Capítulo 2

Processos de Software

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

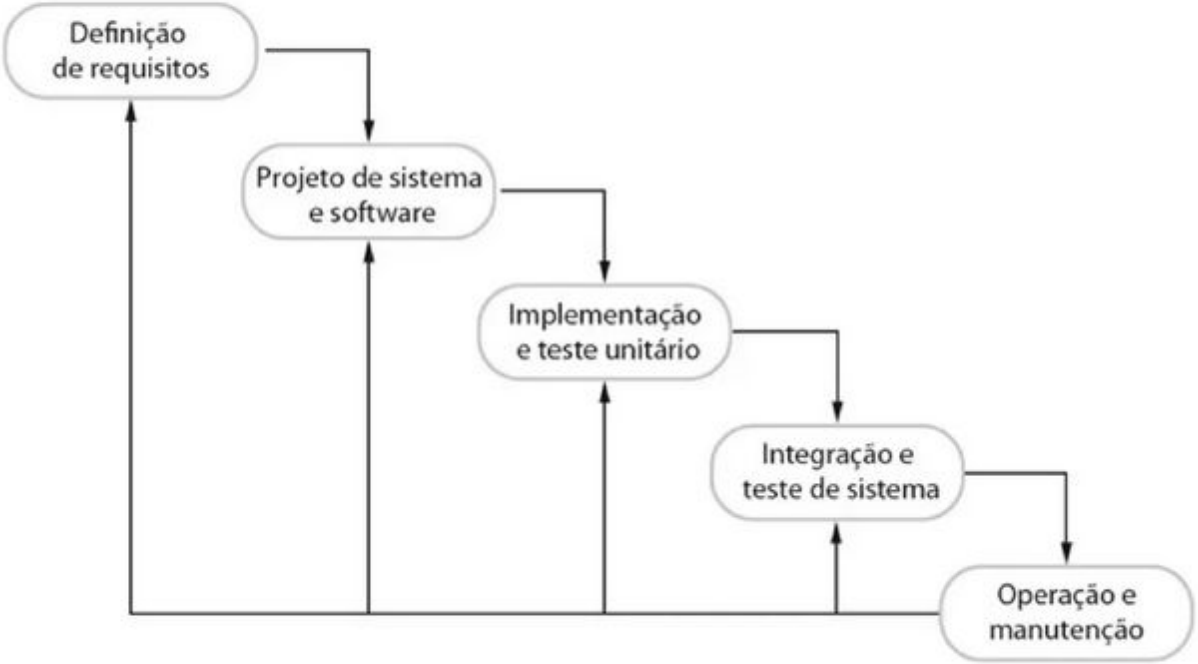
- Processos dirigidos a planos são processos em que todas as atividades do processo são planejadas com antecedência e o progresso é medido em relação a esse plano.
- Nos processos ágeis o planejamento é incremental e é mais fácil modificar o processo para refletir alterações nos requisitos do cliente.
- 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.

O modelo cascata



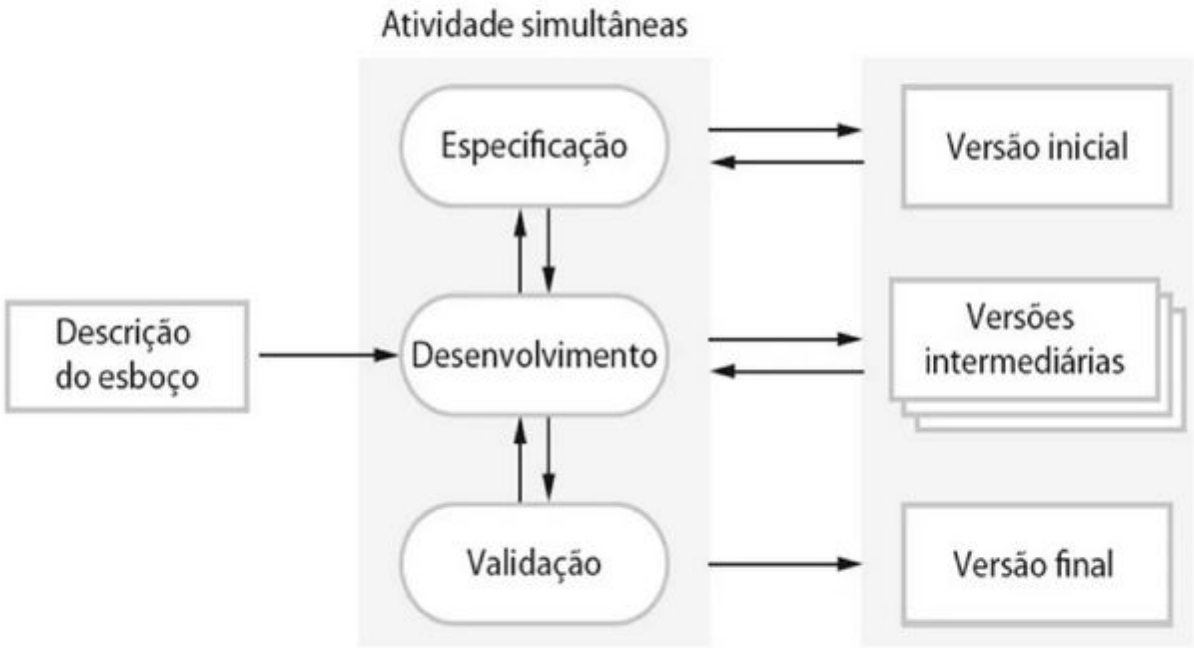
Fases do modelo cascata

- 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.

