Engenharia de Software

Natália Schots

31/março/2015 – Aula 03

Agenda

- Introdução à Engenharia de Software
  - Princípios de Hooker
- Processos de Desenvolvimento de Software
- Modelos de Ciclos de Vida

Na aula passada...

Introdução à ES

- Elementos básicos
  - Processos
  - Métodos
  - Ferramentas
- Histórico
  - Crise do software
- Mitos da ES
  - Mitos gerenciais
  - Mitos do cliente
  - Mitos do desenvolvedor

Princípios de Hooker

7 princípios de Hooker (1/6)

- Tem que existir uma razão para se fazer software
  - Se não for possível identificar essa razão, é melhor não fazer
  - Fazer software, em última instância, consiste em "agregar valor para o usuário"
  - É importante enxergar os reais requisitos do software!

7 princípios de Hooker (2/6)

- Keep it simple, sir! (KISS)
  - “um projeto deve ser o mais simples possível, mas não mais simples que isso”
  - As soluções mais elegantes normalmente são simples
  - Fazer algo simples usualmente demanda mais tempo do que fazer de forma complexa

7 princípios de Hooker (3/6)

- Mantenha o estilo
  - O projeto de um software deve seguir um único estilo
  - A combinação de diferentes estilos corretos pode levar a um software incorreto
  - Padrões e estilos devem ser estabelecidos no início e seguidos por todos

7 princípios de Hooker (4/6)

- O que é produzido por você é consumido por outros
  - Sempre especifique, projete e codifique algo pensando que outros vão ler
  - Sempre exija qualidade nos produtos que você consome e forneça qualidade nos produtos que você produz

7 princípios de Hooker (5/6)

- Esteja pronto para o futuro
  - Sistemas de boa qualidade têm vida longa
  - Projete desde o início pensando na manutenção
- Planeje para reutilização
  - Pense no problema geral, e não só no problema específico
  - Busque por soluções já existentes

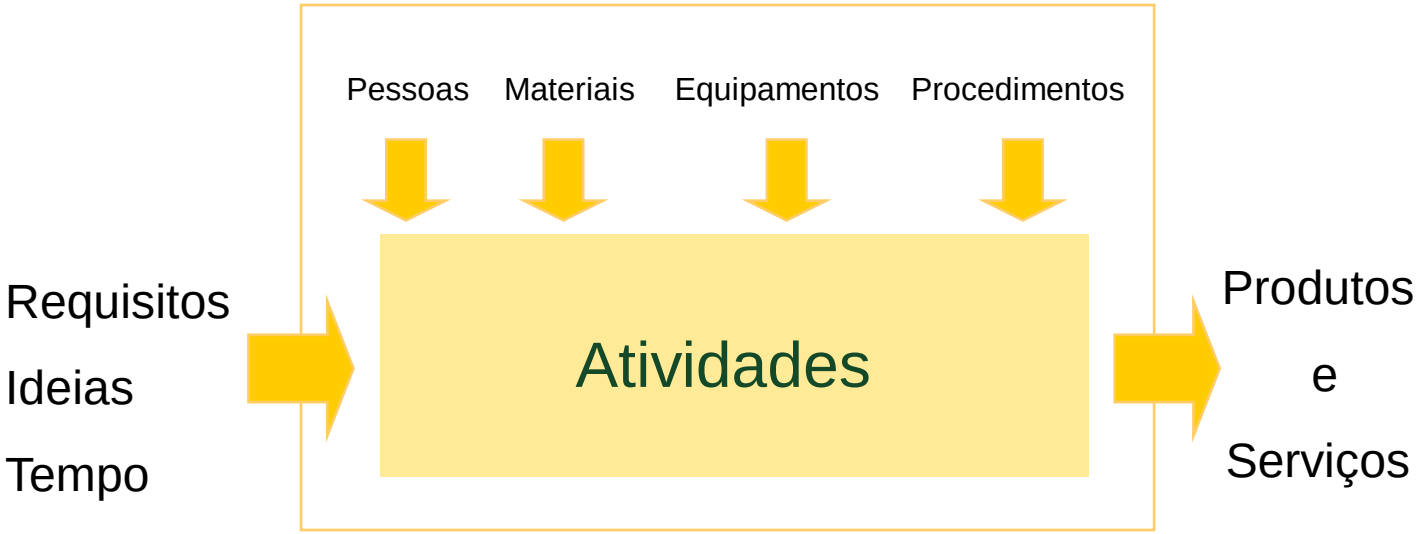
7 princípios de Hooker (6/6)

- Pense!
  - “Plano é desnecessário, mas planejar é indispensável” – D. Eisenhower
  - Avalie alternativas
  - Mitigue os riscos

