



# MODELOS E PADRÕES DA ENGENHARIA DE SOFTWARE

Prof. Luis Fernando dos Santos Pires

# Introdução à Engenharia de Software

- Engenharia
- É a ciência de adquirir e de aplicar conhecimentos matemáticos, técnicos e científicos na criação, aperfeiçoamento e implementação de utilidades, tais como materiais, estruturas, máquinas, aparelhos, sistemas ou processos, que realizem uma determinada função ou objetivo.



# Introdução à Engenharia de Software

- Software
- Software de computador é um produto que profissionais de software desenvolvem e ao qual dão suporte no longo prazo. (PRESSMAN, 2011)
- Programas de computador e documentação associada. Os produtos de software podem ser desenvolvidos para um cliente específico ou para um mercado geral. (SOMMERVILLE, 2019)



# Introdução à Engenharia de Software

- O que seria então Engenharia de Software?
- Engenharia de Software = programar?
- Erroneamente, muitas vezes, desenvolver software é confundido com programação.



# Introdução à Engenharia de Software

- Programação é uma abordagem (abordagem de construção) corretamente adotada quando se pensa em problemas pequenos. Por exemplo: calcular médias e ordenar conjuntos de dados.
- Para problemas mais complexos é necessário adotar uma abordagem de engenharia. Por exemplo: desenvolver um sistema de informações para um banco.



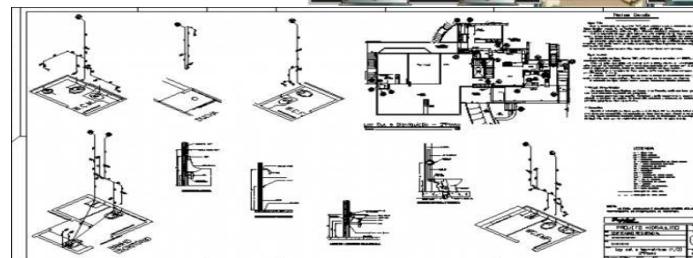
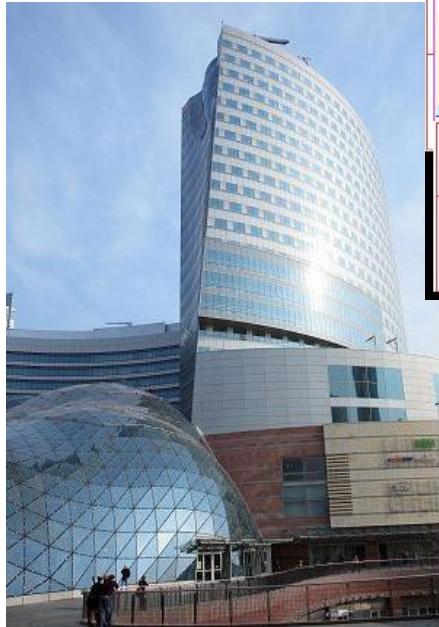
# Introdução à Engenharia de Software



