

Plano de ensino & Revisão Arch

Relembrando arquitetura e as
novidades para o semestre

Prof. Sandro Ramos



Sejam bem-vindos!

Fico muito feliz em velos novamente!

Que assim como foi nossa última experiência, que possamos colaborativamente construir essa jornada de conhecimento!



Overview

- Plano de ensino
- Arquitetura: Definição
- Arquitetura: Benefícios
- Arquiteto de Software
- Design
- Arquitetura x Design

Plano de ensino



Avaliação

Conteúdo programático

Ementa

Metodologia

Competências

Recursos, Projeto integrador...

Avaliação

Primeiro bimestre

- 2 trabalhos - **Peso 3,0**
 - 2 Formulários - 1,5 cada
- 1 Prova - **Peso 7,0**

Segundo bimestre

- 2 trabalhos - **Peso 7,0**
 - 1 Formulário - 2,0 cada
 - 1 Trabalho - 5,0

Prova multi - 3,0

Arquitetura

Definição

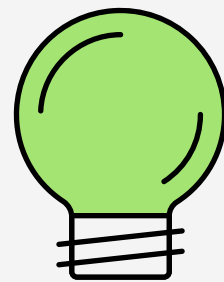
Segundo Mary Shaw "Arquitetura de software é a estrutura de estruturas de um sistema, composta por elementos de software, suas propriedades externamente visíveis e os relacionamentos entre eles."



Em resumo, são os princípios de alto nível que orientam o projeto e sua evolução ao longo do tempo.

Arquitetura

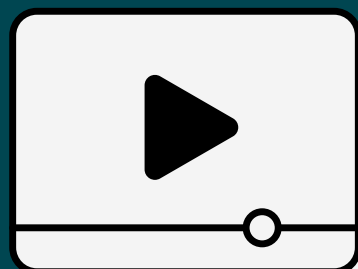
Definição



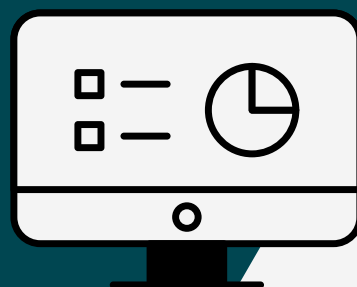
Relacionamentos entre os elementos: São as conexões e dependências entre os diferentes elementos de software. Estes relacionamentos podem incluir chamadas de função, trocas de mensagens, dependências de dados, e interações entre módulos.



Propriedades externamente visíveis: Referem-se às interfaces públicas e ao comportamento observável dos elementos de software. Definem como os elementos interagem e se comunicam uns com os outros. Ex: métodos, funções, APIs.



Elementos de software: São os componentes fundamentais que constituem o sistema. Estes podem ser módulos, classes, componentes ou serviços e etc.



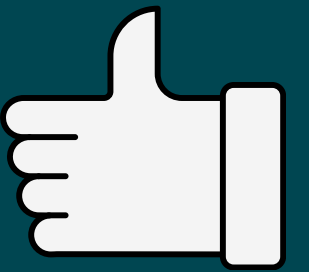
Estrutura de estruturas: Não é uma única estrutura, mas uma coleção de estruturas necessárias para compor o software, conforme suas diferentes preocupações e aspectos. Ex: Estrutura de organização de módulos, de execução, de desenvolvimento e a estrutura física, entre outras.

Arquitetura

Benefícios

O maior benefício de uma boa arquitetura é sem dúvida o baixo acoplamento, que permite nos adaptarmos rapidamente às mudanças, por vezes sem sentido, do negócio.

- Compreensão do negócio
- Baixo acoplamento
- Alta coesão
- Reutilização
- Separação clara de responsabilidades
- Escalabilidade e desempenho
- Manutenibilidade



Para Len Bass, além dos já citados:

- Atendimento a requisitos de qualidade(RNF)
- Gerenciamento de complexidade
- Facilidade de integração
- Redução de riscos
- Suporte a tomada de decisões

Arquiteto de software

É o responsável por toda a definição da estrutura, tecnologias, padrões de comunicação e coordenação da criação de um software.

Garante que o software atenda aos requisitos de qualidade e atua de maneira consultiva durante o desenvolvimento.



Arquiteto de software

Característica



É mais próximo do negócio, traduzindo as demandas/prioridades da empresa para a "fala técnica" que um time de tecnologia compreenda.

No processo de desenvolvimento se diferenciam por darem um foco muito grande na etapa de solução do problema, enquanto o resto do time está focado em implementação.

Na prática, é um Sênior que já domina front, back, cloud e tem uma expertise em várias tecnologias, para escolher a melhor ferramenta, dado um determinado problema.

