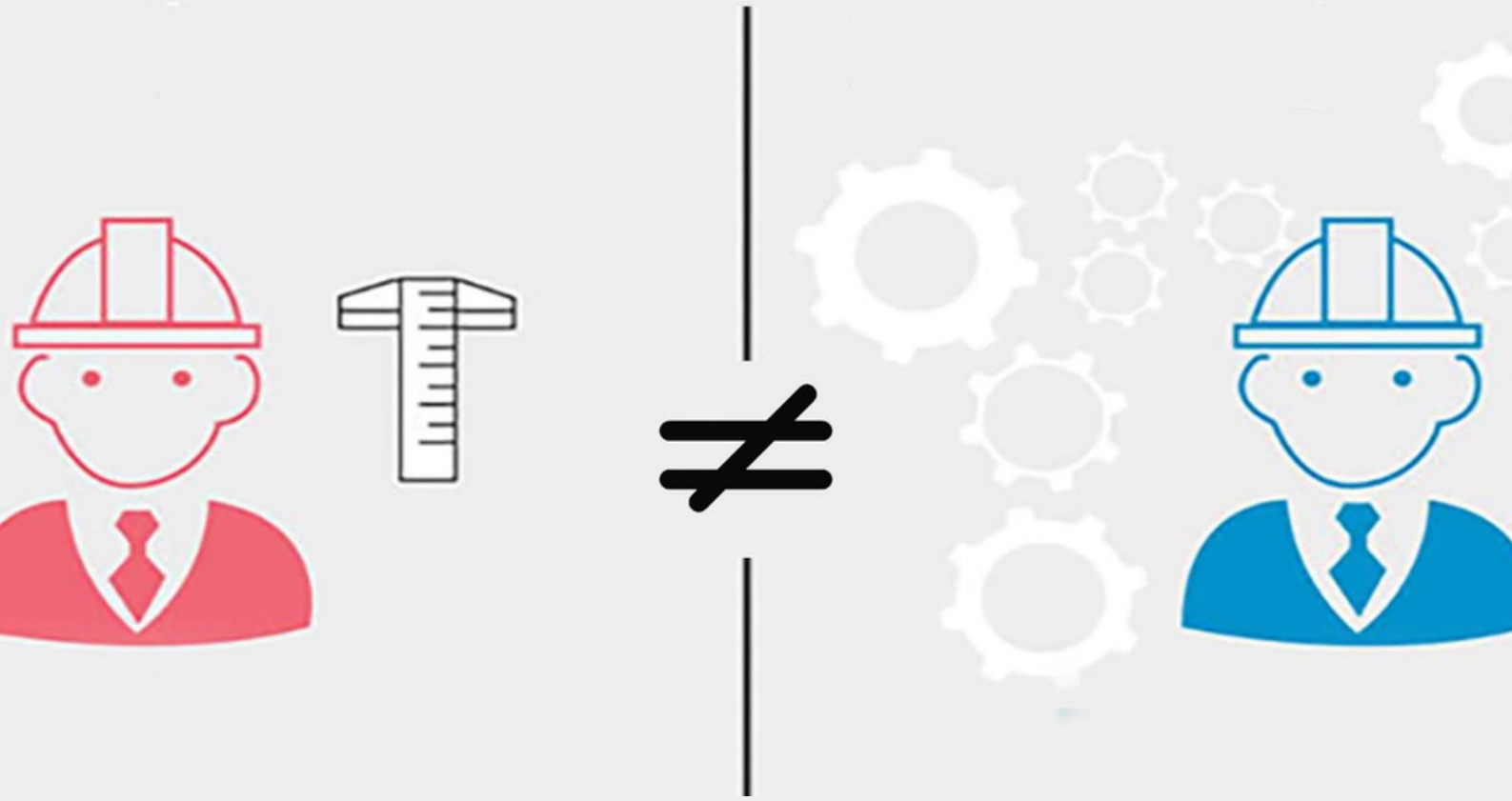
Arquitetura de Software
VS
Design de Código:



Design vs Arquitetura

...

Tem diferença ou é a mesma coisa?

Definições chatas e formais

Design de código

Object-oriented design is the discipline of defining the objects and their interactions to solve a problem that was identified and documented during object-oriented analysis.

Traduzindo:

Design orientado a objetos é a disciplina de definir os objetos e suas interações para resolver um problema que foi identificado e documentado durante a análise orientada a objetos.

Arquitetura de software

A arquitetura de software de um sistema consiste na definição dos componentes de software, suas propriedades externas, e seus relacionamentos com outros softwares.

Como já era de se esperar: Não entendi nada

Analogia com uma casa

Um arquiteto faz a planta da casa.

