

1. Analisando a 'cadeia de efeitos' em qualidade de software, qual das seguintes afirmativas descreve mais precisamente a relação entre erro, defeito e falha?

A. Uma falha é a causa raiz de um problema, levando a um erro no código e, consequentemente, a um defeito no sistema.

B. Um erro, como uma lógica inadequada, pode criar um defeito latente no software, que se manifesta como uma falha sob certas condições de execução.

✓ **Isso mesmo!**

Esta opção descreve corretamente a sequência: o erro (causa) leva ao defeito (anomalia no sistema), que por sua vez pode levar à falha (manifestação para o usuário).

C. Um defeito é a manifestação de uma falha durante a execução, que foi originada por um erro humano.

D. Um erro de codificação introduzido por um desenvolvedor resulta diretamente em uma falha, que pode ou não conter um defeito.

Avançar

2. Qual das seguintes atividades NÃO pertence à fase de 'Análise de Testes'?

A. Definir e priorizar condições de teste com base na análise da base de teste.

B. Avaliar a testabilidade da base de teste e dos itens de teste.

✗ **Não era bem isso**

Avaliar se o software pode ser testado de forma eficaz é uma parte crucial da análise inicial.

C. Projetar e priorizar casos de teste de alto nível e identificar os dados de teste necessários.

✓ **Resposta correta**

O projeto detalhado de casos de teste e a identificação de dados específicos ocorrem na fase de 'Modelagem de Testes', que responde a 'como testar'.

D. Capturar a rastreabilidade bidirecional inicial entre os elementos da base de teste e as condições de teste.

[Voltar](#)

[Avançar](#)

3. O 'Paradoxo do Pesticida' no contexto de testes de software sugere que:

A. Se os mesmos testes são repetidos continuamente, eles perdem a eficácia para encontrar novos defeitos.

✓ **Isso mesmo!**

Assim como os insetos se tornam resistentes a um pesticida, o software se torna 'imune' a um conjunto de testes estático, que deixa de revelar novos problemas.

B. Testes de regressão são inerentemente ineficazes e devem ser substituídos por novas estratégias de teste a cada ciclo.

C. É mais eficaz concentrar os esforços de teste em um pequeno número de módulos, pois a maioria dos defeitos se agrupa ali.

D. A utilização excessiva de ferramentas de automação de teste pode introduzir mais defeitos do que os que são encontrados.

[Voltar](#)

[Avançar](#)

4. Qual dos seguintes itens é um 'produto de trabalho' característico da fase de 'Monitoramento e Controle de Teste'?

A. Plano de testes definindo o escopo e a abordagem.

B. Relatórios de progresso e resumos de teste para as partes interessadas.

✓ Isso mesmo!

Esta fase foca em comparar o progresso com o plano e comunicar o status, o que é feito através de relatórios.

C. Casos de teste detalhados e scripts de automação.

D. Condições de teste definidas e priorizadas.

[Voltar](#)

[Avançar](#)

5. A afirmação de que 'um pequeno número de módulos geralmente contém a maioria dos defeitos' refere-se a qual princípio de teste?

A. Testes exaustivos são impossíveis.

B. Ilusão da ausência de erros.

C. Agrupamento de defeitos.

✓ Isso mesmo!

Este princípio, baseado no Princípio de Pareto, descreve exatamente esse fenômeno, permitindo focar os esforços de teste.

D. Teste depende do contexto.

[Voltar](#)

[Avançar](#)

6. Do ponto de vista da psicologia de teste, qual é a abordagem mais construtiva para um testador comunicar a descoberta de um defeito?

A. Apontar o responsável pelo erro no relatório de defeito para agilizar a correção.

B. Descrever o problema de forma neutra e factual, focando no impacto para o sistema e o usuário, sem críticas pessoais.

✓ **Isso mesmo!**

Uma comunicação neutra e focada nos fatos promove a colaboração e evita que a mensagem seja vista como um ataque pessoal.

C. Minimizar a gravidade do defeito para não desmotivar o desenvolvedor.

D. Comunicar apenas as falhas mais críticas e ignorar as de baixo impacto para não sobrecarregar a equipe.

[Voltar](#)

[Avançar](#)

7. Qual das seguintes opções representa a 'Ilusão da ausência de erros' (ou Falácia da ausência de defeitos)?

A. O software passa em todos os testes de especificação, mas não atende às necessidades reais do usuário e é difícil de usar.

