

Introdução aos processos de software

Prof. Marcelo Werneck

Processos em nossa vida



Por que se preocupar com processos?

- O que pode acontecer quando se cozinha sem processos?



Processos e Produto

- A qualidade do produto é determinada pela qualidade do processo utilizado para produzi-lo.
- A qualidade do produto é preocupação da organização.
- Necessidades do negócio determinam a elaboração de produtos de qualidade.
- Problemas no processo geram defeitos no produto.

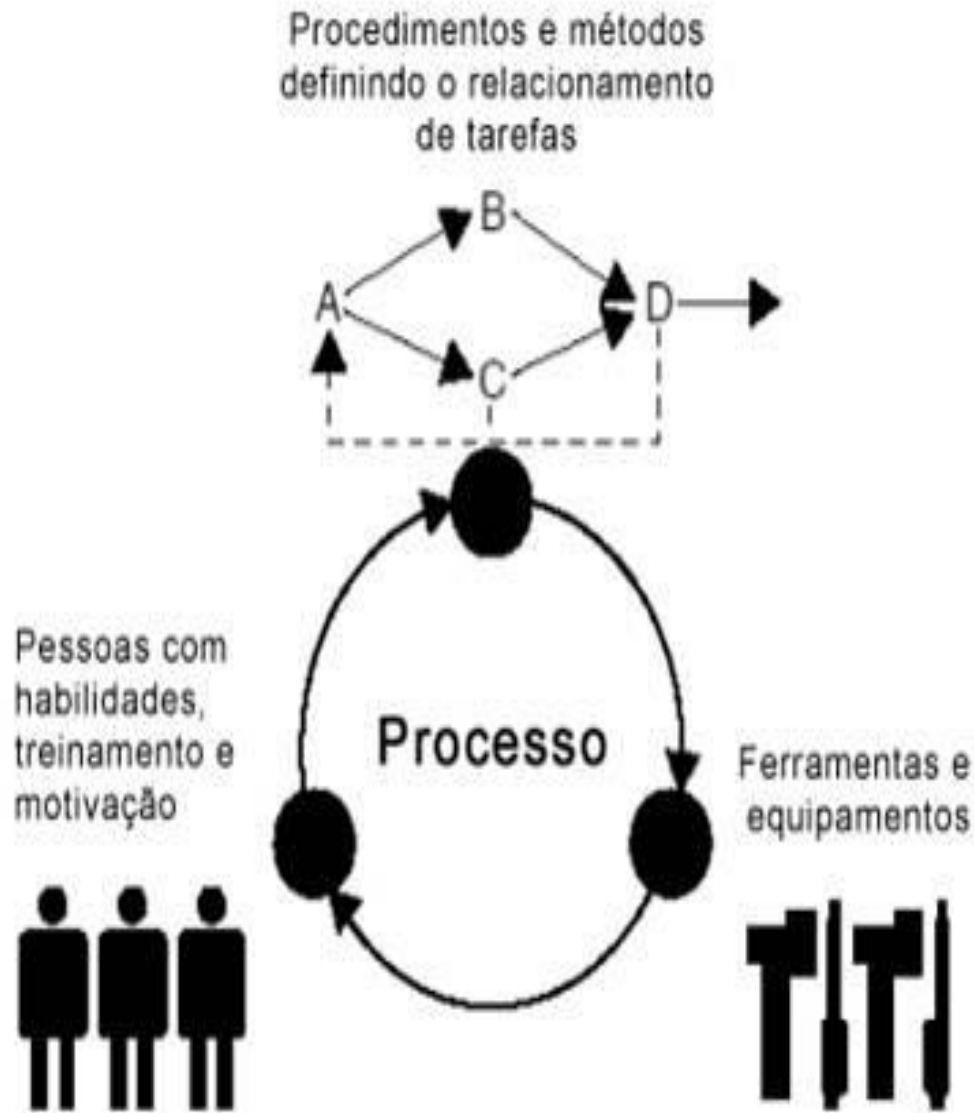
Processos e Produto

- “De cano sujo não sai água limpa”.



