



Processo de Desenvolvimento de Software – Atividades

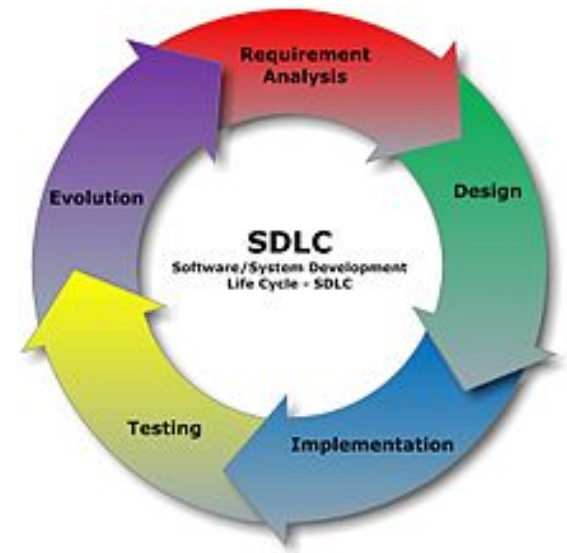
Prof. Ingrid Nunes

INF01127 - Engenharia de Software N

Processo de Desenvolvimento de SW



- Especificar artefatos que devem ser produzidos
- Indicar
 - **quem** deve fazer
 - **o que** deve ser feito e quando
 - se possível, **como** fazer
- Especificar o **ciclo de vida** de um sistema
 - da concepção à entrega e manutenção



Processo de Desenvolvimento de SW



- Conjunto de atividades
 - Meta é o desenvolvimento ou evolução de software
- **Atividades genéricas**
 - Parte de qualquer processo de software
 - Atividades complexas
 - Subdivididas em várias outras atividades
- Processo determina
 - Refinamento em sub-atividades
 - Forma e ordem de execução
 - Pré- e pós-condições
 - Produtos (ou artefatos)
 - Entradas e saídas das atividades
 - Papéis (responsabilidades)

Processo de Desenvolvimento de SW



- Atividades de apoio
 - Gerência de configuração e de mudanças
 - Gerência de projeto
 - Documentação
 - Gerência de ambientes
 - Etc.

Processo de Desenvolvimento de SW

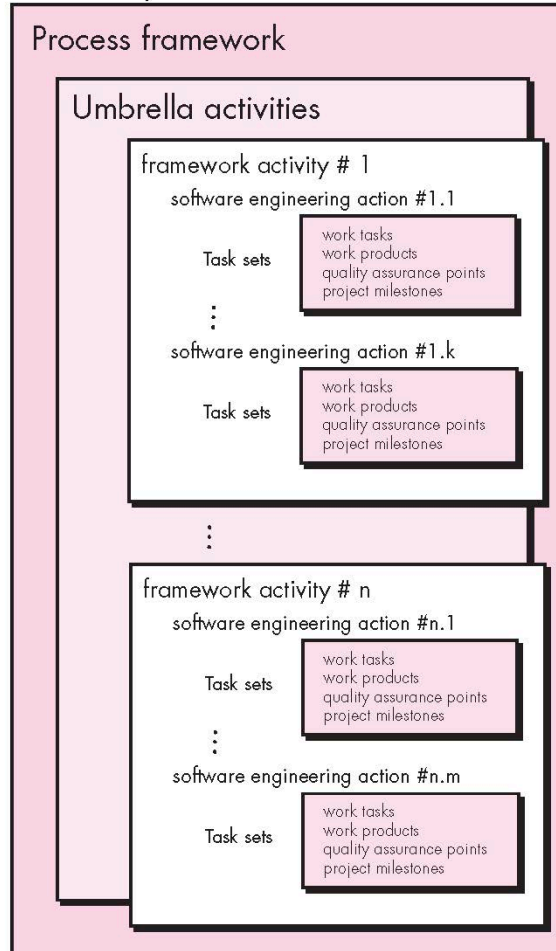


- Metodologia *versus* Processo
 - **Metodologia**
 - Um corpo de métodos, regras e postulados empregados por uma disciplina
 - Um determinado procedimento ou conjunto de procedimentos
 - **Processo**
 - Uma série de ações ou operações levando a um fim
 - Processo de Desenvolvimento **de Software**
 - **Objetivo**: desenvolver um **produto de software**

Processo de Desenvolvimento de SW



Software process



Processo de Desenvolvimento de SW

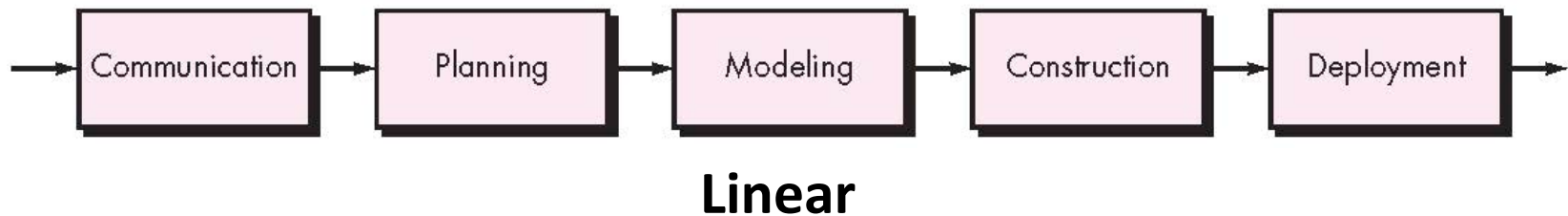


- Atividades genéricas
 - **Especificação**
 - Definição da funcionalidade do software e restrições na sua operação
 - **Projeto e Implementação**
 - Produção de um software que esteja de acordo com a especificação
 - **Validação e Verificação**
 - Verificação de que o software está correto e é o que o cliente deseja
 - **Evolução**
 - Mudança do software em resposta às demandas de mudança

