

CONHECIMENTOS ESPECÍFICOS

Julgue os seguintes itens, acerca de testes de *software*.

- 61 Testes com base na arquitetura do *software* podem ser realizados manualmente ou de forma automatizada em qualquer nível de teste.
- 62 A realização de testes de desempenho deve ser realizada o quanto antes, ainda durante a etapa de arquitetura e implementação da solução. Para este fim, deve-se adotar estratégia que identifique as partes de uma funcionalidade que estejam demorando mais tempo para serem executadas, no nível de teste de componente ou no nível de integração de componente.
- 63 Os testes não funcionais são realizados após a conclusão dos testes funcionais, uma vez que problemas oriundos de requisitos não funcionais apresentam pouco impacto na solução tecnológica.
- 64 A segurança do *software* é um dos itens tratados em testes não funcionais.

No que diz respeito ao desenvolvimento de testes de *software*, julgue os itens a seguir.

- 65 Os testes que se baseiam na arquitetura de *software* e no seu código fonte são denominados testes caixa branca. Nesse tipo de testes, devem ser incluídos a validação de decisões e o relacionamento das estruturas testadas entre si para verificar a decisão tomada ao final de um processo.
- 66 As técnicas para *design* de testes são classificadas em duas abordagens: estática, que se baseia em revisões e análises estáticas; e dinâmica, que se baseia em testes de caixa branca, experiência e testes de caixa preta.
- 67 Caso um desenvolvimento de testes tenha sido realizado com base na documentação de regras de negócio e em casos de uso, é correto afirmar que o tipo de abordagem utilizada na concepção dos testes baseou-se em experiência.
- 68 A análise de limite de valores é uma técnica de teste caixa preta, utilizada para testar problemas no uso de valores extremos em variáveis e estruturas lógicas relacionadas a uma ou mais funcionalidades.

Acerca de gerenciamento de testes de *software*, julgue os itens subsequentes.

- 69 O relacionamento entre a equipe de teste e a equipe de desenvolvimento é um fator determinante para mitigar ou eliminar os problemas encontrados no ciclo de vida da solução tecnológica.
- 70 Uma das atribuições do gerente de testes é planejar modificações nos planos de teste de acordo com os resultados obtidos e estabelecer relatórios de testes úteis à tomada de decisões no projeto.
- 71 Analistas de teste com perfil técnico, especializados em aspectos específicos da solução, são dispensáveis em projetos maiores.

Considere o projeto de uma aplicação com interface *web*, cujo número previsto de usuários é de 4 milhões de pessoas, e que deve atender a aspectos de acessibilidade a pessoas com deficiência visual. Considere, ainda, que durante a etapa de desenvolvimento da aplicação, a disciplina de testes tenha sido negligenciada devido ao curto prazo de entrega do projeto. Assim, quando a aplicação foi implementada em ambiente de produção e liberada para acesso, ocorreram os seguintes problemas:

- ▶ lentidão ao acessar sistema por meio de nome de usuário e senha;
- ▶ as páginas *web* da aplicação não funcionavam satisfatoriamente nos navegadores Internet Explorer e Safari;
- ▶ todas as funcionalidades que envolvem gravação, alteração e consulta a banco de dados apresentaram lentidão superior ao tempo esperado de resposta definido pelo gestor do projeto;
- ▶ existência de campos nas páginas de inserção de dados e consulta que permitem a inclusão de qualquer tipo de dados, sem qualquer tratamento.

No período de desenvolvimento da aplicação, o gerente de projeto optou por utilizar uma metodologia ágil (Scrum), deixando a cargo dos desenvolvedores a criação de testes unitários para testar a solução, sendo esta a única frente de testes do projeto. A fim de solucionar os problemas da aplicação, um analista de testes com conhecimento em programação foi designado para examinar a solução e apontar problemas na aplicação.

Com base nas informações apresentadas na situação hipotética, julgue os itens que se seguem.