- Melhorias nos processos podem levar a melhorias nos produtos finais!!!

Interfaces dos Processos



A Premissa da Gerência de Processos

- A qualidade de um sistema é altamente influenciada pela qualidade do processo usado para adquirir, desenvolver ou mantê-lo.
- Esta premissa implica um foco em processos assim como em produtos:
 - Já é uma premissa antiga em manufatura
 - Visível em movimentos de qualidade em indústrias de serviço e manufatura (ex. padrões ISO)
 - Também é aplicada a desenvolvimento de software

O que são Processos?

- Conjunto de passos parcialmente ordenados que visam a obtenção de um objetivo
- Define
 - quem (papel)
 - faz o quê (produto)
 - quando (passos, tempo) e
 - como (procedimentos, insumos)
 - para obter um **objetivo**.
- Uma seqüência de passos executados com um dado propósito (IEEE)

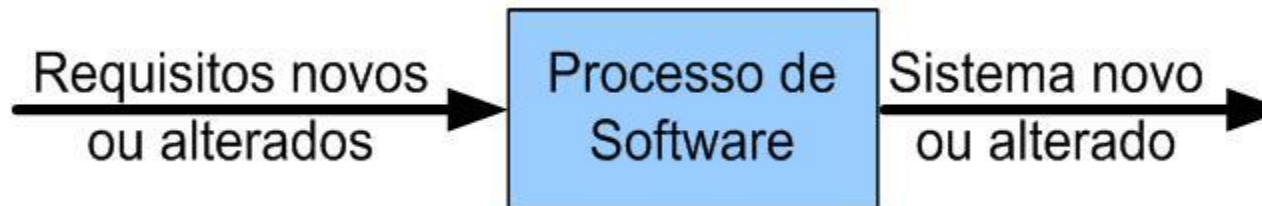
O que são Processos?

- Considerando o processo de se cozinhar:
 - quem (papel): Cozinheiro
 - faz o quê (produto): Comida / Prato
 - quando (passos, tempo): Após o café
 - como (procedimentos, insumos): Receita / Ingredientes
 - para obter um objetivo: Servir a comida.



Processos de Software

- Conjunto de passos parcialmente ordenados utilizados na confecção e evolução de sistemas computacionais (Kruchten)
- Conjunto de atividades, métodos e práticas que são utilizadas na confecção de sistemas computacionais (Humphrey).



Tipos de processos

- Processos prescritivos:
 - Processo Pessoal de *Software* (PSP)
 - Processo de *Software* para Times (TSP)
 - *Unified Process* (UP)
 - Praxis
- “Processos” (Metodologias) ágeis:
 - XP
 - Scrum



Modelos prescritivos

- Criados para colocar ordem no caos do desenvolvimento de software.
- Toda organização define um conjunto básico de atividades que compõem seu processo de desenvolvimento.
- Cada atividade deveria possuir:
 - Descrição do que fazer (passo a passo)
 - Descrição do que deve ser produzido (produtos de trabalho)
 - Descrição dos responsáveis (papéis)

Modelos prescritivos

- Modelos prescritivos prescrevem um conjunto de elementos de processo (atividades, produtos de trabalho, mecanismos de controle, ...)
- Cada modelo de processo define um fluxo de trabalho que invoca as atividades de maneiras diferentes.

Modelos prescritivos

- Modelos convencionais adotam a estratégia de previsibilidade.
- Tenta-se levantar todos os requisitos e compreender o domínio do problema antes de dar início ao processo de desenvolvimento.
- Após o levantamento dos requisitos, é feito um planejamento para que as mudanças possam ser controladas no decorrer do processo de desenvolvimento do software.

Elementos Típicos de um Processo

Elementos típicos de um processo

- Passos, atividades, insumos e resultados estão entre os **elementos** de um processo.
- Os termos apresentados são baseados nos elementos do processo Praxis;
 - embora existam correlações com outros processos.

