



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP2 2º semestre de 2014.

Nome –

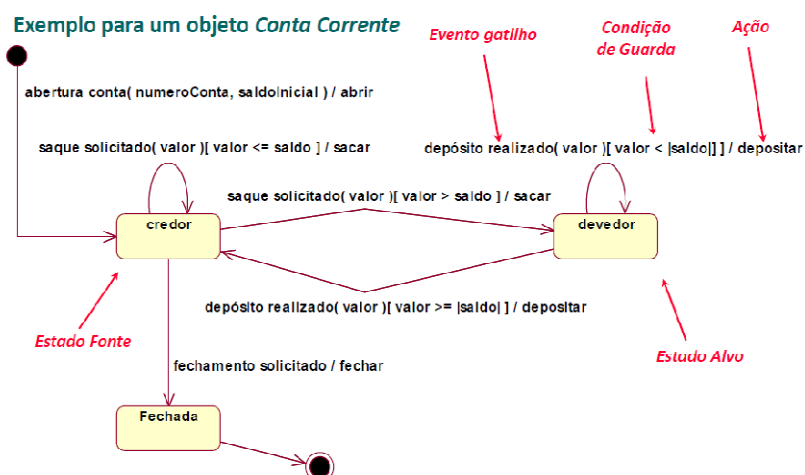
Assinatura –

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 representa uma Expressão-Ação em um diagrama de estados? Ela deve ser opcional ou obrigatória? Dê um exemplo. (Valor: 1,0 ponto; máximo: 10 linhas)

Uma Expressão-ação indica a expressão que deve ser executada quando o objeto receber um evento gatilho ("trigger"). Sua execução depende da condição de guarda ter sido satisfeita ou não.



- 2) Porque o uso de bons nomes para variáveis e rotinas facilita o entendimento de um programa? (valor: 1,0 ponto; máximo: 10 linhas)

O código-fonte de um programa de computador é um artefato bastante complexo e difícil de entender. Assim, todos os esforços devem ser feitos para que estes artefatos sejam mais legíveis, de modo a reduzir as chances de um desenvolvedor introduzir um erro no sistema por desconhecimento sobre como realizar uma alteração no código. Comentários, boa organização, simplicidade, boa nomenclatura para os componentes do programa (como classes, métodos e variáveis) são ferramentas que auxiliam na clareza dos programas. Bom nome para rotina ajuda o desenvolvedor a entender seu objetivo, enquanto um bom nome para as variáveis e parâmetros facilita o entendimento da informação ali armazenada no contexto da rotina.

- 3) Quais são as três características fundamentais de um projeto de software que fazem com que ele seja diferente de uma operação continuada? (valor: 1,0 ponto; máximo: 10 linhas)

O projeto é temporário, tendo data de início e encerramento, gera um produto ou serviço único e é realizado de forma progressiva. Por temporário, entende-se que o projeto possui um início e um término definidos (não dura para sempre!). Por único, entende-se que existem aspectos no produto ou serviço sendo criado que o diferenciam de produtos ou serviços anteriores. Por progressivo, entende-se que as características de cada projeto são mais detalhadas na medida em que maior for o entendimento do produto ou do serviço.

- 4) Marque Verdadeiro (V) ou Falso (F) para as afirmações abaixo (valor até 2,0 pontos – 0,2 por marcação correta. Atenção: cada 2 erros/2 brancos/ 1 erro, 1 branco eliminam 1 acerto!):

A) V; B) V; C) F; D) V; E) F; F) F; G) V; H) V; I) F; J) F

- A. As técnicas de inspeção *ad-hoc*, apenas quando aplicadas por inspetores inexperientes, não garantem a cobertura do documento como um todo.
- B. Inspeções de Software capturam, em geral, cerca de 60% dos defeitos. Por isso, necessitam ser complementadas por outras atividades de garantia de qualidade, tais como análise de desempenho.
- C. Inspeções de Software podem ser aplicadas a diferentes artefatos do ciclo de vida. Por isso, ao usar inspeções não é necessário aplicar testes.
- D. O perfil de experiência do inspetor afeta diretamente a eficiência da inspeção *ad-hoc*.
- E. O custo da correção de um defeito se mantém constante ao longo do ciclo de vida do software.
- F. Os melhores testadores para o software são os inspetores.

- G. Defeitos são usualmente gerados na comunicação e na transformação das informações ao longo do ciclo de vida do software.
 - H. Prevenir Defeitos é melhor que Remover Defeitos.
 - I. Os resultados dos testes provam que o software está completamente livre de defeitos.
 - J. Os custos e resultados inspeções e testes de software são equivalentes
- 5) O que representa a métrica Complexidade Ciclomática? Como ela pode ser calculada? O que seu valor define no contexto do teste estrutural? (Valor: 2,0 pontos; máximo 10 linhas)

Esta métrica que 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 uma vez.

Seu calculo se dá através da identificação do número de regiões no grafo de fluxo do programa, através da formula

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

E: número de arcos

N: número de nós

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

P: número de nós predicados (decisões)

- 6) Explique a diferença entre teste funcional e teste estrutural, apresentando pelo menos dois exemplos de critérios que podem ser utilizados para projetar casos de teste. (Valor: 2,0 pontos):

Teste Funcional: Baseia-se na especificação do software para derivar os requisitos de teste. Aborda o software de um ponto de vista macroscópico. Critérios: Particionamento em Classes de Equivalência; Análise do Valor Limite; Grafo de Causa-Efeito.

Teste Estrutural: É baseada no conhecimento da estrutura interna da implementação. Teste dos detalhes procedimentais. A maioria dos critérios dessa técnica utiliza uma representação de programa conhecida como grafo de programa ou grafo de fluxo de controle. Critérios: Todos os caminhos, todos os usos, todas as definições, todos os usos computacionais, ...

- 7) Liste as fases do processo de inspeção de software, apresente o objetivo de cada uma e que papéis são necessários para sua execução. (Valor: 1,0 ponto, 10 linhas).

Planejamento

Responsável: Moderador.

Tarefas: Definir contexto da inspeção (descrição da inspeção, como a preparação individual deverá ocorrer, documento a ser inspecionado, autor do documento, entre outros); Selecionar inspetores (recomenda-se utilizar entre 3 e 5 inspetores em uma inspeção); Distribuir material.

Preparação Individual

Responsável: Inspetor.

Tarefas: Estudar os artefatos; Fazer anotações sobre os artefatos.

Reunião de Inspeção

Envolvidos: Moderador, Inspetores e Autor.

Tarefas: Leitura do documento, com a equipe discutindo possíveis defeitos (Duração recomendada 2 h); Produzir uma lista de defeitos; Em casos de discordância a decisão sobre registrar um defeito ou não (falso positivo) é do moderador.

Retrabalho

Responsável: Autor.

Tarefas: Corrigir os defeitos encontrados.

Continuação

Responsável: Moderador.

Tarefas: Analisar correções do autor e inspeção como um todo; Re-avaliar qualidade do artefato inspecionado; Decidir sobre a necessidade de uma nova inspeção.