- 72 A existência de campos que possibilitem a inserção de qualquer tipo de dado pelo usuário não compromete a segurança da solução tecnológica descrita.
- 73 Os testes de integração estão entre as ações que possibilitariam apontar a origem do problema de desempenho do banco de dados.
- 74 Um teste de desempenho deve ser realizado para mensurar se a infraestrutura tecnológica de produção é suficiente para suportar a quantidade de acessos simultâneos à aplicação *web*. Uma forma de realizar essa verificação é utilizar ferramenta com suporte à medição de teste de cobertura.
- 75 As páginas *web* que não funcionaram satisfatoriamente nos navegadores citados não necessitam ser testadas semanticamente, visto que a conformidade com padrões W3C não é alvo dos processos de teste de um sistema.
- 76 A lentidão nas operações do sistema está relacionada a não mensuração correta da infraestrutura tecnológica que suporta a solução, devendo o analista de testes focar em um teste de desempenho baseado na quantidade de acessos simultâneos à aplicação *web*.

A respeito de gerenciamento de defeitos, julgue os itens subsequentes.

- 77 Um relatório de incidente deve identificar o defeito por meio de uma ID única, um resumo da anomalia e uma descrição detalhada do defeito, como por exemplo, as pré-condições e pós-condições e demais informações que contextualizam o defeito e sua correção.
- 78 A classificação da prioridade de correção dos defeitos encontrados deve ser realizada por um processo de triagem, cujas principais características a serem consideradas são o custo e o prazo envolvidos.
- 79 Defeitos categorizados como menos prioritários devem ser corrigidos em versões posteriores do *software*.
- 80 O ciclo de vida de um defeito é realizado em três fases: identificação do defeito, ação para reproduzi-lo e ações para corrigi-lo.

Com relação à análise e à avaliação de riscos em projetos de teste de *software*, julgue os itens a seguir.

- 81 Após um sistema ter sido completamente integrado, é possível testá-lo em relação ao desempenho e à confiabilidade. O teste de desempenho significa estressar o sistema por meio de demandas fora dos limites do projeto de *software*.
- 82 Teste é um processo que busca atingir a confiabilidade do *software*, portanto, por meio de testes é possível verificar se um *software* está livre de defeitos ou comportamentos indesejados em todas as circunstâncias.
- 83 A JUnit é uma biblioteca típica de testes de apoio e fornece suporte para a execução de testes, geração de *logs* e verificação de resultados. Por meio do conjunto de classes C# do JUnit, é possível a criação de ambiente de testes automatizado.

No que diz respeito à norma ISO/IEC 27005, de gestão de riscos, julgue os itens seguintes. Nesse sentido, considere que a sigla SGSI, sempre que utilizada, se refere a sistema de gestão de segurança da informação.

- 84 O processo descrito pela norma ISO/IEC 27005 fornece as diretrizes para a gestão de riscos de segurança da informação de uma organização sincronizada com o ciclo de melhoria contínua PDCA utilizado em um SGSI, conforme a ISO/IEC 27001.
- 85 Conforme descreve a norma em questão, a metodologia para estimativa de riscos qualitativa pode ser utilizada: como uma verificação inicial a fim de identificar riscos que exigirão uma análise mais detalhada; quando esse tipo de análise é suficiente para a tomada de decisões; e quando os dados numéricos ou recursos são insuficientes para uma estimativa quantitativa.
- 86 O processo para a gestão de riscos de segurança da informação descrito pela referida norma pode ser usado para avaliações de riscos em um projeto; porém, não pode ser aplicado à operação cotidiana de um SGSI.

Segundo o manual de práticas de contagem (CPM) do IFPUG, versão 4.3, e o roteiro de métricas de *software* do SISF, versão 2.0, julgue os itens subsequentes, a respeito da mensuração de sistemas em pontos de função.

- 87 De acordo com o CPM, *Scope Creep* é o fenômeno que explica a mudança de requisitos anteriores à implantação dos projetos de desenvolvimento e de manutenção de *software*, a partir do maior conhecimento sobre as necessidades e funcionalidades de negócio adquiridas pelo usuário e pelo desenvolvedor.
- 88 No CPM 4.3, estão definidos dois tipos de projetos de *software*: o de desenvolvimento e o de melhoria. Caso seja requisitada no projeto de *software* a migração ou carga inicial de dados para uma nova aplicação, tanto no projeto de desenvolvimento quanto no projeto de melhoria, não serão consideradas as funcionalidades de conversão de dados.
- 89 De acordo com a versão 4.3 do CPM, em caso de requisitos de usuário para o desenvolvimento de funcionalidades de manutenção de *help*, deve-se contar também a função de dados e as funcionalidades de manutenção.