Processos do desenvolvimento de software

Processo de Software

- Um conjunto de atividades inter-relacionadas ou interativas, que transforma insumos (entradas) em produtos (saídas) (ISO 9000, 2000)



[FLORAC e CARLETON, 1999]

Características de um processo

- Tecnologicamente competitivos, adaptáveis e adequados com relação ao tempo
- Capazes de produzir produtos que atingem as necessidades do cliente e do negócio
- Adequados à cultura organizacional

[ROCHA, 2013]

Descrição de um processo

- Pode ser descrito em termos de:
  - Propósito/Resultado
    - Tipo de definição útil quando não se quer definir as atividades de forma detalhada, mas sabe-se o objetivo do processo (propósito) e os resultados que este deve produzir
  - Atividades
    - É a abordagem mais comum, onde são descritas as atividades e suas inter-relações, bem como a sequência de execução de cada atividade
      - Cada atividade deve conter: procedimentos e métodos, ferramentas de apoio, artefatos de entrada e de saída, responsáveis etc.

[ROCHA, 2013]

Exemplo: Processo Cozinhar (1/4)

- Defina o processo em termos de Propósito/Resultado
  - Propósito: Fornecer o prato de comida ao cliente de acordo com o pedido
  - Resultados:
    - Um pedido que indica o prato de comida a ser produzido é comunicado ao cozinheiro
    - Um prato de comida é preparado
    - O garçom é avisado que o prato de comida está preparado

[ROCHA, 2013]

Exemplo: Processo Cozinhar (2/4)

- Defina o processo em termos de Atividades
  - **Artefato de Entrada:** Pedido solicitado
  - **Atividades:**
    - O pedido é impresso na cozinha na ordem em que foi solicitado
    - É verificado qual a comida a ser produzida
    - Os ingredientes que serão utilizados para preparar a comida são separados
    - A comida é preparada de acordo com a receita pré-definida
    - A comida é colocada no prato em que será servida
    - O prato de comida é decorado...

[ROCHA, 2013]

Exemplo: Processo Cozinhar (3/4)

- Defina o processo em termos de Atividades (cont.)
  - O prato de comida é colocado no passa-prato para o garçom pegar
  - O garçom é avisado que o prato de comida referente ao pedido X está pronto
  - **Responsável:** cozinheiro
  - **Artefato de Saída:** Prato de comida pronto
  - **Ferramentas:** Sistema automatizado de pedidos, sistema luminoso de aviso ao garçom indicando que o pedido está pronto

[ROCHA, 2013]

Exemplo: Processo Cozinhar (4/4)

- Defina o processo em termos de Atividades (cont.)
  - **Métodos:** receita detalhada
  - **Treinamento:** cozinheiro ter feito curso de culinária e ter feito estágio como ajudante de cozinha do chef do restaurante por no mínimo 3 meses
  - **Métrica do processo:**
    - Número de pratos devolvidos por não ser o solicitado
    - Quantidade de comida deixada no prato pelo cliente

[ROCHA, 2013]

Por que definir processos?

- Alguns benefícios:
  - Facilitar o entendimento e a comunicação entre pessoas
  - Apoiar a melhoria dos processos
  - Apoiar a gerência dos processos
  - Fornecer apoio automatizado guiando no processo
  - Fornecer apoio na execução automatizada do processo

[ROCHA, 2013]

Processo de Desenvolvimento (1/3)

- Um processo de desenvolvimento de software define:
  - Um ciclo de vida para o software e um paradigma
  - Os métodos que serão utilizados durante o desenvolvimento
  - As ferramentas que apoiarão estes métodos
  - Os papéis das pessoas envolvidas no desenvolvimento

Processo de Desenvolvimento (2/3)

- Principais etapas:
  - Análise de requisitos
    - Descrição do que o software deve fazer
  - Projeto
    - Definição de uma solução computacional
  - Codificação
    - Tradução do projeto para uma linguagem
  - Testes
    - Verificação do código
  - Manutenção
    - Correção de erros e evolução

Processo de Desenvolvimento (3/3)

- Processos de apoio:
  - Gerência de projetos
  - Medição
  - Garantia da qualidade
  - Gerência de configuração
  - Gerência de reutilização

Modelos de Ciclos de Vida

Ciclo de Vida (1/4)

– O que é?

