**Resumo sobre Técnicas de Teste de Software**

Na aula são exploradas três importantes técnicas de teste de software: **Teste do Caminho Básico**, **Testes de Estrutura de Controle** e **Testes Baseados em Modelos**. Essas técnicas são fundamentais para garantir a qualidade, a confiabilidade e a eficácia dos sistemas desenvolvidos.

**1. Teste do Caminho Básico**

O Teste do Caminho Básico é uma técnica de teste de caixa-branca. Ele utiliza grafos de fluxo para representar a lógica de controle de um programa e calcular a **complexidade ciclomática**, que define o número mínimo de testes necessários para cobrir todos os caminhos lógicos independentes de um programa. A ideia é garantir que todas as instruções do programa sejam executadas ao menos uma vez, testando todas as condições lógicas em seus estados verdadeiro e falso.

* **Complexidade Ciclomática**: É uma métrica usada para calcular a quantidade de caminhos independentes no grafo de fluxo. Pode ser calculada de três formas diferentes:
  + Contando o número de regiões no grafo.
  + Usando a fórmula: V(G)=E−N+2V(G) = E - N + 2V(G)=E−N+2, onde E é o número de arestas e N é o número de nós.
  + Contando os nós de decisão e aplicando a fórmula: V(G)=P+1V(G) = P + 1V(G)=P+1.

**2. Testes de Estrutura de Controle**

Esta técnica testa a lógica de controle do software, incluindo condições, loops e ramificações. Existem várias abordagens dentro dos Testes de Estrutura de Controle:

* **Teste de Condição**: Verifica as condições lógicas em um programa.
* **Teste de Ciclo**: Testa loops em diferentes níveis de complexidade, como ciclos simples ou aninhados.
  + Para ciclos simples, a técnica inclui testar o loop com diferentes quantidades de iterações (zero, uma, duas, etc.).
  + Para ciclos aninhados, o teste começa com o ciclo mais interno e avança gradativamente, mantendo os outros ciclos em valores mínimos.

**3. Testes Baseados em Modelos**

Esses testes utilizam **modelos formais** que descrevem o comportamento esperado do sistema. Modelos, como **Máquinas de Estados Finitos (MEFs)**, são usados para representar configurações e transições entre estados. Isso permite simular diferentes cenários de uso do sistema e identificar falhas.

* **Métodos Específicos**: Existem várias técnicas baseadas em modelos, como:
  + **TT (Teste de Transição)**: Foca na cobertura de transições entre estados.
  + **UIO (Unique Input/Output)**: Testa sequências de entrada e saída específicas.
  + **W**: Maximiza a cobertura de estados do sistema.
  + **DS (Domínio de Sequências)**: Gera casos de teste para diferentes domínios de entrada.

Essas técnicas são essenciais para garantir que o software seja testado de maneira abrangente e que sua lógica funcione corretamente, aumentando a qualidade do produto final.