

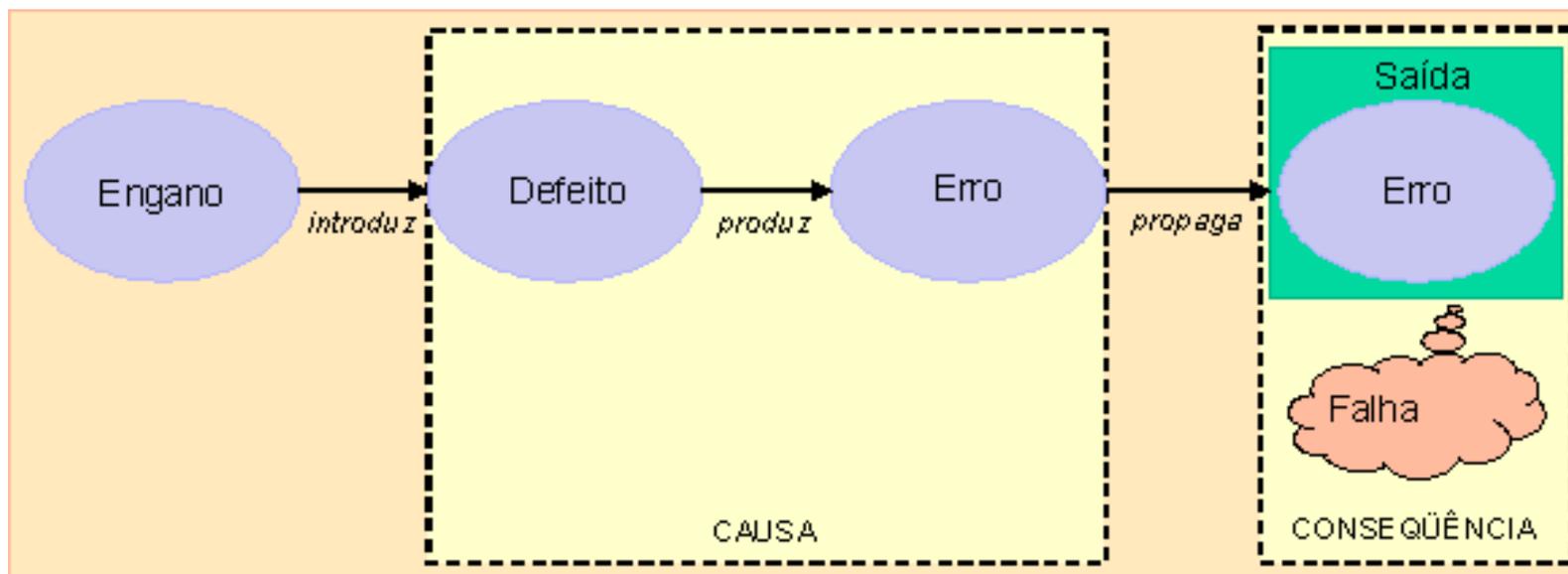
Teste de Software - Conceitos

Parte 02

Simone Souza

ICMC/USP

Recapitulando ...



Recapitulando ...

- Condições para uma falha ser observada: (RIP)
 - **Reachability** – alcançar o local do defeito
 - **Infection** – depois da execução, o estado do programa deve estar incorreto (**Erro**)
 - **Propragation** – o estado incorreto do programa deve se propagar, gerando uma saída incorreta (**Falha**)

Exemplo

```
public static int numZero(int[] x){
//Effects: if x == null throw NullPointerException
// else return the number of occurrences of 0 in x

int count = 0;
for(int i = 1; i < x.length; i++) {
    if (x[i] == 0)
        count++;
}
return count;
}
```

Exemplo

```
public static int numZero(int[] x){
//Effects: if x == null throw NullPointerException
// else return the number of occurrences of 0 in x

int count = 0;
for(int i = 1; i < x.length; i++) {
    if (x[i] == 0)
        count++;
}
return count;
}
```

x = [2 , 3, 0]

