



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP31º semestre de 2015.

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) Cite duas limitações do modelo de ciclo de vida de desenvolvimento de software em cascata. Explique como os ciclos de vida chamados de incrementais tentam resolver estas limitações (valor: 2,0 pontos)

As duas principais limitações do modelo em cascata são: (a) a dificuldade de concluir a etapa de análise de requisitos, devido a modificações nos requisitos do software que ser congelados ao final da análise; e (b) a primeira versão do software só estará disponível após o término das fases de análise, projeto, codificação e testes, aumentando o tempo de latência entre o início do projeto e a primeira versão.

Os ciclos incrementais permitem que as atividades do ciclo de vida sejam executadas em diversas rodadas (incrementos), cada qual abordando parte do escopo do projeto. Sendo assim, haverá novas atividades de análise de requisitos para absorver as mudanças identificadas ao longo do projeto e a primeira versão do software não precisará esperar que todos os requisitos tenham sido desenvolvidos.

- 2) Quais das seguintes perguntas podem ser utilizadas para identificar atores para os casos de uso do sistema? Responda listando os números de todas as alternativas que considerar corretas (valor: 2,0 pontos).
 1. Quem está interessado em uma determinada operação?
 2. Quais são as tabelas do banco de dados do sistema?
 3. Quem será beneficiado pelo sistema?

4. Quem fornece informação para o sistema?
5. O sistema oferece interface gráfica com o usuário?
6. A que horas o sistema deve ser executado?
7. Quem fornece suporte ou manutenção para o sistema?
8. O sistema interage com algum sistema em operação?

1, 3, 4, 7 e 8.

- 3) Considerando o requisito descrito abaixo, defina os casos de teste utilizando **análise de valor limite** que permita testar no software a funcionalidade relacionada ao desconto a ser efetuado por dependente (valor 2,0 pontos):

"... o cálculo do desconto por dependente é feito da seguinte forma: as entradas são o sexo (masculino, feminino) e a idade do dependente que deve estar restrita ao intervalo [0..24]. Para dependentes masculinos até 10 anos (inclusive) o desconto é de 25%. Entre 10 e 15 (inclusive) o desconto é de 15%. Dos 15 aos 18 (inclusive) o desconto é de 10% e dos 18 aos 24 de 5%. Para dependentes femininos até 18 anos (inclusive) o desconto é de 25%. Dos 18 aos 24 é de 15%"

Sexo: Validos: {M,F} Inválidos: qualquer valor diferente de M,F

Casos de Teste para Sexo = {M}

Idade: [0..10]: Valores: 0,10 - resultado: 25%, Valor: 11 cai na outra faixa, 15%.

Idade: [11..15]: Valor: 15 - resultado 15%, Valor: 16 cai na outra faixa, 10%.

Idade: [16..18]: Valor: 18 - resultado 10%, Valor: 19 cai na outra faixa, 5%.

Idade: [19..24]: Valor: 24 - resultado 5%, Valor: 25, resultado 0% (não tem desconto)

Casos de Teste para Sexo = {F}

Idade:[0..18]: Valores:0,18 - resultado:25%, Valor:19 cai na outra faixa, 15%.

Idade:[19..24]: Valor:24 - resultado 15%, Valor:25, resultado 0% (não tem desconto)

- 4) “Manutenção de Software só ocorre em software construído de forma *ad-hoc*”. Esta afirmação é verdadeira ou falsa? Justifique sua resposta, indicando se existem ou não diferentes tipos de manutenção e apresente 3 fatores que podem afetar o esforço de manutenção de software e 3 responsabilidades da equipe de manutenção. (valor: 2,0 pontos).

A afirmação é falsa. Todo software passa por manutenção, principalmente evolutiva. A manutenção pode ser também corretiva (que deve ser evitada com boas práticas de desenvolvimento), perfectiva, adaptativa ou preventiva. Diferentes fatores podem influenciar a manutenção, tais como: Tipo de aplicação, Novidade do sistema, Rotatividade e disponibilidade do pessoal de manutenção, Duração da vida útil do sistema, Dependência de um ambiente que se modifica, Característica de hardware, Qualidade do projeto, Qualidade do código, Qualidade da documentação, Qualidade dos testes.

São responsabilidades da equipe de manutenção: entender o sistema, localizar informação na documentação do sistema, manter a documentação do sistema atualizada, estender as funções existentes para acomodar novos requisitos ou modificações nos requisitos, acrescentar novas funções para o sistema, encontrar a fonte de falhas ou problemas no sistema, localizar e corrigir faltas, responder questões sobre a forma como o sistema funciona, reestruturar o projeto e codificação dos componentes, reescrever o projeto e código dos componentes, apagar os projetos e códigos de componentes que não são mais úteis, gerenciar trocas para o sistema

- 5) Na AD2 foi solicitada uma pesquisa sobre computação ubíqua. De acordo com o que você aprendeu, responda: o que é computação ubíqua? O que são sistemas ubíquos? Que características estes sistemas apresentam que os diferenciam de sistemas convencionais? Dê um exemplo de um sistema ubíquo e aponte que características de ubiquidade este sistema apresenta. (valor 2,0 pontos)

Em linhas gerais, computação ubíqua se faz presente no momento em que os serviços ou facilidades computacionais são disponibilizados às pessoas de forma que o computador não seja uma ferramenta visível ou imprescindível para acesso a esses serviços. Ou seja, esses serviços ou facilidades podem se materializar em qualquer momento ou lugar, de forma transparente, através do uso de dispositivos de uso comum no dia-a-dia.

Um sistema ubíquo possui um escopo mais bem definido e está fortemente relacionado com as diferentes dimensões que compõem a computação ubíqua. Isto por que ubiquidade é uma propriedade do sistema e como tal, pode ser atendida em sua plenitude ou parcialmente. Esta variação está relacionada com o fato de um sistema implementar ou não as funcionalidades que representam as dimensões da computação ubíqua.

As características diferenciais (dimensões) para este tipo de software se relacionam a onipresença dos serviços, invisibilidade, sensibilidade ao contexto, comportamento adaptável ou dinamismo de tarefas, captura de experiências, descoberta de serviços, composição de funcionalidades, interoperabilidade espontânea, heterogeneidade de dispositivos e tolerância a falhas.

Exemplos:

Software com onipresença de serviços: o usuário, ao se deslocar, continua acessando seus serviços nos mais variados ambientes em que se encontra.

Software com invisibilidade: ao passar em frente a um cartaz de um filme, o indivíduo recebe no celular maiores informações sobre o filme em questão como: trilha sonora, diretor, duração, elenco e o resumo

Software com sensibilidade ao contexto: um sistema para controle de temperatura de um frigorífico deve estar constantemente monitorando a temperatura para manter o ambiente no estado ideal para manutenção das carnes.