Elementos estruturais

- **Fluxo:**
 - Subprocesso caracterizado por um tema.
 - No RUP e UP, o fluxo é chamado de **disciplina**.
 - Exemplos: requisitos, análise, desenho, implementação, testes.

Elementos estruturais

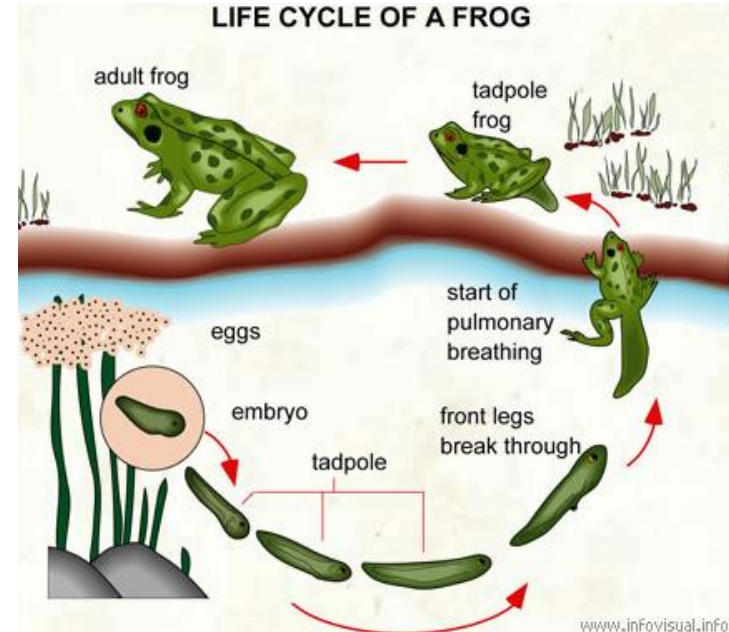
- **Atividade:**
 - Termo genérico para designar unidades de trabalho.
 - Fluxos são constituídos por atividades;
 - parcialmente ordenadas.

Elementos estruturais

- **Subfluxo:**
 - conjunto de atividades mais estreitamente correlatas;
 - que faz parte de um fluxo maior.
- **Técnica:**
 - método ou prática aplicável à execução de um conjunto de atividades.

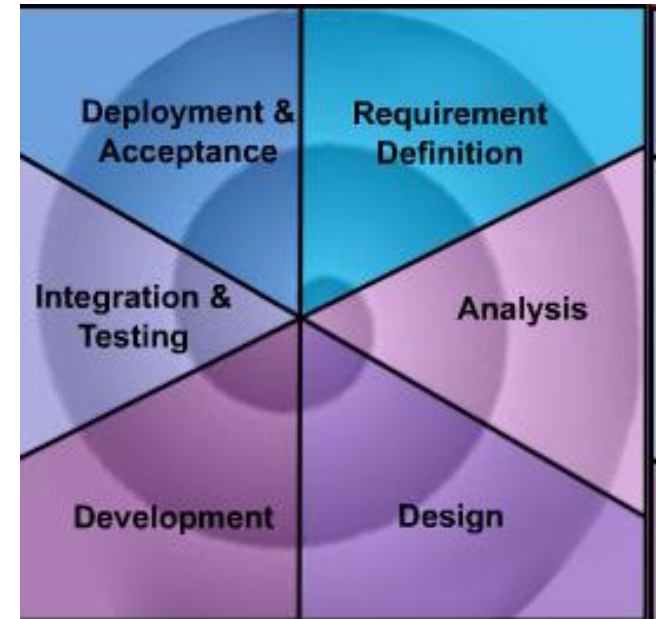
Elementos gerenciais

- **Ciclo de vida:**
 - conjunto da história de um *software*;
 - desde a percepção de sua necessidade;
 - até sua retirada de operação.
 - Determina o caráter temporal do *software*.
- **Fase:**
 - divisão maior de um processo;
 - para fins gerenciais.
 - Corresponde aos pontos principais aceitação por parte do cliente.
 - Chamada de **etapa** no RUP.



