



Técnicas e Estratégias de Testes

José Osvano da Silva, PMP

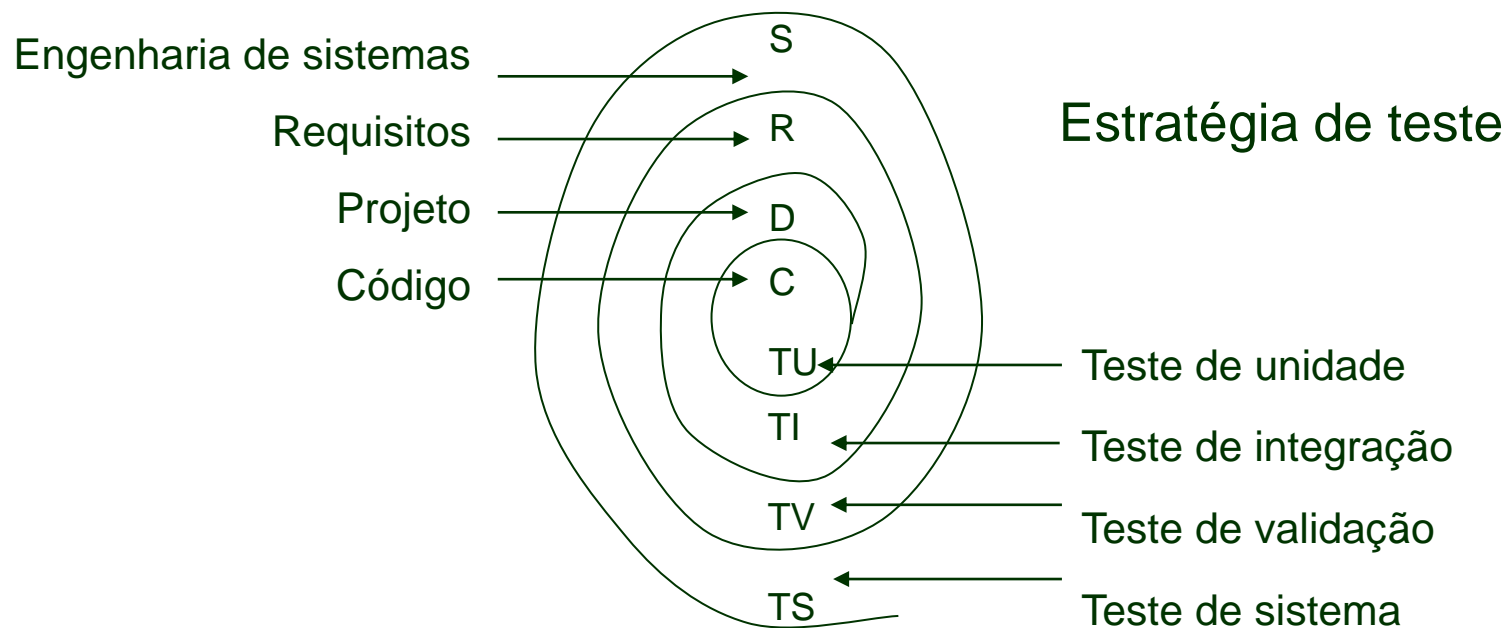
Sumário

- Introdução
- Objetivos do Teste
- Fluxo de Informações de Testes
- Técnicas de Teste de Software
 - Teste de Caixa Preta
 - Teste de Caixa Branca
- Teste do Caminho Básico
 - Complexidade Ciclomática
 - Complexidade Ciclomática Passo a Passo
- Exercícios