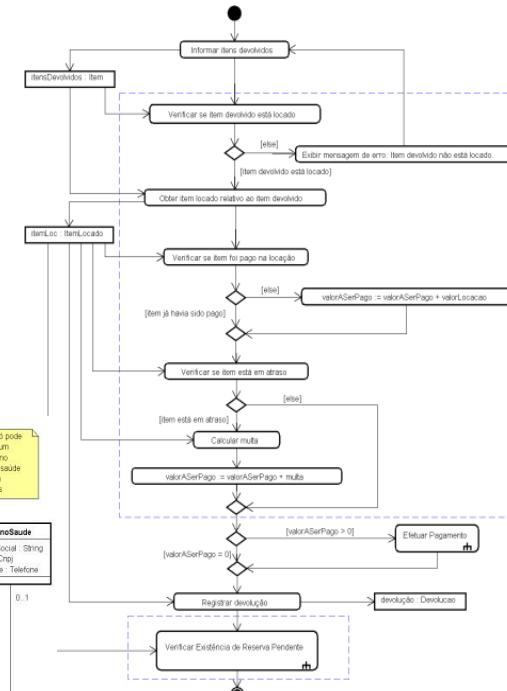
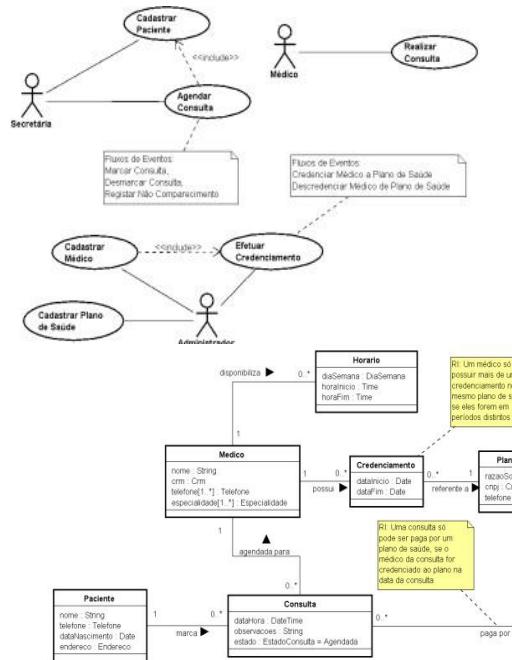
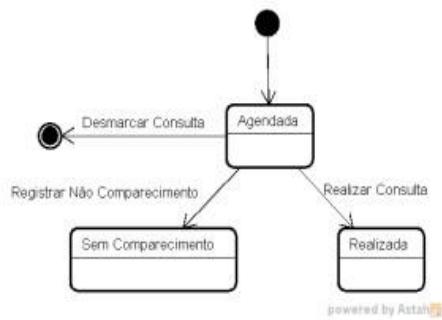
- Construir uma casinha de cachorro é um problema simples. O próprio dono pode comprar os materiais e construir a casinha em um final de semana.
- Construir um edifício requer um projeto de engenharia civil, planejamento da execução da obra e desenvolvimento de modelos (maquetes e plantas de diversas naturezas), até a realização da obra, que deve ocorrer por etapas. Ao longo da realização do trabalho, é necessário realizar um acompanhamento para verificar prazos, custos e a qualidade do que se está construindo.



# Analogia - Engenharia Civil



# Engenharia de Software



# Engenharia de Software

- A Engenharia de Software surgiu com o objetivo de melhorar a qualidade dos produtos de software e aumentar a produtividade no processo de desenvolvimento.
- Trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software.



# Engenharia de Software

- Propõe a divisão do problema em problemas menores, cujas soluções devem ser integradas por uma arquitetura. As soluções devem ser obtidas utilizando-se procedimentos (métodos, técnicas, roteiros etc), bem como ferramentas que automatizam o trabalho (ou parte dele).
- Tipicamente, são exigidas várias pessoas, cujo esforço deve ser planejado, coordenado e acompanhado. Também é requerido que a qualidade do que se está produzindo seja sistematicamente avaliada.



# Engenharia de Software

- “Engenharia de Software é uma disciplina de engenharia que está preocupada com os aspectos da produção de software, desde sua concepção inicial até sua operação e manutenção.” (SOMMERVILLE, 2019)
- Quais são eles?
- Especificação, Desenvolvimento, Validação, Evolução.  
(SOMMERVILLE, 2019)



# Engenharia de Software

- Quais são os principais desafios da engenharia de software?
- Lidar com a diversidade crescente, demandas por tempos de entrega reduzidos e desenvolver software confiável.
- Quais são os custos da engenharia de software?
- Aproximadamente 60% dos custos de software são custos de desenvolvimento; 40% são custos de teste. Para software customizado, os custos de evolução geralmente excedem os custos de desenvolvimento.



# O processo de software

- Um conjunto estruturado de atividades necessárias para desenvolver um sistema de software.
- Existem vários processos de desenvolvimento de software diferentes mas todos envolvem:
  - ✓ especificação – definição do quê o sistema deve fazer;
  - ✓ projeto e implementação – definição da organização do sistema e implementação do sistema;
  - ✓ validação – checagem de que o sistema faz o que o cliente deseja;
  - ✓ evolução – evolução em resposta a mudanças nas necessidades do cliente.
- Um modelo de processo de desenvolvimento de software é uma representação abstrata de um processo. Ele apresenta uma descrição do processo de uma perspectiva em particular.



# Descrições de processo de software

- Quando descrevemos e discutimos processos, geralmente falamos sobre as atividades desses processos, tais como especificação de modelo de dados, desenvolvimento de interface de usuário, etc. e organização dessas atividades.
- Descrições de processos também podem incluir:
  - ✓ Produtos, que são os resultados de uma atividade do processo;
  - ✓ Papéis, que refletem as responsabilidades das pessoas envolvidas no processo;
  - ✓ Pré e pós-condições, que são declarações que são verdadeiras antes e depois de uma atividade do processo ser executada, ou um produto produzido.



