Núcleo de Excelência em Testes de Sistemas

Fundamentos de Teste de Software

Módulo 2- Teste Estático e Teste Dinâmico

Aula 6

Teste Dinâmico: Técnicas de Especificação

SOFTEX RECIFE

SUMÁRIO

INTRODUÇÃO

O entendimento das técnicas para elaboração de um projeto de teste é primordial para a consolidação do planejamento e preparação da execução dos testes dentro de um determinado projeto.

As técnicas de projeto de teste reúnem as melhores práticas e consolidam padrões para apoiar o processo de elaboração do caso de teste, onde o analista de teste pode, a partir de um conjunto de técnicas existentes, selecionar um subconjunto que melhor se aplica ao contexto do projeto em questão.

As técnicas **baseadas na especificação**, ou **caixa preta**, abrangem um conjunto de práticas para exercitar um sistema tomando como entrada as especificações deste. Não apenas os requisitos funcionais, mas também os não-funcionais são o ponto de partida para projetar os testes que podem ser através das técnicas: **partição em equivalência**; **análise de valor fronteira**; **tabelas de decisão** e **diagramas de transição de estados.**

Com o objetivo de complementar o conhecimento básico para apoiar o desenvolvimento dos projetos de teste, este capítulo irá abordar as **técnicas baseadas na estrutura ou caixa branca** e **baseadas na experiência.**

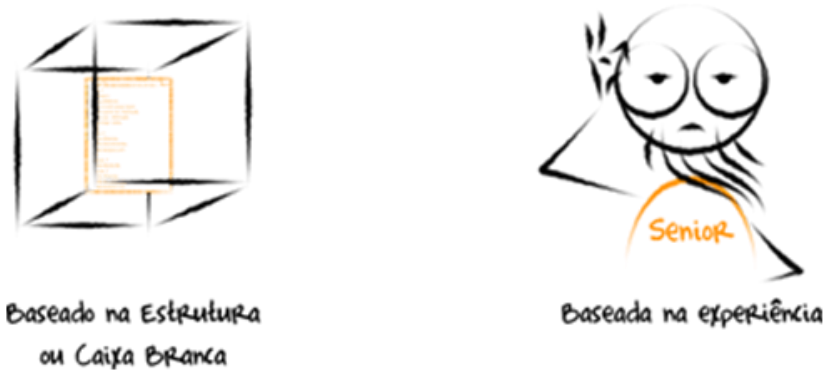

Figura 1: Tipos de Técnicas para Projetar Casos de Teste

TÉCNICAS BASEADAS NA ESTRUTURA

As técnicas de teste baseadas na estrutura, ou caixa branca, são utilizadas para explorar estruturas ou componentes do sistema nos mais diversos níveis. A estrutura de um sistema pode ser compreendida como tudo aquilo utilizado para a construção de um determinado software como, por exemplo, a estrutura do código, aspectos arquiteturais, estrutura de um menu ou até de uma webpage.

Para explorar, no nível de componente, as estruturas de interesse serão as estruturas do programa, como as decisões e loops. No nível de integração, pode-se exercitar como os componentes interagem uns com os outros e, no nível de sistema, como os usuários irão interagir com um menu, por exemplo.

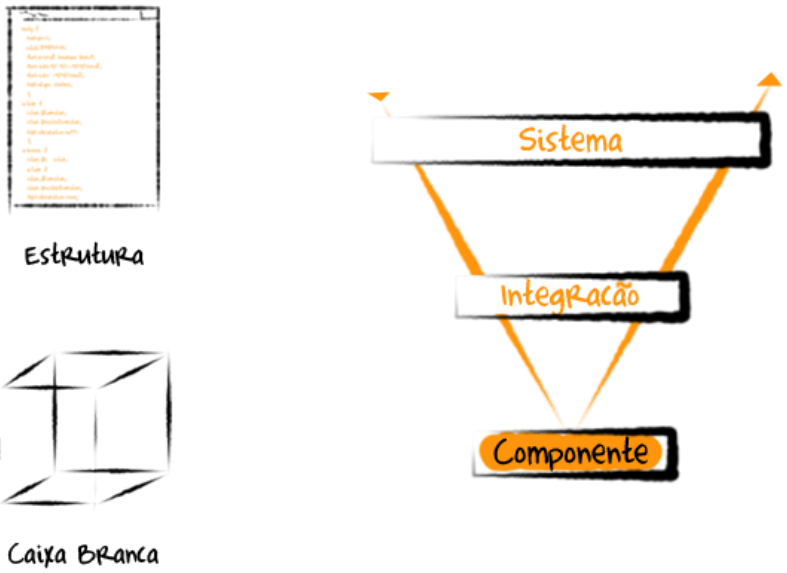

Figura 2: Níveis de Teste Caixa Branca

Ao invés de verificar se um componente ou sistema funciona corretamente, conforme especificado, as técnicas de caixa branca irão focar na garantia de que um determinado elemento dessas estruturas está executando conforme projetado.