Introdução

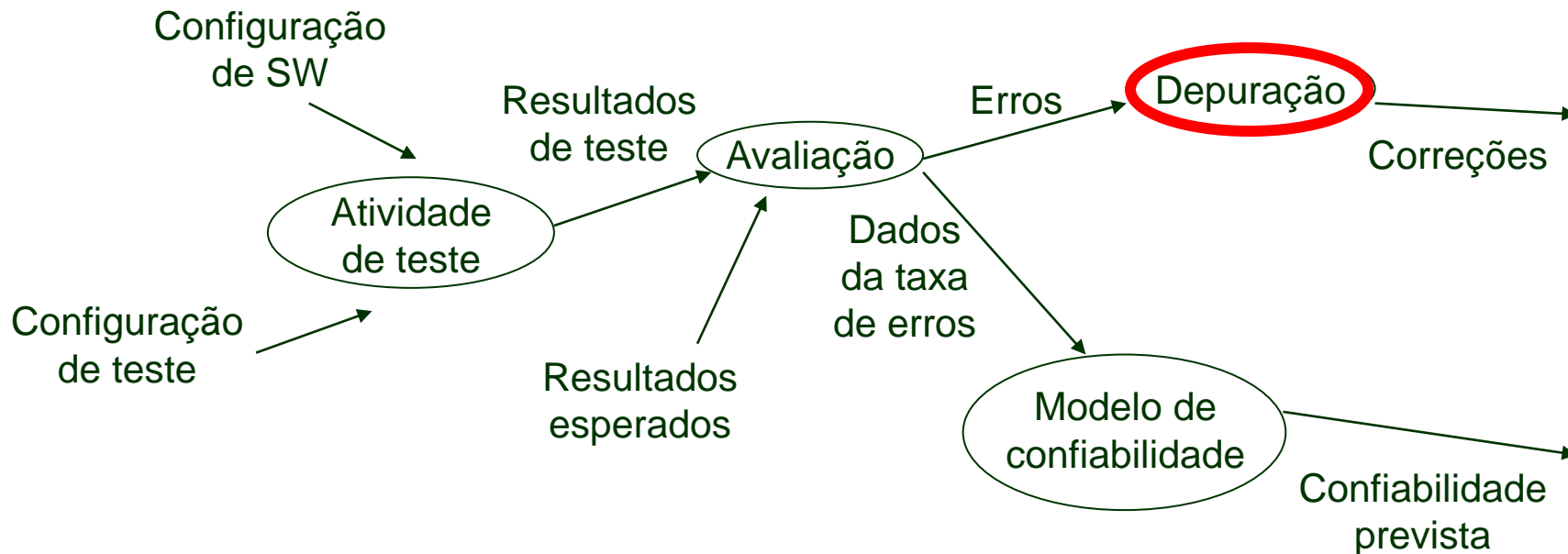
- Teste é um conjunto de atividades que pode ser **planejado** antecipadamente e **realizado** sistematicamente.
- É possível definir um “template” (esqueleto);
- Um conjunto de passos ao qual é possível alocar **técnicas** de projeto de casos de teste e **estratégias** de teste específicos.

Objetivos do Teste



- O Processo de Teste, como qualquer outro processo deve ser revisto continuamente;
- O Objetivo é ampliar sua atuação e possibilitar aos profissionais uma maior visibilidade e organização dos seus trabalhos;
- Isso resulta numa maior agilidade e controle operacional dos projetos de testes.

Fluxo de Informações de Testes



- O processo de depuração é a parte mais imprevisível do processo de teste;
- Um erro que indique uma discrepância de 0,01% entre resultados esperados e reais pode demorar uma hora, um dia ou um mês para ser diagnosticado e corrigido.



Técnicas de Teste de Software

- Conhecendo-se a função específica que um produto projetado deve executar;
- Testes podem ser realizados para demonstrar que **cada função é totalmente operacional** (teste de caixa preta - “black box”);

Técnicas de Teste de Software

- Conhecendo-se o funcionamento interno de um produto, testes podem ser realizados para garantir que **“todas as engrenagens”** funcionam perfeitamente;
- Que a operação interna de um produto tem um desempenho de acordo com as especificações;
- Que os componentes internos foram adequadamente postos à prova (teste de caixa branca - “white box”);

Teste de Caixa Preta

- Teste de caixa preta refere-se aos testes realizados nas **interfaces** do Software;
- A entrada é adequadamente aceita e a saída é corretamente produzida com a integridade das informações externas mantida.