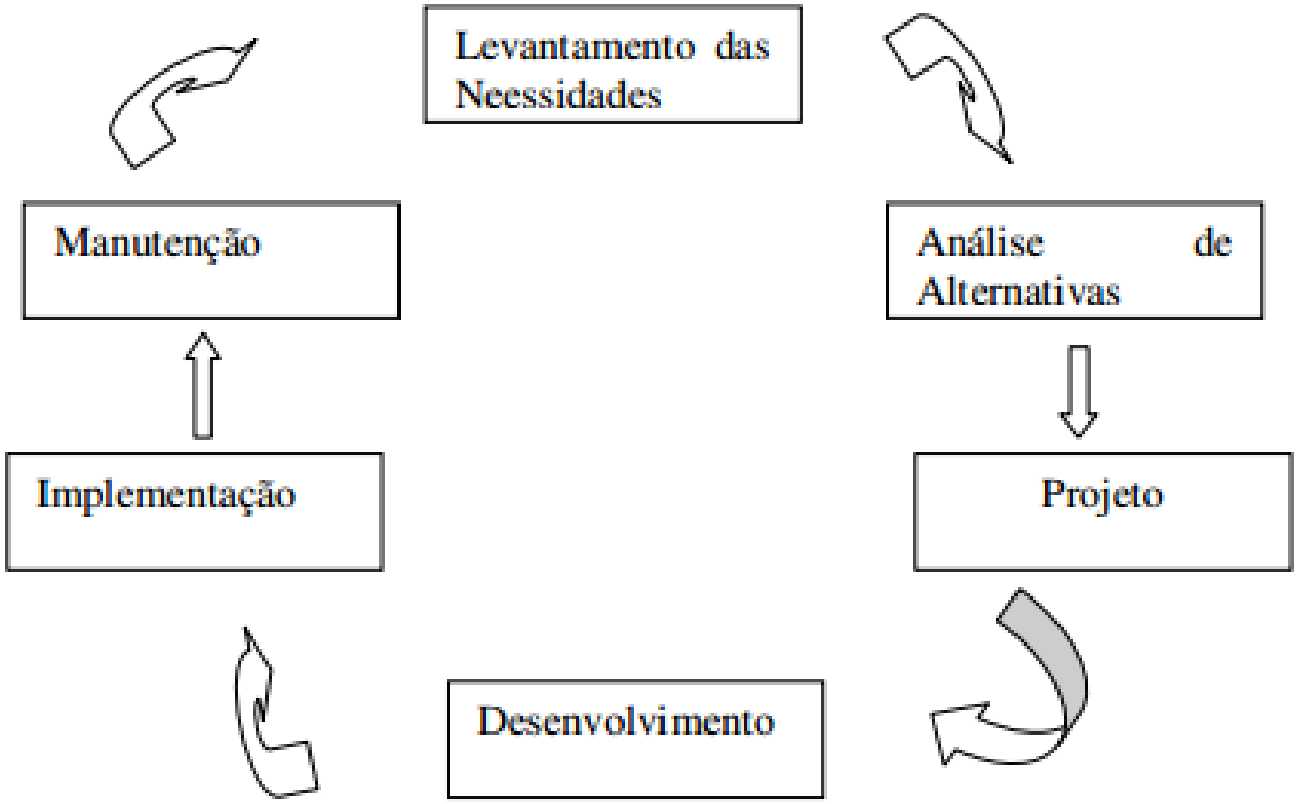
Ciclo de Vida (2/4)

- "É um processo utilizado por um analista de sistemas para desenvolver um sistema de informação"