Processo de Desenvolvimento de SW



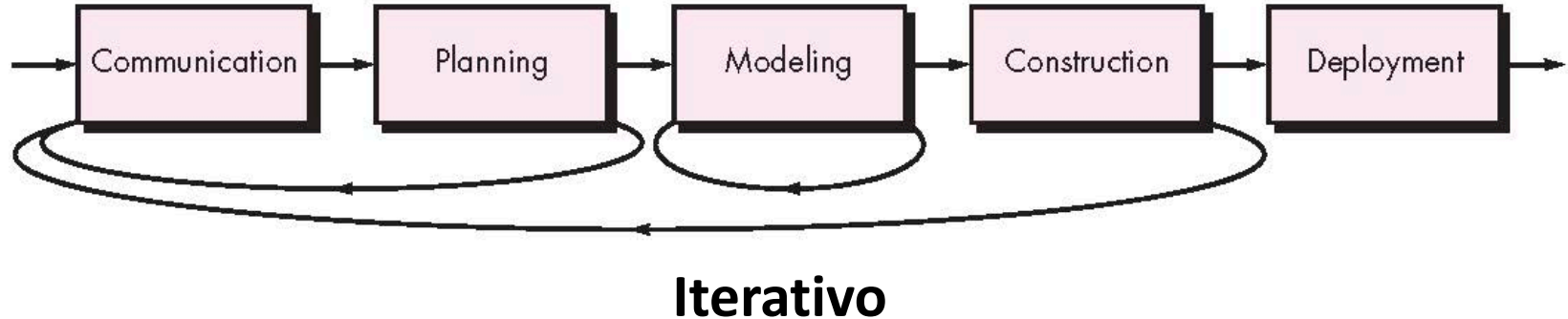
- Característica Importante
 - Fluxos do Processo



Processo de Desenvolvimento de SW



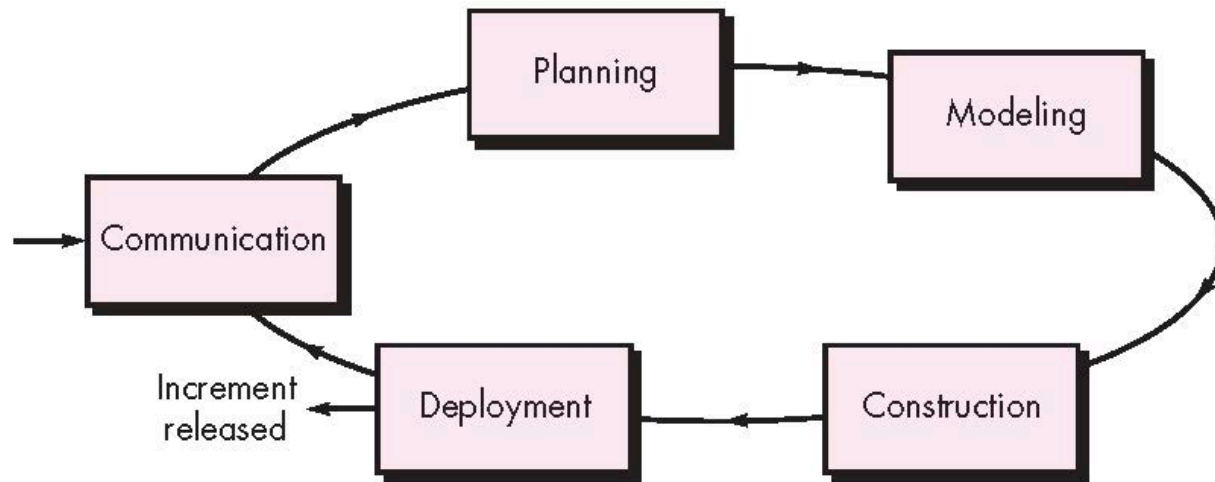
- Característica Importante
 - Fluxos do Processo



