



Fatec

Araras
Antonio Brambilla

ENGENHARIA DE SOFTWARE II

CURSO: DESENVOLVIMENTO DE SOFTWARE MULTIPLATAFORMA

PROFESSOR: BRUNO HENRIQUE DE PAULA FERREIRA

ROTEIRO

- Modelos de Processos de Software
 - Espiral e Prototipação
- Criação de grupos de PI
 - Exercício em grupo

Modelos de processos de software



Modelos de processos de Software (Sommerville)

- Um processo de software é um conjunto de atividades relacionadas que levam à produção de um sistema de software.
- Existem muitos tipos diferentes de sistemas de software e não há um método universal de engenharia de software que seja aplicável a todos
- O processo utilizado nas diferentes empresas depende do tipo de software que está sendo desenvolvido, dos requisitos do cliente e das habilidades das pessoas que o desenvolvem.

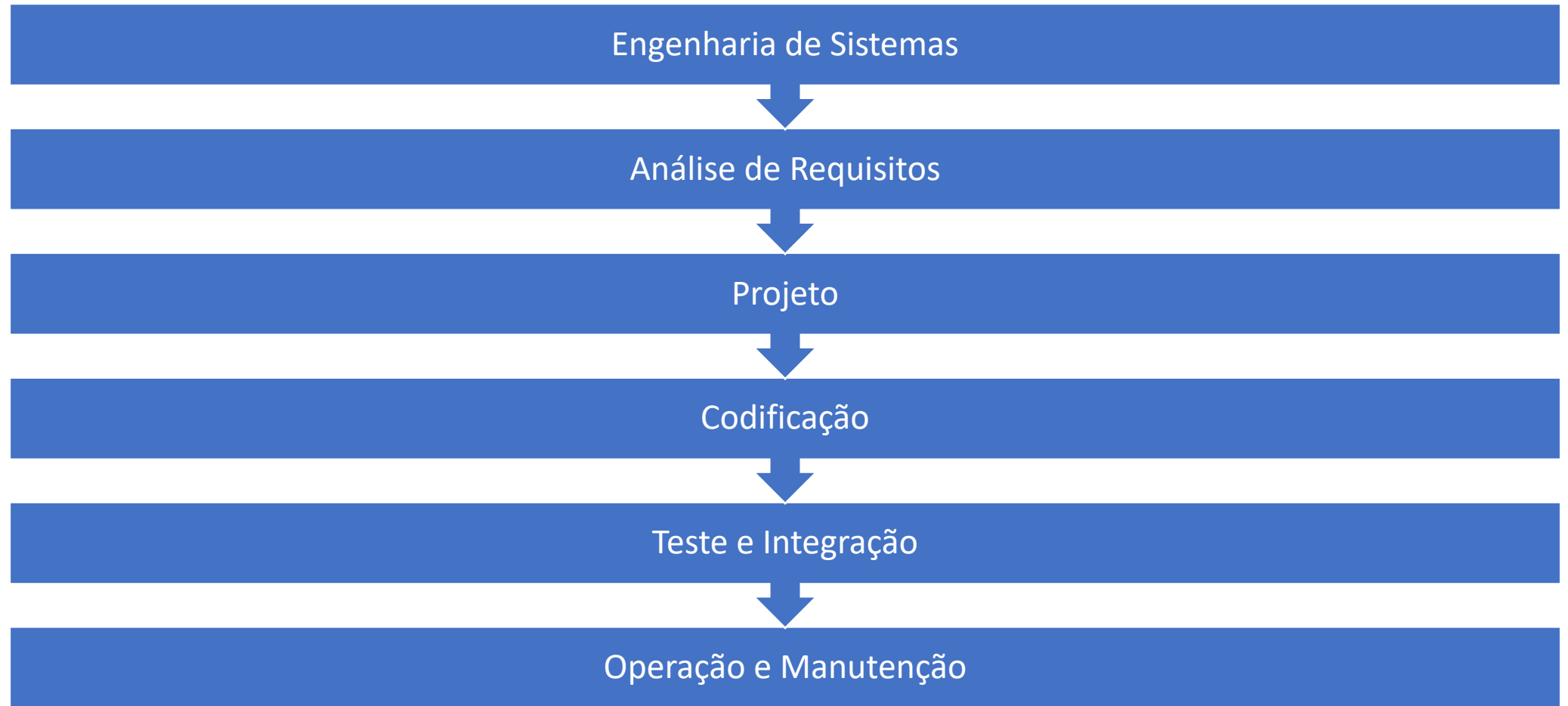
Alguns Modelos existentes

- **Modelo Cascata**
- **Incremental**
- **Prototipagem**
- **Espiral**
- **Modelos Ágeis**

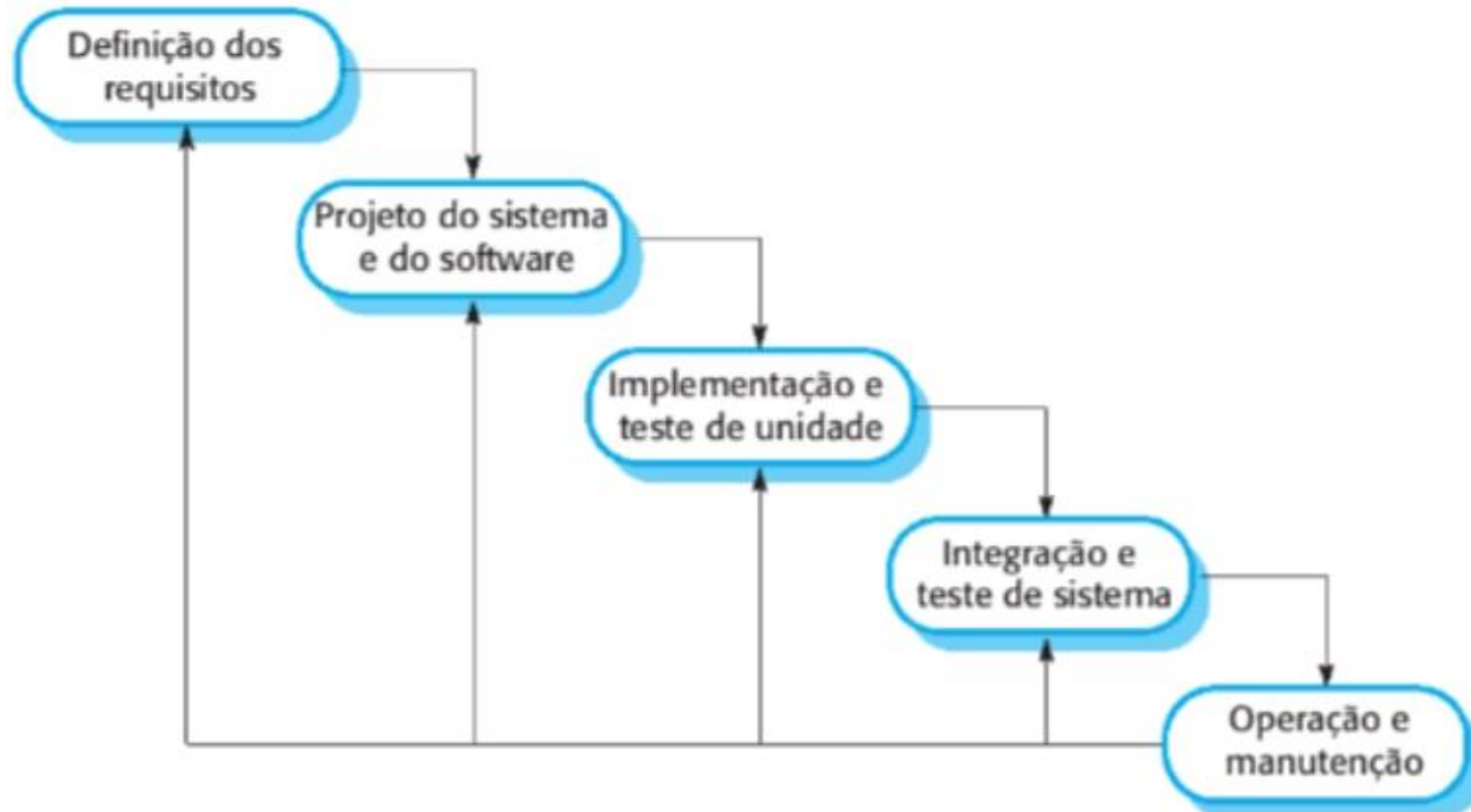
Modelos de processos de Software – Quatro Atividades Fundamentais

