



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

GABARITO - AP3 1º semestre de 2018.

1. Explique qual é a finalidade de um diagrama de casos de uso e cite os principais componentes deste diagrama. (Valor 2,0 pontos)

R. O diagrama de casos de uso tem como principal finalidade representar em um gráfico os requisitos de um sistema e os atores interessados nestes requisitos. Cada requisito é descrito como um caso de uso, representado como uma elipse no diagrama. Os atores, indicados por “bonecos de palito”, representam qualquer pessoa, sistema ou sensor que interaja com o sistema, oferecendo informações a este ou consumindo as informações que ele produz.

2. Dizemos que um projeto "é um esforço temporário, realizado com o intuito de criar um produto ou serviço único de forma incremental". Explique cada um dos três componentes na definição de projeto (temporário, único e incremental) comparando os projetos com seu antagonista, as operações continuadas. (Valor 2,0 pontos)

R. Dizemos que um projeto é único no sentido que o resultado produzido por ele deve introduzir alguma novidade, ou seja, não deve ser apenas a repetição de um produto ou serviço produzido anteriormente. Este produto ou serviço, por sua vez, é criado de forma incremental, suas partes sendo desenvolvidas e agregadas a medida que o projeto está em andamento. Finalmente, por temporário entende-se que o projeto não dura para sempre: ele tem uma data de início e uma data de encerramento.

3. Explique porque um alto grau de acoplamento entre os componentes do software pode tornar sua manutenção muito difícil (valor: 1,0 ponto)

R. O acoplamento indica o grau de dependência entre os componentes (ou módulos) do software. Um software com baixo acoplamento é sempre desejável. Um software com alto grau de acoplamento indica que a dependência entre módulos é grande, aumentando o risco de efeito colateral e, conseqüentemente, de falhas. Em relação a manutenção, o esforço aumenta (e os riscos de falha também) tendo em vista que alterações realizadas em um determinado módulo tem o potencial de afetar os módulos dependentes.

4. Relacione as afirmações com os conceitos associados a elas (1,0 ponto):

(a) Testes de Integração

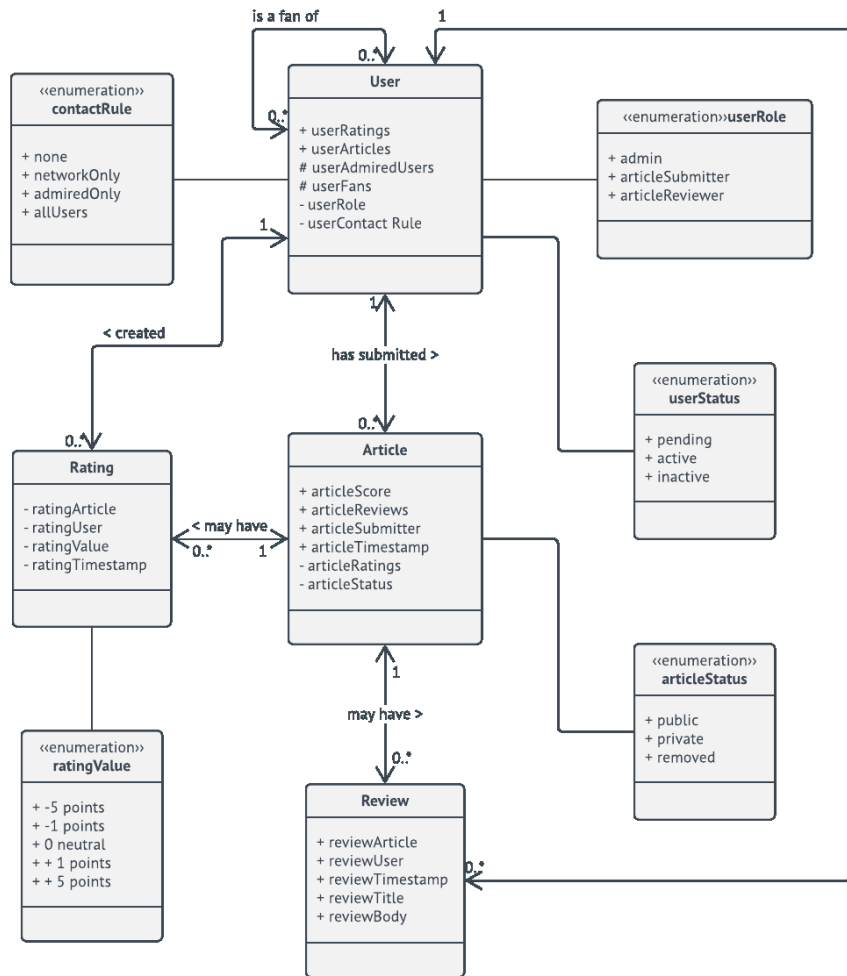
(1) Com a presença do cliente, determina se o sistema funciona de acordo com suas expectativas e com os

requisitos.

