

AULA 11 - Teste de Software

Iniciais dos alunos: *AVC, JPP, PC*

Disciplina: *Programação Modular (INF1301)* – Professor: *Flavio Bevilacqua*
:

1) Definição:

Método de controle de qualidade baseado na execução de experimentos controlados com o objetivo de encontrar erros na aplicação.

Devemos testar o máximo possível para que a gama de usos que o usuário fará não encontre erro. Quanto mais testamos, menor a chance do usuário encontrar um erro.

Existem dois tipos de manutenção, corretiva e evolutiva. Manutenção Corretiva ocorre quando algo na aplicação não está de acordo com os requisitos. Manutenção evolutiva ocorre quando o cliente deseja algo que não está nos requisitos. O cliente não paga pela manutenção corretiva, mas paga pela evolutiva.

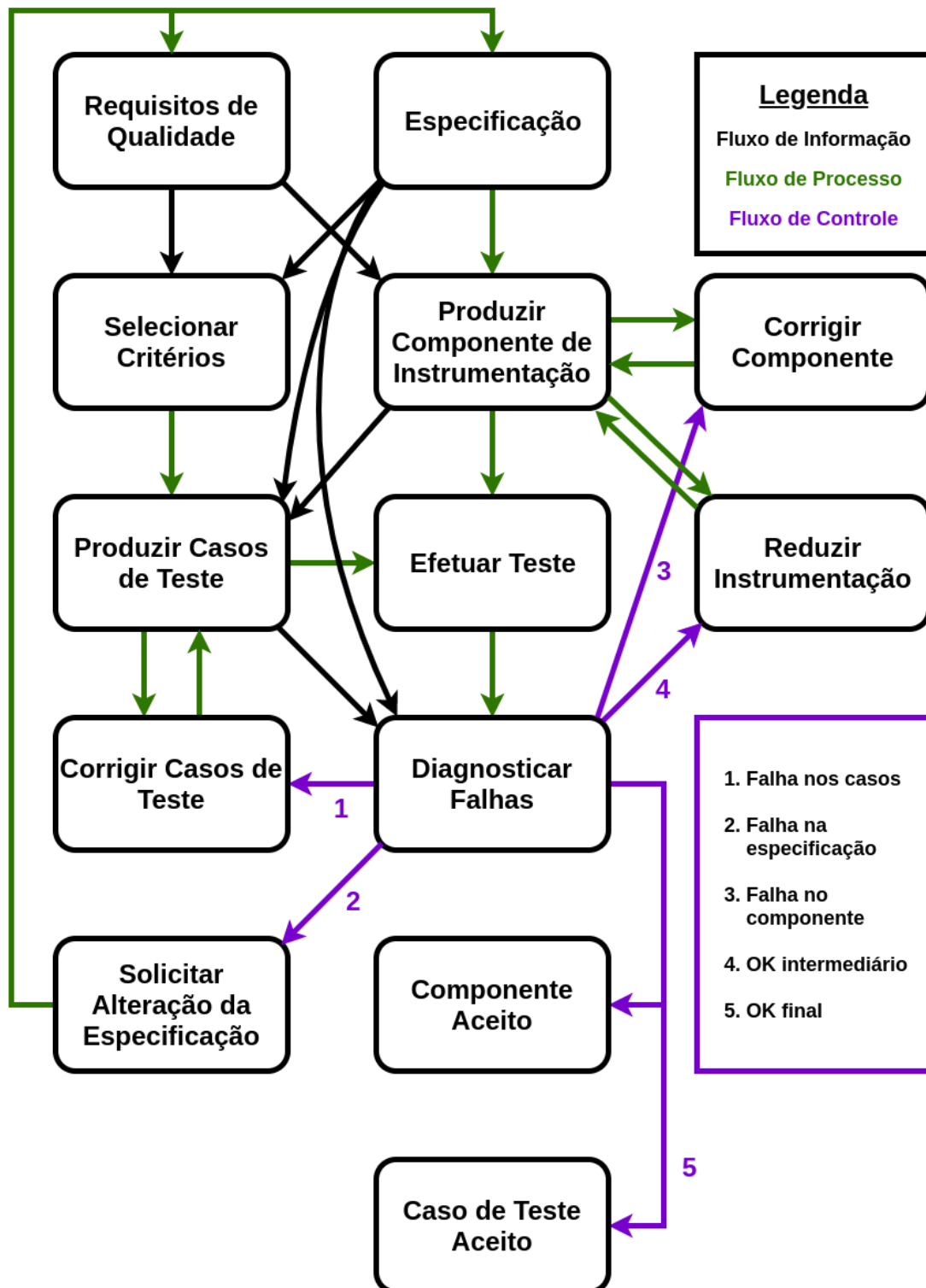
Casos de teste começam a ser elaborados durante a documentação de requisitos.