Existem diversas técnicas para realização do teste caixa branca. Dentre elas, podemos citar:

- Fluxograma;
- Grafos de Controle de Fluxo; e
- Teste e Cobertura de Declaração.

1. Fluxograma

Um fluxograma é uma estrutura visual simples e representa a forma que um código pode ser lido. Sua representação se dá através de dois símbolos:

- Retângulos para representarem declarações sequenciais; e
- Losango para representar as decisões.

Mais de uma declaração sequencial pode ser escrita dentro de um único retângulo, contanto que não exista nenhuma decisão na sequência. Qualquer decisão é representada pelo losango, incluindo seus loops associados.

Por exemplo, podemos analisar o retângulo com um pseudocódigo que representa um programa sequencial. A segunda figura representa uma estrutura de decisão ou seleção, e a terceira figura representa um fluxograma para uma estrutura de loop.

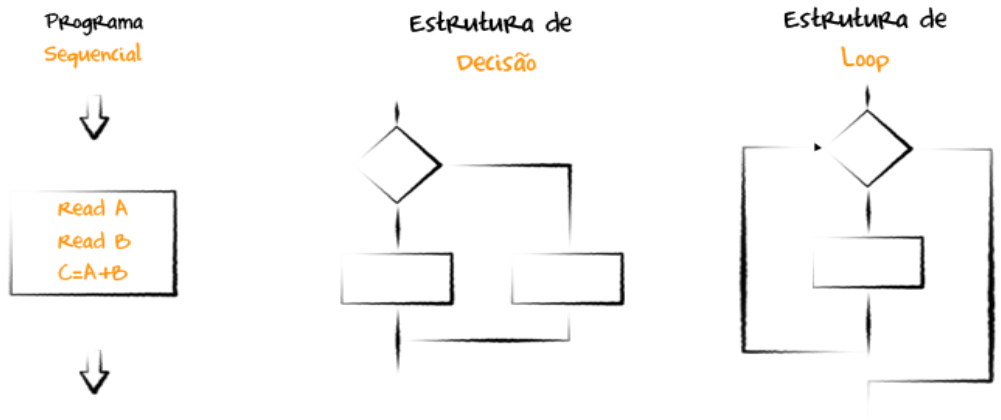

Figura 3: Estruturas de um Fluxograma

Vamos analisar o exemplo com base no código mostrado na Figura 4, que contém estruturas não executáveis, presentes antes do *Begin* e nas linhas brancas do código.

```
Program Maxandmean

A, B, C, Maximun : Integer
Mean: Real

Begin

Read A
Read B
Read C
Mean = (A+B+C) / 3

If A > B
Then
    If A > C
    Then
        Maximum = A
    Else
        Maximum = C
    Endif
Else
    If B > C
    Then
        Maximum = B
    Else
        Maximum = C
    Endif
Endif

Print ("Mean of A, B and C is ", Mean)
Print ("Maximum of A, B, C is ", Maximum)

End
```

