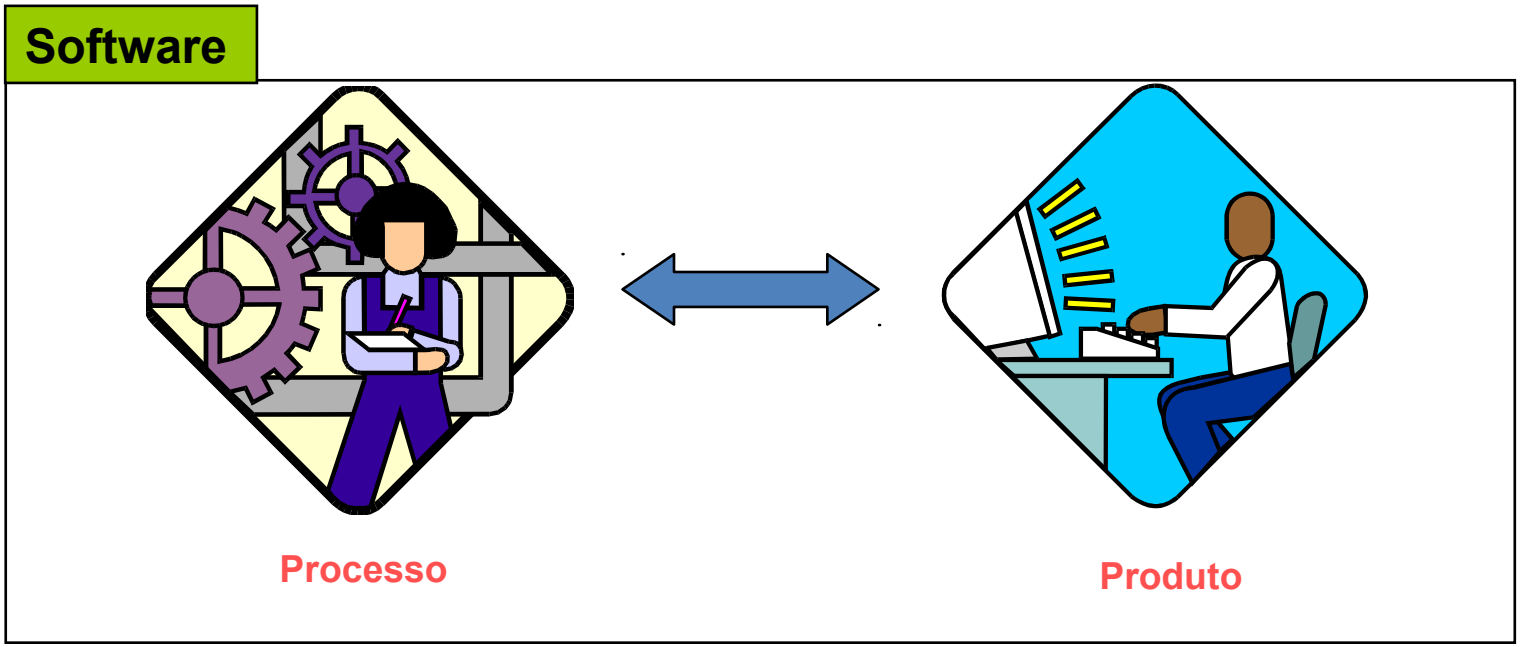
Engenharia de Software
Ciclo de vida e processo

Profa. Flávia B. Blum Haddad
Email: flaviahaddad@utfpr.edu.br

Software



Engenharia de Software

- Elementos
  - Modelos do ciclo de vida do software
  - Linguagens
  - Métodos
  - Ferramentas
  - Processos
- Atividades
  - Modelagem do negócio
  - Elicitação de requisitos
  - Análise e Projeto
  - Implementação
  - Testes
  - Planejamento
  - Gerenciamento
  - Manutenção

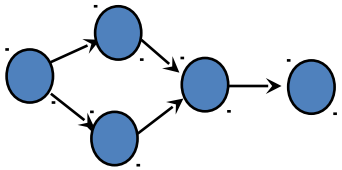
Processo de Construção de Software

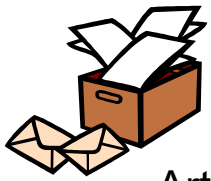
Equipe

Ferramentas

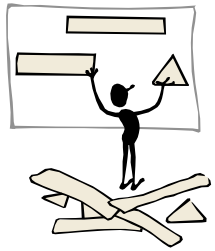
Métodos

Passos/Atividades





Artefatos



Insumos

Processo

Processo de Software

- Um conjunto estruturado de atividades necessárias para desenvolver um sistema de software;
- Um conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software.

