



# PRINCÍPIOS DA ENGENHARIA DE SOFTWARE – AULA 03

Prof.: Franklin M. Correia

1

# NA AULA ANTERIOR.....

- Modelos de processos de software:
  - Evolucionário
    - Tipos
    - Problemas
  - Prototipação
  - Baseado em componentes
  - Iteração de processo de desenvolvimento
    - Entrega incremental

# O QUE TEMOS PARA HOJE.....

- Iteração de processo de desenvolvimento
  - Desenvolvimento espiral
- Atividades de Processo de desenvolvimento
  - Especificação de Software
  - Projeto e implementação de software
  - Validação de software
  - Evolução de software
- Técnicas de 4º Geração
- Ferramentas CASE

# ITERAÇÃO – DESENVOLVIMENTO ESPIRAL

- Proposto por Boehm.
- Processo é apresentado como uma espiral e não como uma sequencia de atividades
- O sistema evolui em espiral para fora;
- Parte de um esboço inicial até o final;

# ITERAÇÃO – DESENVOLVIMENTO ESPIRAL

## ○ Fases

- Desenvolvimento de Objetivos
- Avaliação e Redução de Riscos
- Desenvolvimento e Validação
- Planejamento

# ITERAÇÃO – DESENVOLVIMENTO ESPIRAL

- Desenvolvimento de objetivos
  - Objetivos específicos são definidos
  - Identificação de restrições sobre o projeto e produto
  - Elaboração de plano de gerenciamento detalhado
  - Riscos do projeto são identificados
    - A depender do risco as estratégias são planejadas

# ITERAÇÃO – DESENVOLVIMENTO ESPIRAL

- Avaliação de redução de riscos
  - Para cada risco identificado uma análise detalhada é realizada
  - Ações são executadas para reduzir o risco
    - Se houver o risco de que existem requisitos não apropriados, pode-se desenvolver protótipos do sistema

# ITERAÇÃO – DESENVOLVIMENTO ESPIRAL

- Desenvolvimento e validação

- Um modelo de desenvolvimento para o sistema é selecionado;
  - Se riscos de interfaces forem identificados pode-se adotar o modelo de desenvolvimento de prototipação evolucionária
  - O modelo em cascata pode ser escolhido se o principal risco identificado for a integração de subsistemas



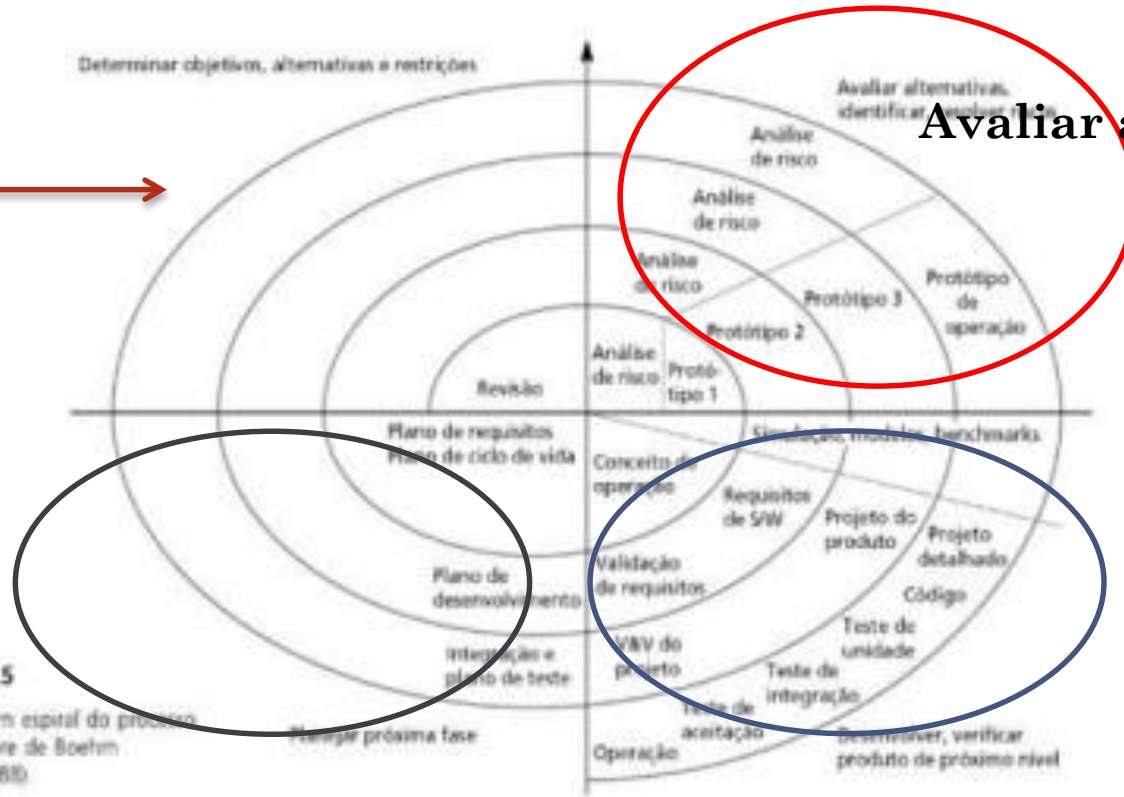
# ITERAÇÃO – DESENVOLVIMENTO ESPIRAL

## ○ Planejamento

- O projeto é revisado e a decisão de prosseguir ao próximo loop é tomada
- Para o próximo loop o planejamento será elaborado para a próxima fase do projeto

# ITERAÇÃO – DESENVOLVIMENTO ESPIRAL

Definição de objetivos



Avaliar alternativas

Figura 4.5

Modelo em espiral do processo de software de Boehm (Boehm, 1988)

