



Fundação CECIERJ – Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP3 2º semestre de 2017.

GABARITO

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) Considerando o requisito descrito abaixo, defina os casos de teste utilizando **particionamento por classes de equivalência** que permita testar o resultado do desconto efetuado por dependente (valor 2,0 pontos):

"... o cálculo do desconto de plano de saúde por dependente é feito da seguinte forma: a entrada é a idade e o sexo do dependente que devem estar restritos aos intervalos [0..18] e [M,F], respectivamente. Para dependentes masculinos até 5 anos (inclusive) o desconto é de 30%. Entre 5 e 10 (inclusive) o desconto é de 20%. Dos 11 aos 15 (inclusive) o desconto é de 15% e dos 16 aos 18 de 10%. Para dependentes femininos até 5 anos (inclusive) o desconto é de 50%. Entre 5 e 12 (inclusive) o desconto é de 30%. Dos 12 aos 16 (inclusive) o desconto é de 18% e dos 16 aos 18 de 10%. Acima de 18 anos não há desconto."

Utilizando particionamento por classes de equivalência, devemos gerar um caso de testes para cada classe válida e inválida do domínio de entrada. Desta forma, a combinação de fatores deve ser:

Masculino:

0..5, 30% -> classe válida [$\geq 0.. \leq 5$]; classe inválida >5 . Não faz sentido idade <0 .
5..10, 20% -> classe válida [$>5.. \leq 10$], classes inválidas ≤ 5 ; >10 .
11..15, 15% -> classe válida [$\geq 11.. \leq 15$], classes inválidas <11 ; >15 .
16..18, 10% -> classe válida [$\geq 16.. \leq 18$], classes inválidas <16 ; >18 .
 ≥ 18 , 0% -> classe válida [≥ 18], classe inválida <18 .

Feminino:

0..5, 50% -> classe válida [≥ 0 .. ≤ 5]; classe inválida >5 . Não faz sentido idade <0 .

5..12, 30% -> classe válida [>5 .. ≤ 12], classes inválidas ≤ 5 ; >12 .

12..16, 18% -> classe válida [>12 .. ≤ 16], classes inválidas ≤ 12 ; >16 .

16..18, 10% -> classe válida [>16 .. ≤ 18], classes inválidas ≤ 16 ; >18 .

≥ 18 , 0% -> classe válida [≥ 18], classe inválida <18 .

Note que as classes inválidas se sobrepõem as classes válidas. Neste caso temos os seguintes casos de teste (qualquer valor nas faixas é possível), apresentados como a tupla $CT = \{\text{sexo, idade, desconto esperado}\}$

$\{M, 2, 30\}; \{M, 6, 20\}; \{M, 13, 15\}; \{M, 17, 10\}; \{M, 18, 0\}$

$\{F, 2, 50\}; \{F, 6, 30\}; \{F, 12, 18\}; \{F, 17, 10\}; \{F, 18, 0\}$

- 2) Na AD2 foi solicitada uma pesquisa sobre métodos ágeis. De acordo com o que você aprendeu, responda: o que é um método ágil? Que características estes métodos apresentam que os diferenciam de métodos convencionais? Dê um exemplo de um método ágil e apresente suas características principais. (valor 2,0 pontos)

Conforme gabarito da AD2, Métodos ágeis, Desenvolvimento ágil ou simplesmente Agile, engloba um conjunto de metodologias utilizadas no desenvolvimento de software que propõem, essencialmente, menor foco no processo e maior foco em pessoas, interações entre pessoas, produto, colaboração com cliente e respostas a mudanças.

Agile engloba diversas metodologias (também chamadas de práticas) como, por exemplo, Adaptive Software Development (ASD), Agile Unified Process (AUP), Crystal Methods, Dynamic Systems Development Methodology (DSDM), eXtreme Programming (XP), Feature Driven Development (FDD), Kanban, Lean Software Development, Scrum, Scrumban.

Como características de XP podem ser citadas grande foco no código fonte, intensidade nos testes e simplicidade nas soluções. Dependendo do método citado as características se modificam.

