



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Engenharia de Software

AD2 2º semestre de 2016

Atenção: para responder a algumas das questões você necessitará 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) Quais são as cinco grandes fases (ou grupos de processos) em que podemos dividir o gerenciamento de projetos de software? O que se faz em cada uma destas etapas? Desenhe um diagrama mostrando como elas se relacionam. (valor 2,5 pontos)

As cinco fases em que podemos dividir o gerenciamento de projetos são a iniciação, o planejamento, a execução, o controle e o fechamento. A iniciação é a primeira fase do projeto, em que a alta administração da empresa autoriza o início do projeto e autoriza o gerente de projetos à sua coordenação. No planejamento, o gerente de projetos e a equipe definem um plano para o desenvolvimento do projeto. A execução consiste em colocar o plano em prática e o controle consiste em comparar o plano com o andamento da execução. Por fim, o fechamento é o encerramento administrativo do projeto.

- 2) Explique quais são as características de um bom programa. Porque é importante escrever o programa pensando em sua manutenção do que apenas nas funcionalidades entregues aos usuários? (valor: 2,5 pontos)

Acima de tudo, um bom programa deve ser correto, ou seja, deve fazer o que sua especificação estabelece e deve se comportar de forma previsível quando dados fora do padrão lhe são apresentados. Depois, um bom programa deve ser claro, no sentido de facilitar o seu entendimento por outros programadores além de quem o escreveu. Por fim, um bom programa é moderado no consumo de recursos, tais como memória, bateria e capacidade de processamento.

- 3) Atividade de pesquisa: O que é *DevOps*? Como *DevOps* afeta o processo construtivo do software? (valor: 2,5 pontos)

DevOps (*Development and Operations*/Desenvolvimento e Operação) é um neologismo que representa um movimento de profissionais de tecnologias de informação e comunicação focado em uma atitude diferenciada em relação a entrega de software através da colaboração entre a equipe de desenvolvimento e operadores. Esta colaboração tem o objetivo de melhorar a qualidade dos processos de desenvolvimento e operação, assim como a qualidade dos produtos e serviços.

DevOps afeta o processo construtivo do software ao propor um conjunto de princípios e práticas, como por exemplo a **automação** de tarefas repetitivas para reduzir os esforços desnecessários permitindo entregas mais rápidas, consistência da infraestrutura, maior produtividade da equipe e facilidade na repetição de tarefas.

Algumas práticas de DevOps são baseadas nos princípios do *Lean Thinking* (**Leanness**) com o objetivo de garantir o fluxo contínuo de desenvolvimento e entregas regulares através de mudanças pequenas e incrementais. Essa agilidade nas entregas promove constante *feedback* entre as equipes de desenvolvimento, operação e os clientes.

Referências

- de França, B. B. N.; Jeronimo Junior, H. & Travassos, G. H. **Characterizing DevOps by Hearing Multiple Voices** Proceedings of the 30th Brazilian Symposium on Software Engineering, ACM, 2016, 53-62
- puppet labs & it revolutions press. state of devops report. puppet labs, 2013.
- mueller, ernest. what is devops? the agile admin. 2010. retrieved from: <http://goo.gl/md3lu1>.
- erich, floris, amrit, chintan, and daneva, maya. report: devops literature review. university of twente, 2014.
- walls, mandi. building a devops culture (2013).
- debois, patrick. devops areas - codifying devops practices. jedi (patrick debois' blog). 2014. retrieved from: <http://goo.gl/25scb3>.

- 4) Explique o que é teste funcional. Apresente três exemplos de critérios de teste que podem ser utilizados para projetar casos de teste para este tipo de teste. (valor: 2,5 pontos)

Teste Funcional é uma técnica utilizada para projetar casos de testes que considera o software como uma caixa preta ignorando os detalhes de implementação. Nos testes funcionais, são fornecidas entradas e as saídas geradas são avaliadas para verificar se estão em conformidade com os requisitos. Além disto, nesta técnica o software é avaliado segundo o ponto de vista do usuário.

Exemplos de critérios que podem ser utilizados são:

- Particionamento em Classes de Equivalência: divide o domínio de entrada de um sistema em classes (categorias) de dados das quais casos de testes podem ser derivados;
- Análise de Valor Limite: explora os limites dos valores para preparar os casos de teste, pois erros tendem a ocorrer nos limites do domínio de entrada;
- Grafo Causa Efeito: técnica para identificação de casos de teste que explora as condições lógicas e as ações correspondentes.