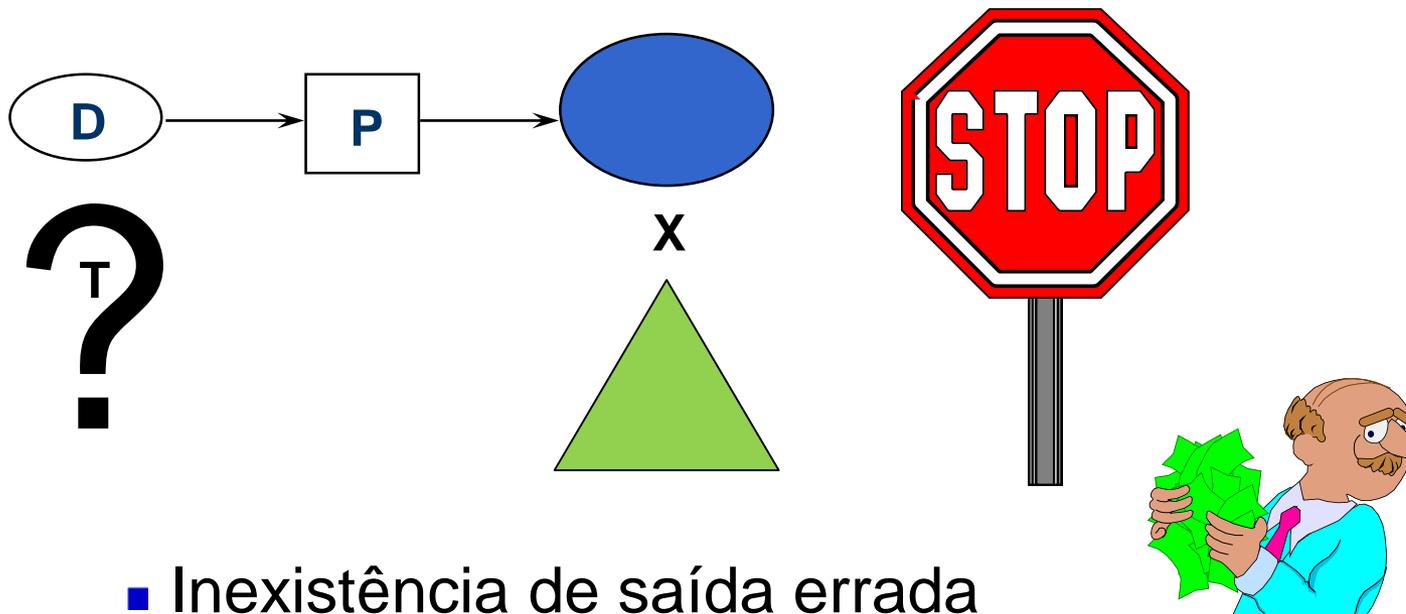
não propaga o erro

x = [0,3,0]

propaga o erro gerando
uma falha

Objetivo do Teste

- Revelar a presença de defeito



- Inexistência de saída errada
 - Software é de alta qualidade?
 - T é de baixa qualidade?

Exemplo: gerar casos de teste para testar o programa TRI

- O programa TRI lê três valores inteiros que representam os lados de um triângulo. A partir dos valores, o programa informa se os lados formam um triângulo isósceles, escaleno ou equilátero.
- Condição: a soma de dois lados tem que ser maior que o terceiro lado.

Exemplo

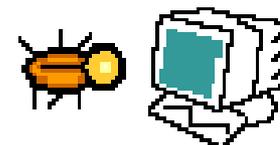
1. Existe c.t. para triângulo escaleno válido?
2. Existe c.t. para triângulo isósceles válido?
3. Existe c.t. para triângulo equilátero válido?
4. Existem pelo menos 3 c.t. para isósceles válido contendo a permutação dos mesmos valores?
5. Existe c.t. com um valor zero?
6. Existe c.t. com um valor negativo?
7. Existe c.t. em que a soma de 2 lados é igual ao terceiro lado?
8. Para o item 7, tem um c.t. para cada permutação de valores?

Exemplo

9. Existe c.t. em que a soma de 2 lados é menor que o terceiro lado?
10. Para o item 9, tem um c.t. para cada permutação de valores?
11. Existe c.t. para os 3 valores iguais a zero?
12. Existe c.t. com valores não inteiros?
13. Existe c.t. com número de valores errados, por exemplo, 2 valores ao invés de 3?
14. Para cada c.t. você especificou a saída esperada para a entrada projetada?

Exemplo

- Questões baseadas em erros encontrados em implementações do programa triângulo!!!