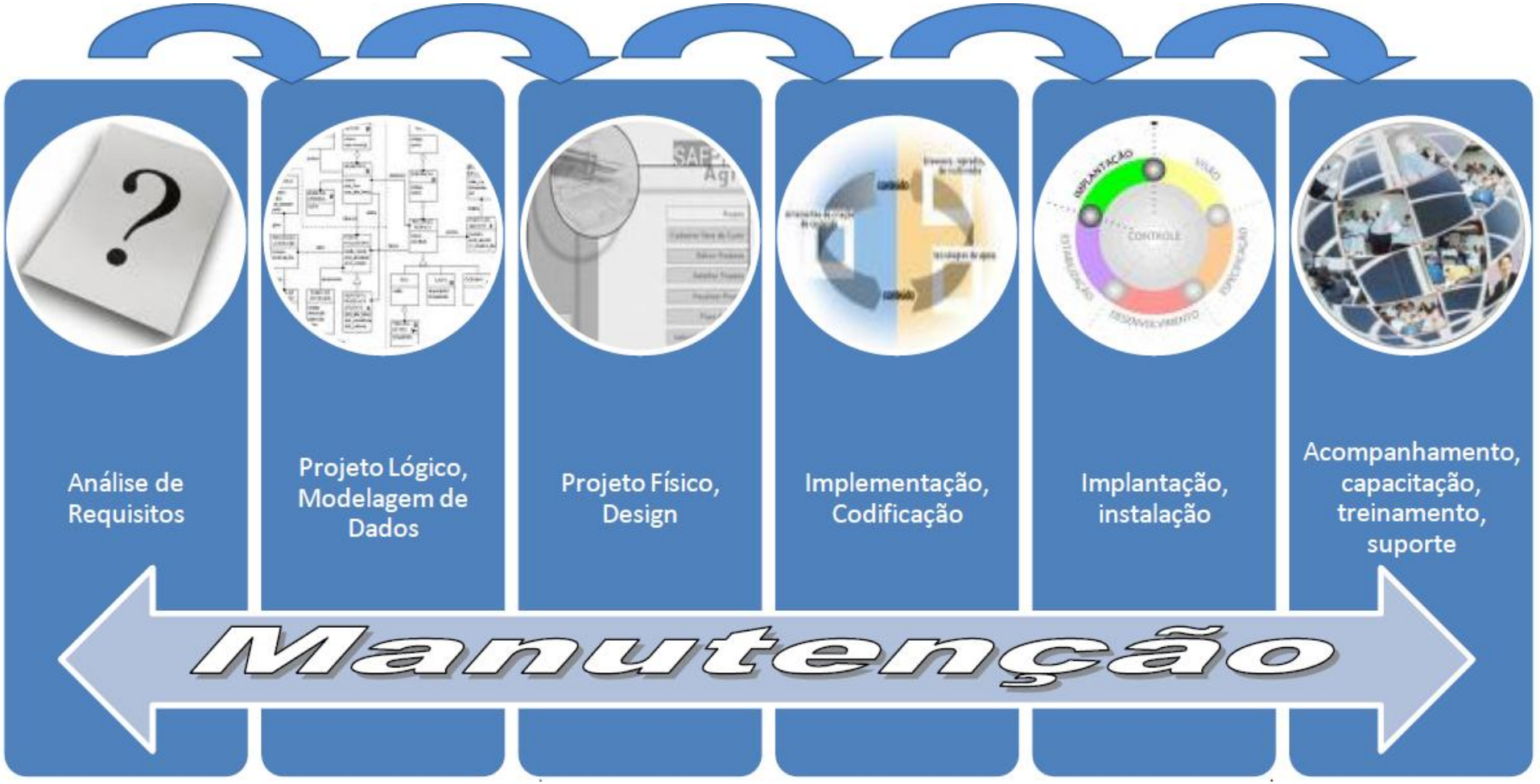
- "É um roteiro, um conjunto de passos bem definidos, que permite o uso de uma ou várias técnicas para o desenvolvimento de um sistema de informação"

- Considerados como "processos pré-fabricados"
  - Apresentam características predefinidas
  - Devem ser adaptados ao contexto real de uso (projeto, equipe, cliente)

Ciclo de Vida (3/4)

