

# Métodos de Desenvolvimento de Software

Introdução ao Design de Software

Material Baseado nos livros:

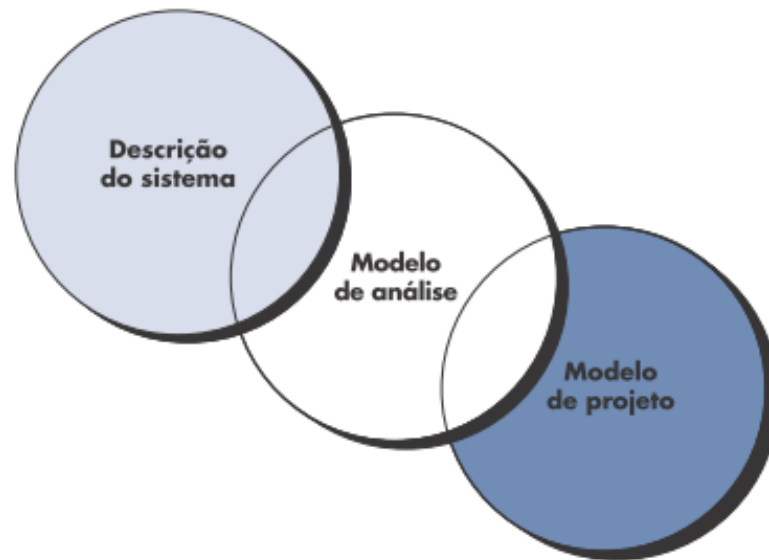
- Swebok - v3
- Introdução a Arquitetura e Design de Software.

*Programar é divertido, mas desenvolver software de qualidade é difícil. Entre as boas ideias, os requisitos ou a "visão" e um produto de software funcional, há muito mais do que programação.*

*Análise e design, visam definir como resolver o problema, o que programar, capturar o design de maneiras que sejam fáceis de comunicar, revisar, implementar e evoluir.*

Philippe Kruchten

# Insumos para o Design



# O que a Disciplina de Design de Software?

- ▶ É o processo de **definição da arquitetura, componentes, interfaces** e outras características de um sistema ou componente e o resultado [desse] processo
- ▶ É onde os **requisitos** de software são analisados a fim de produzir uma descrição da **estrutura interna do software** que servirá de base para sua construção
- ▶ Ou seja, possui o foco em uma **solução conceitual** que atenda aos requisitos identificados, e não a sua implementação, de fato.

 **SWEBOK<sup>®</sup>**  
**V3.0**

*Guide to the Software  
Engineering Body of Knowledge*

 **IEEE**  
IEEE computer society

# O que a Disciplina de Design de Software?

- ▶ A Disciplina de Design de Software, é o elo de ligação (ou tradução) entre a Disciplina de Requisitos e a Disciplina de Construção.
- ▶ Para o Swebok, existem duas atividades principais:
  - ▶ **Projeto de arquitetura de software** (às vezes chamado de projeto de alto nível ou **ARQUITETURA**): desenvolve a estrutura e a organização de nível superior do software e identifica os vários componentes.
  - ▶ **Projeto detalhado do software** (também, denominado **Projeto** ou **DESIGN**, por alguns autores): especifica cada componente com detalhes suficientes para facilitar sua construção.

# O que a Disciplina de Design de Software? (cont.)

- Importante destacar que a ARQUITETURA e o DESIGN são interpretações, visões, leituras críticas de uma implementação



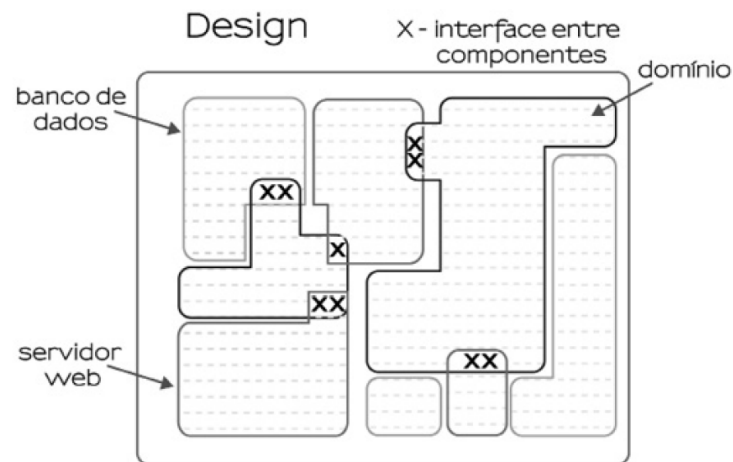
# O que a Disciplina de Design de Software? (cont.)