Com relação a conceitos de qualidade de *software*, julgue os próximos itens.

- 90 Quando ocorre uma falha na execução de um programa devido à sua não conformidade a um requisito de *software*, a correção dessa falha impede a ocorrência de novos erros, os quais poderiam resultar em outras falhas.
- 91 A qualidade de *software* é fundamentada nas necessidades do usuário. A falta de conformidade aos requisitos de *software* é determinante para a falta de qualidade de *software*.

No que se refere à garantia da qualidade de *software* (SQA – *software quality assurance*), julgue os itens que se seguem.

- 92 Segundo a SQA estatística, 80% dos defeitos de *software* podem ser mapeados em 20% de todas as causas possíveis no código.
- 93 SQA pode ser aplicada em algumas fases do processo de engenharia de *software*, como análise, projeto, codificação e testes.

Julgue os próximos itens, a respeito de teste e qualidade de *software*.

- 94 Na norma IEEE 829, são descritas as diretrizes para as atividades de teste de um produto de *software*.
- 95 A norma e o modelo de qualidade de processo ISO 15504 são utilizados como base para a elaboração da norma ISO 12207 e têm como objetivo apoiar as organizações na definição de seus processos de ciclo de vida de desenvolvimento, operação e manutenção de *software*.
- 96 A NBR ISO/IEC 9126 descreve um modelo, segmentado em quatro partes, de qualidade do produto de *software* e também define as características e subcaracterísticas que devem ser medidas e avaliadas nos produtos de *software*.

Acerca dos conceitos da gestão de configuração, julgue os itens subsequentes. Nesse sentido, considere que a sigla SACM, sempre que utilizada, se refere a *service asset and configuration management*.

- 97 Segundo a versão 3 do ITIL, o âmbito do SACM não se estende aos fornecedores de serviços internos e externos, já que eles não são considerados como ativo do serviço de tecnologias de informação.
- 98 Na versão 3 da biblioteca ITIL, o SACM é um processo que não se relaciona, em nenhum momento, com outros processos da biblioteca, tais como *Incident Management*, *Problem Management*, *Change Management* e o serviço *Service Desk*.
- 99 A versão 3 do ITIL define Item de Configuração como um ativo, um serviço ou outro item que está, ou estará, sob o controle da gestão da configuração. São exemplos de componentes dos itens de configuração: *hardware*, *software*, documentação, equipes e grupos de suporte.

De acordo com a IN MPOG n.º 4/2014, julgue os itens a seguir a respeito do processo de contratação de soluções de tecnologia da informação (TI).

- 100 A estratégia geral de TI para a administração direta, autárquica e fundacional do poder executivo federal será elaborada pelo órgão central do SISP, em conjunto com os órgãos setoriais e seccionais do SISP. O documento servirá de subsídio à elaboração dos PDTI pelos órgãos e entidades integrantes do SISP.
- 101 A referida norma pode ser aplicada em qualquer contratação de soluções de TI, mesmo àquelas que possam comprometer a segurança nacional.
- 102 De acordo com a norma em questão, o gestor do contrato deve ser um servidor com atribuições gerenciais, indicado por autoridade competente, para coordenar e comandar o processo de gestão e fiscalização da execução contratual.

Acerca dos conceitos de gerenciamento de projetos, julgue os itens seguintes.

- 103 Segundo o guia PMBOK, SWOT é uma técnica de gerenciamento de projeto útil para identificar as causas dos riscos.
- 104 Segundo o guia PMBOK, as fases do projeto são divisões em que é necessário um controle para gerenciar de forma efetiva o término de uma entrega. Geralmente, as fases são terminadas sequencialmente, mas em algumas situações podem se sobrepor.
- 105 Segundo o guia PMBOK, o *brainstorming* não pode ser considerado como exemplo de técnica de coleta de informações, com a finalidade de identificar erros.

Julgue os itens a seguir, a respeito da engenharia de *software*.