Elementos gerenciais

- **Iteração:**
 - Parte constituinte de uma fase.
 - Atinge um conjunto bem definido de metas parciais de um projeto.



Elementos pessoais



- **Papel:**
 - Responsabilidade dentro de um processo.
 - Papéis não são pessoas:
 - várias pessoas podem ter o mesmo papel;
 - uma pessoa pode acumular diversos papéis.
 - O RUP define diversos papéis:
 - Analista de Requisitos, Arquiteto, Desenvolvedor, Analista de Testes, etc;



Elementos de comunicação e resultados

- **Artefato:**
 - Representa um dos resultados da execução do processo.
 - Muitas vezes os artefatos determinam informações do processo;
 - que são trocadas entre atividades.
 - Servem de insumos para atividades ou fluxos, são transformados e servem como resultados.
 - O término de fases e iterações geralmente está associado com a entrega de um conjunto de artefatos.

Elementos de comunicação e resultados

- **Gabarito:**

- Formato típico de um artefato.
- Também chamado de *template*.
- Contém todas as necessidades de informação relacionadas a um artefato.

- **Exemplo:**

- Preenchimento típico de um artefato.
- Facilita o uso do artefato.

Elementos de comunicação e resultados

- **Padrão:**

- regras estabelecidas para o preenchimento de um ou mais artefatos;
- ou a aplicação de determinadas técnicas.

- **Lista de conferência:**

- lista para verificar a aderência a padrões ou aferir a qualidade dos artefatos.
- Também chamadas de *checklists*.