Processo de Desenvolvimento de SW



- Característica Importante
 - Fluxos do Processo

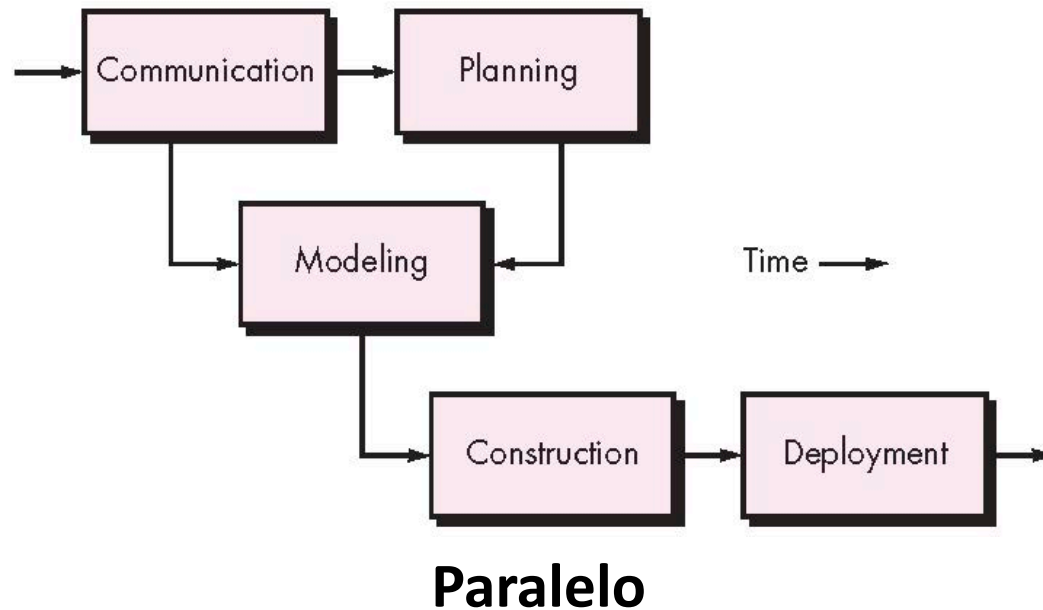


Evolucionário

Processo de Desenvolvimento de SW



- Característica Importante
 - Fluxos do Processo



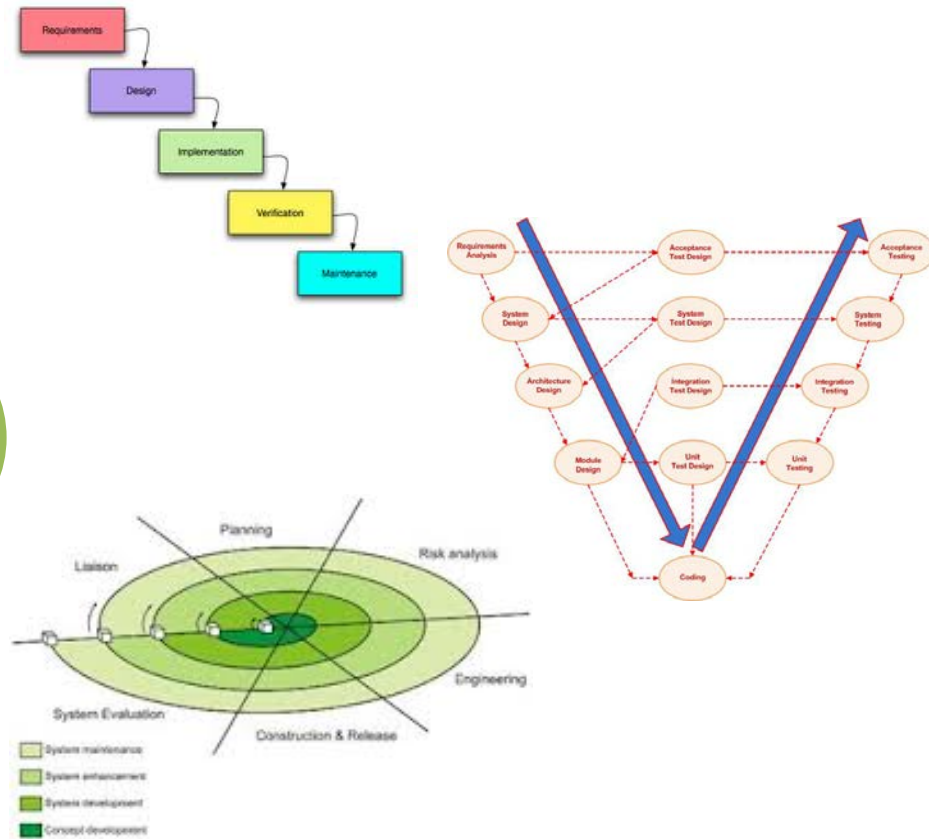
Processo de Desenvolvimento de SW



Atividades Genéricas



Modelos de Processo





Processo de Desenvolvimento de Software



ATIVIDADES GENÉRICAS

Especificação: o início de tudo...



“A intenção do cliente é...”

O mais importante aqui é...



A IDÉIA É VIÁVEL?

O que devo fazer exatamente?



**Ou, em outras
palavras, quais são
os requisitos da
aplicação?**

O que devo fazer então?



Registrar os Requisitos

Haverá um entendimento comum?