- 1. Especificação. A funcionalidade do software e as restrições sobre sua operação devem ser definidas.
- 2. Desenvolvimento. O software deve ser produzido para atender à especificação.
- 3. Validação. O software deve ser validado para garantir que atenda ao que o cliente deseja.
- 4. Evolução. O software deve evoluir para atender às mudanças nas necessidades dos clientes.

Modelo Cascata



Modelo Cascata



Modelo Cascata

- **Análise e definição dos requisitos.** Os serviços, as restrições e as metas do sistema são estabelecidos por meio de consulta aos usuários. Depois, eles são definidos em detalhes e servem como uma especificação do sistema.
- **Projeto do sistema e do software.** O processo de projeto do sistema reparte os requisitos entre requisitos de sistemas de hardware e de software, e estabelece uma arquitetura global do sistema
- **Implementação e teste de unidade.** Durante essa etapa, o projeto do software é realizado como um conjunto de programas ou unidades de programa. O teste de unidade envolve a verificação de cada unidade, conferindo se satisfazem a sua especificação.

Modelo Cascata

- **Integração e teste de sistema.** As unidades de programa ou os programas são integrados e testados como um sistema completo a fim de garantir que os requisitos de software tenham sido cumpridos. Após os testes, o sistema de software é entregue ao cliente.
- **Operação e manutenção.** Normalmente, essa é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em uso. A manutenção envolve corrigir os erros que não foram descobertos nas primeiras fases do ciclo de vida, melhorar a implementação das unidades do sistema e aperfeiçoar os serviços do sistema à medida que novos requisitos são descobertos.

