



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP1 1º semestre de 2010.**

- 1) O que é a prototipação e como pode ser utilizada para apoiar o desenvolvimento de software? (valor: 2,0 pontos; máximo: 10 linhas)

Prototipação é a construção de versões mais simples dos sistemas para verificar o entendimento dos requisitos, a qualidade da interface com o usuário e a capacidade de um projeto resolver o problema. Eles permitem o desenvolvimento rápido de uma versão de baixa qualidade do sistema. Esta versão, embora não possa ser colocada em produção para os usuários (devido a sua baixa qualidade), pode ser apresentada aos usuários para validar o entendimento dos requisitos pelos integrantes da equipe de desenvolvimento.

- 2) Explique em que consiste a atividade de implantação. Cite 3 tarefas que comumente fazem parte do processo de implantação. (valor: 2,0 pontos; máximo: 10 linhas)

A implantação implica em colocar o software previamente desenvolvido em operação no ambiente do usuário. Normalmente ela inclui tarefas para a preparação de dados para serem utilizados pelo software, conversão de dados que estejam em formatos ou mecanismos de armazenamento diferentes daqueles usados pelo sistema, integração do novo sistema com outros sistemas, treinamento de operadores, construção e disponibilização de manuais e suporte à operação (help desk).

- 3) Explique os modelos de ciclo de vida incremental evolutivo, incremental descartável e incremental operacional. Quais as principais diferenças entre estes três modelos de ciclo de vida? (valor: 2,0 pontos; máximo: 15 linhas)

Incremental evolutivo: diversas execuções do modelo clássico de ciclo de vida em cascata, onde ao fim de cada execução é gerado um produto executável. Assim, artefatos produzidos em cada etapa de desenvolvimento são aproveitados em cada nova passagem pela etapa.

Incremental descartável: diversas execuções do modelo clássico de ciclo de vida em cascata, onde ao fim de cada execução todos os artefatos desenvolvidos são descartados. Assim, o produto é construído do início a cada nova iteração.

Desenvolvimento incremental operacional: os requisitos do sistema são identificados e refinados através de protótipos e o produto final é construído paralelamente a construção dos protótipos (por uma equipe separada).

- 4) Qual é a diferença entre os requisitos funcionais e os requisitos não-funcionais? Cite um exemplo hipotético de cada um destes requisitos. (valor: 2,0 pontos; máximo: 10 linhas)

Requisitos funcionais são descrições das funções que o sistema deve prover para o usuário, ligados ao domínio do problema. São declarações do que o sistema recebe como entrada, gera como saída, de como ele deve reagir a entradas específicas e como deve se comportar ao longo do tempo. Requisitos não funcionais são restrições impostas sobre as funções oferecidas pelo sistema, ligadas ao domínio da solução.

- 5) Escolha um princípio de projeto de software e explique seu significado. Descreva um exemplo de como um projeto deve aplicar este princípio. (valor: 2,0 pontos; máximo: 10 linhas)

São quatro os princípios gerais de projeto de software: separação de objetivos, modularização, antecipação de mudanças e generalização. Abaixo, descrevemos o significado dos quatro princípios. Qualquer um deles representa uma resposta válida, mas o exemplo vai depender do princípio escolhido.

Princípio da separação de objetivos: Permite a análise de diferentes aspectos de um problema, de modo a que os analistas possam se concentrar em um aspecto de cada vez. É um princípio fundamental para o entendimento de sistemas de software complexos, pois muitos problemas ocorrem quando diversos objetivos são conflitantes ou se influenciam mutuamente.

Princípio da modularização: a modularização auxilia na separação dos objetivos de um sistema por componentes, onde cada componente (ou módulo) pode ser utilizado por outros componentes sem que os últimos conheçam os detalhes de sua implementação.

Princípio da antecipação de mudanças: sugere que o projeto de um sistema deve ser realizado de forma que o sistema possa acomodar mudanças. A antecipação de mudanças reforça o princípio da modularização, fazendo com que as mudanças afetem um mínimo de módulos.

Princípio da generalização: um conjunto de requisitos descreve um problema, mas é comum que exista um problema mais geral do que o simplesmente descrito pelos requisitos. O princípio sugere que o projeto do sistema deve focalizar o problema mais geral, pois frequentemente este pode ser mais fácil de resolver, pode antecipar alterações nos requisitos e pode ser reutilizado com mais facilidade.