Especificação



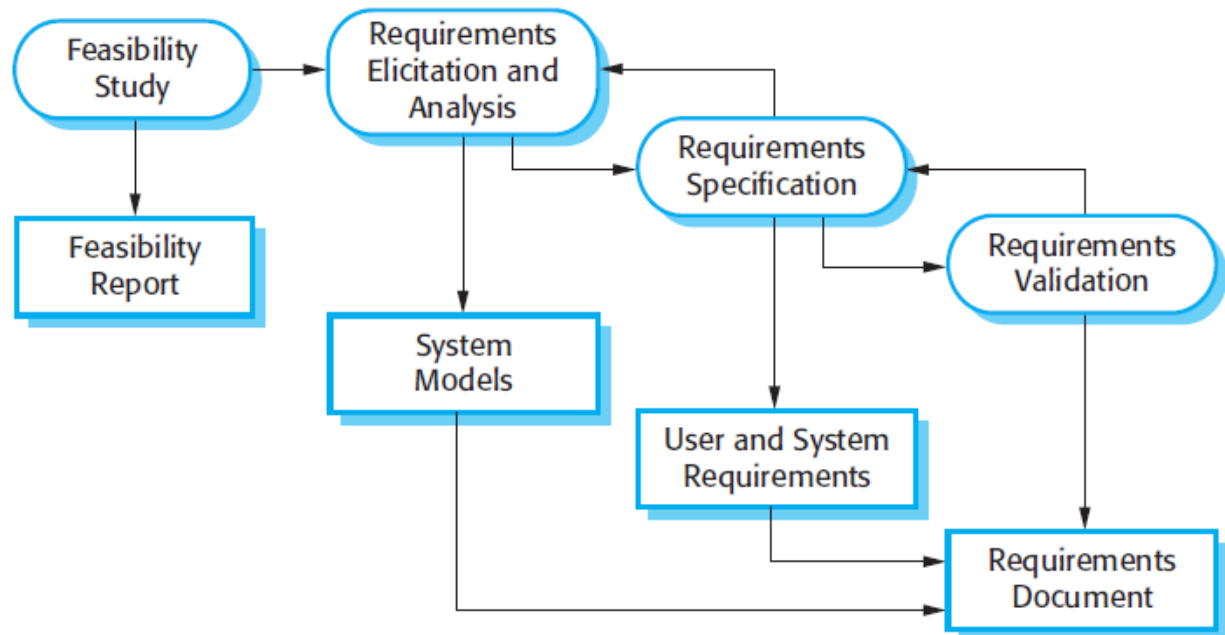
- Especifica
 - Quais serviços são necessários
 - Restrições que têm impacto no desenvolvimento e operação do sistema
- **Estudo de Viabilidade**
 - Necessidades podem ser atendidas pela TI
 - Viável do ponto de vista comercial, custo, tempo, etc.
 - Definição do escopo
 - Rápido, barato e superficial
 - Decisão de prosseguir ou não
 - Visão do produto

Especificação



- **Identificação (Análise) de Requisitos**

- Observação de sistemas existentes, conversas com potenciais usuários, análise de processos, etc.
- Pode envolver protótipos, provas de conceitos, modelos de sistema, etc.



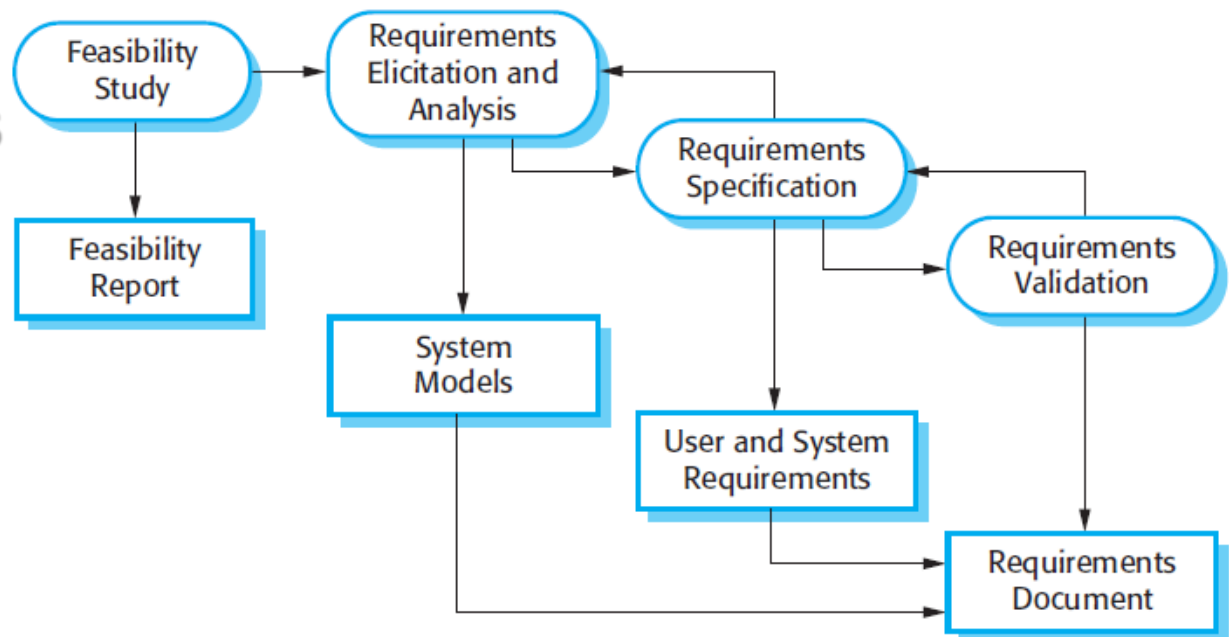
Especificação



- **Especificação de Requisitos**

- Tradução das informações coletadas na fase de elicitação
- Requisitos de negócio (abstratos)
- Requisitos de Sistema: RF e RNF

