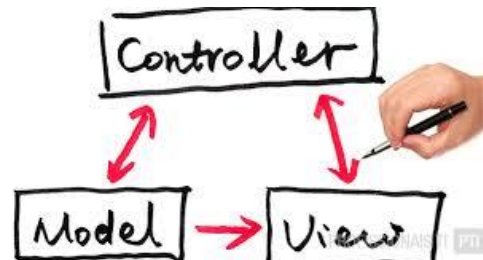




Unidade 23

Arquitetura de Software

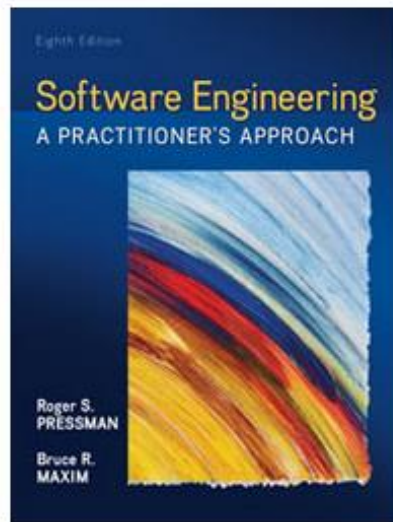


Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPU SP

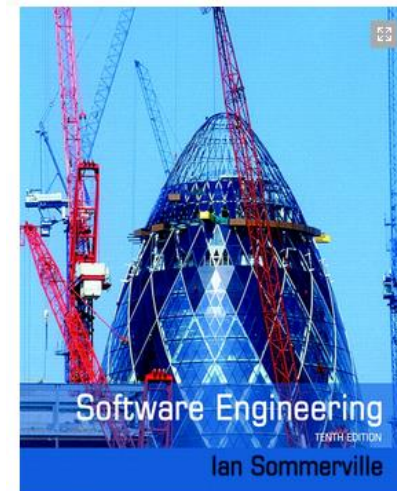


Bibliografia

- **Software Engineering – A Practitioner's Approach – Roger S. Pressman – Eight Edition – 2014**
- **Software Engineering – Ian Sommerville – 10th edition - 2015**
- Engenharia de Software – Uma abordagem profissional – Roger Pressman - McGraw Hill, Sétima Edição - 2011
- Engenharia de Software – Ian Sommerville – Nona Edição – Addison Wesley, 2007



Software Engineering: A
Practitioner's Approach, 8/e





Introdução

- ⊕ A Arquitetura de Software é composta por alguns princípios;
- ⊕ Embora haja ferramentas que auxiliem na concepção da arquitetura, sem o emprego desses princípios não se desenhará com qualidade a arquitetura do software;





O que se pode pensar dessa imagem?





O que se pode pensar dessa imagem?



- ⊕ Quarto desorganizado!
- ⊕ Quarto desarrumado!
- ⊕ Muitos objetos espalhados pelo quarto!



Algumas questões



- ⊕ Você acharia alguma roupa no quarto ?
- ⊕ Você acharia facilmente algum objeto no quarto?
- ⊕ É fácil mudar o layout do quarto?
- ⊕ Você se sentiria bem nesse quarto?



Uma outra visão



- ⊕ Quarto organizado;
- ⊕ Objetos em seus lugares corretos;
- ⊕ Ambiente confortável.



Qual a diferença dos dois ambientes?



Caos x Ambiente Perfeito !





Ambiente Organizado



- ⊕ Você acharia alguma roupa no quarto ?
- ⊕ Você acharia facilmente algum objeto no quarto?
- ⊕ É fácil mudar o layout do quarto?
- ⊕ Você se sentiria bem nesse quarto?





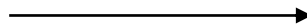
Outra questão

- ⊕ Se num ambiente físico tão simples (um quarto) pode-se facilmente se instalar um caos, imagine como seria num ambiente de desenvolvimento de software?





No desenvolvimento de software, essa situação pode ocorrer (caos), porém as consequências são muito mais drásticas !!!