PROTOTIPAÇÃO

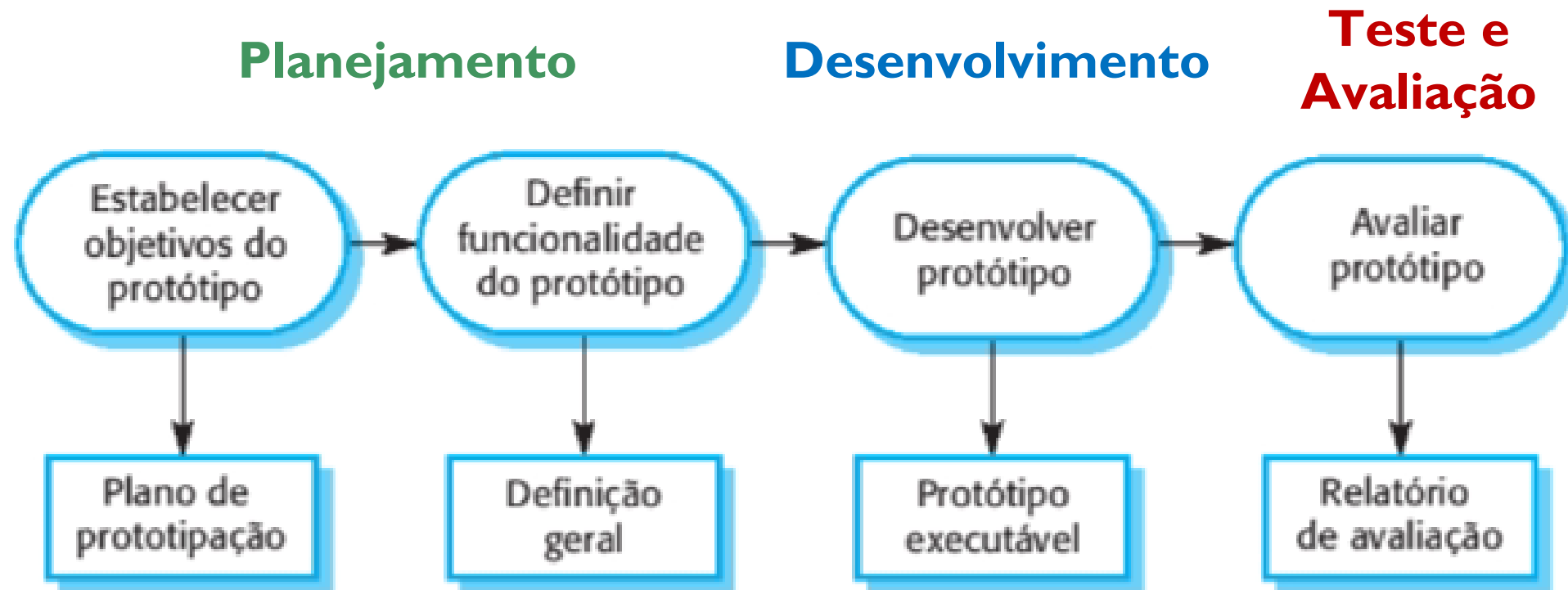
O protótipo é uma versão inicial de um sistema utilizado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções.

O desenvolvimento rápido e iterativo do protótipo é essencial para que os custos sejam controlados e os stakeholders do sistema possam experimentar o protótipo no início do processo de desenvolvimento do software.

PROTOTIPAÇÃO

- Um protótipo de software pode ser utilizado em um processo de desenvolvimento para ajudar a antecipar as mudanças que podem ser necessárias:
 1. No processo de engenharia de requisitos, um protótipo pode ajudar na elicitación e validação dos requisitos do sistema.
 2. No processo de projeto do sistema, um protótipo pode ser utilizado para explorar soluções de software e no desenvolvimento de uma interface com o usuário para o sistema.

PROTOTIPAÇÃO – Desenvolvimento de um Protótipo



SOMMERVILLE

Prototipação



PROTOTIPAÇÃO

- Permite que usuários potenciais observem até que ponto o sistema os ajuda.
 - Pontos fortes e fracos no software
 - Novos requisitos
 - Revelar erros e omissões

OBJETIVOS

- **Validação de Requisitos:** Os protótipos ajudam a esclarecer e validar os requisitos do software com os stakeholders.
- **Comunicação Efetiva:** Permitem uma comunicação mais eficaz entre a equipe de desenvolvimento e os usuários finais.
- **Teste de Conceito:** Permitem testar a viabilidade de ideias e conceitos antes de investir recursos no desenvolvimento completo.

TIPOS DE PROTÓTIPOS

- **Protótipos de Papel:** Desenhos ou esboços de telas de software.
- **Protótipos de Alta Fidelidade:** Protótipos interativos que se assemelham muito à aparência e ao comportamento do produto final.
- **Protótipos Funcionais:** Incluem funcionalidades limitadas, mas reais, para demonstrar a operação do software.