Antes de realizar os testes:

- Critérios de seleção de casos de teste. Aplicações que possuem algoritmos complexos usam testes de caixa aberta.
- Cenários de teste:
 - Modo de uso do artefato.
 - Organização necessária.
 - Ferramentas.
 - Definição e povoamento de base.

- Pessoas envolvidas no teste.
- Massa de teste.
- Resultados esperados.

2) Processo de Teste de um Programa



Testagem mais rigorosa é utilizada em aplicações críticas (serviços com elevado valor ou risco grande).

Testagem menos rigorosa é utilizada em serviços de baixo valor ou risco pequeno.

Testes são utilizados para detectar a presença de erros, nunca a ausência de erros.

3) Registro de Falhas:

Planilha com a seguinte sugestão de campos:

- (a) Data
- (b) idTeste
- (c) Sintoma
- (d) Data de correção
- (e) Artefatos alterados
- (f) Classe da falha (conjunto ao qual ela pertence, ex: especificação projeto, código, plataforma, rede, etc.)
- (g) Correção realizada

4) Critérios de Seleção de Casos de Teste:

Teste Caixa Fechada: são casos de teste gerados a partir das especificações (input e output ou AE e AS).

Teste Caixa Aberta: São casos de teste gerados a partir da estrutura do código. Contadores de cobertura. Deve percorrer todos os caminhos do código.

Teste Estrutura de Dados: São casos de teste gerados a partir das assertivas estruturais e dos modelos das estruturas de dados utilizadas.

OBS: o critério é válido quando acusa falha quando há um erro no artefato, confiável quando acusa falha independente da escolha de dados ou ações, completo quando cobre todas as condições definidas por um padrão de completeza (todos

os caminhos ou todas as situações de retorno), eficaz quanto mais erros encontra e eficiente quanto menos recursos gastar.

5) Processo de Geração de Casos de Teste:

- Caso de teste abstrato (o quê?)

Ex: A repetição deve ocorrer três vezes.

- Caso de teste semântico (como?)

Ex: Para que a repetição ocorra três vezes é preciso é preciso que o arquivo contenha três registros.

- Caso de teste valorado ou simplesmente caso de teste

Ex: Inclusão dos registros de teste necessários na base e execução da aplicação a ser testada.

6) Recomendações Básicas de Valoração de Casos de Teste:

- $a \geq b$:

$a > b$ ($a = b + 1$), $a = b$, $a < b$ ($a = b - 1$)

- Valores em campos de tamanho variável:

- vazio
- tamMin
- tamMax
- tamMin - 1
- tamMax + 1
- tamMax - 1
- tamMin + 1
- conjunto de valores permitidos

- Casos para estruturas criadas dinamicamente:

Ex: Pesquisar em lista

- Lista vazia
- Lista com um único elemento
- Lista com três elementos

Ex: Pesquisa em árvore

- Árvore vazia
- Árvore com um único nível
- Árvore com três níveis

OBS: sempre testar utilizando os valores-limite, pois é nas extremidades que ocorre o maior número de erros.

- Pesquisa num vetor com três ou mais elementos:

a	b	c	d
---	---	---	---

- Elemento no início (a)
- Elemento no final (d)
- Elemento no meio (b ou c)
- Elemento que não está no vetor (e)

- Teste de repetição

Ex: Coloca A em todos os índices do vetor

A	A	A	–	–
---	---	---	---	---

I C C

O terceiro A é colocado em um índice calculado sobre outro índice calculado.

Arrasto = 3 - 1 = 2.

