Fundação CECIERJ - **Vice Presidência de Educação Superior a Distância**

Curso de Tecnologia em Sistemas de Computação
Disciplina de Arquitetura e Projeto de Sistemas II
Gabarito – AD2 2° semestre de 2010.

Nome –

Observações importantes:

1- Prova com consulta.

2- ADs enviadas pelo correio devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.

3- Como a avaliação à distância é individual, caso sejam constatadas semelhanças entre provas de alunos distintos, será atribuída a nota ZERO a TODAS as provas envolvidas. As soluções para as questões podem ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser distinta. ALÉM DISSO, às questões desta AD respondidas de maneira muito semelhantes às respostas oriundas dos gabaritos já publicados de ADs de períodos anteriores, será atribuída a nota ZERO, incluindo também cópias diretas e sem sentido de tópicos dos slides das aulas.

Questão 1 [4 pontos]

Em relação à Reutilização de Software, responda:

a) [2 pontos] Explique o que é a Engenharia de Domínio. Além disso, pesquise e explique o que é Linha de Produtos de Software. Existe alguma diferença ou consideração a ser feita sobre estes conceitos?
b) [2 pontos] Da área de Qualidade de Software, sabe-se que o modelo MPS, no contexto do programa de Melhoria de Processo do Software Brasileiro (MPS.BR), se baseia nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos de software e serviços correlatos. Uma vez que uma das metas da Reutilização é agregar qualidade a produtos de software, o modelo MPS apresenta um processo chamado Desenvolvimento para Reutilização (DRU), que visa identificar oportunidades de

reutilização sistemática de ativos na organização e, se possível, estabelecer um programa de reutilização para desenvolver ativos a partir de engenharia de domínios de aplicação. Para atingir este propósito, o processo DRU possui um conjunto de resultados a serem alcançados pela organização. Apresente e discorra sobre cada um destes resultados (DICA: verificar o Guia de Implementação – Parte 5: Fundamentação para Implementação do Nível C do MR-MPS[1]).

Resposta:

a) A Engenharia de Domínio (ED) é o processo de identificar e organizar o conhecimento sobre uma classe de problemas, o domínio do problema, para suportar sua descrição e solução, ou seja, uma abordagem baseada em reutilização para definição do escopo, especificação da estrutura, e construção de recursos para uma classe de sistemas, subsistemas ou aplicações. Uma Linha de Produtos de Software (LPS), por sua vez, é um conjunto de sistemas de software que compartilham um conjunto de características comuns e controladas, que satisfazem necessidades de um segmento de mercado em particular, e são desenvolvidos a partir de artefatos (*core assets*), de forma predefinida. A priori, as denominações ED e LPS eram utilizadas como sinônimos, sendo LPS uma vertente da ED, cujo foco foi transferido para o âmbito empresarial. No entanto, pela análise dos conceitos apresentados e em conformidade com o Software Engineering Institute[2], considerando o surgimento do termo Engenharia de Linha de Produto de Software (ELPS), uma LPS é um produto (i.e., artefato) da ELPS, que contempla as atividades de ED e Engenharia de Aplicação (EA) como parte do ciclo de vida de uma LPS.

b) Os cinco resultados a serem alcançados são:

   a. **DRU1 – Domínios de aplicação em que serão investigadas oportunidades de reutilização de ativos ou nos quais se pretende praticar reutilização são identificados, detectando os respectivos potenciais de reutilização**: o objetivo é verificar se os ganhos proporcionados pela implantação são maiores que os custos, identificando domínios de atuação da organização (i.e., analisando projetos passados e projetos futuros) e os potenciais de reutilização de cada domínio (i.e., estabilidade do domínio, ativos de domínio preexistentes na organização e ativos de domínio passíveis de construção ou aquisição). Ressalta-se que a inexistência de domínios com potencial de reutilização pode justificar a não adoção de um programa de reutilização, mas é necessária a utilização de mecanismos formais de tomada de decisão;
   b. **DRU2 – A capacidade de reutilização sistemática da organização é avaliada e ações corretivas são tomadas, caso necessário**: o objetivo é avaliar a capacidade da organização para executar o processo de forma sistemática, observando recursos humanos capacitados, recursos financeiros

