Curso: Sistemas de Informação – Unidade São Gabriel
Disciplina: Fundamentos de Testes de Software – Período: 5º - Turno: Noite
Professor: Claudiney Vander Ramos
Data de Entrega: 31/03/2020
Lista 1 – Verificação e Validação (V&V); Abordagens de Testes de Software

1) Descreva a diferença entre validação e verificação.
2) Explique as dimensões do teste de software: O que testar? Como testar? Quando testar?
3) Definir teste de software. Explicar a importância dos testes de software e suas limitações.
4) O que é um teste bem sucedido? Considerando as visões de testes de defeitos e testes de validação.
5) O que é um caso de teste?
6) O que é um bom caso de teste?
7) O que são testes exaustivos?
8) Quais as fases (etapas) dos testes?
9) Qual a diferença entre teste caixa-branca (estrutural) e teste caixa-preta (funcional)?
10) Qual a diferença entre teste baseado em especificação e teste baseado em programa (código ou implementação)?
11) O que é uma técnica de teste?
12) O que é um critério de teste?
13) O que é um grafo de fluxo de programa?
14) O que é teste baseado em fluxo de dados?
15) O que é teste baseado em fluxo de controle?
16) Testes caixa-branca servem para identificar problemas de software com relação à estrutura do produto. Explique qual a importância dos “testes de caminho”.
17) Dê um exemplo de testes de interface onde podem surgir problemas de software devido ao encapsulamento de procedimentos.
18) Um programa lê três valores inteiros. Os três valores são interpretados como representando os comprimentos dos lados de um triângulo. O programa imprime uma mensagem que declara se o triângulo é escaleno, isósceles ou equilátero. Desenvolva um conjunto de casos de teste que você ache adequado para testar esse programa.
19) Projete e implemente o programa (com manipulação de erros onde for apropriado) especificado no exercício anterior. Crie um grafo de fluxo para o programa e aplique teste de caminho básico para desenvolver casos de teste que irão garantir que todos os comandos do programa sejam testados. Execute os casos e mostre seus resultados.

20) Selecione um componente de software que você tenha projetado e implementado recentemente. Projete um conjunto de casos de teste que garanta que todos os comandos tenham sido executados usando teste de caminho básico.

21) Apresente um exemplo no qual o teste caixa-preta pode dar a impressão de que "está tudo certo", enquanto o teste caixa-branca poderia detectar um erro (defeito). Apresente um exemplo no qual o teste caixa-branca poderia dar a impressão de que "tudo está certo", enquanto o teste caixa-preta poderia descobrir um erro (defeito).

22) Teste de caminho.

Para cada uma das funções abaixo:
1) Elaborar o grafo do programa
2) Determinar os caminhos completos
3) Criar casos de teste que executem estes caminhos

a) O método a seguir, escrito em C#, recebe uma String **`str`** e ajusta seu tamanho ao tamanho da variável **`tam`**. Se a String for menor que **`tam`**, caracteres (espaços) adicionais são adicionados à palavra; se a String for maior que **`tam`**, os caracteres adicionais são "cortados".

```
String AjustaStr (String str; Int tam) {
  while (str.Length() < tam) do {
    str = str + ' ';
  }
  if (str.Length() > tam) {
    str = str.Substring(0,tam);
  }
  return str;
}
```

b)

/* Este programa escrito em C lê uma linha de texto e converte conjuntos de
caracteres brancos em um unico caractere. Sugira casos de teste para o programa */

```
void eliminaBrancos(char[] linha) {
  int i,j,tamanho;
  i=0; tamanho = linha.Length();
  while(i<tamanho) {
    if(IsWhiteSpace(linha[i])) /* verifica se é "branco" */
    {
      if(IsWhiteSpace(linha[i+1])) /* verifica se o próximo é "branco" */
        for (j=i;j<tamanho;j++ )
        linha[j]=linha[j+1]; /* copia o proximo caractere para
        /* a posicao vazia*/
      else
        i++;
    }
    else
      i++;
  }
} // fim do eliminaBrancos()
```

c)

O programa abaixo abre um arquivo de texto e imprime todas as linhas.

```
public void cat(File file) {
   RandomAccessFile input = null;
   String line = null;
   try {
      input = new RandomAccessFile(file, "r");
      while ((line = input.readLine()) != null) {
         System.out.println(line);
      }
      return;
```

```
    }
    catch(FileNotFoundException fnf) {
       System.err.println("Arquivo*** "+ file+" ***não encontrado");
    }
    catch(IOException e) {
       System.err.println(e.toString());
    }
    finally {
       if (input != null) {
         try {
            input.close();
         }
         catch(IOException io) {}
       }
    }
}
```

d)

A função recebe três valores e verifica se eles podem formar um triângulo. Três lados formam um triângulo quando um lado é menor que a soma dos outros dois.

```
function verificaTriangulo (a, b, c : real): String;
begin
   if ((a < b + c) AND (b < a + c) AND (c < a + b)) then
     if (a = b) AND (b = c) then
       verificaTriangulo:='Triangulo equilatero' {Três lados iguais}
     else if ((a = b) OR (a = c) OR (b = c)) then
        verificaTriangulo:='Triangulo isosceles' {Dois lados iguais}
     else
        verificaTriangulo:='Triangulo escaleno'
   else verificaTriangulo:='Nao é um triangulo'; {Não satisfez a propriedade}
end;
```

23) Questões do ENADE (2005, 2008, 2011)

Prova 2005 - Questão 34

Julgue os seguintes itens referentes a teste de *software*.