Teste de Caixa Branca

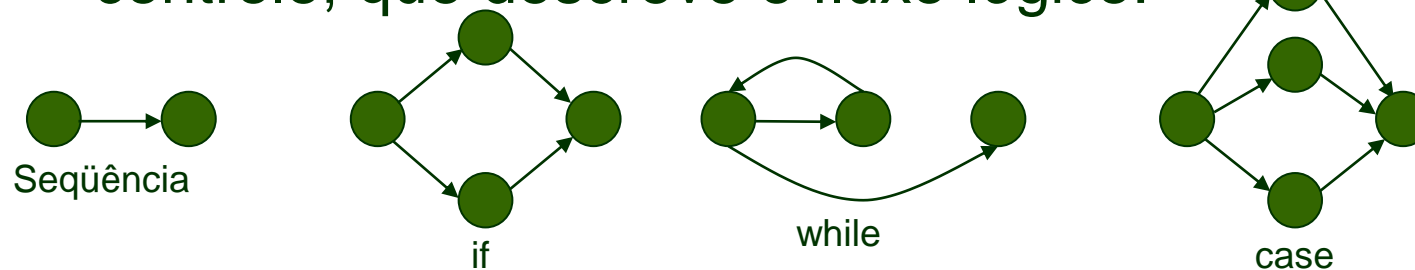
- Teste de caixa branca baseia-se num minucioso exame dos detalhes procedimentais, **através da definição de todos os caminhos lógicos possíveis.**
- Infelizmente estes testes apresentam problemas logísticos, uma vez que o número destes possíveis caminhos lógicos pode ser muito grande, o que levaria a um tempo infinito.
- Entretanto este tipo de teste não pode ser desprezado como pouco prático, podendo-se optar por um número limitado de opções

Teste de caminho básico

- É uma **técnica de teste de caixa branca** que possibilita que o projetista do caso de teste derive uma medida de complexidade lógica de um projeto procedimental e;
- Use essa medida como guia para definir **um conjunto básico de caminhos de execução.**

□ Notação de grafo de fluxo:

- notação simples para representação do fluxo de controle, que descreve o fluxo lógico:



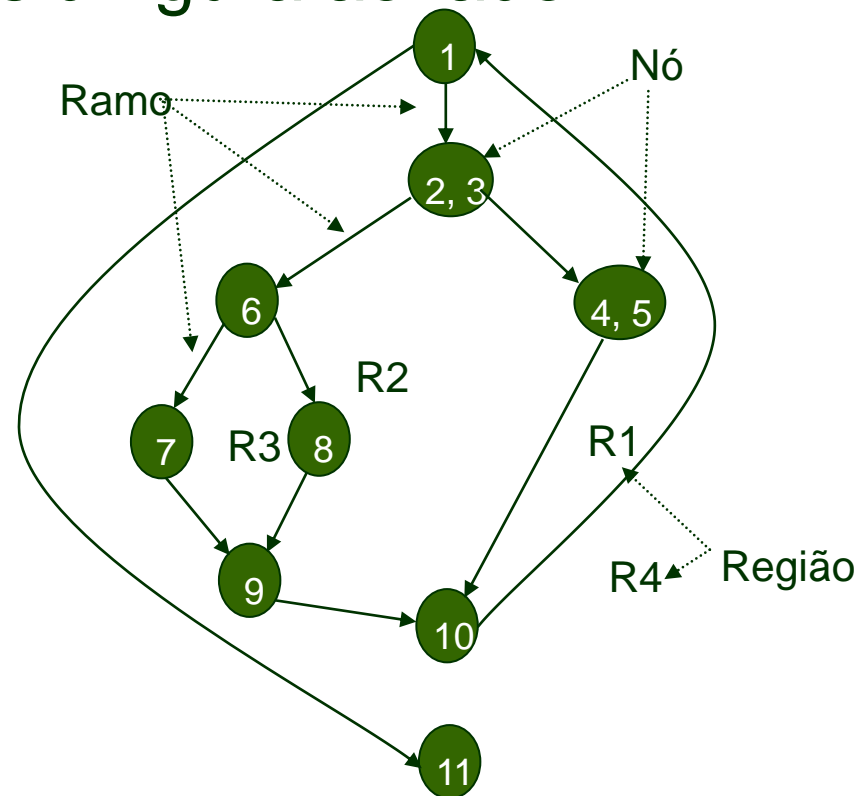
Complexidade Ciclomática

- É uma métrica de Software que proporciona uma medida quantitativa da complexidade lógica de um programa;
- O valor computado da complexidade ciclomática **define o número de caminhos independentes do conjunto básico de um programa;**
- Oferece-nos um limite máximo para o **número de testes que deve ser realizado** para garantir que todas as instruções sejam executadas pelo menos uma vez.

Complexidade Ciclomática

- Por exemplo, um conjunto de caminhos independentes, referentes à figura ao lado:

- caminho 1: 1-11
- caminho 2: 1-2-3-4-5-10-1-11
- caminho 3: 1-2-3-6-8-9-10-1-11
- caminho 4: 1-2-3-6-7-9-10-1-11



Complexidade Ciclomática

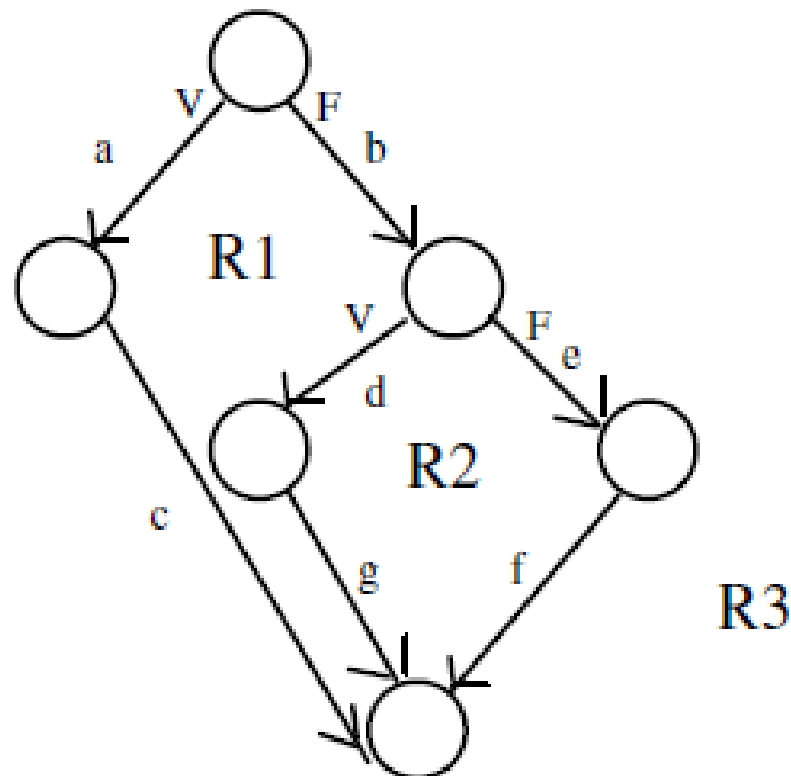
- Para determinarmos os caminhos independentes de um grafo;
- Temos que inicialmente determinar quantos são;
- Existem 3 formas de se determinar este número, também denominado **complexidade ciclomática**:
 - Contar o número de regiões no grafo de controle. No exemplo, temos quatro regiões, R1, R2, R3 e R4: as três internas(R1, R2 e R3) e a externa (R4). Logo, este grafo tem 4 caminhos independentes.
 - As demais são mostradas no material do professor Pasteur no portal.

