

Exercícios de Aplicação

FUNDAMENTOS DE ENGENHARIA DE SOFTWARE:

1. O que é software? Comente sobre seus componentes e diferencie-o de um programa de computador.

R.: Software é um termo geral que se refere a instruções, dados ou programas de computador que permitem a execução de tarefas específicas em um sistema de computador. Em termos mais amplos, o software é o conjunto de informações processáveis pelo hardware de um computador.

Componentes:

- **Código Fonte (Source Code):** Conjunto de instruções escritas em uma linguagem de programação.
- **Código Executável (Executable Code):** Resultado da compilação do código fonte, que pode ser executado pelo computador.
- **Dados de Configuração e Parâmetros:** Informações necessárias para a execução adequada do software.
- **Documentação:** Manuais, guias e documentação técnica que auxiliam no entendimento e uso do software.
- **Bibliotecas:** Conjuntos de código reutilizável que facilitam o desenvolvimento.

Diferença entre Software e Programa de Computador:

Software é o termo mais amplo que inclui todos os elementos necessários para o funcionamento de um sistema de computador, enquanto programa de computador se refere especificamente à parte do software que contém o código executável.

2. O que é Produto de Software?

R.: Um produto de software é um artefato resultante do processo de engenharia de software. Ele pode ser um programa de computador, uma aplicação, ou qualquer outra entrega específica do desenvolvimento de software. Um produto de software é projetado para atender a requisitos específicos e resolver problemas identificados durante a análise.

3. “Engenheiros de software preocupam se com o desenvolvimento de produtos de software.” Quais são os tipos de produto de software que existem? Dê exemplos.

R.: Alguns dos tipos comuns de produtos de software incluem:

-Sistemas Operacionais:

Exemplo: Microsoft Windows, macOS, Linux.

-Software de Aplicação:

Exemplo: Microsoft Office (Word, Excel, PowerPoint), Adobe Photoshop.

-Software de Desenvolvimento:

Exemplo: Eclipse, Visual Studio, IntelliJ IDEA.

-Software Embarcado:

Exemplo: Firmware de dispositivos eletrônicos, como roteadores, impressoras.

-Software de Tempo Real:

Exemplo: Sistemas de controle em carros, sistemas de controle de tráfego aéreo.

-Software de Banco de Dados:

Exemplo: MySQL, Oracle Database, MongoDB.

-Software de Redes:

Exemplo: Wireshark (analisador de protocolos de rede).

-Software de Entretenimento:

Exemplo: Jogos eletrônicos, aplicativos de streaming de vídeo.

-Software de Segurança:

Exemplo: Antivírus, firewalls.

-Sistemas de Informação Empresarial (ERP):

Exemplo: SAP, Oracle ERP.

-Sistemas de Gerenciamento de Conteúdo (CMS):

Exemplo: WordPress, Drupal.

-Aplicativos Móveis:

Exemplo: WhatsApp, Instagram, Spotify.

-Software de Simulação:

Exemplo: Simuladores de voo, simulação de processos industriais.

-Software Educacional:

Exemplo: Plataformas de e-learning, softwares de tutoria.

4. Fale da crise de software.

R.:A "crise de software" refere-se a um período na história do desenvolvimento de software em que surgiram desafios significativos e problemas persistentes na criação de sistemas de software. Essa crise foi identificada pela primeira vez em um famoso relatório encomendado pela OTAN em 1968. Aqui estão alguns dos principais aspectos relacionados à crise de software:

- Crescimento na Complexidade:

À medida que os sistemas de software tornavam-se mais complexos, tornava-se cada vez mais difícil gerenciá-los eficientemente. A complexidade crescente resultava em atrasos, custos excessivos e falhas nos projetos de software.

- Fracassos de Projetos:

Muitos projetos de software estavam falhando em atender aos requisitos, sendo entregues fora do prazo ou excedendo o orçamento previsto. Alguns projetos eram abandonados antes da conclusão.

- Falta de Metodologias Adequadas:

Na época, muitos desenvolvedores trabalhavam sem metodologias claras e estruturadas para o desenvolvimento de software. Isso levava a processos caóticos e falta de controle sobre o ciclo de vida do software.

- Escassez de Profissionais Qualificados:

A demanda por profissionais qualificados em desenvolvimento de software excedia a oferta, levando a uma escassez de mão de obra qualificada. Isso contribuía para a baixa qualidade e eficiência nos projetos.

- Mudanças nos Requisitos:

Os requisitos dos projetos frequentemente mudavam durante o desenvolvimento, o que tornava difícil manter a consistência e a estabilidade nas fases de projeto.

- Falta de Ferramentas Adequadas:

A falta de ferramentas avançadas e eficazes para auxiliar no desenvolvimento, teste e manutenção de software também contribuiu para os desafios enfrentados.

5. O que é Engenharia de Software?

R.: Engenharia de Software é uma disciplina que se dedica à aplicação de princípios, práticas e técnicas sistemáticas na concepção, desenvolvimento, operação e manutenção de sistemas de software. Ela envolve a aplicação de métodos de engenharia para garantir que o desenvolvimento de software seja conduzido de maneira sistemática, eficiente e de alta qualidade.

6. O que torna um projecto de software diferente de projectos de outras áreas de engenharia?

R.: Os projetos de software diferem de projetos em outras áreas de engenharia devido a diversas características intrínsecas à natureza do desenvolvimento de software. Aqui estão algumas das principais distinções:

- Imaterialidade e Intangibilidade:

Software: O produto final é intangível e imaterial, consistindo em código, algoritmos e dados.

Outras Engenharias: Produtos físicos tangíveis, como edifícios, pontes ou dispositivos eletrônicos.

- Mudanças Dinâmicas:

Software: Requisitos podem mudar ao longo do desenvolvimento de maneira mais rápida e dinâmica.

Outras Engenharias: Requisitos em engenharia tradicional geralmente são mais estáveis.

- Ciclo de Vida Contínuo:

Software: O ciclo de vida do software muitas vezes é contínuo, com atualizações e manutenção frequentes após a implementação inicial.

Outras Engenharias: Muitos projetos têm um ciclo de vida mais linear e definido.

- Flexibilidade e Adaptabilidade:

Software: Maior facilidade de realizar mudanças e adaptações durante o desenvolvimento.

Outras Engenharias: Mudanças podem ser mais desafiadoras e custosas.

- Escopo Mal Definido Inicialmente:

Software: Muitas vezes, o escopo do projeto pode ser desafiador de definir completamente no início.

Outras Engenharias: O escopo em muitos projetos é mais concreto desde o início.

- Custo de Reprodução Baixo:

Software: A reprodução do software é geralmente mais fácil e de menor custo.

Outras Engenharias: Reproduzir produtos físicos pode ser mais complexo e caro.

- Heterogeneidade Tecnológica:

Software: Diversidade de tecnologias, linguagens e frameworks.

Outras Engenharias: Uso de materiais e processos mais padronizados.

- Colaboração e Comunicação Intensiva:

Software: A colaboração e a comunicação entre membros da equipe são fundamentais.

Outras Engenharias: Dependendo do projeto, a colaboração pode ser menos intensiva.

7. Por que motivos a engenharia de software é importante?

R.: A Engenharia de Software é crucial para lidar com a complexidade crescente do desenvolvimento de software, garantindo que os projetos sejam entregues de forma eficaz, cumprindo requisitos, prazos e orçamentos.

A importância da Engenharia de Software está intrinsecamente ligada à sua capacidade de criar soluções inovadoras, promover a eficiência, facilitar a interconexão global e impulsionar o progresso em muitos setores da sociedade moderna.

8.O que são problemas fundamentais e acidentais da Engenharia de Software?

R: Na Engenharia de Software, problemas fundamentais referem-se a questões intrínsecas à disciplina e estão relacionados à natureza da construção de software, enquanto problemas acidentais são resultantes de circunstâncias externas ou contingências e são geralmente relacionados a restrições de tempo, recursos ou outras condições específicas do projeto.

9. Sobre problemas fundamentais e essenciais da Engenharia de Software, comente sobre o problema da complexidade, da alterabilidade, da conformidade e da invisibilidade.

R: Complexidade: O software tende a se tornar mais complexo com novos requisitos e evolução, exigindo gerenciamento cuidadoso para garantir qualidade e manutenibilidade.

Alterabilidade: O software deve ser projectado e construído de forma a permitir mudanças sem efeitos colaterais indesejados, adaptando-se aos requisitos em constante evolução.

Conformidade: O software deve atender aos requisitos especificados e às expectativas dos usuários, bem como estar em conformidade com padrões e regulamentos relevantes.

Invisibilidade: Problemas como bugs podem permanecer ocultos até o uso, exigindo métodos de teste e abordagens proativas para garantir a qualidade.

10. O que entende por Gestão de Expectativas?

R: Gestão de Expectativas é o processo de gerenciar e alinhar as expectativas dos stakeholders em relação ao projeto de software. Isso envolve comunicar de forma clara e realista o que pode e o que não pode ser alcançado dentro dos recursos e prazos disponíveis, além de garantir que as expectativas estejam alinhadas com os objetivos e limitações do projeto.

11. Mencione os atributos de um bom software.

R: Confiabilidade: O software deve operar consistentemente e sem falhas sob uma variedade de condições.

Eficiência: O software deve executar suas funções de forma rápida e com uso mínimo de recursos.

Manutenibilidade: O software deve ser facilmente mantido, adaptado e corrigido.

Usabilidade: O software deve ser fácil de aprender e usar, com uma interface intuitiva e amigável.

Portabilidade: O software deve ser capaz de ser executado em diferentes plataformas e ambientes sem modificações significativas.

Segurança: O software deve proteger os dados e garantir que o acesso seja controlado e autorizado adequadamente.

Escalabilidade: O software deve ser capaz de lidar com um aumento na demanda sem perder desempenho ou funcionalidade.

Eficiência de custos: O software deve fornecer um bom retorno sobre o investimento, tanto em termos de desenvolvimento quanto de operação.

12. Apresente as diferenças entre Engenharia de Software e Ciência da Computação.

R: Os engenheiros de software lidam com a criação, implementação e manutenção de software.

Ciência da Computação: é mais teórica e tem como objetivo principal investigar os princípios e os fundamentos por trás dos sistemas computacionais

13. Mencione e descreva as disciplinas relacionadas à Engenharia de Software.

R: • Computação em Nuvem.

- Técnicas de Inteligência Artificial.
- Arquitetura de Software.
- Infraestrutura Ágil.
- Segurança em Engenharia de Software.
- Desenvolvimento Mobile.
- Testes e Manutenção de Software.

14. Defina o conceito de sistema.

R: Sistema é um conjunto integrado de componentes regularmente inter-relacionados e interdependentes criados para realizar um objetivo definido, com relações definidas e mantidas

entre seus componentes e cuja produção e operação como um todo é melhor que a simples soma de seus componentes.

15. Defina os seguintes conceitos: dado, informação e conhecimento.

R: **Dados** são uma coleção de valores discretos que transmitem informações, descrevendo quantidade, qualidade, fatos, estatísticas, outras unidades básicas de significado, ou simplesmente sequências de símbolos que podem ser posteriormente interpretados. Um dado é um valor individual em uma coleção de dados.

Informação é um conhecimento inscrito (gravado) sob a forma escrita (impressa ou numérica), oral ou audiovisual. É resultante do processamento, manipulação e organização de dados, de tal forma que represente uma modificação (quantitativa ou qualitativa) no conhecimento do sistema (humano, animal ou máquina) que a recebe.

O **conhecimento** é um conjunto de informação armazenada por intermédio da experiência ou da aprendizagem (a posteriori), o através da introspecção (a priori).

16. Defina sistemas de informação.

R: Sistema de informação é um conjunto formado por pessoas, software, hardware, procedimentos e dados. O sistema de informação é responsável por difundir as informações através da organização.

17. Quais as três atividades fundamentais num sistema de informação?

R: Entrada de Dados, Processamento de Dados e Saída de Informação.

18. Diga o que ocorre em cada uma das atividades fundamentais dum sistema de informação.

R: **Entrada de Dados:** Esta atividade envolve a coleta e entrada de dados brutos no sistema de informação. Os dados podem ser inseridos manualmente pelos usuários, capturados automaticamente de dispositivos eletrônicos ou importados de outras fontes de dados externas

Processamento de Dados: Após a entrada dos dados, ocorre o processamento, no qual os dados são manipulados e transformados em informações significativas. Isso pode envolver cálculos,

classificações, comparações, análises estatísticas ou qualquer outra operação que agregue valor aos dados

Saída de Informação: Uma vez que os dados tenham sido processados, as informações resultantes são apresentadas aos usuários de forma compreensível e útil. Isso pode incluir a geração de relatórios, gráficos, tabelas, dashboards ou qualquer outra forma de representação visual ou textual dos dados

19.Explique o conceito de feedback?

R: O Feedback é uma resposta ou retorno que uma pessoa, sistema ou processo recebe sobre seu desempenho ou resultado, com o propósito de melhorar ou ajustar seu comportamento, desempenho ou resultados futuros.

20.Qual é a utilidade da informação produzida por um SI para uma organização?

R: Suporte à Tomada de Decisão: Fornece dados e análises para decisões mais informadas.

Melhoria da Eficiência Operacional: Automatiza processos e otimiza fluxos de trabalho.

Melhoria da Qualidade: Inclui controle de qualidade e feedback para melhorias contínuas.

Melhoria do Atendimento ao Cliente: Personaliza produtos e serviços para aumentar a satisfação do cliente.

Vantagem Competitiva: Identifica tendências e ajusta estratégias para superar concorrentes.

Facilitação da Comunicação e Colaboração: Permite uma comunicação eficaz e colaboração entre equipes.

Conformidade Regulatória: Ajuda a cumprir regulamentos e normas, evitando penalidades.

21.O que são sistemas técnicos?

R: Sistemas técnicos são sistemas compostos por elementos físicos e tecnológicos que trabalham juntos para realizar uma função específica ou alcançar um objetivo predefinido.

22.O que são sistemas sociotécnicos?