Figura 4: Código do Programa MaxadMean

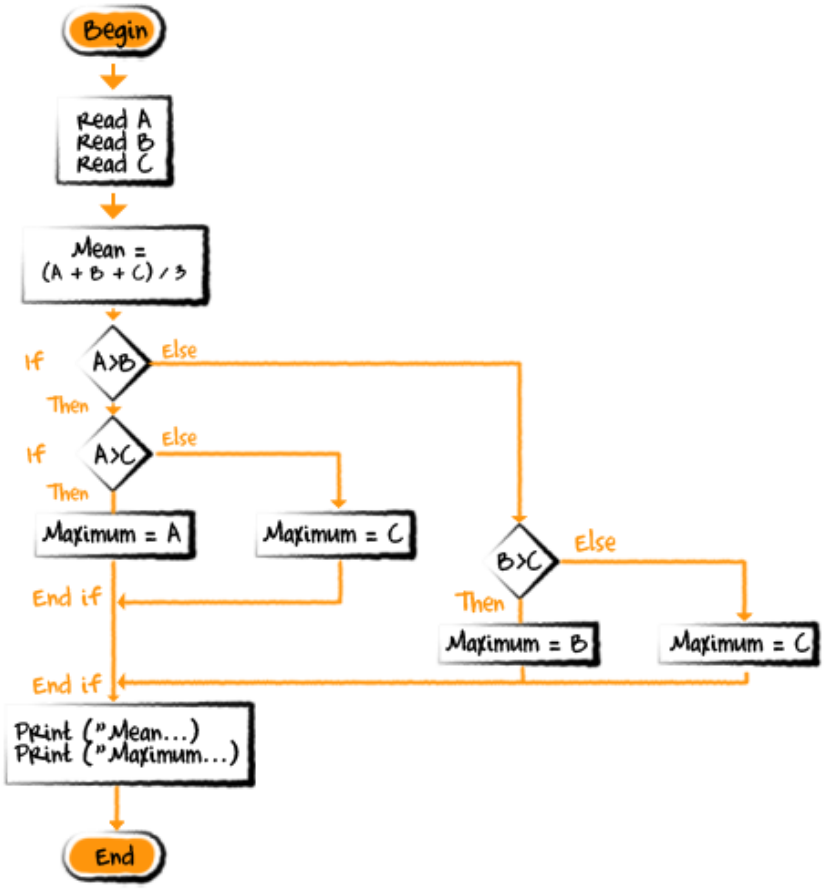

Figura 5: Fluxograma do Programa MaxadMean

O fluxograma na Figura 5 representa o código da Figura 4.

2. Grafos de Controle de Fluxo

Os grafos de controle de fluxo fornecem um mecanismo para representar os pontos de decisão e o fluxo de controle dentro de um código. São similares a um fluxograma, exceto pelo fato de que eles apresentam apenas as decisões. A elaboração do grafo se dá através da análise, exclusivamente das declarações que afetam o fluxo de controle.

O grafo é feito com dois símbolos:

- Nós, que representam qualquer ponto onde o fluxo de controle pode ser modificado, ou pontos, onde a estrutura de controle retorna para o fluxo principal; e
- Arestas, que são as linha que se conectam a qualquer 2 nodos.

A área fechada que contém um conjunto de nós e arestas é chamada de região, conforme mostrado na Figura 6.

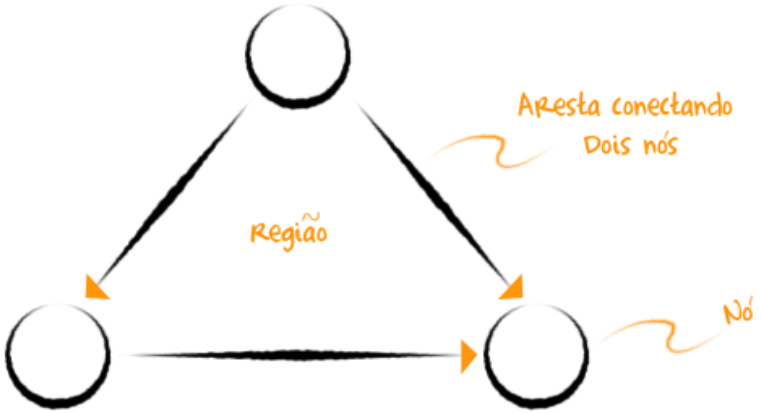

Figura 6: Forma de Representar um Grafo de Controle de Fluxo

É possível desenhar sub-grafos para representar estruturas individuais como, por exemplo:

- IF then else
- Do While
- Repeat Until

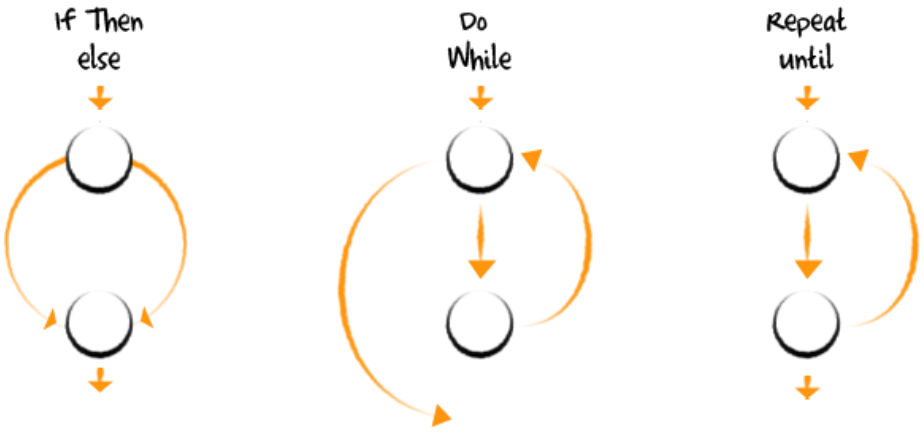

Figura 7: Sub-grafos de Controle de Fluxo

Para desenhar um grafo de controle de fluxo, os passos a seguir devem ser seguidos.

- Primeiramente, devemos fazer a análise do componente para identificar todas as estruturas de controle, conforme a Figura 8 apresentada;
- Depois, adicionar um nó para cada declaração de decisão e expandir os nodos pela substituição do sub-grafo apropriado, representando a estrutura em um determinado ponto de decisão, de acordo com a Figura 9.

