

## Engenharia de Software

1. **Qual é a definição de Engenharia de Software apresentada pelo IEEE em 1993 e qual seu objetivo principal?**

A Engenharia de Software é a "aplicação de uma abordagem sistemática, disciplinada e quantitativa para o desenvolvimento, operação e manutenção de software". Seu objetivo é aplicar os princípios e conceitos da área na implementação de componentes de software como parte de Sistemas de Informação, iniciando pela modelagem de requisitos.

2. **O que foi a "Crise do Software" e quais fatores a desencadearam a partir da década de 1960?**

A "Crise do Software" foi um período em que a capacidade de desenvolver software não acompanhava a evolução e a demanda gerada pelos avanços do hardware. Os principais fatores foram a redução dos custos de hardware, o aumento da velocidade e capacidade de armazenamento dos computadores, e a consequente incapacidade das equipes de desenvolvimento de atender à crescente demanda por produção de software.

3. **Quais são os três tipos de produtos de software discutidos na aula? Dê uma breve definição de cada um.**

- **Produtos Genéricos ("Off-the-shelf"):** Sistemas autônomos desenvolvidos para serem vendidos a qualquer cliente interessado, com especificações controladas pelo desenvolvedor (ex: Microsoft Word, Excel).
- **Produtos Sob Encomenda ("Bespoke"):** Sistemas encomendados por um cliente específico para atender às suas necessidades particulares, com especificações definidas pelo cliente (ex: sistema de controle de tráfego aéreo).

4. **Descreva as principais características, vantagens e desvantagens do Modelo de Processo em Cascata (Waterfall).**

- **Características:** Processo sequencial com fases distintas e bem definidas (análise de requisitos, projeto, implementação, teste, manutenção).
- **Vantagens:** Simplicidade de gerenciamento, clareza nas fases e entregáveis, e abordagem estruturada e documentada.
- **Desvantagens:** Inflexibilidade a mudanças, documentação intensiva, feedback tardio do cliente e separação rígida entre as fases.

5. **O que é o Desenvolvimento Incremental e quais são suas principais vantagens em relação ao modelo em cascata?**

O Desenvolvimento Incremental é uma abordagem que intercala as atividades de especificação, desenvolvimento e validação, entregando o sistema em uma série de versões (incrementos) funcionais. Suas principais vantagens são a redução do custo de mudanças, o feedback antecipado do cliente, a entrega mais rápida de software útil e a redução de riscos.

6. **Quais são as quatro atividades fundamentais presentes em todos os processos de software, segundo Sommerville?**

1. **Especificação de Software:** Definir os serviços e restrições do sistema.

2. **Projeto e Implementação de Software:** Converter a especificação em um sistema executável.
  3. **Validação de Software:** Verificar se o sistema atende à especificação e às necessidades do cliente.
  4. **Evolução de Software (Manutenção):** Modificar o software após a entrega para corrigir defeitos ou adaptar a novas necessidades.
7. **Explique as quatro fases do Rational Unified Process (RUP).**
0. **Concepção:** Estabelecer o "business case" para o sistema.
  1. **Elaboração:** Compreender o problema, estabelecer a arquitetura, planejar o projeto e identificar os principais riscos.
  2. **Construção:** Envolve o projeto, a programação e os testes do sistema.
  3. **Transição:** Transferir o sistema da equipe de desenvolvimento para a comunidade de usuários.
8. **Cite os quatro valores fundamentais do Manifesto Ágil.**
0. **Indivíduos e interações** mais que processos e ferramentas.
  1. **Software em funcionamento** mais que documentação abrangente.
  2. **Colaboração com o cliente** mais que negociação de contratos.
  3. **Responder a mudanças** mais que seguir um plano.
9. **No Scrum, quais são os três papéis principais e quais são suas responsabilidades?**
- **Product Owner:** Representa os interesses do negócio e é responsável por definir e priorizar as funcionalidades do produto no Product Backlog.
  - **Scrum Master:** Facilita o processo Scrum, ajuda a equipe a seguir as práticas corretas e remove impedimentos.
  - **Equipe de Desenvolvimento (Equipe Dev):** Grupo auto-organizado responsável por analisar, projetar, implementar e testar o produto.
10. **Quais são os eventos (cerimônias) "time-boxed" do Scrum?**
- **Reunião de Planejamento da Sprint (Sprint Planning):** Define o trabalho a ser realizado na sprint.
  - **Daily Scrum:** Reunião diária de 15 minutos para sincronizar o progresso.
  - **Revisão da Sprint (Sprint Review):** A equipe apresenta o que foi entregue e obtém feedback.
  - **Retrospectiva da Sprint (Sprint Retrospective):** A equipe reflete sobre o processo de trabalho para identificar melhorias.
11. **Diferencie Requisitos Funcionais de Requisitos Não Funcionais.**
- **Requisitos Funcionais (RFs):** Descrevem o que o sistema deve fazer, seus serviços e como ele deve reagir a entradas específicas.
  - **Requisitos Não Funcionais (RNFs):** São restrições sobre os serviços ou funções oferecidos pelo sistema, como desempenho, segurança e usabilidade.
12. **O que é um Modelo de Contexto e qual seu principal propósito na modelagem de sistemas?**

