

# Engenharia de Software-2003

Mestrado em Ciência da Computação

Departamento de Informática - UEM

Profa. Dra. Elisa H. M. Huzita

# Produto de Software





# Conceitos

## ● Software

“criação intelectual compreendendo os programas, procedimentos, regras e qualquer documentação correlata à operação de um sistema de processamento de dados.” ISO9000-3

## ● Produto de Software

“Conjunto completo de programas de computador, procedimentos e documentação correlata, assim como dados designados para entrega a um usuário.” ISO9000-3

Produto:

- mercado - editor ou
- usuário específico - negócio

## ● Outros termos: sistemas de software, aplicações, aplicativos

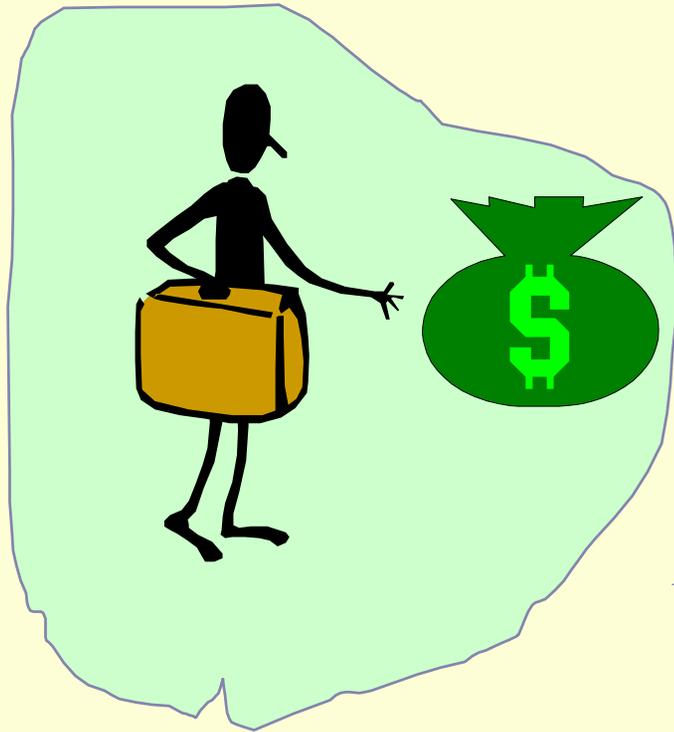




# O Papel do Software na Sociedade

- LAS (*London Ambulance System*) é automatizar o tratamento de chamadas de emergência e a atribuição de ambulâncias a acidentes.
  - É o maior serviço de ambulância do mundo, abrangendo uma população de aproximadamente 6.8 milhões. LAS transporta mais de 5.000 pacientes por dia, recebe entre 2.000 e 2.500 chamadas diariamente.
  - Falhas no sistema em 26 e 27 de Outubro de 1992 causaram problemas tais como:
    - alocação ineficiente de ambulâncias (duplicada e atrasada - múltiplas ambulâncias foram enviadas para o mesmo acidente ou o veículo mais próximo não foi enviado para o acidente mais próximo);
    - uma crescente lista de mensagens de exceção e uma enorme lista de espera;
    - um aumento do tempo de resposta proporcional ao aumento de mensagens e da lista de espera;
    - um número crescente de telefonemas repetidos (“call backs”).
- 

# Como produzir software?



Mundo Real



Automatizar



Computador



# Questões comumente colocadas ...

- Porque demora-se tanto para construir um produto de software?
  - Porque os custos são tão altos?
  - Porque não podemos detectar todos os erros antes de entregar um produto de software para o cliente?
  - Porque é difícil mensurar o progresso do desenvolvimento de um produto de software?
- 



# Características de software

- Software é desenvolvido e não manufaturado no sentido clássico.
  - Software não se deteriora ... não existem componentes de reposição.
  - Software é feito sob encomenda, ao invés de ser construído a partir de componentes (está mudando).
  - A evolução tecnológica afeta diretamente as facilidades que podem ser incorporadas ao software e as técnicas de construção destes.
  - Tempo para transferência de tecnologia.
- 



# Tipos de Sistemas de Software

- Software básico
  - Software para sistema em tempo real
  - Software comercial
  - Software para engenharia e aplicações científicas
  - Software embarcado
  - Software para computadores pessoais
  - Software baseados em inteligência artificial
  - Software de entretenimento
- 