[1] Maiores informações sobre o modelo MPS e sobre os processos do Modelo de Referência MR-MPS estão disponíveis em http://www.softex.br/mpsbr/_guias/default.asp.
[2] http://www.sei.cmu.edu/productlines/. Para mais detalhes sobre LP e ED, acessar http://reuse.cos.ufrj.br/files/publicacoes/mestrado/Mes_PaulaCibele.pdf

para investimento de longo prazo, infra-estrutura apropriada e aspectos culturais trabalhados dentro da organização. Ressalta-se que a inexistência de capacidade de reutilização pode justificar o adiamento da implantação do programa de reutilização, mas deve-se tomar ações corretivas para gerar capacidade e gerenciar os riscos relacionados à implantação do DRU, uma vez que isso esta inexistência não justifica a não adoção de um programa de reutilização;

c. **DRU3 – Um programa de reutilização, envolvendo propósitos, escopo, metas e objetivos, é planejado com a finalidade de atender às necessidades de reutilização de domínios**: este resultado é aplicável quando a organização tem oportunidade e capacidade de reutilização, requerendo a definição de um programa de reutilização, isto é, um documento que inclui propósito, metas, recursos necessários e disponíveis, estágios intermediários a serem atingidos, atividades a serem executadas (incluindo cronograma e responsáveis), indicadores de monitoramento do programa e escopo em que o programa será conduzido;
d. **DRU4 – O programa de reutilização é implantado, monitorado e avaliado**: o objetivo é implantar o programa de reutilização de acordo com o planejado, monitorando a execução do programa de acordo com os indicadores previamente planejados, comparando o planejado com o realizado (e reportando e acompanhando as não-conformidades detectadas), e avaliando periodicamente o programa de reutilização (e tomando as ações corretivas necessárias para a melhoria da execução do processo);
e. **DRU5 – Propostas de reutilização são avaliadas de forma a garantir que o resultado da reutilização seja apropriado para a aplicação alvo**: este resultado busca entender as demandas de reutilização submetidas na forma de propostas de reutilização, as quais permitem a análise da adequação de um ativo de domínio para um problema específico e motivam a construção ou aquisição de ativos de domínio. Ele requer, ainda, a análise das propostas sob a perspectiva do esforço de adaptação de ativo existente, esforço de construção do ativo ou custo de aquisição do ativo, além de uma avaliação da proposta quanto às necessidades e expectativas da organização. Nesse sentido, é um momento propício para que os dados de reutilização dos ativos reutilizáveis sejam registrados a fim de manter a rastreabilidade entre o ativo original e o ativo adaptado para um consumidor específico;
f. **DRU6 – Formas de representação para modelos de domínio e arquiteturas de domínio são selecionadas**: neste resultado, notações capazes de representar domínios e famílias de aplicações em diferentes níveis de abstração devem ser identificadas e selecionadas com o objetivo de definir uma notação para representar modelos de domínio (i.e., fronteira entre domínios, características obrigatórias, características opcionais, características variantes, e dependência e exclusão mútua de características). Adicionalmente, deve-se definir uma notação para representar arquiteturas de domínio, concretizando as características definidas no modelo de domínio;
g. **DRU7 – Um modelo de domínio que capture características, capacidades, conceitos e funções comuns, variantes, opcionais e obrigatórios é**

desenvolvido e seus limites e relações com outros domínios são estabelecidos e mantidos: o objetivo é determinar a fronteira dos domínios com potencial de reutilização e domínios correlatos no contexto do programa de reutilização, o que impacta os domínios a serem incluídos no programa no futuro. Deve-se desenvolver também modelos de domínio para cada domínio no escopo do programa de reutilização, utilizando a notação definida no DRU6, considerando modelos de domínio como ativos reutilizáveis (controlar com o processo Gerência de Reutilização) e controlando a evolução dos modelos de domínio (controlar com o processo Gerência de Configuração);

h. **DRU8 – Uma arquitetura de domínio descrevendo uma família de aplicações para o domínio é desenvolvida e mantida por todo o seu ciclo de vida**: considerando que arquitetura de domínio corresponde a famílias de aplicações para um dado domínio, identificando os ativos de domínio e os seus relacionamentos, o objetivo deste resultado é desenvolver arquiteturas de domínio para cada domínio no escopo do programa de reutilização, utilizando a notação definida no DRU6, considerando as arquiteturas de domínio como ativos reutilizáveis (controlar com o processo Gerência de Reutilização) e controlando a evolução das arquiteturas de domínio (controlar com o processo Gerência de Configuração). Por fim, deve-se priorizar os ativos de domínio identificados;