Será que é fácil efetuar manutenção em softwares desorganizados?





Que problemas usualmente se encontra ao se fazer manutenção em um sistema desorganizado?





Software desorganizado

Observações frequentes



- ⊕ Dificuldade para se localizar código;
- ⊕ Partes internas difíceis de serem alteradas;
- ⊕ Alterando-se alguma parte, quebram-se outras;
- ⊕ Desenvolvedor inseguro e com estresse;
- ⊕ Atribui-se falta de qualidade ao sistema de software;
- ⊕ Atribui-se incompetência do desenvolvedor;
- ⊕ Cliente passa a ter desconfiança da equipe de software;
- ⊕ Credibilidade do desenvolvedor é afetada;
- ⊕ Custos podem afetar o projeto;
- ⊕ Prazos podem afetar o projeto;
- ⊕ Mudanças de negócio não absorvidas pelo software;
- ⊕ Há probabilidade de se redesenvolver o software.





Como desenvolver softwares organizados?





Diretriz Básica



- ⊕ “A única constante de um software é a sua permanente mudança”; (Martin Folwer)
- ⊕ O software deve ser organizado, estruturado, para absorver mudanças;
- ⊕ Software não deve ser desenhado para permanecer estático ao longo do tempo;
- ⊕ Desenvolver experiente emprega essa diretriz no desenvolvimento de software;
- ⊕ Software foi feito para ser alterado;
- ⊕ Software deve fornecer valor agregado ao cliente para o negócio atual e deve se adequar às futuras mudanças tecnológicas e corporativas.





Software para absorver mudanças

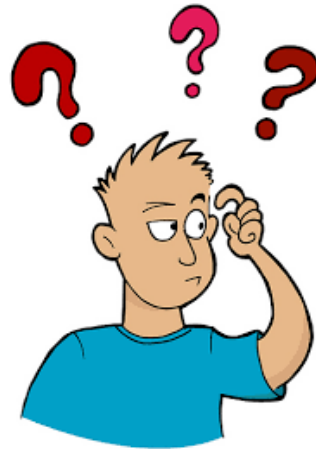


- ⊕ Mudança de paradigma: Focar em Qualidade de Software;
- ⊕ A Engenharia de Software trata a Qualidade focando conceitos de Organização, Práticas e Diretrizes de desenvolvimento de software;
- ⊕ Esses conceitos são tratados na área de Arquitetura de Software;





O que é Arquitetura de Software?





Arquitetura de Software



- ⊕ Corresponde à **estrutura** que abrange os **componentes** do software, as propriedades externamente visíveis desses componentes e as relações entre eles;
- ⊕ Um **componente** pode ser um módulo de programa, uma classe orientada a objetos ou pode ainda ser estendido para abranger um banco de dados, etc;
- ⊕ As **propriedades** dos componentes são características necessárias para o entendimento de como eles interagem com outros componentes;
- ⊕ As **relações** podem ser, por exemplo, uma chamada de módulo, uma conexão a um banco de dados, etc.





Estilos de Arquitetura



- ✦ Arquitetos da Construção Civil usam diversos estilos arquitetônicos para a construção de um casa;
- ✦ Por exemplo, estilo Colonial, Estilo Contemporâneo, Estilo Clássico, etc





Estilos de Arquitetura de Software



- ⊕ Arquiteturas Centralizadas em Dados;
- ⊕ Arquiteturas de Fluxo de Dados;
- ⊕ Arquitetura de Chamadas e Retornos;
- ⊕ Arquiteturas Orientadas a Objetos;
- ⊕ Arquitetura em Camadas.

