



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP1 2º semestre de 2011.**

- 1) Explique a diferença entre técnicas *top-down*, *bottom-up* e *middle-out* na Engenharia de Software. (valor 2.0 pontos)

Top-down: partindo de elementos mais complexos e descendo a níveis cada vez mais detalhados. Bottom-up: partindo de componentes menores, estes são agrupados para formar os principais componentes do sistema (de mais alto nível). Middle-out: partindo de componentes intermediários, os elementos mais complexos são compostos e os mais simples são identificados

- 2) Explique a afirmativa “Software em operação é difícil de manter e a manutenção pode se tornar mais cara do que sua reconstrução”. (valor 2.0 pontos)

Erros encontrados durante a manutenção do software deverão ser corrigidos o mais rapidamente possível, de modo que a sua operação não sofra descontinuidades. As correções realizadas às pressas podem ser resolvidas de forma não-ideal, prejudicando a organização interna e a legibilidade do software e tornando-o, assim, mais difícil de manter. A medida que estes problemas se acumulam, o custo de correção pode atingir patamares onde é mais simples construir outro sistema do que manter o original.

- 3) Dos cinco problemas abaixo, indique quais são observados com frequência quando um modelo de ciclo de vida em cascata é usado no desenvolvimento de um sistema. Escreva os números dos problemas selecionados na folha de resposta. (valor 2.0 pontos)

- ✓ Dificuldade de manter a sequência de tarefas proposta pelo modelo;
- ✓ Incapacidade de concluir a etapa de análise de requisitos, devido a mudanças nas demandas dos clientes, travando as demais atividades do projeto;

- ✖ Alto custo de desenvolvimento, pois os produtos das etapas de desenvolvimento são descartados e cada nova versão do software é construída desde o início;
- ✓ A primeira versão utilizável do software só estará disponível após o término de todas as fases do projeto;
- ✖ Componentes utilizados para acelerar as atividades de projeto e codificação podem apresentar problemas, gerando ruído nas fases seguintes do projeto.

- 4) Explique o que é um caso de uso na modelagem dos requisitos de um software? Qual é a sua relação entre os casos de uso e os atores? (valor 2.0 pontos)

Um caso de uso é um diálogo entre o sistema e um ator, baseado em um cenário descritivo de como o ator interage com o sistema, identificando os eventos que podem ocorrer e as respostas esperadas a estes.

- 5) Quais das técnicas abaixo podem ser utilizadas para apoiar o levantamento de requisitos junto aos usuários durante a análise dos requisitos de um software? Escreva os números das técnicas selecionadas na folha de resposta. (valor 1.0 ponto)

- ✓ Entrevistas estruturadas
- ✖ Ciclo de vida em cascata
- ✖ Gerenciamento de projetos
- ✓ Role playing
- ✓ Reuniões de brainstorming
- ✖ Modularização
- ✓ Storyboarding e prototipação

- 6) Explique o que é coesão na disciplina de projeto de software. (valor 1.0 ponto)

Um módulo com alta coesão realiza uma única tarefa, necessitando de pouca interação com tarefas realizadas em outros módulos. Em termos práticos, um módulo altamente coeso realiza uma única tarefa, sem precisar executar rotinas de outros módulos.