

COMPILADO

DESENVOLVIMENTO PARA ENGENHARIA QUALIDADE E TESTE DE SOFTWARE

[Versão PDF do Excel – by © Garay & Soares Ltda]

Caso de não houver alguma questão informar aos envolvidos – podem haver novas!

PERGUNTAS	RESPOSTAS
1. O engenheiro de software atua no desenvolvimento de programas, sua manutenção e adequação a diferentes processos produtivos, de forma a garantir maior desempenho e produtividade.	disponibilizar um software que se caracterize por ser de baixo custo e que ofereça confiabilidade e eficiência.
2. A aplicação de Lean Software Development pode ajudar equipes de desenvolvimento de software a atingirem maior nível de maturidade em termos de qualidade e velocidade, entendendo e estudando seu processo atual, e trabalhando constantemente na identificação e eliminação de desperdícios.	linha determinística.
3. “A prototipação se dá basicamente com a comunicação que ocorre através de uma reunião com todos os envolvidos a fim de definir objetivos gerais do software e identificar quais requisitos já estão bem conhecidos e esquematizar as áreas que realmente necessitam de uma definição mais ampla.	através de um modelo referenciado em PC ou em um papel, que apresente a interação existente entre homem e máquina.
4. “A reutilização de software se baseia no uso de conceitos, produtos ou soluções previamente elaboradas ou adquiridas para criação de um novo software, visando melhorar significativamente a qualidade e a produtividade”.Fonte: DEVMEDIA. Reutilização de Software. Revista Engenharia de Software Magazine, n. 39. Disponível em: < https://www.devmedia.com.br/reutilizacao-de-software-revista-engenharia-de-software-magazine-39/21956 >. Acesso em: 31 mar. 2019.Assim, considerando as informações apresentadas e os conteúdos estudados, analise as tarefas das unidades de software e associe-as às suas respectivas características:1) Reuso de Sistema.2) Reuso de Componentes.3) Reuso de objetos e funções.() Define qual os elementos que compõem um software e implementam uma única função podem ser reusados.() Acontece quando esses componentes, alterando o seu tamanho independentemente de serem subsistemas ou objetos únicos, possam ser reusados.() Ocorre quando um determinado sistema é reusado na íntegra, ou seja, na sua totalidade, em outros sistemas. Agora, assinale a alternativa que apresenta a sequência correta:	3, 2, 1.
5. “Após o término do primeiro incremento o cliente utiliza e avalia esse incremento fornecendo posteriormente um resultado ou feedback. Com base nesse resultado fornecido pelo cliente o próximo incremento é planejado considerando a modificação do primeiro incremento, caso seja necessário, de acordo com o feedback do cliente.	aplicar sequências lineares de forma escalonada, ao longo do tempo, gerando um incremento dos softwares finalizados para o cliente.
6. “As metodologias ágeis surgiram nos últimos anos como uma alternativa aos métodos tradicionais de gestão de projetos de TI. Por exemplo SCRUM, XP, Lean etc. Elas são conhecidas por sua maior flexibilidade, capacidade de se adaptarem a mudanças no ambiente de trabalho e ciclos com duração reduzida.	entrega rápida do produto aos clientes, que ao receberem o software podem opinar sobre mudanças posteriores no sistema.
7. “Bottom-up”, diferente de “top-down”, é uma abordagem de teste que avalia de forma crescente. A desvantagem é que ela pode fazer com que a empresa crie o hábito de resolver os problemas de forma desorganizada ou consertar um erro sem avaliar a real causa, sem descobrir a fonte do transtorno. Assim, recomenda-se usar métodos diversificados no mesmo teste. Considerando essas informações e de acordo com o conteúdo estudado, é correto afirmar que a vantagem na utilização da abordagem “Bottom-up” é:	encontrar facilmente erros lógicos e de cobertura.

PERGUNTAS	RESPOSTAS
8. “Em uma abordagem dirigida a planos, acontecem iterações no ambiente de atividades com documentos formais que são utilizados para estabelecer a comunicação entre os estágios do processo.	identificar etapas diferentes do processo de software, em que as saídas estão relacionadas a cada etapa.
9. “Equipes ágeis costumam lançar seus produtos mais rapidamente em comparação às equipes que utilizam os métodos tradicionais, pois vão sendo liberados partes que geram valor ao negócio.	resistência cultural das empresas.
10. “Esse é um modelo de processo de software evolucionário que também é iterativo como a prototipação, porém com aspectos sistemáticos e controlados do modelo cascata. O modelo espiral fornece um grande potencial para que possamos ter rápido desenvolvimento de versões cada vez mais completas.	3, 1, 2, 4.
11. “Extreme Programming (XP) é uma metodologia de desenvolvimento de software, nascida nos Estados Unidos ao final da década de 90. Vem fazendo sucesso em diversos países, por ajudar a criar sistemas de melhor qualidade, que são produzidos em menos tempo e de forma mais econômica que o habitual.	ocorre de maneira direta a uma série de atividades onde os programadores desenvolverão testes para cada uma delas, na fase anterior a escritura do código.
12. “Gestores de TI interessados em otimizar a governança e em ampliar os resultados da área precisam estar atentos a essas inovações, e é relevante conhecer melhor o que é DevOps. Em essência, esse modelo aproxima o desenvolvimento da operação com o objetivo de simplificar processos, integrar áreas, alcançar maior qualidade nas entregas e reduzir o tempo de resposta ao cliente.	utilização de processos automatizados direcionados à produção mais veloz e estável dos serviços e aplicações.
13. “Metodologia de desenvolvimento ágil é uma forma de construir software que possibilita obter as vantagens do sistema antes de ele estar pronto. Também encoraja os desenvolvedores a se aprimorarem constantemente. São utilizadas para agilizar o trabalho dos desenvolvedores e gerar melhoria contínua para os processos.	desenvolver softwares úteis ao mercado e que sejam produzidos envolvendo um conjunto de incrementos que incluam uma nova funcionalidade do sistema.
14. “Métodos utilizados para elaboração de sistemas computacionais dando prioridade a sua coesão, isto porque estes métodos são desenvolvidos a partir de princípios matemáticos que garantem a sua exatidão na capacidade de expressão das ideias vinculadas ao projeto de software.	identificar os requisitos funcionais, que definem o que o software deve fazer, e não funcionais, que estabelecem como o software vai executar as suas rotinas.
15. “Modelos evolucionários são caracterizados por serem iterativos e apresentarem características que possibilitem desenvolvermos versões cada vez mais completas do software.	por possibilitar ao profissional da área de Engenharia de Software produzir versões mais abrangentes em relação ao Software.
16. “O desenvolvimento de softwares é uma área que oferece inúmeras ferramentas para potencializar o desempenho dos profissionais envolvidos em seus processos. E um dos fatores que desponta como um grande diferencial é o Rapid Application Development, ou RAD. O método vem sendo amplamente aderido e pode revolucionar os processos de desenvolvimento em uma empresa.	5, 3, 1, 2, 4.
17. “O paradigma Técnicas de Quarta Geração (4GT) da engenharia de software concentra-se na capacidade de se especificar software a uma máquina em um nível que esteja próximo à linguagem natural ou de se usar uma notação que comunique uma função significativa. ”RUP – RATIONAL UNIFIED PROCESS. 4GL: O Que São as Técnicas de 4ª Geração. Disponível em: < https://tec4rup.wordpress.com/2010/05/13/o-que-sao-as-tecnicas-de-4%C2%AA-geracao/ >. Acesso em 31 Mar. 2019.O trecho acima traz a definição de das técnicas de 4ª Geração. A partir dessas informações e do conteúdo estudado, é possível afirmar que o modelo 4 GT se subdivide, sequencialmente, em:	Obtenção dos requisitos, Estratégia do projeto, Implementação usando 4 GL, Testes.