Ex: Fibonacci no vetor

1	1	2	3	5	_
---	---	---	---	---	---

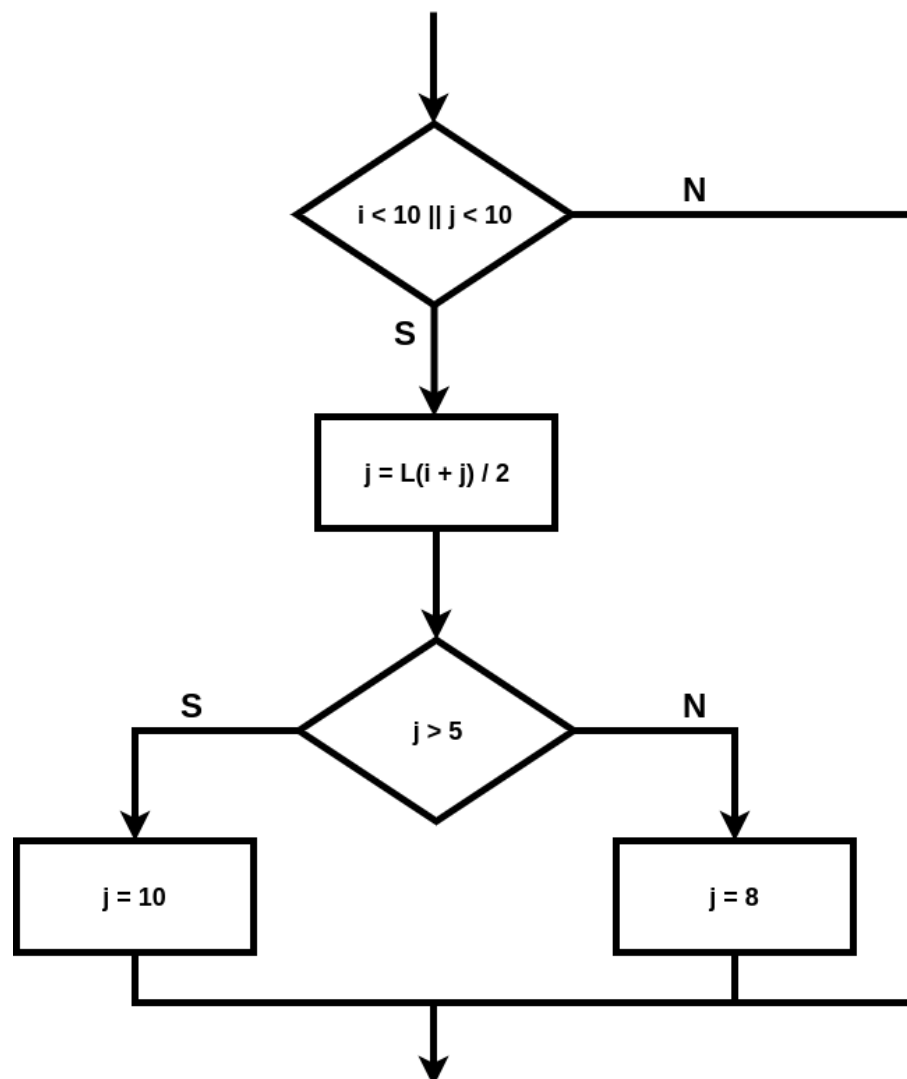
I I C C C

O quinto valor é calculado sobre outros dois valores calculados calculado.

Arrasto = $5 - 1 = 4$.

Arrasto é o menor número de repetições necessárias para que a próxima iteração dependa única e exclusivamente de valores calculados em repetições anteriores.

7) Padrões de Cobertura de Testes Caixa Aberta



Cobertura de Decisões

①

i < 10	j < 10
S	S
S	N
N	S
N	N

②

j > 5
S
N

Caso	1	2	3	4	5	6	7	8
i	4	1	2	1	10	10	10	10
j	4	9	10	10	2	1	10	10
①	SS	SS	SN	SN	NS	NS	NN	NN
②	S	N	S	N	S	N	Não Testa	Não Testa

Cobertura de Arestas

①

i < 10 j < 10
S
N

②

j > 5
S
N

Caso	1	2	3	4
i	4	1	10	10
j	9	10	10	10
①	S	S	N	N
②	S	N	Não Testa	Não Testa

Cobertura de Comandos

①

i < 10 j < 10
S

②

j > 5
S
N

Caso	1	2
i	4	1
j	9	10
①	S	S
②	S	N

8) Teste de Caixa Fechada

Casos de teste gerados a partir da especificação. O teste é completo quando todas as condições de retorno são testadas. O método principal é o de partição em classes de equivalência, que são conjuntos de casos de teste que possuem o mesmo objetivo. O teste é eficiente quando gasta menos recursos. O objetivo é gerar um teste para cada classe de equivalência.

Passo 1: Gerar os grupos principais de teste relacionados com a estrutura utilizada e resultados esperados.

Ex: Pesquisar um valor em uma estrutura sequencial (lista, vetor, etc).

Grupos de teste em estrutura:

- Lista vazia
- Lista com um único elemento
- Lista com três elementos

Resultados esperados:

- Achou
- Não achou

Passo 2: Gerar a tabela de casos de teste.

Caso	Lista	Elemento Pesquisado	Achou	Não Achou
1	Vazia	A	0	1
2	A	A	1	0
3	A	B	0	1
4	ABC	A	1	0
5	ABC	C	1	0
6	ABC	B	1	0
7	ABC	D	0	1

Passo 3: Gerar script

==caso1

=crialista 0

=pesquisa A 1

==caso2

=insere A 0

=pesquisa A 0

==caso3

=pesquisa B 1

==caso4

=insere B 0

=insere C 0

=pesquisa A 0

==caso5

=pesquisa C 0

==caso6

=pesquisa B 0

==caso7

=pesquisa D 1

9) Testes Estatísticos:

São casos de testes gerados automaticamente, utilizando dados aleatórios. Necessita de uma aplicação, função ou módulo gerador.

Vantagem: total automatização.

Desvantagem: geração de casos de teste repetidos.