Técnicas de Teste

- A qualidade da atividade de teste está fortemente ligada à qualidade do **conjunto de casos de teste**.
 - *Como selecionar casos de teste?*

Técnicas de Teste

- Técnica Funcional (ou Caixa Preta)
- Técnica Estrutural (ou Caixa Branca)
- Teste Baseado em Defeitos (Teste de Mutação)
 - Possuem um conjunto de critérios de teste

Critério de Teste

- Maneira sistemática e planejada para conduzir os testes
- Fornece indicações a respeito de quais casos de teste utilizar de modo a aumentar as chances de revelar erros no programa
- Quando erros não forem revelados...
 - Estabelecer um nível elevado de confiança na correção do programa

Critério de Teste

■ Propriedades Mínimas:

- Garantir, do ponto de vista de **fluxo de controle**, a cobertura de **todos os desvios condicionais**
- Requerer, do ponto de vista de **fluxo de dados**, ao menos **um uso de todo resultado computacional**
- Requerer um conjunto de casos de teste **finito**

Critério de Teste

■ Critério de Geração:

- Procedimento para **escolher** um conjunto de casos de teste T para o teste do programa P
- T é **C-adequado** por construção ao critério

■ Exemplo (Função numZero()):

- Construir um conjunto T C-adequado a um critério que deseja executar todos os comandos pelo menos uma vez.

Exemplo

```
public static int numZero(int[] x){
//Effects: if x == null throw NullPointerException
// else return the number of occurrences of 0 in x

int count = 0;
for(int i = 1; i < x.length; i++) {
    if (x[i] == 0)
        count++;
}
return count;
}
```

Critério de Teste

- Critério de Adequação:
 - Predicado para **avaliar** um conjunto de casos de teste T no teste do programa P
 - Verifica se T satisfaz os requisitos de teste do critério

- Exemplo (Função numZero()):
 - T é adequado ao critério “executar todos os comandos pelo menos uma vez”.
$$T = \{ ([2], 0), ([], \text{“entrada nula”}) \}$$

Requisito de Teste

- Cada critério estabelece um **conjunto de requisitos** de teste específico.
- Casos de teste são selecionados de modo a satisfazer os requisitos estabelecidos pelo critério em questão.
- Dados um programa P , um conjunto de casos de teste T e um critério C , diz-se que:
 - T é **C -adequado** para o teste de P se T executar todos os requisitos de teste estabelecidos pelo critério C .

Exemplo

```
public static int numZero(int[] x) {  
    //Effects: if x == null throw NullPointerException  
    // else return the number of occurrences  
  
    int count = 0;  
    for(int i = 0; i < x.length; i++) {  
        if (x[i] == 0)  
            count++;  
    }  
    return count;  
}
```

Requisitos de teste para
o critério todos os
comandos:

1. count=0
2. i = 1
3. i < x.length
4. if (x[i] == 0)
5. count++
6. i++
7. return count

Exemplo

```
public static int numZero(int[] x) {
    //Effects: if x == null throw NullPointerException
    // else return the number of occurrences

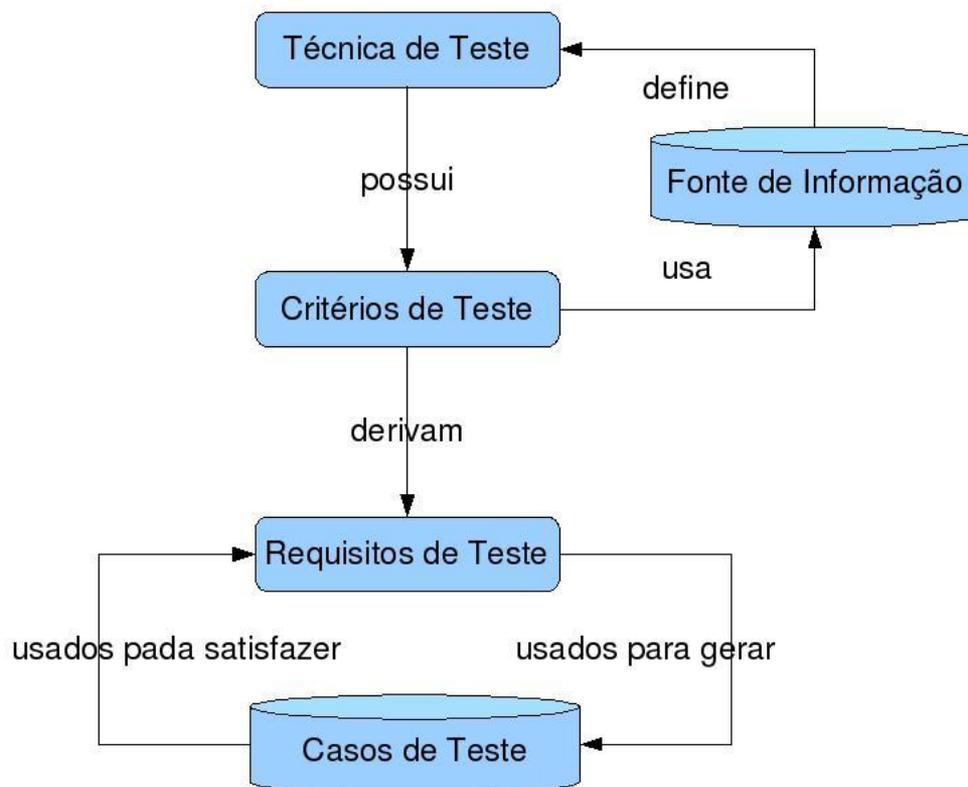
    int count = 0;
    for(int i = 1; i < x.length; i++) {
        if (x[i] == 0)
            count++;
    }
    return count;
}
```