# Problemas relacionados a produção de software

- A sofisticação do hardware ultrapassa nossa habilidade de construir software que utilize todo o potencial do hardware.
  - Nossa habilidade para construir novos produtos de software não acompanha a demanda.
  - Nossa habilidade de manter os produtos de software existentes estão ameaçadas por projetos ineficientes e recursos inadequados (pessoal mal treinado ou mal alocado).
- 



# Sintomas e Causas dos Problemas

- mau entendimento das necessidades dos usuários finais
  - falta de habilidade para tratar com modificações de requisitos;
  - módulos que não “trabalham” juntos;
  - software que é difícil de manter ou extender;
  - descoberta, de forma tardia, das falhas de projeto;
  - baixa qualidade de software;
  - desempenho inaceitável;
  - os membros da equipe não conseguem reconstruir pois não sabem quem modificou o que, quando, onde e porque;
  - um processo “irresponsável” de construir- entregar
- 



# Que resultaram de....

- gerenciamento adhoc dos requisitos
  - comunicação ambígua e imprecisa;
  - arquiteturas frágeis;
  - grande complexidade;
  - inconsistência detectada em requisitos, projetos e implementação;
  - testes insuficientes;
  - avaliação subjetiva do estado do projeto;
  - falha em atacar/avaliar riscos;
  - propagação incontrolada de modificações;
  - automação insuficiente.
- 



# Tamanho e complexidade...

- Fatores que influenciam no tamanho:
  - Contexto (escopo);
  - Número de interfaces com outros sistemas;
  - Número e tipo de usuários;
  - Número de entidades (dados) e atributos.





# Tamanho e complexidade...

- Fatores que influenciam a complexidade:
    - Complexidade do processo de negócio que precisa ser automatizado;
    - Complexidade das regras de negócio;
    - Volume de transações para o processo por unidade de tempo;
    - Localização geográfica dos usuários;
    - Complexidade das relações que existem entre as várias entidades que usam o sistema;
    - Tipo e nível de maturidade que é requerido para executar o sistema.
- 



# Atributos de um bom software

- funcionalidades e desempenho requeridos;
  - Confiança: confiabilidade, segurança;
  - Manutenção;
  - Interoperável;
  - Fácil uso: interface e documentação adequados.
- 



# Grandes desafios (atualmente)

- Sistemas legados;
  - Heterogeneidade;
  - Redução de custo;
  - Redução de tempo de entrega.
- 



# Engenharia de Software

- Uma definição:

- O estabelecimento e uso de um conjunto de princípios para se obter, **economicamente**, um software que seja **confiável** e trabalhe **eficientemente** em máquinas reais.

- Três elementos chaves:

- **métodos**
  - **ferramentas**
  - **procedimentos (gerenciamento de projetos)**
- 

# A Evolução do Software

## Os primeiros anos

- sistemas batch
- distribuição limitada
- software personalizado

## A segunda era

- sistemas multiusuários
- sistemas em tempo real
- banco de dados
- software produto

## A terceira era

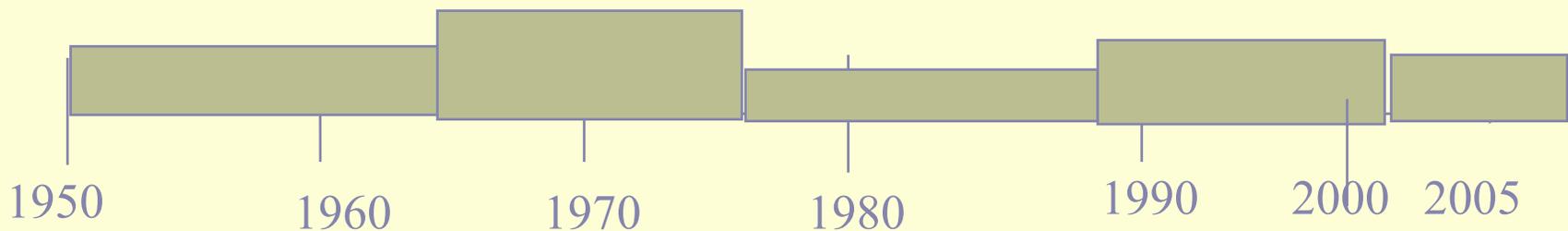
- sistemas distribuídos
- incorporação de inteligência
- hardware de baixo custo
- impacto do consumidor