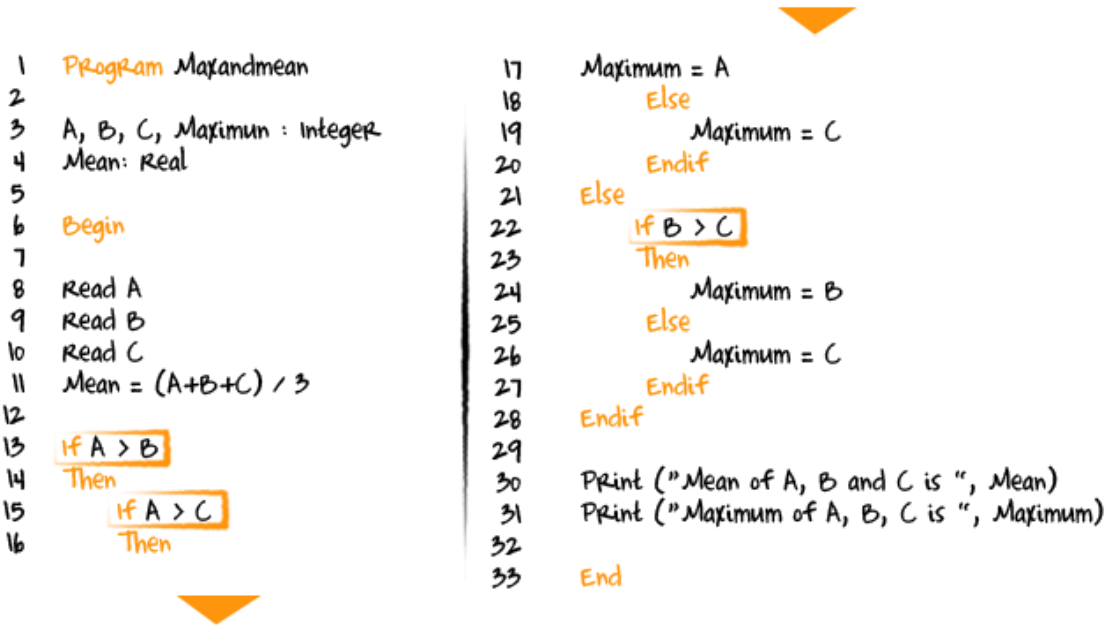

Figura 8: Código do Programa MaxadMean com Estruturas de Controle Identificadas

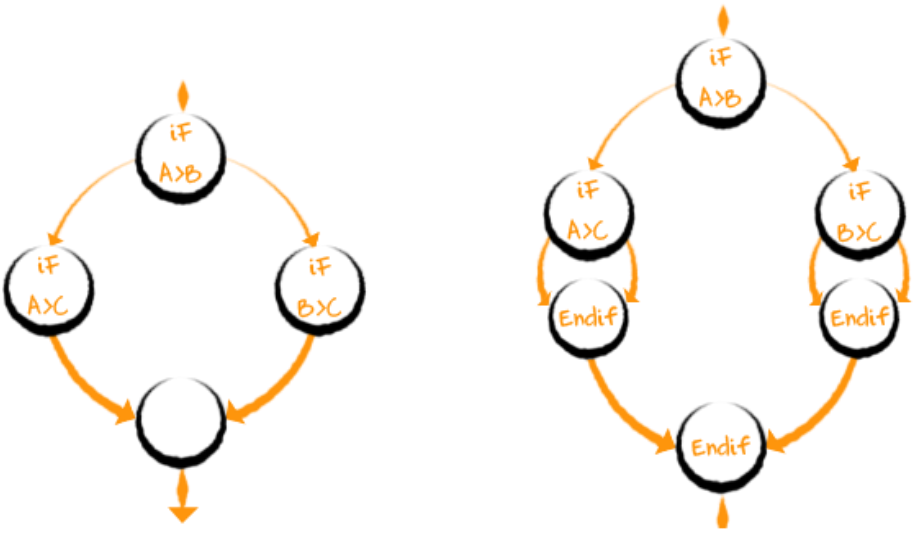

Figura 9: Grafo de Controle de Fluxo com seu Sub-grafo expandido

3. Teste de Cobertura de Declaração

O teste de cobertura de declaração tem o foco de exercitar todas as declarações de programação, assim como analisar apenas a parte executável do código. Se o objetivo é testar todas as declarações executáveis, tem-se 100% de cobertura das declarações e, nesse caso, não se conta as declarações não executáveis.

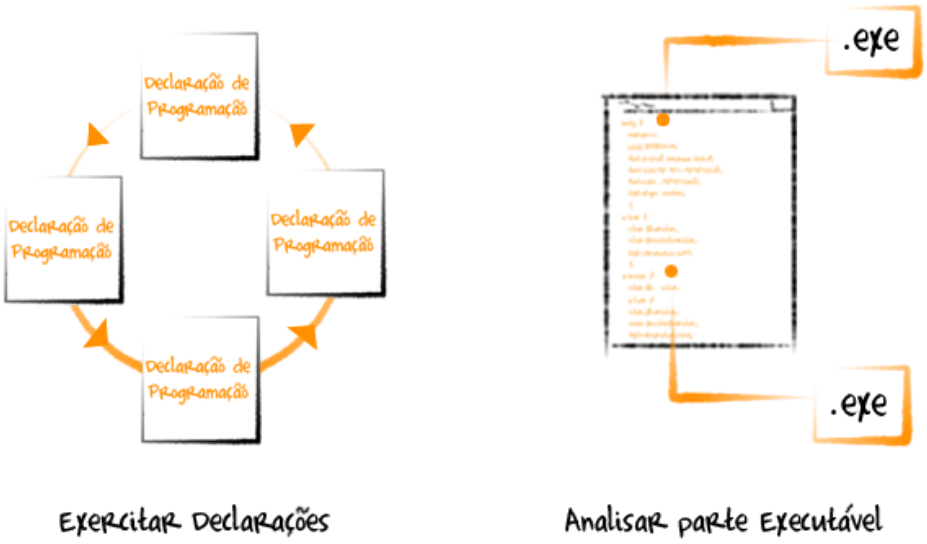

Figura 10: Objetivos do Teste de Cobertura de Declaração

A cobertura do código deve ser mensurada em um projeto de desenvolvimento, haja vista ser uma medida simples, pois um projeto de teste que não tenha exercitado todas as suas estruturas não deve ser considerado completo.

Vamos analisar o seguinte exemplo, com o pseudo-código e o fluxograma representados na Figura 11.