T é C-adequado ao C1:
 $T = \{ ([2],0), ([2,2,0],1) \}$

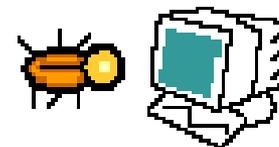
Requisitos de teste para o critério todos os comandos (C1):

1. count=0
2. i = 1
3. i < x.length
4. if (x[i] == 0)
5. count++
6. i++
7. return count

Relacionamento dos termos

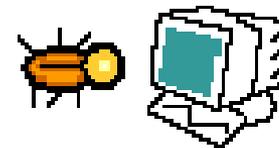


Teste de Software



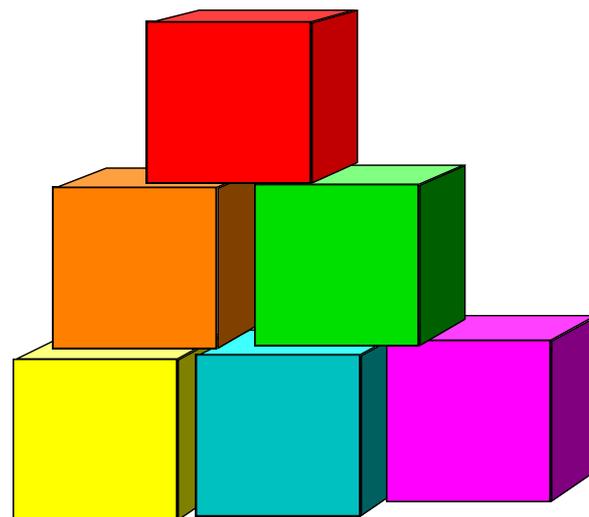
- Em que momento testar?
 - **Níveis de teste:**
 - Qual é o **objeto** de teste?
 - Qual é o **objetivo** do teste?

Nível de Teste

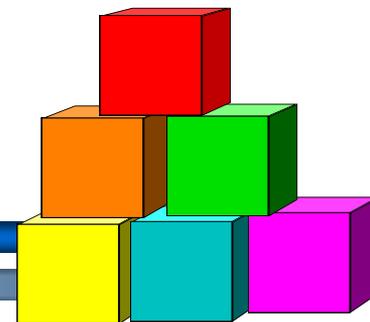


■ Níveis de teste – Objeto de teste:

- Teste de Unidade
- Teste de Integração
- Teste de Sistema



Teste de Unidade

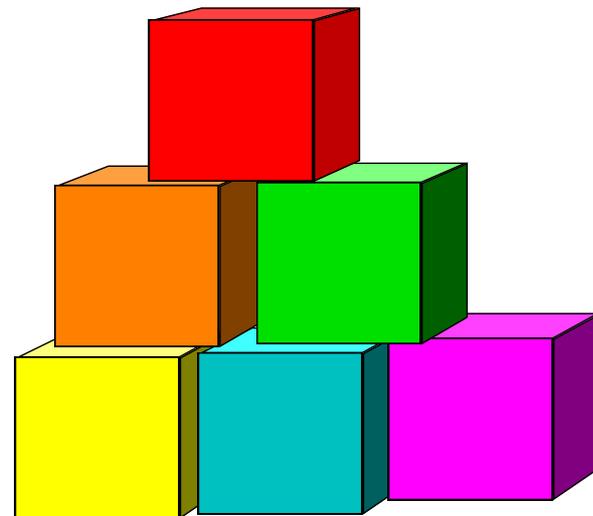


- Cada **unidade** do programa é explorado separadamente...
 - Identificar erros de lógica e de implementação.

