



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP2 1º semestre de 2011.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

-
- 1) O que significa uma relação de uso entre módulos no projeto de software? Qual é a importância destas relações de uso na melhoria do projeto? (valor: 2,0 pontos; máximo: 10 linhas)

A relação de uso é uma associação direcionada entre dois módulos componentes do código-fonte de um sistema de software, que ocorre quando um módulo depende do outro para cumprir seus objetivos. A dependência pode ocorrer pelo uso de uma rotina, uma variável ou um tipo definido no segundo módulo. O conjunto de relações de uso dos módulos de um sistema forma um grafo e se este grafo não for reduzido a uma árvore, poderemos ter um sistema onde “nada funcione até que tudo funcione”. Sendo assim, a organização dos módulos em árvores facilita os testes e a manutenção do sistema.

- 2) Dizemos que um projeto de software deve ser gerenciado em cinco grandes etapas: inicialização, planejamento, execução, controle e fechamento. Explique o papel de cada uma destas etapas no gerenciamento de projetos de software. (valor: 2,0 pontos; máximo: 15 linhas)

A **iniciação** representa o reconhecimento pela alta administração da empresa sobre o início de um projeto.

O **planejamento** envolve a definição, revisão e manutenção de uma organização de trabalho para a realização do projeto.

A **execução** consiste na coordenação das pessoas e recursos necessários para a execução do plano.

O **controle** envolve a monitoração e medição do progresso para garantir que os objetivos do projeto serão atingidos.

Finalmente, o **fechamento** consiste na homologação e encerramento das atividades do projeto.

- 3) O que representa a métrica Complexidade Ciclomática (McCabe)? Que artefato de software (dê um exemplo) pode ser utilizado para observá-la e como ela pode ser obtida ou calculada? (valor: 2,0 pontos)

A métrica fornece uma medida quantitativa da complexidade lógica de um programa. No contexto do teste estrutural, seu valor define o número de caminhos independentes e nos fornece o número máximo de casos de teste que garantem que todos os comandos tenham sido executados pelo menos 1 vez. Um exemplo de artefato de software que ela pode ser observada e no grafo de fluxo de programa.

Pode ser obtida através da identificação do número de regiões do grafo de fluxo

$$V(G)=E-N+2$$

E: número de arcos

N: número de nós

ou, $V(G) = P + 1$

P: número de nós predicados (decisões)

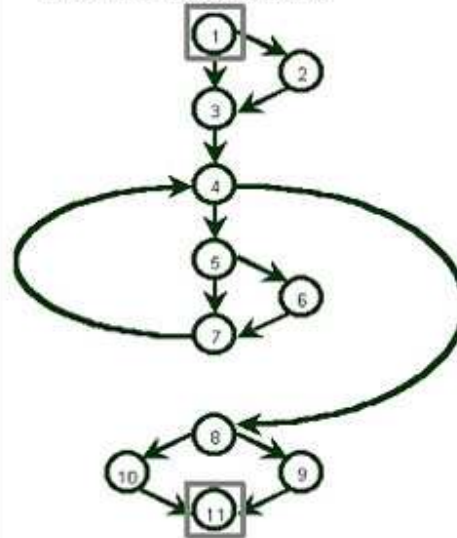
Exemplo de um grafo de programa:

Programa Identifier.c [Função Main]

```
/* 01 */ {
/* 01 */     char achar,
/* 01 */     int length, valid_id,
/* 01 */     length = 0,
/* 01 */     printf("Identificador: "),
/* 01 */     achar = fgetc(stdin),
/* 01 */     valid_id = valid_s(achar),
/* 01 */     if (valid_id)
/* 02 */         length = 1;
/* 03 */     achar = fgetc(stdin);
/* 04 */     while (achar != '\n')
/* 05 */     {
/* 05 */         if (!valid_f(achar))
/* 06 */             valid_id = 0;
/* 07 */         length++;
/* 07 */         achar = fgetc(stdin);
/* 07 */     }
/* 08 */     if (valid_id && (length >= 1) && (length < 6))
/* 09 */         printf("Valido\n");
/* 10 */     else
/* 10 */         printf("Invalido\n");
/* 11 */ }
```

(a)

Grafo de Fluxo de Controle



(b)

- 4) Defina os casos de teste para estes requisitos usando a técnica de análise de valor limite (Valor 2 pontos):

Em uma indústria química o controle da temperatura e pressão de um determinado processo depende de dois fatores: a temperatura de ebulição do produto sendo processado e de um coeficiente Z. As entradas

do programa de controle são a temperatura de ebulição do produto em graus Celsius limitada ao intervalo [80;210] e o coeficiente Z limitado ao intervalo [0.2;2.5]. O sistema de controle deve suspender o processo se a temperatura ou a pressão atingirem valores críticos.

O cálculo do valor crítico da temperatura é obtido pela fórmula: $t_k = t_e * 2$.

O valor crítico da pressão pela fórmula:

$p_k = t_k * Z$, quando t_e for menor que 105;

$p_k = t_e * Z$ nos demais casos.

Onde, t_k =temperatura crítica, t_e =temperatura de ebulição, p_k =pressão crítica.

Como este software foi construído visando *testabilidade*, é possível entrar com um terceiro e quarto valores correspondentes à temperatura e a pressão em um dado instante. Nesse caso, o software efetua os cálculos e imprime uma mensagem sinalizando se interromperia ou não o processo.

t_e : {79, 80, 104, 105, 210, 211}; Z : {0.1, 0.2, 1.2 (intermediário), 2.5, 2.6}

| t_e | Z | t_k | p_k | t_i | p_i | Resultado Esperado |
|-------|-----|-------|-------|-------|-------|--------------------|
| 79 | 0.1 | - | - | - | - | Valores inválidos |
| 80 | 0.2 | 160 | 32 | 159 | 31 | Continuar Processo |
| 104 | 1.2 | 208 | 229.6 | 208 | 220 | Suspender Processo |
| 105 | 1.2 | 210 | 126 | 100 | 126 | Suspender Processo |
| 210 | 2.5 | 420 | 525 | 421 | 526 | Suspender Processo |
| 211 | 2.6 | 422 | - | - | - | Valores inválidos |

- 5) Defina **defeito** e **falha** e indique quais técnicas podem ser usadas para identificar cada um deles. (10 linhas, valor 2 pontos).

- Defeito:
 - o Deficiência mecânica ou algorítmica que se ativada pode levar a uma falha.
- Falha:
 - o Evento notável onde o sistema viola suas especificações.

Defeitos são encontrados por inspeção do software. Falhas são reveladas pelos testes.