Um Modelo de Contexto define as fronteiras do sistema, mostrando o que está dentro e o que está fora dele. Seu propósito é ilustrar como o sistema se encaixa em seu ambiente operacional, incluindo outros sistemas com os quais ele interage.

**13. Em um Diagrama de Casos de Uso UML, o que representam os relacionamentos <<include>> e <<extend>>?**

- **<<include>>**: Indica que um caso de uso base inclui explicitamente a funcionalidade de outro caso de uso. É uma dependência obrigatória.
- **<<extend>>**: Indica que um caso de uso extensor adiciona um comportamento opcional a um caso de uso base em um ponto de extensão específico.

**14. Diferencie Agregação de Composição em Diagramas de Classes UML.**

- **Agregação (losango vazado)**: Representa uma relação "tem-um" onde a parte pode existir independentemente do todo. Se o todo for destruído, a parte continua a existir.
- **Composição (losango preenchido)**: É uma forma mais forte de agregação que representa uma relação "parte-de", onde a parte não pode existir independentemente do todo. Se o todo for destruído, a parte também é.

**15. Explique o padrão de arquitetura Modelo-Visão-Controlador (MVC).**

O padrão MVC separa a aplicação em três componentes interconectados:

- **Modelo (Model)**: Gerencia os dados e a lógica de negócio.
- **Visão (View)**: Renderiza a apresentação dos dados para o usuário.
- **Controlador (Controller)**: Processa a entrada do usuário, interage com o Modelo e seleciona a Visão apropriada para exibir a resposta.

## **Qualidade e Teste de Software**

**1. O que é Qualidade de Software, de acordo com a definição formal apresentada?**

A qualidade de software refere-se ao grau em que um sistema, componente ou processo de software atende aos requisitos especificados e às necessidades ou expectativas do cliente ou usuário.

**2. A norma ISO/IEC 25010 define oito atributos de qualidade. Cite e descreva brevemente quatro deles.**

- **Adequação Funcional**: A capacidade do produto de prover funcionalidades que atendam às necessidades declaradas e implícitas.
- **Eficiência de Desempenho**: O desempenho relativo à quantidade de recursos usados (tempo de resposta, uso de CPU/memória).
- **Confiabilidade**: A capacidade do sistema de desempenhar suas funções sob condições estabelecidas por um período de tempo definido.
- **Usabilidade**: A capacidade do produto de ser entendido, aprendido, usado e atraente para o usuário.
- *(Outras possíveis: Compatibilidade, Segurança, Manutenibilidade, Portabilidade)*