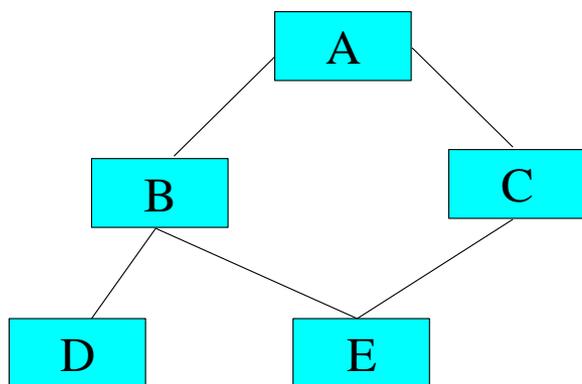
- Critérios de teste utilizados:
 - **Critérios de Fluxo de Controle**
 - **Critérios de Fluxo de Dados**
 - **Critério Análise de Mutantes**

Fases do Teste

- Teste de Unidade
- **Teste de Integração**
- Teste de Sistema

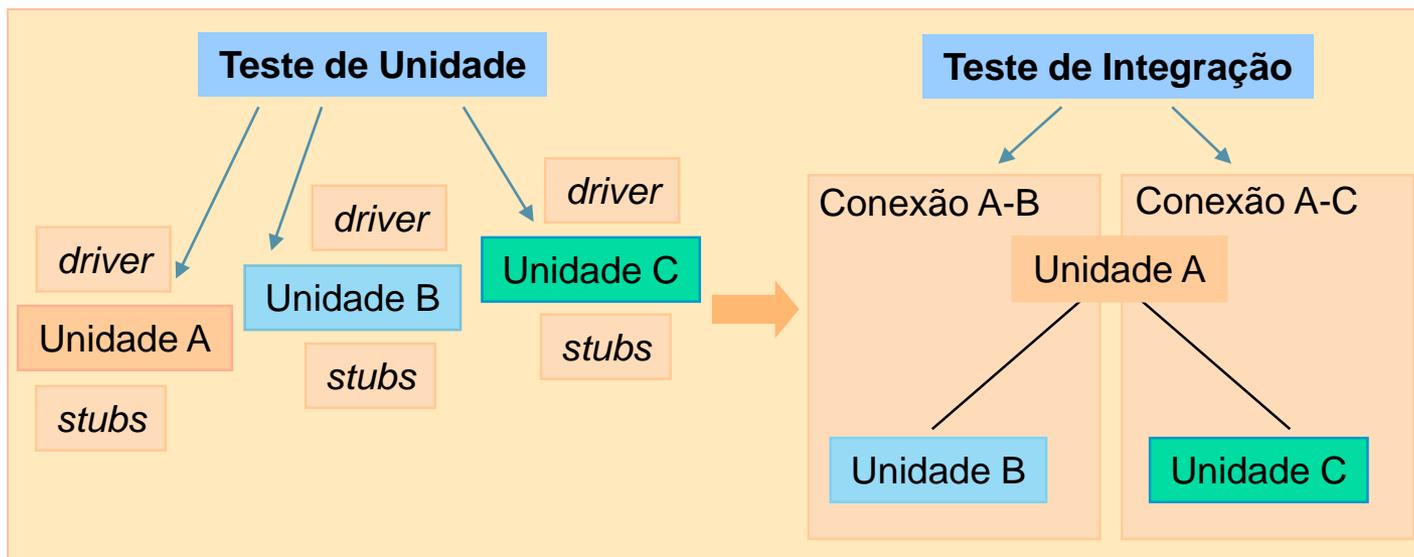


Teste de Integração

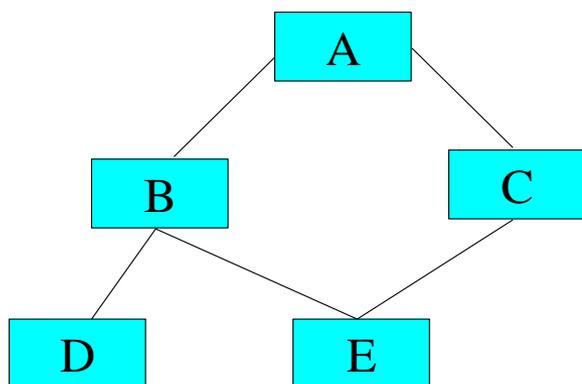
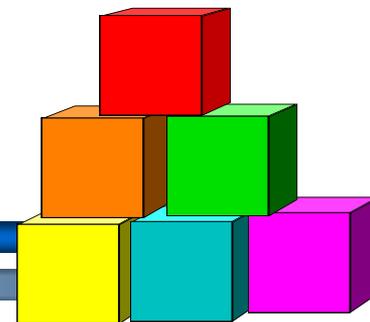


