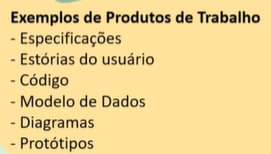
--> Diferentes técnicas de teste:

* Teste estático: Feito na documentação e no código sem ser executado; Antecipação de defeitos.

-Verificar a complexidade do código; Partes mais simples facilitam a manutenção. Documentação deve ser coesa e no padrão correta.

-> Exemplos de coisas a testar:



-> Benefícios: Correção mais barata dos defeitos (quanto antes detectado, mais barato); Reduzir tempo de dev; Aumento da eficiência dos testes.

NOTA: Testes estáticos NÃO são caixa branca

-> Estáticos vs dinâmico:



-> Defeitos de manutenção são normalmente encontrados com testes estáticos;

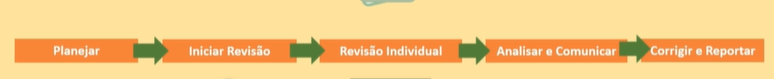
-------------------------------------------------------------------------------------------------------------------------------------

--> O processo de revisão:

-> Formal: Equipe definida, com registro formal do resultado, adequado à normativas internacionais

-Informais: Simplificado, gera resultados mesmo com a desorganização

* Fases do processo de revisão:



-> Os papéis em cada fase:



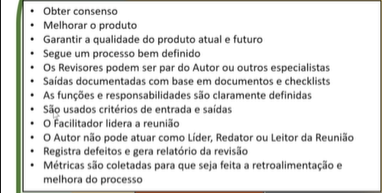
-> Os tipos de revisão:

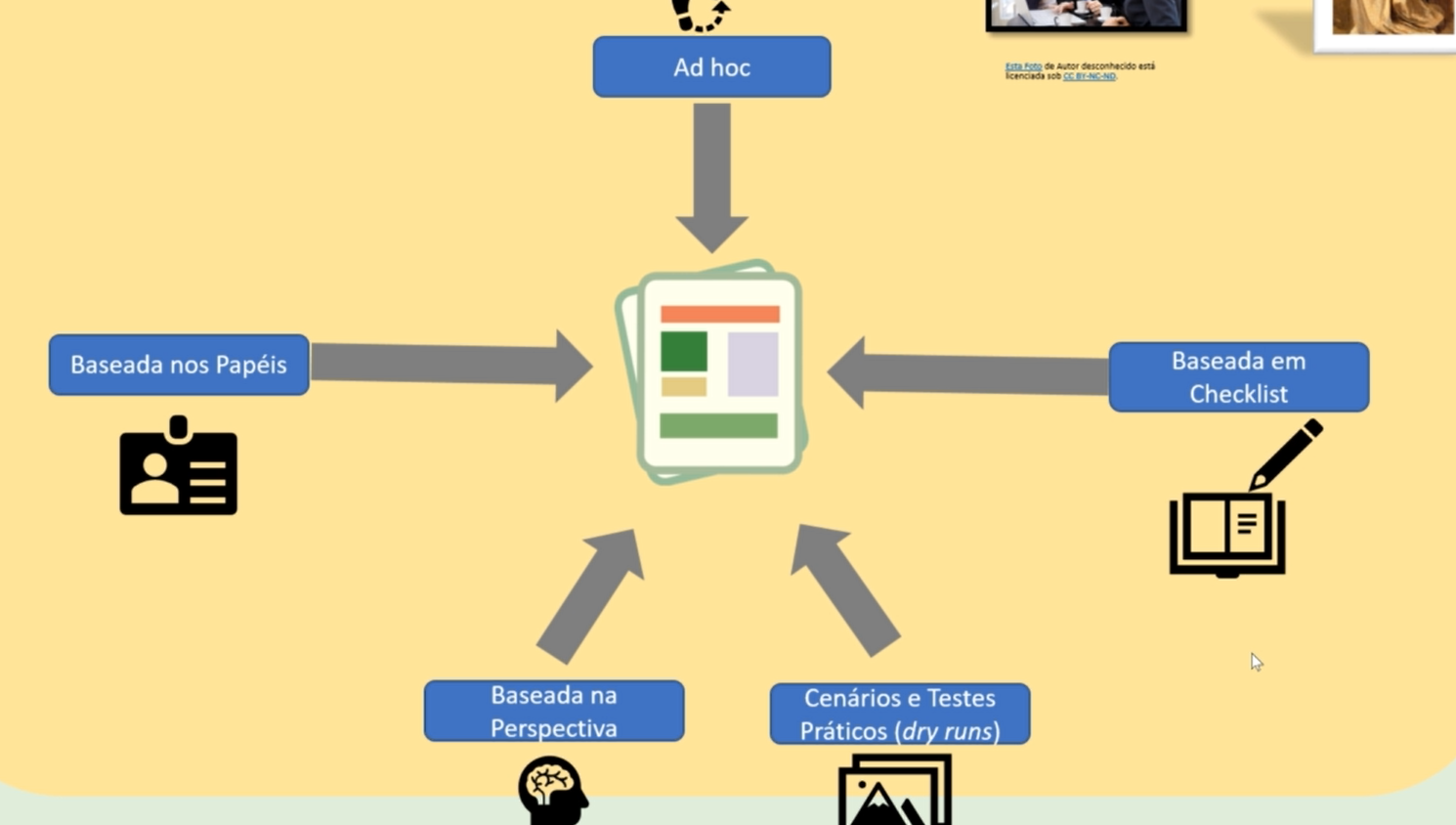
-Informal: Simples, não segue o processo todo; Foco em resolver problemas menores, muito utilizada no método ágil.