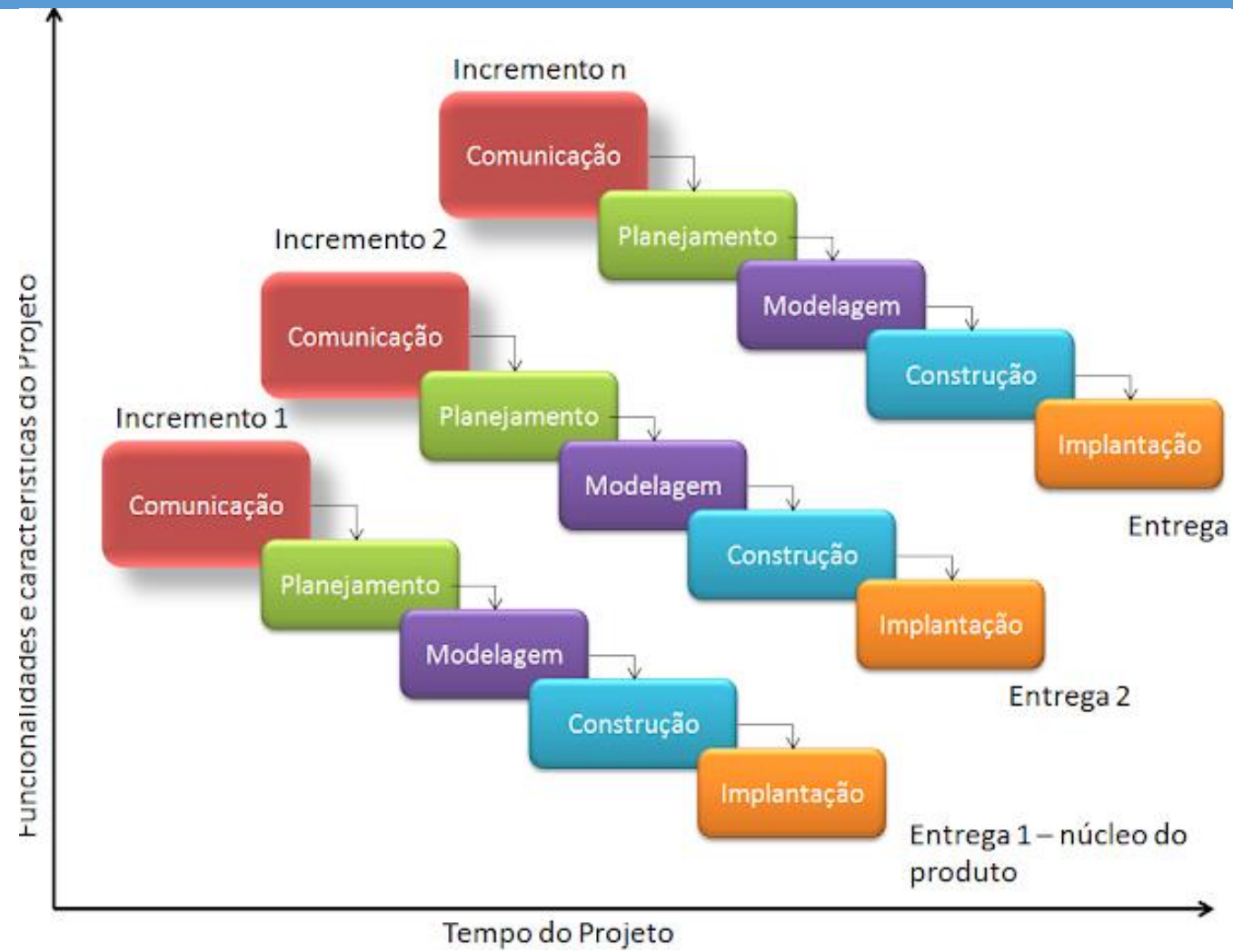
VANTAGENS

- **Redução de Riscos:** Ajuda a identificar problemas e lacunas nos requisitos antes do desenvolvimento completo.
- **Melhoria da Comunicação:** Facilita a compreensão das necessidades do usuário.
- **Economia de Tempo e Dinheiro:** Evita retrabalho caro no desenvolvimento.

DESVANTAGENS

- **Custo e Tempo Inicial:** A criação de protótipos pode consumir tempo e recursos iniciais.
- **Expectativas Irreais:** Os usuários podem confundir protótipos com produtos finais.