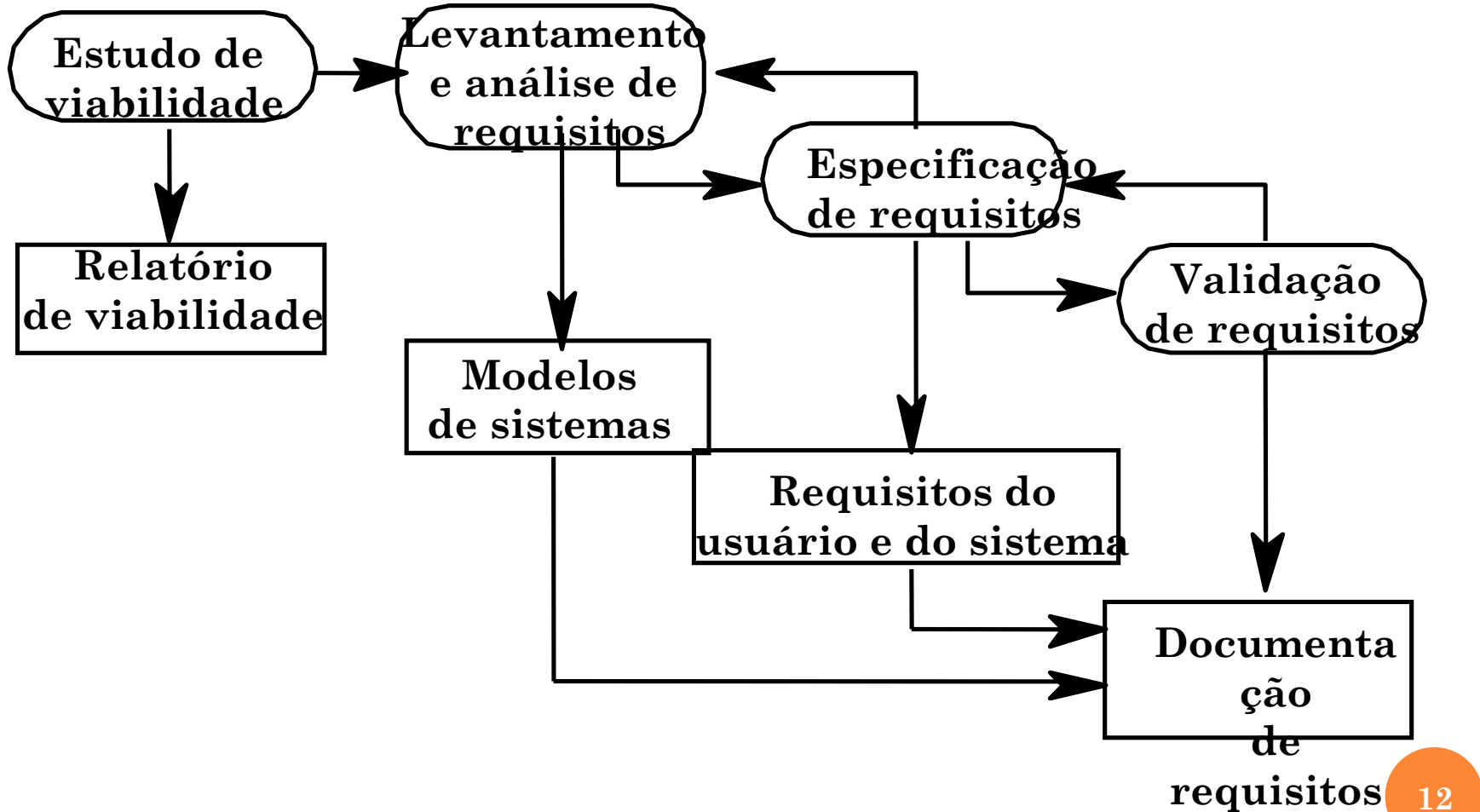
Planejar próxima fase

Desenvolver, verificar produto de próximo nível

# ATIVIDADES DO PROCESSO DE SOFTWARE

- Engenharia de requisitos
  - Processo de compreender e definir quais serviços são necessários e identificar as restrições de operação e desenvolvimento do sistema;
  - Estágio crítico do software
    - Erros significam a problemas no projeto e na implementação do sistema

# ATIVIDADES DA ENGENHARIA DE REQUISITOS



# ATIVIDADES DA ENGENHARIA DE REQUISITOS

- Estudo de viabilidade
  - Avalia se as necessidades do usuário podem ser satisfeitas por meio de tecnologias atuais de software e hardware;
  - Considera se o sistema terá custo adequado comercialmente;
  - Se o sistema pode ser desenvolvido com as restrições e orçamento existentes;

# ATIVIDADES DA ENGENHARIA DE REQUISITOS

- Levantamento de Análise de Requisitos
  - Processo de derivação de requisitos do sistema através:
    - Da observação de sistemas existentes
    - Discussões com usuário e possíveis compradores
    - Pode envolver o desenvolvimento de um ou mais modelos de protótipos
  - Ajuda o analista a compreender o sistema especificado

# ATIVIDADES DA ENGENHARIA DE REQUISITOS

- Especificação de requisitos
  - Traduz as informações coletadas com os usuários em um documento contendo um conjunto de requisitos.
  - Tipos de Requisitos:
  - Requisitos de usuário
    - Declarações para os requisitos do ponto de vista do usuário final
      - Funcionalidades, telas, processo da empresa
  - Requisitos de sistema
    - Declaração para os requisitos do ponto de vista do sistema
      - Detalhamento de funcionalidades

# ATIVIDADES DA ENGENHARIA DE REQUISITOS

- Validação de requisitos
  - Verifica o realismo dos requisitos, consistência e abrangência;
  - Objetivo de descobrir erros no documento de requisitos;
  - Uma vez descoberto erro, devem ser corrigidos;



## ATIVIDADES DA ENGENHARIA DE REQUISITOS- **COMENTÁRIO**

- As atividades do processo de requisitos não tem sua sequência restrita. A análise de requisitos continua durante a definição e a especificação. Com o amadurecimento do conhecimento sobre o negócio, novos requisitos aparecem ao longo do processo.