-Revisão de acompanhamento: Propõem melhorias e visa obter consensos; Processo liderado pelo AUTOR, com PRESENÇA OBRIGATÓRIA DO RELATOR; Pode usar cenários e gerar logs.

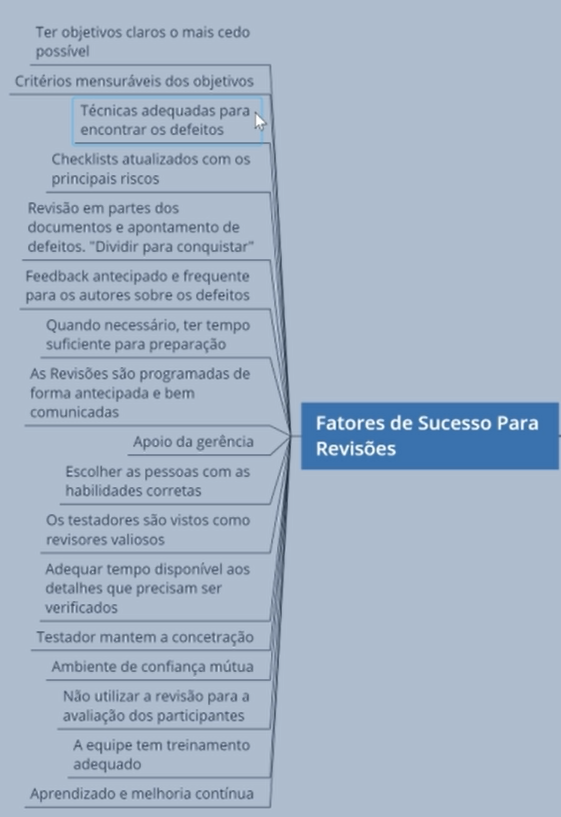
-Revisão técnica: Garantir qualidade e melhorar o produto; Reunião com preparação prévia; REVISOR deve ser par técnico do autor, no papel de facilitador; Elabora relatórios de defeitos

-Inspeção:





-> Fatores de sucesso:



-> Como escolher um teste:

1) Olhe a documentação disponível; Sendo pouca, teste exploratório;

2) Complexidade do sistema/componente; Quanto mais complexo, mais testes.

3) Experiência do testador

4) Modelo de desenvolvimento utilizado

5) Risco do software/necessidade do cliente

6) Ferramentas e orçamento disponível.

7) Tempo disponível para realizar os testes

8) Normas regulatórias a serem seguidas; Expectativa dos defeitos que podem aparecer

* Os testes de caixa preta:

-Particionamento de equivalência: Testar ao menos um valor inválido inferior, um válido, e um inválido superior; COBRE TODO O RANGE DE VALIDADE, mas não é suficiente por si só.