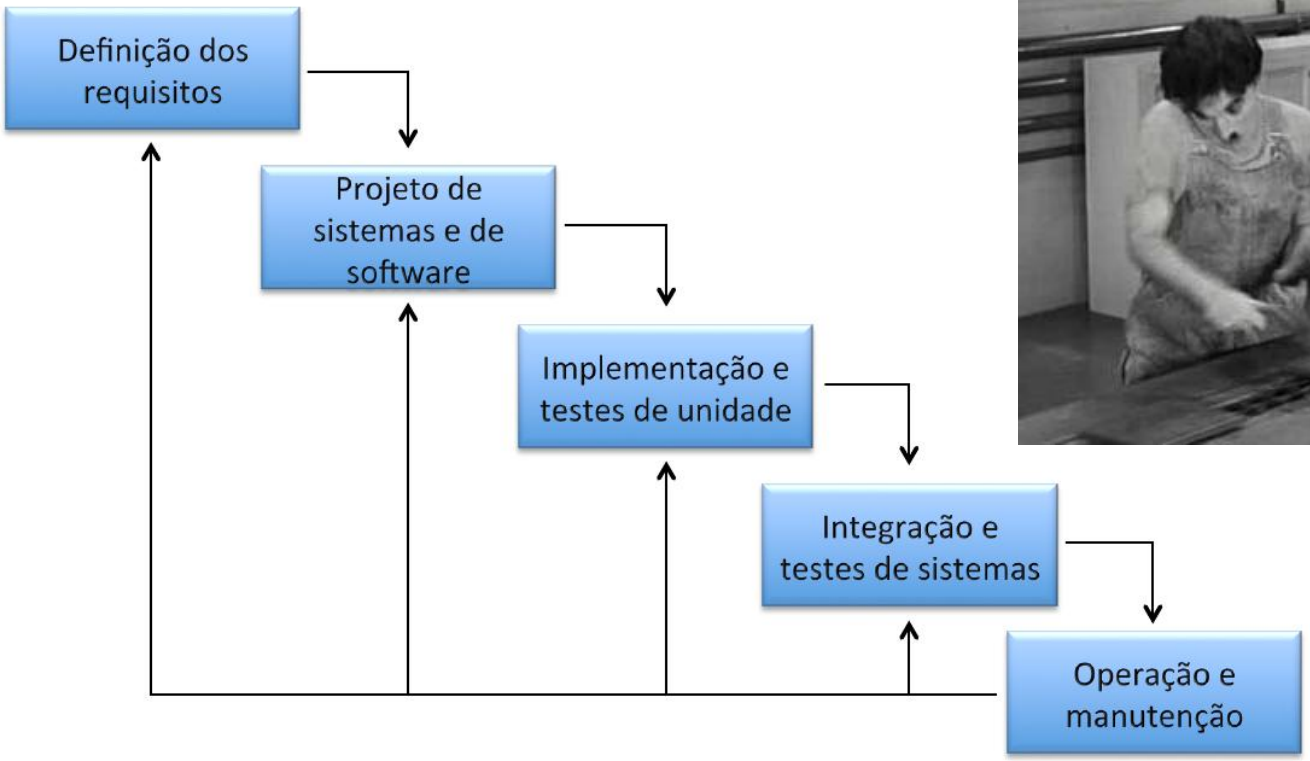


Ciclo de Vida (4/4)



[Angelo, 2009]

Modelo Cascata (1/3)



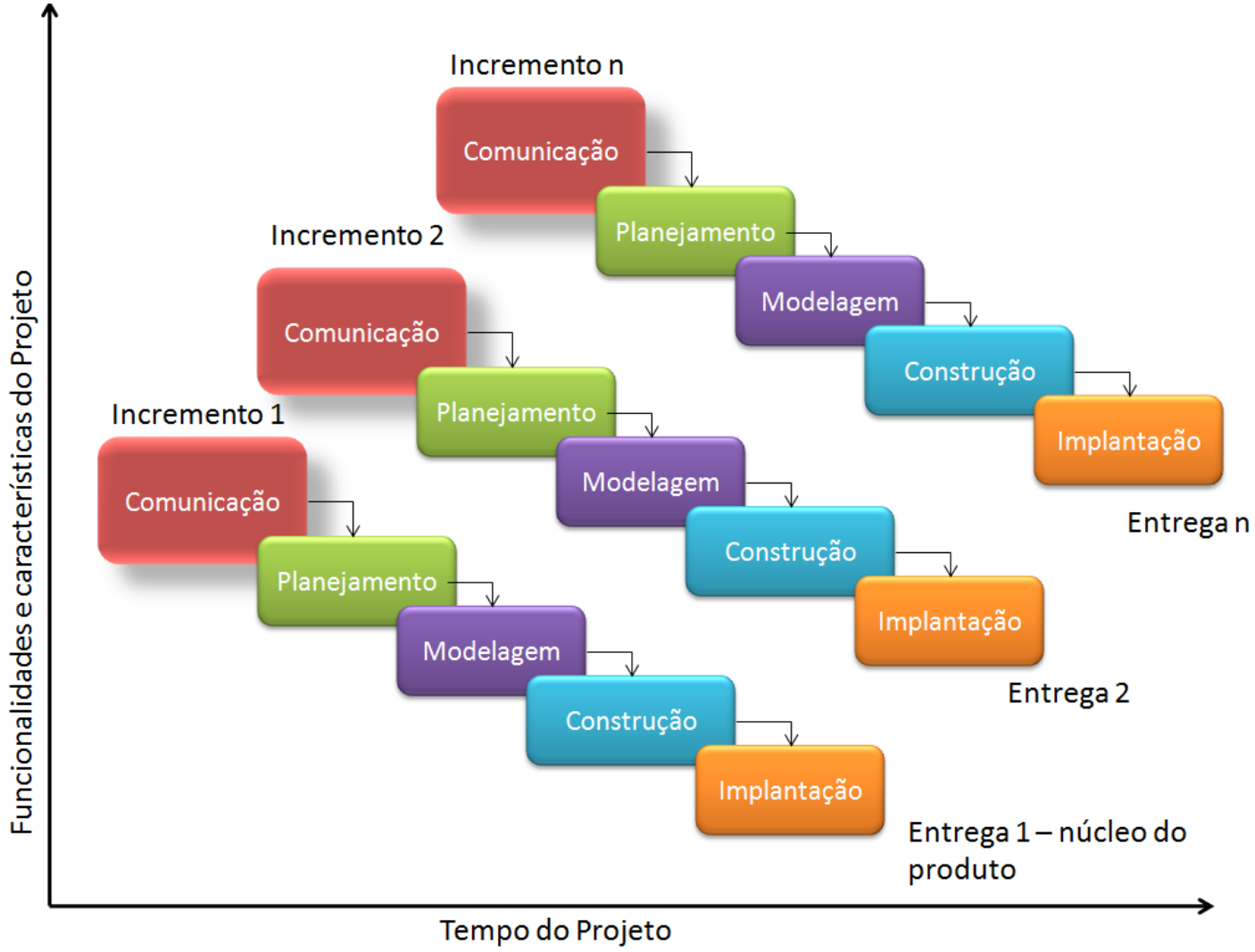
Modelo Cascata (2/3)