i. **DRU9 – Ativos do domínio são especificados; adquiridos ou desenvolvidos, e mantidos por todo o seu ciclo de vida**: o objetivo é especificar todos os ativos de domínio identificados na arquitetura de domínio seguindo a priorização previamente definida, incluindo um detalhamento das funcionalidades do ativo de domínio. Deve-se analisar (custo *versus* benefício) e avaliar a aquisição ou desenvolvimento do ativo, isto é, se a organização deve comprar o ativo (utilizar o processo Aquisição), desenvolver o ativo (fazendo refatoração de ativos existentes em projetos anteriores), não investir em um ativo em particular ou incorporar os ativos na biblioteca de ativos reutilizáveis (utilizar o processo Gerência de Reutilização).

Questão 2 [3 pontos]

O padrão *Information Expert* visa atribuir a responsabilidade ao especialista, isto é, à classe que tem a informação necessária para satisfazer a responsabilidade.

a) [1 ponto] Descreva um exemplo de uma situação onde esse padrão é útil.
b) [1 ponto] Quais são os benefícios alcançados com o uso desse padrão?
c) [1 ponto] Desenhe um modelo de classes exibindo como o padrão pode ser utilizado na situação descrita.

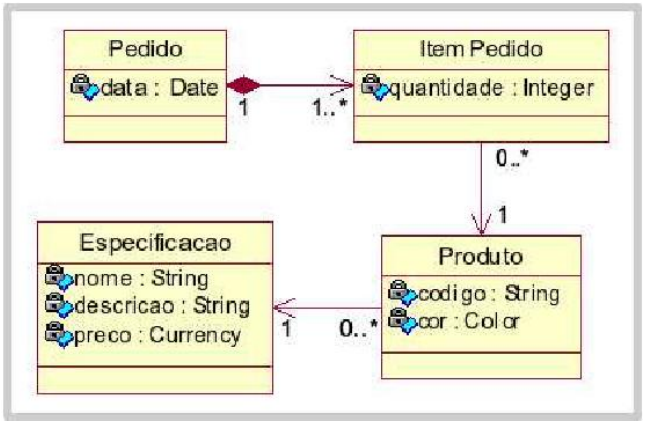
Resposta:

a) Em um sistema de PDV, o responsável pelo cálculo do total de um pedido deveria ser uma classe `Pedido`, considerando o padrão *Information Expert*, uma vez que deve-

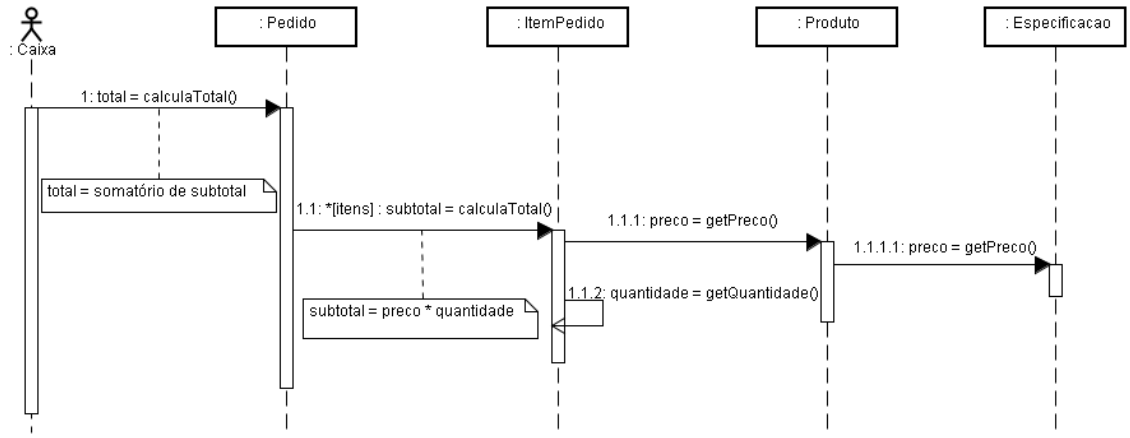
se atribuir a responsabilidade ao especialista, ou seja, à classe que tem a informação necessária para satisfazer a responsabilidade (no caso, o cálculo do total do pedido).