- ▶ **DESIGN** é uma forma de interpretar a implementação, analisando e compreendendo as **interações entre determinadas partes do sistema**, como possíveis impactos que uma mudança em um ponto causa em outro componente que o acessa direta ou indiretamente.
- ▶ Existem diversas interfaces de comunicação entre diversos componentes

# O que a Disciplina de Design de Software? (cont.)

## ► Exemplo:

- o Apache se comunica com o Jetty através do mod\_jk; uma aplicação Web se comunica com o Jetty através da Servlet API; já dentro do domínio da aplicação, outras interfaces conectam partes de uma implementação, como uma List, um Calendar, um DAO ou uma Connection.



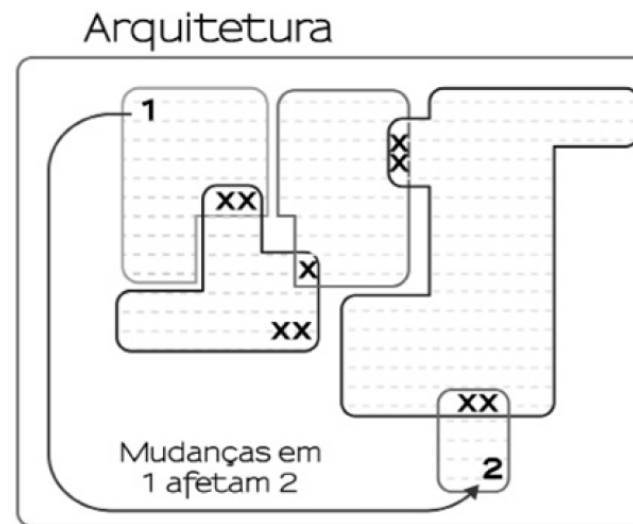
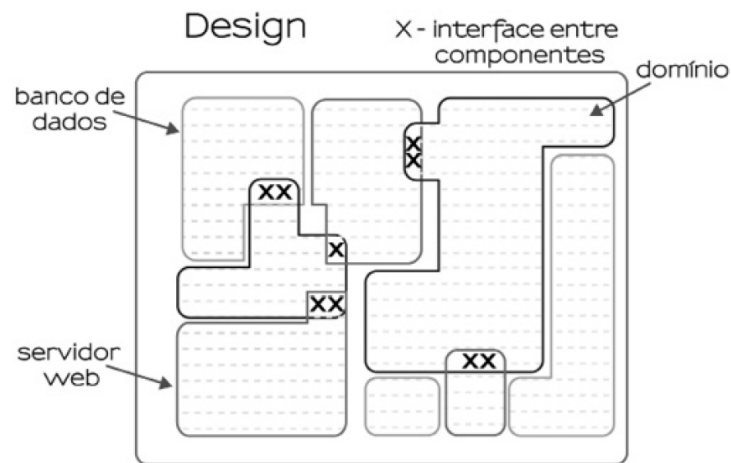


# O que a Disciplina de Design de Software? (cont.)

- ▶ **ARQUITETURA** é outra maneira de ver a implementação, analisando e compreendendo como uma mudança em determinado ponto do sistema afeta o **sistema inteiro**, e de **todas as maneiras possíveis**.
- ▶ Do ponto de vista arquitetural, analisa-se como escolhas de implementação influenciam o sistema inteiro

# O que a Disciplina de Design de Software? (cont.)

- Neste sentido, o **DESIGN** possui uma preocupação mais local, a **ARQUITETURA** uma visão mais global do sistema



# Arquitetura de Software

# Estilos de Arquitetura

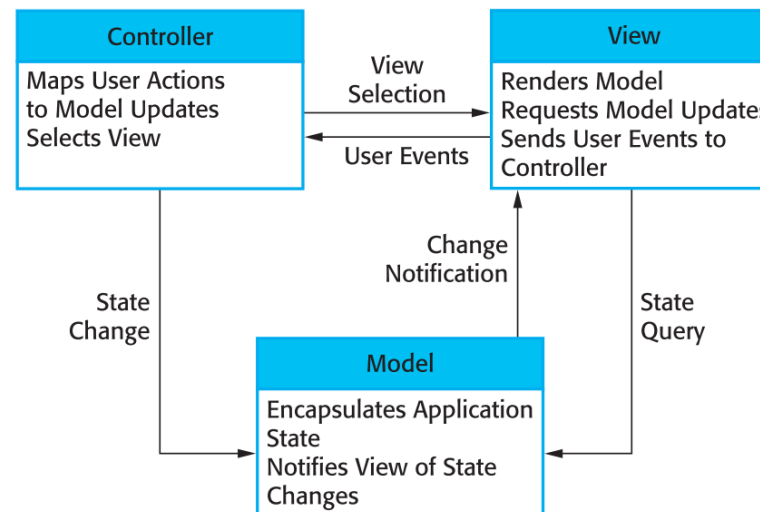
- ▶ Um estilo arquitetural é “uma especialização de elementos e tipos de relação, juntamente com um conjunto de restrições sobre como eles podem ser usados”
- ▶ Um estilo arquitetural pode, portanto, ser visto como fornecendo uma organização de alto nível do software. Vários autores identificaram uma série de estilos arquitetônicos importantes:

# Estilos de Arquitetura

- ▶ Estilos arquiteturais importantes destacados na literatura:
  - ▶ Estruturas gerais (por exemplo, camadas, tubos e filtros, quadro negro)
  - ▶ Sistemas distribuídos (por exemplo, cliente-servidor, três camadas, corretor)
  - ▶ Sistemas interativos (por exemplo, Model-View-Controller, Presentation-Abstraction-Control)
  - ▶ Sistemas adaptáveis (por exemplo, microkernel, reflexão)
  - ▶ Outros (por exemplo, lote, intérpretes, controle de processo, baseado em regras).

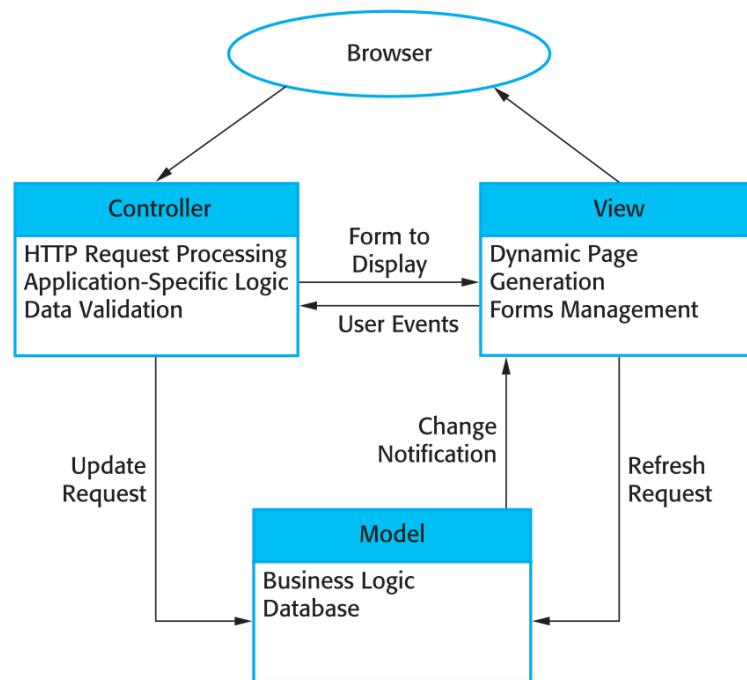
# Arquitetura: MVC

	MVC (Model-View-Controller)
Descrição	Separa apresentação e interação dos dados do sistema. O sistema está estruturado em três componentes lógicos que interagem entre si. O componente <b>Model</b> gerencia os dados do sistema e as operações associadas a esses dados. O componente <b>View</b> define e gerencia como os dados são apresentados ao usuário. O componente <b>Controller</b> gerencia a interação do usuário (por exemplo, pressionamentos de tecla, cliques do mouse, etc.) e passa essas interações para a Visualização e o Modelo.



# Arquitetura: MVC

	MVC (Model-View-Controller)
Exemplo	Arquitetura de um sistema de aplicação baseado na web organizado usando o Padrão MVC.



# Arquitetura: MVC

	MVC (Model-View-Controller)
Quando usar	Quando há várias maneiras de visualizar e interagir com os dados. Também é usado quando os requisitos futuros para interação e apresentação de dados são desconhecidos.
Vantagens	Permite que os dados mudem independentemente de sua representação e vice-versa. Apoia apresentação dos mesmos dados de maneiras diferentes com alterações feitas em uma representação mostrado em todos eles.
Desvantagens	Pode envolver código adicional e complexidade de código quando o modelo de dados e as interações são simples.



# Padrões de Design de Arquitetura

- ▶ De maneira geral, um padrão é “***uma comum solução para um problema comum em um determinado contexto***”.
- ▶ Embora os estilos arquiteturais possam ser vistos como padrões que descrevem a organização de alto nível de software, outros padrões de design podem ser usados para descrever detalhes em um nível inferior (projeto).

