

1. Teste de caixa-preta são testes que se preocupam com a implementação do software, mas sim, se a funcionalidade está de acordo com o esperado, dentro do teste de caixa-preta existem várias subdivisões que se aplicam em situações diferentes, o syllabus cita as seguintes:
 - a. **Particionamento de Equivalência:** particiona os dados e os processa igualmente, o que resulta numa saída válida ou inválida. Uma partição é chamada de válida se tiver o valores válidos e inválida, caso contrário. Se necessário, essas partições ainda podem ser divididas em subpartições. O syllabus diz que partições inválidas devem ser testadas individualmente para evitar que falhas sejam mascaradas
 - b. **Análise de Valor Limite:** Essa é uma técnica semelhante à anterior, mas ao invés de considerar valores binários, como válidos e inválido, ela considera intervalos de valores para dizer se algo é válido para uma determinada situação
 - c. **Teste de tabela de decisão:** São usados para teste implementações de requisitos, identificar condições para de validade para determinados valores. Cada condição pode ser verdadeira, falsa, valor que não importa, ação que não deve ocorrer ou que não deve ocorrer, dessa forma é possível testar diferentes situações e mapeá-las. Uma tabela contém todas as possibilidades possíveis, mas condições absurdas podem ser removidas.
 - d. **Teste de transição de estado:** Testa comportamentos de componentes quando submetidos a eventos, um diagrama de transição mostra como os estados se modificam e mudam a partir de entradas e saídas do software.
 - e. **Teste de caso de uso:** Testa o comportamento baseado uso do software por parte humana, testa se os erros são tratados com o mau uso, se o fluxo de uso está sendo completado e se o comportamento esperado realmente acontece
2. **Técnicas de teste de caixa-branca:** Esse teste se baseia na estrutura interna do algoritmo, sua implementação. Ele é útil para que se entenda o que acontece com o software de também verificação mais profunda de possíveis falta de tratamento de erros e bad smells.
 - a. **Teste de cobertura de instruções:** É uma porcentagem do números de instruções executadas pelo números total de instruções, serve para mensurar a abrangência das instruções executáveis
 - b. **Teste de decisão e cobertura:** Testa as opções possíveis para tomada de decisão, como if/else e switch, cobrindo todas as saídas a partir de determinadas entradas.
 - c. **O valor da instrução e teste de decisão:** é mais abrangente que o teste de instrução, já que além de executar as instruções, também testa a lógica por trás e o fluxo que uma instrução segue permitindo encontrar erros que não foram vistos em testes anteriores
3. **Técnicas de teste baseada na experiência:** São testes baseados na experiência do testador, os erros mais comuns, os fluxos mais seguidos, padrões que se repetem, etc. É menos sistemática que anterior, mas permite ver erros relacionados a comportamento que não foram vistos, pode ser menos abrangente, já que demanda tempo e esforço muito grandes.

- a. **Suposição de erro:** Supor uma situação que pode evoluir para um erro, defeito ou falha a partir de comportamento ou conhecimento anterior dos desenvolvedores e do software
- b. **Teste Exploratório:** Explora o software a fim de conhecê-lo, pode ser baseado em tempo ou em adaptação, e é importante para identificar etapas de fluxos e documentar as descobertas
- c. **Teste baseado em lista de verificação:** A partir de uma lista de verificação, o testador verifica se os requisitos estão sendo cumpridos, são usadas como suporte a outros tipos de teste

2. Na aula foi citado os testes de caixa-preta, onde o testador não precisa saber o que acontece internamente, mas se uma entrada no software resulta em uma saída satisfatória; testes de caixa-branca, quando se testar conhecendo o código e seu comportamento, a fim de identificar erros não tratados, implementações a serem melhoradas e bad smells. e teste baseado na experiência que usa a experiência do testador para testar o software

3. Teste de caixa cinza: é uma junção de testes de caixa preta e branca, ele usa seus pontos fortes para fazer os testes, juntando o conhecimento interno do software com regras e cenários de sucesso esperados