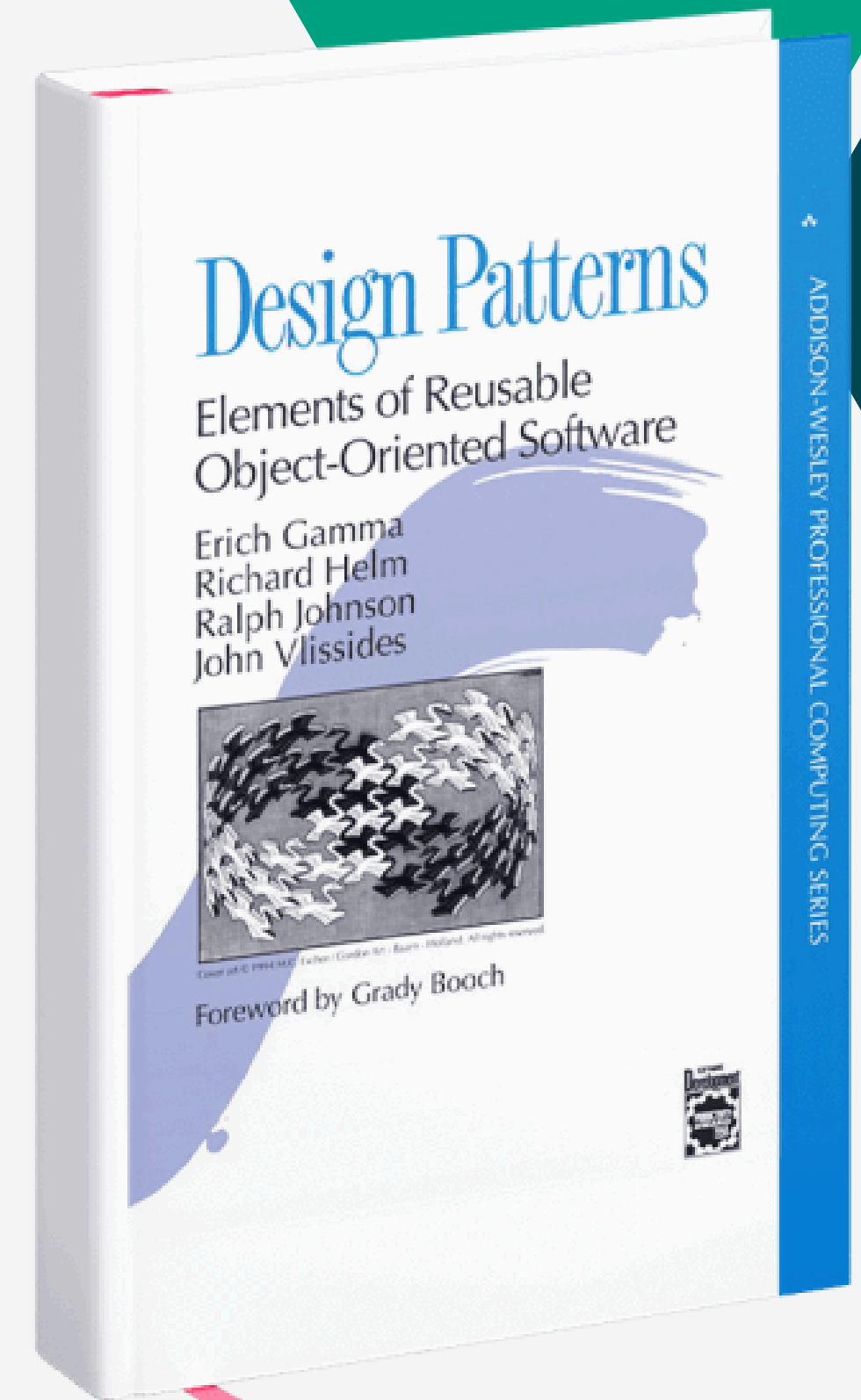
Design de Software

Segundo Erich Gamma, design de software é o processo de definir a estrutura e o comportamento de um sistema de software mediante componentes reutilizáveis e bem definidos, conhecidos como Design Patterns.