# Padrões de Design de Arquitetura

- ▶ Os padrões de projeto de nível incluem:
  - ▶ Padrões de criação
    - ▶ builder, factory, prototype, singleton
  - ▶ Padrões estruturais
    - ▶ adapter, bridge, composite, decorator, façade, flyweight, proxy
  - ▶ Padrões de comportamento
    - ▶ command, interpreter, iterator, mediator, memento, observer, state, strategy, template, visitor

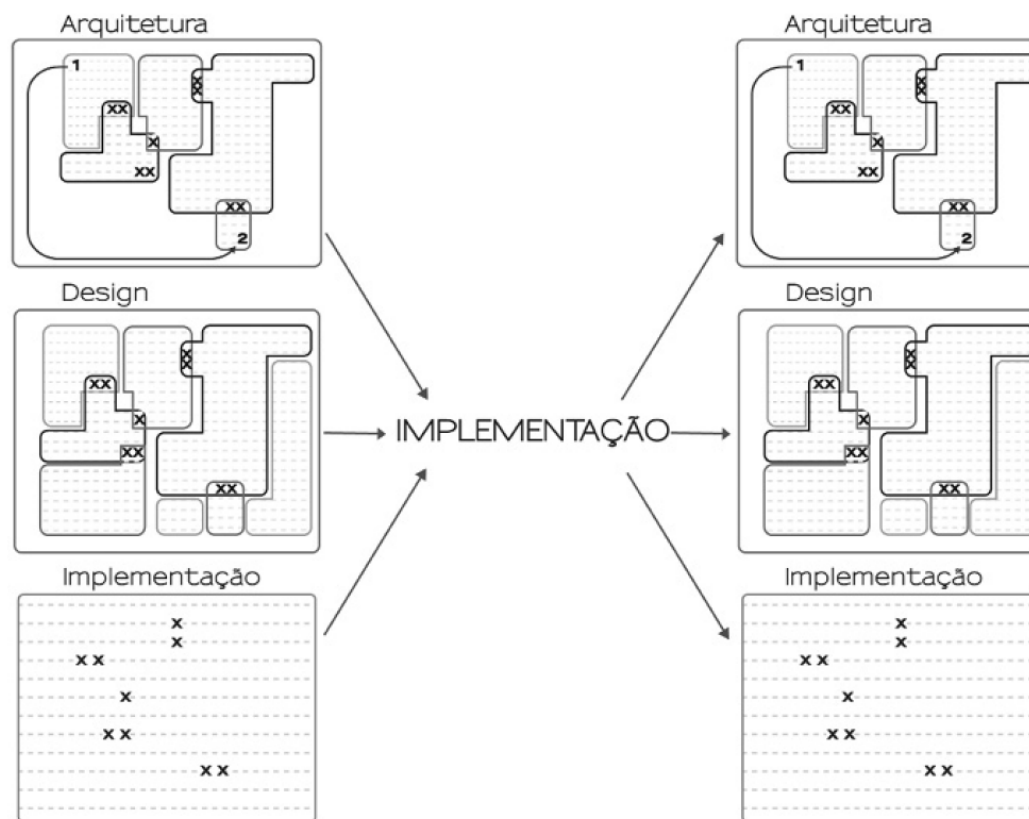
# Decisões de Design Arquitetural

- ▶ O projeto arquitetural é um **processo criativo**.
- ▶ Durante o processo de design, os designers de software precisam tomar várias **decisões fundamentais** que afetam profundamente o **software** e o **processo de desenvolvimento**.
- ▶ É útil pensar no processo de projeto arquitetural, a partir de uma perspectiva de tomada de decisão, e não de uma perspectiva de atividade.

# Design, Arquitetura e Implementação

- ▶ Uma vez que somente implementações são concretas, todo tipo de **mudança implica** em conhecer a **implementação** e, alterá-la causa possíveis mudanças nos aspectos **arquiteturais** e de **design**.
- ▶ Uma **boa implementação, design ou arquitetura** é aquela que permite modificações causando somente um impacto considerado justo a outras partes do sistema, e, ao mesmo tempo, diminuindo as ocorrências de tais situações, **minimizando o custo de desenvolvimento e manutenção**.

# Design, Arquitetura e Implementação (cont.)



# Métodos de Desenvolvimento de Software

Introdução ao Design de Software



Material Baseado nos livros:

- Swebok - v3
- Introdução a Arquitetura e Design de Software.