- 106 A análise de pontos por função é uma métrica que permite mensurar produtividade, qualidade, custo e documentação de um *software*.
- 107 A engenharia de *software*, composta por um conjunto de três elementos fundamentais — métodos, ferramentas e procedimentos — torna disponível ao desenvolvedor uma base para construção de um *software* de alta qualidade em termos de produtividade.
- 108 Na fase de engenharia de requisitos do *software*, do paradigma do ciclo de vida clássico da engenharia de *software* chamado de modelo cascata, são identificadas as necessidades do sistema do ponto de vista do desenvolvedor, sem a presença do solicitante.

Acerca de ferramentas CASE (*computer aided software engineering*), julgue os itens subsequentes.

- 109 IDE (*integrated development environment*) é um exemplo de ferramenta CASE utilizada com o objetivo de acelerar o desenvolvimento do *software*.
- 110 As ferramentas CASE abrangem todas as ferramentas eletrônicas utilizadas em computadores, destinadas a auxiliar o desenvolvedor na etapa de codificação do *software*.

A respeito da linguagem UML, julgue os itens que se seguem.

- 111 A linguagem UML, que utiliza conceitos orientados a objetos, é um sistema de notação dirigida à modelagem de sistemas.
- 112 A linguagem UML padroniza artefatos e notação e define um processo padrão de desenvolvimento de *softwares*.
- 113 Caso de uso é uma narrativa utilizada para descrever a sequência de eventos de um ator que usa um sistema para completar um processo.
- 114 O ícone seguinte é a representação para um ator de um caso de uso.



- 115 Para descrever um processo de forma breve, em duas ou três sentenças, utiliza-se um caso de uso de alto nível.

Com base em desenvolvimento (Ágil) de *software*, julgue os seguintes itens.

- 116 Uma das premissas do desenvolvimento Ágil de *software* é que pessoas ligadas ao negócio e ao desenvolvimento devem trabalhar juntas somente na fase inicial do projeto.
- 117 Uma das entregas necessárias do desenvolvimento (Ágil) de *software* é a completa documentação do sistema e do projeto.
- 118 Acatar as mudanças de requisitos, ainda que o desenvolvimento já esteja avançado, é um dos princípios do Manifesto Ágil.

Acerca do *Scrum*, julgue os itens subsequentes.

- 119 *Daily Scrum*, também conhecido como *Daily Standup*, é uma reunião diária rápida para se atualizar o *Scrum Master* sobre o estado do projeto.
- 120 Uma das vantagens do *Scrum* é que, com ele, aplica-se um mecanismo de redução sistemática de risco, por meio de seu ciclo frequente de inspeção e adaptação.
- 121 Um projeto *Scrum* inicia-se com o *Product Owner*, que coleta informações dos *stakeholders* a fim de que seja elaborada uma lista de requisitos e de um *backlog* de produto.

Julgue os próximos itens com referência a MPS.BR e CMMI.

- 122 MR – MPS, MA – MPS e MN – MPS são as três partes que compõem o MPS.BR.
- 123 O CMMI é um modelo de referência que define práticas necessárias à maturidade em disciplinas específicas.
- 124 São seis os níveis de maturidade do MPS.BR: A – Em Otimização; B – Gerenciado quantitativamente; C – Definido; D – Largamente Definido; E – Parcialmente Definido; F – Gerenciado.

Com relação à orientação a objetos, julgue os itens subsecutivos.

- 125 Objetos são definidos como entidades da modelagem de sistemas que armazenam estados com a utilização de atributos dos próprios objetos, sem interação com outros objetos por meio de mensagens.
- 126 Em um sistema bem elaborado, os métodos são o único meio de se interagir com os objetos.
- 127 O foco da orientação a objetos está nos procedimentos a serem contemplados pelo sistema e nas informações que este manipulará e(ou) armazenará.
- 128 De acordo com o processo RUP, o desenvolvimento de um *software* é dividido em quatro fases: iniciação, elaboração, construção e transição.

Julgue os itens a seguir, com base no padrão MVC (*model – view – controller*).

- 129 O padrão de arquitetura MVC separa as *views* e os *controllers* do modelo do sistema, o que permite alterações na interface sem impacto na camada de lógica do negócio.
- 130 A função do *controller* é enviar comandos para sua *view* associada para alterar a representação dos dados, porém ele não tem autoridade para alterar o modelo.

Espaço livre