# Processos dirigidos a planos e ágeis

- Processos dirigidos a planos são processos em que todas as atividades do processo são planejadas com antecedência e o progresso é medido em relação a esse plano.
- Nos processos ágeis o planejamento é incremental e é mais fácil modificar o processo para refletir alterações nos requisitos do cliente.
- Na realidade, os processos mais práticos incluem elementos dos processos ágeis e dirigidos a planos.
- Não existe processo de software certo ou errado.



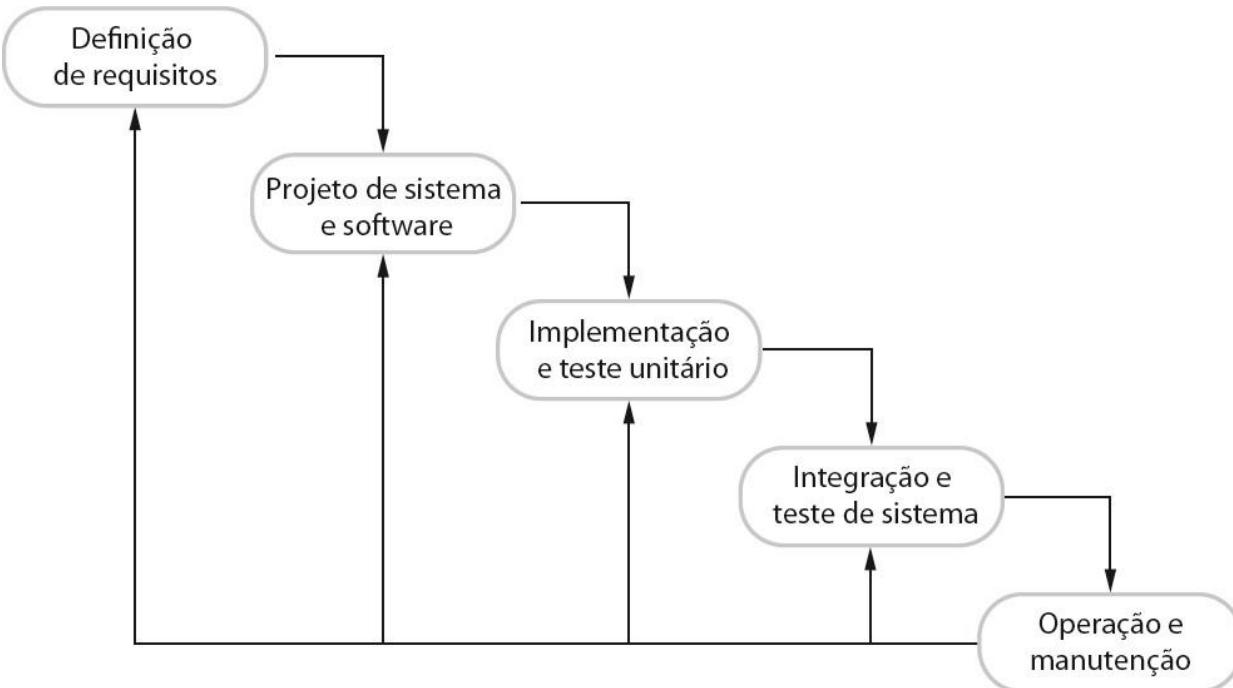
# Modelos de processo de software

- Modelo Cascata - Modelo dirigido a planos. Fases de especificação e desenvolvimento separadas e distintas.
- Desenvolvimento Incremental - Especificação, desenvolvimento e validação são intercaladas. Pode ser dirigido a planos ou ágil.
- Engenharia de software orientada a reúso - O sistema é montado a partir de componentes já existentes. Pode ser dirigido a planos ou ágil.

Na realidade a maioria dos grandes sistemas são desenvolvidos usando um processo que incorpora elementos de todos esses modelos.



# Modelo cascata



# Fases do modelo cascata

- Existem fases identificadas e separadas no modelo cascata:
  - ✓ Análise e definição de requisitos
  - ✓ Projeto de sistema e software
  - ✓ Implementação e teste de unidade
  - ✓ Integração e teste de sistema
  - ✓ Operação e manutenção
- O principal inconveniente do modelo cascata é a dificuldade de acomodação de mudanças depois que o processo já foi iniciado. Em princípio, uma fase precisa ser completada antes de se mover para a próxima fase.