- **Validação de Requisitos**



Projeto e Implementação



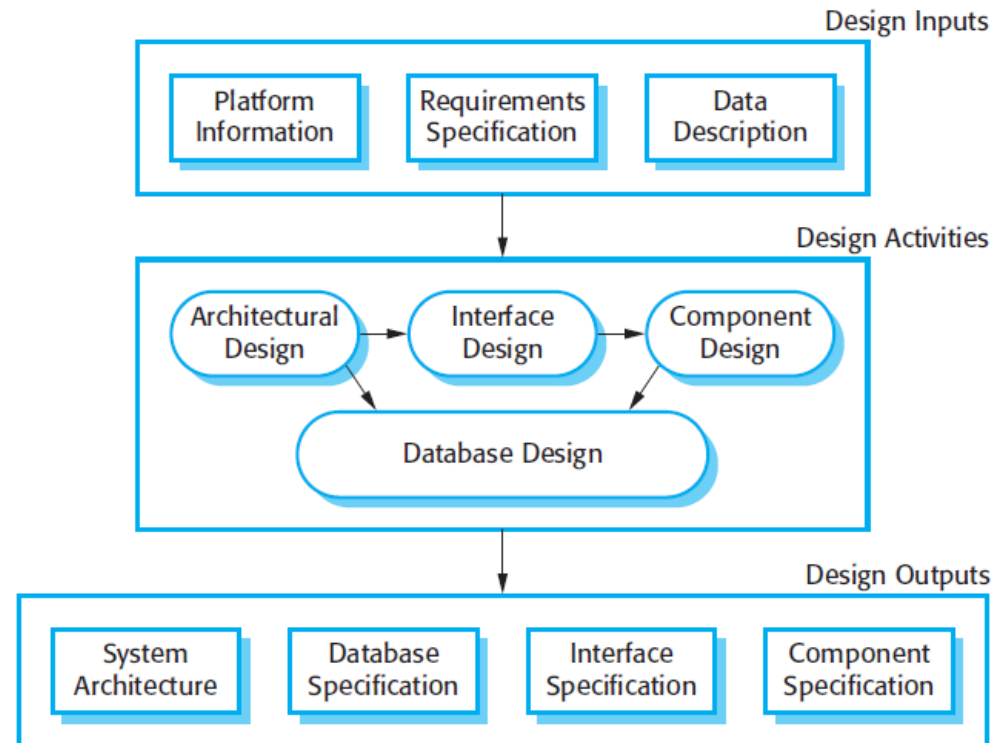
- Conversão da especificação de sistema em um sistema executável
 - **Análise e Projeto** de Software
 - Conceber uma **estrutura** de software que atenda aos requisitos especificados
 - Pode requerer **refinamento** da **especificação**
 - Considerar **requisitos funcionais** e **não funcionais**
 - **Atributos** de **qualidade** de software
 - **Implementação**
 - Transformar essa estrutura em um programa **executável**
- Atividades de projeto e implementação
 - Fortemente relacionadas
 - Frequentemente intercaladas

Projeto e Implementação



- **Projeto da Arquitetura**

- Definição da estrutura geral do sistema
- Principais componentes
 - Módulos ou sub-módulos
- Seus relacionamentos
- Como estão distribuídos