## A quarta era

- sistemas desktop poderosos
- tecnologia de orientação a objetos
- sistemas especialistas
- redes neurais
- computação paralela
- comunicação intergaláctica

## A quinta era

- Computação móvel
- .....
- Tecnologia de componentes





# O que foi feito?

- Crise de Software
  - Aflição crônica (*Pressman, 1991*)
  - Bons Exemplos:
    - sistemas telefônicos
    - sistemas bancários
    - sistemas de reservas de passagens aéreas
    - folhas de pagamento, contabilidade, etc.
- 



# O que foi feito?

- Programação estruturada
  - Análise estruturada
  - Sistemas de gerenciamento de banco de dados
  - Linguagens de 4a. Geração
  - Verificação, validação e teste de software
  - Métodos formais (sequenciais, concorrentes)
  - Análise da complexidade de algoritmos
  - Tipos abstratos de dados
  - Análise orientada a objetos
  - Modelagem e automação de processos
  - Ferramentas CASE
- 



# Processo de software

## ● Definição:

- Conjunto de atividades e resultados associados que geram um produto de software

## ● Envolve:

- Especificação;
  - Desenvolvimento;
  - Validação;
  - Evolução.
- 



# Processo de Software





# Modelos de Processo de Desenvolvimento de Software

- Modelo de ciclo de vida

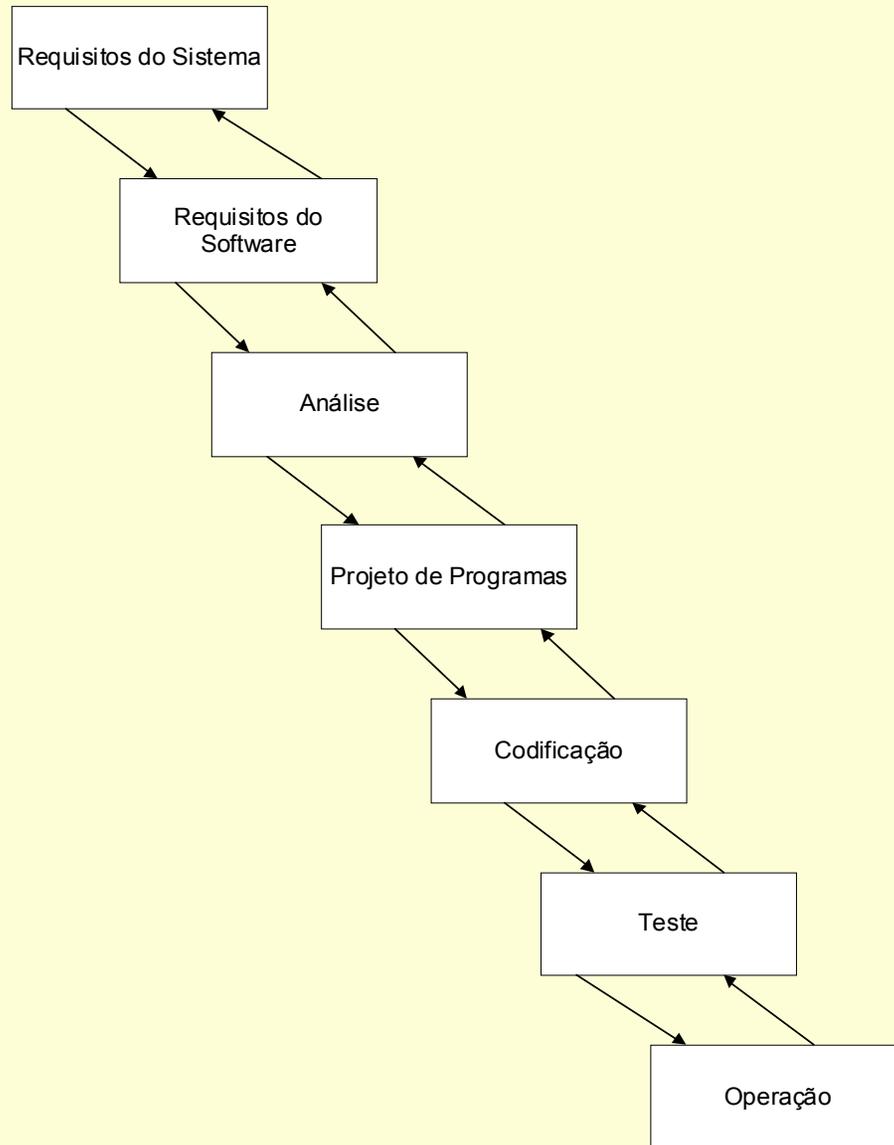
descrições abstratas do processo de desenvolvimento e modificação, tipicamente, mostrando os principais estágios de desenvolvimento e manutenção de um software executável.