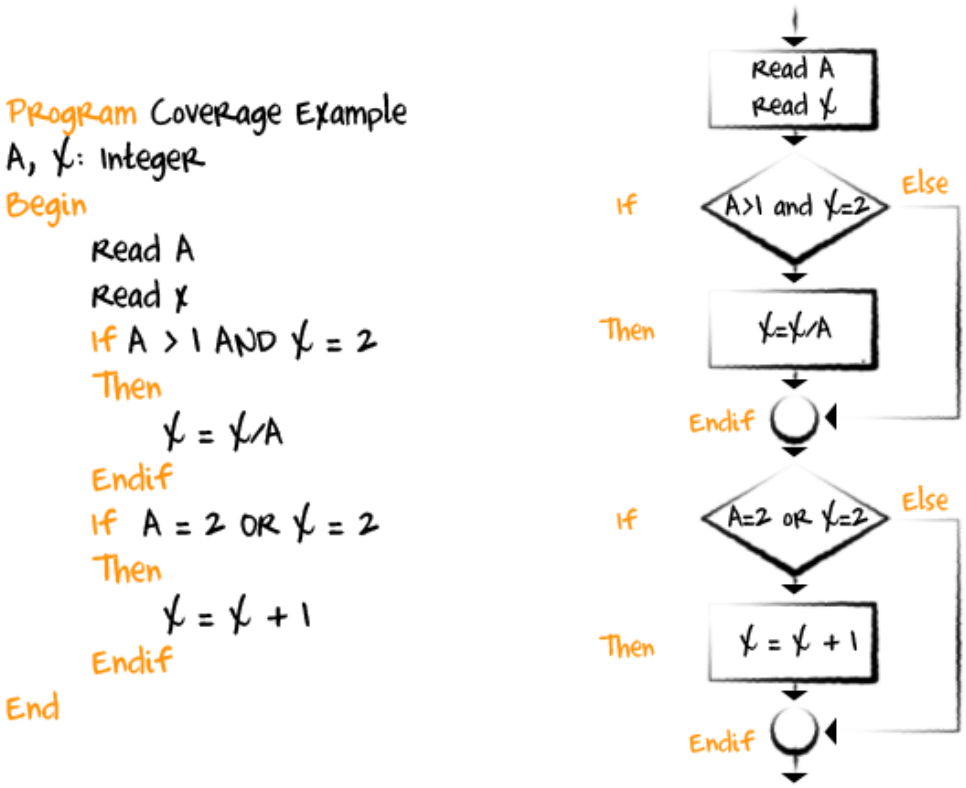

Figura 11: Pseudo-código e Fluxograma

Se você analisar o fluxograma, pode observar que as estruturas executáveis estão representadas pelos retângulos e losangos.

Um grafo de fluxo é menos complexo e mostra apenas os detalhes estruturais, em particular onde o programa se ramifica e se reúne. A partir daí, é possível fazer uma versão híbrida do grafo de fluxo para determinar a cobertura de declaração.

Para se fazer isso, primeiramente, se constrói um gráfico de controle de fluxo convencional e adiciona-se um nó para cada ramo em que há uma ou mais declarações. Com isso, temos o grafo da Figura 12.

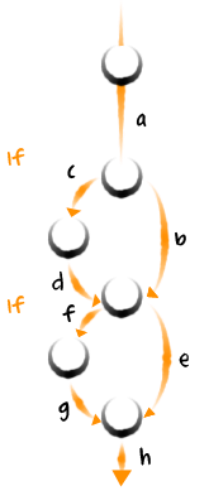

Figura 12: Grafo de Controle de Fluxo Híbrido

Com base nessa estrutura, 100% de cobertura do código pode ser alcançada através da escrita de um único caso de teste que segue o caminho **acdfgh**. Se você definir A=2 e X=2, todas as declarações serão executadas pelo menos uma vez.

TÉCNICAS BASEADAS NA EXPERIÊNCIA

As técnicas baseadas em experiência são aquelas em que se recorre quando não há especificação adequada para utilizar com base e criar os casos de teste. Também conhecidas como exploratórias, as técnicas não sistemáticas podem ser vistas como uma forma suplementar às sistemáticas.

Elas podem ser utilizadas antes ou depois da aplicação dos métodos sistemáticos, no entanto, nunca devem ser encaradas como a única forma de realização dos testes em um projeto.

Elas utilizam a experiência do usuário e do testador para determinar as áreas mais importantes do sistema onde possíveis erros acontecerão. Mesmo quando há especificação disponível, vale à pena complementar os testes estruturados com as técnicas baseadas na experiência.