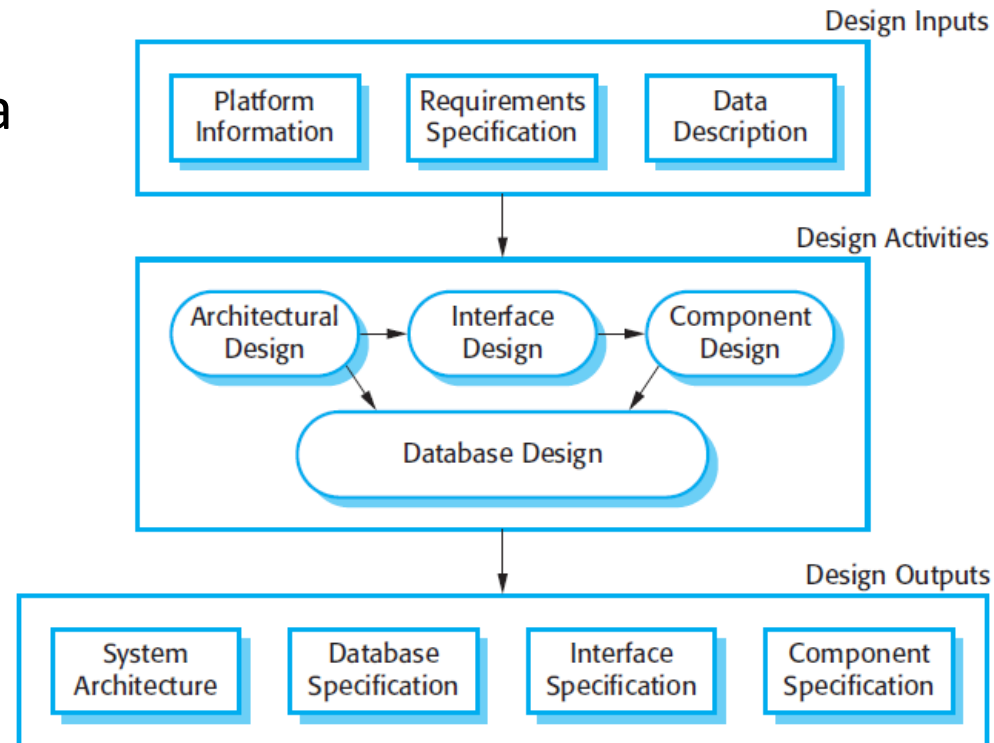


Projeto e Implementação



- **Projeto de Interface**

- Interfaces entre componentes do sistema
- Deve ser não ambígua
- Permite projeto e desenvolvimento **em paralelo**

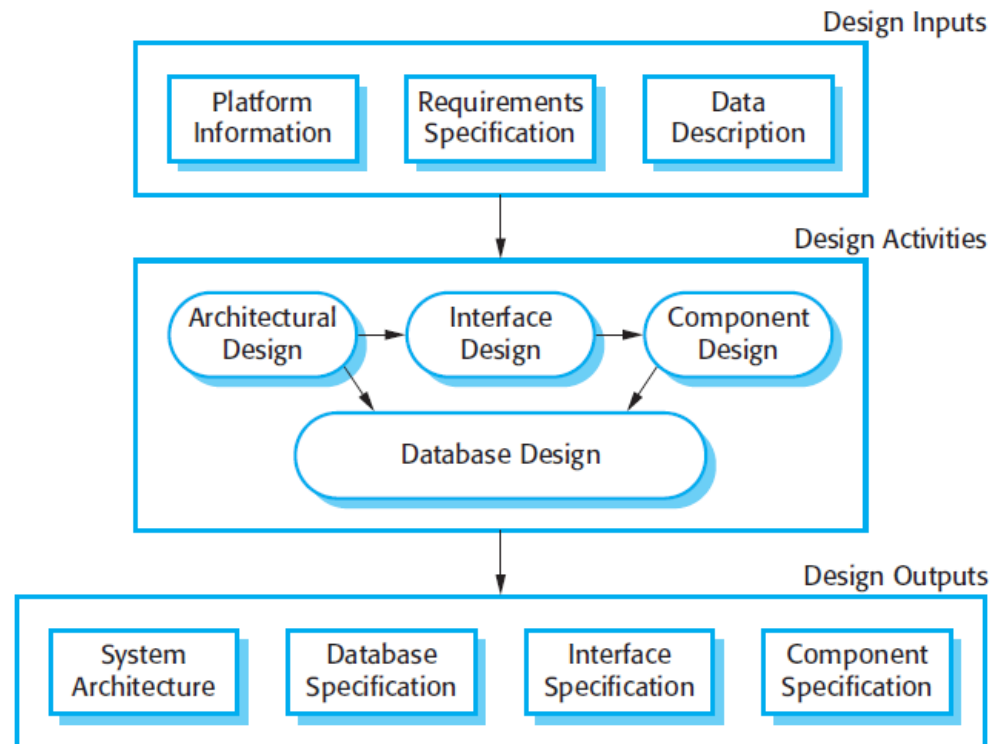


Projeto e Implementação



- **Projeto de Componentes**

- Cada componente é projetado
- Pode ser
 - Mudanças em componentes reusáveis
 - Projeto detalhado para geração de código