- Processo de Software

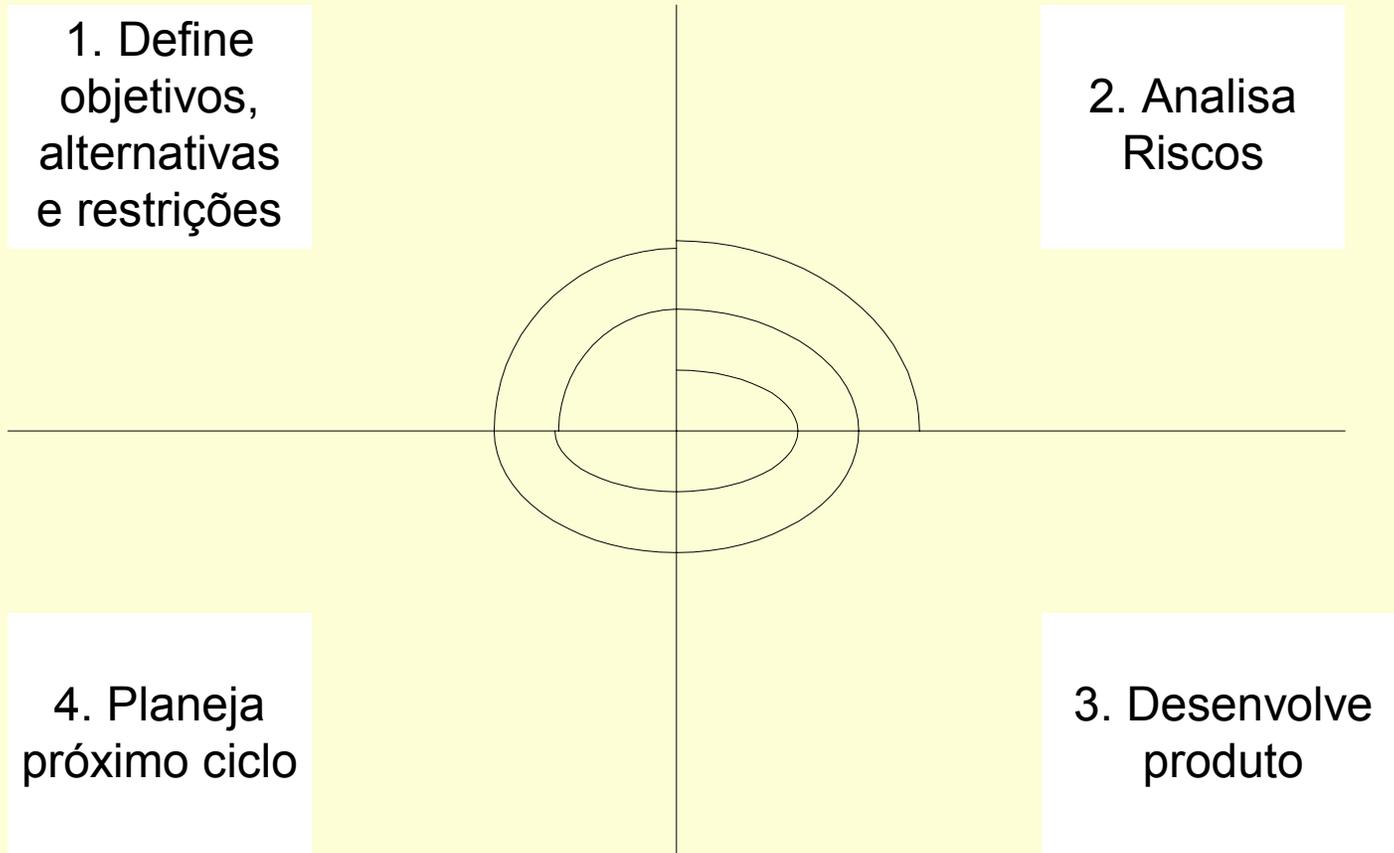
- Desenvolvimento
- Manutenção
- Uso



# O Modelo Cascata



# Modelo Espiral





# Outros modelos

- Modelo em V
  - Modelo incremental
  - Desenvolvimento baseado em reuso
  - Desenvolvimento utilizando métodos formais
- 



# Ciclo de vida canônico

- Estudo de Viabilidade
  - Iniciação do projeto
  - Especificação de requisitos
  - Projeto da arquitetura
  - Projeto detalhado
  - Codificação
  - Teste de unidade
  - Teste de aceitação
  - Teste operacional
  - Encerramento do projeto
  - Operação
  - Desativação do produto
- 

# Seleção de Estágios

- Análise de requisitos
- Especificação do sistema
- Projeto da arquitetura
- Projeto detalhado
- Implementação
- (Manutenção e evolução)

O que?

