



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 da AP1 – 2º semestre de 2010

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

Questão 1 (2 pontos)

Em relação ao processo de software,

- (a) Defina com suas palavras o ciclo de vida cascata e o ciclo de vida evolutivo. Explícite, também com suas palavras, as suas semelhanças e diferenças.

Resposta: O ciclo de vida cascata consiste em executar, em série, as atividades de análise, projeto, codificação e testes. Por outro lado, o ciclo de vida evolutivo prega o refinamento gradativo do conhecimento. Desta forma, ambos têm como propósito a explicitação dos requisitos e transformação dos requisitos em código, mas o ciclo de vida evolutivo permite que versões parciais do software sejam produzidas, e que os requisitos possam ser refinados a partir do conhecimento gerado por essas versões parciais.

- (b) Detalhe com suas palavras as fases do processo unificado, discutindo o foco principal de cada fase.

Resposta: O processo unificado é decomposto em quatro fases: concepção, elaboração, construção e transição. Na concepção o foco principal é entender o problema. Nesse momento é feito um planejamento inicial, ainda propenso a incertezas. Os riscos são identificados. Na elaboração o foco é definir estratégias de solução para o problema. O escopo do produto começa a se consolidar, o que permite que o planejamento fique mais concreto. É feita a mitigação dos riscos. Na construção o foco é a codificação da solução escolhida, apesar dessa codificação ter iniciado de fato desde a concepção. Embora o

planejamento ainda esteja evoluindo, de modo geral o escopo já está bem estabelecido. Finalmente, a transição representa a passagem do produto para o cliente e o início de um novo ciclo de desenvolvimento. Nesse momento, é possível observar com clareza eventuais falhas do ciclo que está terminando para evitar que elas se repitam num novo ciclo.

Questão 2 (3 pontos)

Em relação a projeto de software,

- (a) Discuta com suas palavras para que servem os modelos de sequência e de transição de estados da UML.

Resposta: O modelo de sequência serve para explicitar como objetos colaboram visando implementar uma determinada funcionalidade. Por outro lado, o modelo de transição de estados explicita os estados que um determinado objeto pode atingir, e as razões que levam às transições entre estados.

- (b) Descreva com suas palavras o que são invariantes, pré-condições e pós-condições existentes em contratos de software.

Resposta: Invariantes são restrições de uma classe que devem sempre ser verdadeiras em estados de equilíbrio dessa classe. Pré-condições são restrições que devem ser verdadeiras antes da execução de um determinado método. Pós-condições são restrições que devem ser verdadeiras depois da execução de um determinado método.

- (c) Explique com suas palavras o raciocínio por trás da heurística “Em sistemas com interface com usuário, as classes de modelo não devem depender das classes de interface”.

Resposta: Essa heurística existe para preservar a independência das classes de modelo. Como as classes de interface são mais volúveis, mudando rapidamente em função de evoluções tecnológicas, a independência de modelo para interface permite a substituição da interface com maior facilidade. Ou seja, as classes de modelo podem ser reutilizadas com outras classes de interface de forma facilitada.

Questão 3 (2 pontos)

Utilizando as suas próprias palavras responda:

- (a) Para que servem os modelos de sistemas?

Resposta: Modelos são úteis para o entendimento de problemas, a difusão do conhecimento entre os desenvolvedores e a possibilidade de testar hipóteses antes de realizá-las. Eles são criados devido à limitação do ser humano em lidar com a complexidade.

(b) Quais são as dificuldades na engenharia de requisitos?

Resposta: Algumas das dificuldades são: (i) problemas de comunicação entre desenvolvedor e cliente, (ii) mudanças frequentes nas solicitações realizadas pelos clientes, (iii) falta de entendimento completo sobre o problema a ser tratado, e (iv) a falta de padronização na forma de representar o problema a ser resolvido e os requisitos a serem contemplados pelo sistema.

Questão 4 (3 pontos)

Em relação ao Desenvolvimento de Software Orientado a Objetos (DSOO),

(a) Descreva com suas palavras as principais características e dificuldades do DSOO.

Resposta: Trata-se de um desenvolvimento que envolve um processo iterativo, evolutivo e ágil que explora a reutilização de software e oferece uma representação conceitual única durante o processo de desenvolvimento. Algumas de suas dificuldades são: a identificação de casos de uso, a classificação e organização de classes e objetos e ainda o detalhamento do comportamento de classes e objetos.

(b) O que significa RUP e quais são seus conceitos-chave?

Resposta: RUP é um framework de genérico para processos podendo ser especializado para diferentes classes de sistemas, áreas de aplicação, organizações, níveis de competência e tamanhos de projeto. Seus conceitos-chave são: o direcionamento a casos de uso, centrado em arquitetura e desenvolvimento iterativo-incremental.

(c) Qual é o princípio básico da orientação a objetos e qual é o seu objetivo?

Resposta: O princípio básico é o encapsulamento, cujo objetivo é restringir o escopo de informação para atingir legibilidade, manutenibilidade e reutilizabilidade. O encapsulamento proíbe a visualização da realização de uma entidade.