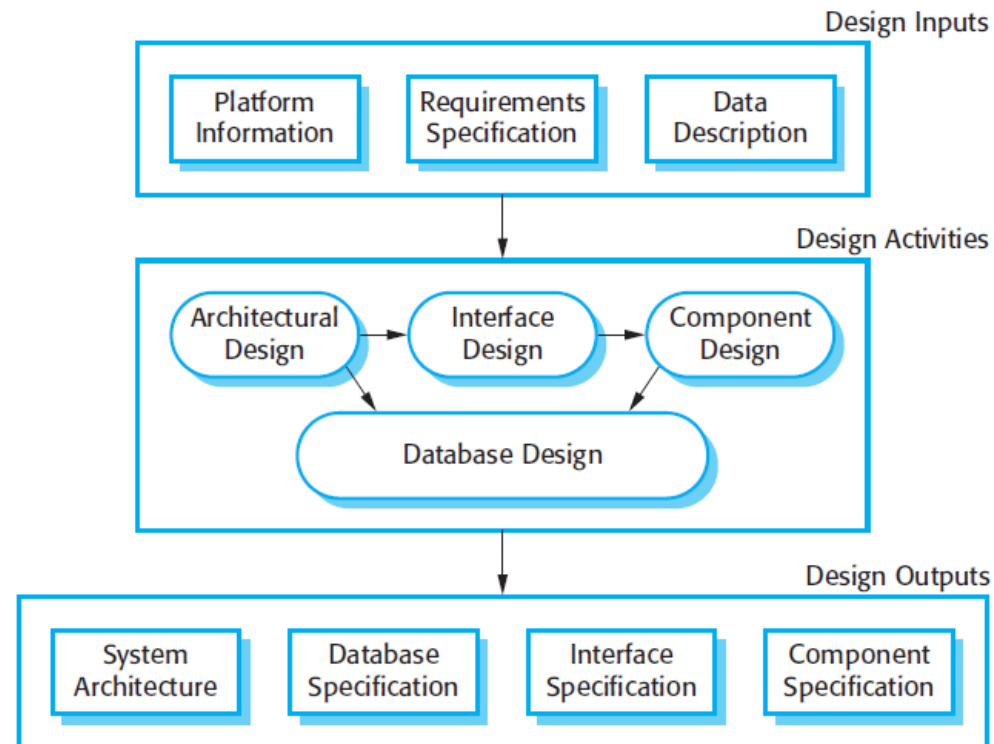


Projeto e Implementação



- **Projeto do Banco de Dados**

- Projeto de estrutura de dados
- Como dados serão representados em um banco de dados



Projeto e Implementação



- Programação e **Debugging**
 - Tradução de um projeto em programas
 - Mapeamento para uma linguagem de programação
 - Programação é freqüentemente uma atividade pessoal
 - Estilos de codificação
 - Melhores práticas
 - Padrões
 - Programadores desenvolvem **testes** para descobrir possível **falhas** em seus programas, e **remover** estas falhas (*debugging*)



Verificação e Validação (V&V)



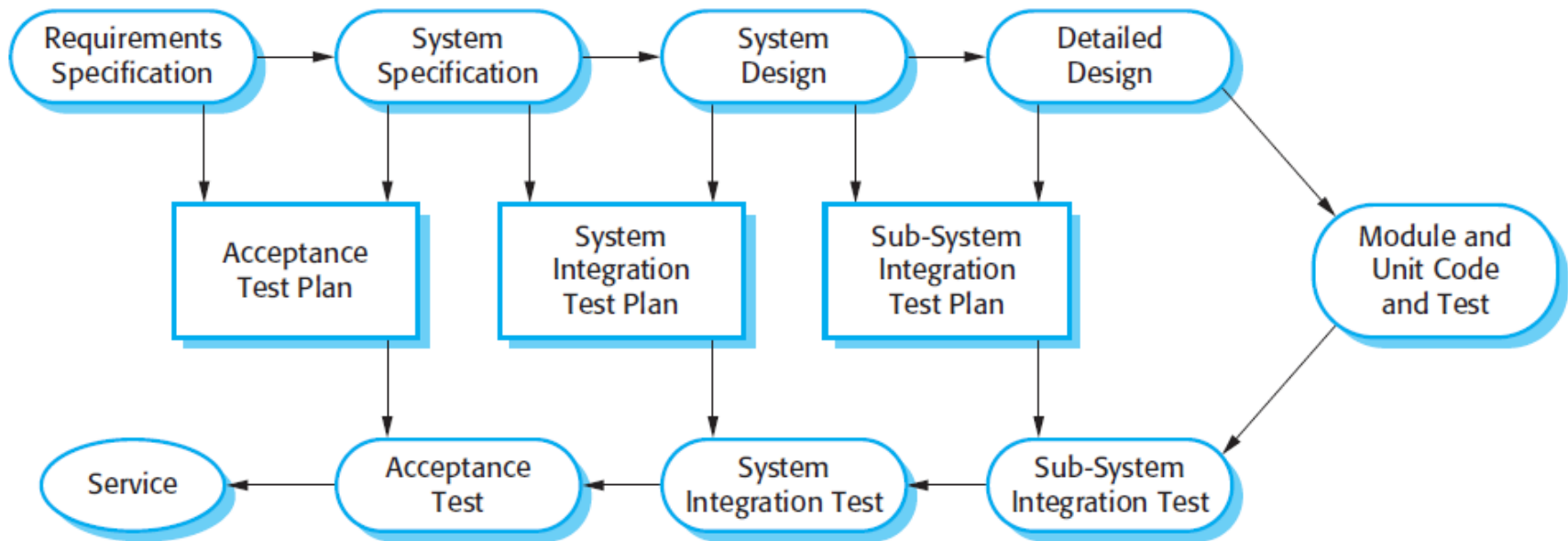
- Mostrar que um sistema
 - Está em conformidade com a sua **especificação**
 - Atende às **necessidades do cliente**
- Envolve processos de **inspeção, revisão e verificação** em todas atividades do processo
- Tipicamente, maior concentração após a implementação
 - Execução do sistema com casos de teste que são derivados da especificação, usando dados similares aos reais
 - Tendência que o desenvolvimento de casos de teste faça parte do processo de refinamento de requisitos e guie o projeto/implementação

Verificação e Validação (V&V)