✓ **Isso mesmo!**

Esta é a definição exata do princípio: um sistema tecnicamente correto (verificado) pode ser um produto fracassado porque não foi validado contra as expectativas e necessidades do usuário.

B. A crença de que testes automatizados podem encontrar todos os tipos de defeitos, incluindo os de usabilidade.

C. Um gerente de projeto que acredita ser possível entregar um software complexo completamente livre de erros.

D. A equipe de teste decide parar de testar após um longo período sem encontrar novos defeitos.

[Voltar](#)

[Avançar](#)

8. A principal razão para realizar 'Testes Antecipados' no ciclo de vida de desenvolvimento de software é:

A. Reduzir o custo total da qualidade, pois o custo para corrigir defeitos aumenta exponencialmente nas fases posteriores do projeto.

✓ **Isso mesmo!**

Este é o argumento central para o teste antecipado. Encontrar e corrigir um defeito em requisitos é muito mais barato do que corrigi-lo no código em produção.

B. Substituir completamente a necessidade de testes de sistema e aceitação no final do ciclo.

C. Permitir que os desenvolvedores escrevam código de maior qualidade desde o início, pois sabem que serão testados.

D. Garantir que a equipe de teste tenha trabalho desde o início do projeto.

[Voltar](#)

[Avançar](#)

9. Qual das seguintes atividades é central para a fase de 'Encerramento de Testes'?

A. Priorizar os defeitos em aberto que precisam ser corrigidos antes do lançamento.

B. Analisar as lições aprendidas para melhorar o processo de teste em projetos futuros.

✓ **Isso mesmo!**

Consolidar a experiência, analisar o que deu certo e errado (retrospectivas) e melhorar a maturidade do processo são atividades-chave do encerramento.

C. Criar os scripts de teste automatizados que serão usados no projeto.

D. Executar testes de regressão finais para garantir a estabilidade do sistema.

[Voltar](#)

[Avançar](#)

10. O princípio 'Teste depende do contexto' implica que:

A. A mesma abordagem de teste rigorosa usada para um software de controle de voo deve ser aplicada a um site de e-commerce.

B. O contexto do teste é definido exclusivamente pelas ferramentas de automação disponíveis para a equipe.

C. A estratégia, o rigor e as técnicas de teste devem ser adaptados com base nos riscos, no domínio da aplicação e nos objetivos do projeto.

✓ **Isso mesmo!**

Esta opção captura a essência do princípio: o teste não é uma atividade única para todos, mas deve ser moldado pelas circunstâncias específicas do projeto.

D. Existe um conjunto universal de 'melhores práticas' de teste que garante o sucesso em qualquer projeto.

[Voltar](#)

[Avançar](#)

11. A rastreabilidade bidirecional no processo de teste é fundamental para:

A. Apenas garantir que o código escrito por cada desenvolvedor seja testado.

B. Avaliar a cobertura do teste, analisar o impacto das mudanças e tornar o processo auditável.

✓ **Isso mesmo!**

Esses são os principais benefícios da rastreabilidade, permitindo verificar se todos os requisitos foram testados e entender as consequências de uma mudança em um requisito.

C. Determinar o tempo exato que cada teste levará para ser executado.

D. Substituir a necessidade de comunicação direta entre testadores e desenvolvedores.

[Voltar](#)

[Avançar](#)

12. Qual das seguintes NÃO é considerada uma característica de um bom testador, de acordo com o material?

A. Curiosidade e atenção aos detalhes.

B. Pessimismo profissional e olhar crítico.

C. Boa capacidade de comunicação.

D. Otimismo e confiança de que o software funcionará como esperado.

✓ Isso mesmo!

Enquanto a colaboração é importante, uma mentalidade excessivamente otimista pode ir contra o objetivo de encontrar defeitos. O pessimismo profissional é o atributo listado.

[Voltar](#)

[Avançar](#)

13. Na fase de 'Modelagem de Testes' (ou Design de Teste), a equipe de teste está primariamente preocupada em responder qual pergunta?

A. Como testar?

✓ Isso mesmo!

Esta é a questão central da modelagem, onde as condições de teste são transformadas em casos de teste concretos, com dados e ambiente definidos.