- Composto por uma determinada sequência de atividades
- Uma atividade começa a ser executada quando a anterior termina
- Resultado de uma etapa é utilizado na etapa seguinte
- Guiado por documentos
- Ciclo de vida mais antigo e mais utilizado

Modelo Cascata (3/3)

- Útil quando se tem requisitos estáveis e bem definidos
  - Ex.: Adicionar um novo dispositivo legal em um sistema de contabilidade
- Não lida bem com incertezas
- Fornece pouca visibilidade do estado do projeto
  - Tempo longo para a primeira entrega
  - Dificuldade na obtenção de feedback do cliente

Modelo Incremental (1/2)

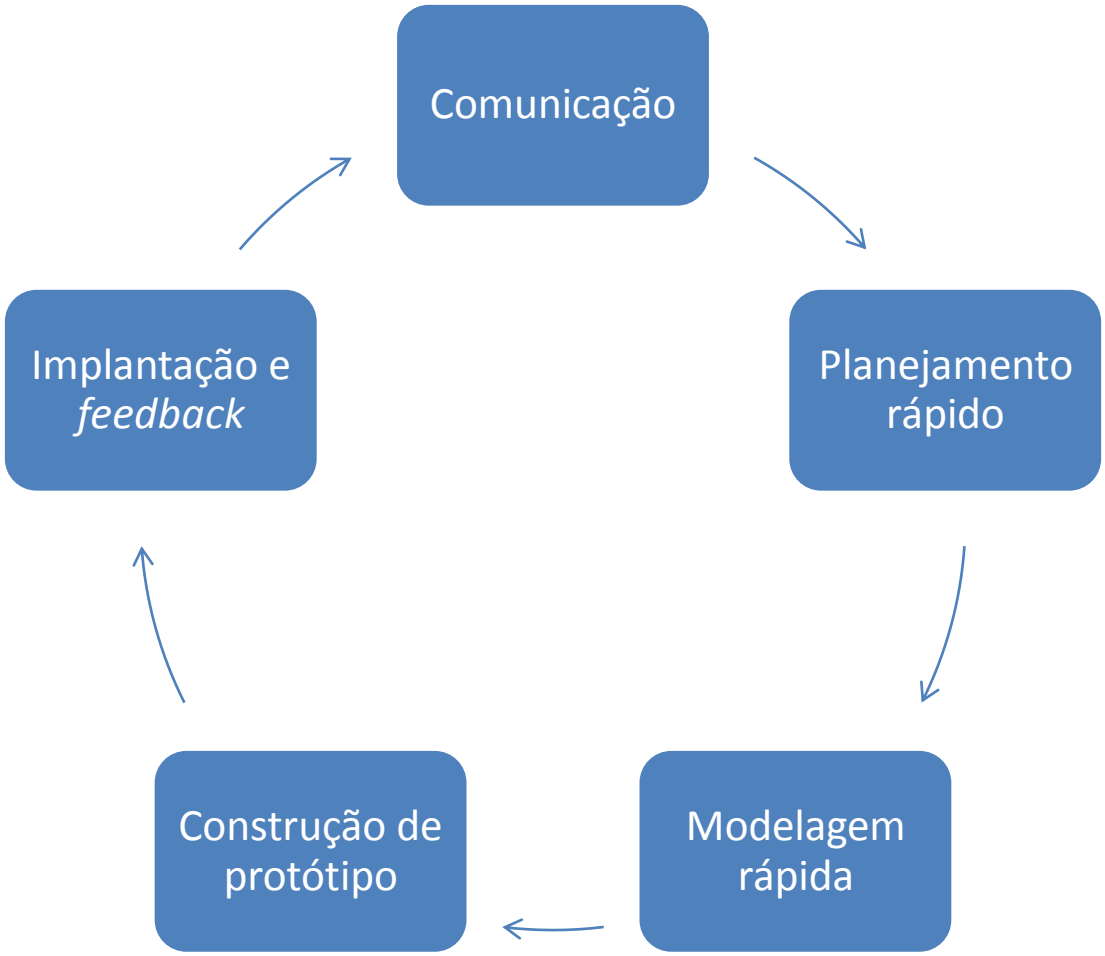


Modelo Incremental

Modelo Incremental (2/2)