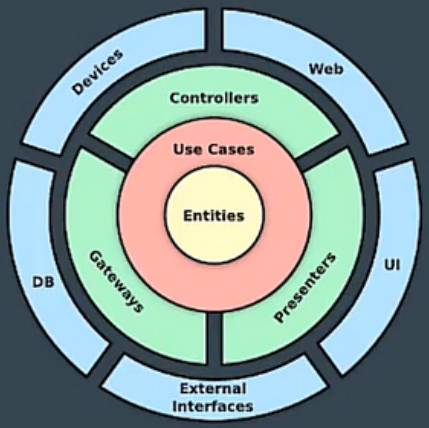
Um designer de interiores define como os móveis vão estar.

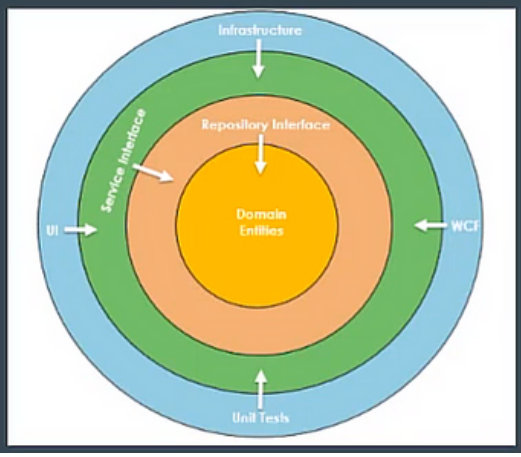
Design

Arquitetura

Agora trazendo pra TI

Arquitetura de software

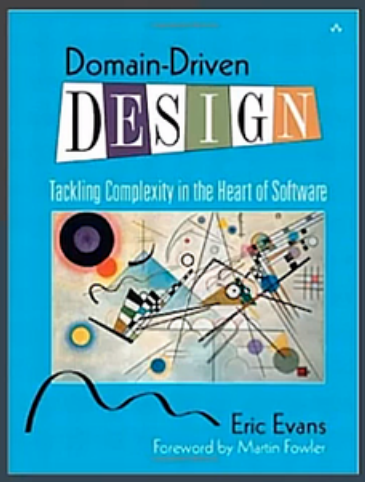




Design de código

SOLID, DRY & KISS

Princípios de POO



Resumindo

Arquitetura: Visão de mais alto nível. Separação de camadas, pastas da aplicação.

Design: Visão de mais baixo nível. Como escrever cada classe. Quais padrões aplicar

Arquitetura de Software:

A arquitetura de software refere-se à estrutura fundamental de um sistema de software. Ela lida com decisões de alto nível sobre como o sistema é organizado e como seus componentes interagem entre si. A arquitetura de software é uma visão de alto nível do sistema, que define seus componentes principais e como eles se relacionam para atender aos requisitos do usuário e às metas do negócio.

Principais Aspectos:

1. **Componentes:** Define os principais componentes do sistema, como módulos, classes ou serviços.
2. **Relacionamentos:** Descreve como esses componentes interagem e se comunicam uns com os outros.
3. **Padrões:** Estabelece padrões e diretrizes para o design e desenvolvimento do sistema.
4. **Escalabilidade:** Considera como o sistema pode ser escalado para lidar com um aumento na carga.
5. **Manutenibilidade:** Planeja para facilitar futuras modificações e atualizações do sistema.

Design de Código:

O design de código refere-se à estrutura interna dos componentes de software individuais. Ele está mais focado nos detalhes de implementação, como classes, métodos, variáveis e algoritmos. O design de código preocupa-se com a legibilidade, eficiência, reusabilidade e facilidade de manutenção do código-fonte.

Principais Aspectos:

1. **Clareza e Legibilidade:** Código deve ser fácil de ler e entender para colaboração eficaz e manutenção.
2. **Eficiência:** O código deve ser otimizado para desempenho, uso de recursos e velocidade de execução.
3. **Reusabilidade:** Componentes de código devem ser projetados para serem reutilizáveis em diferentes partes do sistema ou em projetos futuros.
4. **Manutenibilidade:** Deve ser fácil fazer alterações no código sem introduzir erros.
5. **Simplicidade:** Deve seguir o princípio KISS (Keep It Simple, Stupid) para evitar complexidade desnecessária.

Relação entre os Dois:

- **Arquitetura e Design:** A arquitetura de software influencia o design de código, pois define a estrutura global do sistema. As decisões de arquitetura afetam a forma como o código é organizado e como os componentes individuais são projetados.

- **Colaboração:** Arquitetos de software e desenvolvedores trabalham juntos para garantir que a arquitetura seja implementada corretamente no nível do design de código.