- | | |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (b) Testes Unitários | (2) Processo que verifica se os componentes ou módulos do sistema de software funcionam corretamente em conjunto e de acordo com a especificação do projeto (<i>design</i>). |
| (c) Testes funcionais | (3) Aplicado a uma nova versão do software para verificar se esta ainda executa as mesmas funções da mesma maneira que a versão anterior. |
| (d) Testes de regressão | (4) Avalia o sistema quando submetido aos seus limites por um curto período de tempo. |
| (e) Testes de Aceitação | (5) Avalia o sistema para determinar se a funcionalidade descrita na especificação de requisitos executa corretamente pelo sistema já integrado. |
| (f) Testes de estresse | (6) Cada componente ou módulo do software é testado de forma isolada para verificar se este funciona corretamente. |

R. E-1; A-2; D-3; F-4; C-5; B-6

5. 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 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
ContactRule	0	0	1
User	0	0	7
UserRole	0	0	1
Rating	0	0	3
Article	0	0	4
userStatus	0	0	1
ArticleStatus	0	0	1
ratingValue	0	0	1
Review	0	0	2

As classes, neste modelo, apresentam NOC e DIT com valor nulo (não possuem filhos e não definem hierarquia). Entretanto, o acoplamento entre objetos varia de classe a classe, sendo as classes **User**, **Article** e **Rating** que apresentam maior grau de dependência. Nestes casos, estas classes deveriam ser observadas no projetos pois o alto nível de acoplamento implica que elas podem afetar outras classes ou serem afetadas pelas outras classes. Desta forma, estas classes apresentam maior risco de falha, sendo a classe **User** a mais sensível neste modelo.

6. Marque Verdadeiro (V) ou Falso (F) para as afirmações abaixo (Até 2 pontos – 0,13 por marcação correta. Atenção: cada 2 erros/2 brancos/1 erro + 1 branco eliminam 1 acerto!):
- a) Inspeções de Software encontram defeitos e contribuem para a produtividade, qualidade e estabilidade dos projetos de software.
 - b) O melhor testador para o código é o próprio programador.
 - c) Os resultados dos testes não provam que o software está completamente livre de defeitos.
 - d) Ferramentas como analisadores de código, comparadores de arquivos, repositórios de controle de configuração auxiliam no estágio de manutenção do software.
 - e) O esforço de manutenção independe do tipo de sistema, seu tempo de vida, e da qualidade do projeto (*design*), uma vez que a primeira lei de evolução afirma que o software vai mudar continuamente.
 - f) O perfil de experiência do inspetor não afeta a eficiência da inspeção *ad-hoc*.
 - g) É irrelevante considerar as necessidades de diferentes *stakeholders* no planejamento e elaboração de materiais de treinamento e documentação.
 - h) Remover Defeitos é tão bom quanto Prevenir Defeitos.
 - i) O Critério de Cobertura dos Testes influencia a percepção do esforço de teste.
 - j) O fenômeno de rejuvenescimento do software representa o aumento da qualidade de um sistema existente por meio de atividades como reestruturação, engenharia reversa e reengenharia.
 - k) Inspeções de Software podem ser aplicadas a diferentes artefatos do ciclo de vida. Por isso, ao usar inspeções não é necessário aplicar testes.
 - l) O esforço despendido em manutenção perfectiva, conforme observado, é proporcionalmente equivalente ao esforço de manutenções preventiva, adaptativa e corretiva juntas.
 - m) O manual de operação explica as configurações de hardware e software, os métodos para permitir e negar o acesso de um usuário, os procedimentos para adicionar ou excluir periféricos de um sistema e as técnicas para duplicar ou realizar o backup de arquivos e documentos.
 - n) As técnicas de inspeção *ad-hoc*, apenas quando aplicadas por inspetores experientes, garantem a cobertura do documento como um todo.
 - o) O conhecimento adquirido no treinamento pode ser facilmente esquecido para aquelas funções que não são executadas regularmente.

R. V; F; V; V; F; F; F; V; V; V; F; F; V; F; V

Boa prova!