

SSC0721 – Teste e inspeção de software


# SSC0721 – Teste e inspeção de software

## Introdução

Márcio Delamaro  
(Slides de: Auri M. R. Vincenzi)

SSC – ICMC – USP

março de 2009



SSC0721 – Teste e inspeção de software

## Organização

- Teste
  - Desafios do Teste
- Casos de Teste
  - Projeto de Casos de Teste
  - Entradas
  - Saídas
  - Oráculo
  - Ordem de Execução
- Tipos de Teste
  - Técnicas de Teste
- Fases de Teste
  - Teste de Unidade
  - Teste de Integração
  - Teste de Sistema
  - Teste de Aceitação
- Limitações do Teste
- Automatização
- Algumas definições
- Exercícios
- Resumo
- Referências

SSC0721 – Teste e inspeção de software

## Teste

- Teste
  - Desafios do Teste
  - Casos de Teste
    - Projeto de Casos de Teste
    - Entradas
    - Saídas
    - Oráculo
    - Ordem de Execução
  - Tipos de Teste
    - Técnicas de Teste
  - Fases de Teste
    - Teste de Unidade
    - Teste de Integração
    - Teste de Sistema
    - Teste de Aceitação
  - Limitações do Teste
  - Automatização
  - Algumas definições
  - Exercícios
  - Resumo
  - Referências

SSC0721 – Teste e inspeção de software

## Teste

### O que é teste?

- De acordo com a norma IEEE Standard 610.12-1990 (IEEE, 1990):  
“The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.”
- Craig e Jaskiel (2002) apresentam outra definição:  
“Testing is a concurrent lifecycle process of engineering, using and maintaining testware in order to measure and improve the quality of the software being tested.”

SSC0721 – Teste e inspeção de software

## Desafios do Teste

- Teste
  - Desafios do Teste
  - Casos de Teste
    - Projeto de Casos de Teste
    - Entradas
    - Saídas
    - Oráculo
    - Ordem de Execução
  - Tipos de Teste
    - Técnicas de Teste
  - Fases de Teste
    - Teste de Unidade
    - Teste de Integração
    - Teste de Sistema
    - Teste de Aceitação
  - Limitações do Teste
  - Automatização
  - Algumas definições
  - Exercícios
  - Resumo
  - Referências

SSC0721 – Teste e inspeção de software

## Desafios do Teste

### Desafios

- Alguém já testou algum produto de software?

SSC0721 – Teste e inspeção de software

## Desafios do Teste

### Desafios

- Alguém já testou algum produto de software?
- Quais foram os maiores desafios?

SSC0721 – Teste e inspeção de software

## Desafios do Teste

### Desafios

- Alguém já testou algum produto de software?
- Quais foram os maiores desafios?
- Alguns problemas comuns são:

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

▶ Muitas combinações de entrada para serem exercitadas.

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

▶ Muitas combinações de entrada para serem exercitadas.

▶ Dificuldade em determinar os resultados esperados para cada caso de teste.

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

▶ Muitas combinações de entrada para serem exercitadas.

▶ Dificuldade em determinar os resultados esperados para cada caso de teste.

▶ Requisitos do software inexistentes ou que mudam rapidamente.

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

▶ Muitas combinações de entrada para serem exercitadas.

▶ Dificuldade em determinar os resultados esperados para cada caso de teste.

▶ Requisitos do software inexistentes ou que mudam rapidamente.

▶ Não há tempo suficiente para o teste.

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

▶ Muitas combinações de entrada para serem exercitadas.

▶ Dificuldade em determinar os resultados esperados para cada caso de teste.

▶ Requisitos do software inexistentes ou que mudam rapidamente.

▶ Não há tempo suficiente para o teste.

▶ Não há treinamento no processo de teste.

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

▶ Muitas combinações de entrada para serem exercitadas.

▶ Dificuldade em determinar os resultados esperados para cada caso de teste.

▶ Requisitos do software inexistentes ou que mudam rapidamente.

▶ Não há tempo suficiente para o teste.

▶ Não há treinamento no processo de teste.

▶ Não há ferramenta de apoio.

SSC0721 – Teste e inspeção de software

Desafios do Teste

Desafios

▶ Alguém já testou algum produto de software?

▶ Quais foram os maiores desafios?

▶ Alguns problemas comuns são:

▶ Não há tempo para o teste exaustivo.

