

Testes Caixa-Branca

Esse tipo de teste preocupa-se com o quando os casos de teste cobrem a lógica do programa. O caso de teste ideal seria testar todos os caminhos do programa, o que não é possível, principalmente para programas com loops.

Um bom critério de cobertura lógica é conhecido por cobertura de decisões, que determina que cada decisão tenha uma saída verdadeira e uma falsa ao mesmo uma vez, ou seja, que todos os caminhos sejam percorridos.

Existem três exceções para cobertura usando esse critério:

- Programas sem decisões
- Programas com múltiplos pontos de entrada
- Trechos em ON-units.

Um critério melhor é a cobertura de condições, mas também não garante uma cobertura de todos os caminhos. O que é resolvido pelo critério decisão/condição, que exige que cada condição em uma decisão tenha todas as saídas possíveis testadas, e que cada ponto de entrada seja executado ao menos uma vez. Esse último critério também possui falhas, já que condições envolvendo or e and podem mascarar o resultado de outras condições.

Um critério que visa solucionar esse último problema é a cobertura de múltiplas condições, que requer que sejam escritos casos de testes de forma que todas as combinações de saída de condições em cada decisão e em cada ponto de entrada seja executada ao menos uma vez.

Os testes começam dos módulos que não são invocados por outros, usando 'stubs' no lugar de suas dependências. Começam com os módulos terminais de uma aplicação.