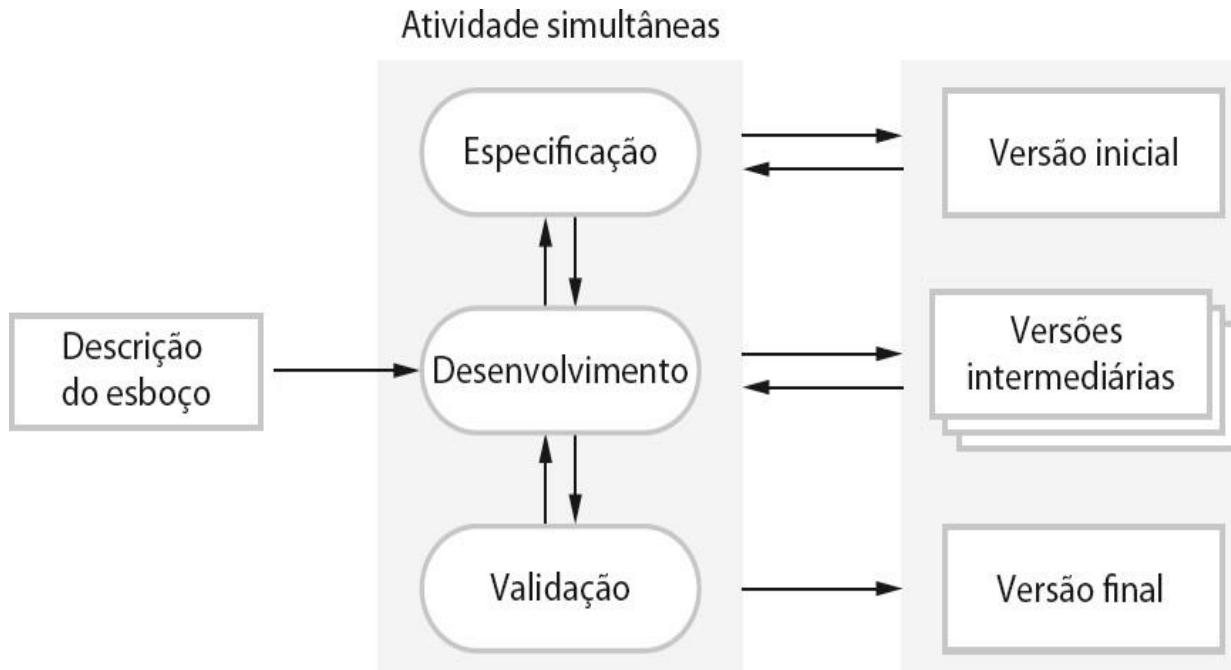


# Problemas do modelo cascata

- Divisão inflexível do projeto em estágios distintos torna difícil responder às mudanças nos requisitos do cliente.
  - ✓ Por isso esse modelo só é apropriado quando os requisitos são bem entendidos e as mudanças durante o processo de projeto serão limitadas.
  - ✓ Poucos sistemas de negócio possuem requisitos estáveis.
- O modelo cascata é mais usado em projetos de engenharia de grandes sistemas onde o sistema é desenvolvido em vários locais.
  - ✓ Nessas circunstâncias, a natureza do modelo cascata dirigida a planos ajuda a coordenar o trabalho.



# Desenvolvimento incremental



# Benefícios do desenvolvimento incremental

- O custo para acomodar mudanças nos requisitos do cliente é reduzido.
  - ✓ A quantidade de análise e documentação que precisa ser feita é bem menor do que o necessária no modelo cascata.
- É mais fácil obter feedback do cliente sobre o trabalho de desenvolvimento que tem sido feito.
  - ✓ Os clientes podem comentar demonstrações do software e ver quanto foi implementado.
- Possibilidade de mais rapidez na entrega e implantação de software útil para o cliente.
  - ✓ Os clientes podem usar e obter ganhos do software mais cedo do que é possível no processo cascata.



# Problemas do desenvolvimento incremental

- O processo não é visível.
  - ✓ Gerentes precisam de entregas regulares para medir o progresso. Se os sistemas são desenvolvidos de forma rápida, não é viável do ponto de vista do custo produzir documentação para refletir todas as versões do sistema.
- A estrutura do sistema tende a degradar conforme novos incrementos são adicionados.
  - ✓ A menos que tempo e dinheiro sejam gastos na reconstrução para melhorar o software, as mudanças regulares tendem a corromper a estrutura do sistema. A incorporação posterior de mudanças no software se torna progressivamente mais difícil e cara.