**3. Qual a diferença entre Qualidade de Processo e Qualidade de Produto?**

- **Qualidade de Produto:** Refere-se às características intrínsecas do software (os 8 atributos da ISO 25010). É o que o usuário final experiencia diretamente.
  - **Qualidade de Processo:** Refere-se à eficácia e eficiência das atividades, métodos e ferramentas usadas na criação e manutenção do software. Um bom processo visa prevenir defeitos.
4. **Diferencie os papéis de Garantia da Qualidade (QA), Controle de Qualidade (QC) e Teste.**
- **QA (Garantia da Qualidade):** É um conjunto de atividades **proativas** focadas em **prevenir** defeitos, definindo processos e padrões.
  - **QC (Controle de Qualidade):** É um conjunto de atividades **reativas** focadas em **identificar** defeitos no produto antes da entrega, como a execução de testes e revisões de código.
  - **Teste:** É uma atividade específica dentro do QC que consiste em operar o sistema para encontrar evidências de falhas.
5. **Qual a diferença fundamental entre Verificação e Validação de software?**
- **Verificação:** Responde à pergunta "Estamos construindo o produto **corretamente**?". Avalia se o produto atende às especificações e padrões definidos.
  - **Validação:** Responde à pergunta "Estamos construindo o produto **certo**?". Avalia se o produto final atende às necessidades e expectativas do usuário em seu ambiente real.
6. **O que é o Ciclo de Vida do Teste de Software (STLC)?**
- O STLC (Software Testing Life Cycle) é uma sequência de atividades, análoga ao SDLC, que define as fases do processo de teste para garantir que os objetivos de qualidade sejam alcançados.
7. **Cite as sete fases do STLC (Software Testing Life Cycle).**
1. Planejamento do Teste
  2. Análise e Desenho dos Testes
  3. Implementação dos Testes
  4. Configuração do Ambiente de Teste
  5. Execução dos Testes
  6. Avaliação dos Critérios de Saída e Relatórios
  7. Atividades de Fechamento do Teste
8. **Qual é o principal entregável da fase de "Planejamento do Teste" no STLC?**
- O principal entregável é o **Plano de Teste**, que define a estratégia, objetivos, escopo, recursos, estimativas e riscos do processo de teste.
9. **O que ocorre durante a fase de "Análise e Desenho dos Testes" no STLC?**
- Nesta fase, a equipe de teste analisa os requisitos para identificar as condições de teste e desenha os casos de teste de alto nível, além de identificar os dados de teste necessários.

10. Qual a diferença entre a fase de "Implementação dos Testes" e a de "Execução dos Testes"?

- **Implementação:** É a fase onde os casos de teste de alto nível são transformados em casos de teste concretos e detalhados, incluindo a criação de scripts de automação e a obtenção dos dados de teste.
- **Execução:** É a fase onde os testes preparados são efetivamente executados no ambiente configurado, os resultados são registrados e os defeitos são reportados.

11. De acordo com o atributo de "Segurança" da ISO 25010, quais são algumas de suas subcaracterísticas?

As subcaracterísticas incluem Confidencialidade, Integridade, Não repúdio/Responsabilização e Autenticidade.

12. O que significa a "Manutenibilidade" como um atributo de qualidade de software?

É a capacidade do produto de ser modificado de forma eficaz e eficiente, seja para correções, melhorias ou adaptações, envolvendo aspectos como modularidade, reusabilidade e testabilidade.

13. Explique a "Portabilidade" como um atributo de qualidade e dê um exemplo.

É a capacidade de um produto de software ser transferido de um ambiente para outro de forma eficaz e eficiente. Por exemplo, uma API Java Spring que pode ser implantada facilmente em um servidor Linux local ou em um contêiner Docker na nuvem com mínimas alterações.

14. Use uma analogia para explicar a relação entre QA, QC e Teste.

Em uma fábrica de carros (analogia):

- **QA** é quem constrói a "fábrica": define os processos, padrões e ferramentas para garantir que bons carros sejam produzidos.
- **QC** é o inspetor na linha de montagem: ele verifica o carro (software) para encontrar defeitos antes que chegue ao cliente, seguindo as diretrizes do QA.
- **Teste** é a principal ferramenta do inspetor de QC: é o ato de ligar o motor, verificar os freios, a pintura, etc.

15. Quando ocorre a Verificação e quando ocorre a Validação no ciclo de desenvolvimento de software?

- **Verificação:** Ocorre continuamente ao longo de todo o ciclo de desenvolvimento (ex: revisões de código, testes unitários).
- **Validação:** Ocorre principalmente no final das iterações ou do ciclo (ex: testes de sistema, testes de aceitação do usuário), buscando feedback do cliente o mais cedo possível.