Modelo Incremental

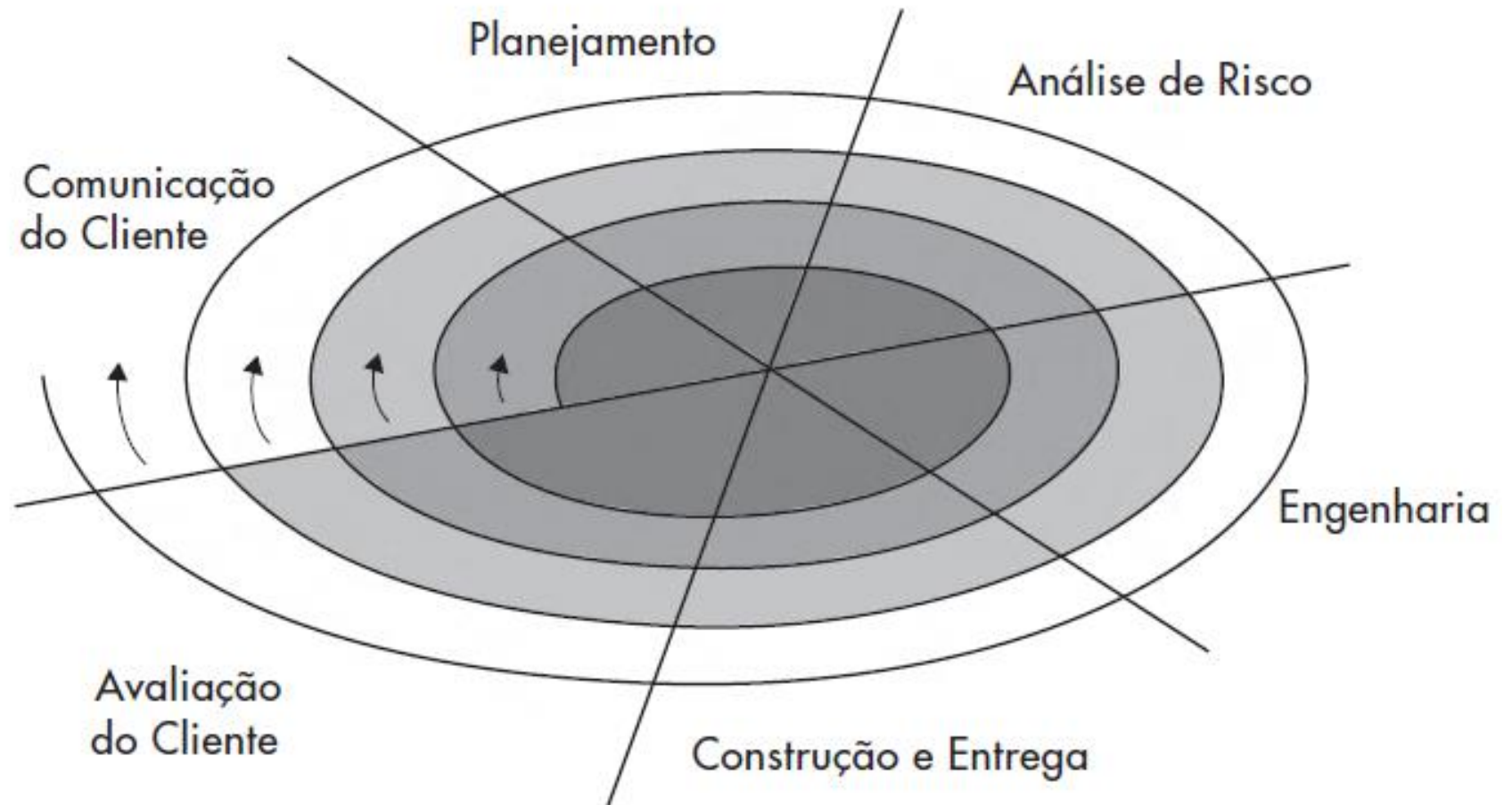


Modelo Incremental

- O desenvolvimento é dividido em etapas, denominadas “incrementos”, que produzirão incrementalmente o sistema, até a sua versão final.
- Em cada incremento é realizado todo o ciclo do desenvolvimento de software, do planejamento aos testes do sistema já em funcionamento. Cada etapa produz um sistema totalmente funcional, apesar de ainda não cobrir todos os requisitos.

