



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP2 2º semestre de 2010.

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) Explique a relação existente entre os processos da área de conhecimento de tempo e da área de conhecimento de custo em um projeto de desenvolvimento de software. (valor: 2,0 pontos; máximo: 10 linhas)

O principal componente de custo dos projetos de software é o custo do trabalho de equipe responsável pelo desenvolvimento do projeto. Assim, o tempo necessário para realizar o projeto está diretamente relacionado ao seu custo. Os processos da área de conhecimento de tempo permitem o planejamento e controle do tempo de projeto, decompondo o projeto em atividades e indicando as dependências e duração destas atividades. Os processos de gerenciamento de custos indicarão os recursos necessários para realizar estas atividades, informando seus custos. Assim, as atividades que compõem o projeto (definidas no gerenciamento de tempo) influenciam diretamente os custos do projeto (gerenciamento de custos).

- 3) Utilize a abordagem de particionamento por equivalência para representar os casos de teste para o seguinte requisito de software (2,0 pontos):

“Uma companhia telefônica realiza a tarifação das ligações comerciais considerando as categorias de horário F1 e F2. Nos dias úteis as ligações para as categorias F1 e F2 são tarifadas por minuto em R\$ 0,04 e R\$ 0,05 das 8:00 as 20:00 hrs e R\$ 0,01 e R\$ 0,02 das 20:01 as 07:59 hrs, respectivamente. Para os finais de

semana e feriados, as tarifas das ligações para F1 e F2 são R\$ 0,02 e R\$0,03 respectivamente e em qualquer horário.”

Condições:

Dia da Semana: {útil, não útil}

Horários: {8:00, 20:00} e {20:01, 7:59}

Tarifas F1_util: {0,04, 0,01}

Tarifas F2_util: {0,05, 0,02}

Tarifa F1_nao útil: {0,01}

Tarifa F2_nao útil: {0,02}

Classes Válidas

Horarios da ligacao: qualquer valor entre 8:00 e 20:00 e qualquer valor entre 20:01 e 7:59 (as classes invalidas na verdade representam complementos pois os horários são complementares)

Dia da semana: útil (segunda a sexta) e não útil (feriados e finais de semana) (neste caso não existe classe invalida pois um dia vai ser util ou não)

Tarifas: F1 e F2

O valor das tarifas se refere ao resultado esperado. Neste caso, e preciso construir casos de teste para atender as classes validas e para classe invalida de tarifa (valor diferente de F1, F2):

CT: {F1, segunda, 9:00}, {F1, sexta, 21:00}, {F2, segunda, 9:00}, {F2, sexta, 21:00} {F2, sábado, --}, {F1, domingo, --}, {F9, --, --}

- 4) No método de inspeção de software, para a fase de detecção de defeitos, é possível utilizar, a princípio, 3 tipos de técnicas. Cite quais são estas técnicas, apresentando a diferença básica entre elas. Destas 3 técnicas, indique aquela que permite ter mais controle na inspeção e a que permite mais liberdade ao inspetor para identificação de defeito. (Valor: 2,0 pontos, máximo: 10 linhas)

As três técnicas conhecidas são **ad-hoc** (a revisão do documento é realizada seguindo a experiência do próprio inspetor), **checklist** (um roteiro básico descrevendo as características de qualidade que devem ser observadas é fornecido ao inspetor) e **técnicas de leitura** (onde todo um conjunto de instruções concretas para se identificar o defeito é entregue ao inspetor, que deve seguir estes procedimentos para ler o documento).

Destas técnicas a que permite maior liberdade, porém menor rendimento é a **ad-hoc**. A que permite mais controle é a **técnica de leitura**.

- 5) Defina o que é métrica objetiva e métrica subjetiva. Apresente 1 exemplo para cada uma no contexto do software e indique para que a métrica pode ser usada. (Valor: 2,0 pontos, máximo: 10 linhas)

Uma métrica objetiva é aquela que representa uma medida que pode ser extraída diretamente de um produto ou de um processo. Exemplo de métrica objetiva é Linhas de Código ou número de defeitos. Estas métricas, por exemplo, podem ser usadas para fornecer uma indicação sobre o tamanho do software ou o número de defeitos identificados em um determinado artefato, respectivamente.

Uma métrica subjetiva é aquela que representa uma estimativa de extensão ou aplicação de alguma técnica. Uma classificação ou qualificação do problema ou experiência. Exemplos de métricas subjetivas são qualidade de uso de uma técnica ou nível de experiência de um inspetor. Estas técnicas apresentação, em geral, a percepção do desenvolvedor frente a uma determinada característica de qualidade não mensurável e associada ao software.