- Diferentes níveis de teste
 - **Teste de Desenvolvimento**
 - Componentes testados independentemente
 - Uso de ferramentas de automação
 - E.g. JUnit
 - **Teste de Integração e Sistema**
 - Componentes são integrados
 - Testa a interação entre componentes
 - V&V de requisitos funcionais e não funcionais
 - **Teste de Aceitação** (alpha e beta)
 - Teste feito pelo usuário ao invés de dados simulados

Verificação e Validação (V&V)



Evolução



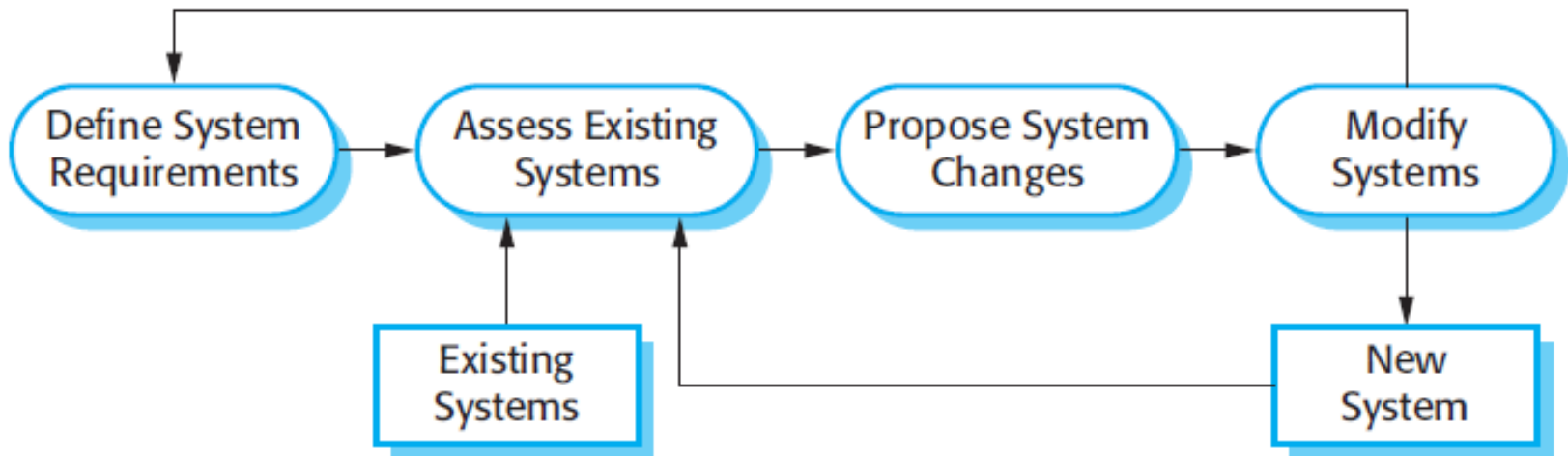
- **Software é flexível** por natureza
 - Por isso, pode mudar
- Por exemplo
 - Um requisito pode ser alterado devido a uma mudança no panorama econômico
 - Porque o hardware ficou mais barato
 - Porque o concorrente lançou um produto inovador
 - Porque o dono teve uma ideia brilhante
- É **imprevisível**
- **Manutenção**
 - Costuma representar até 70% do ciclo de vida do sistema
 - Costuma ser vista como uma atividade “desinteressante” (“castigo”)

Evolução



- Causas
 - Correção
 - Defeitos descobertos pelo usuário
 - Adaptação
 - Acomodação de mudanças no ambiente externo
 - Infraestrutura
 - Legislação
 - Processos de negócio
 - Melhoria
 - Adição de funcionalidades que vão melhorar o software
 - Melhor interface, mais funções, mais processos, etc.
 - Prevenção
 - Reengenharia
 - Torná-lo mais robusto a mudanças
- Tendência dos métodos de desenvolvimento incrementais
 - Sem distinção entre desenvolvimento/manutenção

Evolução



Pontos Importantes



- **Processos** de Software
 - **Atividades** envolvidas na produção de um sistema de software
- Engenharia de **requisitos**
 - Processo de desenvolvimento de uma **especificação** de software
 - Especificações devem comunicar as **necessidades** do usuário aos desenvolvedores de software
- Processos de **projeto** e **implementação**
 - Se preocupam em **transformar** especificação de requisitos em um sistema de software **executável**
 - **Métodos sistemáticos** de projeto podem ser usados como parte dessa transformação

Pontos Importantes



- **Validação** de software
 - Processo de checar que o sistema esteja de acordo com a especificação e **satisfaz** as necessidades **reais** dos usuários
- **Evolução** de software
 - Software existente é **modificado** para atender novos requisitos
 - Mudanças são **contínuas** e o software deve evoluir para continuar **útil**

Referências



- **Leitura Obrigatória**
 - **Sommerville, I. Engenharia de software, 9a edição. Pearson, 2011.**
 - **Capítulo 2**
- **Leitura Complementar**
 - Pressman, Roger. Engenharia de Software: Uma Abordagem Profissional, 7a edição. McGraw-Hill, 2011.
 - Capítulo 2

Perguntas?



- Este material tem contribuições de
 - Ingrid Nunes
 - Karin Becker
 - Lucinéia Thom
 - Marcelo Pimenta

Prosoft

