

FACOM - Faculdade de Computação - UFU

GESTÃO DA QUALIDADE DE SOFTWARE - 1a. PROVA - B

Nome do aluno:

1. Por que linguagens como HTML, SQL, Python, dentre outras são adotadas como padrão técnico?

Essas linguagens são adotadas pela interoperabilidade universal, estabilidade normativa (ISO/W3C) e pela redução de custos (TCO) proporcionada por seus vastos ecossistemas e farta mão de obra qualificada ([TAVARES et al., 2025](#))

2. Quais foram os principais fatores de mercado e tecnologia que motivaram a evolução do SWEBOK para a versão 4.0?

A evolução foi motivada pela necessidade de integrar metodologias ágeis/DevOps, o avanço de tecnologias disruptivas como Inteligência Artificial e Nuvem, e a exigência de maior foco em segurança e governança de dados.

([WASHIZAKI, 2024](#))

3. Qual o principal fator de fracasso em projetos de software?

O principal fator de fracasso é a má gestão de requisitos, manifestada por requisitos incompletos, flutuantes ou pela falta de envolvimento do usuário, o que gera produtos que não atendem às necessidades do negócio.

([ALVES et al., 2021](#))

4. O que é um Requisito Não Funcional negligenciado?

É um atributo de qualidade (como segurança ou desempenho) que, embora crítico para a operação, é ignorado durante o desenvolvimento. Resulta em sistemas funcionalmente corretos, mas inviáveis na produção.

([WIEGERS; BEATTY, 2013](#))

5. Cite uma situação onde não é interessante aplicar o TDD?

O TDD não é recomendado em prototipagem exploratória ou prova de conceito (PoC), onde o foco é validar a viabilidade de uma ideia rapidamente e os requisitos ainda são incertos ou voláteis demais para testes estáveis.

(BECK, 2003)

6. Qual a principal diferença entre os testes de Caixa Preta e Caixa Branca em relação ao conhecimento que o testador tem sobre o código-fonte?

No teste de Caixa Preta, o testador não possui conhecimento do código-fonte, focando apenas em entradas e saídas; já no de Caixa Branca, o testador tem acesso total à estrutura interna, lógica e implementação do software.

(MYERS, 2006)

7. Como as LLMs auxiliam os profissionais na execução dos testes de regressão?

LLMs auxiliam na automação da geração de scripts, na identificação de casos de teste impactados por mudanças no código e na análise rápida de logs de falhas, acelerando o ciclo de feedback da regressão.

(WANG et al., 2024)

8. O que caracteriza o *scope creep* e por que ele é prejudicial ao desenvolvimento de um projeto de software?

O scope creep caracteriza-se pela expansão descontrolada do escopo sem ajustes em prazo ou orçamento, resultando em atrasos, estouro de custos e queda na qualidade devido à sobrecarga da equipe.

(TELES et al., 2022)

9. O que significa "congelar" os artefatos em uma *baseline* e por que esse processo é fundamental para garantir a rastreabilidade e a estabilidade do código?

Significa estabelecer um ponto de controle imutável para versões específicas de requisitos e código, garantindo que mudanças futuras sejam auditáveis e que o sistema possa ser reconstruído exatamente como estava em um momento aprovado.

(PRESSMAN et al., 1995)

10. Qual é a função do CCB (Comitê de Controle de Configuração) na manutenção de uma baseline e por que a intervenção humana é necessária mesmo com o uso de ferramentas de versionamento como o Git?

O CCB avalia o impacto técnico e financeiro de mudanças na baseline; a intervenção humana é necessária porque ferramentas como o Git apenas automatizam versões, mas não julgam o valor de negócios ou prioridades estratégicas.

(TAVARES et al., 2025)

FACOM - Faculdade de Computação - UFU

GESTÃO DA QUALIDADE DE SOFTWARE - 1a. PROVA - A

Nome do aluno:

1. O que significa "congelar" os artefatos em uma *baseline* e por que esse processo é fundamental para garantir a rastreabilidade e a estabilidade do código?

Significa estabelecer um ponto de controle imutável para versões específicas de requisitos e código, garantindo que mudanças futuras sejam auditáveis e que o sistema possa ser reconstruído exatamente como estava em um momento aprovado.