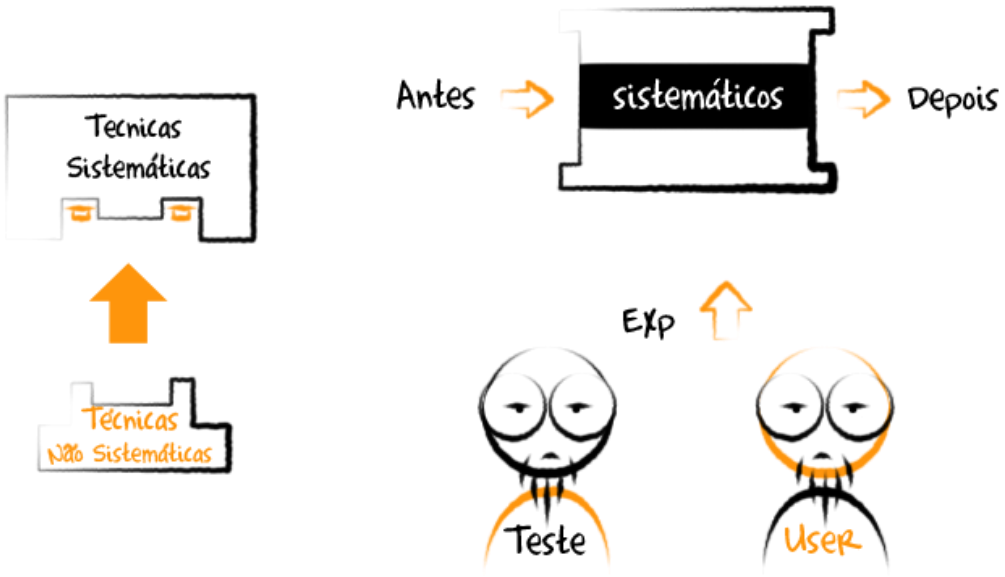

Figura 13: Características das Técnicas Baseadas na Experiência

As técnicas podem variar desde uma abordagem simplista, como o teste *ad hoc* e descoberta de erro, até as mais sofisticadas como o teste exploratório. Todas fazem uso do conhecimento e experiência de quem está realizando os testes ao invés de sistematicamente explorar o sistema.

1. Descoberta de Erro

Descoberta de Erro é a técnica onde a experiência do testador é utilizada para antecipar falhas que possam vir ocorrer em um sistema e projetar testes para atingir essas falhas. Se aplicada após a utilização de técnicas sistemáticas e pode destacar áreas suspeitas e aspectos da aplicação que não foram vistos anteriormente.

A principal desvantagem da técnica é que sua efetividade varia de acordo com a experiência de quem irá executá-la. No entanto, se um grupo de testadores e usuários contribuírem para a construção de uma lista dos possíveis erros e os testes para atacá-los, a fraqueza da técnica pode ser facilmente superada.

Figura 14: Características da Descoberta de Erro

2. Exploratória

O teste exploratório é a técnica que combina a experiência do testador com uma abordagem estruturada para testar onde as especificações estão inadequadas ou indisponíveis e onde existe a pressão do tempo. Pode ser entendido como uma forma estruturada de buscar erros e usar seus resultados para decidir o caminho a ser seguido, se devemos ou não continuar o teste. O teste exploratório, simultaneamente, aprende, projeta e executa os testes. Dessa forma, o teste exploratório maximiza a quantidade de teste que pode ser alcançada dentro de um intervalo pré-definido de tempo.

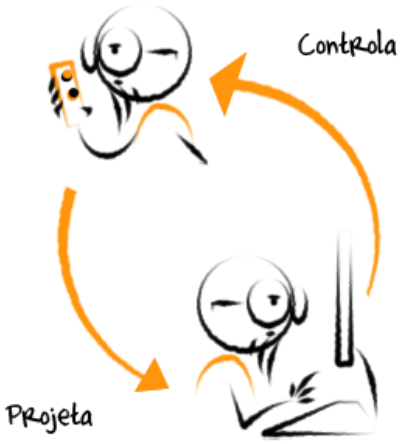

Figura 14: Característica do Teste Exploratório

CONCLUSÃO

Portanto, nesse curso finalizamos as principais técnicas para apoiar a escrita dos casos de teste e a apostila irá complementar o que apresentamos na aula.

A tabela abaixo consolida as técnicas apresentadas nesse modulo. No entanto, a literatura e prática de mercado é bastante abrangente e existem outras técnicas que não foram abordadas nesse curso e podem ser utilizadas para otimizar o processo de elaboração dos casos de teste.

Tabela 1: Técnicas para Especificação de Casos de Teste

Tipo	Técnica
Baseado na Especificação ou Caixa Preta	Partição em equivalência
	Análise de Valor de Fronteira
	Tabelas de Decisão
	Diagramas de Transição de Estados
Baseado na Estrutura ou Caixa Branca	Fluxograma
	Grafos de Controle de Fluxo
	Teste de Cobertura de Declaração
Baseado na Experiência	Descoberta de Erro
	Teste Exploratório