



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AD2 1º semestre de 2016

Atenção: para responder a algumas das questões você necessitará de ler o material de aula e eventualmente realizar pesquisas na internet. ****não copie! Defina e explique com suas próprias palavras, informando quais fontes de informação você utilizou ****

1. O conhecimento aplicado no gerenciamento de projetos pode ser dividido em várias disciplinas. Algumas destas disciplinas são chamadas de "disciplinas centrais" por dominarem a etapa de planejamento no gerenciamento de projetos. Quais são as três disciplinas centrais e qual é o papel de cada uma? (valor 2,0 pontos).

As três disciplinas centrais são o gerenciamento de escopo, de tempo e de custos. O gerenciamento de escopo tem como principal objetivo definir o trabalho que deve ser realizado no contexto do projeto. A divisão deste trabalho em tarefas e sua distribuição pelos integrantes da equipe responsável pelo projeto são os principais objetivos do gerenciamento de tempo, que resultará em um cronograma para o projeto. O gerenciamento de custos parte do cronograma, em especial do número e perfil das pessoas envolvidas na realização das suas tarefas, para chegar ao custo total do projeto e sua linha de desembolso.

2. No material de aula falamos sobre diagramas de sequência e de colaboração. Ambos se propõem a representar a troca de mensagens pela qual se realizam cenários de um caso de uso. Mas quais são as principais diferenças entre os diagramas de sequência e colaboração? (valor: 2,0 pontos)

Os dois diagramas apresentam a mesma informação e as diferenças entre eles são apenas visuais. Os diagramas de colaboração não apresentam a linha de tempo dos objetos: os objetos são representados por retângulos espalhados na área do diagrama e as mensagens são representadas por setas numeradas. Os números nas setas representam a ordem em que as mensagens são trocadas entre os objetos.

3. Atividade de pesquisa: O que é um sistema de sistemas (*system of systems*)? Que características estes sistemas apresentam que os diferenciam de sistemas convencionais? Dê um exemplo de um sistema de sistemas e aponte que características este sistema apresenta. (valor: 2,0 pontos)

Um sistema de sistemas, na perspectiva da engenharia de software, representa um sistema complexo, formado por diferentes partes que são, por si, sistemas de software. As propriedades esperadas para um sistema convencional devem ser abstraídas e extrapoladas para um sistema de sistemas, incluindo preocupações adicionais quanto a interoperabilidade sistêmica, o que torna o projeto destas soluções complexo.

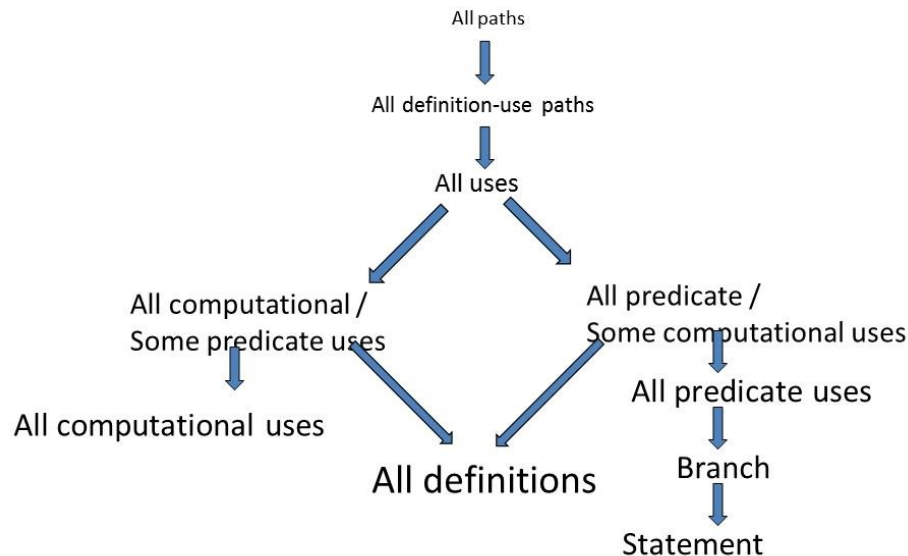
Sistemas de Informação integrados constituem um exemplo de um sistema de sistemas bem simples. Outros exemplos dizem respeito a, por exemplo, os sistemas de controle de edifícios inteligentes, nos quais diferentes sistemas de software (associados a diferentes dispositivos tais como painéis solares, iluminação, refrigeração, etc) devem ser usados de forma integrada e ótima. Uma outra maneira de naturalmente obter sistema de sistemas é pensar na possibilidade de alta interoperação de dispositivos e a capacidade de embarcar software em praticamente todos os elementos presentes na vida diária da sociedade, conforme podemos perceber na tendência tecnológica inerente a Internet das Coisas (IoT – Internet of Things).

A seguir alguns exemplos de fonte de informação (existem outros):
<http://rs.ieee.org/component/content/article/9/77-system-of-systems.html>
<http://www.sei.cmu.edu/sos/>
https://en.wikipedia.org/wiki/System_of_systems

4. Explique o que é teste estrutural. Apresente três exemplos de critérios de teste que podem ser utilizados para projetar casos de teste para este tipo de teste. (valor: 2,0 pontos)

Teste Estrutural toma por base a estrutura interna do software para planejar o teste. Neste sentido, fluxo de controle ou de dados servem como critérios para estabelecer os casos de teste e roteiros de teste. Usualmente, a estrutura do software é visualizada através do grafo do programa ou grafo de fluxo de controle.

Diferentes critérios podem ser utilizados: testar todos os nós do grafo, todos os arcos, todos os caminhos e outras opções. Quanto mais forte o critério (maior cobertura), maior o esforço do teste (planejamento e execução).



5. Explique o que você entende pelos termos defeito, falta, erro e falha. Explique a diferença entre eles, e cite pelo menos uma técnica de verificação, validação e teste que pode ser usada para identificar cada um deles. Dê um exemplo de como defeitos podem ser classificados. (valor 2,0 pontos)

A falha representa um comportamento inesperado do software, ou seja, observa-se um comportamento diferente do que foi especificado. Uma falha pode ser consequência de uma ou mais faltas presentes no software, consequência de um erro do desenvolvedor. Usualmente, refere-se a estas faltas como defeitos. Falhas podem ser reveladas pelos testes de software. Da falha se pode chegar aos defeitos. Por sua vez, defeitos podem ser detectados por inspeção do software.

Inspeções e revisões de software são adequadas para a identificação de defeitos (faltas) nos artefatos de software.

Testes de Software são adequados para revelar falhas no software, que usualmente são devido a defeitos no produto, que devem ser encontrados e corrigidos.

Defeitos podem ser classificados de diferentes formas. Uma possível taxonomia é classifica-los como omissão, fato incorreto, inconsistência, ambiguidade ou informação estranha.