Como?



# Estágios...

- Análise de Requisitos (sistemas e software)
    - Software é sempre parte de um sistema maior que envolve hardware, pessoas, etc.modelo de negócios
    - Coleta de requisitos do software, especificamente – técnicas
  - Especificação do sistema
    - Expressr os requisitos de maneira formal através de diagramas bem definidos ou especificações matemáticas
  - Projeto da arquitetura
    - Determinar a estrutra do software com seus componentes e conectores
- 



# Estágios...

## ● Projeto detalhado

- Projetar a concretização da especificação do software, definindo base de dados, representações de interfaces, algoritmos, etc...

## ● Implementação

- Escrever o projeto do sistema em uma linguagem de programação

## ● Manutenção e evolução

- Corrigir eventuais erros no software e efetivar atualizações
- 



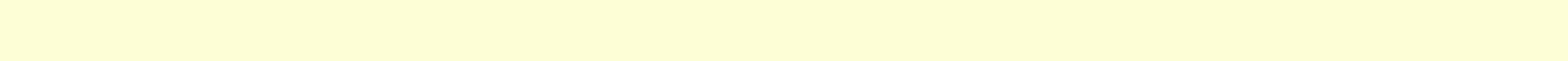
# Seleção de Métodos

- Em geral, mais de um método é utilizado para cobrir todo o ciclo de vida.
- A escolha deve considerar:
  - características técnicas da aplicação
  - características técnicas do método
  - cobertura do ciclo de vida





# Modelos

- Um **modelo** é uma visão abstrata de um sistema.
  - Modelos de sistema complementar apresentam informações diferentes sobre o sistema
  - Princípio da modelagem: modificação de requisitos é mais fácil e mais barato enquanto se está modelando do que em produção
- 



# Modelos

## Por que construir modelos?

- Possibilita melhor entendimento da área sob estudo;
  - Possibilita validar os requisitos
  - Possibilita confirmar o entendimento dos requisitos com o usuário
- 



# Tipos de modelos

- Modelo de contexto;
  - Modelo de fluxo de dados;
  - Modelo de máquina de estados;
  - Modelo de arquitetura;
  - Modelo semântico de dados;
  - Modelo de objetos.
- 



# Modelagem de software

- Análise estruturada
  - Orientada a objetos
  - Baseada em componentes
  - Baseada em agentes
  - Orientada a aspectos
- 



# Pontos de Vista

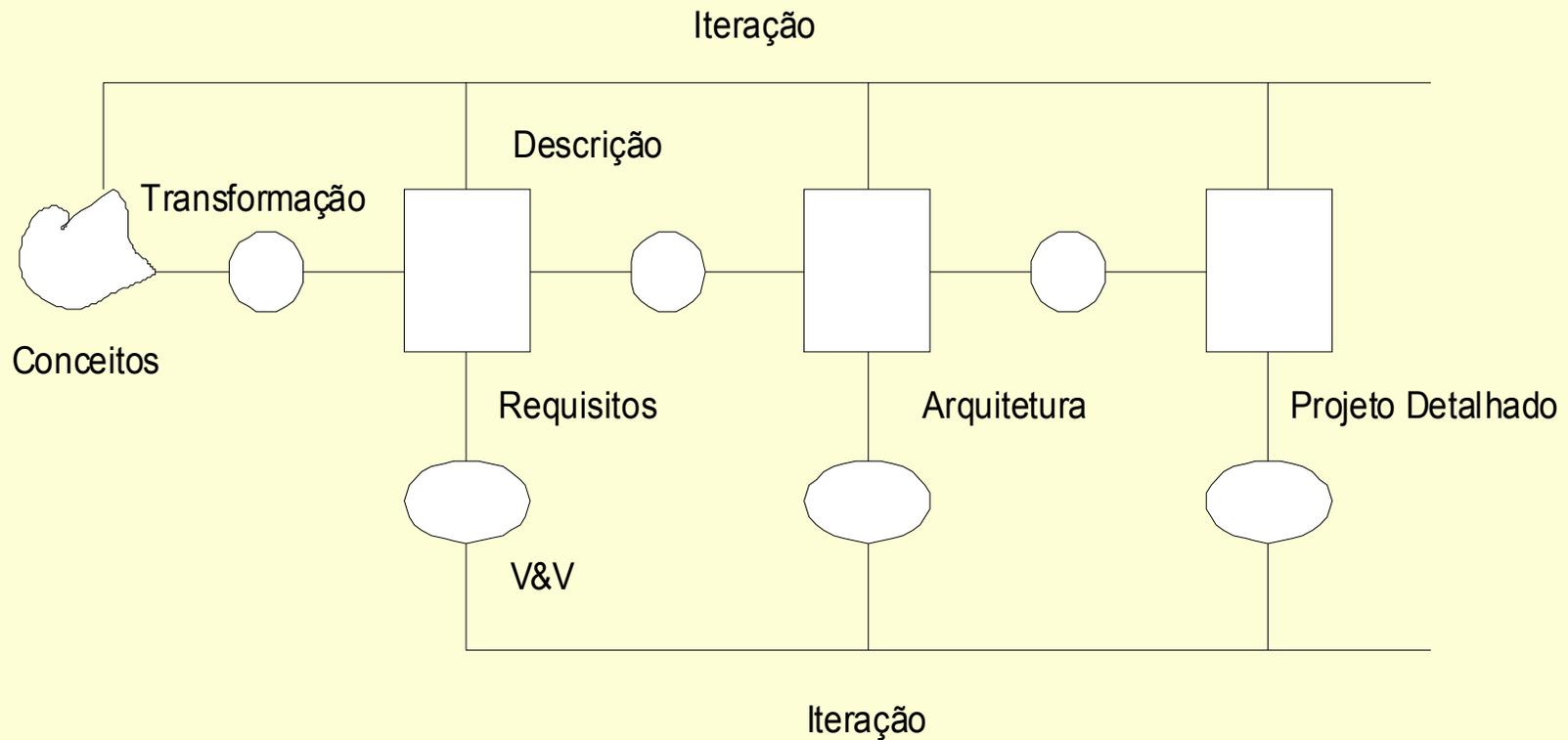
## Técnico

- envolve estágios compostos de atividades tecnicamente relacionadas. Ex. atividades de elicitação e validação na análise de requisitos.

## Gerencial

- envolve fases compostas de grupos de atividades, temporariamente relacionadas. Essas fases começam e terminam em um ponto bem definido e terminam com a revisão de alguns produtos.
- 

