



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AD2 2º semestre de 2007.

1. Explique o que é um módulo no que se refere ao projeto de software.

Um módulo é um componente bem definido de um sistema, que pode conter rotinas, dados e definições de tipos.

2. Dizemos que o projeto de software pode ser analisado segundo duas visões ortogonais. Na primeira, o projeto pode ser detalhado ou arquitetônico. Na segunda, o projeto pode tratar da lógica, da interface com o usuário ou do armazenamento. De acordo com a segunda visão, explique o que é tratado em cada uma das três divisões dos projetos de software.

O projeto de interface aborda as características da interface do sistema com o meio externo, através da qual o sistema recebe e envia dados para elementos externos a ele (como seus usuários, por exemplo). O projeto de dados trata da organização, armazenamento e recuperação das informações manipuladas pelo sistema em um meio persistente (como arquivos e tabelas em um banco de dados). Finalmente, o projeto lógico aborda as características de processamento das informações manipuladas pelo sistema, destacando a organização dos módulos do sistema e a comunicação realizada entre eles para a resolução de um problema.

3. Explique o que você entende pelos termos defeito, erro e falha. Explique a diferença entre eles, e cite pelo menos uma técnica de VV&T que pode ser usada para identificar cada um deles. Dê um exemplo de como defeitos podem ser classificados.

Defeito: Deficiência mecânica ou algorítmica que se ativada pode levar a uma falha. Defeitos podem ser classificados como de omissão, inconsistência, fato incorreto, ambigüidade e informação estranha.

Técnicas que podem ser utilizadas para identificar defeitos: Inspeção Ad-hoc, checklist ou Técnica de Leitura.

Erro: Item de informação ou estado de execução inconsistente.

Falha: Evento notável onde o sistema viola suas especificações. Técnica que pode ser utilizada para identificar erros: Testes de software

4. Indique as fases do processo de inspeção de software, apresente o objetivo de cada uma e que papéis são necessários para sua execução. Para a fase de detecção, indique as técnicas que podem ser usadas para identificar defeitos.

Fase: Planejamento

Responsável: Moderador.

Tarefas/objetivos: Definir contexto da inspeção (descrição da inspeção, como a preparação individual deverá ocorrer, documento a ser inspecionado, autor do documento, entre outros); Selecionar inspetores (recomenda-se utilizar entre 3 e 5 inspetores em uma inspeção); Distribuir material.

Fase: Preparação Individual (detecção de defeitos)

Responsável: Inspetor.

Tarefas/objetivos: Estudar os artefatos; Fazer anotações sobre os artefatos.

Técnicas que podem ser usadas: ad-hoc, checklist ou Técnica de Leitura

Fase: Reunião de Inspeção

Envolvidos: Moderador, Inspetores e Autor.

Tarefas/objetivos: Leitura do documento, com a equipe discutindo possíveis defeitos (Duração recomendada 2hrs); Produzir uma lista de defeitos; Em casos de discordância a decisão sobre registrar um defeito ou não (falso positivo) é do moderador.

Fase: Retrabalho

Responsável: Autor.

Tarefas/objetivos: Corrigir os defeitos encontrados.

Fase: Continuação

Responsável: Moderador.

Tarefas/objetivos: Analisar correções do autor e inspeção como um todo; Re-avaliar qualidade do artefato inspecionado; Decidir sobre a necessidade de uma nova inspeção.

5. Explique a diferença entre teste funcional e teste estrutural. Apresente um exemplo de critério de teste que pode ser utilizado para projetar casos de teste para cada um destes tipos de teste.

Teste Funcional: baseia-se na especificação do software para derivar os requisitos de teste. Aborda o software de um ponto de vista macroscópico.

Critérios: Particionamento em Classes de Equivalência; Análise do Valor Limite; Grafo de Causa-Efeito.

Teste Estrutural: É baseada no conhecimento da estrutura interna da implementação.

Teste dos detalhes procedimentais. A maioria dos critérios dessa técnica utiliza uma representação de programa conhecida como grafo de programa ou grafo de fluxo de controle.

Critérios: Todos os caminhos, todos os usos, todas as definições, todos os usos computacionais, ...

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:

"... 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 8 anos (inclusive) o desconto é de 20%. Entre 8 e 12 (inclusive) o desconto é de 12%. Dos 12 aos 16 (inclusive) o desconto é de 5% e dos 16 aos 18 de 3%..."