I A técnica de teste funcional, que estabelece os requisitos de teste com base em determinada implementação, permite verificar se são atendidos os detalhes do código e solicita a execução de partes ou de componentes elementares do programa; a técnica de teste estrutural aborda o *software* de um ponto de vista macroscópico e estabelece os requisitos de teste, com base em determinada implementação.

II Na fase de teste de unidade, o objetivo é explorar-se a menor unidade de projeto, procurando-se identificar erros de lógica e de implementação de cada módulo; na fase de teste de integração, o objetivo é descobrir erros associados às interfaces entre os módulos quando esses são integrados, para se construir a estrutura do *software*, estabelecida na fase de projeto.

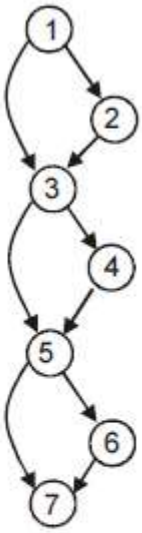
III Critérios com base na complexidade, em fluxo de controle e em fluxo de dados, são utilizados pela técnica estrutural de teste.

Assinale a opção correta.

A Apenas um item está certo.
B Apenas os itens I e II estão certos.
C Apenas os itens I e III estão certos.
D Apenas os itens II e III estão certos.
E Todos os itens estão certos.

Prova 2008 - Questão 12

Ao longo de todo o desenvolvimento do *software*, devem ser aplicadas atividades de garantia de qualidade de *software* (GQS), entre as quais se encontra a atividade de teste. Um dos critérios de teste utilizados para gerar casos de teste é o denominado critério dos caminhos básicos, cujo número de caminhos pode ser determinado com base na complexidade ciclomática. Considerando-se o grafo de fluxo de controle apresentado na figura ao lado, no qual os nós representam os blocos de comandos e as arestas representam a transferência de controle, qual a quantidade de caminhos básicos que devem ser testados no programa associado a esse grafo de fluxo de controle, sabendo-se que essa quantidade é igual à complexidade ciclomática mais um?



A 1. **B** 3. **C** 4. **D** 7. **E** 8.

Prova 2008 - Questão 16

O gerenciamento de configuração de *software* (GCS) é uma atividade que deve ser realizada para identificar, controlar, auditar e relatar as modificações que ocorrem durante todo o desenvolvimento ou mesmo durante a fase de manutenção, depois que o *software* for entregue ao cliente. O GCS é embasado nos chamados itens de configuração, que são produzidos como resultado das atividades de engenharia de *software* e que ficam armazenados em um repositório. Com relação ao GCS, analise as duas asserções apresentadas a seguir.

No GCS, o processo de controle das modificações obedece ao seguinte fluxo: começa com um pedido de modificação de um item de configuração, que leva à aceitação ou não desse pedido e termina com a atualização controlada desse item no repositório

porque

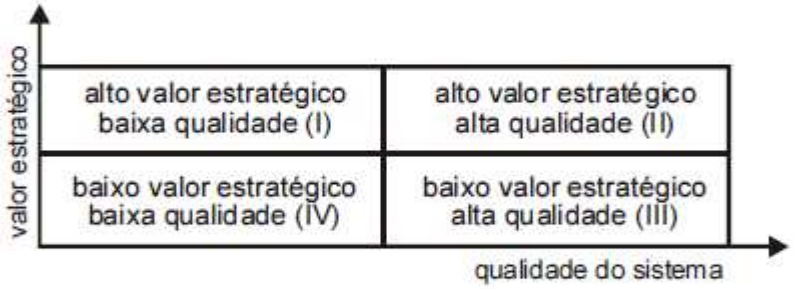
o controle das modificações dos itens de configuração baseia-se nos processos de *check-in* e *check-out* que fazem, respectivamente, a inserção de um item de configuração no repositório e a retirada de itens de configuração do repositório para efeito de realização das modificações.

Acerca dessas asserções, assinale a opção correta.

- **A** As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- **B** As duas asserções são proposições verdadeiras, e a segunda não é uma justificativa correta da primeira.
- **C** A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
- **D** A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
- **E** As duas asserções são proposições falsas.

Prova 2008 - Questão 69

Um ponto crítico para as organizações é a gerência de seus sistemas legados. Quanto a esses sistemas, é importante decidir se eles devem sofrer uma reengenharia, sendo reimplementados, ou não. Essa decisão é tomada após se avaliarem os sistemas legados com base em dois parâmetros: valor estratégico para a organização, ou seja, o valor que ele agrega para os serviços e produtos da organização; e qualidade do sistema, ou seja, o custo de manutenção uma vez que sistemas de baixa qualidade possuem alto custo de manutenção. Essa avaliação classifica esses sistemas de acordo com as situações de I a IV indicadas abaixo.



Em qual(ais) dessas situações um sistema legado deve ser classificado para ser indicado a uma reengenharia?

- (A) Apenas na situação I.
- (B) Apenas na situação IV.
- (C) Apenas nas situações I e II.
- (D) Apenas nas situações II e III.
- (E) Apenas nas situações III e IV.

Prova 2011 - Questão 19

Uma equipe está realizando testes com base nos códigos-fonte de um sistema. Os testes envolvem a verificação de diversos componentes individualmente, bem como das interfaces entre os componentes.

No contexto apresentado, essa equipe está realizando testes em nível de

- (A) unidade.
- (B) aceitação.
- (C) sistema e aceitação.
- (D) integração e sistema.
- (E) unidade e integração.