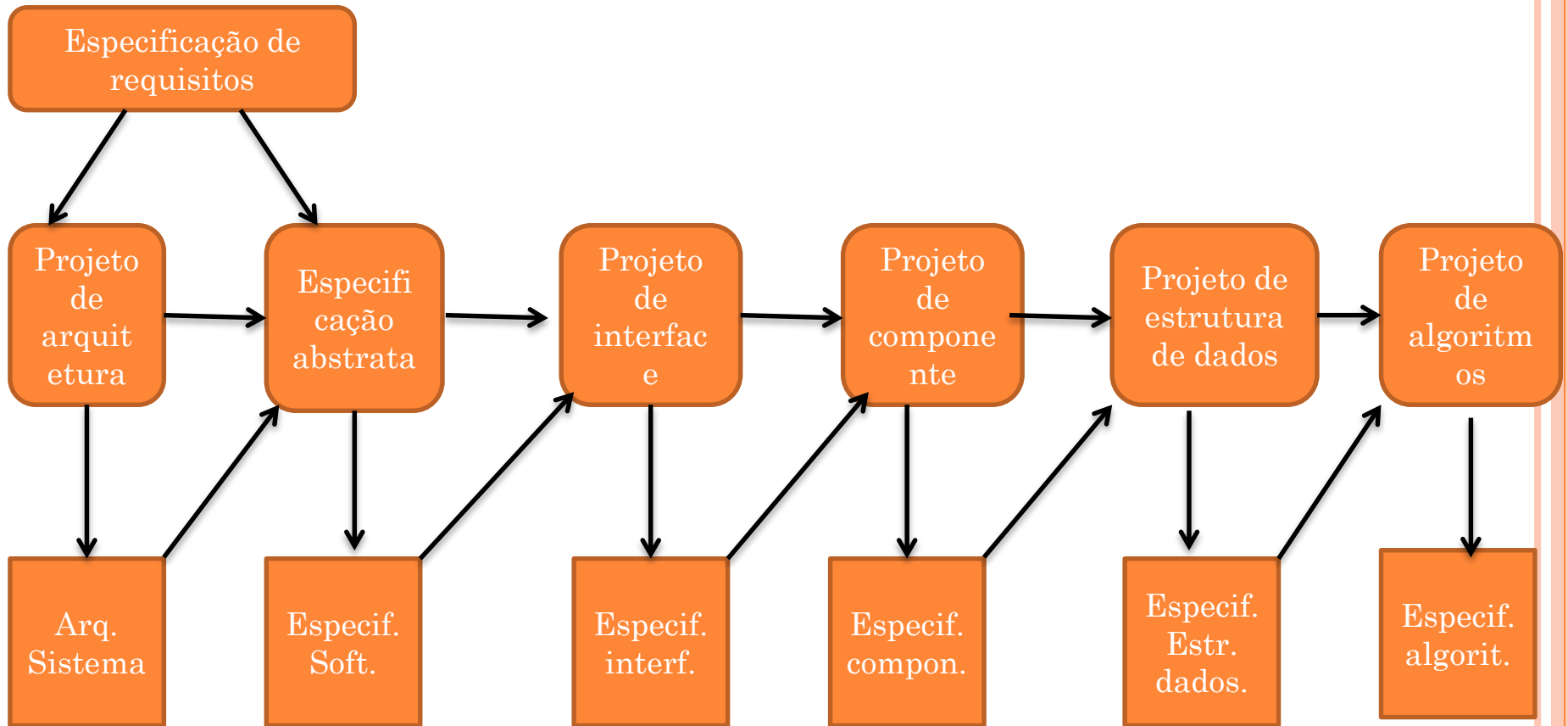
# PROJETO DE IMPLEMENTAÇÃO DE SOFTWARE

- Tradução de uma especificação de um sistema em um sistema executável
- Sempre envolve processos de projeto e programação
- Projeto de software é a descrição da estrutura do software, dos dados do sistema, interfaces entre os componentes do sistema e algoritmos usados

# PROJETO DE IMPLEMENTAÇÃO DE SOFTWARE

- Desenvolvimento do projeto de forma iterativa
  - Desenvolvimento baseado em versionamento
- Pode envolver o desenvolvimento de modelos de sistemas com diferentes níveis de abstração
- Erros e omissões dos estágios anteriores podem ser descobertos

# PROJETO DE IMPLEMENTAÇÃO DE SOFTWARE



# PROJETO DE IMPLEMENTAÇÃO DE SOFTWARE

- Projeto de Arquitetura
  - Os subsistemas constituintes do sistema e seus relacionamentos são identificados e documentados
- Especificação abstrata
  - Para cada subsistema são produzidas uma especificação abstrata dos serviços e as restrições que ele deve operar
- Projeto de interface
  - Para cada subsistema é projetada e documentada a interface com outros sistemas
  - Não deve ser ambígua

