



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP1 2º semestre de 2013.

- 1) Justifique a seguinte afirmação: “o grau de progresso no desenvolvimento de um produto de software é difícil de medir”. (valor 2.0 pontos)

O material componente de um projeto de software (seus artefatos e código-fonte) é maleável. Esta maleabilidade, embora traga uma aparente facilidade de alteração ao software, esconde potenciais problemas de consistência entre as partes do produto. Estas inconsistências podem aparecer mais tarde sobre a forma de defeitos, o que torna o processo de desenvolvimento do software menos previsível. Além disso, partes desenvolvidas e aparentemente concluídas de um produto de software podem vir a apresentar problemas, dificultando assim uma medição consistente do progresso de um processo de desenvolvimento.

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

Técnicas *top-down* partem de elementos mais complexos e descem a níveis cada vez mais detalhados (refinamento a partir dos componentes mais complexos). Técnicas *bottom-up* partem de componentes menores, que são agrupados para formar os principais componentes do sistema (agrupamento dos componentes de mais alto nível a partir dos componentes menores). Por fim, técnicas *middle-out* partem de componentes intermediários, compondo os elementos mais complexos e refinando os intermediários para identificar os mais simples.

- 3) A IEEE define a Engenharia de Software como “a aplicação de uma abordagem sistemática, disciplinada e quantitativa para o desenvolvimento, operação e manutenção de software, ou seja, a aplicação da engenharia ao software”. Esta definição chama a atenção para a relação entre as Engenharias em geral e o desenvolvimento de produtos de software. Neste sentido, explique quais são as etapas da Engenharia e como elas se relacionam com as etapas de um ciclo clássico de desenvolvimento de software (como o cascata, por exemplo). (valor 2.0 pontos)

As Engenharias normalmente desenvolvem seus produtos em três grandes etapas: análise, síntese e correções. A análise é realizada quando um novo problema deve ser resolvido, devendo este ser dividido em partes menores e mais simples, até que estas partes possam ser resolvidas por técnicas conhecidas. A análise da Engenharia é equivalente à análise de requisitos no desenvolvimento de software. A síntese consiste em, tendo dividido o problema em pequenas partes e resolvido estas partes isoladamente, unir as soluções de cada parte em uma estrutura maior, que atenda a todo o problema. No contexto da Engenharia de Software, o projeto de software e a codificação são equivalentes à síntese. Por fim, as correções consistem na resolução de problemas decorrentes de tradução durante a síntese (verificação) ou de elicitação durante a análise (validação). Os testes são a atividade da Engenharia de Software equivalente às correções.

- 4) Explique o que é um requisito funcional de um software? Cite uma forma como este tipo de requisito pode ser modelado e dê um exemplo. (valor 2.0 pontos)

Requisitos funcionais são descrições das funções que o sistema deve prover para o usuário. São declarações ligadas ao domínio do problema descrevendo o que o sistema recebe como entrada, gera como saída, como ele deve reagir a entradas específicas e como deve se comportar ao longo do tempo.

- 5) Quais das técnicas abaixo não podem ser utilizadas para apoiar a identificação 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)

1. Entrevistas estruturadas
2. Ciclo de vida em cascata
3. Gerenciamento de projetos
4. Role *playing*
5. Reuniões de *brainstorming*
6. Modularização
7. *Storyboarding* e prototipação

Resp: 2, 3, 6.

- 6) Qual é a relação entre os conceitos de *fan-out* e acoplamento na disciplina de projeto de software? (valor 1.0 ponto)

O *fan-out* de um módulo indica o número de módulos que são utilizados por ele, ou seja, o número de módulos com quem ele tem relações de uso. Acoplamento é uma

medida de interconexão entre módulos – módulos com alto acoplamento dependem de muitos outros módulos para cumprir seus objetivos. Sendo assim, módulos com alto acoplamento possuem alto *fan-out*.