

3 Modelos no desenho e implementação

vistas de arquitetura

- arquitetura
- conjunto de decisões importantes sobre a organização de um software
 - procura saber quais os elementos estruturais e as interfaces
 - decisões acerca de colaborações inter-elementos
 - entende como funciona o sistema e qual o seu comportamento

objetivo é definida para satisfazer os requisitos não funcionais e as restrições de operações determinantes

arquitetura lógica | organização geral dos blocos de software (packages), independentemente das tecnologias de implementação

arquitetura de instalação | visão dos equipamentos e configuração concreta de produção

arquitetura de componentes | peças construídas com uma tecnologia concreta
é normal dividir sistemas complexos em subsistemas + geríveis

componente | peça substituível e reusável de um sistema maior, cujos detalhes de implementação são abstraídos

arquitetura

- estática → não afeta o correr do sistema
- dinâmica → muda a forma de correr do sistema

arquitetura por camadas

- divisão modular de blocos de software em camadas / níveis
- as camadas são sobrepostas, pedindo "as de cima" serviços às de baixo
- cada camada tem uma especialização

papel do arquiteto | explorar a resolução entre a arquitetura do sistema e os atributos de qualidade

desenha (cria e envolve) e analisa se o desenho suporta as necessidades funcionais

acoplamento mede a força/intensidade da dependência de uma classe das outras
coesão mede a força/intensidade do relacionamento dos elementos de uma classe entre si

Princípios do design orientado a objetos (OOD)

↓ acoplamento ↑ coesão

classes e
design de
métodos

GRASP (Generic Responsibility Assignment Software Principles)

diretrizes para atribuir responsabilidades a classes e objetos em programações orientada a objetos

baixo acoplamento

menor dependência entre classes

alta coesão

elemento da classe fortemente relacionado

especialista na informação

princípio para determinar onde delegar responsabilidades (por ex métodos)
criador

definir qual a classe responsável por criar objetos

controlador

atribui a responsabilidade de manipular eventos do sistema, sendo
o objeto controlador da interface não-usuário, responsável por
receber ou manipular um evento do sistema