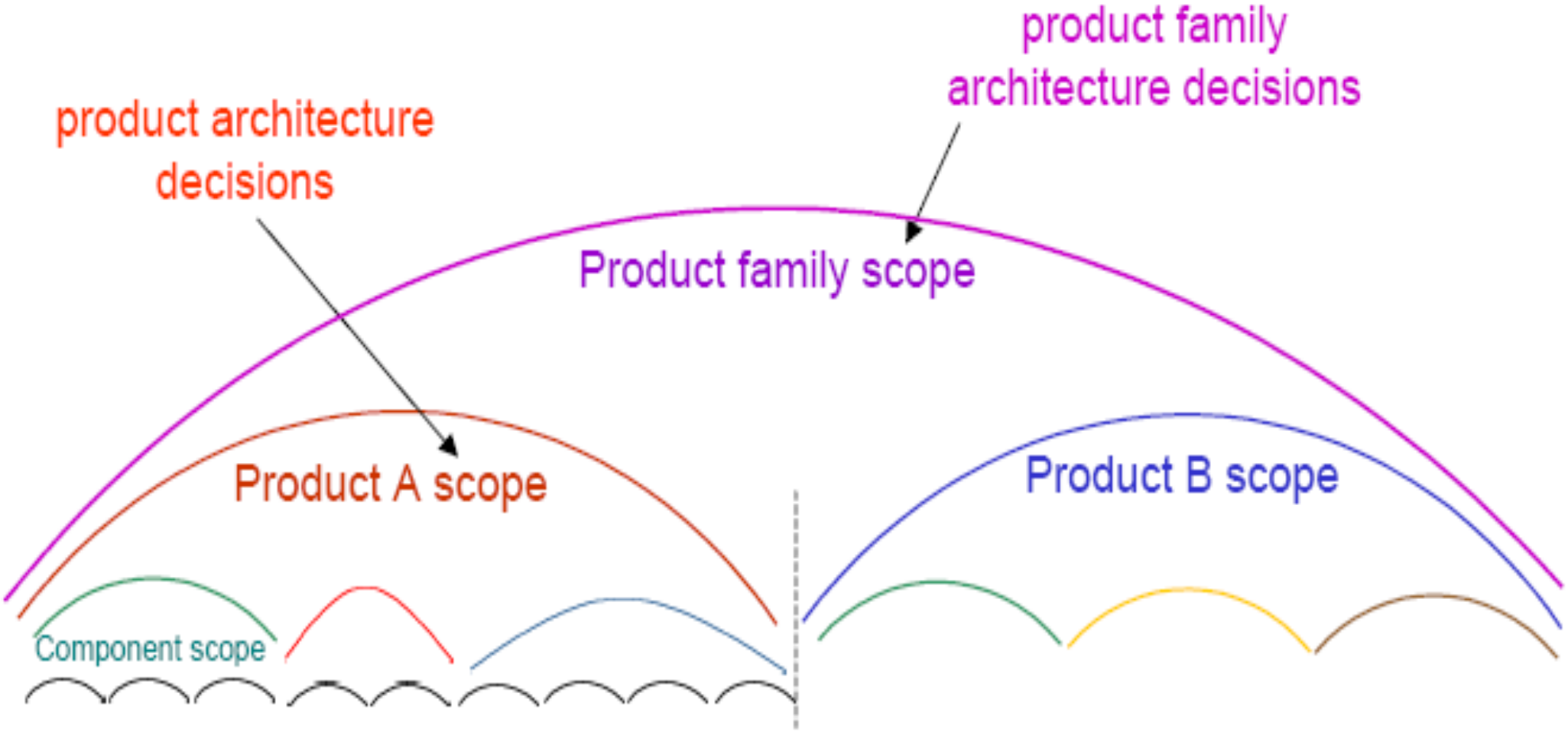
Bottom-Up:

começar por avaliar os pequenos problemas e depois juntar as peças.