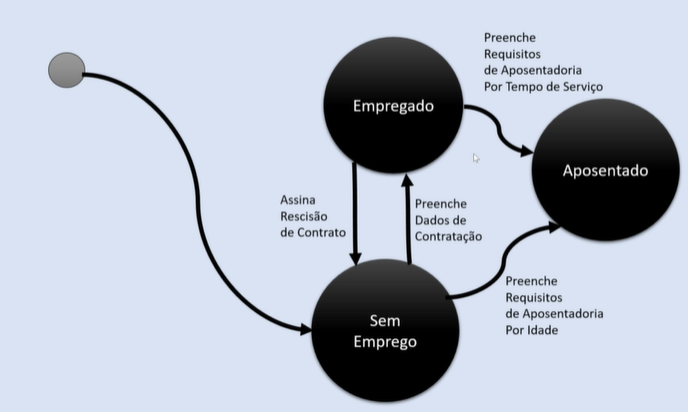
-Análise de valor limite: Testa o limite de cada range. Se o limite não ser aceito, é provável que todos os valores também não sejam; Dois testes válidos e dois inválidos. COSTUMA SER USADO JUNTO COM O PARTICIONAMENTO.

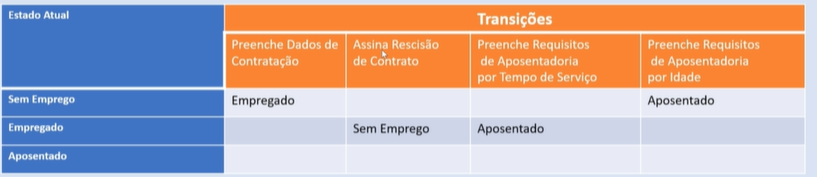
-Teste de transição de estado: Tira “fotografias” do sistema em vários momentos, testando se o resultado obtido é o esperado.

* Transições são sempre iniciadas por EVENTOS.
* Podem ser usados diagramas e tabelas de posição, devendo considerar todos os valores válidos e inválidos.

-> Cobertura é dada por

REGRAS DE DECISÃO DE AO MENOS UM TESTE/ N TOTAL DE REGRAS DE DECISÃO (em %)



  
-O teste de tabela ajuda a identificar todas as combinações de condições, ajudando também a encontrar lacunas nos requisitos.

----//----

-Tabela de decisão: A partir das regras de negócio, cria-se as condições e as ações esperadas, que serão testadas

-Testes de caso de uso: Representam a interação entre o ator e a funcionalidade sendo testada. Cobertura: % casos testados/ % de casos total.

* Existem cenários normais e cenários de exceção e alternativos, todos devem ser testados.
* Testes de caixa branca: Testa a estrutura interna do software; TESTE ESTÁTICO NÃO É CAIXA BRANCA!

-> Geralmente são usados nos testes de componente, mas são aplicáveis em todos.

Teste de instruções e cobertura: Os testes devem passar por todas as possibilidades do código.

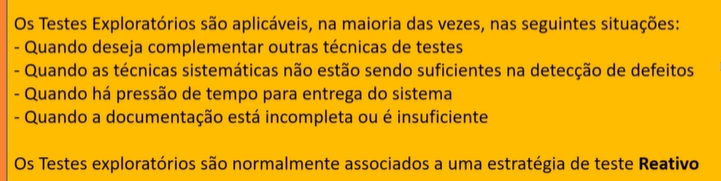
--//---

* Teste baseado em experiência: A montagem dos casos derivam da experiência prévia dos testadores; Servem para complementar as outras técnicas, alcançando possibilidades diferentes.

-Uma pessoa experiente é capaz, também, de supor quais erros podem ocorrer.

* Teste exploratório: Testador conhece a aplicação em si, sabendo como cobrir a quase totalidade da aplicação para descobrir erros a partir de testes práticos;

-Geralmente, o teste é definido temporalmente. Os resultados são registrados em folhas de sessão, com tudo definido no termo de teste.



-Checklist: Lista de “coisas que dão/podem dar defeito” para serem testadas; Essa lista, claro, é ampliada durante o teste, e para uso futuro.