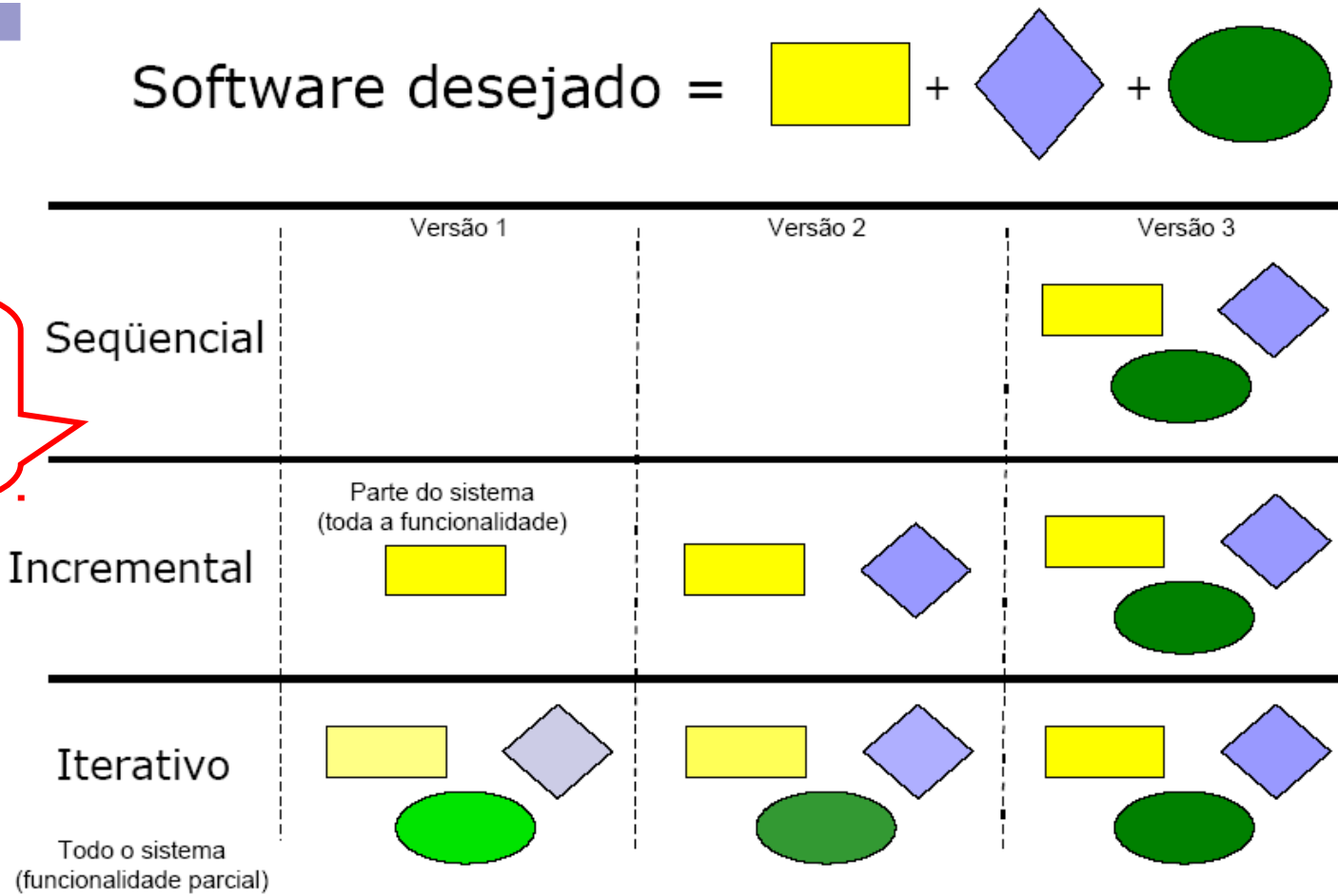
  - Concepção/Especificação
  - Análise e Design
  - Implementação
  - Testes/Validação
  - Manutenção/Evolução

Processo de Software - elementos

Processo de Software
- **Processos**
  - **Atividades**
    - Pré-atividades
    - Subatividades
    - **Artefatos**
      - Insumos
      - Produtos
    - **Recursos**
      - Recursos Humanos
      - Ferramentas de Software
      - Hardware
    - **Procedimentos**
      - Métodos
      - Técnicas
      - Roteiros

Modelo de Processo (ciclo de vida e processos de software)

- Um modelo de processo de software deve ser escolhido com base:
  - Na natureza do projeto e da aplicação;
  - Nos métodos e ferramentas a serem utilizados;
  - Nos controles e produtos que precisam ser entregues.

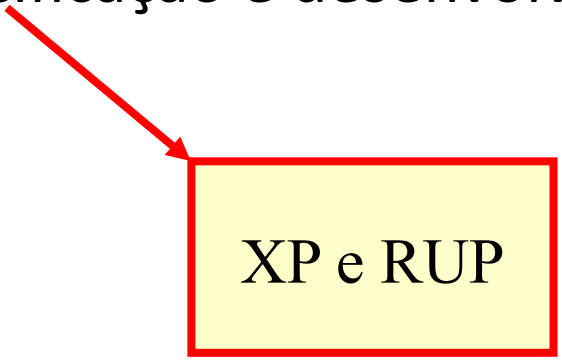


3 abordagens principais

Ciclo de vida de software

- O modelo cascata (Clássico)
  - Fases separadas e distintas de especificação e desenvolvimento.

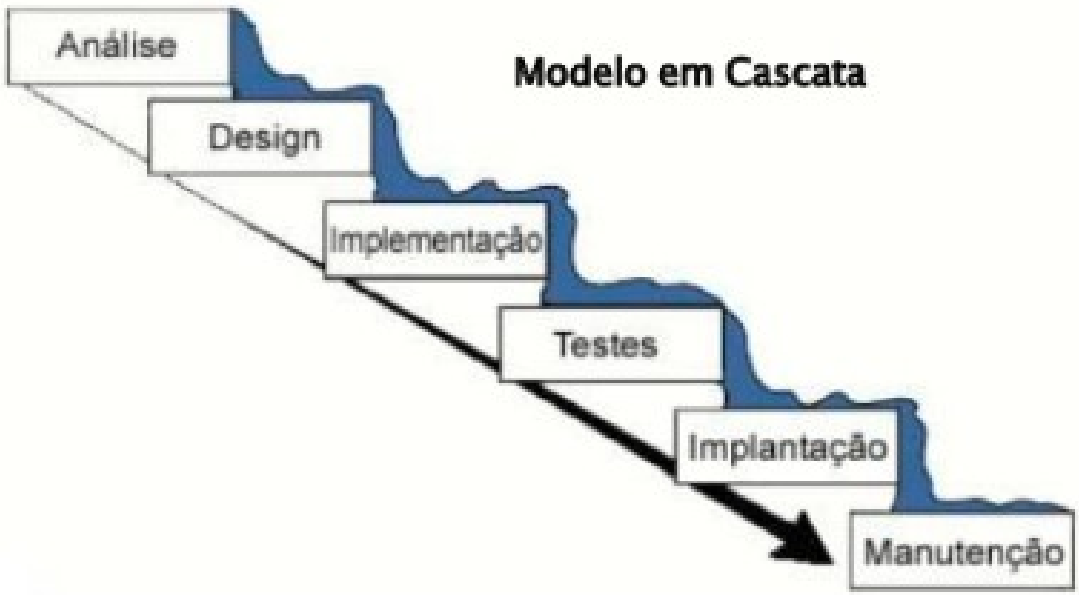
- Prototipação e Espiral
  - Especificação e desenvolvimento são intercalados.