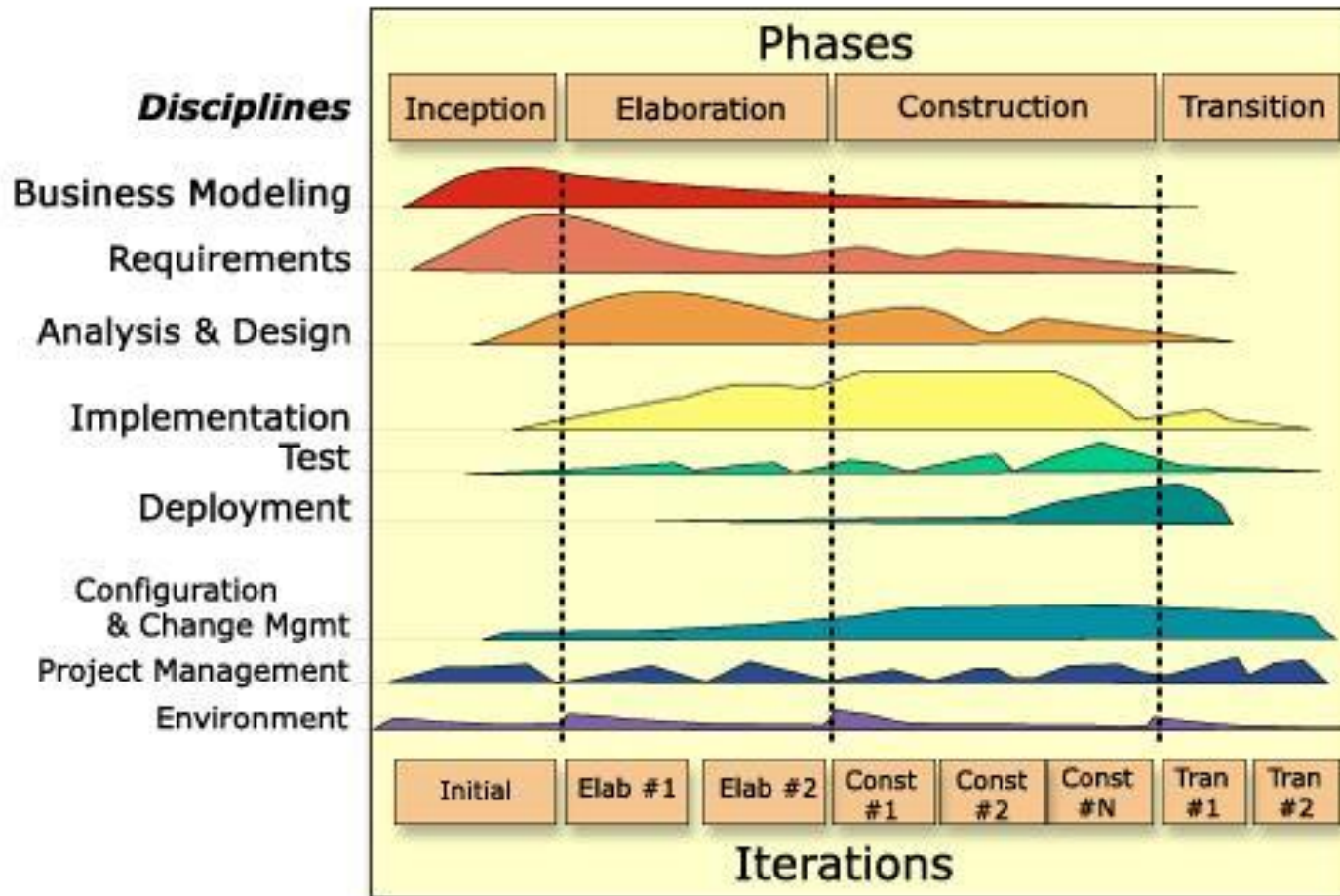
Processo Unificado

- Proposto por Booch, Jacobson e Rumbaugh;
 - proponentes da UML como notação de modelagem orientada a objetos.
- Descende de métodos anteriores propostos pelos mesmos autores.
- Utiliza a UML como notação de uma série de modelos;
 - que compõem os principais resultados das atividades do processo.

Características do Processo Unificado

- Iterativo e incremental:
 - iterações referem-se a passos do fluxo de trabalho;
 - incrementos referem-se ao crescimento do produto.
 - É prático em produtos do mundo real dividir sua entrega em iterações;
 - que incrementalmente apresentam mais e mais do sistema.
 - Em cada iteração, os engenheiros de *software* apresentam os casos de uso mais relevantes;
 - guiados pela arquitetura;
 - de forma a cada vez mais estarem próximos do sistema final.

“Gráfico das Baleias”