b) Entre os benefícios deste padrão, estão: (i) leva a projetos onde o objeto de software faz o que o objeto real faria; (ii) mantém o encapsulamento e não aumenta o acoplamento, pois utiliza informações próprias; e (iii) distribui o comportamento entre as classes do sistema, aumentando a coesão das mesmas.

c) Para o seguinte modelo:



a aplicação do padrão supracitado para o cálculo do total de um pedido em um sistema de PDV pode ser visualizado pelo diagrama de seqüência abaixo:



Questão 3 [3 pontos]

Em relação a Componentes, responda:

a) [2 pontos] Explique o conceito de componente de forma a incluir pelo menos **cinco** de suas propriedades fundamentais. Além disso, faça um desenho que exemplifique estas propriedades, seja exemplificando-as por meio de um exemplo de componente, seja montando um modelo ou analogia para descrevê-las.

b) [1 ponto] No que se refere ao escopo do projeto de um componente, o que é mais vantajoso ou interessante: *um componente com muitas funcionalidades, ou com poucas funcionalidades*? Argumente a sua resposta.

Resposta:

a) Componente é uma unidade de software independente, que encapsula dentro de si seu projeto e implementação, oferecendo serviços por meio de interfaces bem definidas. A seguir, são explicadas as cinco propriedades fundamentais destacadas na ordem em que foram sublinhadas:
   i. **Identificação**: componentes devem ser facilmente identificados, ou seja, devem estar armazenados em um único lugar, ao invés de espalhados e misturados com outros artefatos de software ou documentação;
   ii. **Autocontido**: característica dos componentes de poderem ser reutilizáveis sem a necessidade de incluir ou depender de outros componentes. Caso haja alguma dependência, todo o conjunto deve ser abstraído como um componente reutilizável;
   iii. **Documentação**: A existência de documentação é indispensável para a reutilização. O tipo de componente e a sua complexidade indicarão a conveniência do tipo de documentação;
   iv. **Funcionalidade**: componentes têm uma funcionalidade clara e específica que realizam e/ou descrevem. Assim, componentes podem realizar funções ou podem ser simplesmente descrições de funcionalidades (e.g., artefatos do ciclo de vida);
   v. **Interface**: componentes devem possuir interfaces claras, que indicam como estes podem ser reutilizados e conectados a outros componentes, devendo-se ocultar os detalhes que não são necessários para a reutilização per si (encapsulamento).

b) Dependendo do contexto do componente e/ou do projeto e da experiência do desenvolvedor e/ou organização, por exemplo, diversas vantagens (e respectivas desvantagens justificadas sobre cada uma dessas vantagens) para cada uma das escolhas mencionadas poderiam ser apontadas. No entanto, analisar o escopo de um componente requer uma análise de contexto e experiência. Um componente de conversão de moeda para um sistema *web* de uma companhia aérea é bem delimitado em seu escopo, isto é, possui poucas funcionalidades (ou uma, no caso) e é pequeno em termos da medida de uma métrica como LoC (linhas de código). Por outro lado, um *framework* para desenvolvimento *web* como Struts ou JSF, com diversas funcionalidades e grande em termos da medida da métrica LoC, no contexto de um sistema nacional para integrar subsistemas de saúde, educação, receita, eleitoral etc. com milhões de usuários e com vários requisitos não funcionais como disponibilidade e confiabilidade, também é bem delimitado em seu escopo e, diante de outros componentes deste sistema (onde um subsistema pode ser um componente), pode parecer oferecer poucas funcionalidades e até ser pequeno em termos de própria medida mencionada, no contexto em que está inserido. Adicionalmente, um *framework* destes, no contexto do sistema *web* de uma companhia aérea, pode parecer

grande em termos da medida em questão, e com várias funcionalidades. Assim, o vantajoso, ou interessante, é que o componente atenda ao conceito mostrado em seu conceito em (a), apresentando um escopo bem definido em seu projeto e construção.