B. Está tudo pronto para começar?

C. O teste foi bem-sucedido?

D. O que testar?

[Voltar](#)

[Avançar](#)

14. A atividade de 'Comparar o progresso real com o plano de teste usando métricas' é uma responsabilidade central de qual processo de teste?

A. Execução de Testes

B. Análise de Testes

✗ **Não era bem isso**

A análise foca no produto a ser testado para definir o que será testado.

C. Planejamento de Testes

D. Monitoramento e Controle de Teste

✓ **Resposta correta**

A essência do monitoramento e controle é exatamente essa: verificar o andamento, usar métricas para comparar com o planejado e tomar ações corretivas.

[Voltar](#)

[Avançar](#)

15. Um desenvolvedor comete um erro de sintaxe ao declarar uma variável. Isso representa um:

A. Erro, que é a ação humana incorreta. O código com a sintaxe errada contém um defeito.

✓ **Isso mesmo!**

A ação do programador é o erro. A consequência no código, a sintaxe incorreta que causa um erro de compilação, é o defeito.

B. Falha, pois o compilador não consegue construir o programa, impactando o processo de desenvolvimento.

C. Defeito, que é a causa raiz, levando a um erro na compilação e a uma falha no ambiente de desenvolvimento.

D. Erro, que causará uma falha de compilação, mas não é considerado um defeito até que o código seja executado.

[Voltar](#)

[Avançar](#)

16. Qual é a principal saída (produto de trabalho) da fase de 'Análise de Testes'?

A. Relatórios de defeitos detalhados.

B. Casos de teste reutilizáveis com passos detalhados.

C. Um cronograma de execução de teste.

D. Condições de teste definidas e priorizadas.

✓ **Isso mesmo!**

A análise da base de teste resulta na identificação do 'que' precisa ser testado, que são as condições de teste.

[Voltar](#)

[Avançar](#)

17. A 'Definição de Feito' (Definition of Done) ou 'Critérios de Saída' são utilizados no processo de teste para:

A. Determinar quando iniciar a fase de execução de testes.

B. Determinar se o nível de qualidade é suficiente e se mais testes são necessários.

✓ **Isso mesmo!**

Eles são um conjunto de condições acordadas (ex: cobertura de teste, número de defeitos críticos) que, quando atendidas, indicam que a atividade de teste pode ser concluída.

C. Avaliar o desempenho individual dos testadores da equipe.

D. Planejar o orçamento e os recursos necessários para o projeto de teste.

[Voltar](#)

[Avançar](#)

18. Qual das seguintes ações ajuda a mitigar o 'Paradoxo do Pesticida' em testes que buscam encontrar novos defeitos?

A. Focar os testes apenas nas áreas onde defeitos foram encontrados anteriormente (agrupamento de defeitos).

B. Reduzir o número total de testes executados para economizar tempo.

C. Repetir os mesmos testes de regressão automatizados em todos os ciclos de desenvolvimento.

D. Adicionar novos casos de teste e modificar os testes existentes e seus dados a cada ciclo.

✓ **Isso mesmo!**

Para superar o efeito do paradoxo, é necessário variar e atualizar constantemente o conjunto de testes para exercitar o software de novas maneiras.

[Voltar](#)

[Avançar](#)

19. Durante a fase de 'Execução de Testes', um testador executa um caso de teste e o resultado real é diferente do resultado esperado. Qual é o próximo passo mais apropriado?

A. Modificar o resultado esperado no caso de teste para que corresponda ao resultado real.

B. Analisar a anomalia para determinar se é um defeito real e, em caso afirmativo, comunicá-lo formalmente.

✓ **Isso mesmo!**

Antes de registrar um defeito, é preciso analisar a discrepância para garantir que não foi um erro no ambiente, nos dados ou na execução do teste. Se for um defeito, ele deve ser registrado.

C. Corrigir o código imediatamente para que o teste passe.

D. Ignorar a discrepância e passar para o próximo teste para manter o cronograma.

[Voltar](#)

[Avançar](#)

20. A atividade de 'Finalizar e arquivar o ambiente de teste, dados de teste e infraestrutura' pertence a qual fase do processo de teste?

A. Monitoramento e Controle de Teste

B. Encerramento de Testes

✓ Isso mesmo!

Esta é uma atividade de 'limpeza' que ocorre no final do projeto para garantir que os ativos de teste sejam preservados para referência futura ou reutilização.

