

CONCURSO PÚBLICO – 05/2025
Área de Conhecimento: CESMO - Engenharia de Software
PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 1: Fale sobre o Processo de Levantamento de Requisitos definindo e apresentando técnicas que podem ser utilizadas nesse processo. Apresente e detalhe os tipos de requisitos existentes.

Definição

Segundo Pressman (Capítulo 7 do Livro de Engenharia de Software de Roger Pressman) o **processo de levantamento de requisitos** é uma etapa fundamental da Engenharia de Software, responsável por identificar e documentar as necessidades e expectativas dos stakeholders em relação ao sistema a ser desenvolvido. Segundo Sommerville e Pressman, essa fase visa compreender **o que o software deve fazer e quais restrições devem ser atendidas**, servindo como base para todo o projeto. Um levantamento mal conduzido pode resultar em falhas graves, como retrabalho, aumento de custos e insatisfação do cliente.

Técnicas Utilizadas

Diversas técnicas podem ser aplicadas para obter requisitos, entre elas:

- **Entrevistas:** conversas estruturadas ou semiestruturadas com usuários e especialistas.
- **Questionários:** coleta de informações de um grande número de pessoas.
- **Observação direta:** acompanhamento das atividades do usuário para entender processos reais.
- **Workshops e reuniões de brainstorming:** interação colaborativa para eliciar requisitos.
- **Prototipação:** criação de modelos iniciais do sistema para validar expectativas.
- **Análise de documentos:** estudo de normas, relatórios e sistemas existentes.

Tipos de Requisitos

Os requisitos podem ser classificados em:

- **Requisitos Funcionais:** descrevem as funcionalidades que o sistema deve oferecer (ex.: “O sistema deve permitir cadastro de clientes”).
- **Requisitos Não Funcionais:** especificam restrições ou características de qualidade, como desempenho, segurança e usabilidade (ex.: “O sistema deve responder em até 2 segundos”).
- **Requisitos de Domínio:** refletem regras específicas do contexto ou área de negócio (ex.: cálculos tributários em um sistema fiscal).

Sommerville enfatiza que requisitos devem ser **claros, completos e verificáveis**, enquanto Pressman destaca a importância de **comunicação eficaz com o cliente e documentação adequada** para evitar ambiguidades.

Membros da Banca:

Avaliador 1 (nome e assinatura)
assinatura

Avaliador 2 (nome e

Avaliador 3 (nome e assinatura)
(nome e assinatura)

Presidente da Banca

CONCURSO PÚBLICO – 05/2025
Área de Conhecimento: CESMO - Engenharia de Software
PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 2: O que é escopo do software e qual a sua importância no processo de desenvolvimento de sistemas?

Segundo Pressman (Capítulo 25 do Livro de Engenharia de Software de Roger Pressman), o **escopo do software** é a definição clara dos limites e objetivos do sistema a ser desenvolvido. Ele descreve o que será entregue, quais **funcionalidades** estarão presentes, quais **restrições** devem ser respeitadas e quais **interfaces** serão necessárias. Segundo Sommerville e Pressman, o escopo é essencial para estabelecer uma compreensão comum entre cliente e equipe de desenvolvimento, evitando ambiguidades e mudanças descontroladas durante o projeto.

A importância do escopo no processo de desenvolvimento é significativa. Primeiramente, ele serve como base para **estimativas de custo, prazo e recursos**, pois define o tamanho e a complexidade do projeto. Além disso, um escopo bem definido reduz riscos, facilita o planejamento e contribui para a **qualidade do produto**, garantindo que os requisitos do cliente sejam atendidos. Pressman destaca que a falta de definição clara do escopo é uma das principais causas de falhas em projetos de software, levando a atrasos, aumento de custos e insatisfação do cliente.

Em síntese, o escopo do software é um elemento estratégico que orienta todo o processo de desenvolvimento, sendo indispensável para o sucesso do projeto.

Membros da Banca:

Avaliador 1 (nome e assinatura)
assinatura)

Avaliador 2 (nome e

Avaliador 3 (nome e assinatura)
(nome e assinatura)

Presidente da Banca

CONCURSO PÚBLICO – 05/2025
Área de Conhecimento: CESMO - Engenharia de Software
PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 3: Um elemento-chave de qualquer processo de engenharia é a medição. Medições podem ser aplicadas ao processo de software com a intenção de melhorá-lo continuamente. Elas podem ser usadas durante um projeto de software para ajudar nas estimativas, no controle de qualidade, na produtividade e no controle de projeto. Neste contexto conceitue e descreva métricas de software.

De acordo com Sommerville e Pressman (Capítulo 23 do Livro de Engenharia de Software de Roger Pressman), a medição é um elemento-chave em qualquer processo de engenharia, e na Engenharia de Software ela se concretiza por meio das **métricas de software**. Essas métricas são instrumentos quantitativos que permitem avaliar atributos do **produto**, do **processo** e do **projeto**, com o objetivo de melhorar continuamente a qualidade, a produtividade e a previsibilidade do desenvolvimento.