- Faz entregas incrementais do software
  - Cada incremento é construído via um mini-cascata
  - Cada incremento é um software operacional
- Versões anteriores ajudam a refinar o plano
  - Feedback constante do cliente
- Diminuição da ansiedade do cliente
  - O cliente rapidamente recebe uma versão funcional do software

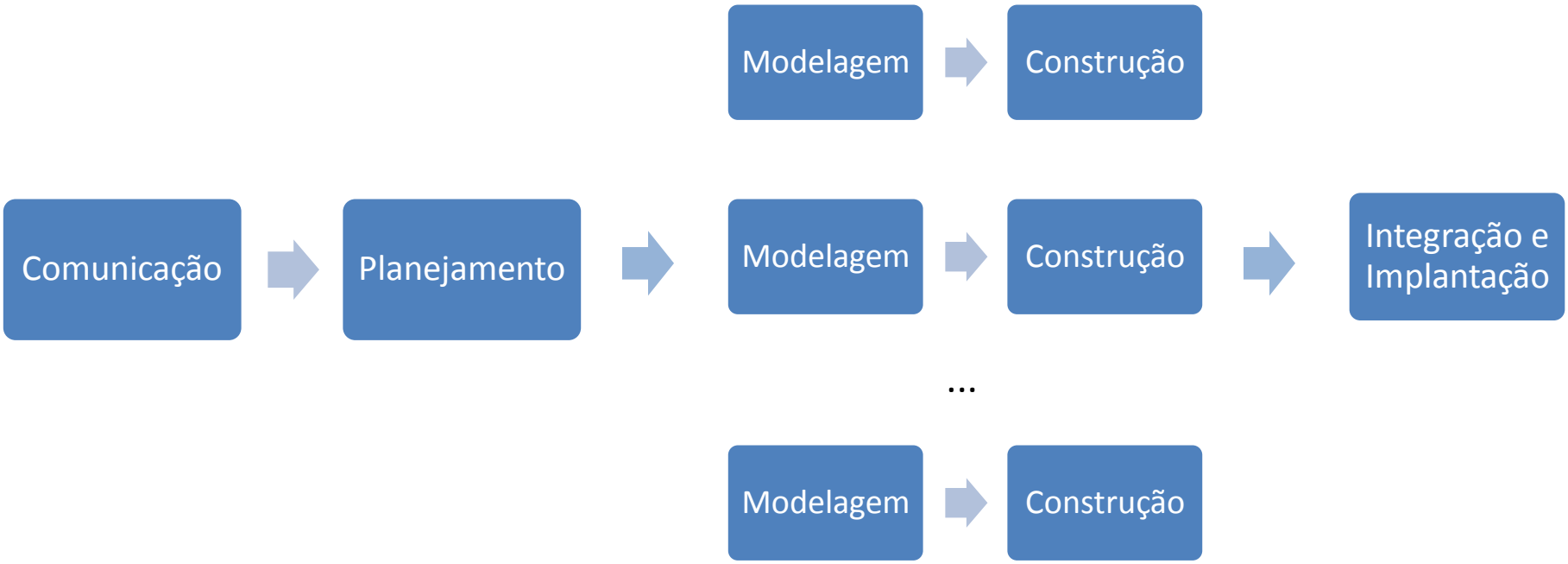
Prototipação (1/2)



Prototipação (2/2)

- Usualmente utilizado como auxílio a outro modelo de ciclo de vida
- Útil para
  - Validar um requisito obscuro com o cliente
  - Verificar o desempenho de um algoritmo específico
- Deveria ser descartado (“jogado fora”) no final
  - Protótipos não são produtos
  - Apesar disto, alguns clientes tendem a desejar colocar protótipos em ambiente de produção

RAD (*Rapid Application Development*) (1/2)



RAD (*Rapid Application Development*) (2/2)

- Funcionamento equivalente ao cascata
- Principais diferenças
  - Visa entregar o sistema completo em 60 a 90 dias
  - Múltiplas equipes trabalham em paralelo nas etapas de modelagem e construção
  - Assume a existência de componentes reutilizáveis e geração de código
- Difícil de ser utilizado em domínios novos ou instáveis

Espiral (1/2)


Planejamento
(análise de riscos)
Modelagem
Construção
Implantação
Comunicação

Espiral (2/2)

- Foco principal no gerenciamento de riscos
- A cada ciclo
  - O conhecimento aumenta
  - O planejamento é refinado
  - O produto gerado no ciclo anterior é evoluído (não é descartado)
- Cada ciclo evolui o sistema, mas não necessariamente entrega um software operacional
  - Modelo em papel
  - Protótipo
  - Versões do produto
  - etc.