Complexidade Ciclomática Passo a Passo

```
procedure Teste(A,B,C : integer);  
begin  
    if A=B then  
        A := A + 1  
    else  
        if B = C then  
            B := B+1  
        else  
            B := B -1;  
        end;  
    end;  
end;
```

Complexidade Ciclomática Passo a Passo

- Teria o seguinte grafo de controle:



Complexidade Ciclomática Passo a Passo

- No grafo de controle identificamos os chamados caminhos independentes.
- Dentre um conjunto de caminhos identificados para um grafo, dizemos que eles são independentes quando nenhum deles é formado da combinação de dois ou mais outros.

Complexidade Ciclomática Passo a Passo

- Para determinarmos os caminhos independentes de um grafo, temos que inicialmente determinar quantos são.
- Existem 3 formas de se determinar este número, também denominado **complexidade ciclomática**:

Complexidade Ciclomática Passo a Passo

- 1ª) Contar o número de regiões no grafo de controle. No exemplo anterior, temos três regiões, R1, R2 e R3: as duas internas(R1 e R2) e a externa (R3). Logo, este grafo tem 3 caminhos independentes.

Complexidade Ciclomática Passo a Passo

- 2ª) Aplicar a fórmula:

$$C = E - N + 2$$

Onde:

- C = número de caminhos independentes
- E = número de arestas
- N = número de nós
- No exemplo temos $E = 7$, $N = 6$, logo $C = 7 - 6 + 2 = 3$

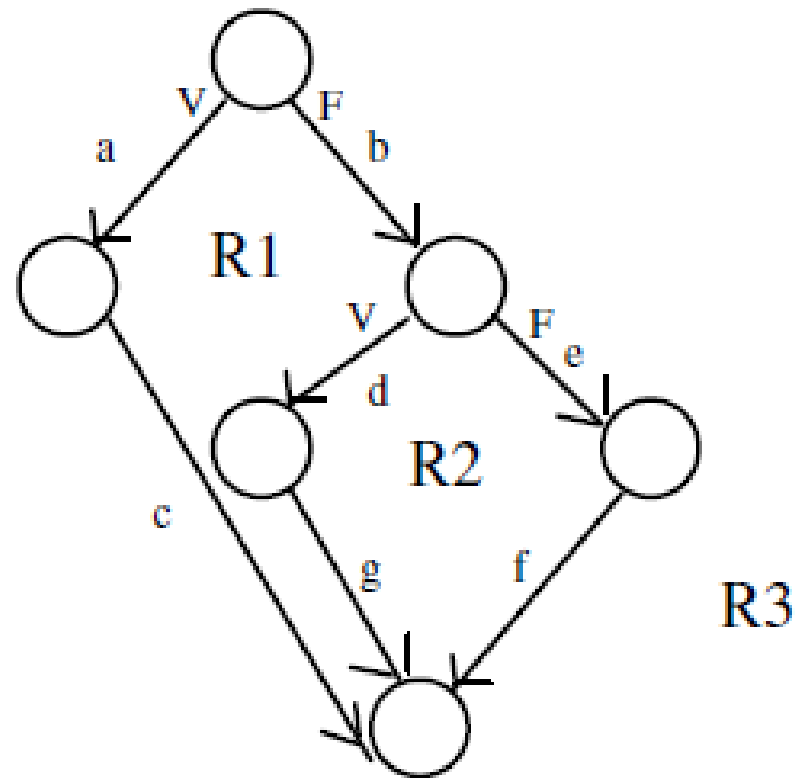
Complexidade Ciclomática Passo a Passo

- 3ª) Contar o número de estruturas de decisão no programa e somar 1. No exemplo, temos 2 "if". Logo, o número de caminhos independentes é 3.

Complexidade Ciclomática Passo a Passo

- Para identificar os caminhos independentes em um grafo fazemos o seguinte:
 - Determinar o número de caminhos independentes conforme visto anteriormente.
 - Tomar o caminho mais a esquerda possível no grafo como primeiro caminho independente.
 - No exemplo, este caminho é o **ac**.