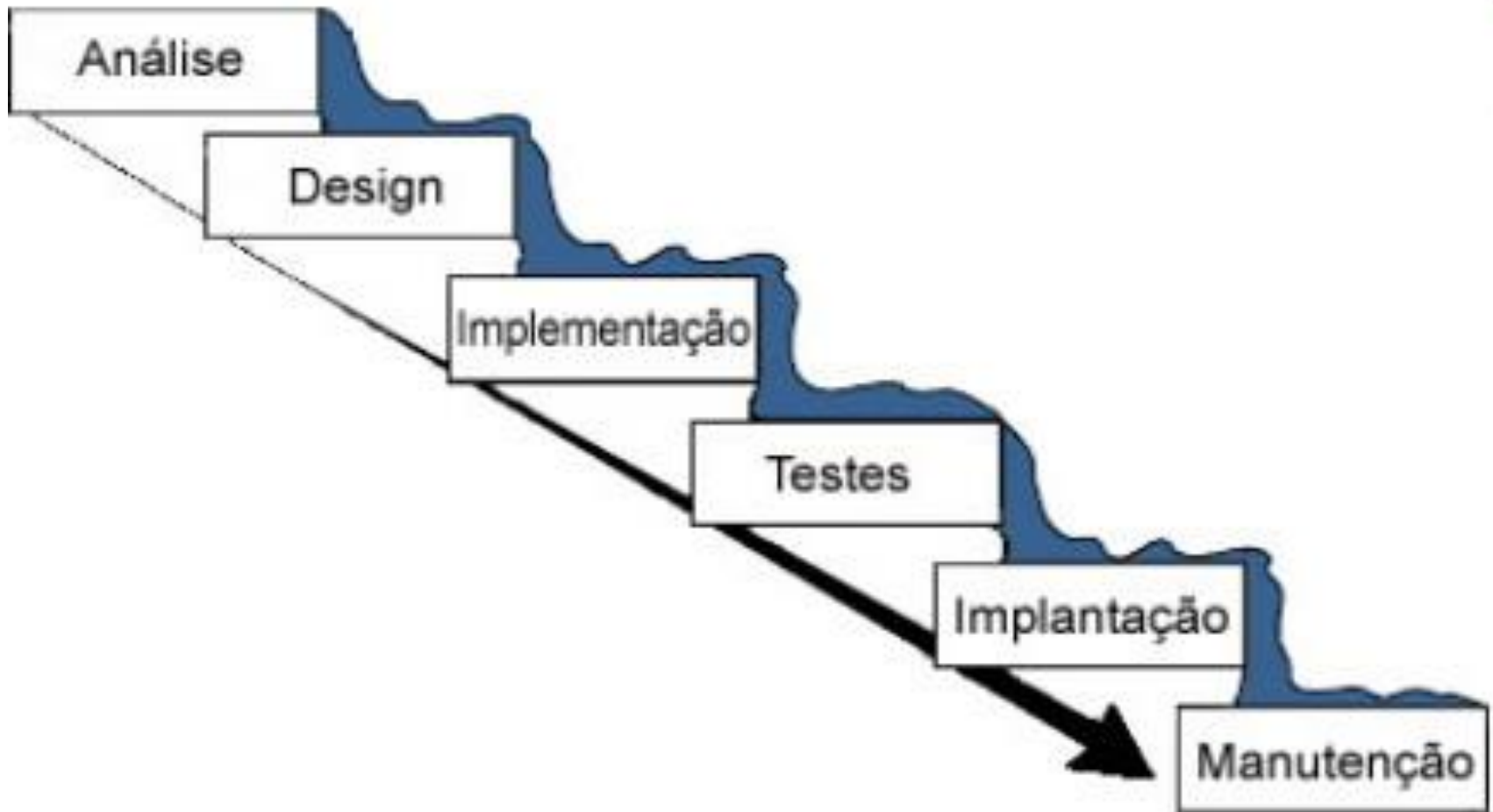


Ciclos de Vida

Modelo em Cascata

- Melhor aplicado quando:
 - Requisitos do problema são razoavelmente bem compreendidos.
 - Trabalho flui de um modo razoavelmente linear.
 - Adaptações bem definidas ou aperfeiçoamento de um sistema existente precisam ser feitos.
 - Requisitos bem definidos e estáveis.
- Sugere abordagem sequencial.
- Paradigma mais antigo da Engenharia de Software.

Modelo em Cascata



Modelo em Cascata

- Problemas encontrados:
 - Projetos reais raramente seguem o fluxo sequencial proposto.
 - Difícil para o cliente estabelecer todos os requisitos explicitamente. Difícil acomodar incerteza natural que existe no início dos projetos.
 - Cliente precisa esperar. Versão executável só é disponibilizada ao final.