(PRESSMAN et al., 1995)

2. Por que podemos afirmar que o TDD é uma forma limpa e previsível de desenvolver códigos limpos?

O TDD garante previsibilidade ao forçar o design de pequenas unidades testáveis antes da implementação, resultando em um código com alta coesão, baixo acoplamento e uma suíte de testes que permite refatorações seguras e contínuas.

(MARTIN, 2019)

3. Qual a principal diferença entre os testes de Caixa Preta e Caixa Branca em relação ao conhecimento que o testador tem sobre o código-fonte?

No teste de Caixa Preta, o testador não possui conhecimento do código-fonte, focando apenas em entradas e saídas; já no de Caixa Branca, o testador tem acesso total à estrutura interna, lógica e implementação do software.

(MYERS, 2006)

4. Como as LLMs auxiliam os profissionais na execução dos testes de regressão?

LLMs auxiliam na automação da geração de scripts, na identificação de casos de teste impactados por mudanças no código e na análise rápida de logs de falhas, acelerando o ciclo de feedback da regressão.

(WANG et al., 2024)

5. O que caracteriza o *scope creep* e por que ele é prejudicial ao desenvolvimento de um projeto de software?

O **scope creep** caracteriza-se pela expansão descontrolada do escopo sem ajustes em prazo ou orçamento, resultando em atrasos, estouro de custos e queda na qualidade devido à sobrecarga da equipe.

(TELES et al., 2022)

6. Qual é a função do CCB (Comitê de Controle de Configuração) na manutenção de uma baseline e por que a intervenção humana é necessária mesmo com o uso de ferramentas de versionamento como o Git?

O **CCB avalia o impacto técnico e financeiro de mudanças na baseline; a intervenção humana é necessária porque ferramentas como o Git apenas automatizam versões, mas não julgam o valor de negócio ou prioridades estratégicas.**

(TAVARES et al., 2025)

7. Quais foram os principais fatores de mercado e tecnologia que motivaram a evolução do SWEBOK para a versão 4.0?

A evolução foi motivada pela necessidade de integrar metodologias ágeis/DevOps, o avanço de tecnologias disruptivas como Inteligência Artificial e Nuvem, e a exigência de maior foco em segurança e governança de dados.

(WASHIZAKI, 2024)

8. Qual o principal fator de fracasso em projetos de software?

O **principal fator de fracasso é a má gestão de requisitos, manifestada por requisitos incompletos, flutuantes ou pela falta de envolvimento do usuário, o que gera produtos que não atendem às necessidades do negócio.**

(ALVES et al., 2021)

9. O que é um Requisito Não Funcional negligenciado?

É um atributo de qualidade (como segurança ou desempenho) que, embora crítico para a operação, é ignorado durante o desenvolvimento. Resulta em sistemas funcionalmente corretos, mas inviáveis na produção.

(WIEGERS; BEATTY, 2013)

10. Por que a Matriz de Rastreabilidade é considerada uma ferramenta técnica essencial para lidar com mudanças durante a codificação?

A **Matriz de Rastreabilidade vincula requisitos a códigos e testes, permitindo identificar instantaneamente quais componentes serão afetados**

por uma mudança, o que evita efeitos colaterais e garante que nenhum requisito seja esquecido ou mal implementado.

(WIEGERS; BEATTY, 2013)

FACOM - Faculdade de Computação - UFU

GESTÃO DA QUALIDADE DE SOFTWARE - Testes - A

Nome do aluno:

1. A disciplina nos testes transforma uma atividade meramente reativa, focada na caça a bugs, a um mecanismo proativo e **PREVENTIVO**.
2. Podemos afirmar que Verificação é garantir que seja entregue o que o cliente realmente queria? **SIM**.
3. QC (Quality Control) caracteriza-se como uma atividade detectiva tentando descobrir falhas? **SIM**.
4. O DDD é uma prática que se aplica bem aos testes unitários? **SIM**.
5. Teste de Componente foca na lógica interna verificando sua interface para assegurar que ele possa interagir corretamente no sistema maior. **SIM**.
6. Qual Teste foca no comportamento do sistema como um todo? **TESTE DE SISTEMA**.
7. Que teste garante a estabilidade e a integridade antes que o produto seja disponibilizado aos usuários? **TESTE DE ACEITAÇÃO**.
8. Qual box oferece uma validação imparcial focada na experiência do usuário e nos requisitos de negócio? **BLACK-BOX TESTING**.
9. Encontrar bugs é o motivo único e principal ao se implementar uma estratégia de testes madura e eficaz? **NÃO**.
10. Testes de regressão não são muito indicados em evolução incremental? **ERRADO**.
11. Manutenção preventiva é acionada após vulnerabilidades que comprometem a operação do sistema comprometendo podendo ser medida pelo MTTR? **ANULADA**.
12. Refatorações para reduzir complexidade e remover duplicações podem ser ações decorrentes da Manutenção preventiva. **SIM**.