- Testar a comunicação entre as unidades funcionais do sistema
- Formas de comunicação:
 - **parâmetros de entrada.**
 - **Retorno** por meio de parâmetros de entrada (**passagem por referência**).
 - **Variáveis globais.**
 - **Comandos *return*.**

Teste de Integração



Teste de Integração



- Como testar a conexão entre A-B:
 - Testar pontos em que A chama B
 - Testar pontos dentro de B que usam dados enviados por A (relacionados com as variáveis de comunicação).

```

/* 01 */      {
/* 01 */      char  achar;
/* 01 */      int  length, valid_id;
/* 01 */      length = 0;
/* 01 */      printf ("Identificador: ");
/* 01 */      achar = fgetc (stdin);
/* 01 */      valid_id = valid_s(achar);
/* 01 */      if (valid_id)
/* 02 */          length = 1;
/* 03 */      achar = fgetc (stdin);
/* 04 */      while (achar != '\n')
/* 05 */      {
/* 05 */          if (!(valid_f(achar)))
/* 06 */              valid id = 0;
/* 07 */      }
/* 07 */      printf ("%d", length);
/* 08 */      if (valid_id)
/* 09 */          printf ("Valido");
/* 10 */      else
/* 10 */          printf ("Invalido");
/* 11 */      }

```

```

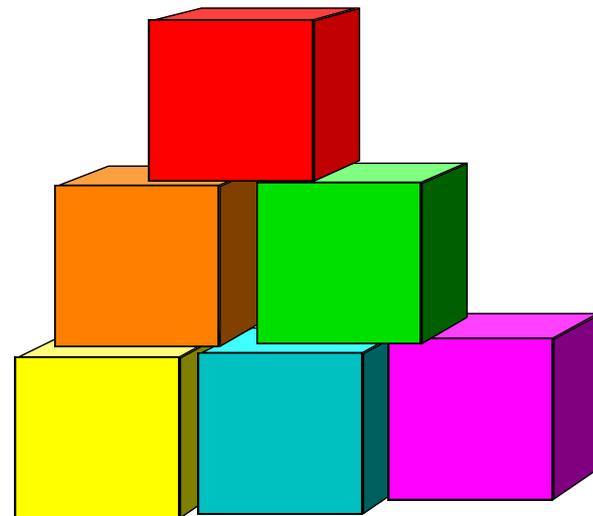
int valid_s (char ch) {
/*1*/  if (((ch >= 'A') && (ch < 'Z')) || ((ch >= 'a') && (ch <= 'z')))
/*2*/      return (1);
/*3*/  else return (0); }

int valid_f (char ch) {
/*1*/  if (((ch >= 'A') && (ch <= 'Z')) || ((ch >= 'a') && (ch <= 'z'))
|| ((ch > '0') && (ch <= '9')))
/*2*/      return (1);
/*3*/  else return (0); }