C. Implementação de Testes

D. Modelagem de Testes

[Voltar](#)

[Avançar](#)

21. A incapacidade de um aplicativo de e-commerce processar pagamentos durante um pico de tráfego é um exemplo clássico de:

A. Um risco de negócio que deveria ter sido aceito durante o planejamento.

B. Um erro de lógica na codificação do processamento de pagamento.

C. Uma falha, pois o sistema não realizou sua função esperada do ponto de vista do usuário.

✓ Isso mesmo!

A falha é a manifestação externa de um defeito durante a interação do usuário. A transação não concluída é exatamente essa manifestação.

D. Um defeito no módulo de pagamento que não foi detectado em testes de baixo volume.

[Voltar](#)

[Avançar](#)

22. Por que a análise de risco é crucial em um contexto onde 'testes exaustivos são impossíveis'?

A. Para concentrar os esforços de teste limitados nas áreas do sistema que apresentam maior probabilidade e impacto de falha.

✓ Isso mesmo!

Dado que não se pode testar tudo, a análise de risco é a abordagem lógica para priorizar e focar os recursos de teste onde eles podem agregar mais valor, mitigando os maiores riscos.

B. Para transferir toda a responsabilidade por falhas em produção para a equipe de gerenciamento de produto.

C. Para eliminar completamente a necessidade de testes de regressão, economizando tempo e recursos.

D. Para provar matematicamente que as áreas não testadas do software estão livres de defeitos.

[Voltar](#)

[Avançar](#)

23. Um 'viés de confirmação' (confirmation bias) na psicologia de teste é mais provável de ser um problema quando:

A. Um desenvolvedor testa seu próprio código para verificar se ele atende aos requisitos positivos.

✓ Isso mesmo!

O autor do código tem uma tendência natural a criar testes que provem que seu código funciona (confirmem sua crença), em vez de testes que tentem quebrá-lo.

B. O cliente final realiza o Teste de Aceitação do Usuário (UAT) para validar o produto.

C. Um testador independente executa testes exploratórios em um sistema pela primeira vez.

D. Uma equipe de automação de testes converte casos de teste manuais existentes em scripts.

[Voltar](#)

[Avançar](#)

24. Qual das seguintes situações melhor ilustra a aplicação do princípio 'Agrupamento de Defeitos' para otimizar o esforço de teste?

A. Distribuir o tempo de teste igualmente entre todos os módulos do sistema para garantir uma cobertura uniforme.

B. Permitir que os desenvolvedores decidam quais partes de seu próprio código devem ser testadas.

C. Após uma análise de dados históricos de produção que mostra que 80% das falhas ocorrem no módulo de faturamento, a equipe decide alocar recursos adicionais para testes mais rigorosos e revisões de código nesse módulo.

✓ Isso mesmo!

Esta é uma aplicação direta do princípio: usar dados para identificar um 'cluster' de defeitos e focar os esforços de qualidade nessa área de alto risco.

D. Executar o mesmo conjunto de testes de regressão em cada novo build do software.

[Voltar](#)

[Avançar](#)

25. Um produto de software foi lançado e, embora tecnicamente robusto e com poucos defeitos relatados, teve uma adoção muito baixa pelos usuários, que o consideraram confuso e inadequado para suas tarefas. Este cenário é um exemplo claro de falha na:

A. Implementação de Testes, pois o ambiente de teste não refletia o ambiente de produção.

B. Verificação, pois a equipe de teste não conseguiu encontrar todos os defeitos ocultos no sistema.

C. Validação, pois o software, embora construído corretamente, não era o produto certo para o usuário.

✓ **Isso mesmo!**

Isso se alinha perfeitamente com a 'Ilusão da ausência de erros'. O sistema foi verificado (construído corretamente) mas não validado (não era o produto certo).

D. Execução de Testes, pois os testes foram executados de maneira inadequada.

[Voltar](#)

[Avançar](#)

26. Na fase de 'Implementação de Testes', qual das seguintes atividades é a mais central?

A. Desenvolver e priorizar procedimentos de teste, criar suítes de teste e agendá-las para execução.

✓ **Isso mesmo!**

A implementação é sobre organizar e preparar todos os ativos de teste para que a execução possa ocorrer de forma suave e ordenada. Isso inclui agrupar casos de teste em suítes e criar uma agenda.