▶ Muitas combinações de entrada para serem exercitadas.

▶ Dificuldade em determinar os resultados esperados para cada caso de teste.

▶ Requisitos do software inexistentes ou que mudam rapidamente.

▶ Não há tempo suficiente para o teste.

▶ Não há treinamento no processo de teste.

▶ Não há ferramenta de apoio.

▶ Gerentes que desconhecem teste ou que não se preocupam com qualidade.

SSC0721 – Teste e inspeção de software

- Casos de Teste

Teste  
Desafios do Teste  
**Casos de Teste**  
Entradas  
Saídas  
Oráculo  
Ordem de Execução  
Tipos de Teste  
Técnicas de Teste  
Fases de Teste  
Teste de Unidade  
Teste de Integração  
Teste de Sistema  
Teste de Aceitação  
Limitações do Teste  
Automatização  
Algumas definições  
Exercícios  
Resumo  
Referências

SSC0721 – Teste e inspeção de software

- Casos de Teste
- Projeto de Casos de Teste

## Projeto de Casos de Teste

- ▶ O segredo do sucesso do teste está no projeto dos casos de teste.
- ▶ Segundo (Pressman, 2006):  
“The design of tests for software and other engineering products can be as challenging as the initial design of the product itself. Yet ... software engineers often treat testing as an afterthought, developing test cases that ‘feel right’ but have little assurance of being complete. Recalling the objectives of testing, we must design tests that have the highest likelihood of finding the most errors with a minimum amount of time and effort.”

SSC0721 – Teste e inspeção de software

- Casos de Teste
- Projeto de Casos de Teste

## Partes de um Caso de Teste

- ▶ Um caso de teste bem projetado é dividido em três partes:
  - ▶ Entradas (*inputs*);
  - ▶ Saídas Esperadas (*outputs*);
  - ▶ Ordem de execução (*order of execution*).

SSC0721 – Teste e inspeção de software

- Casos de Teste
- Entradas

## Partes de um Caso de Teste – entradas

- ▶ Geralmente identificadas como dados fornecidos via teclado para o programa executar.
- ▶ Entretanto, os dados de entrada podem ser fornecidos por outros meios, tais como:
  - ▶ Dados oriundos de outro sistema que servem de entrada para o programa em teste;
  - ▶ Dados fornecidos por outro dispositivo;
  - ▶ Dados lidos de arquivos ou banco de dados;
  - ▶ O estado do sistema quando os dados são recebidos;
  - ▶ O ambiente no qual o programa está executando.

SSC0721 – Teste e inspeção de software

- Casos de Teste
- Saídas

## Partes de um Caso de Teste – saídas esperadas

- ▶ As saídas também podem ser produzidas de diferentes formas.
- ▶ A mais comum é aquela apresentada na tela do computador.
- ▶ Além dessa, as saídas podem ser enviadas para:
  - ▶ Outro sistema interagindo com o programa em teste;
  - ▶ Dados enviados para um dispositivo externo;
  - ▶ Dados escritos em arquivos ou banco de dados;
  - ▶ O estado do sistema ou o ambiente de execução podem ser alterados durante a execução do programa.

SSC0721 – Teste e inspeção de software

- Casos de Teste
- Oráculo

## Partes de um Caso de Teste – oráculo (1)

- ▶ Todas as formas de entrada e saída são relevantes.

SSC0721 – Teste e inspeção de software

- Casos de Teste
- Oráculo

## Partes de um Caso de Teste – oráculo (1)

- ▶ Todas as formas de entrada e saída são relevantes.
- ▶ Durante o projeto de um caso de teste, determinar correção da saída esperada é função do **oráculo** (*oracle*).

SSC0721 – Teste e inspeção de software

- Casos de Teste
- Oráculo

## Partes de um Caso de Teste – oráculo (1)

- ▶ Todas as formas de entrada e saída são relevantes.
- ▶ Durante o projeto de um caso de teste, determinar correção da saída esperada é função do **oráculo** (*oracle*).
- ▶ Oráculo corresponde a um mecanismo (programa, processo ou dados) que indica ao projetista de casos de testes a saída obtida para um caso de teste é aceitável ou não.

