



Fundação CECIERJ – Vice-Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP3 2º semestre de 2016.**

**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 **grafo de causa efeito** 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."

SEXO	M				
IDADE	<=5	>5 e <=10	>10 e <=15	>15 e <=18	>18
DESCONTO	30%	20%	15%	10%	0%
SEXO	F				
IDADE	<=5	>5 e <=12	>12 e <=16	>16 e <=18	>18
DESCONTO	50%	30%	18%	10%	0%

SEXO	M					F				
IDADE	<=5	>5 E <=10	>10 E <=15	>15 E <=18	>18	<=5	>5 E <=12	>12 E <=16	>16 E <=18	>18
DESCONTO (%)	30	20	15	10	0	50	30	18	10	0

CT: {(M,5); (M,10); (M,15), (M,19); (F,5);(F,12);(F,16), (F,18); (F,19)}

- 2) Uma ferramenta CASE é utilizada para apoiar as atividades de desenvolvimento de software. Entretanto, sua escolha e uso dependem de algumas características do processo de desenvolvimento e do produto a ser construído. Indique 2 (duas) características associada ao processo de desenvolvimento e 2 (duas) associadas ao produto que podem afetar a escolha de uma ferramenta CASE. Explique o significado de cada uma destas características. Em complemento, cite um exemplo de ferramenta CASE e explique como, quando e para quê ela pode ser usada no processo de desenvolvimento de software. (valor: 2,0 pontos; máximo: 10 linhas)

Produto de software que pode ser utilizado para apoiar a execução de atividades inerentes ao processo de desenvolvimento de software. Conjunto de software cujo propósito é apoiar a criação, análise e manutenção de outro software.

Características que devem ser observadas: Escopo coberto pela ferramenta, Ambiente de Hardware, Ambiente Operacional, Tipo de sistema de software desenvolvido, Natureza do trabalho, Facilidade de Aumento de Escopo, Geração de código, Geração de Casos de Teste, Facilidades de personalização a metodologias, Documentação produzida, Interface com o Usuário, Apoio ao aprendizado, Segurança

Um exemplo de ferramenta CASE pode ser representado por um diagramador de modelos UML. As ferramentas StarUML, Enterprise Architect, BOUML, dentre outras são representantes deste grupo. Elas podem ser usadas nas fases de especificação de requisitos e projeto (ver exemplos adicionais no material do curso).

- 3) O que é um diagrama de estado? Quando devemos considerar sua utilização num projeto de software? Mostre como um diagrama de estados é representado em UML, indicando seus elementos constituintes e as possíveis relações entre eles. (valor: 2,0 pontos; máximo: 10 linhas).

Diagramas de estado modelam o comportamento de uma classe no tempo e são utilizados apenas em classes com estados relevantes, como classes que monitoram eventos ao longo de sua vida e serviços de monitoração com algoritmos complexos. Eles são construídos a partir de um estudo das operações realizadas sobre as classes ou pelas próprias classes. Eles são compostos de estados (representados como retângulos com bordas arredondadas) e transições (representadas como setas entre os estados). As transições mostram as possíveis trocas de estado em função de ações executadas sobre ou pela classe. Dois

estados especiais podem ser encontrados em um diagrama de estados: o estado inicial (representado por um pequeno círculo) e o estado final (representado por um “olho de boi”).

- 4) O que é manutenção de software? Que atividades são normalmente executadas pelas equipes de manutenção de software? Qual o tipo de manutenção todo software está sujeito a sofrer? Indique 3 (três) fatores que podem afetar o esforço de manutenção, explicando de que forma eles podem afetar as atividades de manutenção do software. (valor: 2,0 pontos; máximo: 15 linhas).

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) Por que documentar software é importante? Dê 2 (dois) exemplos de documentos que normalmente são entregues com o software, explicando seus objetivos e indicando quais são seus principais usuários. (valor: 2,0 pontos; máximo: 10 linhas).

Documentar o software é de fundamental importância para sua existência. Os documentos apoiam o entendimento e utilização do software, facilitam a manutenção e permitem explicitar as características do software que devem ser preservadas ao longo do ciclo de vida. Diferentes documentos devem ser criados ao longo do processo de desenvolvimento

Possíveis exemplos:

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.