PERGUNTAS	RESPOSTAS
18. “O solicitante do software computacional pode ser uma pessoa da mesa do lado. Em muitos casos, acredita em mitos sobre software porque gerentes e profissionais da área pouco fazem para corrigir falsas informações. Mitos conduzem às falsas expectativas e, em uma última instância, à insatisfação com o desenvolvedor.	Mito dos Clientes.
19. “Os paradigmas da Engenharia de Software frequentemente são descritos como abordagens alternativas à engenharia de Software em vez de abordagens complementares. Em muitos casos, porém, os paradigmas podem e devem ser combinados de forma que as potencialidades de cada um possam ser obtidas num único projeto.	modelo cascata, prototipação, espiral e 4GT.
20. “Scrum é uma metodologia ágil para gestão e planejamento de projetos de software. No Scrum, os projetos são divididos em ciclos (tipicamente mensais) chamados de Sprints. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado.	Planejamento Geral, Ciclos de Sprint, Encerramento do projeto.
21. “Uma forma de utilizar o modelo cascata é quando precisamos fazer adaptações ou aperfeiçoamentos em um sistema já existente. Por exemplo, quando temos um sistema já pronto e precisamos fazer uma adaptação porque alguma lei governamental foi alterada ou criada”.Fonte: DEVMEDIA. Introdução ao Modelo Cascata. Disponível em: < https://www.devmedia.com.br/introducao-ao-modelo-cascata/29843 >. Acesso em 31 mar. 2019.O trecho acima trata de uma das utilizações do método cascata. Baseado no conteúdo e no trecho citado, podemos afirmar que o modelo Cascata também se caracteriza por:	ser um modelo que se origina de uma abordagem sistemática, sequencial ao desenvolvimento do software.
22. “XP é um apelido carinhoso de uma nova metodologia de desenvolvimento designada Extreme Programming, com foco em agilidade de equipes e qualidade de projetos, apoiada em valores como simplicidade, comunicação, feedback e coragem que nos submetem ao reconhecimento de que XP é uma metodologia baseada em comportamentos e atitudes.	Seleção de estórias, Divisão em Tarefas, Planejamento de Release, Desenvolvimento do Software, Liberação do software e Avaliação do sistema.
23. A abordagem Big Bang testa tudo; ela conta com uma sequência calculada por algoritmos internos, sendo vantajosa para ter agilidade ao testar o programa. Avaliando as desvantagens, percebe-se que nesta abordagem há dificuldade de se achar a causa do problema, dificuldade de se achar o troubleshooting, isto é, a localização do erro.Considerando essas informações e de acordo com o conteúdo estudado, é correto afirmar que a abordagem Big Bang testa:	todas as classes ao mesmo tempo, de forma unitária e eficiente.
24. A atividade de teste de software consiste em uma das etapas do desenvolvimento cujo objetivo primordial é avaliar a possibilidade e a existência de erros no sistema, para que então possam ser solucionados ou evitados.Considerando essas informações e o conteúdo estudado sobre teste de sistema, pode-se afirmar que esse tipo de teste:	é realizado em ambientes reais e com usuários reais ou representativos, podendo acontecer dentro da própria empresa.
25. A atividade de teste de software consiste em uma das etapas do desenvolvimento de um software, cujo objetivo primordial é avaliar a possibilidade e a existência de erros no sistema, para que então possam ser solucionados ou evitados.Considerando essas informações e o conteúdo estudado sobre teste de software, analise as afirmativas a seguir:I. Os métodos de desenvolvimento e complexidade dos softwares são altamente passíveis de erros.II. Os erros podem ocorrer devido a problemas na especificação dos requisitos, na modelagem de negócio, no modo pelo qual a funcionalidade deve ser desempenhada, na complexidade do sistema e na mudança de requisitos.III. Os desenvolvedores estão isentos de cometer erros de programação devido à detecção de erros das ferramentas modernas.IV. O teste de software é realizado de diversas maneiras e usando diversas metodologias.Está correto apenas o que se afirma em:	I, II e IV.
26.	

PERGUNTAS	RESPOSTAS
27. A base teórica do CMM são os princípios da qualidade total, que é definida pela ISO 8402 como uma abordagem de gerenciamento para uma organização centrada na qualidade, baseada na participação de todos os seus membros, visando o sucesso a longo prazo por meio da participação do cliente e dos benefícios para os membros da organização e da sociedade. Nesse modelo, as organizações de software são enquadradas em um dos cinco níveis de maturidade. Considerando essas informações e o conteúdo estudado sobre o nível de maturidade “Definido” do método CMMI, pode-se afirmar que ele:	tem foco no desenvolvimento de requisitos, solução técnica, integração do produto, verificação e validação, foco no processo organizacional.
28. A demanda por qualidade de software tem motivado as empresas de software para o desenvolvimento com qualidade. Um software de qualidade é fácil de usar, funciona corretamente, é de fácil manutenção e que mantém a integridade dos dados para evitar possíveis falhas. Qualidade de software é dividida em categorias, entre elas está a manutenibilidade, nesta categoria são analisadas se há facilidade para se fazer correções, nas atualizações e alterações do produto. Considerando essas informações e o conteúdo estudado em relação ao atributo de Estabilidade, pode-se afirmar que:	o atributo de estabilidade identifica os riscos de falhas quando se faz alterações ou manutenções no código ou mesmo remoção de defeitos.
29. A Engenharia de Software se dedica a melhorar os processos de desenvolvimento. A qualidade é uma área de conhecimento que está dentro da Engenharia de Software e que deve ser colocada em prática quando se desenvolve sistemas, mesmo com todas as dificuldades de entregar dentro do prazo, dentro do escopo e dentro do custo planejado e acordado com o cliente. Considerando essas informações e o conteúdo estudado sobre de qualidade de um software, analise as afirmativas a seguir.I. Qualidade de um software significa desenvolver um sistema sem defeitos.II. É muito comum encontrar softwares perfeitos, dificilmente eles precisam ser melhorados, pois são reconhecidos como um produto de boa qualidade.III. A qualidade de um sistema deve atender as necessidades do cliente, deve ser fácil de usar e funcionar corretamente e de fácil manutenção.IV. Qualidade de Software é um processo das etapas do desenvolvimento de um sistema com o objetivo de garantir a conformidade dos processos, prevenindo e eliminando falhas. Está correto apenas o que se afirma em:	III e IV.
30. A etapa de teste de software busca verificar se o sistema se comporta de acordo com o especificado nos requisitos levantados junto ao cliente, reduzindo a probabilidade de erros quando o sistema estiver em produção. O teste de software busca a execução de um determinado sistema para avaliar se este alcançou os objetivos propostos, como também se processa corretamente para o seu fim específico. Considerando essas informações e o conteúdo estudado sobre teste de aceitação, pode-se afirmar que:	neste teste, um grupo de usuários simula operações de rotina do sistema verificando se o levantamento de requisitos está de acordo.
31. A inspeção pode ser feita tanto em produtos de software como em projetos de software, dependendo do aspecto que será analisado durante a revisão. De acordo com os aspectos analisados, podem ser classificados dois tipos básicos de revisão: a inspeção de código-fonte tem o intuito de encontrar defeitos no código-fonte, realizando uma análise estática do código; e a inspeção de documentos de requisitos analisa documentos de requisitos em busca de erros. A inspeção em documentos de requisitos pode revelar inúmeros defeitos. Considerando essas informações e o conteúdo estudado sobre defeito de omissão, pode-se afirmar que:	informações necessárias ao sistema são omitidas, como a falta de uma funcionalidade ou da capacidade de desempenho do sistema.
32. A ISO/IEC 9126 (NBR13596) define que a qualidade de software deve ser dividida em seis amplas categorias que têm como objetivo servir base na avaliação de produto de software, na qual o gestor de projetos deve utilizar de mecanismos para verificar seus atributos no software que será entregue. Diante dessas informações e o conteúdo estudado sobre qualidade de software, é correto afirmar que a categoria de funcionalidade:	descreve todas as funções que satisfazem as necessidades às quais se destina o produto, de acordo com levantamento de requisitos.
33.	

PERGUNTAS	RESPOSTAS
34. A ISO/IEC 9126 (NBR13596) define que a qualidade de software deve ser dividida em seis amplas categorias. A portabilidade é a categoria que mede como o sistema pode ser transferido e adaptado em outros ambientes e infraestruturas. Fonte: ISO/IEC DIS. 14598. Information Technology Evaluation of software product Part5: Process for evaluators. Disponível em: https://www.iso.org/standard/24906.html . Acesso em: 09/10/2019.Considerando essas informações e o conteúdo estudado em relação ao atributo de Adaptabilidade, pode-se afirmar que:	o atributo de adaptabilidade representa a forma que o sistema vai poder ser utilizado em diferentes sistemas operacionais, banco de dados e hardware.
35. A metodologia CMMI foi desenvolvida com o intuito de definir um ponto inicial para modelos integrados, aprimorar as melhores práticas para a criação de modelos baseados em lições aprendidas, estabelecer um framework que possibilite a integração futura de novos modelos e para a criação de uma forma associada de avaliação de desempenho e treinamento de produtos. Esse método é dividido em 5 níveis.Considerando essas informações e os conteúdos estudados sobre a metodologia CMMI, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F a(s) falsa(s):I. () No nível 1 – Inicial, as atividades de gerenciamento são documentadas, padronizadas e integradas em processos padrão para a organização.II. () No nível 2 – Repetitivo, os processos básicos de gerenciamento de projeto são determinados para controlar custos, cronogramas e funcionalidade.III. () No nível 3 – Definido, o processo de software é desorganizado, podendo até mesmo ser caótico, quando possui poucos processos definidos, cujo sucesso depende dos esforços individuais.IV. () No nível 4 – Gerenciado, são efetuadas medições detalhadas do processo de software e qualidade do produto e controlados quantitativamente.V. () No nível 5 – Otimizado, a melhoria contínua do processo é possibilitada pela realimentação quantitativa do processo e conduzida a partir de ideias novas tecnologias.Agora assinale a alternativa que apresenta a sequência correta:	F, V, F, V, V.
36. A norma IEEE 829 é uma empresa dos EUA criadora de conhecimentos na área tecnológica; ela criou oito passos básicos para explicar e nortear o gerenciamento de testes e para um perfeito funcionamento de determinados softwares.Considerando essas informações e o conteúdo estudado sobre planejamento e estimativa de teste, analise as frases a seguir e associe-as com suas respectivas partes correspondentes:1) Especificação do projeto.2) Plano de teste.3) Especificação do caso de teste.4) Relatório do sumário de teste.() O líder elabora cronogramas e define atribuições.() A empresa avalia todos os testes feitos e toma decisões para futuros testes.() O líder define os tipos de teste a serem realizados.() Neste estágio, o líder de teste delimita os casos a testar.Agora, assinale a alternativa que apresenta a sequência correta:	2, 4, 3, 1.
37. A partir do momento em que se entende o que é qualidade, a Software Quality Assurance (SQA) deve criar um Plano de Garantia da Qualidade de Software para definir as estratégias de uma equipe de software, a fim de garantir de qualidade de software. Fonte: PRESSMAN, R. S. Engenharia de Software: Uma abordagem Profissional. 7.ed Porto Alegre: McGraw-Hill, 2011. Considerando essas informações e o conteúdo estudado relacionado às funções da garantia de qualidade de software, é correto afirmar que é uma dessas funções:	verificar se os processos de teste, inspeção e auditoria estão sendo executados.
38. A Qualidade de software definida pela ISO/IEC 9126 (NBR13596) é dividida em seis categorias, entre elas está a confiabilidade, que descreve a capacidade de tolerância a falhas, ou seja, se o sistema consegue garantir que o desempenho do dele se manterá dentro do esperado. Fonte: ISO/IEC DIS. 14598. Information Technology Evaluation of software product Part5: Process for evaluators. Disponível em: https://www.iso.org/standard/24906.html . Acesso em: 09/10/2019.Considerando essas informações e o conteúdo estudado em relação ao atributo de Maturidade, pode-se afirmar que:	o atributo de maturidade demonstra a capacidade de estar e permanecer livre de falhas.