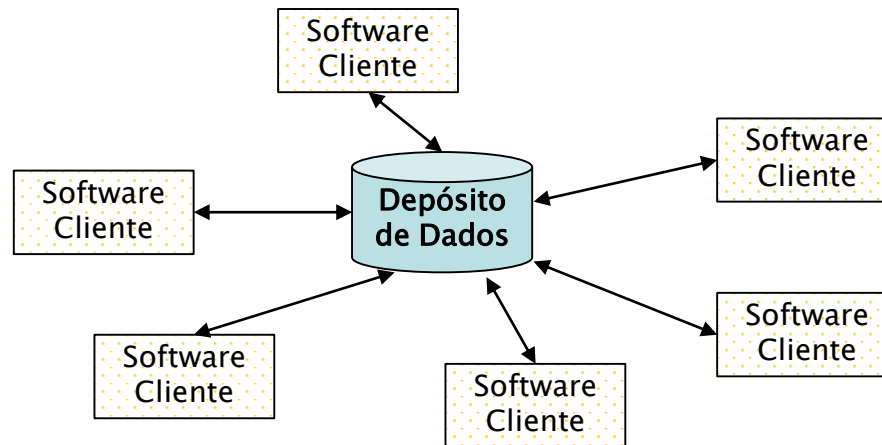




Arquiteturas Centralizadas em Dados



- ✦ Um repositório de dados (banco de dados) reside no centro dessa arquitetura e é em geral acessado por outros componentes que atualizam, acrescentam, eliminam ou modificam os dados contidos nesse repositório;
- ✦ O software-cliente acessa um repositório central e executam processos de forma independente.

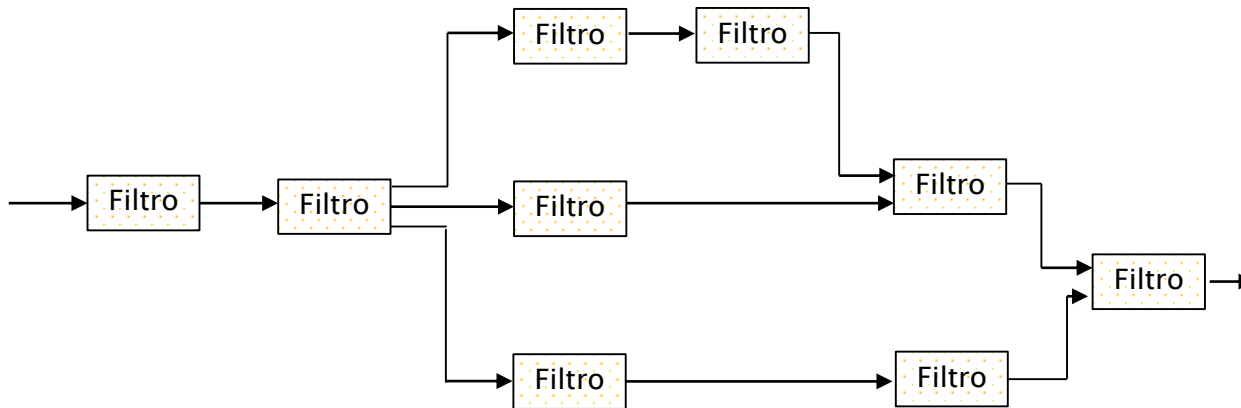




Arquiteturas de Fluxo de Dados



- ⊕ Estilo no qual dados de entrada devem ser **transformados** por meio de uma série de componentes computacionais ou de manipulação de dados de saída;
- ⊕ Exemplo: Processo **batch** de compilação de linguagens de programação.