```

Fases do Teste

- Teste de Unidade
- Teste de Integração
- Teste de Sistema



Teste de Sistema

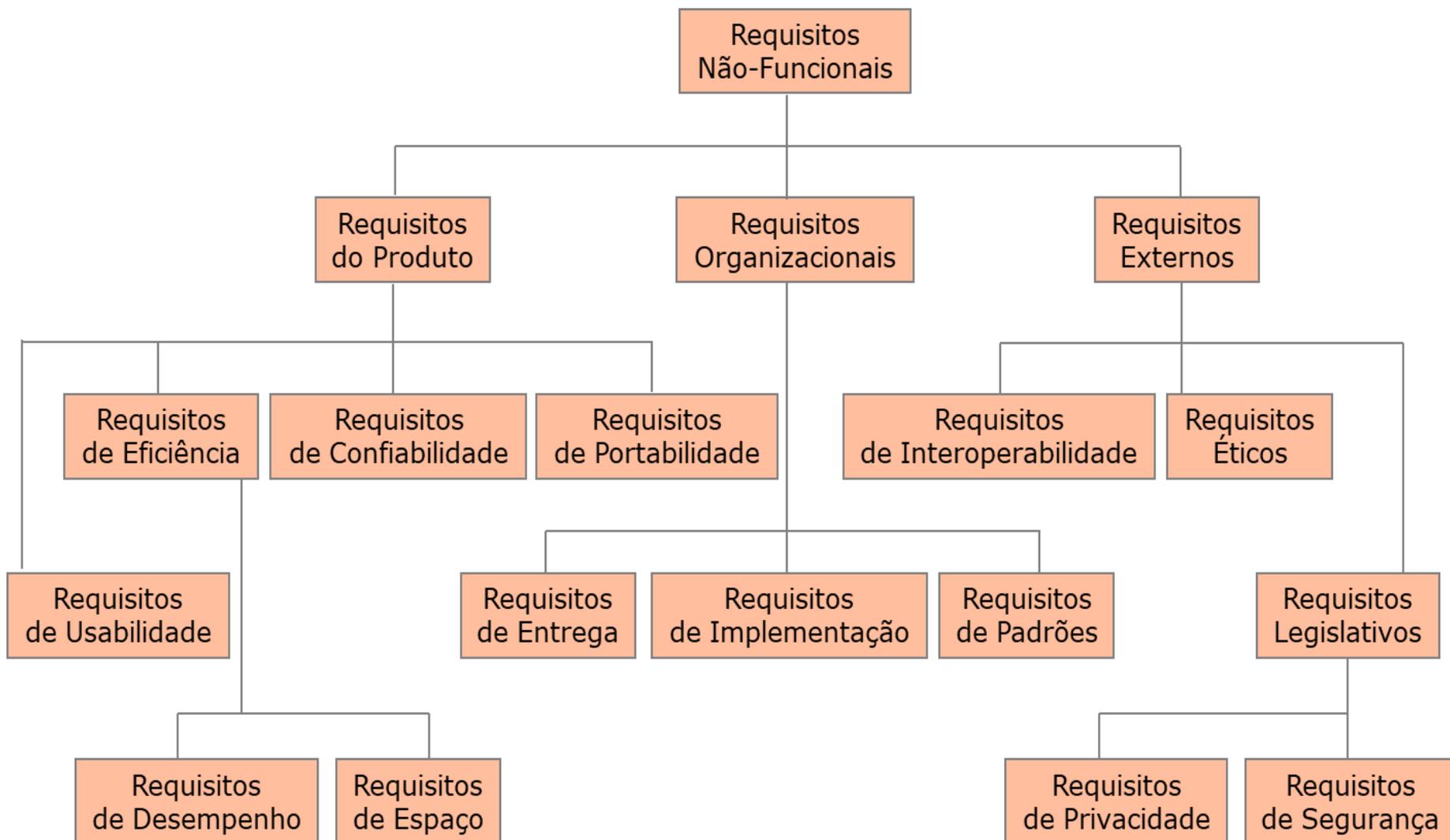
- Após integrar (e testar!) todos os componentes do software...
 - O software deve ser capaz de funcionar apropriadamente no ambiente para o qual foi projetado.

Teste de Sistema

- Adequado para testar requisitos não funcionais do software:
 - Velocidade
 - Confiabilidade
 - Segurança
 - ...



Requisitos não funcionais

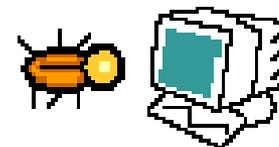


Exemplo de testes de sistema

- Teste de Recuperação e Estresse
- Teste de Segurança
- Teste de Desempenho
- Teste de Usabilidade



Teste de Software



- Em que momento testar?
 - **Níveis de teste:**
 - Qual é o **objeto** de teste (Fases)?
 - Qual é o **objetivo** do teste?

Nível do Teste - Objetivo

- Testar um objetivo ou mais objetivos
 - Os objetivos podem se relacionar com as etapas do teste
- Exemplo de objetivos de teste:
 - Funcionalidades, qualidade esperada, medir a confiabilidade e vulnerabilidade, a aceitação do software, a instalação...

