



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP3 2º semestre de 2009.**

**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) Manutenção de software pode ser classificada em diferentes tipos, dependendo da ação que precisa ser realizada no software. Descreva os diferentes tipos de manutenção que o software pode receber. Dentre estes tipos, qual normalmente se refere a melhoria do software? Dê um exemplo. (valor: 1,0 ponto; máximo: 10 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)

Os tipos de manutenção que normalmente tratam melhoria do software são Perfectiva, Preventiva ou Adaptativa, dependendo de que melhoria se deseja tratar.

- 2) Indique 3 documentos que devem ser entregues junto com o software. Explique o objetivo, quem usa e um possível problema que pode ocorrer na falta de cada um destes 3 documentos que você indicou. (valor: 1,5 pontos; máximo: 10 linhas).

**Manual do Usuário**

- Usuários querem saber os detalhes de uso e das funções do sistema
- Guia de referência ou tutorial para usuários do sistema:

- Os objetivos e propósitos do sistema
  - As capacidades e funções do sistema
  - As características do sistema e suas vantagens, apresentando um cenário claro do que o sistema faz
- x A falta deste documento pode levar o usuário a não compreender/entender em como utilizar o software

#### Manual do Operador

- Apresenta material para operadores da mesma maneira que manuais d usuário
- Operadores querem conhecer detalhes de acesso e desempenho do sistema
- A falta deste documento pode levar a uma operação errada do software, provocando perdas de informação ou mal funcionamento de alguma característica do software

#### Guia Geral do Sistema

- O cliente representa a maior audiência para este manual
- Descreve o sistema sem detalhar cada função. Entretanto, ele contém informação suficiente que permite ao cliente decidir se o sistema é completo ou adequado para as demandas da organização
- Fornece referência cruzada para os outros documentos
- A falta deste documento pode levar ao cliente ter um entendimento inadequado do software, afetando a tomada de decisão frente as possibilidades de utilização do produto

3) O que é uma ferramenta CASE? Se você tiver que escolher uma ferramenta CASE para ser usada em um projeto, cite três características que precisam ser observadas para reduzir o risco de problema no desenvolvimento. Dê um exemplo de ferramenta CASE, indicando para quê ela pode ser utilizada no processo de desenvolvimento. (valor: 1,5 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

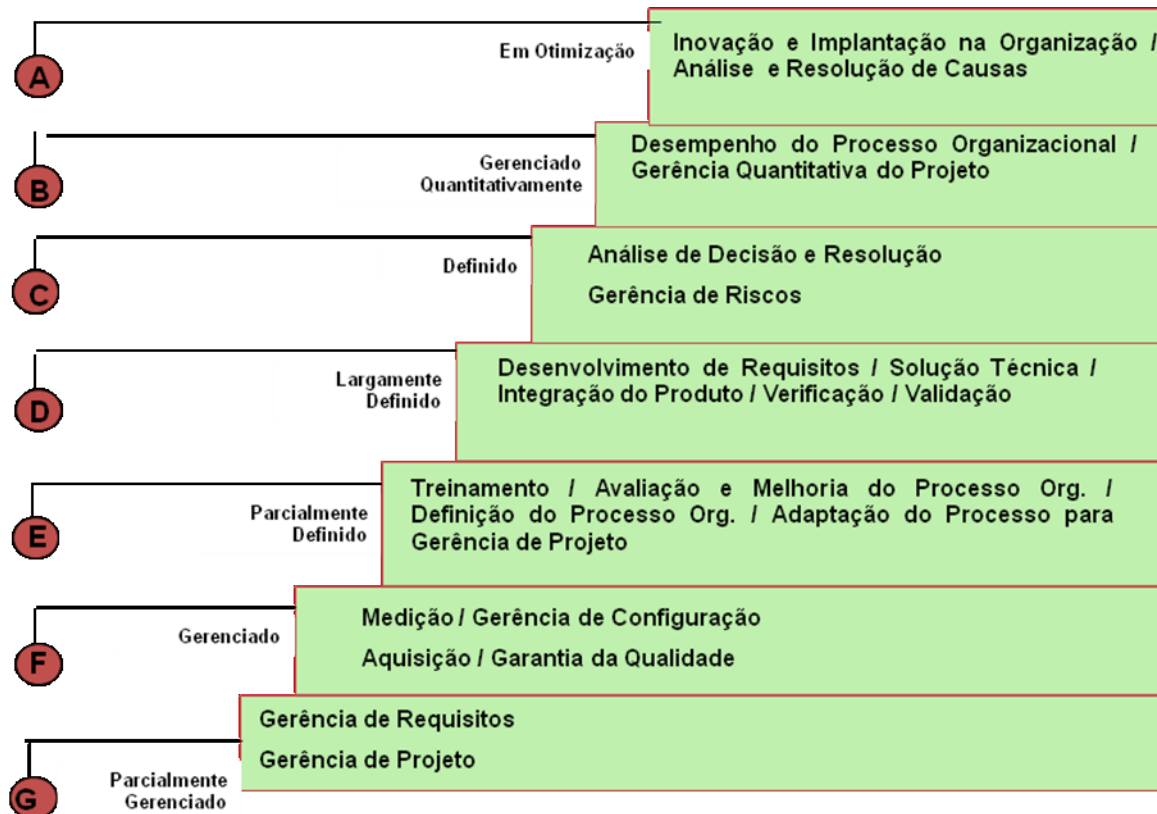
Um exemplo de ferramenta CASE pode ser representado por um diagramador de modelos UML. As ferramentas StarUML e Enterprise Architect, dentre outras são representantes deste grupo. Elas podem ser usadas nas fases de especificação de requisitos e projeto (ver exemplos adicionais no material do curso).

