



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AP1 1º semestre de 2018.

- 1) Explique a diferença entre técnicas *top-down*, *bottom-up* e *middle-out* na Engenharia de Software. (valor 2.0 pontos)

Técnicas *top-down* partem de elementos mais complexos e descem a níveis cada vez mais detalhados (refinamento a partir dos componentes mais complexos). Técnicas *bottom-up* partem de componentes menores, que são agrupados para formar os principais componentes do sistema (agrupamento dos componentes de mais alto nível a partir dos componentes menores). Por fim, técnicas *middle-out* partem de componentes intermediários, compondo os elementos mais complexos e refinando os intermediários para identificar os mais simples.

- 2) Explique o que é um caso de uso e como ele é utilizado no levantamento de requisitos em um projeto de software. Deixe claro o que diferencia um caso de uso das técnicas anteriores de levantamento de requisitos. (valor 2.0 pontos)

Um caso de uso é um conjunto de ações, incluindo variantes de ações, que o sistema pode realizar e que oferece algum benefício observável para o usuário. Ele é baseado em um cenário descritivo de como um ator interage com o sistema, identificando os eventos que podem ocorrer e as respostas esperadas a estes. O conjunto de casos de uso de um sistema representa todas as operações disponíveis para seus usuários, ou seja, seus requisitos. A principal diferença entre os casos de uso e técnicas anteriores de levantamento de requisitos é a clareza com que os casos de uso se relacionam com os usuários interessados no seu uso, fazendo com que seja mais fácil atribuir cada requisito ao conjunto de patrocinadores (*stakeholders*) interessados.

- 3) O que é prototipação e porque este grupo de técnicas é importante na Engenharia de Software? (valor 2.0 pontos)

Prototipação é a construção de versões mais simples dos sistemas para verificar o entendimento dos requisitos, a qualidade da interface com o usuário e a capacidade de um projeto resolver o problema. Estas técnicas são importantes para validar o

entendimento do analista de requisitos com seus patrocinadores antes de investir todo o esforço necessário para desenvolver o software definitivo.

- 4) Explique as principais diferenças entre as atividades de análise e projeto no ciclo de vida de desenvolvimento de software. (valor: 2,0 pontos)

A atividade de análise de requisitos tem como objetivo descrever o que o software deve fazer e para quem devem ser construídas estas funcionalidades, sem se preocupar com questões técnicas do desenvolvimento de software. A atividade de projeto tem como objetivo, dado um conjunto de requisitos, definir uma solução computacional para sua implementação.

- 5) Quais das técnicas abaixo não podem ser utilizadas para apoiar a identificação de requisitos junto aos usuários durante a análise dos requisitos de um software? Escreva os números das técnicas selecionadas na folha de resposta. (valor 2.0 pontos)

1. *Storyboarding* e prototipação
2. Gerenciamento de projetos
3. Reuniões de *brainstorming*
4. Entrevistas estruturadas
5. Ciclo de vida em cascata
6. Role *playing*
7. Modularização

Resp: 2, 5, 7