



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP1 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) É comum que programadores pensem que os únicos artefatos que interessam no desenvolvimento de um software são os módulos de código-fonte. Explique porque esta afirmativa está incorreta, pelo menos em projetos de larga escala. Dica: outra maneira de responder a pergunta é explicar porque os outros artefatos (exceto o código fonte) são importantes. (valor: 2,0 pontos; máximo: 10 linhas).

Os módulos de código-fonte são fundamentais para o desenvolvimento de software: em muitos casos, eles são o objetivo final de todo um processo de desenvolvimento. No entanto, em projetos de larga escala, precisamos tomar conhecimento do que efetivamente precisa ser desenvolvido e registrar este conhecimento de forma que nos aproximemos, passo a passo, do nível de abstração do código-fonte. Precisamos registrar o conhecimento sobre o problema (modelos de análise), sobre a solução computacional proposta (modelos de projeto), até que finalmente possamos criar o código-fonte. Os artefatos intermediários ajudam no entendimento do domínio onde o software será desenvolvido, ajudando os desenvolvedores a se comunicar com os usuários, clientes e entre si.

- 2) Explique o que você entende por um processo de software. Quais são os principais elementos envolvidos quando descrevemos um processo de software? (valor: 2,0 pontos; máximo: 10 linhas).

Um processo de desenvolvimento de software é um conjunto de atividades, métodos, práticas e tecnologias utilizadas para o desenvolvimento e manutenção de software e produtos relacionados. Ele envolve a equipe de desenvolvimento (pessoas), as

ferramentas utilizadas pela equipe (hardware e software), os métodos de construção de software aplicados pela equipe e os subprodutos construídos por ela.

- 3) Explique as diferenças entre os tipos de atividade de manutenção: corretiva, adaptativa, preventiva e evolutiva. (valor: 1,0 ponto; máximo: 10 linhas)

A manutenção é um processo de correção de erros que ficaram latentes no software após do processo de desenvolvimento. A manutenção pode ser evolutiva (quando as modificações no software dizem respeito à introdução ou modificação das funcionalidades do software), corretiva (quando as modificações têm como objetivo corrigir erros que passaram pela fase de testes), adaptativa (que diz respeito à alteração do software para um novo hardware ou plataforma) ou preventiva (que trata da modificação do software para facilitar futuras atividades de manutenção).

- 4) Cite dois artefatos de software que são construídos durante a atividade de análise de requisitos. Explique o papel e a importância de cada um destes artefatos. (valor: 2,0 pontos; máximo: 10 linhas)

Na análise estruturada, são construídos diagramas de fluxos de dados (que mostram o ciclo de processamento das informações que entram e saem do sistema), o dicionário de dados (que descreve formalmente os termos utilizados no domínio), as mini-especificações de processos (que descrevem, de forma estruturada, uma unidade de processamento de um diagrama de fluxo de dados) e o diagrama entidade relacionamento (que apresenta os relacionamentos entre os dados que são manipulados pela aplicação).

Na análise orientada a objetos, são construídos os diagramas de casos de uso, que representam os diálogos entre os atores que utilizam o sistema e o próprio sistema, descrevendo assim as funcionalidades que o sistema deve oferecer a seus usuários.

Qualquer combinação de dois dos cinco elementos acima responde corretamente a pergunta.

- 5) Porque modelos são úteis no desenvolvimento de sistemas de software? Cite casos ou situações em que um sistema de software poderia ser construído sem o uso de modelos além do próprio código-fonte. (valor: 1,0 ponto; máximo: 10 linhas)

Modelos são úteis para o entendimento de problemas, a difusão do conhecimento entre os componentes da equipe de projeto e testar hipóteses antes de realizá-las. Eles são criados devido à limitação dos seres humanos em lidar com complexidade. Eles auxiliam na comunicação, organização de idéias, visualização e entendimento do projeto. Projetos de larga escala, com equipes envolvendo diversas pessoas, raramente podem ser conduzidos com sucesso sem o uso de modelos de algum tipo.

- 6) Um dos grandes problemas no levantamento de requisitos em projetos de desenvolvimento de software é a identificação de requisitos irrelevantes ou requisitos falsos. Cite dois potenciais problemas causados por este tipo de requisito em um projeto de software. (valor: 2,0 pontos; máximo: 10 linhas)

A identificação de requisitos irrelevantes faz com que a equipe de desenvolvimento invista esforço no desenvolvimento de funcionalidades que não atenderão a nenhum usuário. Desta forma, se estes requisitos fossem desconsiderados, o projeto poderia ser desenvolvido com menor custo e em menor tempo.

Requisitos irrelevantes ou incorretos geralmente também fazem com que o sistema seja mais complexo do que poderia ser. Isto pode influenciar negativamente na qualidade da interface com o usuário, no desempenho do sistema e, em última análise, na sua eficiência. Desta forma, requisitos deste tipo podem afetar de forma negativa a satisfação do usuário com o sistema.