Top-Down:

começar por dividir o sistema em grandes blocos, que caracterizam as propriedades mais importantes do sistema, e continuar a divisão até se conseguir um nível de complexidade que possa ser implementado facilmente.

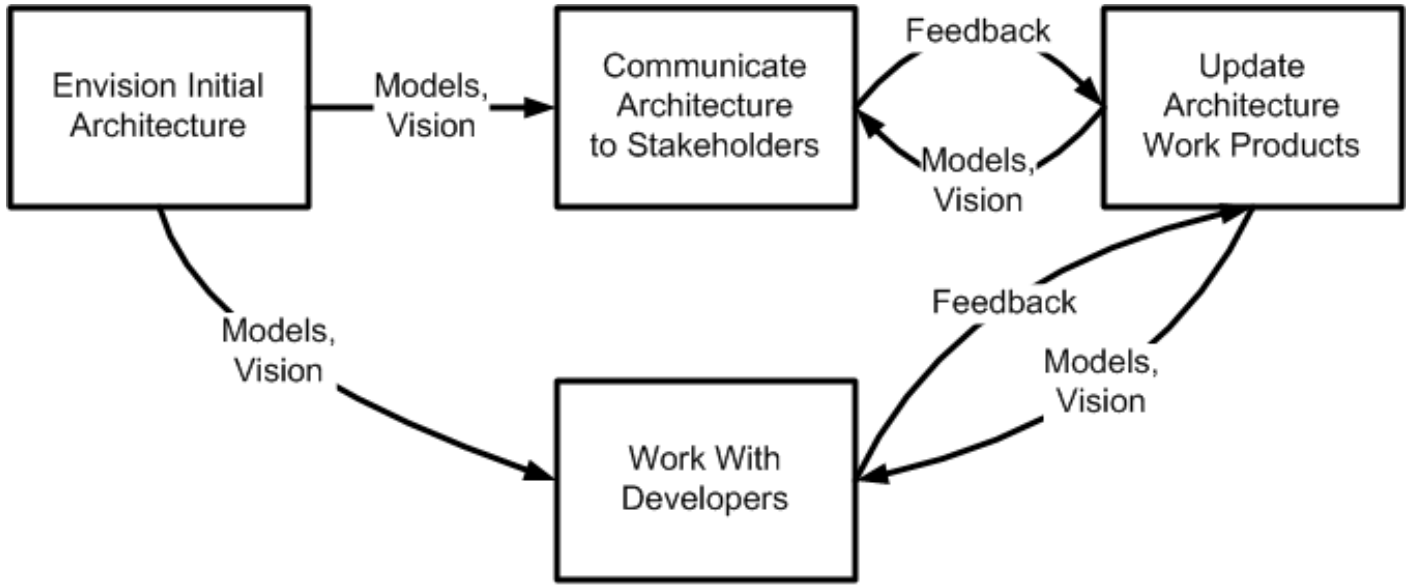
Qual a aproximação que se deve seguir ?