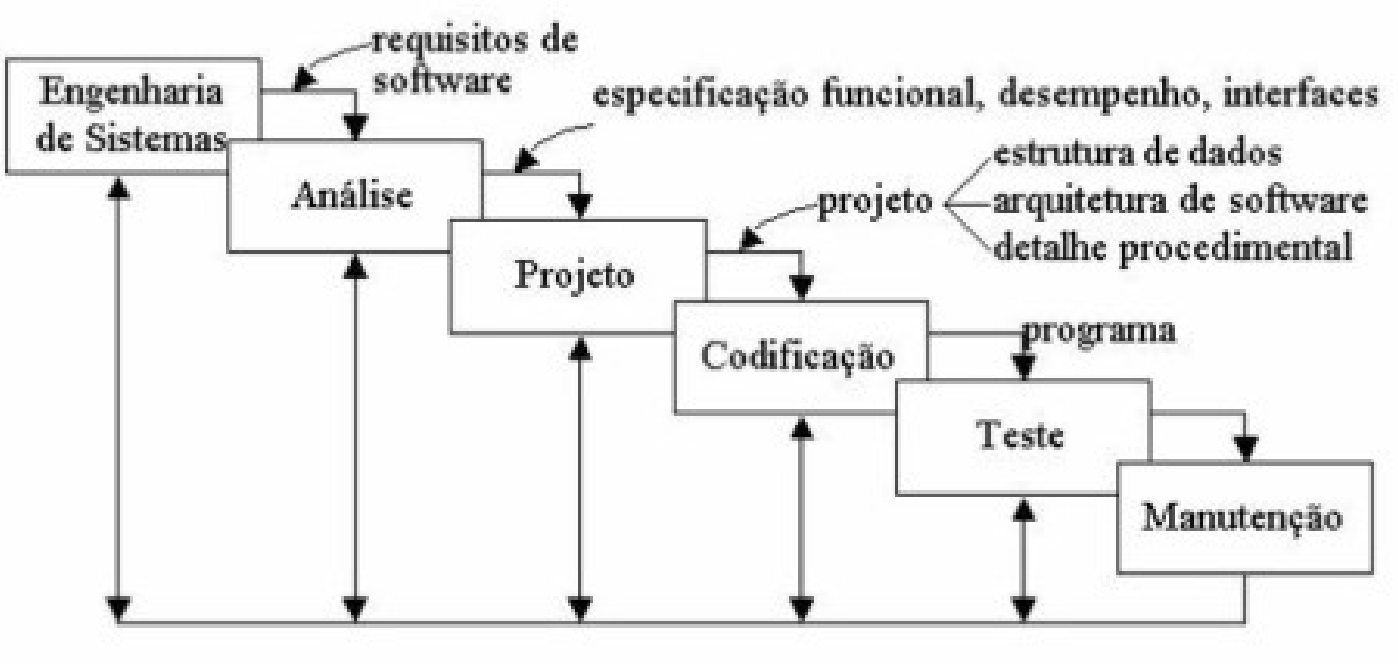
Modelo em Cascata

- Um dos primeiros modelos (Royce, 1970).
- O desenvolvimento de um estágio deve terminar antes do próximo começar.
- Derivado do mundo do hardware (linhas de montagens).

Modelo em Cascata



Modelo em Cascata



Modelo em Cascata

Problemas

- Projetos reais raramente seguem o fluxo seqüencial que ele propõe. Ocorrem iterações que trazem problemas na aplicação do paradigma.

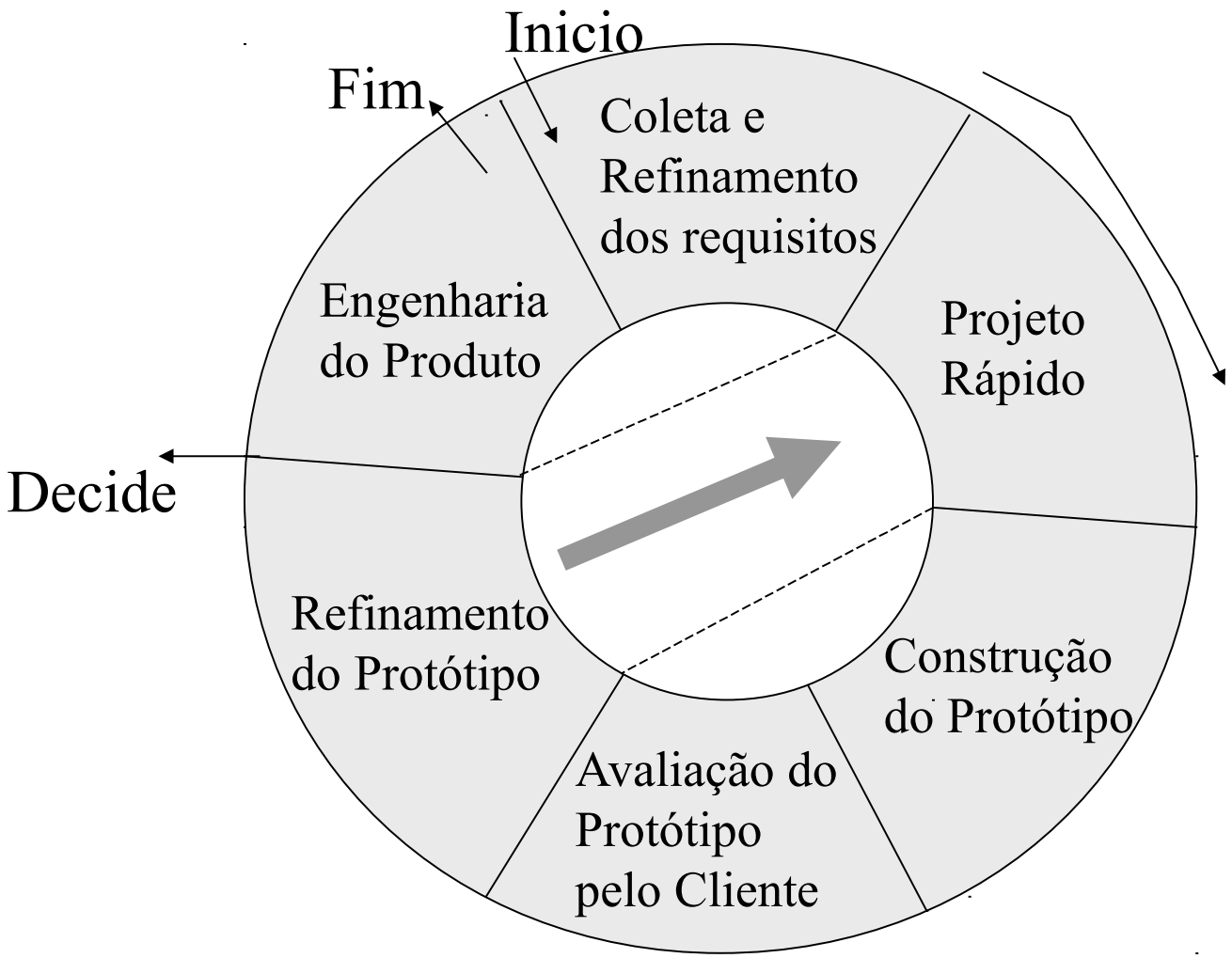
- É difícil para o cliente declarar todas as exigências explicitamente. É difícil acomodar as incertezas naturais que existem no começo de muitos projetos.

