



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AD2 1º semestre de 2008.

1. Relacione os princípios de projeto de “Modularização” e “Separação de Objetivos”, ou seja, explique como a modularização de um sistema contribui para a separação de objetivos (máximo: 10 linhas).

Um dos objetivos da modularização em um projeto de software é a divisão do projeto em unidades que permitam a separação dos objetivos. Estes objetivos podem ser tanto gerenciais (unidades de implantação do sistema, divisão do trabalho pelas equipes, etc.), quanto técnicos (como redução de dependências entre os componentes do sistema, permitir que os diferentes perfis de profissionais que devem trabalhar em um projeto de software possam conviver gerando um mínimo de impacto no trabalho dos demais, etc.).

2. Explique a diferença entre os modelos dinâmicos da UML: os diagramas de estados e os diagramas de sequência. Explique que tipo de informação é representado em cada diagrama e qual é o objetivo de cada diagrama no processo de modelagem (máximo: 10 linhas).

Um diagrama de estado mostra como o estado interno de uma classe varia ao longo do tempo, sendo composto por estados e transições. Estados são configurações dos atributos de uma classe que identificam situações relevantes daquela classe no contexto do sistema. Transições são fluxos entre estados que pode ocorrer a troca de estados ao longo do tempo.

Um diagrama de sequência mostra como um conjunto de objetos de classes do sistema troca mensagens ao longo do tempo para concluir uma tarefa. Eles são compostos de objetos e mensagens.

3. O que são “números mágicos” na codificação de um sistema? Como eles devem ser tratados no código? (máximo: 5 linhas)

Números mágicos são informações numéricas utilizadas ao longo de um programa e cujo significado não fica explícito no código. Eles devem ser declarados e

referenciados como constantes, ao invés do seu valor aparecer explicitamente ao longo do código.

4. Qual a diferença entre defeito e falha de software? Que técnicas podem ser utilizadas para sua identificação? (máximo: 10 linhas)

Defeito representa o problema existente nos artefatos. Normalmente é inserido devido a atividade humana no desenvolvimento. A falha representa o comportamento apresentado pelo software que não satisfaz aos requisitos. Normalmente uma falha pode ser consequência de um ou mais defeitos existentes no software. Defeitos normalmente são identificados através de técnicas utilizadas no método de inspeção de software. As falhas, por sua vez, são reveladas através dos testes de software.

5. No método de inspeção de software, para a fase de detecção de defeitos, é possível utilizar, a princípio, 3 tipos de técnicas. Cite estas técnicas, descrevendo a diferença básica entre elas. Ao final, indique a que permite mais controle e a que permite mais liberdade para identificação de defeito. (máximo: 10 linhas)

As técnicas de inspeção usualmente mais utilizadas são:

Ad-hoc – a leitura é realizada tendo como base a experiência do inspetor.

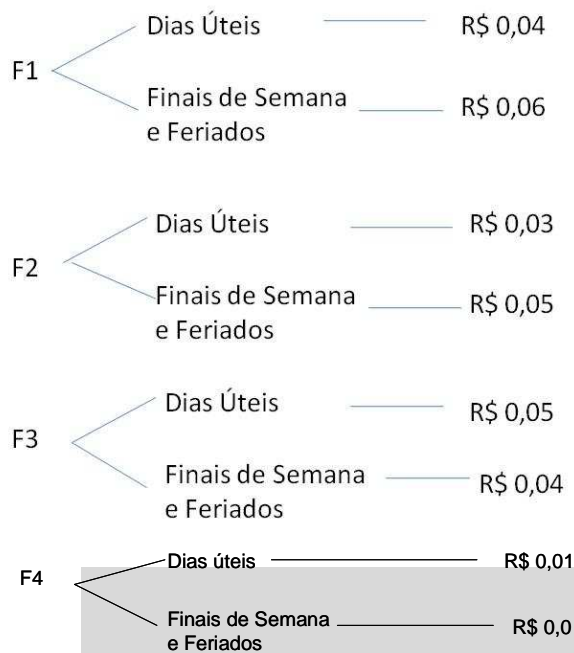
Checklist – a leitura é realizada tendo como base, além da experiência do inspetor, uma lista de características que devem ser observadas visando a identificação ou não dos defeitos

Técnicas de Leitura – a leitura é realizada tendo como base uma sequência de instruções de leitura que direcionam o inspetor a encontrar os defeitos

Ad-hoc permite mais liberdade, porém com rendimento menor. Técnicas de leitura permitem maior controle e melhor rendimento.

6. Utilize a abordagem de grafo de causa e efeito (apresente árvore e tabela de decisão) para representar os casos de teste para o seguinte requisito:

“ Uma companhia telefônica realiza a tarifação das ligações comerciais considerando as categorias de horário F1, F2, F3 (de 06h00min a 21h00min) e F4 (de 21h01min a 05h59min hrs). Nos dias úteis as ligações para as categorias F1, F2 e F3 são tarifadas em R\$ 0,04, R\$0,03 e R\$ 0,05, respectivamente e F4 em R\$ 0,01. Para os finais de semana e feriados, as tarifas das ligações para F1, F2 e F3 são R\$ 0,06, R\$0,05 e R\$ 0,04 respectivamente e F4 é grátis.”



Base Tarifária	F1	F2	F3	F4	F1	F2	F3	F4
Dias da semana	Úteis	Úteis	Úteis	Úteis	Não Útil	Não Útil	Não Útil	Não Útil
Valor da Tarifa	0,04	0,03	0,05	0,01	0,06	0,05	0,04	0,00 (Grátis)

7. Defina métrica objetiva e subjetiva, dando pelo menos 1 exemplo de cada uma no contexto do software. (máximo: 5 linhas)

Uma métrica objetiva representa uma medida absoluta extraída do produto ou do processo. Exemplos: tempo de desenvolvimento, número de linhas de código, número de erros ou modificações.

Uma métrica subjetiva representa uma estimativa de extensão ou grau de aplicação de alguma técnica. Uma classificação ou qualificação do problema ou experiência. Exemplos: qualidade do uso de um método ou técnica, experiência dos programadores na aplicação ou processo.

8. 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
Cria Mapa	0	1	2	

Menu Superior (subclasse)	0	1	3	A principio esta classe seria mais propensa a falhas devido a sua maior dependência com outras classes (maior CBO)
Cria Mapa	0	1	2	REPETIDA
Principal	1	0	1	