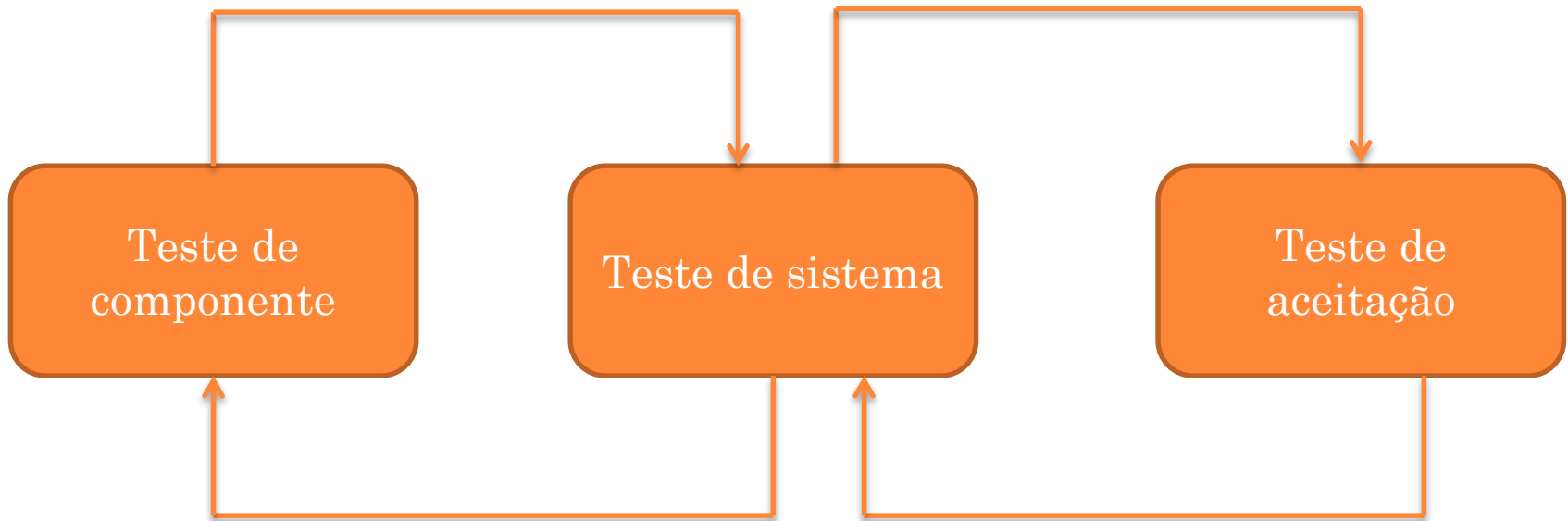
# PROJETO DE IMPLEMENTAÇÃO DE SOFTWARE

- Projeto de componente
  - Serviços são alocados aos componentes e as interfaces desses componentes são projetadas
- Projeto de estrutura de dados
  - As estrutura de dados utilizadas na implementação são projetadas detalhada e especificamente
- Projeto de algoritmo
  - Algoritmos usados para fornecer serviços são projetados detalhada e especificamente

# VALIDAÇÃO DE SOFTWARE

- Destina-se a mostrar que um sistema está em conformidade com sua especificação
- Envolve a verificação de processos, inspeção e revisão
- Defeitos de componente são encontrados no início do processo
- Problemas de interface descobertos na integração

# VALIDAÇÃO DE SOFTWARE



Estágios do processo de teste



# VALIDAÇÃO DE SOFTWARE

- Teste de componente
  - Componentes testados para garantir operação correta
  - Componentes testados independentemente
  - Podem ser simples (funções ou classes de objetos) ou grupos coerentes de entidades;

# VALIDAÇÃO DE SOFTWARE

## ○ Teste de sistema

- Os componente são integrados para compor o sistema;
- Relacionado com a busca de erros das iterações não previstas entre os componentes e as interfaces de componentes
- Verifica se o sistema atente aos requisitos funcionais e não funcionais e ao teste de propriedades emergentes do sistema
- Pode ser processo de vários estágios para sistemas grandes