FACOM - Faculdade de Computação - UFU

GESTÃO DA QUALIDADE DE SOFTWARE - Testes - B

Nome do aluno:

1. Testes de regressão não são muito indicados em evolução incremental? **ERRADO.**
2. Manutenção preventiva é acionada após vulnerabilidades que comprometem a operação do sistema comprometendo podendo ser medida pelo MTTR? **ANULADA.**
3. Refatorações para reduzir complexidade e remover duplicações podem ser ações decorrentes da Manutenção preventiva. **SIM.**
4. O DDD é uma prática que se aplica bem aos testes unitários? **SIM.**
5. Teste de Componente foca na lógica interna verificando sua interface para assegurar que ele possa interagir corretamente no sistema maior. **SIM.**
6. Qual Teste foca no comportamento do sistema como um todo? **TESTE DE SISTEMA.**
7. Que teste garante a estabilidade e a integridade antes que o produto seja disponibilizado aos usuários? **TESTE DE ACEITAÇÃO.**
8. A disciplina nos testes transforma uma atividade meramente reativa, focada na caça a bugs, a um mecanismo proativo e **PREVENTIVO.**
9. Podemos afirmar que Verificação é garantir que seja entregue o que o cliente realmente queria? **SIM.**
10. QC (Quality Control) caracteriza-se como uma atividade detectiva tentando descobrir falhas? **SIM.**
11. Qual box oferece uma validação imparcial focada na experiência do usuário e nos requisitos de negócio? **BLACK-BOX TESTING.**
12. Encontrar bugs é o motivo único e principal ao se implementar uma estratégia de testes madura e eficaz? **NÃO.**

=====

FACOM - Faculdade de Computação - UFU

GESTÃO DA QUALIDADE DE SOFTWARE - Testes Métricas - a

Nome do aluno:

1. O paradigma Goal/Question/Metric (GQM) é usado para evitar o mau uso de dados, garantindo que as métricas sejam definidas com base em objetivos reais do projeto?
SIM
2. O foco central da Gerência de Engenharia de Software reside na coordenação eficaz dos três pilares (Pessoas, Produto e Processo), não incluindo o pilar Projeto?
ERRADO
3. Um dos riscos de ignorar o uso de Baselines é a perda de rastreabilidade, tornando impossível saber qual versão dos requisitos gerou qual versão do código?
CORRETO
4. A Baseline do Produto é estabelecida logo após a aprovação inicial dos requisitos do sistema, definindo o que o software deve fazer?
ERRADO
5. De acordo com o texto, a Gestão de Configuração de Software (SCM) é a atividade responsável por garantir que as mudanças sejam implementadas corretamente, mas não envolve o Controle de Versão (Version Control)?
ERRADO
6. Os quatro pilares centrais da Gerência de Engenharia de Software são Pessoas, Produto, Processo e Projeto?
CORRETO
7. De acordo com o texto, a Gestão de Configuração de Software (SCM) é a atividade responsável por garantir que as mudanças sejam implementadas corretamente, mas não envolve o Controle de Versão (Version Control)?
ANULADA
8. A Gerência de Engenharia de Software é definida no texto como apenas uma atividade tática de cronograma e orçamento, sem um esforço estratégico que equilibre as necessidades humanas?
ERRADO
9. De acordo com o texto, métricas como Linhas de Código (LOC) ou Litros de Café/Dia são consideradas exemplos de medições que geralmente oferecem valor real para orientar decisões de produtividade e qualidade?
ERRADO
10. A Lei de Goodhart, mencionada no texto, defende o uso de métricas como instrumento direto de punição ou recompensa para motivar a equipe?
ERRADO