Nível do Teste - Objetivo

- Alguns tipos de teste (baseado em objetivo):
 - Teste de conformidade (conformance testing)
 - Teste de aceitação
 - Teste alfa e beta
 - Teste de instalação
 - Teste de regressão
 - Teste de configuração
 - Todos os testes de sistema são objetivos

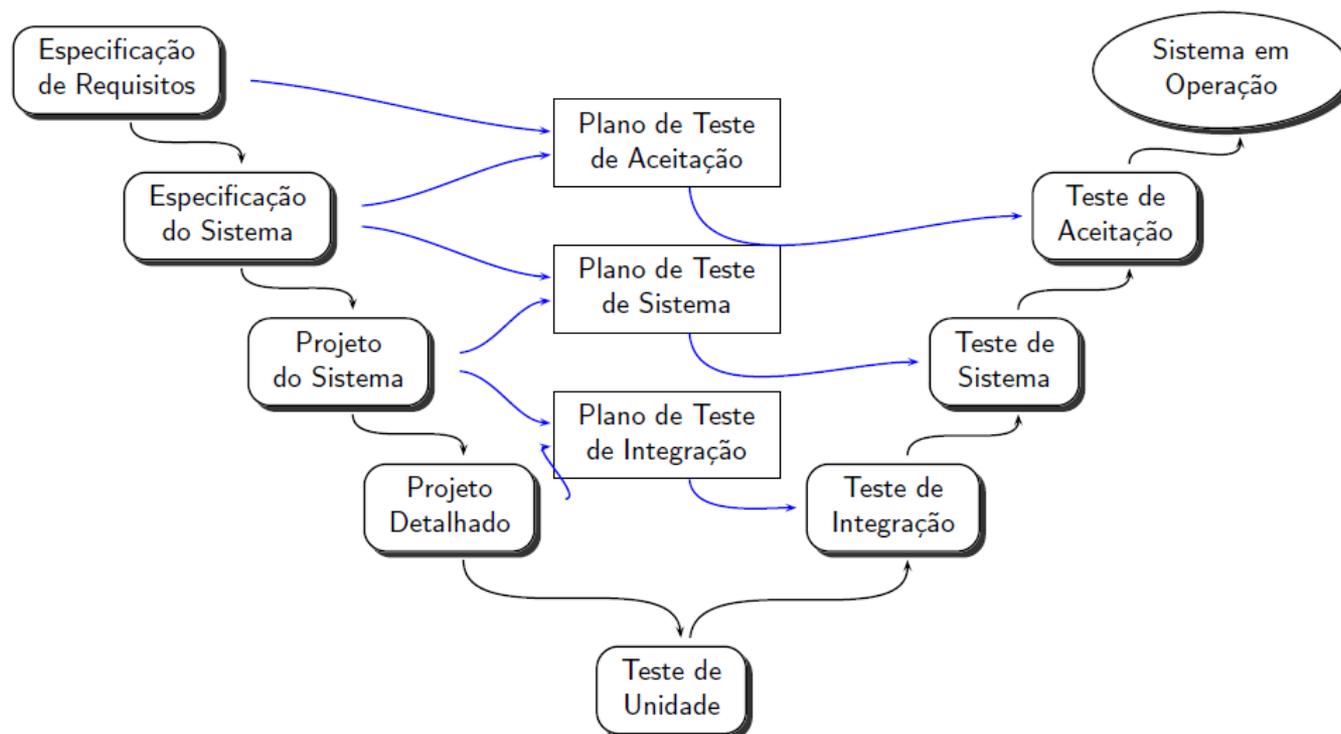
Teste de Software

- Como organizar a atividade de teste ao longo do desenvolvimento?
 - Planejar
 - Documentar



Teste de Software e Fases do Desenvolvimento

Modelo V (V-Model)



Etapas de Teste

- Planejamento.
 - Desenvolvimento **Plano de Testes.**
- Projeto de casos de teste.
 - Seleção e aplicação de critérios.
- Execução do programa.
- Análise de resultados.

Processo de teste

- A realização da atividade de teste deve seguir um **processo**
 - Processo deve estar integrado ao ciclo de vida do software
 - Processo deve considerar:
 - Pessoas, Ferramentas, Políticas e Medições

Processo de teste

- Decisões do processo:
 - Pessoal interno ou externo ao projeto?
 - Estimar custo e esforço da atividade de teste
 - Quando finalizar os testes?
 - Considerar os riscos associados e questões críticas
 - Como serão gerados os casos de teste?
 - Ambiente de teste
 - Ferramentas, local dos testes, pessoas, registros...
 - Rastreamento de defeitos
 - ...

Processo de teste

- O processo deve incluir planejamento, monitoramento e controle

- Sugestão de processo:
 - Norma ISO/IEC 29119-1

Norma ISO/IEC – 29119-1