- 3) Diferentes documentos devem ser criados ao longo do processo de desenvolvimento. O processo de desenvolvimento, por sua vez, pode assumir diferentes configurações dependendo do nível de maturidade da organização. Desta forma, indique 2 documentos que devem ser criados por uma organização de software nível C no MPS-BR, explicando seus objetivos e sugerindo quais são seus usuários em potencial. (valor: 2,0 pontos; máximo: 10 linhas)

Qualquer documento produzido nas áreas de processo previstas no nível C (veja o material do curso). Por exemplo:

gerencia de requisitos: especificação de requisitos descrevendo os requisitos do software usado pelos desenvolvedores, testadores e clientes.

gerencia de projeto: plano de projeto descrevendo os diferentes modelos e artefatos que deverão ser criados ao longo do projeto usado pelo gerente, desenvolvedores e testadores

medição: conjunto de medidas que devem ser coletadas e consideradas para observar o “comportamento” do software e do processo usado pelos desenvolvedores e gerentes

gerencia de configuração: plano de configuração descrevendo a estrutura organizacional dos diferentes artefatos, módulos e componentes. Usado pelos desenvolvedores, testadores e gerentes.

gerencia de aquisição: plano de aquisição relacionado ao processo de desenvolvimento. Utilizado pelos gerentes do projeto.

garantia da qualidade: plano de qualidade do software estabelecendo os critérios e comportamentos esperados para o software usado pelos desenvolvedores, testadores, gerentes e stakeholders

gestão de portfólio de projetos: documento descrevendo os diferentes projetos em andamento e suas características. Utilizado pelos gerentes de projeto.

dentre outros.

- 4) “Manutenção de Software sempre vai ocorrer, independente do processo de desenvolvimento utilizado para a construção do software”. 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; máximo: 15 linhas).

A afirmação é verdadeira.

Qualquer trabalho realizado para alterar o sistema depois que ele já se encontra em operação é considerado manutenção.

Podem ser:

Correções: Corrige um defeito – i.e. uma discrepância entre o comportamento requerido para um produto/aplicação e o comportamento observado

Melhorias: Implementam uma mudança para o sistema que modifica seu comportamento ou implementação. Melhorias podem ser: Troca de requisitos (Manutenção Perfectiva), adiciona um novo requisito ao sistema (Manutenção Adaptativa) e troca a implementação, mas não o requisito (Manutenção Preventiva)

O tipo de manutenção que normalmente tem recebido mais atenção das equipes é a perfectiva.

Fatores: 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

Responsabilidades da Equipe: 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) As decisões em relação ao uso de tecnologias em projetos de software devem ser sempre baseadas em evidência. A avaliação das tecnologias pelos Engenheiros de Software representa uma fonte importante de evidência. Neste sentido, diferentes estratégias podem ser usadas para realizar avaliações de tecnologias em Engenharia de Software. Quais são elas? Em que situação cada uma pode ser aplicada? (valor: 2,0 pontos; máximo: 10 linhas).

Análise de características

- Tipo mais simples de avaliação
- atribuir valor e classificar atributos de vários produtos
- Uso interessante para apoiar escolhas de métodos ou ferramentas, por exemplo

• Pesquisa de opinião

- Estudo em retrospectiva, documenta relações e resultados de certa situação. Pode ser usado para capturar o nível de satisfação dos desenvolvedores após a execução de alguma tarefa no processo.

• Estudos de caso

- Estudo não retrospectivo. Necessária a identificação dos fatores principais que podem afetar o resultado de uma atividade. Definição de hipóteses: orienta a medição e a análise dos resultados. Podem ser usados para observar a utilização de alguma técnica, por exemplo.

• Estudo Experimental (Experimento formal)

- Estudo não retrospectivo. Valores das variáveis são controlados. São utilizados vários métodos para reduzir tendências e eliminar fatores que se confundem. Podem ser usados para comparar tecnologias, por exemplo.

Boa Prova!