## Partes de um Caso de Teste – oráculo (2)

- ▶ Beizer (1990) lista cinco tipos de oráculos:
  - ▶ Oráculo "Kiddie" - simplesmente execute o programa e observe sua saída. Se ela parecer correta deve estar correta.
  - ▶ Conjunto de Teste de Regressão - execute o programa e compare a saída obtida com a saída produzida por uma versão mais antiga do programa.
  - ▶ Validação de Dados - execute o programa e compare a saída obtida com uma saída padrão determinada por uma tabela, fórmula ou outra definição aceitável de saída válida.
  - ▶ Conjunto de Teste Padrão - execute o programa com um conjunto de teste padrão que tenha sido previamente criado e validado. Utilizado na validação de compiladores, navegadores Web e processadores de SQL.
  - ▶ Programa existente - execute o programa em teste e o programa existente com o mesmo caso de teste e compare as saídas. Semelhante ao teste de regressão.

## Partes de um Caso de Teste – ordem de execução (1)

- ▶ Existem dois estilos de projeto de casos de teste relacionados com a ordem de execução:
  - ▶ Casos de teste em cascata.
  - ▶ Casos de teste independentes.

## Partes de um Caso de Teste – ordem de execução (2)

- ▶ Casos de teste em cascata - quando os casos de teste devem ser executados um após o outro, em uma ordem específica. O estado do sistema deixado pelo primeiro caso de teste é reaproveitado pelo segundo e assim sucessivamente. Por exemplo, considere o teste de uma base de dados:
  - ▶ criar um registro
  - ▶ ler um registro
  - ▶ atualizar um registro
  - ▶ ler um registro
  - ▶ apagar um registro
  - ▶ ler o registro apagado
- ▶ Vantagem - casos de testes tendem a ser pequenos e simples. Fáceis de serem projetados, criados e mantidos.
- ▶ Desvantagem - se um caso de teste falhar os casos de teste subsequentes também podem falhar.

## Partes de um Caso de Teste – ordem de execução (3)

- ▶ Casos de teste independentes - Cada caso de teste é inteiramente autocontido.
  - ▶ Vantagem - casos de teste podem ser executados em qualquer ordem.
  - ▶ Desvantagem - casos de teste tendem a ser grandes e complexos, mais difíceis de serem projetados, criados e mantidos.

Teste  
 Desafios do Teste  
 Casos de Teste  
 Projeto de Casos de Teste  
 Entradas  
 Saídas  
 Oráculo  
 Ordem de Execução  
**Tipos de Teste**  
**Técnicas de Teste**  
 Fases de Teste  
 Teste de Unidade  
 Teste de Integração  
 Teste de Sistema  
 Teste de Aceitação  
 Limitações do Teste  
 Automação  
 Algumas definições  
 Exercícios  
 Resumo  
 Referências