As métricas podem ser classificadas em três categorias principais:

- **Métricas de Produto:** Avaliam características do software, como tamanho, complexidade e qualidade.
Exemplos práticos:
 - **Linhas de Código (LOC):** mede o tamanho do software pelo número de linhas de código-fonte.
 - **Complexidade Ciclomática:** indica a complexidade lógica do programa, baseada no número de caminhos independentes.
 - **Defect Density (Densidade de Defeitos):** calcula a quantidade de defeitos por unidade de tamanho (ex.: defeitos por mil linhas de código).
- **Métricas de Processo:** Avaliam a eficiência do processo de desenvolvimento.
Exemplos práticos:
 - **Tempo de Ciclo:** mede o tempo necessário para concluir uma funcionalidade.
 - **Taxa de Retrabalho:** indica o percentual de esforço gasto corrigindo erros ou ajustando requisitos.
- **Métricas de Projeto:** Auxiliam no planejamento e acompanhamento do projeto.
Exemplos práticos:
 - **Function Points (Pontos de Função):** mede o tamanho funcional do software com base nas entradas, saídas e consultas, independente da linguagem de programação.
 - **Esforço (Pessoa-hora):** estima o trabalho necessário para concluir o projeto.
 - **Cumprimento de Cronograma:** compara o progresso real com o planejado.

Pressman ressalta que métricas devem ser usadas para **tomada de decisão**, não para punição, e que sua coleta deve ser consistente e interpretada no contexto do projeto. Sommerville complementa que métricas são fundamentais para **melhoria contínua**, pois permitem identificar gargalos, prever riscos e ajustar processos.

Em síntese, métricas de software são ferramentas estratégicas que transformam dados em informações úteis para garantir qualidade, reduzir custos e aumentar a previsibilidade no desenvolvimento de sistemas.

Membros da Banca:

Avaliador 1 (nome e assinatura)
assinatura)

Avaliador 2 (nome e

Avaliador 3 (nome e assinatura)
(nome e assinatura)

Presidente da Banca

CONCURSO PÚBLICO – 05/2025
Área de Conhecimento: CESMO - Engenharia de Software
PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 4: No contexto de Engenharia de Software, descreva processos de software e modelos de processo de software! Deseja-se também uma comparação entre os modelos de software.

Baseado em Pressman (Capítulo 21 do Livro de Engenharia de Software de Roger Pressman), um **processo de software** é um conjunto estruturado de atividades que orienta a especificação, desenvolvimento, validação e manutenção de sistemas. Ele define **como** o trabalho será realizado, quais produtos serão gerados e quais papéis estarão envolvidos. Ambos os autores destacam que não existe um único processo ideal, mas sim diferentes abordagens que devem ser escolhidas conforme o contexto do projeto.

Os **modelos de processo de software** são abstrações que organizam essas atividades de formas distintas. Sommerville enfatiza modelos como **Cascata**, **Incremental**, **Espiral** e **Ágil**, enquanto Pressman também inclui **Prototipação** e **Modelos Evolutivos**.

- O **Modelo Cascata** segue uma sequência linear, adequada para projetos com requisitos estáveis, mas pouco flexível a mudanças.
- O **Modelo Incremental** entrega o sistema em partes funcionais, permitindo feedback antecipado e redução de riscos.
- O **Modelo Espiral**, proposto por Boehm e amplamente discutido por Pressman, combina desenvolvimento iterativo com análise de riscos, sendo indicado para projetos complexos.
- Os **Modelos Ágeis**, como Scrum e XP, priorizam colaboração, adaptação e entregas rápidas, conforme destacado por Sommerville, sendo ideais para ambientes dinâmicos.

A **comparação entre os modelos** evidencia que o Cascata é simples, mas rígido; o Incremental e o Ágil favorecem mudanças e interação com o cliente; enquanto o Espiral é robusto para projetos críticos, embora mais caro e complexo. Pressman reforça que a escolha do modelo deve considerar fatores como tamanho do projeto, riscos, requisitos e equipe disponível.

Não existe um processo universal: a seleção do modelo deve ser estratégica, visando equilibrar custo, risco, flexibilidade e qualidade do produto.

Membros da Banca:

Avaliador 1 (nome e assinatura)
assinatura)

Avaliador 2 (nome e

Avaliador 3 (nome e assinatura)
(nome e assinatura)

Presidente da Banca

CONCURSO PÚBLICO – 05/2025
Área de Conhecimento: CESMO - Engenharia de Software
PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 5: Na Engenharia de Software, a gestão de riscos é uma prática essencial para garantir o sucesso do projeto, pois permite identificar, analisar e mitigar fatores que podem comprometer prazos, custos e qualidade. Explique o que é gestão de riscos, descreva suas etapas fundamentais e apresente exemplos de riscos comuns em projetos de software.

Gestão de riscos é o processo sistemático de identificar, analisar, priorizar e monitorar riscos que podem afetar o projeto.

Etapas segundo Pressman (Capítulo 26 do Livro de Engenharia de Software de Roger Pressman):

- **Identificação dos riscos** (técnicos, cronograma, custos, requisitos, pessoal).
- **Análise qualitativa e quantitativa** (probabilidade e impacto).
- **Priorização** (classificação dos riscos mais críticos).
- **Planejamento de mitigação** (ações preventivas e contingências).
- **Monitoramento contínuo** (revisão durante todo o ciclo de vida).

Exemplos práticos:

- Risco técnico: uso de tecnologia nova e não testada.
- Risco de cronograma: atraso na entrega de componentes.
- Risco de requisitos: mudanças frequentes solicitadas pelo cliente.
- Risco de pessoal: falta de profissionais qualificados.

Membros da Banca:

Avaliador 1 (nome e assinatura)
assinatura

Avaliador 2 (nome e

Avaliador 3 (nome e assinatura)
(nome e assinatura)

Presidente da Banca



Assinaturas do documento



Código para verificação: **H59TN12P**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:

 **JULIBIO DAVID ARDIGO** (CPF: 578.XXX.009-XX) em 01/12/2025 às 15:00:56
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:36:41 e válido até 30/03/2118 - 12:36:41.
(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTIwMjJfMDAwNDg3MjFfNDg3NTJfMjAyNV9INTIUTjEyUA==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00048721/2025** e o código **H59TN12P** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.