



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AD1 1º semestre de 2010.

- 1) Cite uma razão pela qual foi criada a área de conhecimento da Engenharia de Software. Explique a relação entre a Engenharia de Software e a Engenharia em geral. (valor: 2,0 pontos; máximo: 10 linhas).

A Engenharia de Software surgiu com o intuito de identificar e analisar as causas dos problemas envolvidos com o desenvolvimento de software; propor soluções economicamente viáveis para a resolução destes problemas; e organizar o conhecimento sobre técnicas disponíveis para o desenvolvimento de software. Qualquer um dos itens acima responde a primeira parte da pergunta.

O principal problema da Engenharia de Software era a falta de conhecimento científico para ser aplicado. Por sua vez, as engenharias buscam a resolução de problemas práticos através de soluções que sejam economicamente viáveis, sendo esta busca motivada pela limitação de recursos, como tempo, dinheiro e pessoal capacitado para resolver os problemas. A estratégia da engenharia é aplicar o conhecimento científico sobre um determinado domínio tecnológico. Esta estratégia exige o desenvolvimento contínuo de tecnologias e conhecimento sobre as diversas áreas. Assim, a engenharia se especializa em diversos ramos de conhecimento (como elétrica, civil, software, ...).

- 2) Explique a diferença entre sistemas de informação e sistemas de engenharia, do ponto de vista dos diferentes objetivos e tarefas realizadas por estes sistemas. (valor: 2,0 pontos; máximo: 10 linhas)

Sistemas de informação são sistemas de apoio a processos de negócio e que consistem na manutenção e processamento de grandes bases de informações usando algoritmos geralmente simples. Sistemas de engenharia também apóiam atividades de negócio, não manipulando grandes bases de dados, mas processando informações através de algoritmos complexos. Assim, as principais diferenças entre os dois tipos de sistemas são a complexidade dos algoritmos e o volume de informações tratadas.

- 3) Cite pelo menos duas razões porque construímos modelos na Engenharia de Software. Cite um modelo utilizado em algum paradigma de desenvolvimento e explique qual é a sua função. (valor: 2,0 pontos; máximo: 10 linhas)

Modelos são úteis para o entendimento de problemas complexos, para a difusão do conhecimento entre os componentes da equipe de projeto e para testar hipóteses antes de investir recursos para colocá-las em prática. Qualquer destes itens serve como resposta para a primeira parte da pergunta. A segunda parte é respondida citando qualquer modelo referenciado a partir do slide 11 no conjunto de slides que trata de paradigmas de desenvolvimento de software. Exemplos de modelos incluem: diagramas de fluxos de dados, dicionários de dados, mini-especificações de processo, modelos entidade-relacionamento, modelos de casos de uso, modelos de classe, modelos de estado e diagramas de colaboração.

- 4) Dizemos que sistemas de informação são compostos de dados e processos. O que é cada um destes elementos e como eles interagem entre si? (valor: 2,0 pontos; máximo: 10 linhas)

Os dados são informações manipuladas por um sistema de informação, enquanto processos são seqüências de comandos que transformam estas informações.

- 5) Quais são os dois principais componentes de um diagrama de casos de uso? Explique o papel de cada um destes componentes na modelagem dos casos de uso de uma aplicação. (valor: 2,0 pontos; máximo: 10 linhas)

Diagramas de casos de uso são compostos por atores e casos de uso. Atores representam qualquer elemento que possa interagir com o sistema. Eles não são parte do sistema, mas representam os elementos externos que interagem com ele. Atores podem ser pessoas, outros sistemas e equipamentos ligados ao sistema. Um caso de uso representa um conjunto de ações, incluindo variantes de ações, que o sistema pode realizar e que oferece algum benefício observável para o usuário. Ele captura um contrato entre um patrocinador e um sistema acerca do comportamento do sistema, descrevendo este comportamento em diversas condições, a medida que ele responde a uma requisição gerada por um de seus atores.