PERGUNTAS	RESPOSTAS
39. A qualidade de software está dividida em categorias, entre elas está a categoria de funcionalidade. Nela é descrita todas as funções que satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto, conforme o que foi estabelecido no levantamento de requisitos. Considerando essas informações e o conteúdo estudado em relação ao atributo de Segurança de acesso, pode-se afirmar que:	o atributo de segurança de acesso impede que os acessos não autorizados, acidental ou deliberado, por programas externos.
40. A qualidade é a adequação dentro das normas para atender às exigências definidas pela organização em conformidade com a necessidade do cliente, interessado em um processo, serviço ou produto que será entregue pela sua empresa. Um software de qualidade deve atender seis categorias. Eficiência é uma destas categorias. Diante dessas informações e do conteúdo estudado sobre qualidade de software, pode-se afirmar a respeito da categoria eficiência que:	essa categoria se preocupa em medir o consumo dos recursos de capacidade física e analisa se os recursos e os tempos utilizados são compatíveis com o nível de desempenho.
41. A recuperação é a capacidade de reiniciar operações após a perda da integridade de uma aplicação. Ela garante a continuidade das operações após qualquer imprevisto, avaliando se ocorre salvamento de dados caso aconteça queda de energia ou de rede e se o programa possui a capacidade de salvar as últimas ações feitas. De acordo com o conteúdo estudado sobre falhas e recuperação, analise as afirmativas a seguir. I. É preciso avaliar se ocorre sincronização do aplicativo com outras redes. II. Deve-se avaliar se acontece a proteção do sistema, a partir dos testes e técnicas de desligamento do aparelho em processo de utilização. III. É necessário definir uma navegação intuitiva e completa. IV. O teste ocorre por tentativas de invasão no banco de dados, propositalmente causadas por hackers. Está correto apenas o que se afirma em:	I e II.
42. A tabela de risco é composta por três colunas: na primeira, há os erros; na segunda, os níveis de erros; e na terceira, os grupos heurísticos. Os seus indicadores são: risco alto, risco médio e risco baixo. Esses dados são a métrica relacionada ao software, capaz de ajudar a prevenir problemas de falha geral. Considerando essas informações e o conteúdo estudado em métricas relacionadas ao software, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s). I. () Quando um erro é encontrado várias vezes no programa, ele pertence à categoria popular e pode ser de risco médio e alto. II. () A categoria “completo” é quando o erro acontece sempre e pode ser de risco alto. III. () “Falhou recentemente” é um índice de severidade do programa e denota risco alto. IV. () Quando o erro pode prejudicar o programa e destruir seu funcionamento, é “crítico” e pode ser de risco médio e alto. Agora, assinale a alternativa que apresenta a sequência correta:	V, F, F, V.
43. A utilização de métricas tem sido uma ferramenta essencial na produção de um sistema. Ela se torna importante, pois é possível medir as características com objetivo de descobrir se os requisitos foram atendidos e se possui um grau de qualidade satisfatório. Alguns tipos de métricas que podem ser utilizadas na produção do produto. Diante dessas informações e do conteúdo estudado sobre métricas, é correto afirmar que a métrica de implantação:	mede o esforço médio de cada tarefa implantação como: configuração do sistema, instalação do servidor, do banco de dados e de bibliotecas.
44. A utilização de métricas tem sido uma ferramenta essencial na produção de um sistema. Elas se tornaram importantes porque com elas se consegue medir as características com objetivo de descobrir se os requisitos foram atendidos e se o produto possui um grau de qualidade satisfatório. Há alguns tipos distintos de métricas que podem ser utilizados na produção do produto. Diante dessas informações e o conteúdo estudado sobre métricas, é correto afirmar que a métrica de requisitos:	verifica se o software atendeu às necessidades solicitadas pelo cliente. Nessa medição, são levadas em consideração as condições levantadas inicialmente e o que é apresentado na interface do sistema.

PERGUNTAS	RESPOSTAS
45. Analise a figura a seguir: Testar software é uma atividade crítica para a qualidade do produto final de software, pois representa a revisão final da especificação, projeto e geração de código. Mesmo que durante todo o processo de desenvolvimento de software sejam utilizados métodos, técnicas e ferramentas a fim de evitar que erros sejam introduzidos no produto, a atividade de teste é de fundamental importância para eliminar os erros que persistem. Para que seja possível obter sucesso na atividade de teste, é imprescindível que um planejamento seja realizado. A atividade de teste é planejada de acordo com as etapas do ciclo de vida do software, em que cada nível de teste corresponde a uma etapa. De acordo com a figura e o texto apresentados, assinale a alternativa que corresponde, respectivamente, aos itens #1, #2 e #3 da figura:	Especificação de requisitos, codificação, teste de sistema.
46. Apesar dos modelos aplicados na engenharia de software para garantir a qualidade, o principal objetivo de qualidade de um software é garantir um produto final satisfaça as expectativas do cliente dentro daquilo que foi acordado inicialmente e detalhado no levantamento dos requisitos. Considerando essas informações e o conteúdo estudado sobre controle de qualidade de um projeto de software, analise as afirmativas a seguir: I. O Controle de qualidade deve ser realizado antes da entrega ao cliente. II. O Controle de qualidade deve ser realizado em todas as fases do ciclo de vida de um software. III. O Controle de qualidade deve ser realizado em projetos de grande porte. IV. O Controle de qualidade, por ser muito trabalhoso, deve ser realizado nas fases finais. Está correto apenas o que se afirma em:	I e II.
47. As inspeções podem fazer parte dos processos de verificação e validação de software. Elas complementam os testes, pois não exigem que o programa seja executado. Isso significa que podem ser verificadas versões incompletas do sistema e que representações, tais como modelos UML, podem ser checados. Considerando essas informações e o conteúdo estudado sobre o analista de teste no processo de inspeção, pode-se afirmar que este profissional:	é responsável por trabalhar juntamente com o cliente na realização de testes de aceitação e verificação de problemas.
48. As inspeções podem fazer parte dos processos de verificação e validação de software. Elas complementam os testes, pois não exigem que o programa seja executado. Isso significa que podem ser verificadas versões incompletas do sistema e que representações, tais como modelos UML, podem ser checados. Considerando essas informações e o conteúdo estudado, pode-se afirmar, no caso de defeito de fato incorreto, que:	há informações contraditórias com o domínio da aplicação.
49. As inspeções têm como foco central o código-fonte de um sistema, mas toda e qualquer representação legível do software, tais como seus requisitos ou modelo de projeto, pode e deve ser inspecionada. Durante o inspecionamento de um sistema são utilizados os conhecimentos sobre o sistema, o domínio da aplicação e a linguagem de programação ou modelagem para descobrir possíveis erros. Considerando essas informações e o conteúdo estudado acerca das vantagens da inspeção de software sobre os testes, analise as afirmativas a seguir: I. Durante o teste, erros podem esconder outros erros. Como a inspeção não é preciso se preocupar com as interações entre os erros. II. Caso um programa esteja incompleto, é necessário desenvolver dispositivos de teste para testar as partes, aumentando o custo. III. As inspeções são mais eficazes na descoberta de defeitos do que os testes de programa. IV. As inspeções podem substituir os testes de software. Está correto apenas o que se afirma em:	I, II e III.