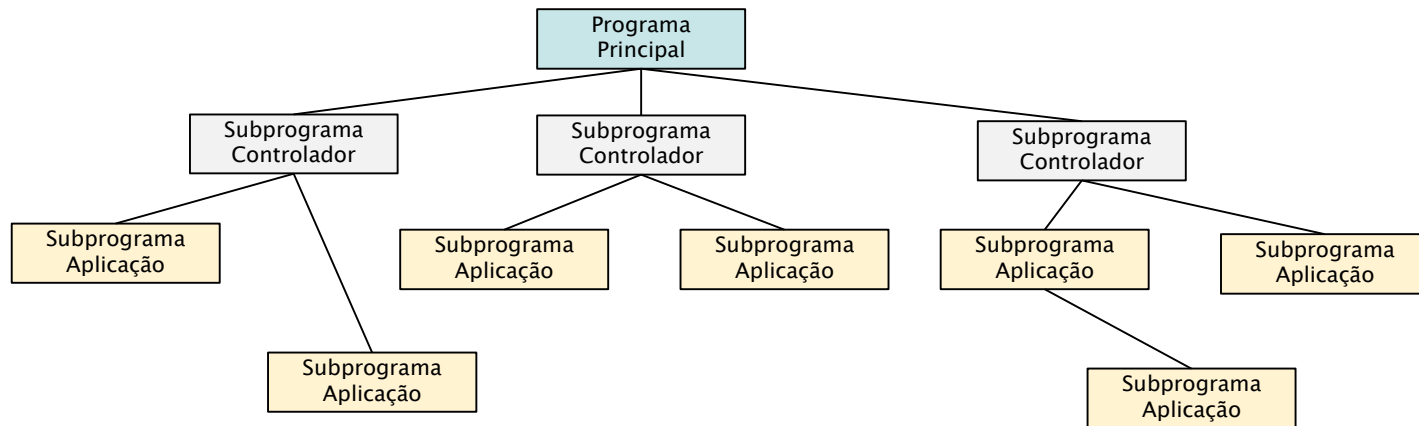




Arquiteturas de Chamadas e Retornos



- ⊕ Estrutura clássica de programação no qual decompõe-se a função em uma **hierarquia de controle** e um “**programa principal**” invoca uma série de componentes de programa que, por sua vez, podem invocar outros. A chamada pode ser feita por procedimentos remotos em um ambiente distribuído.

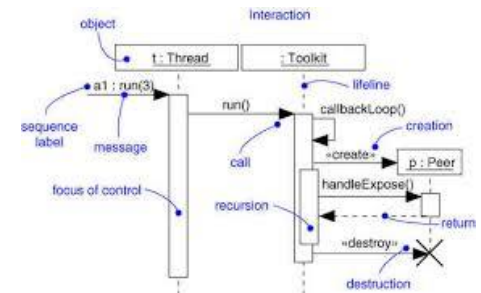
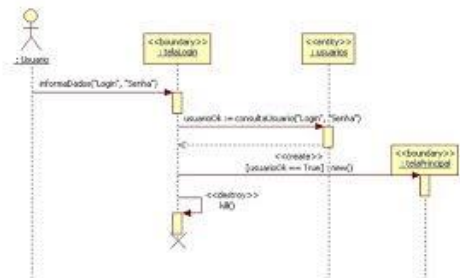




Arquiteturas Orientadas a Objeto



- Essa estrutura combina dados com funções numa única entidade (**objeto**), facilitando a decomposição do problema, manutenção e reuso;
- É comum utilizar a arquitetura orientada a objetos em sistemas de informação como sistemas de consulta e empréstimos online de bibliotecas de instituições de ensino que dispõem de componentes de cadastro de usuários e componentes de autenticação de usuários.
- Note que **componentes similares** existem em outros sistemas de informações, tais como sites de conteúdos, que exigem cadastro e autenticação de qualquer usuário antes de disponibilizar o conteúdo;
- Os componentes encapsulam dados e as operações que devem ser aplicadas para manipular dados. A comunicação e a coordenação entre componentes é feita por meio de passagem de **mensagens**.





Arquitetura em Camadas



- ⊕ A arquitetura do sistema de software é organizada num conjunto de camadas, oferecendo maior flexibilidade e suporte a portabilidade.
- ⊕ A identificação do nível de abstração nem sempre é evidente e perde-se desempenho à medida que o número de camadas cresce.
- ⊕ Exemplo desse estilo compreende os sistemas **Web** de múltiplas camadas que separa cliente, servidores de aplicação, servidores Web e outros clientes Web.