## Tipos de Teste (1)

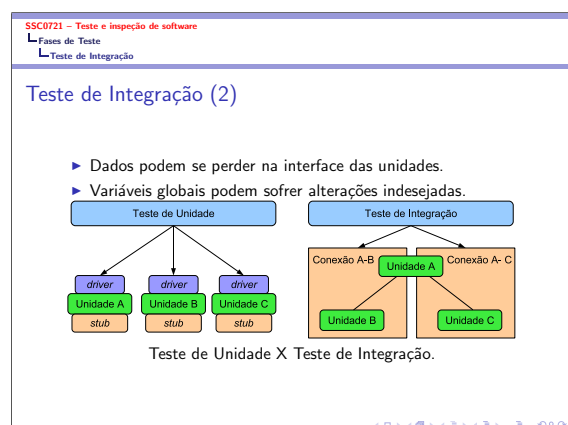
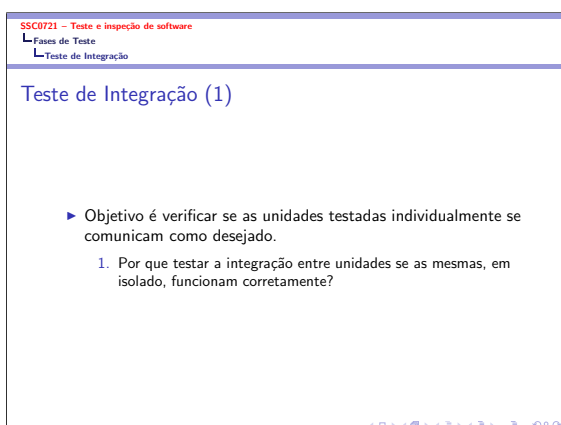
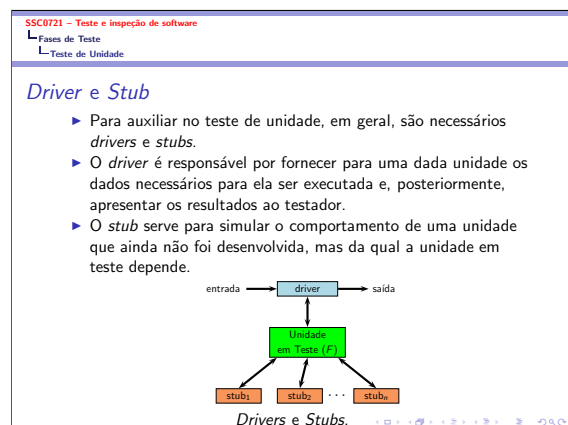
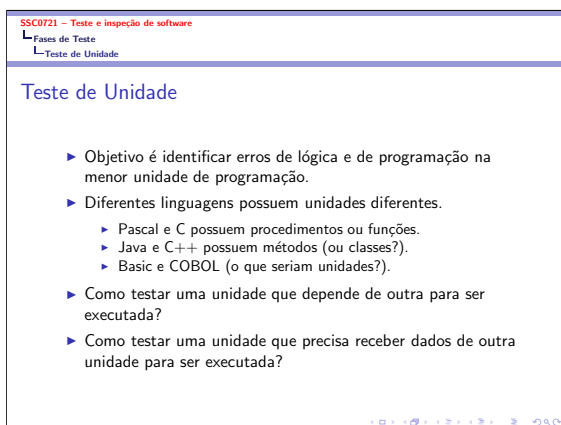
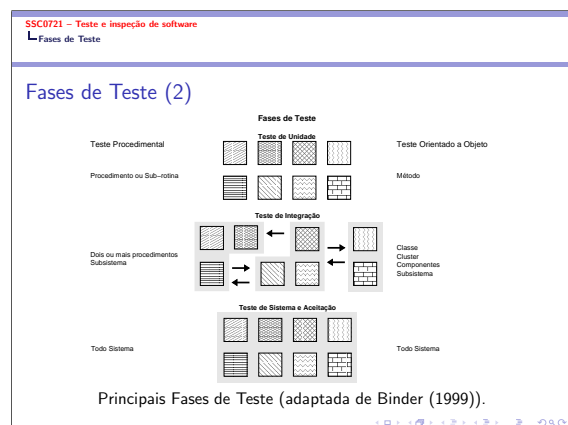
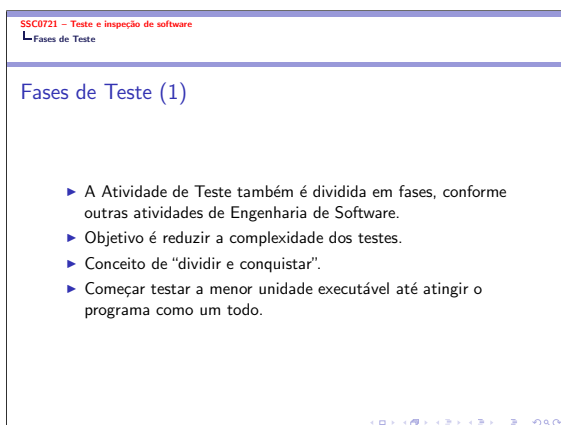
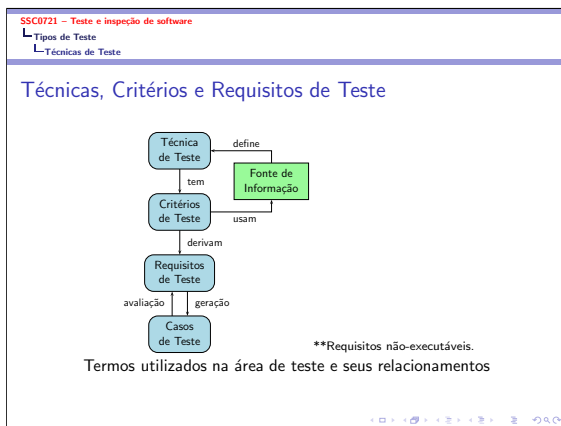
- ▶ Diferentes tipos de testes podem ser utilizados para verificar se um programa se comporta como o especificado.
- ▶ Basicamente, os testes podem ser classificados em teste funcional, teste estrutural ou teste baseado em defeito.
- ▶ Esses **tipos de teste** correspondem às chamadas **técnicas de teste**.