# VALIDAÇÃO DE SOFTWARE

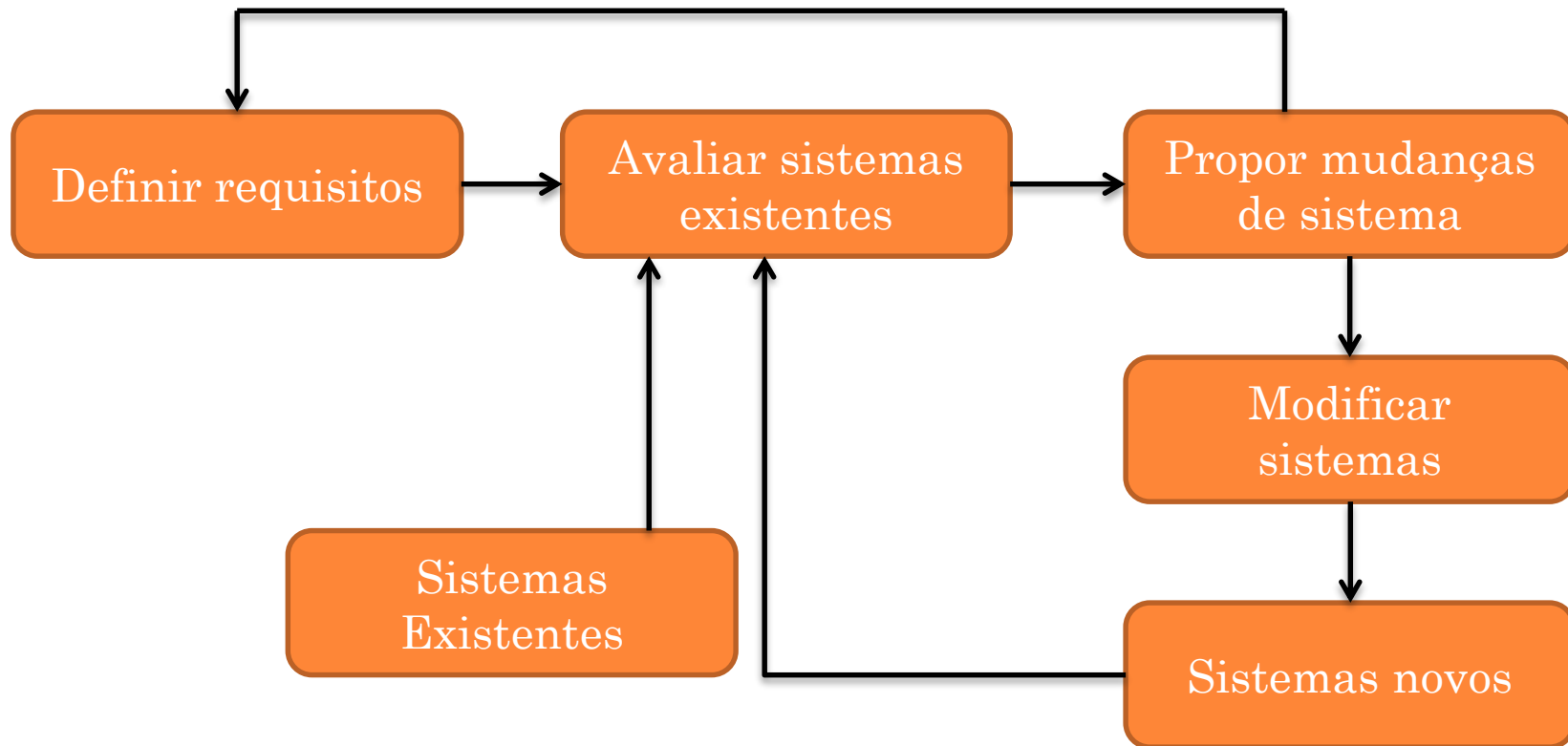
## ○ Teste de aceitação

- Estágio final do teste antes de ser aceito para uso;
- O sistema é testado com dados do cliente;
- Pode revelar erro e omissão de definição de requisitos;
- Pode revelar problema de requisitos onde os recursos não atendem realmente as necessidades do usuário ou o desempenho do sistema é inaceitável

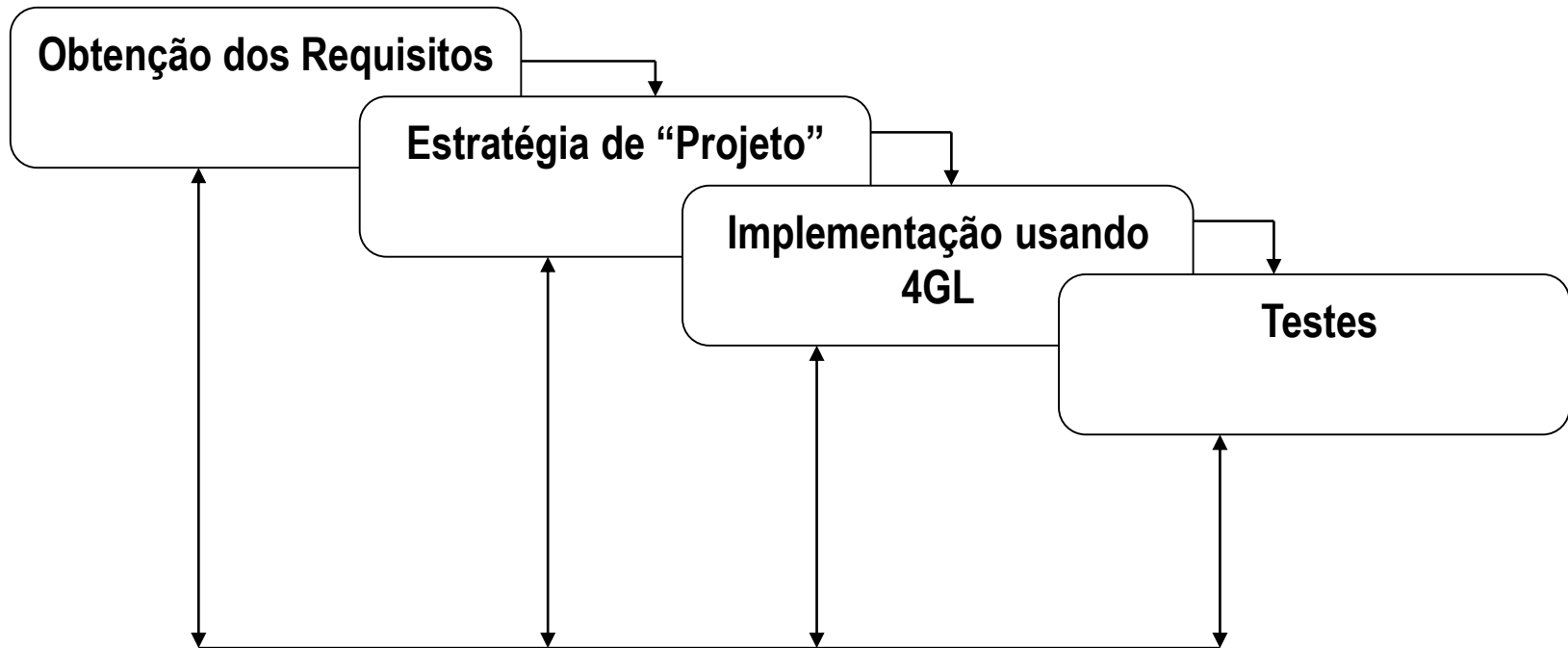
# EVOLUÇÃO DE SOFTWARE

- Mudanças podem acontecer a qualquer momento em um sistema
- São mais baratas as mudança em sistema que as mudanças de hardware
- Evolução em sistemas são constantes

