



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AD2 2º semestre de 2008.

- 1) Explique o que é uma relação de uso entre módulos de um projeto de software. Porque estas relações de uso são importantes no projeto de um sistema? (valor 2,0 pontos)

Uma relação de uso é uma associação estática e direcional entre dois módulos A e B de um sistema que é percebida quando um módulo A utiliza algum tipo, rotina ou variável definido no módulo B.

A relação é estática porque independe do fluxo seguido por um programa em uma determinada execução. A relação é direcional porque dados dois módulos A e B pode ocorrer de A ter uma relação de uso com B e B não ter uma relação de uso com A.

As relações de uso são importantes porque o projeto de software tem como um dos seus objetivos principais a redução destas dependências entre os módulos de um sistema, em particular as dependências que formem ciclos.

- 2) Explique a diferença entre as linguagens de programação interpretadas e as linguagens de programação compiladas. (valor 1,0 ponto)

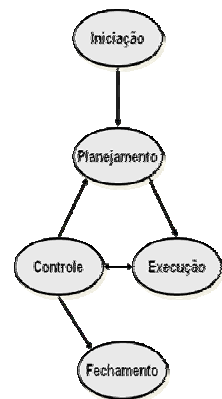
Programas construídos em uma linguagem compilada são transformados em código nativo de uma determinada plataforma computacional e podem ser executados diretamente nesta plataforma. Eles tendem a apresentar maior desempenho (pois são executados diretamente), mas também uma limitação de ser executados somente nesta plataforma.

Programas interpretados são transformados em uma linguagem intermediária – em geral mais simples que a linguagem de programação em si – e executados em um interpretador ao invés de diretamente sobre uma plataforma computacional. Eles tendem a ter menor desempenho, dado que disputam processador com o próprio interpretador que lhes executa, mas podem ser executados em uma diversidade de plataformas computacionais, desde que estas tenham versões do interpretador.

- 3) Quais são os cinco grupos de processos encontrados no gerenciamento de um projeto de software? O que se deve fazer em cada um destes grupos? Desenhe um diagrama mostrando como eles se relacionam? (valor 2,0 pontos)

1. Iniciação: nestes processos são obtidas as autorizações para início do projeto.

2. Planejamento: definição, revisão e manutenção de uma organização de trabalho para a realização do projeto;
3. Execução: coordenação das pessoas e recursos necessários para a execução do plano;
4. Controle: monitoração e medição do progresso para garantir que os objetivos do projeto serão atingidos;
5. Fechamento: homologação e encerramento das atividades do projeto.



- 4) Explique o que é teste funcional. Apresente 3 exemplos de critérios de teste que podem ser utilizados para projetar casos de teste para este tipo de teste. (valor: 1,5 pontos)

Teste funcional toma por base a especificação do software para derivar os requisitos de teste. Aborda o software de um ponto de vista macroscópico, considerando suas funcionalidades. Envolve dois passos principais: identificar as funções que o software deve realizar (especificação dos requisitos, casos de uso), criar casos de teste capazes de verificar se essas funções estão sendo executadas corretamente. A dificuldade do teste funcional está em quantificar a atividade de teste - não se pode garantir que partes essenciais ou críticas do software foram executadas.

Critérios da Técnica Funcional:

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

- 5) Defina métrica objetiva e subjetiva, forneça pelo menos 1 exemplo de cada uma no contexto do software, explicando o que seus valores podem indicar sobre o comportamento do software. (Valor: 1,5 pontos)

Métrica Objetiva: Uma medida absoluta extraída do produto ou do processo.

Usualmente representada numa escala intervalo ou razão

Exemplos: tempo de desenvolvimento, número de linhas de código, número de erros ou modificações

CBO(acoplamento entre objetos): indica o numero de objetos relacionados, ou de certa maneira, o grau de dependência entre objetos. Quanto maior o CBO maior o risco de falha para os objetos da classe.

Métrica Subjetiva: Uma estimativa de extensão ou grau de aplicação de alguma técnica. Uma classificação ou qualificação do problema ou experiência. Normalmente representada numa escala nominal ou ordinal

Exemplos: percepção de qualidade do uso de um método ou técnica, experiência dos programadores na aplicação ou processo.

Experiência do programador: {alta, média, baixa}. Programadores com baixa experiência não são adequados para o desenvolvimento de sistemas complexos e críticos.

- 6) Considerando o Diagrama de Classes abaixo, encontre os valores de WMC (Métodos Ponderados por Classe), 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? (valor: 2,0 pontos):

CLASSE	WMC	DIT	CBO	Propensa à Falha
Computador	3	1	4	Esta classe apresenta maior dependência com outras classes, o que pode representar um risco para seu funcionamento e portanto neste modelo é a mais crítica
Usuário	2	0	0	
Administrador	11	1	0	Esta classe, apesar de ter WMC alto (herança) esta na mesma hierarquia que usuário e gerente. Nesta hierarquia, ela é a mais propensa a falhas.
Gerente	8	2	0	