



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP3 1º semestre de 2019 (GABARITO).**

- 1) Explique o que é a inversão de comando na codificação de rotinas e porque uma situação como esta deve ser evitada. (valor 2,0 pontos)

Os parâmetros de uma rotina devem conter apenas operandos, não opções de execução da rotina. Um operando representa um objeto ou valor que a rotina utiliza para calcular seus resultados. Uma opção indica um modo de operação da rotina, ou seja, uma informação que determina a função cumprida pela rotina. Se a rotina depende de um modo de operação, ela não é independente ou está implementando dois códigos diferentes. No primeiro caso, a rotina deve ser reprojetaada pois ela não deve depender de seus clientes. No segundo caso, a rotina deve ser dividida em duas, uma para cada operação.

- 2) Quais são as principais características de um projeto no gerenciamento de projetos? Dica: estas são as características que diferenciam o projeto de uma operação continuada. (valor 2,0 pontos)

Um projeto é um esforço temporário que tem como objetivo desenvolver um produto ou serviço único de forma incremental.

- 3) Para as métricas (a-e) abaixo, indique seu nível de precisão (objetiva, subjetiva), escala (nominal, ordinal, intervalar, razão) e objeto de medição (processo, produto). Apresente para cada uma delas um exemplo de comportamento que é possível observar no contexto do projeto de software. (valor: 2.0 Pontos)
- a) Dívida Técnica:
  - b) Acoplamento entre Objetos:
  - c) Nível de Maturidade em Processo:
  - d) Facilidade de Uso do Software:
  - e) Cobertura dos Testes:

**RESPOSTA:**

a) Dívida Técnica: objetiva, escala intervalar ou razão, produto. Indica o nível de dívida (diretamente proporcional ao esforço ou dificuldade de manutenção e/ou evolução) do produto de software. Diferentes medidas podem ser extraídas de diferentes artefatos de projeto. Independentemente de onde é extraída, seu valor está diretamente ligado ao conceito da qualidade interna do produto.

b) Acoplamento entre Objetos: objetiva, razão, produto. Indica o nível de dependências entre classes, que se relaciona diretamente com o conceito de acoplamento. Neste caso, o aumento do valor da medida comparado com aquelas extraídas de outros objetos do sistema pode indicar um possível aumento da probabilidade de falhas do objeto.

c) Nível de Maturidade em Processo: subjetiva, pode ser ordinal ou intervalar (Likert), processo. Indica a percepção da maturidade da organização frente aos processos de software executados. O valor da medida está usualmente associado a níveis de atendimento ao processo (por exemplo, no MPS-BR varia de G até A) e a capacidade de atendimento da organização. Projetos complexos exigem níveis de maturidade mais altos (no caso do MPS-BR o nível A é o máximo que se pode atingir)

d) Facilidade de Uso do Software: subjetiva, ordinal, produto. Indica a percepção da facilidade de uso de um produto de software pelo usuário. Baixos valores da medida podem indicar um produto com pouca qualidade (usabilidade) e de pouca atratividade para seus usuários.

e) Cobertura dos Testes: objetiva, razão, produto (ou processo de testes). Indica a eventual cobertura (elementos cobertos pelo teste) que um procedimento de testes (ou conjunto de casos de teste) oferece em relação a avaliação de um produto de software (ou a execução do processo de testes). Baixos valores da medida indicam que os resultados dos testes apresentam baixa confiança (ou seja, quanto menor a cobertura, menor a chance de revelar falhas)

- 4) Explique qual é a finalidade de um diagrama de casos de uso e mostre os principais componentes deste diagrama através de um exemplo. (Valor 2,0 pontos)

Um diagrama de casos de uso apresenta os requisitos de um sistema sob a ótica dos agentes envolvidos com estes casos de uso. Os requisitos são descritos como casos de uso e representados no diagrama por elipses. Os agentes são descritos como atores e representados no sistema como bonecos-palito. Atores são associados aos casos de uso nos quais participam recebendo informações, apresentando informações ou ambos. Casos de uso são ligados a outros casos de uso quando os utilizam ou estendem.

- 5) Marque Verdadeiro (V) ou Falso (F) para as afirmações abaixo (Até 3 pontos – 0,2 por marcação correta. **Atenção: cada 2 erros/2 brancos/1 erro + 1 branco eliminam 1 acerto!**):

1. Inspeções de Software encontram defeitos alterando a produtividade, qualidade e estabilidade dos projetos de software.
2. O melhor testador para o código não é o próprio programador.
3. Os resultados dos testes provam que o software está completamente livre de defeitos.
4. Ferramentas como analisadores de código, comparadores de arquivos, repositórios de controle de configuração dificultam a manutenção do software.

5. O esforço de manutenção depende do tipo de sistema, seu tempo de vida, e da qualidade do projeto (*design*).
6. O perfil de experiência do inspetor não afeta intensamente a eficiência da inspeção com técnicas de leitura.
7. É irrelevante considerar as necessidades de diferentes *stakeholders* no planejamento e elaboração de materiais de treinamento e documentação.
8. Remover Defeitos é tão bom quanto Prevenir Defeitos.
9. O Critério de Cobertura dos Testes não influencia na quantidade de casos de teste.
10. O fenômeno de rejuvenescimento do software representa o aumento da qualidade de um sistema existente por meio de atividades como pagamento da dívida técnica, refatoração, e recompilação.
11. Inspeções de Software podem ser aplicadas a diferentes artefatos do ciclo de vida. Por isso, ao usar inspeções elimina-se a necessidade de testes.
12. O esforço despendido em manutenção perfectiva é proporcionalmente equivalente ao esforço de manutenções preventiva, adaptativa e corretiva juntas.
13. O manual do usuário explica as configurações de hardware e software, os métodos para permitir e negar o acesso de um usuário, os procedimentos para adicionar ou excluir periféricos de um sistema e as técnicas para duplicar ou realizar o backup de arquivos e documentos.
14. As técnicas de inspeção *checklist* sempre garantem a cobertura do documento como um todo.
15. O conhecimento adquirido no treinamento pode ser facilmente esquecido para aquelas funções que não são executadas regularmente.

RESPOSTA: 1-V; 2-V; 3-F; 4-F; 5-V; 6-V; 7-F; 8-V; 9-F; 10-F; 11-F; 12-F; 13-F; 14-F; 15-V