# EVOLUÇÃO DE SOFTWARE



# TÉCNICAS DE 4º GERAÇÃO



# TÉCNICAS DE 4º GERAÇÃO

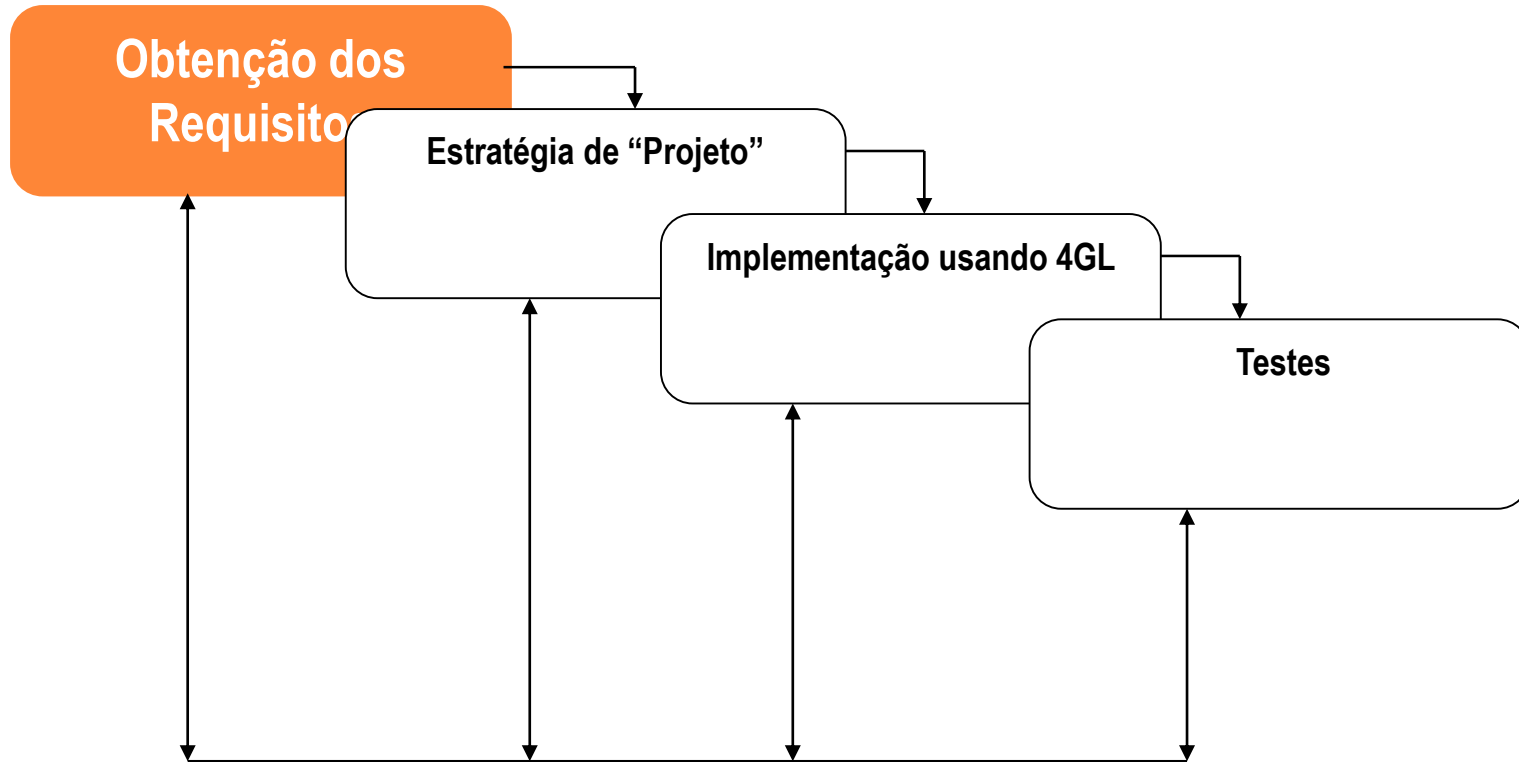
- Concentra-se na capacidade de se especificar o software a uma máquina em um nível que esteja próximo à linguagem natural.
- Engloba um conjunto de ferramentas de software que possibilitam que:
  - ⇒ o sistema seja especificado em uma linguagem de alto nível e
  - ⇒ o código fonte seja gerado automaticamente a partir dessas especificações

# TÉCNICAS DE 4º GERAÇÃO

- O ambiente de desenvolvimento inclui as ferramentas:
  - linguagens não procedimentais para consulta de banco de dados
  - geração de relatórios
  - manipulação de dados
  - interação e definição de telas
  - geração de códigos
  - capacidade gráfica de alto nível
  - capacidade de planilhas eletrônicas