PERGUNTAS	RESPOSTAS
50. As métricas são as medidas brutas, atômicas e de composição simples, como formatos de valores e quantidades. Não são indicadas para serem usadas como base em tomadas de decisões estratégicas, já que mais fazem a medição do que de fato apontam algum resultado concreto. Por outro lado, a utilização de métricas tem sido uma ferramenta essencial na produção de um software, pois se consegue medir as características com base na qualidade. Diante dessas informações e do conteúdo estudado sobre métricas, é correto afirmar que a Métrica de Performance:	mede em milissegundos o tempo que leva para executar ou abrir uma determinada funcionalidade do sistema através de ferramentas automatizadas.
51. As normas são delimitadas pelo seu alcance geográfico, político ou econômico. De maneira organizada, diversos países participam da fundação, elaboração, aprovação e divulgação dessas diretrizes. Uma pirâmide comumente é utilizada para representar os níveis de normatização. Considerando essas informações e os conteúdos estudados sobre os níveis de normatização, analise as afirmativas a seguir. I. Nível internacional é a abrangência mundial estabelecida pela Organização Internacional de Normalização. II. O nível regional é formado pelas entidades técnicas que se desenvolvem com seus associados para facilitar transações entre si. III. O nível nacional elabora normas pelas partes de interesse que são geridas por um organismo nacional de normalização. IV. O nível empresarial consiste nas normas implementadas por um conjunto de empresas, com o intuito de orientar a fabricação, compra e venda de suas operações. V. O nível de associação é estabelecido por um grupo de países ou região geográfica ou política, que é representado por organismos regionais. Está correto apenas o que se afirma em:	I, III e IV.
52. As normativas ISO que definem a qualidade de software estão definidas na ISO/IEC 9126. No Brasil, essas normas estão definidas por meio da ABNT, que organiza as comissões de estudo que se baseiam nas normas estabelecidas pela ISO. Fonte: ISO/IEC DIS. 14598. Information Technology Evaluation of software product Part5: Process for evaluators. Disponível em: https://www.iso.org/standard/24906.html . Acesso em: 09/10/2019. Em relação ao conjunto de normas estabelecido pela ISO 9126, analise as afirmativas a seguir: I. A ISO 9126 estabelece um conjunto de normas para: processos, produtos e qualidade em uso. II. Para a ISO 9126, produto é onde estão definidas as normas, das etapas que compõem o desenvolvimento de um software. III. Para a ISO 9126, é no processo que estão definidos os atributos que compõem a qualidade do software. IV. Para a ISO 9126, é na qualidade em uso que são definidas a aferição da qualidade na visão do usuário e a facilidade de uso e operação do sistema. Está correto apenas o que se afirma em:	I e IV.
53. As revisões não têm como finalidade apenas a validação da conformidade com os padrões, têm utilidade em ajudar a descobrir problemas e omissões no software ou em sua documentação de projeto, envolvendo um exame do software, de sua documentação e de seus registros de processos, a fim de descobrir erros e omissões, e verificar se foram seguidos os padrões de qualidade. Considerando essas informações e o conteúdo estudado sobre inspeção, pode-se afirmar que:	as inspeções de programa são “revisões em pares” em que os membros da equipe colaboram para encontrar bugs no programa que está sendo desenvolvido.
54. As vantagens de uma equipe de desenvolvimento de software prezar pela qualidade em todas as atividades do desenvolvimento são diminuição do retrabalho; redução da quantidade de reformulações e consequentemente, queda no custo e no tempo para se produzir um software. A Software Quality Assurance (SQA) lida com os processos de desenvolvimento do software que envolve algumas atividades definidas por Pressman, (2011). Fonte: PRESSMAN, R. S. Engenharia de Software: Uma abordagem Profissional. 7.ed Porto Alegre: McGraw-Hill, 2011. Considerando essas informações e o conteúdo estudado, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s)	V, V, V, F.

PERGUNTAS	RESPOSTAS
<p>55. Asserts são usados tanto em QUnit quanto no Selenium, e servem para testar uma condição – são aceitos nas linguagens Java, C#, C++, HTML, Javascript, PHP, entre outras – e testar muitas funções diferentes. Aliás, eles têm as características de ser escaláveis, ou seja, pode-se criar novos asserts.Considerando essas informações e o conteúdo estudado em métricas relacionadas ao software, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s).I. <input type="checkbox"/> assertChecked: o teste funciona apenas se uma caixa de seleção tipo escolha estiver marcada.II. <input type="checkbox"/> assertEditable: avalia se o texto do campo de texto pode ser editável.III. <input type="checkbox"/> click: verifica se o elemento pode ser alterado.IV. <input type="checkbox"/> deleteAllCookies: Verifica se o botão fecha cookies da página.Agora, assinale a alternativa que apresenta a sequência correta:</p>	V, V, F, F.
<p>56. Capability Maturity Model Integration (CMMI) ou Ciclo de Vida do Desenvolvimento de Software, CMMI é um modelo de boas práticas de qualidade e maturidade no processo de desenvolvimento de um software. Ele foi criado pelo Software Engineering Institute (SEI) e fomentado pelo departamento de defesa dos Estados Unidos.Analise os estágios do CMMI a seguir e associe-as com suas respectivas características:1) Nível 1 – Inicial.2) Nível 2 – Gerenciado.3) Nível 3 – Definido.4) Nível 4 - Quantitativamente Gerenciado.5) Nível 5 – Otimização.() Políticas e procedimentos de gerenciamento e desenvolvimento de software estão definidas e são exercidas.() Organização imatura, processos improvisados, pouco compromisso com prazos ou custos. Os conhecimentos não pertencem aos projetos, mas às pessoas. As chances de desenvolvimento estão à mercê das habilidades pessoais dos gerentes.() Engajamento na melhoria contínua de processos e ações preventivas de problemas. () Processos estabelecidos e padronizados, pertencentes à organização e não a uma única pessoa.() Existem metas quantitativas para processos, medidas de qualidade e controle estatístico.Agora, assinale a sequência correta:</p>	2, 1, 5, 3, 4.
<p>57. Conceitualmente, podemos explicar métrica como um conjunto de regras e/ou formas diferenciais que definem espaço e distância entre formas. Os problemas que ocorrem na criação de softwares e seus processos de web-desenvolvimento são experiências às quais as empresas estão sujeitas.Considerando essas informações e de acordo com o conteúdo estudado sobre métricas para o teste de software, analise as afirmativas a seguir.I. Programas com funcionamento bom garantem a qualidade do software.II. São métricas avaliativas: profundidade, quantidade e severidade.III. Métrica de profundidade tem a capacidade de testar o banco de dados.IV. Na métrica de quantidade, o testador tem o objetivo de criar indicativos de erros.Está correto apenas o que se afirma em:</p>	II e III.
<p>58. Durante uma inspeção, frequentemente se usa um checklist de erros comuns de programação para ajudar na busca de bugs. Esse checklist pode basear-se em exemplos de livros ou no conhecimento de defeitos comuns em um domínio de aplicação específico. Diferentes checklists são usados para diferentes linguagens de programação, pois cada linguagem tem seus próprios erros característicos.Considerando essas informações e o conteúdo estudado sobre inspeção, analise os itens a seguir sobre as classes de defeitos e associe-os com suas respectivas verificações:1) Defeitos de dados.2) Defeitos de controle.3) Defeitos de entrada/saída.4) Defeitos de interface.5) Defeitos de gerenciamento de armazenamento.6) Defeito de gerenciamento de exceção.() Verifica se cada instrução condicional está correta.() Verifica se as entradas inesperadas podem causar corrupção de dados.() Verifica se o armazenamento dinâmico é usado e se o espaço foi alocado corretamente.() Verifica se todas as variáveis de programa são iniciadas antes que seus valores sejam usados.</p>	2, 3, 5, 1, 4, 6.