# Ponto de Vista Técnico: modelo transformacional





# Ponto de Vista Técnico: modelo transformacional

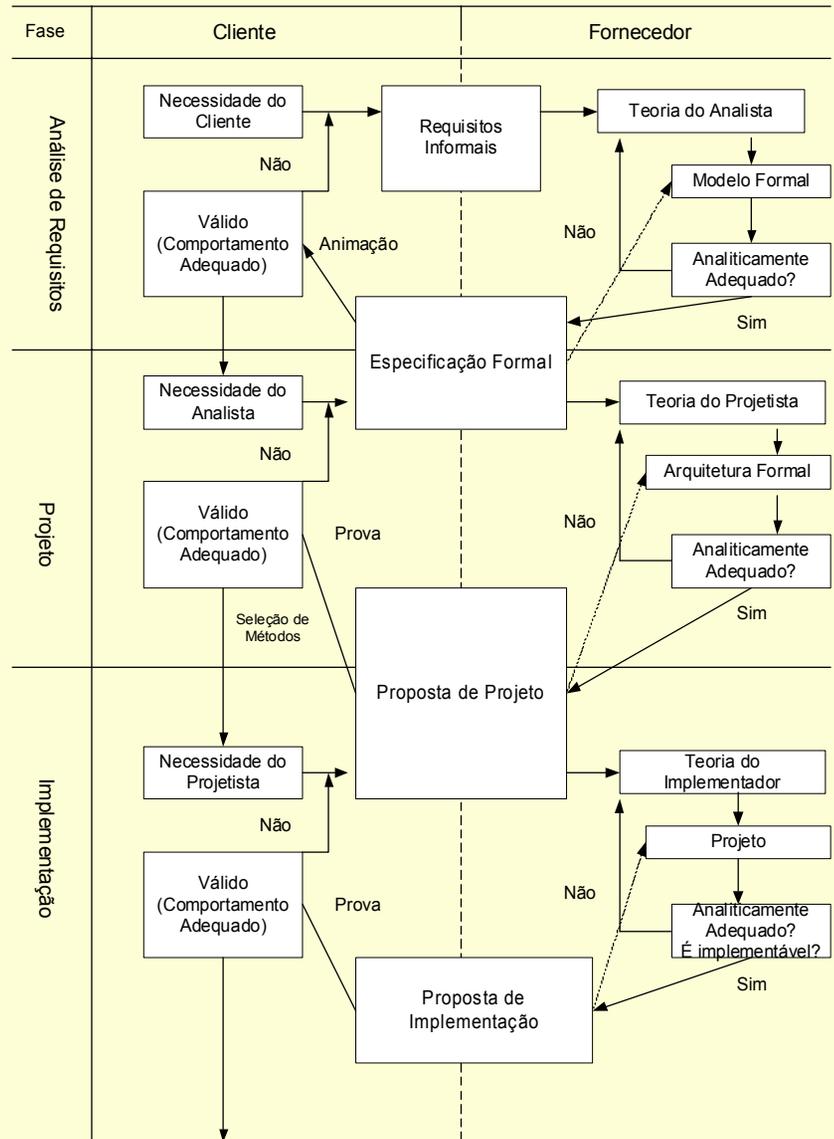
- Baseado em uma visão abstrata das atividades desenvolvidas no processo de software como uma seqüência de transformações.
  - Várias descrições intermediárias são produzidas durante o processo.
  - Erros são descobertos no desenvolvimento, o que causa iteração.
  - V&V são usadas para determinar as atividades que levam a iteração.
    - Verificação: estamos construindo o produto corretamente?
    - Validação: estamos construindo o produto certo?
  - Oferece uma base para escolha de um método: notação,
- 



# Ponto de Vista Técnico: Modelo Contratual

- Considera algumas das especificações como contratos entre o pessoal que trabalha em estágios adjacentes.
  - Ilustração do modelo para desenvolvimento formal:
    - especificação: descrição
    - validação: lado esquerdo
    - verificação: lado direito
- 

# Modelo Contratual





# Gerenciamento de Risco Técnico

- Riscos técnicos: detectar requisitos incorretos, seguir um método errado; má interpretação das descrições entre equipes.
  - Garantir que os estágios serão completados com sucesso.
  - Abordagens
    - Contratual
    - Prototipação
    - Incremental
- 



# Abordagem contratual

- A abordagem principal do gerenciamento de riscos é que todo esforço deve ser colocado nas atividades de V&V antes de tornar uma descrição em *baseline*.
  - Existem *Breakpoints* bem definidos nos quais se pode avaliar o progresso do projeto.
- 



# Abordagem Prototipação

- Validar a precisão dos requisitos ou aceitabilidade das decisões.
  - Validar a viabilidade de uma estratégia proposta.
  - Observações:
    - protótipos só são válidos se construídos rapidamente
    - protótipos devem ser desprezados.
- 



# Abordagem incremental

- Definir e desenvolver uma pequena parte do sistema de cada vez.
  - Desenvolver um núcleo do sistema inicialmente e depois adicionar funcionalidades em subprojetos.
  - Grande atração é que gera sistemas parciais executáveis e utilizáveis de onde se pode obter feedbacks e ganhar credibilidade do usuário.
  - Na prática é difícil de desenvolver uma abordagem incremental sem ter um entendimento completo dos requisitos.
- 



# Ponto de Vista Gerencial

- Objetivo Principal: controlar o processo de desenvolvimento para que o produto resultante seja entregue:
    - no prazo
    - com o orçamento previsto
    - com qualidade e confiabilidade aceitáveis
    - obedecendo às expectativas e requisitos dos usuários
    - satisfazendo as especificações
- 



# Ponto de Vista Gerencial

- Questões principais:
  - análise de riscos
  - planejamento e controle
  - tomada de decisão





# Comentários

- Sempre deve existir um processo de software definido - padrões de qualidade.
  - O modelo canônico deve ser tratado como uma referência que deve ser adaptada para cada situação.
  - Criar um processo baseado em fases específico para o projeto.
- 