



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Engenharia de Software**

**AP3 2º semestre de 2010.**

**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) O que você entende pelo termo ferramenta CASE? Dê três características que precisamos observar quando tivermos que considerar o uso de ferramentas CASE para apoiar o desenvolvimento de software. Cite um exemplo de ferramenta CASE, onde pode ser utilizada no processo de desenvolvimento e indique sua categoria (que pode ser vertical ou horizontal). (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 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).

2) Defina manutenção de software. 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.

3) Considere os documentos: Manual do Usuário, Manual do Operador e Guia Geral do Sistema. Explique o objetivo de cada um destes documentos e defina qual seu principal usuário. (valor: 2,0 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

Glossário

Manual do Operador

Apresenta material para operadores da mesma maneira que manuais d usuário

Operadores querem conhecer detalhes de acesso e desempenho do sistema

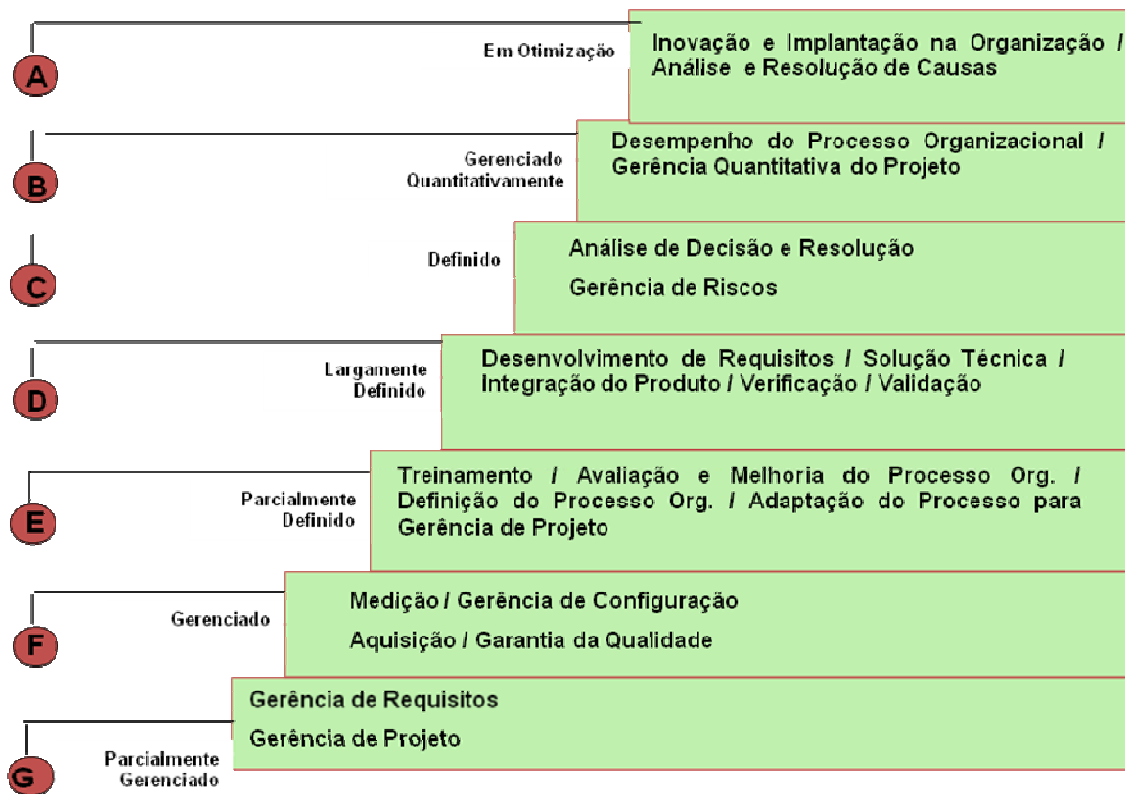
Guia Geral do Sistema

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

4) 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).



5) Qual a finalidade de se construir diagramas de estado durante a modelagem dinâmica com UML? Mostre os elementos que compõem o diagrama e suas relações através de um exemplo de diagrama de estados (valor: 1,5 pontos; máximo: 10 linhas).

Um diagrama de estados (ou diagrama de transição de estados) serve para descrever o que ocorre com um objeto (objetos de uma classe) ao longo do seu ciclo de vida tendo em vista os diferentes eventos que acontecem durante a utilização do sistema. Basicamente, um diagrama de estados é construído para aquelas classes/objetos que podem ter seu comportamento alterado de acordo com o estado em que se encontram e os eventos que acontecem. O diagrama, portanto, possui construtores sintáticos para indicar estados, transições (eventos que fazem com que um objeto passe de um estado a outro) e marcações de início e fim de ciclo de vida.

Um exemplo de diagrama de estados pode ser visto abaixo:

