

Engenharia de Software I

Exercícios

- Introdução a Engenharia de Software

* Nas questões dissertativas (a partir da questão 6), você deve se basear nos slides de Introdução, porém deve escrever com suas palavras e pode dar exemplos se achar pertinente, os comentários podem ser sucintos.

Obs. Você também pode fazer pesquisas na internet ou na bibliografia da disciplina para **completar** sua resposta.

Na primeira aula após a entrega da atividade faremos a correção e comentários das respostas feitas por vocês.

1) De acordo com Pressman, Sistema de Informação é um conjunto ou disposição de elementos, organizado para executar certo método, procedimento ou controle ao processar informações. Um sistema de informação possui entradas e saídas. Quais são os elementos de um Sistema de Informação?

- A) Técnicas, Máquinas, Banco de Dados, Pessoas, Software
- B) Métodos, Hardware, Software, Banco de Dados, Telecomunicações, Máquinas
- C) Procedimento, Hardware, Software, Pessoas, Banco de Dados, Telecomunicações
- D) **Procedimento, Hardware, Software, Pessoas, Banco de Dados**

2) O software é um dos elementos de um sistema de informação, ele é composto por:

- A) **Instruções, Estruturas de Dados e Documentos**
- B) Instruções, Banco de Dados e Documentos
- C) Instruções, Dados e Documentos
- D) Métodos, Estruturas de Dados e Manuais

3) Existem vários tipos de software, são eles: Básico, Tempo Real, Comercial, Científico, Embarcado, Computador Pessoal e Inteligência Artificial. Qual a alternativa que corresponde respectivamente a 1 exemplo de cada tipo de software?

- A) Compilador, Financeiro, Montadora de carros, Teclado de Microondas, Biologia molecular, Sistema Especialista e Editor de texto
- B) **Compilador, Montadora de carros, Financeiro, Biologia molecular, Teclado de Microondas, Editor de texto e Sistema Especialista**
- C) Sistema Especialista, Editor de texto, Compilador, Biologia molecular, Teclado de Microondas, Financeiro e Montadora de carros
- D) Sistema Especialista, Editor de texto, Compilador, Biologia molecular, Financeiro, Montadora de carros e Teclado de Microondas

4) Durante a evolução do software: 1) o hardware sofreu contínuas mudanças, o software era uma arte "secundária" para a qual havia poucos métodos sistemáticos e não havia documentação; 2) Surgiram sistemas multiusuários, sistemas de gerenciamento de banco de dados e empresas software houses; 3) Cresce o número de sistemas baseados em computador, manutenção de software quase impossível. Com isso, veio a crise do software a que referente a um conjunto de

problemas encontrados no desenvolvimento de software. Os problemas desta crise estão relacionados a:

- A) Estimativas de prazo, Insatisfação do cliente, Custo e Dificuldade de manutenção
- B) Custo, Insatisfação do cliente, Qualidade de software
- C) Custo, Insatisfação do cliente e Dificuldade de manutenção
- D) Estimativas de prazo, Insatisfação do cliente, Qualidade de software e Dificuldade de manutenção

5) As etapas do ciclo de vida de um software são compostas pelas seguintes ações: Estabelecer os requisitos e restrições do sistema; Produzir um modelo documentado do sistema; Construir o sistema; Verificar se o sistema atende às especificações requeridas; Liberar o sistema para o cliente e garantir que ele seja operacional; Eliminar defeitos e evoluir o sistema conforme demanda. Qual a sequência das etapas?

- A) Análise, Projeto, Implementação, Testes, Implantação e Manutenção
- B) Análise, Projeto, Implementação, Implantação, Testes e Manutenção
- C) Projeto, Análise, Implementação, Implantação, Testes e Manutenção
- D) Projeto, Análise, Implementação, Testes, Implantação e Manutenção

Questões dissertativas:

6) “A crise do Software refere-se a um conjunto de problemas encontrados no desenvolvimento de software”. Quais foram esses problemas? Cite e explique cada um com suas palavras.

A crise do software refere-se à uma série de problemas que sistemas computacionais apresentaram após a década de 70. Causados por estimativas de prazos e de custos extremamente imprecisas pois não havia precauções tomadas na hora de projetar os sistemas, tudo era feito a “toque de caixa”, sem preocupações com qualidade e principalmente, sem preocupações com manutenção. Isso causava uma imensa insatisfação dos clientes para com esses sistemas, pois não havia estudo aprofundado de suas necessidades, o que resultava em um produto de baixíssima qualidade e manutenção quase impossível.

7) Sobre as causas da Crise do Software comente: “O software não se desgasta, mas se deteriora”.

Softwares não sofrem desgaste físico, pois não são coisas materiais. Porém a evolução do hardware, a transição de sistemas operacionais, mudanças de paradigmas de I/O e de UI causam com que esses sistemas tornem-se antiquados, obsoletos. Existe um limite à vida útil de um software, pois a costura de novas atualizações torna o produto cada vez mais instável e a sua substituição cada dia mais próxima.

8) Cite e comente as demais causas da Crise de Software.

As principais causas foram: a confusão geral entre software e hardware, especialmente por parte dos administradores. A ignorância do fato de que, apesar de não ser um produto físico, existe uma deterioração do software no decorrer da sua vida útil. O

problema da capacitação profissional para saber como lidar com algo tão novo quanto o desenvolvimento de softwares, especialmente vindo de gerentes. Também temos a alta resistência às mudanças como um dos fatores primordiais para o agravamento da crise.

9) Cite e comente cada um dos Mitos da Crise do Software.

Alguns dos mitos da crise do software são: A confusão entre hardware e software. Muitos administradores não tinham conhecimento suficiente para entender que ambos precisam andar juntos, achavam que o investimento em hardware de ponta resolveria qualquer problema. Mudanças repentinas no projeto na ilusão da “flexibilidade” de um sistema, sem pensar no custo e dificuldade de todo retrabalho. A ilusão de pensar que a forma final de um software se apresenta na entrega dele pelos desenvolvedores, quando a maioria absoluta do trabalho se encontra após sua entrega, na adequação aos requisitos e pormenores do trabalho e da empresa.

10) Pesquise na internet uma falha famosa de software e comente.

Uma das falhas mais famosas de software foi o bug do milênio, mais conhecido como Y2K. Se tratava de um problema de computação que ocorreu no final do século XX, especificamente na passagem de 31 de dezembro de 1999 para 1º de janeiro de 2000. Muitos computadores usavam apenas dois dígitos para representar o ano, em vez de quatro, para economizar espaço de armazenamento. Porém os desenvolvedores não pensaram nas consequências disso na virada do milênio, onde todas as datas sofreriam com overflow e, portanto, os softwares não teriam como distinguir entre o ano 2000 e o ano 1900, por exemplo.

11) A Engenharia de software pode ser definida como:

“Estabelecimento e uso de sólidos princípios de engenharia para que se possa obter um software economicamente viável que seja confiável e que funcione eficientemente em máquinas reais” [Pressman].

“Aplicação de uma abordagem sistemática, disciplinada e possível de ser medida para o desenvolvimento, operação e manutenção do software”. [IEEE]

Nas definições acima, comente os termos em negrito:

- Princípios da engenharia: uso de modelos abstratos e precisos para melhorar o projeto e desenvolvimento de software.
- Software economicamente viável: sistema capaz de gerar receita ou economizar dinheiro suficiente para justificar seus custos de desenvolvimento, manutenção e operação ao longo do tempo.
- Software confiável: sistema que resolve os problemas necessários sem causar novos problemas no ambiente de implementação.
- Software eficientemente funcional: sistema que atende aos requisitos e necessidades dos usuários de forma eficiente, realiza as tarefas para as quais foi projetado de maneira eficaz e sem erros, sem consumir excessivamente recursos do sistema ou do usuário
- Abordagem sistemática: processo disciplinado e estruturado para a criação de softwares. Envolve uma série de etapas sequenciais que incluem a análise dos

requisitos, o projeto da solução, a implementação do software, os testes de validação e a manutenção contínua.

- Abordagem possível de ser medida: junto da abordagem sistemática, implementação de ferramentas que permitam a medida de qualidade, produtividade e eficiência.

12) Quais os objetivos da Engenharia de Software citados na apresentação? Copie e comente cada um.

O objetivo principal da engenharia de software é auxiliar no processo de produção dos softwares usando das seguintes maneiras:

- Criar softwares de alta qualidade: evitar criação de softwares que não cumpram com seu propósito e sejam abandonados rapidamente, ou de produtos que causam mais problemas do que resolvem.
- Produzir mais rapidamente: aumentar a produtividade evitando retrabalho, partindo já de premissas e de um planejamento sólido.
- Custo reduzido: diminuir o número de alterações e modificações tardias, diminuir a necessidade de manutenção e o tempo de desenvolvimento contribui e muito para a redução de custos.

13) O Ciclo de vida do software é composto por etapas. Cite e explique cada uma dessas etapas.

- Análise: estabelece as bases, premissas e restrições do sistema, verifica-se a viabilidade do mesmo.
- Projeto: documenta-se tudo que for necessário para desenvolvimento do sistema. Levanta-se todos os requisitos e formaliza-se o produto e os processos.
- Implementação: Desenvolvimento, propriamente dito, do sistema, segundo os requisitos levantados.
- Teste: Teste de tudo que foi desenvolvido. Em caso de falhas correções serão realizadas.
- Implantação: Etapa onde o sistema é colocado em uso, em produção.
- Manutenção: Durante sua vida útil o sistema terá de passar por correções e modificações, a manutenção de sistemas pode ser esporádica ou constante.

14) Pesquise, cite e descreva sucintamente as funções de uma ferramenta Case usada na Engenharia de Software.

Ferramentas CASE ((Computer-Aided Software Engineering) são softwares que auxiliam os engenheiros de software a desenvolver e manter outros softwares.

Suas principais funções são:

- Modelagem e análise: criação de modelos visuais para entender o sistema e identificar problemas e oportunidades de melhoria.
- Documentação: criação de documentação para garantir que todos os envolvidos no projeto tenham uma compreensão clara do que está sendo construído.
- Gerenciamento de requisitos: coleta, análise, documentação e rastreamento de requisitos ao longo do ciclo de vida do projeto.
- Gerenciamento de projetos: programação, alocação de recursos, acompanhamento de progresso e gerenciamento de mudanças.

- Desenvolvimento e implementação: criação de código, depuração, teste de unidade e integração de componentes de software.
- Teste e validação: criação, execução e gerenciamento de casos de teste para o sistema de software.