PERGUNTAS	RESPOSTAS
59. É através das métricas que você será capaz de avaliar o andamento de uma campanha, identificar tendências sejam elas de consumo, interação, navegação e para obter e comparar resultados. A utilização de métricas se torna importante na produção de sistema, pois se consegue medir as características do grau de qualidade satisfatório. Diante dessas informações e do conteúdo estudado sobre métricas, é correto afirmar que a Métrica de Banco de Dados:	avalia o banco de dados físico e lógico, ajuda saber com qual período é necessário para a realização de um backup e o tamanho desse backup num determinado tempo de uso do sistema.
60. Eficiência é a palavra-chave para todas as pessoas em todas as empresas, quer queiram, quer não. Com base nisso, o programa Selenium ajuda a agrupar processos iguais em poucos cliques, para testar vários campos de uma só vez. Considerando essas informações e o conteúdo estudado sobre Assert not Text, é correto afirmar que:	testa a obrigatoriedade de inserção de texto.
61. Em diversos produtos e serviços que as pessoas comprem diariamente existem padrões estabelecidos por meio de estudos, testes e aferições, realizados pelos órgãos reguladores, como a ISO (Organização Internacional de Normalização). As normativas da qualidade de software estão definidas na ISO/IEC 9126. Considerando essas informações e os conteúdos estudados sobre o conjunto de normas que a ISO 9126 estabelece, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s). I. () Nos processos, estão definidas as normas das etapas que compõem o desenvolvimento de um software. II. () Nos produtos, estão definidos os atributos que compõem a qualidade do software, podendo ser divididos em: internos e externos. III. () Há uma padronização dos parafusos e roscas que irão compor o hardware. IV. () A partir da qualidade em uso, são aferidas a qualidade na visão do usuário e a facilidade de uso e operação do sistema. Agora assinale a alternativa que apresenta a sequência correta:	V, V, F, V.
62. Em diversos produtos e serviços que as pessoas comprem diariamente, seja uma roupa de marca, um aparelho celular ou um simples brinquedo eletrônico, existem padrões estabelecidos por meio de estudos, testes e aferições, realizados pelos órgãos reguladores. Na área de desenvolvimento de software, a organização responsável por estabelecer as normas para a qualidade de software é chamada de ISO. Em relação aos princípios de qualidade, aquele que está diretamente ligado ao tempo de resposta de processamento e aos recursos utilizados no sistema é conhecido como:	eficiência.
63. Empresas que adotam as metodologias ágeis, utilizam métricas para obter um entendimento mais amplo do processo de desenvolvimento do software, para melhorar os acertos, para medir o desempenho e a eficiência das soluções. Diante dessas informações e do conteúdo estudado, é correto sobre Sprint Burndown que:	essa métrica ajuda no monitoramento e na finalização do trabalho ao longo da sprint. O objetivo é ter todo o trabalho que foi previsto, concluído.
64. Especificamente para a melhoria de processo de software, têm sido utilizados os modelos e normas: CMM, CMMI, ISO/IEC 12207 e ISO/IEC 15504. Esses modelos identificam processos fundamentais para a Engenharia de software. Basicamente todos eles identificam, direta ou indiretamente, teste de software com um desses processos. Sabe-se que os testes são fundamentais para a avaliação do software desenvolvido, porém, testá-lo não é uma atividade trivial, e exige conhecimentos, habilidades e infraestrutura específica. Considerando essas informações e o conteúdo estudado sobre os aspectos gerais do sobre os níveis de maturidade do CMMI, analise as afirmativas a seguir. I. O nível 1 – Inicial tem foco no melhoramento contínuo do processo com prevenção de defeitos, gerenciamento de mudanças. II. O nível 2 – Repetitivo tem foco na qualidade do produto e do processo e gerenciamento do processo. III. O nível 3 – Otimizado tem foco nas pessoas competentes e especializadas. IV. O nível 4 – Gerenciado tem foco nos processos de gerenciamento, planejamento, acompanhamento e configuração. V O nível 5 –	I, II e IV.

PERGUNTAS	RESPOSTAS
65. Existem diversos sistemas de medição de teste de software, e também muitos programas que gerenciam os testes medidos, como o TestMaster, o TestComplete e o Klaros Management, por exemplo. Esses programas criam pastas para armazenar testes, e estas pastas são criadas pelo usuário e classificadas por categoria; por exemplo, o testador cria uma pasta para categoria “testes pendentes” e cria outra pasta para a categoria “testes completos”.Considerando essas informações e o conteúdo estudado em métricas relacionadas ao software, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s).I. () As medidas “pendente”, “semipendente” e “concluído” fazem parte da métrica de status.II. () O sumário de ocorrências é gerado para ordenar os problemas do programa.III. () Falha de código pendente é exemplo de erro do sumário de ocorrências.IV. () A navegação intuitiva e completa está no status de medida “concluída”.Agora, assinale a alternativa que apresenta a sequência correta:	V, V, F, F.
66. Geralmente as inspeções enfocam o código-fonte, mas qualquer representação legível do software, como seus requisitos ou um modelo de projeto, pode ser inspecionada. Quando se inspeciona um sistema, você usa o conhecimento do sistema, seu domínio de aplicação e a linguagem de programação ou o modelo de projeto para se descobrir erros.Considerando essas informações e o conteúdo estudado, pode-se afirmar que em casos de defeito de ambiguidade:	a informação leva a múltiplas interpretações.
67. Há um estágio do modelo CMM cujas características são desenvolvimento de requisitos, solução técnica, integração do produto, verificação e validação, foco no processo organizacional, definição do processo organizacional, treinamento organizacional, gerenciamento de riscos, gerenciamento integrado do projeto, análise da decisão e resolução.Considerando essas informações e o conteúdo estudado sobre o modelo CMM, é correto afirmar que as características mencionadas são referentes ao estágio:	definido.
68. Há uma estrutura de modelo CMMI na qual os níveis de capacidade proveem uma ordem de melhoria, abordada em cada área de processo separadamente, que melhor atende aos objetivos de negócio das empresas. Considerando essas informações e os conteúdos estudados sobre CMMI, quanto à abordagem descrita, pode-se afirmar que o tipo de representação descrito se refere à estrutura:	contínua.
69. Hoje em dia, com o avanço web, os sites são criados em HTML5 e decorados com CSS3, e as animações são criadas com JavaScript, inclusive já se fala e já virou moda usar bibliotecas e linguagens como jQuery, Angular, React e outras linguagens destinadas à criação de sites Front-End, que fazem animações.Considerando essas informações e o conteúdo estudado sobre dragAndDropToObject, é correto afirmar que ele pode servir para testar ações como:	arrastar objetos.
70. Inspeções representam um tipo de revisão formal por pares. Trata-se de técnicas de análise para avaliação de forma, estrutura e conteúdo de um documento, código-fonte ou outro produto de trabalho. Essa técnica é realizada por um grupo de pessoas que têm o mesmo perfil, a fim de identificar discrepâncias do produto com base em padrões e especificações.Considerando essas informações e o conteúdo estudado sobre defeito de inconsistência, pode-se afirmar que, nesses casos:	a informação aparece de forma diferente em cada aparição, causando incoerência.
71. ISO é uma organização independente e não governamental, fundada em 1947. Foi criada com o objetivo de desenvolver e promover normas internacionais para produtos e serviços, que possam ser usadas por todos os países.Considerando essas informações e os conteúdos estudados em relação a ISO, é correto afirmar que:	a ISO faz normas a respeito de tudo e essas normas se tornaram a base para especificar produtos, organizar o fornecimento de serviços e até mesmo para a elaboração de legislação em vários países.

PERGUNTAS	RESPOSTAS
72. É a distribuição de serviços de computação – servidores, armazenamento, bancos de dados, redes, software, análises, inteligência e muito mais pela Internet, proporcionando inovações mais rápidas, recursos flexíveis e economia na escala.	Computação em Nuvem.
73. o CMM vem sendo usado para tudo, desde o desenvolvimento de softwares, onde teve origem, até a entrega de produtos e a construção de arranha-céus. Infelizmente, como acontece com a Biblioteca de infraestrutura de tecnologia da informação (ITIL), a arquitetura orientada para serviços (SOA) e o Six Sigma, acredita que o CMM acaba sendo subaproveitado.”ADATO, Leon. Data Center: Como saber se “bom o suficiente” é suficientemente bom. Disponível em: < https://cio.com.br/data-center-como-saber-se-201cbom-o-suficiente201d-e-suficientemente-bom/ >. Acesso em: 13 abr. 2019.O trecho trata das funcionalidades do CMM. A partir dessas informações e do conteúdo estudado, é possível afirmar que as fases da maturidade consistem em:	implementar sistemas baseados nas experiências do setor de recursos humanos e, por fim, perceber a aplicação de um processo flexível e com aspecto organizado.
74. A gestão de qualidade pessoal tem como objetivo auxiliar o engenheiro de software a trabalhar de forma real e objetiva com os defeitos que ele próprio insere no programa. Estes níveis adicionam métodos para gestão da qualidade ao PSP.	apresenta uma forma de promover integração entre a equipe de desenvolvimento e de testes.
75. A integração não é um ‘big bang’ no final do projeto; em vez disso, os elementos são integrados progressivamente. O sistema é montado por meio de várias integrações, aproximadamente entre seis e nove, que ocorrem durante as interações. Como muitos riscos do projeto geralmente estão associados às integrações entre os componentes e subsistemas, este mecanismo permite aumentar significativamente as chances do sucesso.	no big bang os objetos são conexos progressivamente com várias integrações.
76. A realização do processo de automação requer ferramentas que possibilitem a interação entre as rotinas automatizadas e os softwares a serem testados. Também requer que essas ferramentas possuam recursos de conferências automáticas.	são constituídos por comandos que simulam as entradas no sistema, além das análises, estabelecendo comparações com as saídas.
77. Além dos tipos convencionais de testes de sistema, outros testes podem ser elaborados no projeto de software. O teste de desempenho é idealizado para avaliar o desempenho do software de forma isolada e dentro do contexto de um sistema integrado.	verificar, no mínimo, um caso de teste, identificado um para cada sentença dentro da especificação suplementar ou para caso de uso crítico.
78. Alguns exemplos de indicadores de processo incluem quantidade de erros descobertos antes da entrega do software, defeitos entregues aos usuários finais, produtos de trabalho entregues, esforço humano despendido, tempo gasto, dentre outros.	uma disparidade entre a solicitação do usuário e o desempenho apresentado pela aplicação executável.
79. Apesar de existirem várias linguagens de programação, o Java e o C continuam a ser as linguagens de programação mais populares do mundo. Nos últimos anos têm existido algumas variações em termos de ranking, mas estas linguagens têm-se mantido nas duas primeiras posições e isso significa muita coisa.	I e III.
80. As métricas de qualidade não devem apenas informar, mas também orientar sobre como realizar a melhoria da qualidade do software, mostrando o que fazer primeiro, como fazê-lo, próximos passos etc.”SUBRAMANYAM, Jitendra. Cinco métricas para avaliar a qualidade de seus softwares. Disponível em: https://computerworld.com.br/2015/09/09/cinco-metricas-para-avaliar-qualidade-de-seus-softwares/ >. Acesso em 13 abr. 2019.O trecho acima retrata a finalidade do uso das métricas de qualidade. A partir dessas informações e do conteúdo apresentado, é possível verificar que as métricas de produto são basicamente avaliadas por meio de duas classes. São elas:	Métricas Dinâmicas e Estáticas.
81. Chama-se Gerenciamento de risco o meio pelo qual estas incertezas são sistematicamente gerenciadas, permitindo avaliar e enfrentar os riscos, evitando o que pode ser evitado, controlando o que pode ser controlado e minimizando a imprevisibilidade.	é uma ferramenta executada de maneira automatizada, com a função de auxiliar a gestão de defeitos.

