



Fundação CECIERJ – Vice-Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP3 2º semestre de 2018.**

**Nome –**

**Assinatura –**

---

Observações:

1. Prova sem consulta, sem uso de máquina de calcular. E sem celular!
  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. Dos cinco problemas listados abaixo, indique quais são observados com frequência quando um modelo de ciclo de vida em cascata é usado no desenvolvimento de um sistema. Escreva apenas as letras correspondentes aos problemas selecionados na folha de resposta. (Valor: 2.0 pontos).
  - a. Dificuldade de manter a sequência de tarefas proposta pelo modelo;
  - b. Incapacidade de concluir a etapa de análise de requisitos, devido a mudanças nas demandas dos clientes, travando as demais atividades do projeto;
  - c. Alto custo de desenvolvimento, pois os produtos das etapas de desenvolvimento são descartados e cada nova versão do software é construída desde o início;
  - d. A primeira versão utilizável do software só estará disponível após o término de todas as fases do projeto;
  - e. Componentes utilizados para acelerar as atividades de projeto e codificação podem apresentar problemas, gerando ruído nas fases seguintes do projeto.

A, B, D.

2. Diagramas de Casos de Uso representam informações importantes que podem ajudar nas decisões sobre funcionalidades, arquitetura, interfaces e outras características do sistema. Desta forma, descreva os elementos componentes de um diagrama de caso de uso. Desenhe um exemplo de diagrama contendo estes elementos e indique como cada um se relaciona com as características do sistema? (Valor: 2,0 pontos).

Um diagrama de casos de uso é composto por atores e casos de uso.

Atores representam elementos que possam interagir com o sistema. Eles não são parte do sistema, mas representam os elementos externos que interagem com o sistema. Atores podem ser: pessoas, outros sistemas ou equipamentos ligados ao sistema. Eles se relacionam entre si através de herança.

Casos de uso são um conjunto de ações, incluindo variantes de ações, que o sistema pode realizar e que oferecem algum benefício observável para o usuário do sistema. Eles se relacionam entre si através de relações de <<include>> e <<exclude>>.

Casos de uso se relacionam com os atores por relacionamentos de participação, que indicam que determinado ator participa (oferece informações, recebe informações ou ambos) do caso de uso.

3. Explique a diferença entre coesão e acoplamento entre classes quando analisamos a estrutura de um projeto de software. (Valor: 2,0 pontos).

Uma classe é coesa quando seus elementos internos (atributos e métodos) colaboram para atingir os objetivos da classe, ou seja, o seu papel no sistema. Sendo assim, a coesão mede a força interna de uma classe. O acoplamento mede o quanto uma classe depende de outras classes do sistema para cumprir a sua função, medindo o quanto a classe é conectada com o restante do sistema.

4. Relacione cada elemento da coluna da esquerda com um e somente um elemento da coluna da direita. (Valor 1,0 ponto).

(a) diagrama de transição de estados	(1) Explicita os processos de negócio do cliente.
(b) diagrama de sequencia	(2) Explicita as possibilidades de interação entre os usuários e o sistema.
(c) diagrama de classes	(3) Detalha o comportamento de um objeto no decorrer da sua vida.
(d) diagrama de casos de uso	(4) Detalha uma determinada possibilidade de interação entre o usuário e o sistema.
(e) descrição de casos de uso	(5) Explicita a estrutura estática interna do sistema.
(f) diagrama de atividades	(6) Detalha a interação entre diferentes objetos do sistema para atender a uma funcionalidade específica.

R: A-3; B-6; C-5; D-2; E-4; F-1

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

- a. Inspeções de Software encontram defeitos, porém alteram a produtividade, qualidade e estabilidade dos projetos de software.

- b. O próprio programador pode e é o mais adequado para testar seu próprio código.
- c. Os resultados dos testes não provam que o software está completamente livre de defeitos.
- d. Ferramentas como analisadores de código, comparadores de arquivos, repositórios de controle de configuração não contribuem muito no estágio de manutenção do software.
- e. Uma vez que a primeira lei de evolução afirma que o software vai mudar continuamente, o esforço de manutenção independe do tipo de sistema, seu tempo de vida, e da qualidade do projeto (*design*).
- f. O perfil de experiência do inspetor não afeta diretamente a eficiência da inspeção *ad-hoc*.
- g. É irrelevante considerar as necessidades de diferentes *stakeholders* no planejamento e elaboração de materiais de treinamento e documentação.
- h. Remover defeitos é melhor que prevenir defeitos.
- i. O Critério de Cobertura dos Testes não influencia no esforço do teste.
- j. O fenômeno de rejuvenescimento do software representa o aumento da qualidade de um sistema existente por meio de atividades como reestruturação, engenharia reversa e reengenharia.
- k. Inspeções de Software podem ser aplicadas a diferentes artefatos do ciclo de vida. Por isso, ao usar inspeções não é necessário aplicar testes.
- l. O esforço despendido em manutenção perfectiva é desproporcionalmente equivalente ao esforço de manutenções preventiva, adaptativa e corretiva juntas.
- m. O manual de operação 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.
- n. As técnicas de inspeção *ad-hoc*, apenas quando aplicadas por inspetores inexperientes, não garantem a cobertura do documento como um todo.
- o. O conhecimento adquirido no treinamento pode ser facilmente esquecido para aquelas funções que não são executadas regularmente.

R: A-V; B-F; C-V; D-F; E-F; F-F; G-F; H-F; I-F; J-V; K-F; L-F; M-V; N-F; O-V

**Boa Prova!**