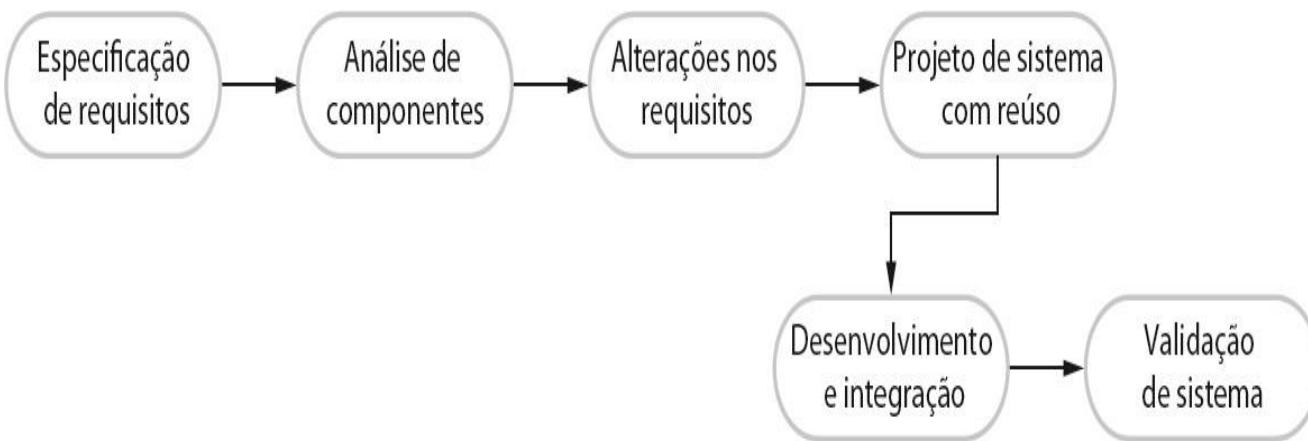


# Engenharia de software orientada a reúso

- Baseada no reúso sistemático em que os sistemas são integrados com componentes existentes ou Sistemas COTS (Commercial--off--the--shelf).
- Estágios do processo:
  - ✓ Análise de componentes;
  - ✓ Modificação de requisitos;
  - ✓ Projeto de sistema com reúso;
  - ✓ Desenvolvimento e integração.
- Atualmente, o reúso é a abordagem padrão para a construção de vários tipos de sistemas de negócio.



# Engenharia de software orientada a reúso



# Tipos de componente de software

- Web services que são desenvolvidos de acordo com padrões de serviço e ficam disponíveis para chamada remota.
- Coleções de objetos que são desenvolvidas como um pacote para ser integrado com um framework como .NET ou J2EE.
- Sistemas de software stand-alone (COTS) que são configurados para uso em ambientes específicos.



# Atividades do processo

- Processos de software reais são sequências intercaladas de atividades técnicas, colaborativas e gerenciais com o objetivo geral de especificar, projetar, implementar e testar um sistema de software.
- As quatro atividades de processo básicas, especificação, desenvolvimento, validação e evolução são organizadas de forma diferente em processos de desenvolvimento distintos.
- No modelo cascata, elas são organizadas em sequências, enquanto no desenvolvimento incremental são intercaladas.



# Especificações de software

- O processo de estabelecer quais serviços são necessários e as restrições na operação e desenvolvimento do sistema.
- Processo de engenharia de requisitos
  - ✓ Estudo de viabilidade  
É técnica e financeiramente viável construir o sistema?
  - ✓ Elicitação e análise de requisitos  
O que os stakeholders do sistema precisam ou esperam do sistema?
  - ✓ Especificação de requisitos  
Definição dos requisitos em detalhes.
  - ✓ Validação de requisitos  
Verificação da completude dos requisitos.



# Referências

- SOMMERVILLE, I., Engenharia de Software, 10<sup>a</sup> Edição. São Paulo: Pearson Prentice Hall, 2019.
- PRESSMAN, R. S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. McGraw-Hill, 2011.





# MODELOS E PADRÕES DA ENGENHARIA DE SOFTWARE

Prof. Luis Fernando dos Santos Pires