



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP2 2º semestre de 2009.**

**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) Dizemos que clareza é uma característica de um bom programa (código). Justifique esta afirmativa, considerando a relação entre o custo de desenvolvimento e o custo de manutenção de um sistema de software. (valor: 1,0 ponto; máximo: 10 linhas).

Em geral, observamos que manter um sistema (criar novas funcionalidades, alterar funcionalidades previamente existentes, corrigir erros, etc) é mais caro do que construir um sistema. Isto ocorre porque a equipe envolvida na construção original geralmente não é responsável pela manutenção e o conhecimento sobre o projeto se perde entre uma tarefa e outra. Para evitar o custo associado a esta manutenção, devemos desenvolver o sistema pensando em quem vai cuidar de sua manutenção no futuro. Comentários, boa organização, simplicidade, boa nomenclatura para os elementos componentes do programa (como classes, métodos e variáveis) são ferramentas que auxiliam a clareza.

- 2) Explique o que é um “número mágico” no código-fonte de um programa. Como este problema de programação pode ser resolvido? (valor: 1,0 ponto; máximo: 10 linhas).

Um número mágico é uma constante com significado próprio (por exemplo, PI), mas que é apresentada numericamente no código de um sistema (por exemplo, 3,14). Números mágicos devem ser declarados como constantes e seu nome deve ser utilizado ao invés de seu valor no código. Geralmente, um desenvolvedor terá mais facilidade de entender o porquê de um valor se ele estiver associado a um nome que o descreva.

- 3) Diferencie os “projetos” das “operações continuadas” (ou apenas “operações”), segundo a interpretação corrente do gerenciamento de projetos (visão do PMBOK). Cite as três características fundamentais de um projeto. (valor: 2,0 pontos; máximo: 10 linhas)

Operações continuadas representam tarefas que se repetem no dia-a-dia de uma empresa (como manutenção dos serviços e processos). Um projeto é um esforço temporário, realizado com o intuito de criar um produto ou serviço único. O resultado único a ser gerado pelo projeto, o fato de ser temporário (com data para começar e acabar) e o fato de ser realizado de forma incremental diferenciam os projetos das operações continuadas.

- 3) Descreva o significado dos conceitos de coesão e acoplamento em software. Indique qual a combinação mais adequada entre eles. Explique como estes conceitos podem ser observados/capturados do software, dando um exemplo para cada um. (Valor: 2,0 pontos; máximo 10 linhas)

Acoplamento: Intuitivamente, refere-se ao grau de interdependência entre diferentes pedaços de um projeto. Desde que objetos de uma mesma classe possuem as mesmas propriedades, duas classes estão acopladas quando métodos declarados numa classe utilizam métodos ou variáveis de instância de outra classe. Excesso de acoplamento aumenta o risco de falha, já que torna os módulos muito dependentes entre si, podendo levar a efeito colateral (um módulo influencia outro). Uma métrica que pode ser usada para observar acoplamento é a CBO (acoplamento entre objetos)

Coesão: Intuitivamente, refere-se à consistência interna dos pedaços de um projeto. O grau de similaridade de métodos pode ser visto como o maior aspecto de coesividade das classes/objetos. Se uma classe/objeto tem diferentes métodos executando diferentes operações para um mesmo conjunto de variáveis de instância, a classe é coesa. Pouca coesão também pode levar a problemas de falha e a uma modularidade inadequada do software. Uma métrica que pode ser usada para observar a coesão em classes é LCOM (Perda de Coesão em Métodos).

O ideal é sempre ter alta coesão e baixo acoplamento.

- 4) No método de inspeção de software, para a fase de detecção de defeitos, é possível utilizar *checklists*. Explique o que é um *checklist* e cite pelo menos 2 diferenças quando comparada com a técnica *ad-hoc*. (Valor: 2,0 pontos; máximo 10 linhas)

*ad-hoc*: a revisão do documento é realizada seguindo a experiência do próprio inspetor, não há garantia de repetição e o resultado é bastante dependente de quem realiza a inspeção. Portanto a relação custo/eficiência (número de defeitos/tempo de inspeção) pode não ser muito adequada. Para se ter maior rendimento é preciso utilizar inspetores experientes, aumentando assim o custo da inspeção.

*Checklist*: um roteiro básico descrevendo as características de qualidade que devem ser observadas durante a revisão é fornecido ao inspetor, que utiliza este

roteiro como guia para realizar a inspeção. Assim, é possível direcionar a revisão, aumentando o rendimento dos inspetores e melhorando a relação custo/eficiência. Por outro lado, um checklist deve ser previamente preparado e ajustado para ser usado num projeto de software.

- 5) Alguns critérios podem ser utilizados para projetar casos de teste. Descreva um destes critérios, indicando se ele deve ser usado para teste funcional ou teste estrutural e dê um exemplo de sua utilização (valor: 1,5 pontos; máximo: 10 linhas).

**\*\* existem diferentes respostas aqui. Veja o material do curso \*\*** Abaixo uma possível resposta:

Um critério que pode ser utilizado para projetar casos de teste funcionais é denominado análise de valor limite. Por este critério, casos de teste são projetados observando-se os limites das informações. Por exemplo, no caso de um determinado conceito que se encontra na faixa 2..10, os possíveis casos de teste poderiam ser: não permitido: qualquer valor  $< 2$  ou  $> 10$ ; qualquer valor entre 2..10, sendo que se dá preferência para os valores 2,10 e algum valor entre estes.