Para o teste da faixa 1 (20%) 0-8 anos: CT={-1,0,8,9}

Para o teste da faixa 2 (12%) 9-12 anos: CT={8,9,12,13}

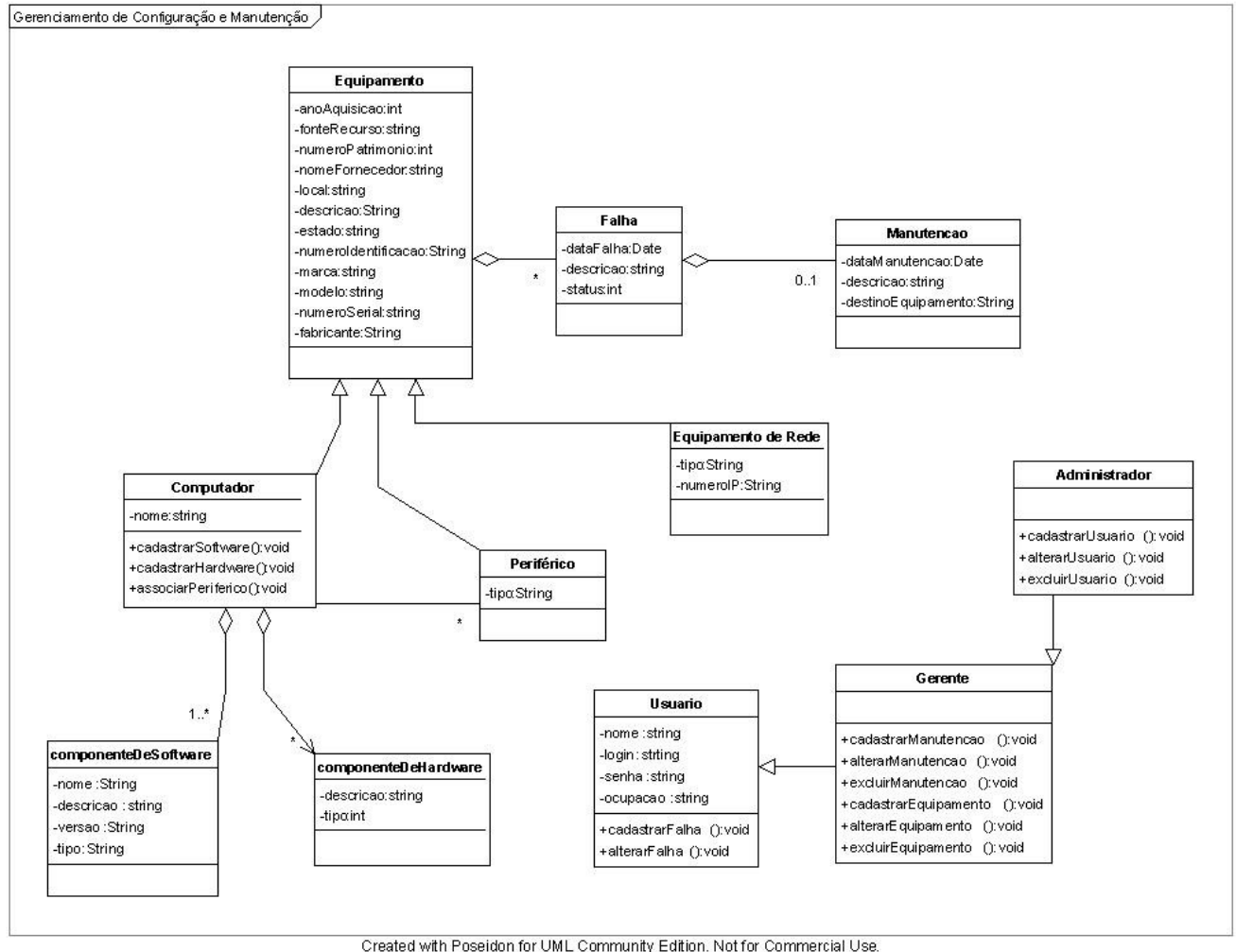
Para o teste da faixa 3 (5%) 13-16 anos: CT={12,13,16,17}

Para o teste da faixa 4 (3%) 17-18 anos: CT={16,17,18,19}

Reunindo estes casos de teste, temos para realizar o teste acima as seguintes entradas da idade do dependente:

CT = {-1,0,8,9,12,13,16,17,18,19}

7. Considerando o Diagrama de Classes abaixo, encontre os valores de NOC (número de filhos), DIT (profundidade de herança) e CBO (acoplamento entre objetos) para as classes identificadas na tabela. Destas quatro classes a que poderia ser mais propensa à falha quando comparada com as outras e por quê? :



CLASSE	NOC	DIT	CBO	Propensa à Falha
Equipamento	3	0	1	
Computador	0	1	3	Em princípio esta classe estaria mais propensa a falha (repare que CBO poderia ser 4 se contássemos a recebida de Equipamento)
Manutenção	0	0	1	
ComponenteDeSoftware	0	0	1	