## Tipos de Teste (2)

- ▶ A técnica de teste é definida pelo tipo de informação utilizada para realizar o teste.
  - ▶ **Técnica funcional** - os testes são baseados exclusivamente na especificação de requisitos do programa. Nenhum conhecimento de como o programa está implementado é requerido.
  - ▶ **Técnica estrutural** - os testes são baseados na estrutura interna do programa, ou seja, na implementação do mesmo.
  - ▶ **Técnica baseada em defeito** - os testes são baseados em informações histórica sobre defeitos cometidos freqüentemente durante o processo de desenvolvimento de software.

## Tipos de Teste (3)

- ▶ Cada técnica de teste possui um conjunto de **critérios de teste**.
- ▶ Os critérios sistematizam a forma como **requisitos de teste** devem ser produzidos a partir da fonte de informação disponível (especificação de requisitos, código fonte, histórico de defeitos, dentre outras).
- ▶ Os requisitos de teste são utilizados para:
  - ▶ Gerar casos de teste.
  - ▶ Avaliar a qualidade de um conjunto de teste existente.



## Teste de Sistema

- ▶ Objetivo é verificar se o programa em si interage corretamente com o sistema para o qual foi projetado. Isso inclui, por exemplo, o SO, banco de dados, hardware, manual do usuário, treinamento, etc.
- ▶ Corresponde a um teste de integração de mais alto nível.
- ▶ Inclui teste de funcionalidade, usabilidade, segurança, confiabilidade, disponibilidade, performance, backup/restauração, portabilidade, entre outros (Norma ISO-IEC-9126 para mais informações (ISO/IEC, 1991))

## Teste de Aceitação

- ▶ Objetivo é verificar se o programa desenvolvido atende as exigências do usuário.

Teste  
 Desafios do Teste  
 Casos de Teste  
 Projeto de Casos de Teste  
 Entradas  
 Saídas  
 Oráculo  
 Ordem de Execução  
 Tipos de Teste  
 Técnicas de Teste  
 Fases de Teste  
 Teste de Unidade  
 Teste de Integração  
 Teste de Sistema  
 Teste de Aceitação  
**Limitações do Teste**  
 Automação  
 Algumas definições  
 Exercícios  
 Resumo  
 Referências

## Limitações do Teste (1)

Observe o exemplo abaixo extraído de Binder (1999):

```

1 int blech(int j) {
2   j = j - 1; // deveria ser j = j + 1
3   j = j / 30000;
4   return j;
5 }
  
```

Código fonte: ./src/blech.java

- ▶ Considerando o tipo inteiro com 16 bits (2 bytes) – o menor valor possível seria -32.768 e o maior seria 32.767, resultando em 65.536 valores diferentes possíveis.
- ▶ Haverá tempo suficiente para se criar 65.536 casos de teste? E se os programas forem maiores? Quantos casos de teste serão necessários?

## Limitações do Teste (2)

```

1 int blech(int j) {
2   j = j - 1; // deveria ser j = j + 1
3   j = j / 30000;
4   return j;
5 }
  
```

Código fonte: ./src/blech.java

Quais valores escolher?

Entrada (j)	Saída Esperada	Saída Obtida
1	0	0
42	0	0
40000	1	1
-64000	-2	-2

Quais valores de entrada revelam o erro no programa acima?

## Limitações do Teste (3)

- ▶ Os casos de testes anteriores não revelam o erro.

## Limitações do Teste (3)

- ▶ Os casos de testes anteriores não revelam o erro.
- ▶ Somente quatro valores do intervalo de entrada válido revelam o erro:

## Limitações do Teste (3)

- ▶ Os casos de testes anteriores não revelam o erro.
- ▶ Somente quatro valores do intervalo de entrada válido revelam o erro:
- ▶ Os valores abaixo revelam o erro:

Entrada (j)	Saída Esperada	Saída Obtida
-30000	0	-1
-29999	0	-1
30000	1	0
29999	1	0

## Limitações do Teste (3)

- ▶ Os casos de testes anteriores não revelam o erro.
- ▶ Somente quatro valores do intervalo de entrada válido revelam o erro:
- ▶ Os valores abaixo revelam o erro:

Entrada (i)	Saída Esperada	Saída Obtida
-30000	0	-1
-29999	0	-1
30000	1	0
29999	1	0

- ▶ Qual a chance desses valores serem selecionados???