Entrega incremental

- Ao invés de entregar o sistema em uma única entrega, o desenvolvimento e a entrega são distribuídos em incrementos, nos quais cada incremento entrega parte da funcionalidade necessária.

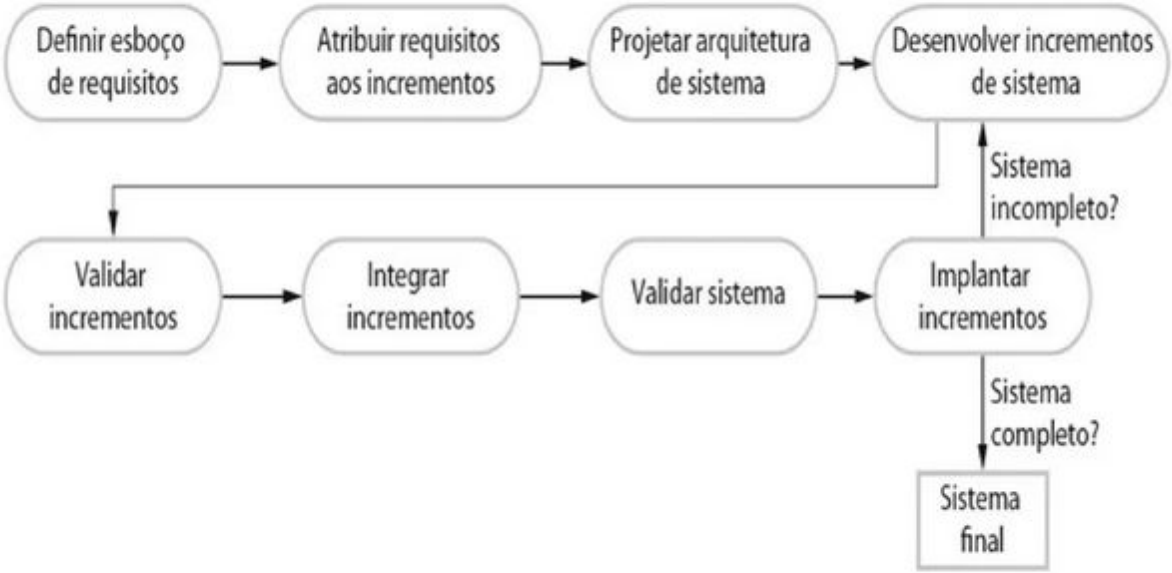
- Os requisitos do usuário são priorizados e os requisitos de mais alta prioridade são incluídos nos primeiros incrementos.

- Assim que o desenvolvimento de um incremento é iniciado os requisitos são congelados, mas os requisitos dos incrementos posteriores podem continuar a evoluir.

Desenvolvimento e entrega incremental

- Desenvolvimento incremental
  - ✓ Desenvolve o sistema em incrementos e avalia cada incremento antes de proceder com o desenvolvimento do próximo incremento;
  - ✓ Abordagem normalmente usada em métodos ágeis;
  - ✓ Avaliação feita por representantes do usuário/cliente.
- Entrega incremental
  - ✓ Implanta um incremento para uso do usuário-final;
  - ✓ Avaliação mais realística sobre o uso prático do software;
  - ✓ Difícil de implementar para sistemas substitutos devido aos incrementos possuírem menos funções do que o sistema que está sendo substituido.

Entrega incremental



Vantagens da entrega incremental

- Os valores podem ser entregues ao cliente junto com cada incremento, e funções do sistema ficam disponíveis mais rápidamente.
- Primeiros incrementos agem como protótipos para ajudar a deduzir requisitos para incrementos posteriores.
- Menor risco de falha geral do projeto.
- Os serviços mais prioritários do sistema tendem a serem mais testados.

Problemas da entrega incremental

- A maioria dos sistemas requer um conjunto de funções básicas que são usadas por diferentes partes do sistema.

  - ✓ Como os requisitos não são definidos em detalhes até que um incremento seja implementado, pode ser difícil identificar funções comuns que são necessárias a todos os incrementos.

- A essência dos processos iterativos é que a especificação seja desenvolvida em conjunto com o software.

  - ✓ No entanto, essa pode entrar em conflito com o modelo de aquisição de muitas organizações, nos quais a especificação completa do sistema é parte do contrato de desenvolvimento do sistema.

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.