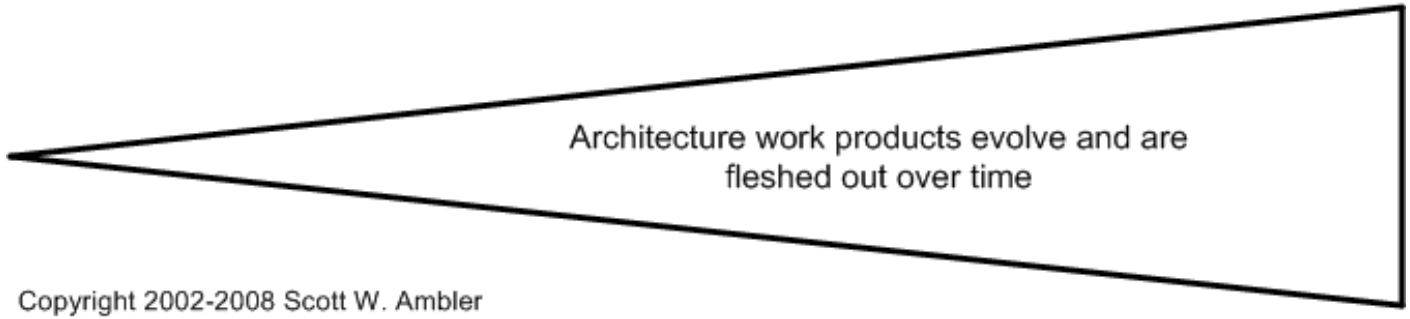


Architecture is the set of decisions that cannot be delegated without compromising overall system objectives.

The Role of an Architect

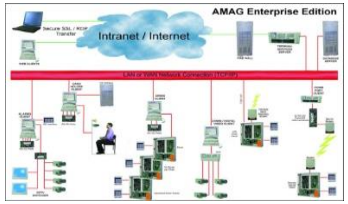
- Central activities:
  - Design
  - Document
  - Assess
  - Recover
  - Maintain



