Nome: Raquel Temóteo Eucaria Pereira da Costa
Matícula: 202045268

Delamaro Cap 2

- Teste funcional = caixa-preta
- Avaliado segundo o ponto de vista do usuário
- Detecta defeitos submetendo possíveis entradas *(pode ser infinito – podendo o tempo atividade de teste ser inviável)
- Conduzir a atividade de teste, de forma mais sistemática
- Técnicas: Funcional, estrutural e baseada em defeitos
- Critérios: Particionamento de Equivalência, Análise do Valor Limite, Teste funcional sistemático, Grafo Causa-Efeito, Teste funcional sistemático e Error-Guessing
    - Particionamento de Equivalência: Divide os dados de entrada em classes de equivalência; Cada classe representa um conjunto de dados de entrada que é tratado da mesma forma pelo software; Testa um representante de cada classe de equivalência; Reduz a quantidade de testes necessários, focando em testar os casos mais significativos.
    - Análise do Valor Limite: É uma extensão do particionamento de equivalência; Foca nos valores limites das classes de equivalência.
    - Testa os valores no limite inferior, no limite superior e imediatamente além desses limites; Ajuda a identificar erros comuns relacionados a problemas de borda.
    - Teste funcional sistemático: É uma abordagem estruturada que segue um plano de teste predefinido; Cobre todas as funcionalidades do software de acordo com um roteiro de teste; Geralmente envolve a criação de casos de teste com base em requisitos ou especificações; É abrangente, mas pode ser demorado.
    - Grafo Causa-Efeito: Também conhecido como teste de tabela de decisão; Modela as condições de entrada e as ações de saída do software como uma tabela de decisão; Cria casos de teste com base nas combinações de condições que podem afetar as saídas; Útil para testar situações complexas com múltiplas condições de entrada.
    - Teste funcional sistemático: Também conhecido como teste de tabela de decisão; Modela as condições de entrada e as ações de saída do software como uma tabela de decisão; Cria casos de teste com base nas combinações de condições que podem afetar as saídas; Útil para testar situações complexas com múltiplas condições de entrada.
    - Error-Guessing: Também conhecido como teste de tabela de decisão; Modela as condições de entrada e as ações de saída do software como uma tabela de decisão; Cria casos de teste com base nas combinações de condições que podem afetar as saídas; Útil para testar situações complexas com múltiplas condições de entrada.
- Qualidade do critério depende da boa especificação do requisito
- Podem ser aplicados em todas as fases de teste, não levam em consideração os detalhes da aplicação

- Estratégias
- Considerações finais
    - Precisa somente da especificação do produto para ser aplicado
    - Testes complementares, pois na prática pode dar errado

Gonçalvez - Níveis de Teste (pag. 73 a 80)

Níveis de teste

- Os testes devem ocorrer em todos os estágios do ciclo de vida do projeto, a cada estágio são definidos **níveis de teste,** com diferentes níveis de esforço.
- 3 práticas:
    i. Analistas diferentes daqueles que realizaram a implementação e a codificação do software realizam os testes depois que uma funcionalidade é desenvolvida e antes que ela seja entregue ao usuário final. Pode gerar atraso.
    ii. Funcionalidades testadas a medida que são desenvolvidas – Desde o início.
    iii. Feito o tempo todo (XP)
- Níveis de Teste
    - Unidade
    - Integração
    - Sistema
    - Aceitação
    - Regressão

Teste de sistema

- Nível de teste, onde o propósito é executar o sistema como um todo.
- Leva em consideração a visão do **usuário final.**
- Tem a **intenção de identificar problemas que diferenciem o software entregue daquele software original. -> identificar se ele é ou se parece com aquele produto que havia sido definido no escopo do projeto pela equipe e pelos usuários.**
- Nível de teste de sistema permite testar o sistema como um todo -> tbm permite -> validar a integração entre outros sistemas
- Ciclo de vida iterativo permite uma maior realização deste teste, pois a cada entrega existe um produto funcional que pode ser testado
- Ponta a ponta das funcionalidades

Teste de unidade x Teste de integração

"O nível de teste de unidade se relaciona com a validação de cada parte desenvol-vida do software, normalmente realizada pelo próprio analista desenvolvedor. O teste de integração, por sua vez, tem relação com a integração entre partes do software já desenvolvidas, ou mesmo do software com outros sistemas interdependentes."

- <u>Teste de Unidade</u>
    - Conhecido como teste de módulo
    - Testado as menores partes
    - Validação de chamadas de métodos, classes, funções, sub-rotinas e partes de código-fonte

    - De maneira isolada
    - Documentos: Especificação de requisitos, casos de uso e arquitetura de software (DAS)
    - Com os documentos é possível elaborar os casos de teste
- Teste de integração
    - Encontrar problemas na integração interna dos componentes ou partes do software
    - Garante que os componentes funcionam de forma adequada
    - Forma modelos de pacotes de integração
    - Detecta problemas como: erros, falhas ou defeitos, na própria interface do software
    - analistas + testadores = cada um faz a sua parte
    - apenas a integração
    - pré-requisitos:
        - preparar previamente um ambiente para que o teste aconteça;
        - preparar previamente um banco de dados para servir de teste;
        - registrar formalmente todos os resultados obtidos com os testes;
        - avaliar todos os resultados obtidos por meio dos testes;
        - relatar ao gerente de projeto toda ocorrência de problema mais grave que possa impactar negativamente o andamento do restante do projeto de desenvolvimento do software

Teste de aceitação x regressão

Normalmente, o nível de teste de aceitação diz respeito à aceitação, por parte dos usuários finais, do software entregue. Já o teste de regressão se relaciona com uma nova validação e uma nova repetição de testes já feitos. A ideia é averiguar se modificações não acarretaram algum problema de execução que anteriormente não existia ou já havia sido corrigido

- Teste de aceitação
    - Realizado por umgrupo de amostra
    - “Teste de aceitação do usuário”
    - Teste mais formal com os usuários se envolvendo, se satisfaz para eles os critérios de aceitação
    - Antes que seja implementado
    - Não busca necessariamente defeitos, mas quer estabelecer a confiança e segurança no que se esta sendo feito
- Teste de regressão
    - Aplicado sempre que uma nova versão fica pronta
    - Também aplicado quando é necessário realizar um novo ciclo de teste em toda a aplicação para verificar se os elementos novos não afetaram os antigos
    - “Teste recorrente de mudança”
    - Realização de todos os testes novamente
    - Identificar problemas
    - Bastante demorado, por isso deve ser aplicado de maneira planejada e detalhada
    - Mudanças de estrutura iniciam testes de regressão
    - Principais causas de mudança de estrutura:
        - Correção de erro
        - Alteração de funcionalidade
        - Desenvolvimento de nova funcionalidade
    - Diz se o software regrediu ou não

Gonçalvez - Testes Caixa Preta (pag. 131 a 144)