Outros ciclos de vida

- Métodos formais
  - Uso de formalismos matemáticos
  - Alto nível de complexidade
  - Usualmente aplicado somente no desenvolvimento de softwares críticos
- Processo Unificado (RUP – *Rational Unified Process*)
  - Tentativa de obter o que há de melhor em cada modelo de ciclo de vida (iterativo + evolutivo)
  - Fases: Concepção, Elaboração, Construção e Transição

Critérios para selecionar ciclo de vida

- Relacionados à equipe e aos usuários
  - Experiência dos usuários no domínio da aplicação
  - Facilidade de expressão dos usuários
  - Experiência da equipe no domínio da aplicação
  - Disponibilidade de recursos para a equipe
  - Grau de acesso aos usuários
- Relacionados ao problema
  - Grau de maturidade do domínio da aplicação
  - Complexidade do problema
  - Frequência e complexidade das mudanças nos requisitos

Norma ISO/IEC 12207 (1/2)

- Norma que define um framework para processos de ciclo de vida com terminologia bem definida
- Contém processos, atividades e tarefas que devem ser aplicadas durante a aquisição de sistemas que contém software, durante o fornecimento, desenvolvimento, operação e manutenção de produtos de software

Norma ISO/IEC 12207 (2/2)

- Descreve os processos do ciclo de vida de software mas não especifica os detalhes de como implementar ou realizar as atividades e tarefas dos processos

- Não prescreve:
  - nome, formato e conteúdo da documentação
  - um modelo específico de ciclo de vida
  - um método de desenvolvimento de software

PROCESSOS NO CONTEXTO DO SISTEMA

Processos de Acordo

- Aquisição
- Fornecimento

Processos Organizacionais de Apoio aos Projetos

- Ger. do Modelo de Ciclo de Vida
- Ger. da Infra-estrutura
- Ger. do Portfolio de Projetos
- Ger. de Recursos Humanos
- Ger. da Qualidade

Processos do Projeto

- Planejamento do Projeto
- Avaliação e Controle do Projeto
- Ger. de Decisão
- Ger. de Riscos
- Ger. de Configuração
- Ger. de Informações
- Medição

Processos Técnicos

- Def. de Requisitos
- Análise de Req. do Sistema
- Projeto da Arq. do Sistema
- Implementação
- Integração do Sistema
- Teste de Qualif. do Sistema
- Instalação do Software
- Apoio à Aceitação do Software
- Operação do Software
- Manutenção do Softwre
- Descarte do Software

PROCESSOS ESPECÍFICOS DE SOFTWARE

Processos de Implementação de Sw

- Implementação do Software
- Análise de Requisitos do Sw
- Projeto da Arq. de Software
- Projeto Detalhado do Software
- Construção do Software
- Integração do Software
- Teste de Qualif. do Software

Processos de Apoio de Sw

- Ger. da Docum. do Software
- Ger. de Config. do Software
- Garantia da Qualidade do Sw
- Verificação do Software
- Validação do Software
- Revisão do Software
- Auditoria do Software
- Resolução de Problemas do Sw

Processos de Reutilização de Software

- Engenharia de Domínio
- Ger. do Programa de Reutilização
- Ger. de Ativos Reutilizáveis

Exercício individual (1/2)

- Para cada cenário a seguir, indique que ciclo(s) de vida pode(m)/deve(m) ser utilizado(s) e justifique.

- Escreva NO MÁXIMO cinco (5) linhas por cenário.

Exercício individual (2/2)

- Cenários:
  1. Jair é um cliente ansioso que precisa de um *website* seguro para vender seus produtos.
  2. Regina precisa PRA ONTEM de um *website* para promover eventos culturais de sua cidade, e sabe exatamente o que quer do *site*.
  3. Raul quer disponibilizar, a cada quinzena, um jogo de cartas diferente em seu *website* de entretenimento.
  4. Reginaldo quer uma aplicação *desktop* para controle das atividades de seus funcionários, mas ainda não sabe exatamente de que controle / nível de controle precisa.

Referências

- Slides Engenharia de Software – Professor Leonardo Murta
- Slides Engenharia de Software – Professor Marcelo Schots
- Slides Processo de Software – Professora Ana Regina Rocha
- Pressman, R.S.; “Engenharia de Software”; 6ª edição, Ed. McGraw-Hill, 2006
- Slides Introdução à Engenharia de Softaware, Professor Márcio Barros

Obrigada!