Pontos cruciais de sua visão:

- Padrões de Design (Design Patterns)
- Reutilização e flexibilidade
- Comunicação eficaz entre devs
- Foco em soluções POO



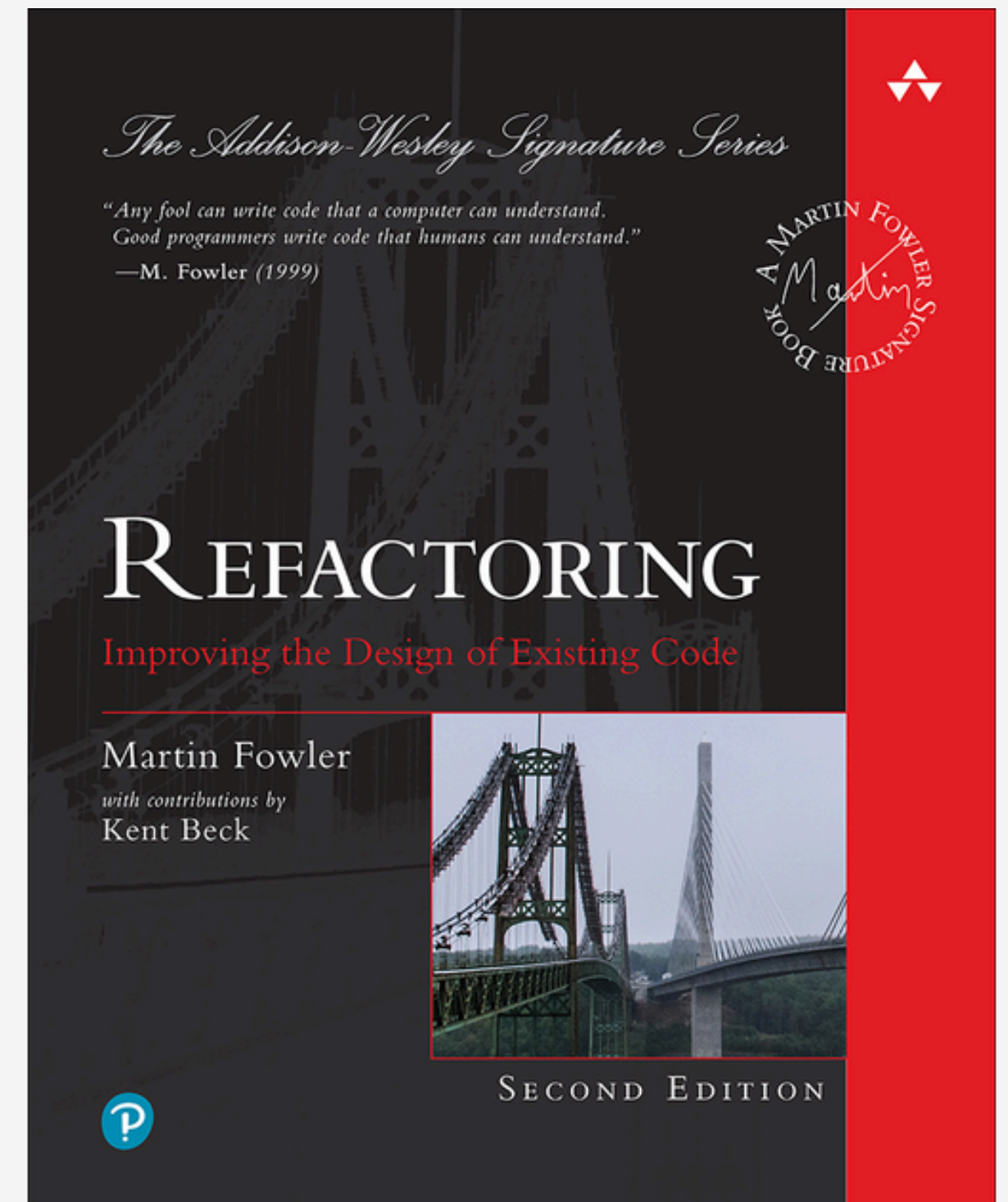
Já para Martin Fowler design de software é definição de uma estrutura que permita que o software seja compreensível, flexível e sustentável.

Fowler enfatiza que um bom design favorece mudanças e melhorias contínuas ao longo do tempo.



Pontos cruciais de sua visão:

- Refatoração
- Padrões de Design
- Simplicidade e clareza
- Evolução contínua

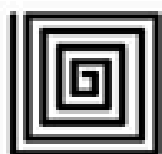




Design

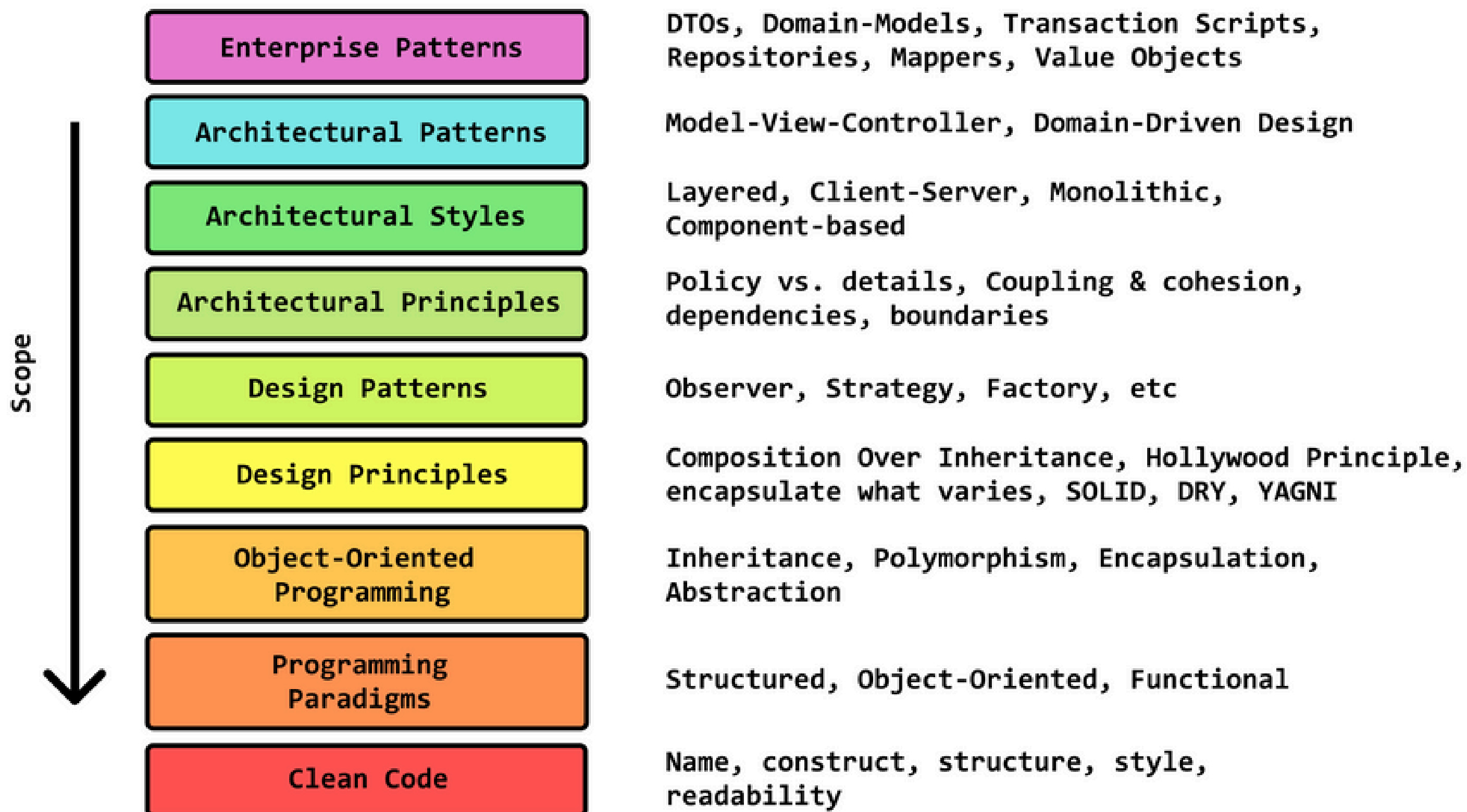
Em resumo podemos concluir que desing de software compreende a definição da estrutura/comportamento dos componentes de um sistema, ou seja, de seus elementos de baixo nível.

Buscando criá-los de maneira flexível, passíveis de refatoração, simples e claros, logo os padrões de design podem nos auxiliar a chegar nesse objetivo.



The Software Design & Architecture Stack

Khalil Stemmler
@stemmlerjs



Arquitetura X Design

E qual a diferença?



Design é mais granular, está nas escolhas feitas durante o processo de desenvolvimento e impacta partes específicas. Foco na implementação concreta e nas práticas de codificação que resultam em um código limpo, sustentável e eficiente.

Arquitetura é mais abrangente, nesse escopo teremos decisões estruturais que impactam diretamente como o sistema será desenvolvido e como evoluirá. Os aspectos definidos aqui são mais complexos de serem alterados e impactam diretamente nos atributos de qualidade do software.

Referências

Shaw, M., & Garlan, D. (1996). Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall.

Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice (3rd ed.). Addison-Wesley Professional.

Fowler, M. (1999). Refactoring: Improving the Design of Existing Code. Addison-Wesley.

Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

<https://blog.xpeducacao.com.br/o-que-e-arquitetura-de-software/>

<https://ehgomes.com.br/software/arquitetura-de-software-fundamentos-principios-e-melhores-praticas/>

<https://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf>

<https://blog.xpeducacao.com.br/arquiteto-de-software/>

<https://querobolsa.com.br/carreiras-e-profissoes/arquiteto-de-software>

<https://www.lucidchart.com/blog/pt/como-implementar-o-design-de-software>

<https://www.freecodecamp.org/portuguese/news/como-aprender-design-e-arquitetura-de-software-um-guia/>