FACOM - Faculdade de Computação - UFU

GESTÃO DA QUALIDADE DE SOFTWARE - Testes Métricas - B

Nome do aluno:

1. A Gerência de Engenharia de Software é vista como um esforço estratégico que equilibra as necessidades humanas, os requisitos do produto e a implementação de um processo rigoroso, em vez de ser apenas uma atividade tática? **CORRETO**
2. De acordo com o texto, o Comitê de Controle de Configuração (CCB) é o grupo responsável por avaliar o impacto de uma solicitação de mudança no custo e prazo do projeto antes de aprová-la? **CORRETO**
3. O texto afirma que uma Baseline, na Gerência de Configuração de Software, deve ser tratada simplesmente como um backup ou uma cópia de segurança dos arquivos do projeto? **ERRADO**
4. O paradigma Goal/Question/Metric (GQM) deve começar definindo as métricas quantificáveis para, em seguida, determinar os objetivos de negócios? **SIM**
5. Os quatro pilares centrais da Gerência de Engenharia de Software são Pessoas, Produto, Processo e Projeto? **CORRETO**
6. O texto sugere que a avaliação de métricas de resultado, devido à sua natureza, deve ocorrer com uma cadência de avaliação diária ou semanal? **ERRADO**
7. Uma das críticas à métrica de Velocidade é que, quando tratada como alvo, ela pode levar a equipe a inflar estimativas para artificialmente aumentar o número da métrica? **CORRETO**
8. A Lei de Goodhart, mencionada no texto, defende o uso de métricas como instrumento direto de punição ou recompensa para motivar a equipe? **ERRADO**
9. De acordo com o texto, a Gestão de Configuração de Software (SCM) é a atividade responsável por garantir que as mudanças sejam implementadas corretamente, mas não envolve o Controle de Versão (Version Control)? **ERRADO**
10. De acordo com o texto, métricas como Linhas de Código (LOC) ou Litros de Café/Dia são consideradas exemplos de medições que geralmente oferecem valor real para orientar decisões de produtividade e qualidade? **ERRADO**

REFERÊNCIAS

ALVES, L. M. et al. Longevity of risks in software development projects: a comparative analysis with an academic environment. *Procedia Computer Science*, Elsevier, v. 181, p. 827–834, 2021. Citado 2 vezes nas páginas [1](#) e [4](#).

BECK, K. *Test-driven development: by example*. [S.l.]: Addison-Wesley Professional, 2003. Citado na página [2](#).

MARTIN, R. C. *Código limpo: habilidades práticas do Agile software*. [S.l.]: Alta Books Grupo Editorial, 2019. Citado na página [3](#).

MYERS, G. J. *The art of software testing*. [S.l.]: John Wiley & Sons, 2006. Citado 2 vezes nas páginas [2](#) e [3](#).

PRESSMAN, R. S. et al. *Engenharia de software*. [S.l.]: Makron books São Paulo, 1995. v. 6. Citado 2 vezes nas páginas [2](#) e [3](#).

TAVARES, M. F. d. S. et al. Engenharia de requisitos no contexto da engenharia de software: aproximações possíveis entre prática e teoria na formação profissional. Belo Jardim, 2025. Citado 3 vezes nas páginas [1](#), [2](#) e [4](#).

TELES, M. A. et al. Gerenciar para quê? uma análise do gerenciamento de projetos em lojas de roupas com base no guia pmbok®. *Revista Gestão e Organizações*, v. 7, n. 4, p. 25–25, 2022. Citado 2 vezes nas páginas [2](#) e [4](#).

WANG, J. et al. Software testing with large language models: Survey, landscape, and vision. *IEEE Transactions on Software Engineering*, IEEE, v. 50, n. 4, p. 911–936, 2024. Citado 2 vezes nas páginas [2](#) e [3](#).

WASHIZAKI, H. Guide to the software engineering body of knowledge. *IEEE Computer Society*, 2024. Citado 2 vezes nas páginas [1](#) e [4](#).

WIEGERS, K.; BEATTY, J. *Software requirements*. [S.l.]: Pearson Education, 2013. Citado 3 vezes nas páginas [1](#), [4](#) e [5](#).