PROCESSOS DE SOFTWARE - ESPIRAL

- Proposto por Barry Boehm, em 1988. As atividades do processo Espiral evoluem do centro para fora, no sentido horário.



PROCESSO ESPIRAL - CARACTERÍSTICAS

- O modelo Espiral da Engenharia de Software, proposto no ano de 1988, foi desenvolvido combinando as melhores práticas dos modelos de Ciclo de Vida Clássico e Prototipação, acrescentando ao mesmo tempo, um elemento que falta a esses paradigmas: a **Análise dos Riscos**.
- Usando o Modelo Espiral, o software é desenvolvido em uma série de versões evolucionárias. Durante as primeiras iterações, as versões podem ser um modelo de papel ou protótipo. Durante as últimas iterações, são produzidas versões cada vez mais completas do sistema submetido à engenharia.

PROCESSO ESPIRAL - CARACTERÍSTICAS

- Durante a evolução do projeto, são seguidos estes passos:
- Determinação dos objetivos, alternativas e restrições;
- Análise de risco e prototipação;
- Validação e verificação;
- Planejamento da fase seguinte.
- Esta concepção tende a criar um roteiro de atividades e etapas, para que se alcance uma maturidade do processo evolutivo de desenvolvimento de sistemas complexos. Ao final obtêm-se um produto em sua forma mais completa possível.

PROCESSO ESPIRAL - VANTAGENS

- Estimativas mais realísticas, pois problemas são descobertos mais rapidamente.
- Adaptabilidade, pois permite gerenciar mudanças que podem acontecer em qualquer etapa do projeto;
- Com o feedback do cliente, pode-se desenvolver um software que atenda cada vez mais as suas necessidades.

PROCESSO ESPIRAL - DESVANTAGENS

- Pode ser difícil convencer os clientes que esta abordagem é adequada (principalmente em questões contratuais);
- Se um risco importante não for descoberto ou não for gerenciado, fatalmente ocorrerão problemas.
- O modelo em espiral, por suas características de avaliação e planejamento baseadas em risco, exige que se tenha gerentes e técnicos experientes.

Características de um bom software

Características do produto	Descrição
Manutenibilidade	O software deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes. Esse é um atributo crítico, porque a mudança de software é um requisito inevitável de um ambiente de negócio em mudança.
Confiança e proteção	A confiança do software inclui uma série de características como confiabilidade, proteção e segurança. Um software confiável não deve causar prejuízos físicos ou econômicos no caso de falha de sistema. Usuários maliciosos não devem ser capazes de acessar ou prejudicar o sistema.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, eficiência inclui capacidade de resposta, tempo de processamento, uso de memória etc.
Aceitabilidade	O software deve ser aceitável para o tipo de usuário para o qual foi projetado. Isso significa que deve ser compreensível, usável e compatível com outros sistemas usados por ele.

BIBLIOGRAFIA

- SOMMERVILLE , Ian Engenharia de software; tradução Luiz Cláudio Quiroz: revisão técnica Fábio Levy Siqueira - 10 ed - São Paulo: Pearson Education do Brasil. 2018.
- SBROCCO, José Henrique Teixeira de Carvalho Metodologias ágeis: engenharia de software sob medida / José Henrique Teixeira de Carvalho Sbrocco, Paulo Cesar de Macedo. -- 1. ed. -- São Paulo: Érica, 2012.
- Pressman, Roger S. Engenharia de software: uma abordagem profissional



OBRIGADO