Modelo em Cascata

Problemas:

- O cliente deve ter paciência. Uma versão do software só estará disponível em um ponto tardio do cronograma. Um erro crasso, pode ser desastroso.
- Desenvolvedores Ociosos.
- Só é apropriado quando os requisitos são bem conhecidos.

Prototipação



Prototipação

APROPRIADO QUANDO

- o cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento e saída com detalhes.

Prototipação

- Permite o refinamento iterativo dos requisitos.
- A cada iteração é produzido um protótipo do software final.
- Este protótipo pode ser um:
  - **Protótipo em Papel**, primeiras versões que permitem ao usuário ter uma visão abstrata do sistema;
  - **Protótipo incompleto**, implementa algum subconjunto de funções exigidas;
  - **Protótipo final**, um software que executa parte ou toda a função desejada, mas que tem outras características que serão melhoradas e ainda não podem ser disponibilizadas.

Prototipação

- **Coleta e Refinamento dos Requisitos:**
  - Nesta etapa o desenvolvedor e o cliente devem definir os objetivos gerais do software (Protótipo)
  - Identificar quais requisitos são conhecidos e as áreas que necessitam de definição adicional
  - Análise de Sistema

- **Projeto Rápido:**
  - Representação dos aspectos do software que são visíveis ao usuário

Prototipação

- **Construção do Protótipo:**
  - Implementação rápida do Projeto.

- **Avaliação do Protótipo:**
  - Cliente e desenvolvedor avaliam o protótipo.
  - Sugestões ou mudanças serão trabalhadas na próxima fase.

Prototipação

- **Refinamento do Protótipo:**
  - São trabalhados os problemas encontrados na fase anterior. Ou seja, são refinados os requisitos.
  - Neste ponto pode ocorrer, no caso de necessidade de alterações, um retorno na fase de projeto Rápido para desenvolver um novo protótipo que incorpore as mudanças.

- **Construção do Produto:**
  - Identificados todos os requisitos necessários, o protótipo pode ser descartado e a versão final do produto deve ser construída considerando os critérios de qualidade.

Prototipação

Problemas:

- O cliente muitas vezes não aceita mais uma iteração, aquela versão mesmo incompleta já serve.
- Não há necessidade de desenvolver uma versão final, modifica-se o protótipo.
- O desenvolvedor freqüentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo.

Solução:

- Definir as regras do jogo logo no começo, o cliente deve concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos

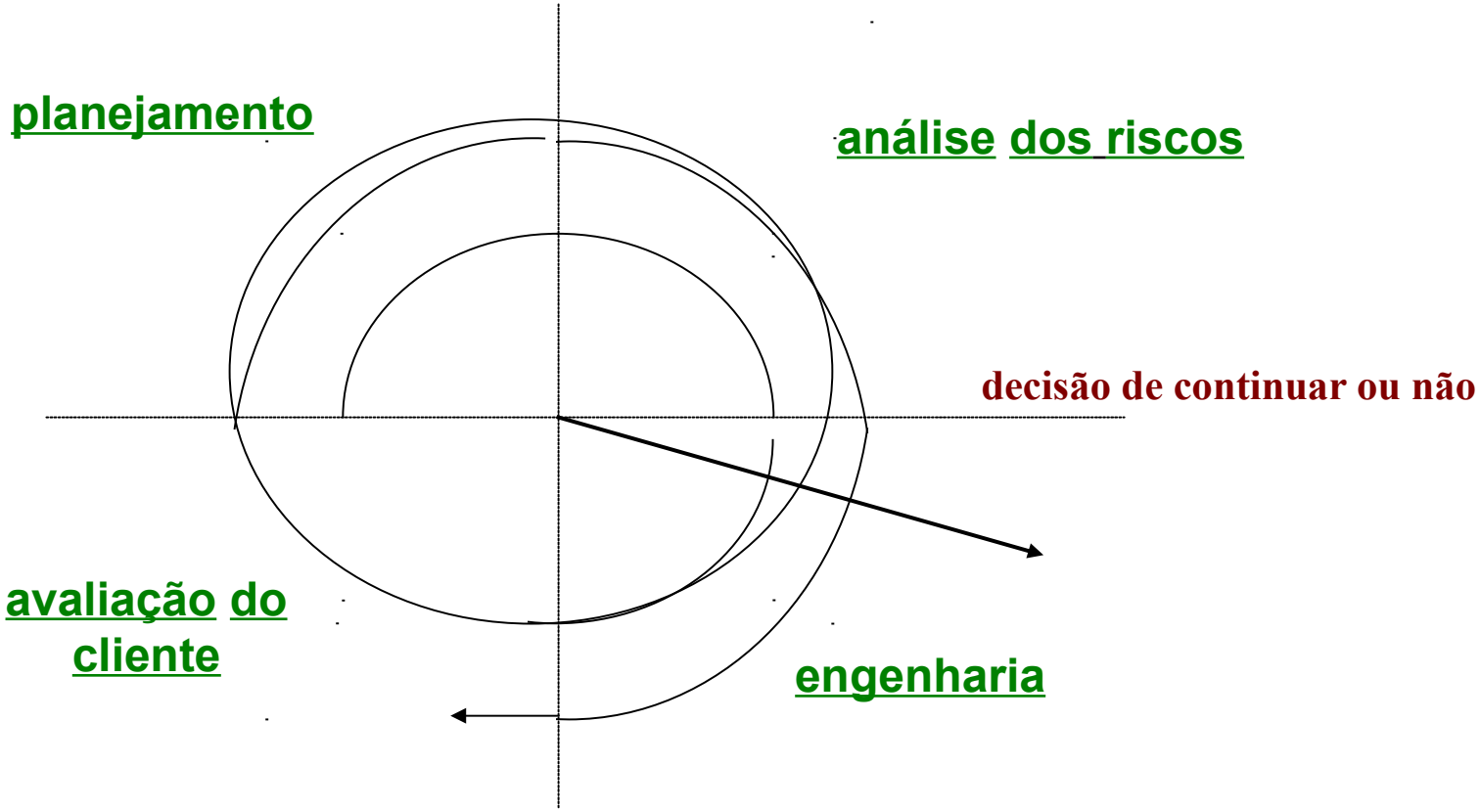
Modelo em Espiral

- O Modelo em espiral é um modelo incremental.
- Este tipo de modelo combina elementos do modelo cascata (aplicado repetidamente) com a filosofia iterativa da prototipação.
- Acrescenta mais uma atividade: Análise de Risco.
- O objetivo é trabalhar junto do usuário para descobrir seus requisitos, de maneira incremental, até que o produto final seja obtido.