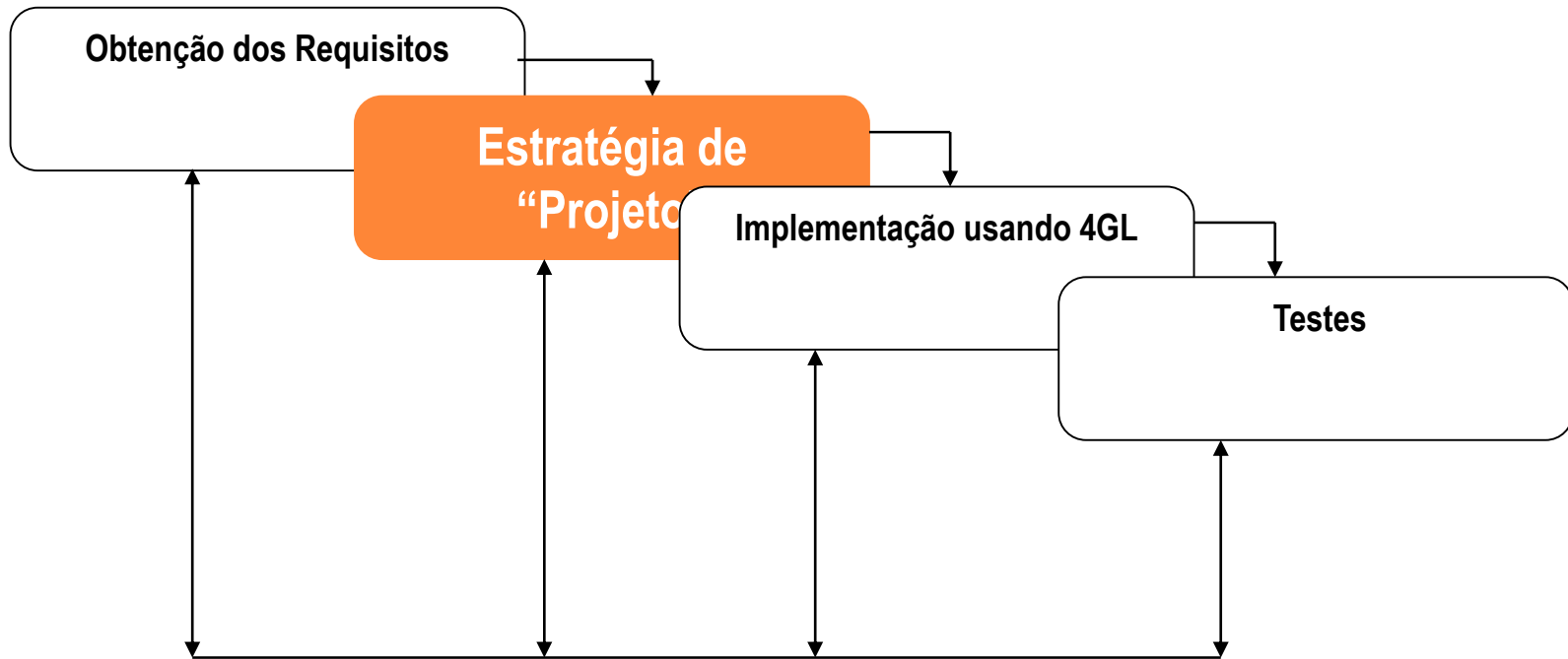
# ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO



## ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO

- **OBTENÇÃO DOS REQUISITOS:** o cliente descreve os requisitos os quais são traduzidos para um protótipo operacional
  - O cliente pode estar inseguro quanto aos requisitos
  - O cliente pode ser incapaz de especificar as informações de um modo que uma ferramenta 4GL possa consumir

# ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO

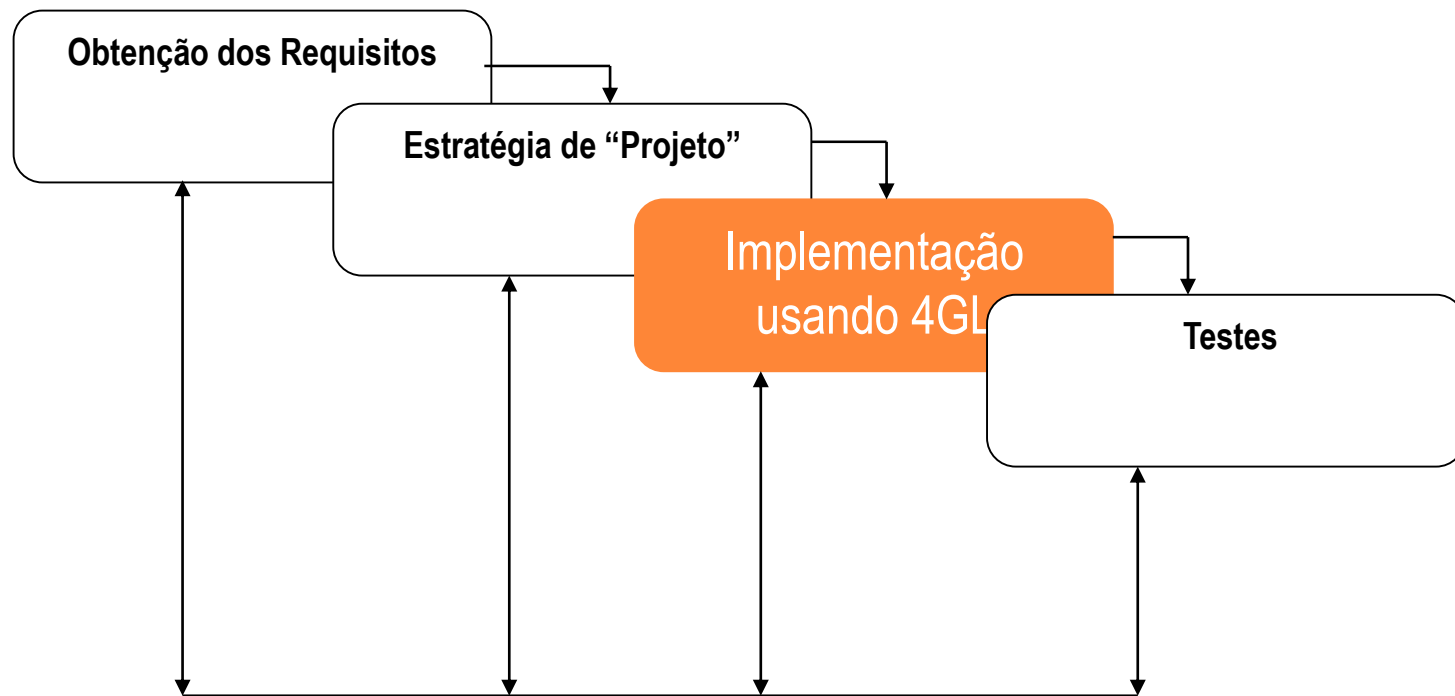


# ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO

## ○ ESTRATÉGIA DE "PROJETO":

- para pequenas aplicações é possível mover-se do passo de Obtenção dos Requisitos para o passo de Implementação
- Para grandes projetos é necessário desenvolver uma estratégia de projeto. De outro modo ocorrerão os mesmos problemas encontrados quando se usa abordagem convencional (baixa qualidade, manutenibilidade ruim, má aceitação do cliente)