Teste  
Desafios do Teste  
Casos de Teste  
Projeto de Casos de Teste  
Entradas  
Saídas  
Oráculo  
Ordem de Execução  
Tipos de Teste  
Técnicas de Teste  
Fases de Teste  
Teste de Unidade  
Teste de Integração  
Teste de Sistema  
Teste de Aceitação  
Limitações do Teste  
**Automatização**  
Algumas definições  
Exercícios  
Resumo  
Referências

## Automatização

- ▶ A atividade de teste é cara e consome muito tempo.
- ▶ Automatizar significa:
  - ▶ Diminuir custos
  - ▶ Diminuir erros
  - ▶ Facilitar teste de regressão
- ▶ Atividades intelectuais × atividades braçais.
- ▶ Suporte a técnicas de teste
- ▶ Geração de dados de teste
- ▶ Oráculos

Teste  
Desafios do Teste  
Casos de Teste  
Projeto de Casos de Teste  
Entradas  
Saídas  
Oráculo  
Ordem de Execução  
Tipos de Teste  
Técnicas de Teste  
Fases de Teste  
Teste de Unidade  
Teste de Integração  
Teste de Sistema  
Teste de Aceitação  
Limitações do Teste  
Automatização  
**Algumas definições**  
Exercícios  
Resumo  
Referências

## Verificação e Validação

- ▶ Validação: o processo de avaliar o software, ao final do seu desenvolvimento com o objetivo de garantir conformidade com o seu uso esperado.
  - ▶ Requer conhecimento do domínio para o qual o software foi desenvolvido
- ▶ Verificação: o processo de determinar se os produtos de uma determinada fase de desenvolvimento do software estão de acordo com os requisitos estabelecidos nas fases anteriores.
  - ▶ Atividade mais técnica, que usa conhecimento sobre artefatos particulares, requisitos e especificações.
- ▶ Teste: Avaliação do software por meio da observação de sua execução.

## Erros, etc

- ▶ Defeito: uma incorreção estática no código.
- ▶ Erro: um estado interno incorreto, que é consequência da execução de um defeito.
- ▶ Falha: comportamento externo incorreto, com respeito aos requisitos ou qualquer outra descrição do comportamento esperado.

## Uma analogia(Ammann e Offut, 2008)

- ▶ Considere um médico fazendo um diagnóstico num paciente doente. O paciente entra no consultório com uma lista de “falha” (ou seja, os sintomas). O médico tem que descobrir o “defeito”, ou seja, a raiz, a causa do problema. Para isso, o médico pode solicitar exames exames que vão procurar condições internas anômalas como pressão alta, batida irregular, colesterol, etc. Essas condições anômalas, na nossa terminologia seriam os “erros”.
- ▶ Defeitos de software tem origem num “engano” humano.

## Exemplo

```
1 public static int numZero (int[] x) {  
2     // Effects: if x == null throw NullPointerException  
3     // else return the number of occurrences of 0 in x  
4     int count = 0;  
5     for (int i = 1; i < x.length; i++)  
6     {  
7         if (x[i] == 0)  
8         {  
9             count++;  
10        }  
11    }  
12    return count;  
13 }
```

Código fonte: ./src/defeito.java





## Para uma falha ocorrer

- ▶ O ponto do programa que contém uma falha deve ser executado (alcançabilidade).
- ▶ Após a execução deste ponto, o estado da execução deve ser incorreto (infecção).
- ▶ O estado infectado deve se propagar de modo a produzir uma saída incorreta (propagação).

## Para uma falha ocorrer

- ▶ O ponto do programa que contém uma falha deve ser executado (alcançabilidade).
- ▶ Após a execução deste ponto, o estado da execução deve ser incorreto (infecção).
- ▶ O estado infectado deve se propagar de modo a produzir uma saída incorreta (propagação).
- ▶ RIP (Reachability, Infection, Propagation).