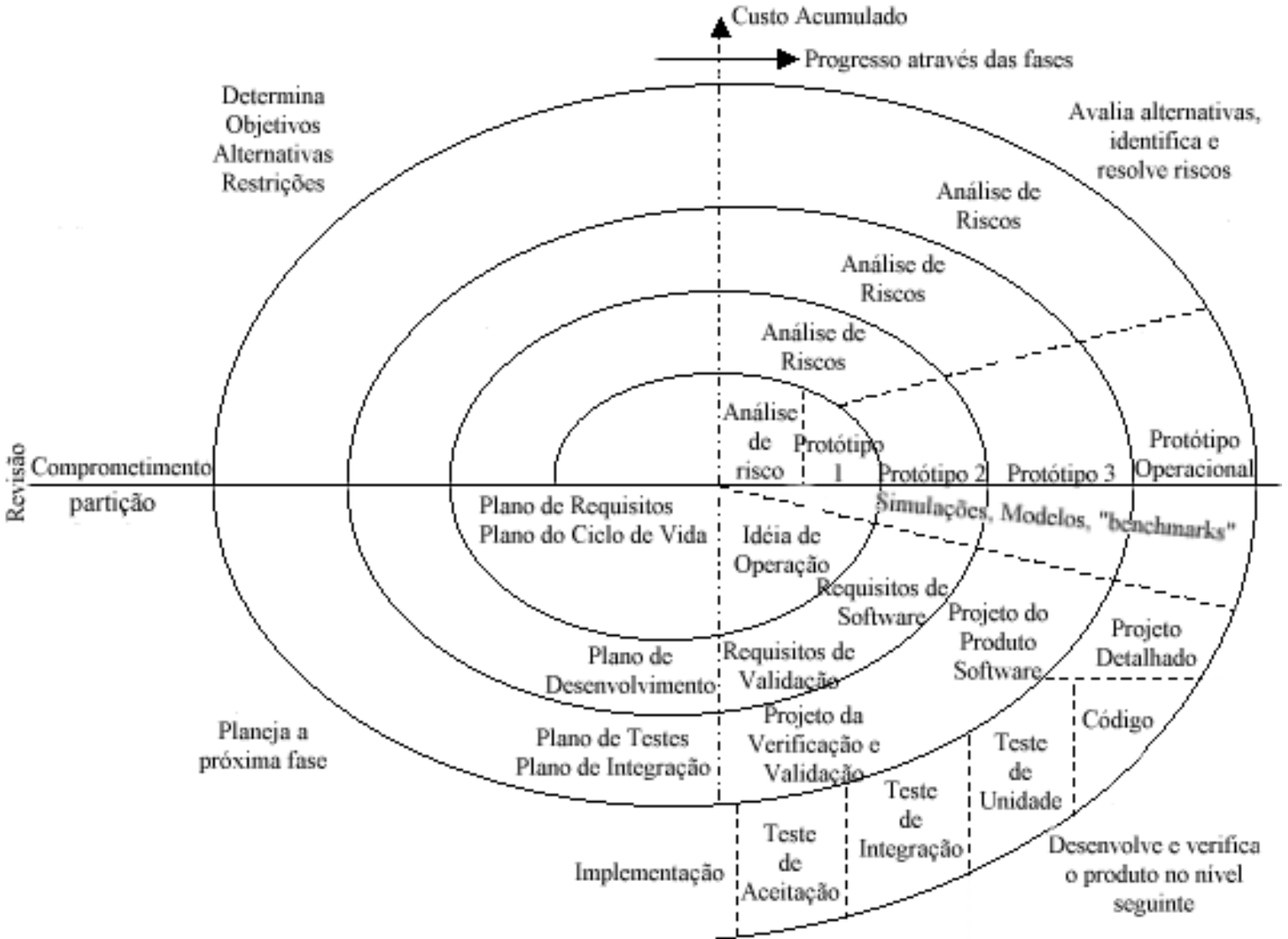
Modelo em Espiral

- O desenvolvimento começa com as partes do produto que são mais bem entendidas.
- A evolução acontece quando novas características são adicionadas à medida que são sugeridas pelo usuário.
- A cada iteração é desenvolvida uma versão usável, não um protótipo.

Modelo em Espiral



Modelo em Espiral



Modelo em Espiral

- Fornece o potencial para desenvolvimento rápido de versões incrementais do software.
- Nas primeiras iterações são desenvolvidas versões que podem ser protótipos.
- Nas iterações mais adiantadas são produzidas versões incrementais mais completas e melhoradas.

Modelo em Espiral

1- PLANEJAMENTO:

- Determinação dos objetivos, alternativas e restrições;
- Comunicação com os clientes;
- Definição de Recursos.

2- ANÁLISE DE RISCO:

- Análise das alternativas e identificação / resolução dos riscos

Modelo em Espiral

3- CONSTRUÇÃO:

- Desenvolvimento do produto no nível seguinte;
- Constrói protótipos ou versões mais avançadas do produto.
- Realiza testes, implantação, suporte.

4- AVALIAÇÃO DO CLIENTE:

- Obtém um feedBack do cliente baseado na avaliação da versão do software.
- São levantadas as necessidades de mudança para o software

Modelo em Espiral

- É uma abordagem realística para o desenvolvimento de software em grande escala;
- Usa uma abordagem que capacita o desenvolvedor e o cliente a entender e reagir aos riscos em cada etapa evolutiva;
- Exige considerável experiência na determinação de riscos e depende dessa experiência para ter sucesso.

Outros Modelos

RAD (Rapid Application Development)

- Adaptação do seqüencial (Cascata).
- Modelo de desenvolvimento de software incremental que enfatiza um ciclo de desenvolvimento bastante curto (60 a 90 dias).
- Desenvolvimento em equipe (team) e modular.
- Fases: Modelagem do Negócio, Modelagem dos Dados, Modelagem do Processo, Geração da Aplicação, Teste e Implantação.