PERGUNTAS	RESPOSTAS
82. Como o foco DevOps está exatamente na qualidade, o teste automatizado é um dos seus principais pilares. Neste sentido, o movimento DevOps propõe então que a estratégia automatizada substitua os testes manuais, não garante que o desenvolvedor realize testes com a cobertura necessária para a qualidade do software.	custo inferior.
83. Dentro desta ótica é que a gestão de riscos do projeto de testes deve ser tratada. Sendo descobrir/corrigir defeitos e melhorar a qualidade do produto final os objetivos de testar o software, o exercício de gestão dos riscos é um instrumento.	o TestLink tem como uma das melhores funções o gerenciamento de testes.
84. É importante medir a qualidade do software, mas é igualmente importante executar a atividade de forma correta. Essa ação é muito útil no desenvolvimento de software, mas, muitas vezes, é melhor não ter medição alguma do que contar com uma errada.”SUBRAMANYAM, Jitendra. Como medir com eficiência a qualidade de software na sua empresa. Disponível em: < https://computerworld.com.br/2014/05/13/como-medir-com-eficiencia-a-qualidade-de-software-na-sua-empresa/ >. Acesso em: 12 abr.2019.O trecho acima cita a importância de o software ser executado de maneira correta para evitar problemas. Baseado no conteúdo e no trecho citado, a Garantia da Qualidade é aplicada ao longo do processo de Engenharia de Software e abrange:	ações que visam adequar-se aos modelos de desenvolvimento de Software.
85. É interessante possuir uma suíte de testes voltada à navegação através das principais rotas da aplicação, tal como navegação por menus, para garantir que quando um usuário real estiver usando tal aplicação, será direcionado para a página correta.	apresentar informações anteriores em relação ao fluxo de casos de teste que serão executados.
86. É muito importante que os erros nos sistemas sejam descobertos o quanto antes para diminuir os prejuízos às empresas. Defeitos encontrados nas fases iniciais de desenvolvimento chegam a ser dez vezes mais baratos de serem corrigidos do que aqueles encontrados na etapa de produção, quando clientes reais encontram o bug.	gerar relatórios com base em dados para verificar a evolução dos testes e a qualidade do sistema.
87. Em um ambiente no qual as entregas são rápidas e o produto é tratado de maneira mais dinâmica do que nunca, dessa forma, os profissionais de testes precisam garantir que, além do produto ser entregue com qualidade, que todo o processo de desenvolvimento siga tanto os critérios de SQA (Software Quality Assurance) como de Agile”.IT FORUM 365. O perfil multidisciplinar do analista de teste. Disponível em: < https://itforum365.com.br/o-perfil-multidisciplinar-do-analista-de-testes/ >. Acesso em: 12 abr. 2019.O trecho acima trata dos critérios de qualidade e rapidez na entrega adotada pelo SQA. A partir dessas informações e do conteúdo estudado, é possível afirmar que a mensuração, uma das fases da atividade SQA, é utilizada com a finalidade de:	acompanhar a qualidade do software e avaliar os efeitos das alterações na metodologia e nos procedimentos.
88. Entretanto, uma vez que exista código executável, torna-se possível executar casos de teste, isto é, realizar testes baseados em execução. Não obstante, a existência do código não exclui a possibilidade de testes que não se baseiam em execução.	nas regras estabelecidas que possam apresentar uma manutenção com alto custo.
89. Inteligência artificial (IA) e machine learning (ML) para o bem e para o mal – conforme o IDC, até o final do próximo ano pelo menos metade dos SOCs (Security Operating Centers) adotam ferramentas de automação e inteligência artificial. Os ganhos de produtividade, com agilidade e escala, são mandatórios, inclusive, porque o outro lado também conta com recursos para automatizar testes.	é compatível com aplicações nativas ou híbridas e permite a escrita simplificada da automação de testes de caixa-preta.
90. Na abordagem Bottom-up, a estimativa do esforço de desenvolvimento começa com as atividades do nível mais baixo do WBS e termina com as do nível superior. O sistema é decomposto em unidades pequenas, de modo que os designers, programadores e DBA possam fazer estimativas mais precisas. [...] A estimativa global para o projeto resulta da somatória das estimativas das diversas atividades e fases que o compõe. Esta abordagem geralmente é	a desvantagem do bottom-up é não resolver erros simultâneos.

PERGUNTAS	RESPOSTAS
utilizada quando o software for desenvolvido com diferentes linguagens e tecnologias, complicando o uso de modelos matemáticos.	
91. No contexto de teste de especificação de POO, o que tem também sido investigado é o desenvolvimento de critérios de teste que utilizam diferentes tipos de diagramas utilizados no projeto orientado a objetos para auxiliar na geração de casos de teste.	I e III. .
92. O grafo de fluxo de um programa é obtido colocando-se todos os comandos em nós e os fluxos de controle em arestas. Comandos em sequência podem ser colocados em um único nó, e estruturas de seleção e repetição devem ser representadas.	2, 4, 3, 1, 5.
93. O sistema real, nativo ou hospedeiro (host system), contém os recursos reais de hardware e software do sistema. Já o sistema virtual, também denominado de sistema convidado (guest system) executa suas ações no sistema virtualizado. Em alguns casos, vários sistemas virtuais podem coexistir, atuando simultaneamente no mesmo sistema real.	apresentar uma quantidade expressiva de cenários de teste.
94. O teste de invasão é uma atividade recomendada para avaliar a estrutura tecnológica da informação de acordo com uma visão externa e interna. Além de avaliar a segurança da informação, esse serviço avalia a qualidade dos controles de SI implementados.	por meio de testes automatizados, que permitem realizar verificações constantes para evitar a presença de hackers dentro um código falho.
95. O teste de recuperação é um teste de sistema que força o software a falhar de diversas maneiras e verifica se a recuperação é adequadamente executada. Já o teste de segurança tenta verificar se todos os mecanismos de proteção embutidos no sistema o protegerão de fato de acessos indevidos etc.	se criarmos casos de teste que garantam aos agentes o poder de realizar os casos de uso, considerando as diferenças inerentes ao seu fluxo de eventos.
96. O teste de regressão é executado sempre que um sistema em operação sofre alguma manutenção. O problema é que a correção de um defeito no software, ou a modificação de algumas das suas funções, pode ter gerado novos defeitos.	planilhas ou ferramentas automatizadas.
97. Os requisitos funcionais do sistema podem ser capturados através de casos de usos, cenários de uso, história de uso e a features. Normalmente, o sistema é decomposto em problemas menores, dando origem a subsistemas. Depois cada subsistema é decomposto em funcionalidades, até que seja possível fazer o projeto técnico.	F, F, V, V.
98. Os termos 'teste de caixa preta' e 'teste de caixa transparente' são particularmente descritivos. Quando testamos em relação às especificações, tratamos o código como uma caixa preta totalmente opaca. Ao contrário, quando testamos em relação ao código, temos de ser capazes de enxergar dentro da caixa.	é a combinação das possíveis entradas em um sistema estabelecido, usando uma maneira inteligente para gerar um número mínimo das saídas de teste.
99. Quando criamos testes automatizados, às vezes precisamos rodar somente um novo teste recém-criado, sem a necessidade de testar todo o resto. Ou mesmo, podemos precisar rodar somente um arquivo de teste específico, sem a necessidade de rodar todos os outros".Fonte: FILHO, W. Protractor: Lições sobre testes end-to-end automatizados. [S.l]: Caso do código, 2016, p. 22.O trecho acima trata da utilidade dos testes automatizados. Considerando essas informações e o conteúdo estudado, analise os métodos utilizados na criação de casos de testes automáticos a seguir e associe-os com as suas respectivas características.1) Rapoo.2) JMobile Tools.3) TGV.4) LTS.() Auxiliar as ferramentas utilizadas nas simulações e na criação de espaços nos estados, apresentando à sua API classes e metodologias.() Ferramenta de auxílio dos testes funcionais (caixa-preta), que se baseia em metodologias de análise de modelos da implementação automática dos casos de teste.() Formalismo desenvolvido para aproveitar os modelos com características estruturais disponibilizadas pela orientação a objetos, junto com os modelos das redes de Petri.() Procedimento formal de entrada que a grande maioria das ferramentas usa na criação e escolha automática de casos de teste.Agora, assinale a alternativa que apresenta a sequência correta:	2, 3, 1, 4.