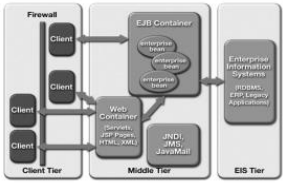


Levels of Architecture*

Enterprise architecture

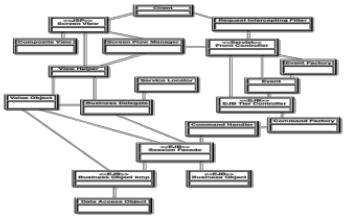


System architecture



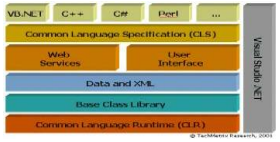
Subsystem

Application architecture



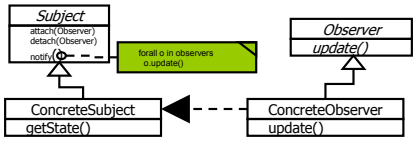
Application

Macro-architecture



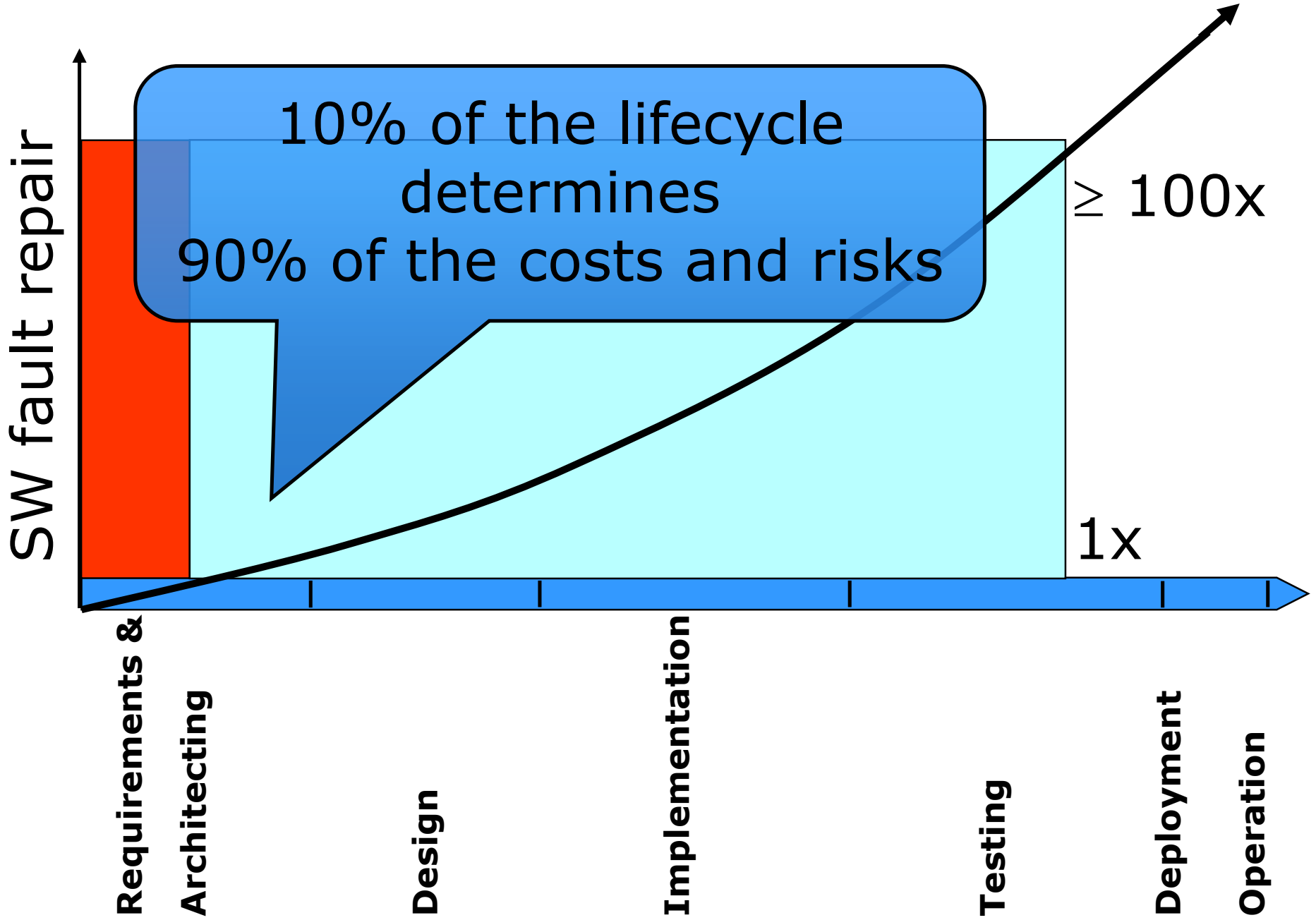
Frameworks

Micro-architecture



Design patterns

The Importance of Architecture



The Importance of Architecture

- Software architecture:
  - provides a **communication** among stakeholders
  - captures **early design** decisions
  - acts as a **transferable abstraction** of a system
  - defines **constraints** on implementation
  - dictates **organizational** structure
  - inhibits or enables a system's **quality attributes**
  - is analyzable and a vehicle for **predicting** system qualities
  - makes it easier to reason about and **manage change**
  - helps in evolutionary **prototyping**
  - enables more accurate **cost** and **schedule** estimates

Wrap-up

- Architecture should be the product of a **single architect or a small group of architects** with an identified leader.
- Architect team should have **functional requirements** for the system and an articulated prioritized list of **quality attributes** that the architecture is expected to satisfy.
- Architecture should be well **documented**, and circulated and **reviewed** by system stakeholders.
- Architecture should be **analyzed** for applicable quantitative measures and formally evaluated for quality attributes before it is too late to make changes to it.
- Architecture should lend itself to **incremental refinement and implementation**.

Representação da Arquitectura

Uma Arquitectura de Software é mais do que um esboço/desenho do sistema

Modelos da Arquitectura:

permitem explorar alternativas e ideias

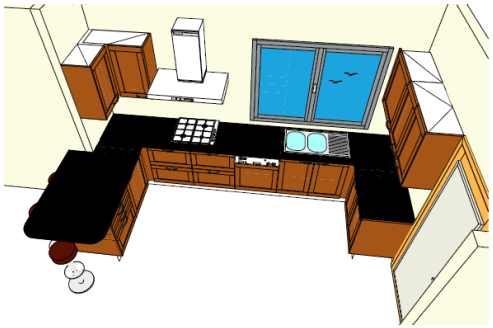
documentam a arquitectura

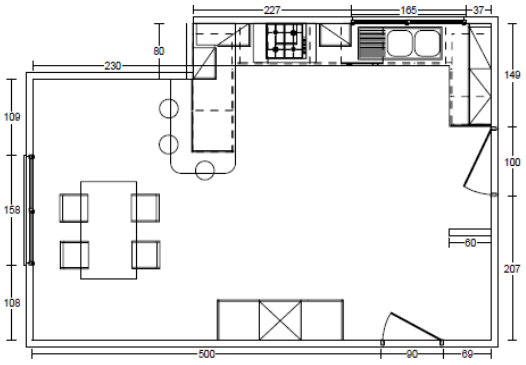
permitem uma visualização do sistema

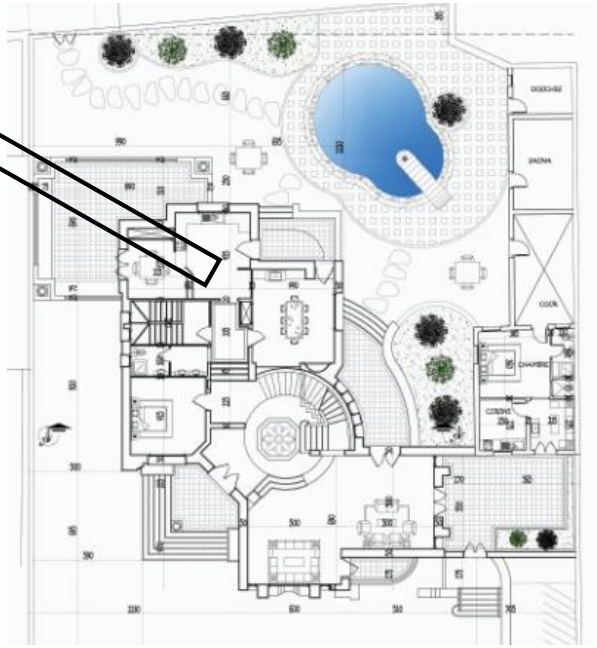
Documentação de Arquiteturas de Software