Outros Modelos

DBC - Desenvolvimento Baseado em Componentes

- Evolução da Tecnologia OO.
- Adaptação do modelo espiral para o desenvolvimento de software.
- Modelo de Componentes: CORBA (Common Object Request Broker Architecture), COM (Component Object Model - Microsoft), UML.
- Componentes são construídos/empacotados para serem reutilizados em diferentes aplicações (Reuso).

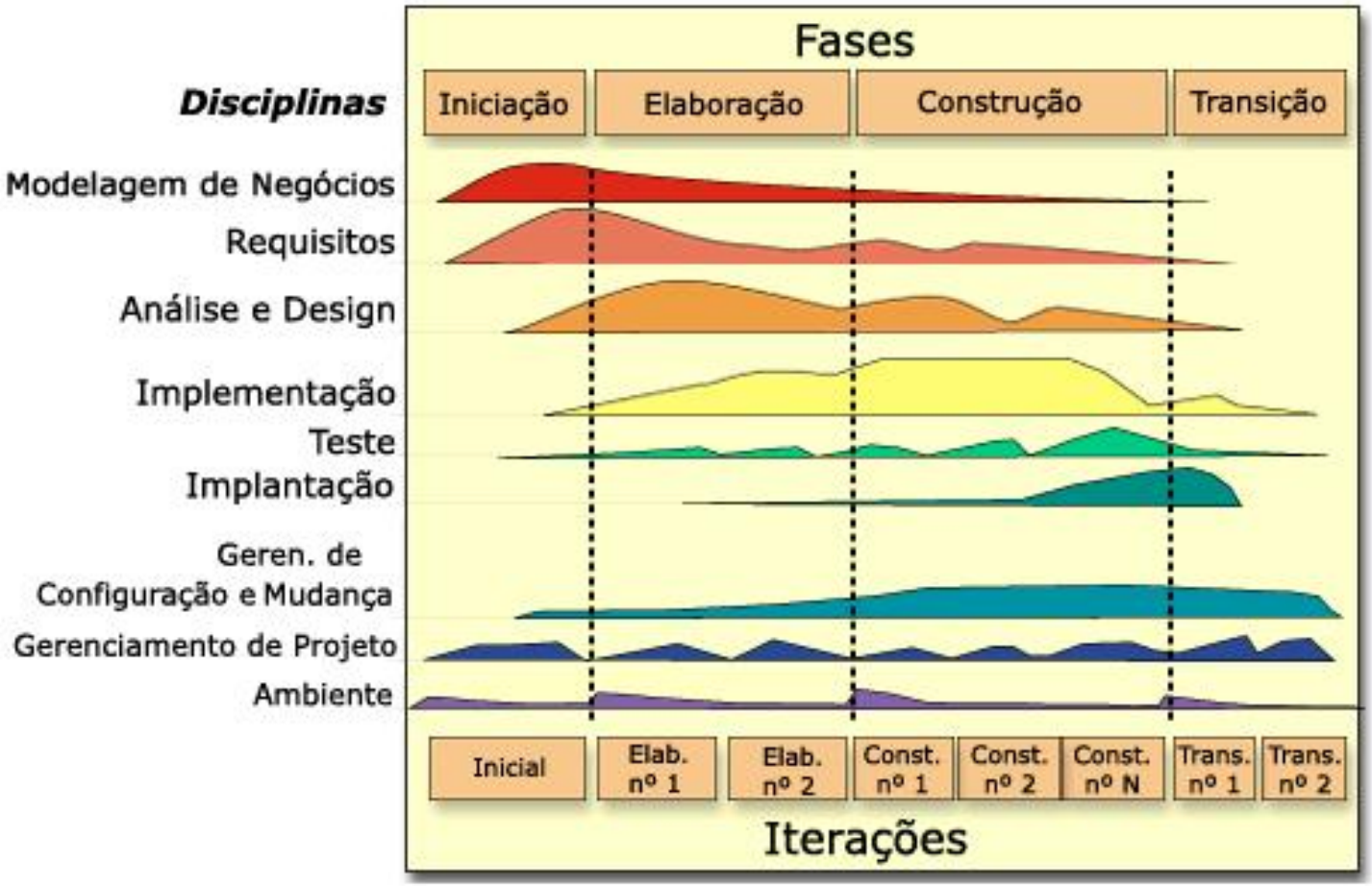
Outros Modelos

RUP(*Rational Unified Process)*

- O RUP é um processo de engenharia de software desenvolvido pela empresa Rational.
- Ele serve como um guia de como utilizar de maneira eficiente a UML (*Unified Modeling Language*).
- Utiliza desenvolvimento Iterativo e Incremental.
- Tem como objetivo oferecer um processo de desenvolvimento "bem definido" e "bem gerido".