Teste  
Desafios do Teste  
Casos de Teste  
Projeto de Casos de Teste  
Entradas  
Saídas  
Oráculo  
Ordem de Execução  
Tipos de Teste  
Técnicas de Teste  
Fases de Teste  
Teste de Unidade  
Teste de Integração  
Teste de Sistema  
Teste de Aceitação  
Limitações do Teste  
Automatização  
Algumas definições  
Exercícios  
Resumo  
Referências

## lastZero

```
1 public static int lastZero (int[] x)
2 {
3     //Effects: if x==null throw NullPointerException
4     // else return the index of the LAST 0 in x.
5     // Return -1 if 0 does not occur in x
6
7     for (int i = 0; i < x.length; i++)
8     {
9         if (x[i] == 0)
10        {
11            return i;
12        }
13    }
14    return -1;
15 }
16 // test: x=[0, 1, 0]
17 // Expected = 2
```

Código fonte: src/lastzero.java

## lastZero

- ▶ a) Identifique o defeito
- ▶ b) Se possível, identifique um caso de teste que não executa o defeito.
- ▶ c) Se possível, identifique um caso de teste que executa o defeito mas não resulta em um estado de erro.
- ▶ d) Se possível, identifique um caso de teste que resulta em um estado de erro mas não em uma falha.
- ▶ e) Para o caso de teste dado, identifique o primeiro estado de erro. Descreva o estado por completo.
- ▶ f) Corrija o defeito e verifique que o caso de teste dado produz o resultado correto.

## oddOrPos

```
1 public static int oddOrPos(int[] x)
2 {
3     //Effects: if x==null throw NullPointerException
4     // else return the number of elements in x that
5     // are either odd or positive (or both)
6     int count = 0;
7     for (int i = 0; i < x.length; i++)
8     {
9         if (x[i]%2 == 1 || x[i] > 0)
10        {
11            count++;
12        }
13    }
14    return count;
15 }
16 // test: x=[-3, -2, 0, 1, 4]
17 // Expected = 3
```

Código fonte: src/oddorpos.java

Teste  
Desafios do Teste  
Casos de Teste  
Projeto de Casos de Teste  
Entradas  
Saídas  
Oráculo  
Ordem de Execução  
Tipos de Teste  
Técnicas de Teste  
Fases de Teste  
Teste de Unidade  
Teste de Integração  
Teste de Sistema  
Teste de Aceitação  
Limitações do Teste  
Automatização  
Algumas definições  
Exercícios  
Resumo  
Referências

## Resumo

- ▶ A atividade de teste é um processo executado em paralelo com as demais atividades do ciclo de vida de desenvolvimento do software.
- ▶ A principal tarefa da atividade de teste é o projeto de casos de teste. Eles é que são responsáveis por revelar erros no programa.
  - ▶ O projeto de casos de teste deve levar em consideração o objetivo do teste que é demonstrar que o programa em teste possui erros.
  - ▶ Segredo é selecionar aqueles casos de teste com maior probabilidade de revelar os erros existentes.
- ▶ Diferentes técnicas de teste existem para auxiliar na atividade de teste.
- ▶ Além disso, o teste deve ser conduzido em fases para reduzir a complexidade.

Teste  
Desafios do Teste  
Casos de Teste  
Projeto de Casos de Teste  
Entradas  
Saídas  
Oráculo  
Ordem de Execução  
Tipos de Teste  
Técnicas de Teste  
Fases de Teste  
Teste de Unidade  
Teste de Integração  
Teste de Sistema  
Teste de Aceitação  
Limitações do Teste  
Automatização  
Algumas definições  
Exercícios  
Resumo  
**Referências**



## Referências Bibliográficas

Ammann, P.; Offut, J. *Introduction to software testing*. 1a. ed. Cambridge University Press, 2008.  
Beizer, B. *Software testing techniques*. 2nd ed. New York: Van Nostrand Reinhold Company, 1990.  
Binder, R. V. *Testing object-oriented systems: Models, patterns, and tools*, v. 1. Addison Wesley Longman, Inc., 1999.  
Craig, R. D.; Jaskiel, S. P. *Systematic software testing*. Artech House Publishers, 2002.  
IEEE. *IEEE standard glossary of software engineering terminology*. Standard 610.12-1990, IEEE Computer Society Press, 1990.  
ISO/IEC. *Quality characteristics and guidelines for their use*. Padrão ISO/IEC 9126, ISO/IEC, 1991.  
Pressman, R. S. *Engenharia de software*. 6 ed. McGraw-Hill, 2006.