PERGUNTAS	RESPOSTAS
100. Top-down é o inverso da estratégia anterior. Uma vantagem desta estratégia é que alguns resultados podem ser apresentados para os usuários antes de a construção de componentes, programas, módulos ou subsistemas de níveis mais baixos ser completada. É claro que artifícios devem ser criados a fim de simular a ausência destes níveis.	a abordagem top-down é boa em encontrar problemas que se repetirão.
101. Um software precisa ser testado para descobrir erros que foram feitos durante o projeto e a construção. Os testes são conduzidos através de uma estratégia que integra os métodos de teste, os passos e os roteiros. A estratégia de teste deve ser planejada sob medida para cada projeto.	custo-benefício e alto nível de complexidade.
102. Leia o excerto abaixo: “A crise de software foi uma decorrência da imaturidade do mercado e dos profissionais da computação da época, pois vinha de um período onde o desenvolvimento do software não exigia requisitos e configurações complexas” Fonte: MANZANO, Augusto. A engenharia de software, a qualidade final do software e o papel do profissional de desenvolvimento. Disponível em: < https://www.revista-programar.info/artigos/a-engenharia-de-software-a-qualidade-final-do-software-e-o-papel-do-profissional-de-desenvolvimento/ >. Acesso em: 23 mar. 2019. Esse trecho refere-se à motivação principal para a crise tecnológica ocorrida entre meados dos anos 1960 e os anos 1970. A partir dessas informações e do conteúdo estudado a respeito, podemos afirmar que a Crise de Software estava diretamente ligada ao:	modelo de trabalho em equipe que era praticado, pois, além do funcionamento incorreto do sistema, havia um ambiente de incertezas quanto a quantidade de softwares e as possíveis demandas futuras.
103. A ABNT NBR ISO 9001 é a versão brasileira da norma internacional ISO 9001 que estabelece requisitos para o Sistema de Gestão da Qualidade (SGQ) de uma organização, não significando, necessariamente, conformidade de produto às suas respectivas especificações. O objetivo da ABNT NBR ISO 9001 é lhe prover confiança de que o seu fornecedor poderá fornecer, de forma consistente e repetitiva, bens e serviços de acordo com o que você especificou.” INMETRO. O que significa a ABNT NBR ISO 9001 para quem compra? Disponível em: < http://www.inmetro.gov.br/qualidade/pdf/cb25docorient.pdf >. Acesso em: 13 abr. 2019. p. 1. Assim, considerando as informações apresentadas e os conteúdos estudados, é possível afirmar que uma das vantagens do uso da norma ISO 9001 é:	atender aos atributos legais e regulamentares, além da possibilidade de se expandir mercados.
104. O MPS.BR (Melhoria de Processo de Software Brasileiro) é um modelo de maturidade desenvolvido por consórcio de empresas, com foco principal de atender aos micros, pequenas e médias empresas de software brasileiras, que necessitam melhorar seu processo de desenvolvimento de software.” PERUCCI, C. C.; CAMPOS, F. C. de. Técnicas de qualidade aplicadas em Software: um estudo bibliométrico. Revista de Ciência e Tecnologia, v. 19, n. 38, p. 5-15. Disponível em: < https://www.metodista.br/revistas/revistas-unimep/index.php/cienciatecnologia/article/view/3091 >. Acesso em 13 abr. 2019. p. 7. O trecho acima traz a definição do MPS.BR. A partir dessas informações e do conteúdo estudado, é possível visualizar que o modelo de Referência (MR-MPS) possui três guias. São elas:	Guia Geral, Guia de Aquisição, Guia de Implementação.
105. Provê informações sobre conceitos de avaliação de processo e seu uso em dois contextos, o de melhoria de processo e o de determinação de capacidade de processo. Ela descreve como as partes deste conjunto de normas se integram e provê orientações para seleção e uso”. ABNT. Código 15504. Disponível em: < https://www.abntcatalogo.com.br/curs.aspx?ID=63 >. Acesso em: 12 abr. 2019. O trecho refere-se a uma das características do histórico da norma ISO/IEC 15504. Assim, considerando as informações apresentadas e os conteúdos estudados, é possível definir a ISO/IEC 15504 como:	um modelo de referência criado a partir de um framework para analisar processos da área de Engenharia de Software.

PERGUNTAS	RESPOSTAS
106. Qualidade não é obtida espontaneamente. Ela tem de ser construída. Assim, a qualidade do produto depende fortemente da qualidade de seu processo de desenvolvimento, das pessoas envolvidas – stakeholders, desenvolvedores, gestores – dos requisitos e de muitos outros fatores. ”DEV MEDIA. Em busca da Qualidade de Software. Disponível em: < https://www.devmedia.com.br/em-busca-da-qualidade-de-software/28717 >. Acesso em 12 abr. 2019. Com base nesse texto e nas informações do conteúdo, é possível afirmar que os fatores de qualidade do software sob a ótica da Operação do Produto caracterizam-se pelas seguintes descrições:	Corretitude (Correção), Confiabilidade, Usabilidade, Integridade e Eficiência.
107. A realização de inspeções em sistemas incompletos não é custosa. Entretanto, se um sistema incompleto é testado, então testes que se adequem a essa situação devem ser criados, o que acrescenta um custo a mais no processo.	não substituem os testes de software tradicionais.
108. Geralmente, os modelos de contexto mostram que o ambiente inclui vários outros sistemas automatizados. No entanto, eles não mostram os tipos de relacionamentos entre os sistemas no ambiente e o sistema que está sendo especificado.	uma atividade que envolve a presença de Stakeholders do sistema, com o objetivo de estabelecer qual funcionalidade este deve incluir.
109. O negócio, que dava mais trabalho do que resultado, era visto como pouco estratégico para a Totvs, que tem voltado seus esforços para o desenvolvimento de software. A venda foi finalizada em outubro, por 25 milhões de reais. A parte de software da Bematech continua com a Totvs.	2, 3, 1, 4, 5.
110. Para planejar o teste de software de forma adequada, devem ser coletadas as métricas durante o teste e fazer uso de modelos existentes de confiabilidade de software, de modo que possam ser estabelecidas diretrizes significativas para que a equipe saiba quando parar de testar.	I e III.
111. Também são executados os testes de recuperação, onde o testador força um conjunto de falhas no software para verificar se a recuperação ocorre de forma adequada. Os testes de estresse são projetados para submeter o programa às situações anormais de uso como o volume e frequência anormal do cargo.	testar o comportamento de falha do sistema e apresentar defeitos que geralmente não são revelados.
112. Um conjunto de atividades, métodos, ferramentas e práticas que são utilizadas para construir um produto de software. Esta prática leva a produção de uma série de documentos que ao final são traduzidos em um programa executável que seja satisfatório. ”Revista Programar. Processo de desenvolvimento de software, o ciclo de vida clássico sobre a ótica da norma ISO/IEC/IEEE 12207:2017. Disponível em: < https://www.revista-programar.info/artigos/processo-de-desenvolvimento-de-software-o-ciclo-de-vida-classico-sobre-a-otica-da-norma-iso-iec-ieee-122072017/ >. Acesso em: 24 abr. 2019. O trecho acima trata da definição do desenvolvimento de software como um processo. A partir dessas informações e do conteúdo estudado, podemos considerar uma vantagem ao se definir a ação de desenvolvimento de software como um processo, quando:	houver redução no tempo de treinamento, já que a definição e documentação dos procedimentos e funções ajudam na inserção de um novo membro à equipe de trabalho.
113. Um software precisa ser testado para descobrir erros que foram feitos durante o projeto e construção. Os testes são conduzidos através de uma estratégia, que integra os métodos de teste, o passo e o roteiro. A estratégia de testes deve ser planejada sob medida para cada projeto, considerando o tempo que será investido neste trabalho.	a garantia de qualidade e simbolizar a análise recente da especificação, projeção e codificação.
114. baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido. Atividades de especificação, desenvolvimento e validação são intercaladas, e não separadas, com rápido feedback entre todas as atividades.	Desenvolvimento Incremental.