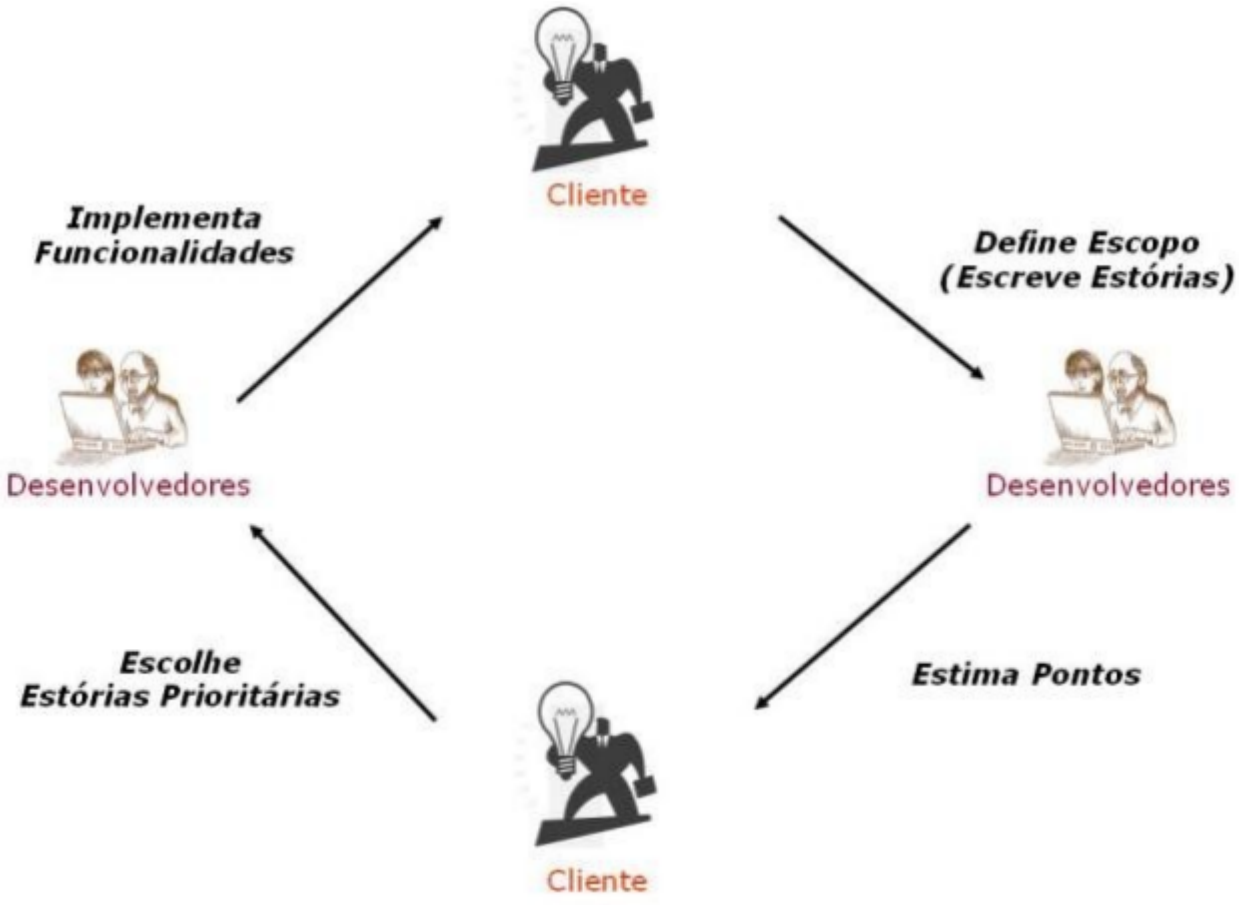
Outros Modelos

RUP(*Rational Unified Process*)



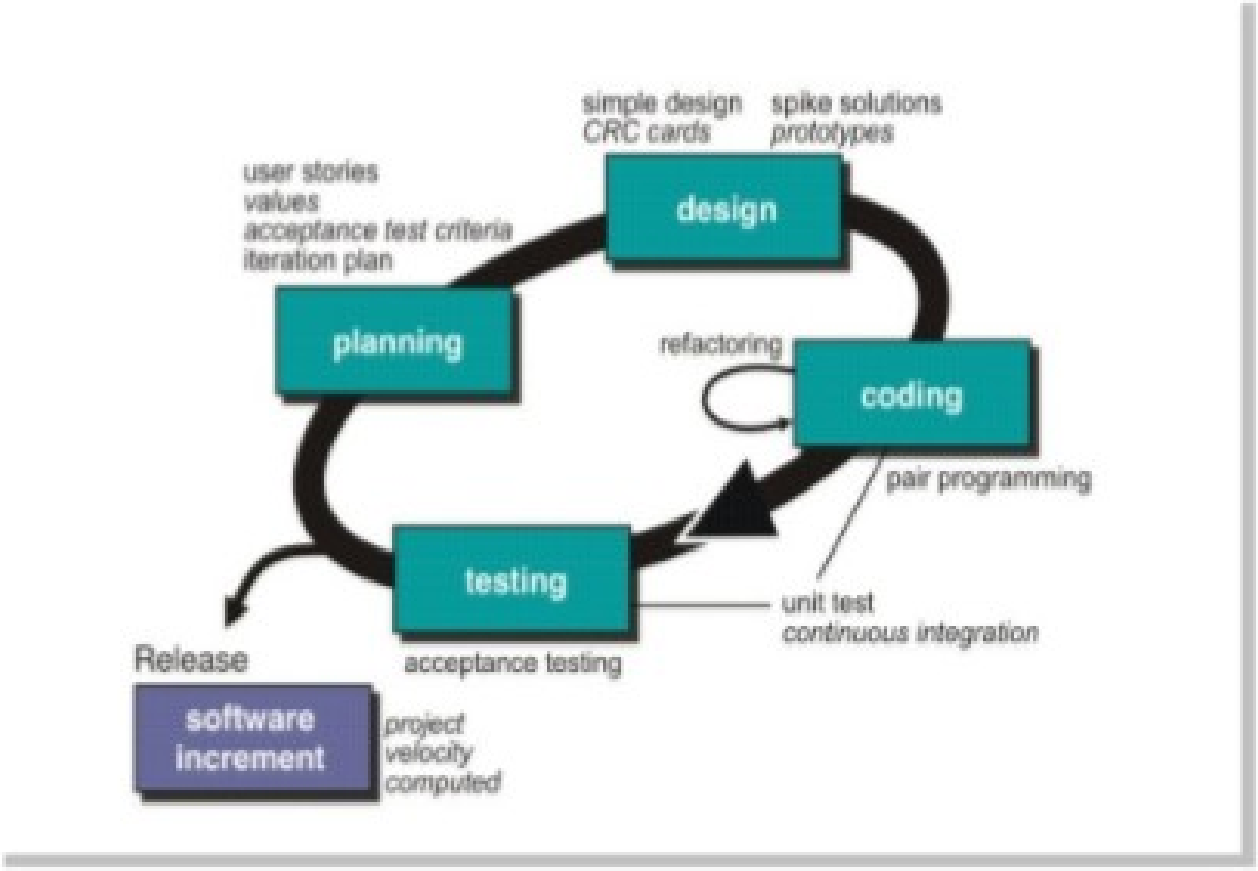
Outros Modelos

XP (Extreme Programming)



Outros Modelos

Extreme Programming (XP)