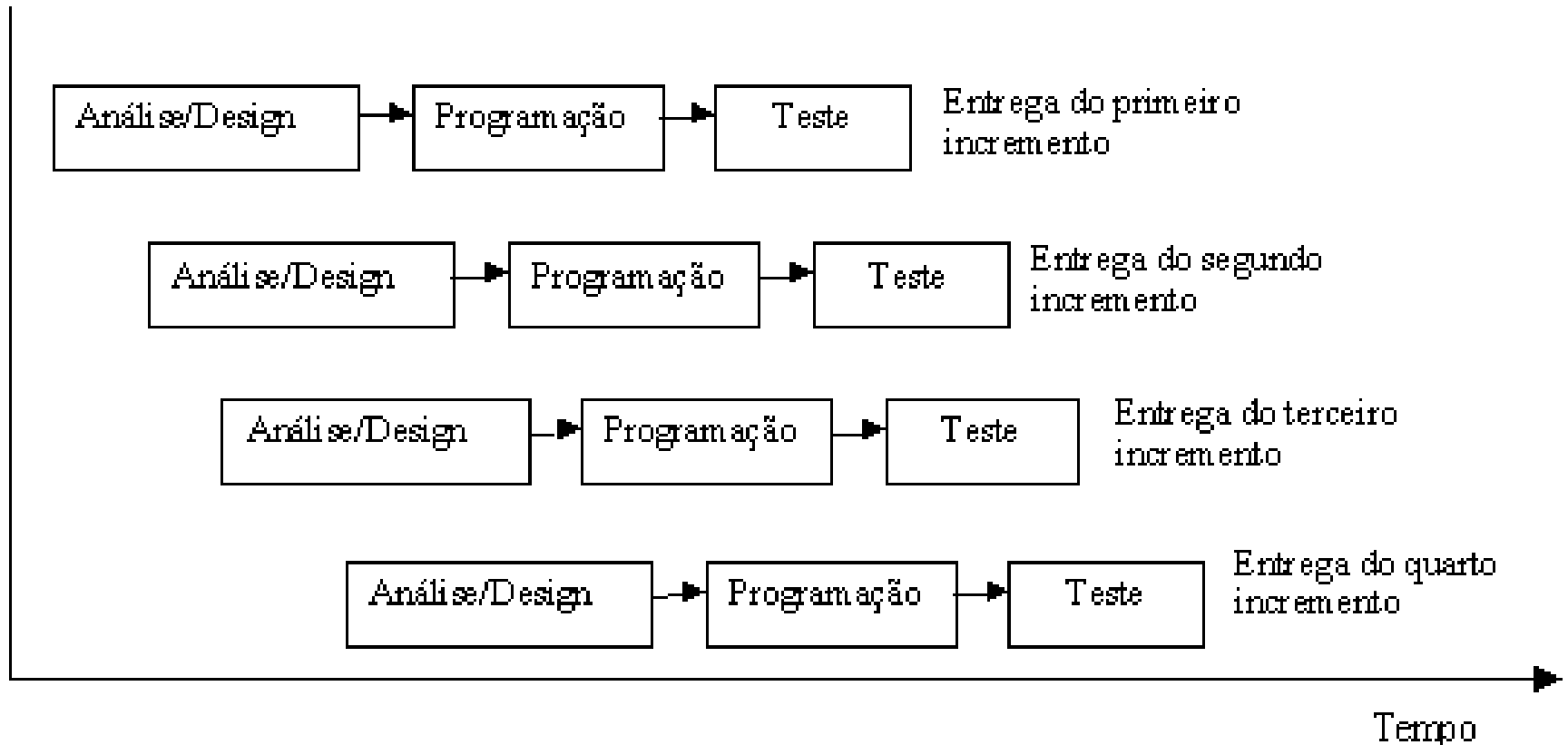
Modelo Incremental

- Combina elementos do modelo em cascata de maneira iterativa.
- Aplica sequências lineares. Cada uma produz “incrementos” do software passíveis de serem entregues.
- Apresenta um produto operacional a cada incremento.

Modelo Incremental

- Útil quando:
 - Não há mão-de-obra disponível para uma implementação completa. Primeiros incrementos com equipe menor.
 - Incrementos podem ser planejados para gerir riscos técnicos.