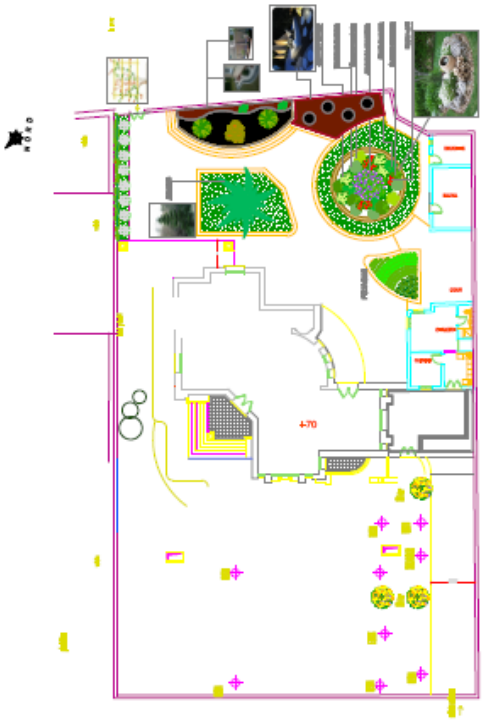
Hugo Paredes

Views

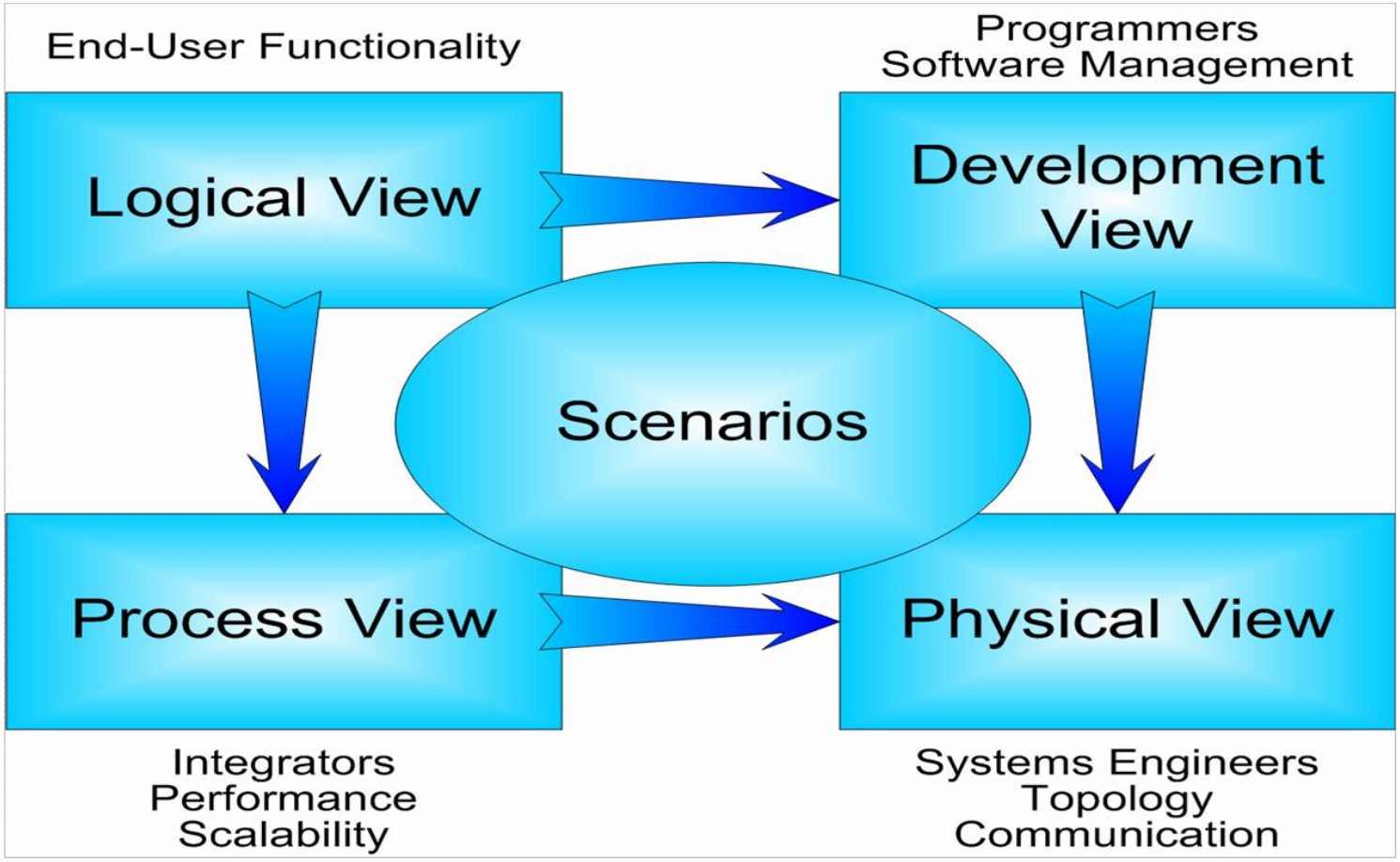








’4 + 1’ View Model*



* Philippe Kruchten

’4 + 1’ View Model*

- **Logical**
  - Focus: Functional requirements of the system.
  - Contents: Class diagrams, Sequence diagrams, Layer diagrams.
- **Development** (implementation)
  - Focus: Static organization of the software in its development environment
  - Contents: Component diagram, Package diagrams.
- **Process**
  - Focus: Runtime behavior of the system, such as the system processes and communication, concurrency, performance and scalability.
  - Contents: Activity diagrams.
- **Physical** (Deployment)
  - Focus: System Engineer’s perspective, looking at the system topology, deployment and communication.
  - Contents: Deployment diagrams.
- **Scenarios**
  - Focus: Use cases for illustrating and validating the architecture.
  - Contents: Use case diagrams.