B. Comparar os resultados reais com os resultados esperados para encontrar anomalias.

C. Analisar a base de teste (requisitos, histórias de usuário) para identificar o que deve ser testado.

D. Consolidar a experiência do projeto e analisar as lições aprendidas em uma retrospectiva.

[Voltar](#)

[Avançar](#)

27. A principal distinção entre 'produtos de trabalho' da fase de 'Análise de Testes' e da fase de 'Modelagem de Testes' é:

A. A Análise produz um Plano de Testes, enquanto a Modelagem produz Relatórios de Progresso.

B. A Análise gera relatórios de defeitos, enquanto a Modelagem cria o ambiente de teste.

C. A Análise foca em 'o quê' testar, resultando em Condições de Teste, enquanto a Modelagem foca em 'como' testar, resultando em Casos de Teste detalhados.

✓ **Isso mesmo!**

Esta é a distinção fundamental. A análise é abstrata e identifica os itens a serem testados. A modelagem é concreta e projeta os passos específicos para testar esses itens.

D. A Análise define os Critérios de Saída, enquanto a Modelagem define a estratégia de automação.

[Voltar](#)

[Avançar](#)

28. Se uma equipe de testes, ao longo de vários meses, utiliza o mesmo conjunto de dados de teste para validar um formulário de cadastro, e deixa de encontrar novos defeitos, mesmo que a equipe de desenvolvimento continue a fazer alterações no back-end relacionado ao formulário, qual princípio de teste está sendo demonstrado?

A. Agrupamento de Defeitos, pois todos os defeitos já foram encontrados naquela área.

B. Testes Mostram a Presença de Defeitos, pois os testes estão confirmando que não há mais defeitos.

C. Paradoxo do Pesticida, pois os testes repetitivos com os mesmos dados perderam a eficácia para detectar novos tipos de defeitos.

✓ **Isso mesmo!**

A repetição do mesmo estímulo (dados de teste) torna o sistema 'resistente' a essa forma de verificação. Novos defeitos introduzidos pelas alterações no back-end podem não ser ativados pelos dados antigos.

D. Teste Depende do Contexto, pois este contexto não exige novos testes.

[Voltar](#)

[Avançar](#)

29. Uma organização está desenvolvendo um sistema de software para uma sonda espacial. Outra está desenvolvendo um aplicativo de rede social. De acordo com o princípio 'Teste depende do contexto', qual das seguintes afirmações é mais provável que seja verdadeira?

A. Ambos os projetos devem adotar a mesma metodologia de teste ágil, pois é a prática mais moderna da indústria.

B. O projeto da sonda espacial exigirá um nível muito mais alto de rigor, documentação, rastreabilidade e cobertura de teste do que o aplicativo de rede social, devido ao altíssimo custo da falha.

✓ **Isso mesmo!**

O contexto, definido pelo risco e pelo impacto da falha, é drasticamente diferente. O custo de uma falha na sonda é catastrófico, justificando um processo de teste muito mais rigoroso e formal.

C. A equipe do aplicativo social gastará mais tempo em testes estáticos e revisões formais, enquanto a equipe da sonda focará em testes exploratórios e dinâmicos.

D. O projeto da sonda espacial provavelmente enfatizará testes de usabilidade e A/B testing, enquanto o aplicativo social focará em testes de integração de hardware.

30. A comunicação de um testador que afirma 'Seu código quebrou o build' para um desenvolvedor é considerada menos eficaz do que 'O build falhou após a integração da funcionalidade X, e os logs indicam um erro de referência nula no módulo Y'. Por que a segunda abordagem é preferível, de acordo com a psicologia de teste?

A. Porque ela usa jargão técnico que demonstra a superioridade do conhecimento do testador.

B. Porque ela se comunica de forma neutra, sobre fatos observáveis, sem críticas pessoais, e fornece informações construtivas para a solução do problema.

✓ **Isso mesmo!**

Esta abordagem foca no problema ('o build falhou') e nos fatos ('erro no módulo Y'), em vez de soar como uma acusação pessoal ('seu código quebrou'). Isso reduz a defensividade e promove a colaboração.

C. Porque ela remove qualquer responsabilidade do testador, colocando-a inteiramente no processo de build.

D. Porque ela é mais longa e detalhada, o que sempre é melhor em relatórios de defeitos.

[Voltar](#)

[Concluído](#)