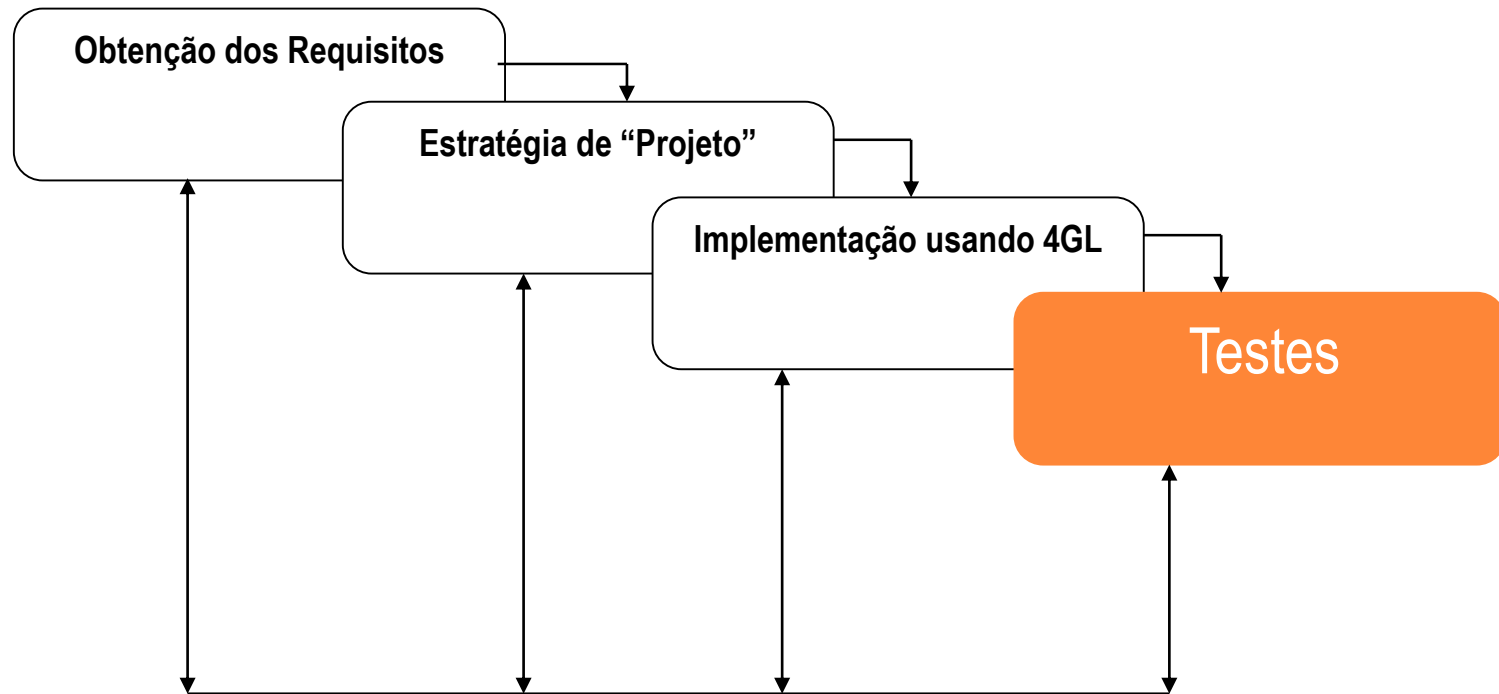
# ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO



# ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO

- IMPLEMENTAÇÃO USANDO 4GL:
  - os resultados desejados são representados de modo que haja geração automática de código.

# ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO



# ATIVIDADES - TÉCNICAS DE 4º GERAÇÃO

## ○ TESTE:

- o desenvolvedor deve efetuar testes e desenvolver uma documentação significativa.
- O software desenvolvido deve ser construído de maneira que a manutenção possa ser efetuada prontamente.



## FERRAMENTAS CASE

Computer-Aided Software Engineering  
(Engenharia de Software Auxiliada por  
Computador)

# FERRAMENTAS CASE

- Fornecem apoio ao processo de software pela automação de algumas atividades referentes ao processo:
  - Desenvolvimento de modelos gráficos
  - Dicionário de dados
  - Geração de interface com o usuário

# TIPOS DE FERRAMENTAS CASE

- Perspectiva funcional
  - As ferramentas são classificadas de acordo a sua função específica;
- Perspectiva de processo
  - As ferramentas são classificadas de acordo com as atividades de apoio que fornecem
- Perspectiva de integração
  - As ferramentas são classificadas de acordo com sua organização em unidades integradas
    - Apoiam uma ou mais atividades do processo

# FERRAMENTAS CASE

- Ferramentas que apoiam tarefas individuais do processo:
  - compilação de um programa
  - comparação de resultado de teste
- Workbenches que apoiam fases ou atividades do processo
  - Especificação
  - Projeto
- Ambientes que apoiam todo ou parte do processo.

# FERRAMENTAS CASE

