



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Engenharia de Software
AP3 1º semestre de 2012. 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) Explique que diferenças existem entre atividades de manutenção de correção e melhoria, e indique 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).

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

As atividades executadas podem ser de:

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)

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

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

- 2) Dê 2 exemplos de documentos que devem ser entregues juntamente com o software, explicando seus objetivos e indicando quais são seus principais usuários. (valor: 1,0 ponto; máximo: 10 linhas)

A documentação do software tem sua importância relacionada a própria existência do produto. Na verdade, assim como o código fonte, ela compõe o conjunto de materiais/artefatos que devem ser entregues ao cliente. Existem diferentes tipos de documentos que devem ser construídos ao longo do ciclo de vida do produto. Cada um tem por objetivo descrever características distintas relacionadas ao produto. A entrega da documentação contribui para garantir a qualidade do produto sendo entregue. Alguns exemplos são:

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

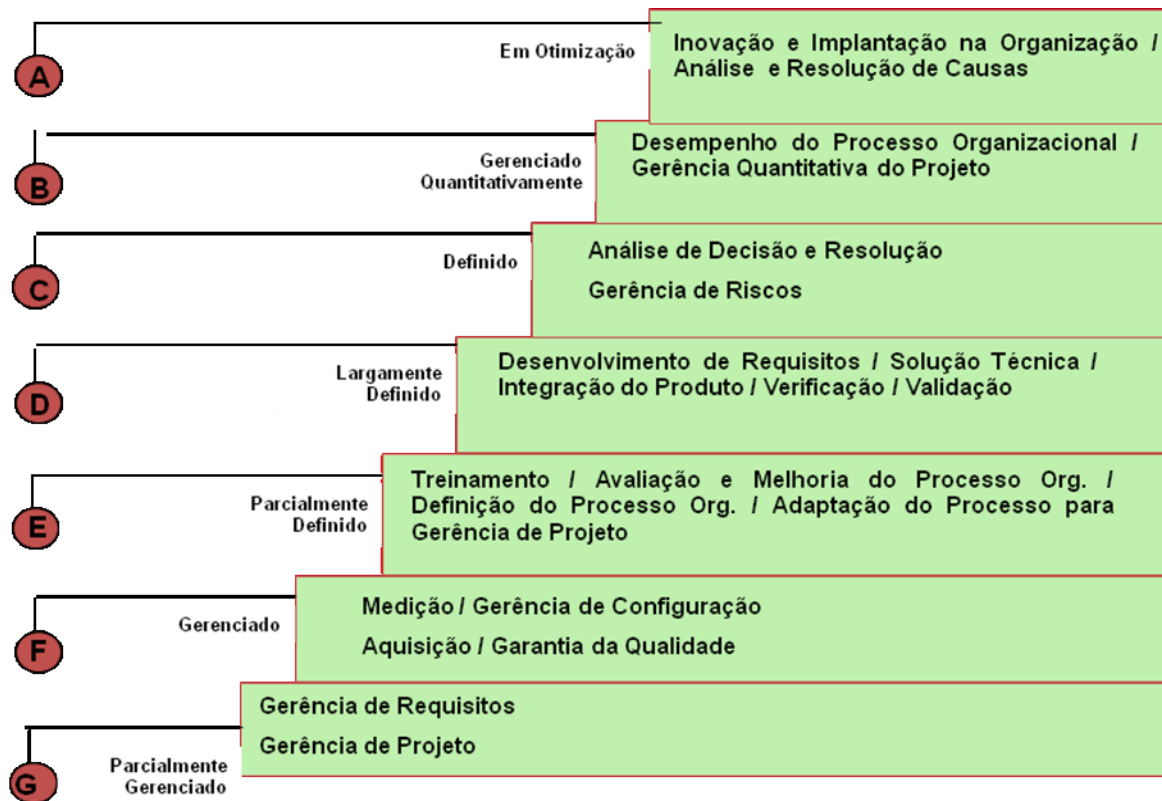
Manual do Operador

- Apresenta material para operadores da mesma maneira que manuais d usuário
- Operadores querem conhecer detalhes de acesso e desempenho do sistema

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

- 3) 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: 1,0 ponto; máximo: 10 linhas)



- 4) Considerando o requisito descrito abaixo, defina os casos de teste (valor de entrada; resultado esperado) utilizando **particionamento por classes de equivalência** que permita testar o desconto a ser 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 22%. Entre 5 e 12 (inclusive) o desconto é de 18%. Dos 12 aos 16 (inclusive) o desconto é de 10% e dos 16 aos 18 de 5%. 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%..."

	Válida	Válida	Inválidas	Inválidas
Sexo	--	M	!= {M,F}	--
Idade p/ desconto 22%	<=5	--	>5	<0
Idade p/ desconto 18%	>5, <=12	--	<=5	>12
Idade p/ desconto 10%	>12, <=16	--	<=12	>16
Idade p/ desconto 5%	>16, <=18	--	<=16	>18
Sexo	--	F	!= {M, F}	--
Idade p/ desconto 50%	<=5	--	>5	<0
Idade p/ desconto 30%	>5, <=12	--	<=5	>12
Idade p/ desconto 18%	>12, <=16	--	<=12	>16
Idade p/ desconto 10%	>16, <=18	--	<=16	>18

Casos de Teste: (Valor, Valor, Resultado)

CT_M_Validos: (M,3, 22), (M, 7, 18), (M, 13, 10), (M, 18,5)

CT_F_Validos: (F, 3, 50), (F,7,30), (F, 13, 18), (F, 18, 10)

CT_M_Invalidos: (M, -1, erro), (M, 19, Sem desconto)

CT_F_Invalidos: (F, -1, erro), (F, 19, Sem desconto)

CT_Invalidos: (A, 10, erro)

- 5) Uma ferramenta CASE é utilizada para apoiar as atividades de desenvolvimento do software. Entretanto, sua escolha e uso dependem de algumas características que devem ser sempre observadas. Indique 4 dessas características, explicando seu significado. Em complemento, cite um exemplo de ferramenta CASE explicando quando e como 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 BOUML, Enterprise Architect, StarUML, 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).

- 6) 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 e indique qual delas pode oferecer risco à qualidade do produto, explicando ao menos um problema que poderia ocorrer. Leve em consideração as estruturas hierárquicas para extração das medidas (valor: 2,0 pontos):

CLASSE	NOC	DIT	CBO	Propensa à Falha
Cria Mapa	0	2	3	
Menu Superior (subclasse)	0	2	4	Com esta classe se deve ter um pouco mais de atenção tendo em vista seu acoplamento mais alto, indicando alto nível de dependência e portanto, mais sujeita a provocar problemas ou sofrer influencias de outras classes
Principal	1	1	1	