Modelo Incremental



Modelos Evolucionários

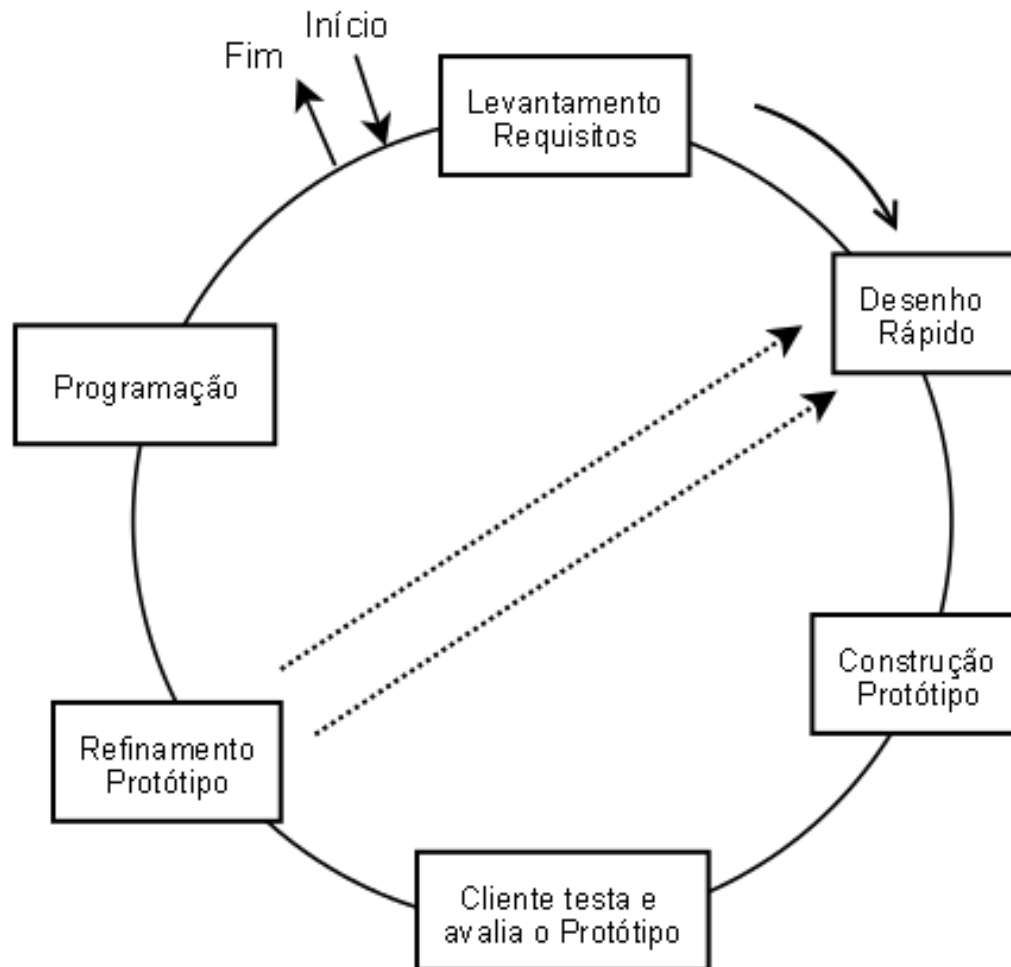
- Uma versão reduzida do software pode ser elaborada e incrementada.
- Um conjunto de requisitos básico é bem entendido mas detalhes ainda precisam ser esclarecidos.
- São iterativos. Permitem desenvolver versões cada vez mais completas do software.
- Adequado para acompanhar um produto que evolui com o tempo.

Prototipagem

- Recomendado quando:
 - Cliente define objetivos gerais do sistemas mas não detalha os requisitos.
 - Incertezas em relação à eficiência de um algoritmo, adaptabilidade de um componente, requisitos de usabilidade
- Pode ser uma técnica implementada dentro do contexto de outros modelos.

Prototipagem

Modelo de Desenvolvimento: "Prototipagem"



Prototipagem

- Problemas possíveis:
 - Cliente não espera que o software irá funcionar “precariamente” ou com grandes limitações.
 - Cliente pode tentar transformar o protótipo num produto executável.
 - Desenvolvedor pode fazer opções simplificadas e mantê-las no desenvolvimento do produto (além do protótipo).
- Deve-se definir se o protótipo será descartado ou não.

Modelo Espiral

- Combina a natureza iterativa da prototipagem com os aspectos controlados e sistemáticos do modelo em cascata.
- Desenvolvimento rápido de versões cada vez mais completas.
- Série de versões evolucionárias.
- Cada ciclo pode ser visto com um projeto.
- Adequada para software de grande porte.

Modelo Espiral