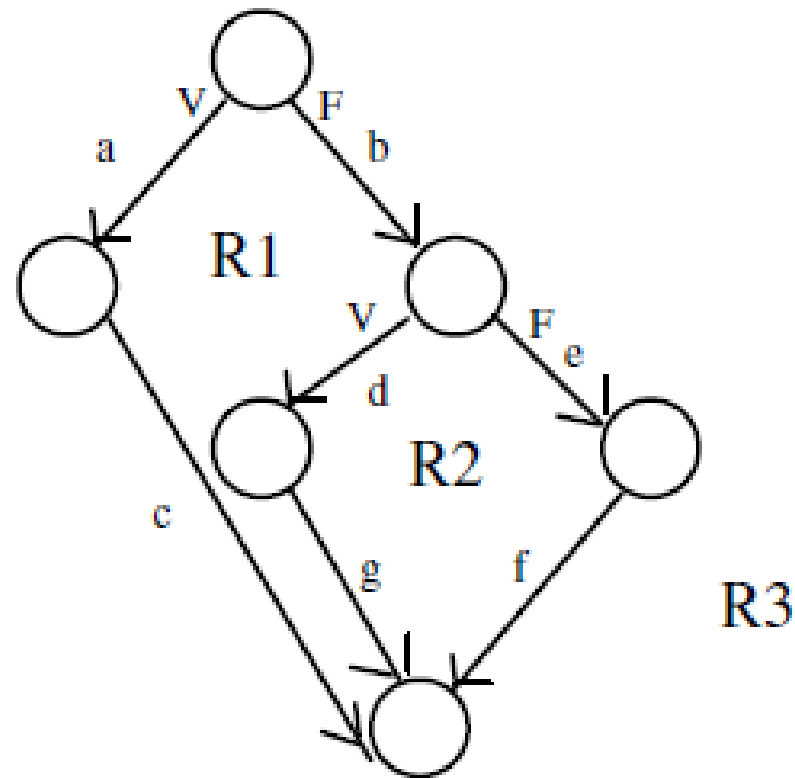
Complexidade Ciclomática Passo a Passo



Complexidade Ciclomática Passo a Passo

- Para identificar os caminhos independentes em um grafo fazemos o seguinte:
 - Tomar o próximo caminho à direita, tendo o cuidado de incluir nele pelo menos uma aresta que não tenha sido incluída nos outros caminhos já determinados. No exemplo, caminho **bdg**.
 - Repetir o passo anterior até que se obtenham todos os caminhos. No exemplo, só nos resta o caminho **bef**.

Complexidade Ciclomática Passo a Passo



1) Avalie o código a seguir:

```
public String getDiaSemana(Integer diaDaSemana) {  
    String retorno = "";  
    switch (diaDaSemana) {  
        case 1:  
            retorno = "Domingo";  
            break;  
        case 2:  
            retorno = "Segunda-feira";  
            break;  
        case 3:  
            retorno = "Terça-feira";  
            break;  
        case 4:  
            retorno = "Quarta-feira";  
            break;  
        case 5:  
            retorno = "Quinta-feira";  
            break;  
        case 6:  
            retorno = "Sexta-feira";  
            break;  
        case 7:  
            retorno = "Sábado";  
            break;  
        default:  
            retorno = "Este não é um dia válido!";  
    }  
    return retorno;  
}
```

Exercício de Fixação

Referências

- SOMMERVILLE, Ian, **Engenharia de Software**, 8ª Edição, São Paulo, Editora Pearson Prentice Hall, 2007.
- PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. 7ª Edição. Porto Alegre: AMGH, 2011. 780 p.
- MIRANDA JR, Pasteur Ottoni de, **Testes de Caixa Branca e Caixa Preta**. Belo Horizonte: PUC-Minas, 2013.

Dúvidas



José Osvano da Silva
joseosvano@unipac.br