4) Explique as abordagens que podem ser utilizadas para realizar avaliações em engenharia de software. Comente, para cada uma, em que situação poderia ser aplicada. (valor: 1,5 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.
- **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.

5) Do ponto de vista do MPS BR (Modelo de Referência da Melhoria do Processo de Software Brasileiro), os níveis de maturidade variam de G até A. Escolha dois destes níveis, indicando seus nomes juntamente com 1 atividade importante de ser executada para cada nível escolhido. (valor: 2,0 pontos; máximo: 10 linhas).



6) Considerando o requisito descrito abaixo, defina os casos de teste utilizando **análise de valor limite** que permita testar o desconto a ser utilizado para realizar os testes relacionados ao desconto efetuado por dependente (valor 1,5 pontos):

"... o cálculo do desconto por dependente é feito da seguinte forma: a entrada é a idade do dependente que deve estar restrita ao intervalo [0..18]. Para dependentes até 6 anos (inclusive) o desconto é de 23%.

Entre 6 e 12 (inclusive) o desconto é de 18%. Dos 12 aos 16 (inclusive) o desconto é de 10% e dos 17 aos 18 de 5%. Caso o dependente seja do sexo feminino, o desconto é de 4% dos 17 aos 18 anos"

Idade 0..6: 23% - Casos de Teste (válidos): {0,6}

Idade >6..12: 18% - Casos de Teste (válidos) : {7,12}

Idade >12..16: 10% - Casos de Teste (válidos): {15,16}

Idade 17..18: 5%, Sexo:M - Casos de Teste (válidos): {(17,M),(18,M)}

Idade 17..18: 4%, Sexo:F - Casos de Teste (válidos): {(17,F),(18,F)}

Idade negativa: não faz sentido.

Idade > 18 anos: sem desconto.

7) Considerando o Diagrama de Classes abaixo, encontre os valores de NOC (numero de filhos), DIT (profundidade de herança) e CBO (Acoplamento entre objetos) para as classes identificadas na tabela. Indique qual a classe estaria mais propensa a falha e por quê? (valor: 1,0 ponto):

CLASSE	NOC	DIT	CBO
Item	0	0	4
Relacionamento	0	0	2
Usuário	0	0	2
Projeto	0	0	2

A principio a classe Item é a mais propensa a falha pois é a que apresenta maior acoplamento, ou seja, dependência de outras classes (CBO)