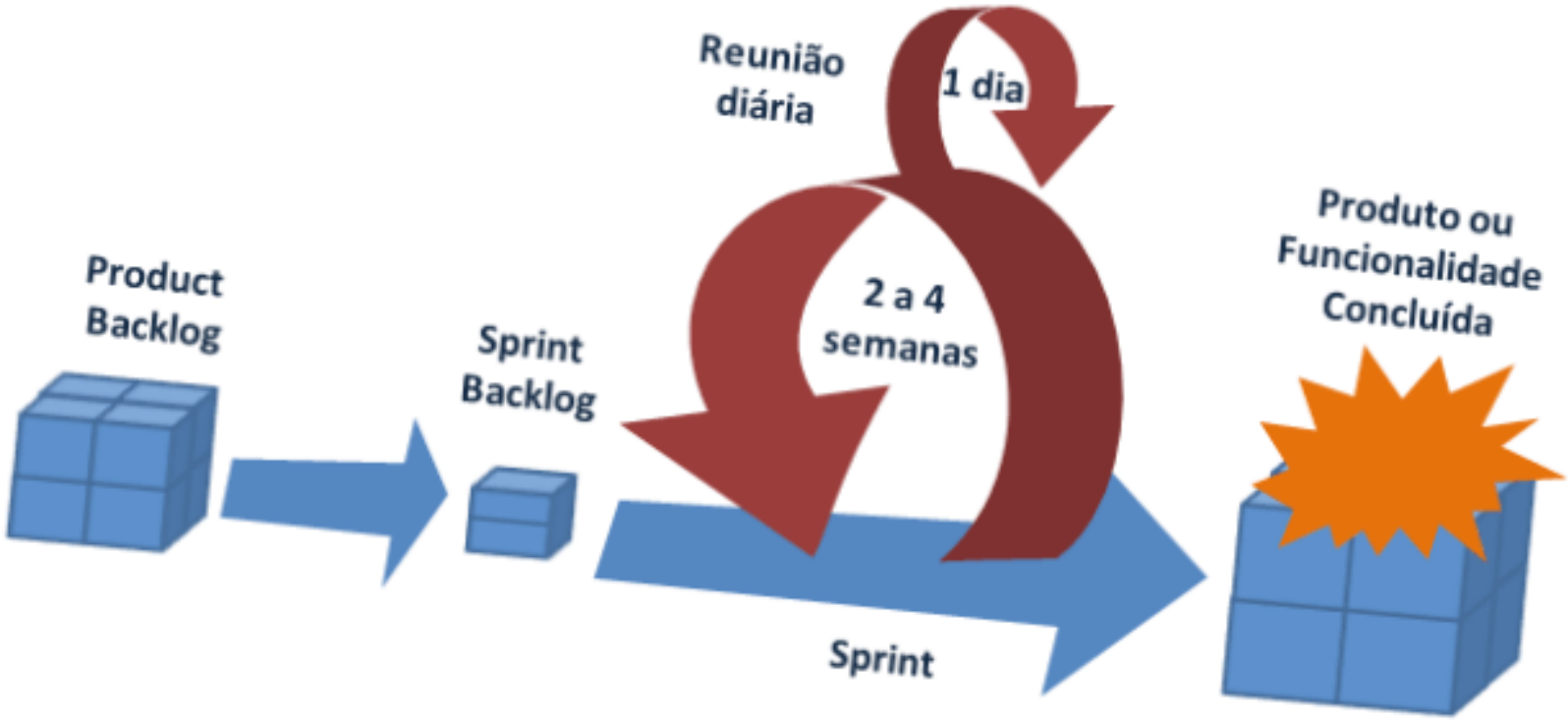
Outros Modelos

XP (Extreme Programming)

- O processo consiste basicamente num ciclo de quatro etapas, que itera até que o produto esteja pronto:
  - **Planejamento** - Coleta e negociação de requisitos com o cliente.
  - **Desenho** – Reconstrução do sistema para a incorporação de novas funcionalidades.
  - **Implementação** – Codificação do sistema de modo a atender os testes.
  - **Teste** – Elaboração de testes com base nas estórias do cliente.

Outros Modelos

SCRUM



Metodologia Ágil x Tradicional

Abordagem Tradicional	Abordagem Ágil
Preditivo: detalhar o que ainda não é bem conhecido	**Adaptativo**: conhecer problema e resolver o crítico primeiro
Rígido: seguir especificação predefinida, a qualquer custo	**Flexível**: adaptar-se a requisitos atuais, que podem mudar
Burocrático: controlar sempre, para alcançar objetivo planejado	**Simplista**: fazer algo simples de imediato e alterar se necessário
Orientado a processos: segui-los possibilita garantir a qualidade	**Orientado a pessoas**: motivadas comprometidas e produtivas

Metodologia Ágil x Tradicional

Abordagem Tradicional	Abordagem Ágil
Documentação gera confiança	**Comunicação** gera confiança
Sucesso = **entregar o planejado**	Sucesso = **entregar o desejado**
Gerência = "**comando-e-controle**" voltado para trabalho em massa, ênfase: papel do gerente, com planejamento e disciplina fortes	Gerência = **liderança/orientação** trabalhadores do conhecimento, ênfase: criatividade, flexibilidade e atenção às pessoas
Desenvolvedor hábil (variedade)	**Desenvolvedor** ágil (colaborador)

Metodologia Ágil x Tradicional

Abordagem Tradicional	Abordagem Ágil
Cliente pouco envolvido	**Cliente** comprometido (autonomia)
Requisitos conhecidos, estáveis	**Requisitos** emergentes, mutáveis
Retrabalho/reestruturação caro	**Retrabalho/reestruturação** barata
Planejamento direciona os resultados (incentiva controlar)	**Resultados** direcionam o planejamento (incentiva mudar)

Metodologia Ágil x Tradicional

Abordagem Tradicional	Abordagem Ágil
Premia a garantia da qualidade	**Premia** o valor rápido obtido
Foco: grandes projetos ou os com restrições de confiabilidade, planej. estratégico / priorização (exigem mais formalismo)	**Foco:** projetos de natureza exploratória e inovadores, com equipes pequenas/médias (exigem maior adaptação)
Objetivo: controlar, em busca de alcançar o objetivo planejado (tempo, orçamento, escopo)	**Objetivo:** simplificar o processo de desenvolvimento, minimizando e dinamizando tarefas e artefatos
Premia a garantia da qualidade	**Premia** o valor rápido obtido

Metodologia Ágil x Tradicional (quando usar?)

Abordagem Tradicional	Abordagem Ágil
Projetos altamente críticos	Projetos pouco críticos
Equipe iniciante	Equipe experiente
Projeto com poucas mudanças	Projeto com mudanças constantes
Equipes maiores (>= 20 pessoas)	Pequena equipe (< 20 pessoas)
Equipe distribuída	Equipe co-localizada
Cultura de controle	Cultura de adaptação

Manifesto Ágil

Valores

Indivíduos e Iterações	Processos e Ferramentas
Software funcionando	Documentação detalhada
Colaboração com cliente	Negociação de contratos
Responder a mudanças	Seguir um plano