

Exercícios para Entrega – Engenharia de Software

QUESTÃO 01: Apresente um exemplo de:

- a) Software embutido: impressoras;
- b) Software básico: compiladores;
- c) Software de tempo real: sistemas de controle de tráfego aéreo;
- d) Software de Computador pessoal: Excel.

QUESTÃO 02: Ao atingir 50% de progresso em um projeto, você identificou que o mesmo se encontra atrasado. A melhor solução para esse atraso seria contratar mais programadores? Justifique a sua resposta.

Provavelmente a melhor solução não seria contratar mais programadores, visto que essa contratação pode gerar um conflito de ideias entre a equipe devido às diferentes competências de cada um. Além disso, o tempo usado para a habituação dos possíveis novos membros também pode atrasar o processo, já que será necessário que os atuais programadores destinem um tempo para auxiliar nas demandas iniciais dos novos integrantes, reduzindo assim, o tempo produtivo para o desenvolvimento de fato do produto.

QUESTÃO 03: Cite dois mitos de software.

- “Todo software para uma determinada atividade deve seguir o mesmo padrão”: o fato de se prender a soluções já conhecidas pode não otimizar o trabalho da equipe, impedir que sejam adotadas novas ferramentas e até inviabilizar até a produtividade da empresa. É importante que cada caso seja analisado separadamente.
- “Erros comuns podem ser desprezados para serem corrigidos futuramente”: a aceleração do processo de desenvolvimento de um software pode levar a uma sucessão de erros de difícil solução/entendimento no futuro, acarretando em retrabalho na revisão do código inteiro, gastos desnecessários e possível atraso na entrega do produto.

QUESTÃO 04: Defina ciclo de vida de software e dê ao menos quatro exemplos de ciclos de vida de software.

O ciclo de vida de um software é uma estrutura que define quais processos serão utilizados em toda a vida do sistema (fase de desenvolvimento, operação e manutenção). São exemplos dos modelos: Cascata, Incremental, Prototipagem, V e Espiral.

QUESTÃO 05: Qual ciclo de vida concentra-se na capacidade de se especificar o software a uma máquina em um nível que esteja próximo à linguagem natural?

Trata-se das Técnicas de 4º Geração, onde no processo de desenvolvimento são utilizadas ferramentas como: linguagens para consulta de banco de dados, geração de códigos e fluxogramas e capacidade gráfica de alto nível.

QUESTÃO 06: Qual o cenário ideal para o uso da prototipação?

A prototipação, que consiste basicamente na materialização de uma ideia/produto (construção de um exemplar), feita com o intuito de facilitar o entendimento dos desenvolvedores, assegurar que o produto atenda o mais fielmente possível a necessidade do cliente e evitar possíveis gastos desnecessários, apresenta seu melhor cenário de uso quando um cliente expõe uma ideia de software, porém, não especifica detalhadamente o que deve ser entregue (como os requisitos de entrada, saída e processamento).

QUESTÃO 07: Diferencie o ciclo de vida clássico do ciclo de vida em espiral. Para cada um desses dois ciclos de vida, apresente ao menos um exemplo de tipo de projeto ou equipe ou situação em que o uso do ciclo em questão se apresenta como uma melhor escolha.

O ciclo de vida clássico (modelo cascata ou waterfall) é um modelo simples, de fácil entendimento e com prazos de início, fim e custo do projeto bem estabelecidos. Apresenta uma sequência das fases, onde cada fase só começa quando a anterior termina e a próxima utiliza os resultados obtidos na anterior, o que justifica seu nome. Nesse modelo, é dada muita importância às análises para a compreensão dos requisitos do sistema antes de iniciar o processo de programação em si, a fim de que o objetivo do software esteja bem definido e que sejam evitados retrabalhos, ou seja, é indicado quando já se tem essas informações no início do projeto.

O modelo espiral apresenta uma abordagem cíclica das fases do processo, onde a cada iteração é apresentado versões evolucionárias do sistema. A constante atualização do sistema tem como consequência o ajuste do custo e cronograma no processo de desenvolvimento. É um modelo guiado por risco, que geralmente recorre à prototipação, e suporta sistemas complexos e/ou de grande porte, onde falhas não são toleráveis, como em softwares astronômicos e científicos.

QUESTÃO 08: Cite os princípios dos métodos ágeis em uma lista em que o primeiro princípio é aquele que você julga ser o mais importante, seguido do segundo mais importante e assim sucessivamente. Justifique o motivo da escolha das posições de cada princípio em sua lista.

- Pessoas, não processos: é extremamente necessário para o desenvolvimento eficiente do projeto que as pessoas estavam satisfeitas com sua ocupação profissional e com uma boa convivência com todo o time;
- Manter a simplicidade: a adoção de soluções simples (que podem ser sofisticadas ao mesmo tempo) facilita o entendimento e o desenvolvimento de todo o processo;
- Aceitar mudanças: princípio extremamente importante para praticar a adaptabilidade e evitar possíveis frustrações já que essas mudanças são recorrentes e esperadas no desenvolvimento de um software;
- Entrega incremental: facilita o entendimento tanto do desenvolvedor quanto do cliente.

garantindo que o produto final atenda o mais fielmente possível as requisições solicitadas.

- Envolvimento do cliente: a confiança do cliente no processo é extremamente importante para que sejam passadas corretamente todas as informações a equipe de desenvolvimento a fim de atender suas necessidades e evitar possíveis gastos desnecessários ou retrabalho.