PERGUNTAS	RESPOSTAS
115. Conforme já descrito anteriormente, a engenharia de software compreende um conjunto de camadas que envolvem métodos, ferramentas e processos. Qualquer desenvolvimento de produto inicia com uma ideia e termina com o produto pretendido.	reunir e organizar o desenvolvimento do software.
116. É um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. Diferente de computadores de propósito geral, como o computador pessoal, realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos.	Controles Embutidos.
117. uma das atividades importantes no processo de desenvolvimento de software embutido de tempo real é a análise de timing. Em uma análise desse tipo, calcula-se com que frequência cada processo do sistema deve ser executado .	produtos, papéis, pré e pós-condições.
118. A depuração com asserções permite o mapeamento de erros para sintomas internos, pois elas indicam pontos do programa nos quais ocorrem discrepâncias em relação à especificação . O custo em termos de espaço e tempo de execução dos programas anotados com asserções é insignificante, o que viabiliza sua utilização em sistemas reais.	o assertEquals verifica, por exemplo, se uma função de soma está realmente adicionando dois valores.
119. A expressão caixa-preta é utilizada para a designação de algo que não compreendemos, portanto, algo obscuro para nós mortais. Ela pode ser utilizada para nomear tudo o que não compreendemos, em que podemos ver uma ação e uma reação, mas sem sabermos exatamente por que ocorreu.	o teste de integração verifica se a conexão entre telas é correta.
120. A implantação do sistema é a última fase do desenvolvimento de um software quando se utiliza metodologia para desenvolvimento de sistemas ou software. Essa fase sucede os projetos de teste, treinamento e documentação. Não se deve envolver o cliente e alta administração neste momento. Eles devem estar envolvidos desde a fase inicial do estudo preliminar.	projeto de dados, arquitetura e procedimental.
121. À medida que a equipe de software desce em direção ao lado esquerdo do V, os requisitos básicos do problema são refinados em representações cada vez mais detalhadas e técnicas do problema e de sua solução. Uma vez gerado o código, a equipe passa para o lado direito V, basicamente realizando uma série de testes	um modelo capaz de estabelecer uma relação entre garantia de qualidade e modelagem.
122. A métrica McCabe pode ser calculada quase tão facilmente quanto o número de linhas de código. Em alguns casos, ela se mostrou uma boa métrica para prever o número de falhas; quanto maior o número de M, maior a chance de um artefato de código conter uma falha”. SCHACH, Stephen R. Engenharia de Software: Os Paradigmas Clássico e Orientado a Objetos. 7. ed. McGrawHill, 2010. p. 493. O trecho aponta as características da Métrica McCabe e suas funcionalidades. A partir dessas informações e do conteúdo estudado, pode-se concluir que a Métrica McCabe se baseia:	em uma representação do fluxo do controle de um programa.
123. A qualidade de software depende da capacitação dos processos. Há pouco investimento das empresas em certificações que comprovem a qualidade e a maturidade dos seus processos na fabricação de software, impossibilitando a venda deste produto no mercado internacional.	a sequência de passos que visam a produção e manutenção de um software além da inter-relação, dentre outras coisas, com recursos humanos e materiais.
124. A Super Micro Computer, Inc. (NASDAQ: SMCI), membro nível ouro da Oracle PartnerNetwork (OPN), ampliou seus relacionamentos com a mesma, testando e credenciando inúmeros servidores com Linux e Oracle VM para x86 (Oracle VM), atendendo aos requisitos de clientes para nuvem, virtualização e aplicativos de banco de dados corporativos.	o retorno de investimentos no projeto.
125. A técnica estrutural (ou caixa-branca) estabelece os requisitos de teste com base em uma dada implementação, requerendo a execução de partes ou de componentes elementares do programa. Os caminhos lógicos do software são testados, fornecendo-se casos de teste que põem a prova tanto conjuntos específicos de condições e/ou laços bem como pares de definições e usos de variáveis.	V, F, V, F.

PERGUNTAS	RESPOSTAS
126.A Teoria da ciência de software de Halstead é provavelmente a mais conhecida e a mais completamente estudada... medidas compostas de complexidade (de software). A ciência de software propõe as primeiras leis analíticas para o software de computador.”PRESSMAN, Roger S. Engenharia de Software/Roger S. Pressman: São Paulo: Pearson Education do Brasil, 1995, p.757.O trecho acima estabelece o conceito de ciência definido por Halstead. A partir dessas informações e do conteúdo estudado, é correto afirmar que as medidas primitivas empregadas por Halstead objetivam:	desenvolver expressões que atendam ao comprimento global do programa.
127.A transformação digital revolucionou a maneira como as organizações geram receita, e o surgimento de conceitos como o Always in Beta reforçam ainda mais a necessidade do teste de desenvolvimento de software, que tem se apresentado como um dos aspectos mais importantes do crescimento organizacional.	por incluir todas as atividades de testes executadas pela equipe responsável pelo desenvolvimento do sistema.
128.Alguns casos de uso podem estar associados a vários atores, mas geralmente somente um deles é que indica o caso de uso, enquanto os outros fornecem ou extraem informações. O nome dado ao caso de uso deve destacar o valor que ele vai gerar para o usuário.	especificações presentes nas entradas do teste e na saída prevista pelo sistema.
129.Apesar das novas e eficientes técnicas de engenharia de software, os projetos de desenvolvimento de software continuam muitas vezes não atendendo às expectativas dos clientes por extrapolarem prazos e orçamentos, além de não satisfazerem completamente o usuário. Muitos problemas ocorrem por falhas no processo desses, consequência, sobretudo, das definições de requisitos de software incompletas e inconsistentes.	o projeto representará a primeira atividade técnica exigida para o desenvolvimento e análise de um software.
130.Aplicados a cada estágio de teste: Teste de caixa-preta: Black box: Visam verificar a funcionalidade e a aderência aos requisitos, em uma ótica externa ou do usuário, sem se basear em qualquer conhecimento do código e da lógica interna do componente testado.	no teste dos caminhos básicos o testador monta um nó ou grafo em cada parte do código.
131.As atividades específicas de garantia de segurança devem ser incluídas em todos os estágios do processo de desenvolvimento de software. Essas atividades de garantia de segurança registram as análises efetuadas e a pessoa ou pessoas responsáveis por essas análises.	2, 3, 1, 4.
132.As disciplinas do AUP devem ser executadas de forma iterativa, de modo a definir quais atividades os membros da equipe de desenvolvimento devem realizar para construir, validar e entregar um sistema que atenda às necessidades do negócio identificadas ao longo das fases e dos marcos AUP.	visualizar os erros no momento da validação do sistema.
133.As manutenções solicitadas pelos usuários são fontes de novos defeitos, inclusive gerando problemas em partes do programa que não foram modificados. Para identificar estas situações, sempre devem ser aplicados os testes de regressão completos, evitando testar apenas modificações.	o nível unidade realiza testes de códigos unitários.
134.As principais falhas verificadas em projetos de software são relativas aos requisitos, devidas às dificuldades no entendimento das necessidades do usuário. Portanto, realizar corretamente o levantamento e administração de requisitos é essencial para a qualidade de software.	estabelece a arquitetura e os componentes de um sistema.
135.Componentes de software fornecem função em seus padrões de sistema virtual. É possível configurar componentes e definir interações entre eles para criar um ambiente do WSRR totalmente operacional quando se implementa seu padrão na nuvem.”IBM. Componente do software. Disponível em: < https://www.ibm.com/support/knowledgecenter/pt-br/SSWLGF_8.5.0/com.ibm.sr.wsrrhypervisor.doc/topics/cwhe_sware_components.html >. Acesso em 13 abr. 2019.Considerando as informações apresentadas e os conteúdos estudados, é possível afirmar que o processo de medição do produto pode ser dividido, sequencialmente, em:	Escolher medições a serem efetuadas, selecionar componentes a serem avaliados, medir características de componentes, identificar medições anômalas e analisar componentes anômalos.

PERGUNTAS	RESPOSTAS
136. Da mesma forma que um arquiteto ao construir uma casa, o projetista de software, utilizando a UML, vai construir o sistema de forma incremental e iterativa, especificando separadamente cada uma das suas partes modeladas isoladamente”.	a descrição computacional é apresentada e o usuário fica sabendo o que o software deve realizar.
137. De fato, em meados de 1965 o termo crise de software não havia sido usado, isto ocorreu durante a década de 1970 quando as dificuldades relacionadas ao desenvolvimento do software começaram a ser mais graves “.	o ano de 1975, quando o crescimento dos sistemas baseados em computadores se deu ao mesmo tempo em que a manutenção foi se tornando impossível.
138. É possível criar modelos estruturais de um sistema quando se está discutindo e projetando sua arquitetura. O projeto de arquitetura é um tema particularmente importante na Engenharia de Software”.	à inclusão de componentes e diagramas de implantação da UML.
139. É um artefato criado quando uma descrição detalhada de todos os aspectos do software ser construído deve ser especificado antes do projeto começar. É importante lembra que uma SRS formal nem sempre é por escrito. Na verdade, há ocasiões em que os esforços gastos em uma SRS talvez fosse mais bem aproveitado em outras atividades da engenharia de software”.	Especificação de Software.
140. É uma disciplina científica relacionada com a compreensão das interações entre humanos e outros elementos de um sistema. Tem como conhecimentos relacionados: computação gráfica, processo de desenvolvimento, técnicas de aplicação, entre outros.	incluir novas áreas de conhecimento em detrimento de outras.
141. Entender a base de código e avaliar a qualidade do software é essencial para simplificar a complexidade e estabelecer uma linha de base para os esforços de modernização, manutenção e transformação.”VERON, Frederic. 10 coisas que os CIOs precisam saber sobre seu software.	o nível organizacional, em que o gerenciamento de qualidade se dedica em definir um framework de processos organizacionais e de padrões.
142. Existem diferentes metodologias para abordar os projetos de implementação de software, o que inclui a construção e implantação. A metodologia descreve as atividades envolvidas na definição, construção e implementação de um sistema: a metodologia é um framework.	2, 4, 3, 1.
143. JUnit é um framework de teste que vem sendo muito utilizado e viabiliza a documentação e a execução automática de casos de teste. O framework JUnit é de código aberto e pode ser utilizado para escrever e executar de forma automática um conjunto de gteste, fornecendo relatórios sobre quais casos de teste não se comportaram de acordo com o que foi especificado.	F, F, V, V, F.
144. Mitos que ainda sobrevivem em um determinado grupo, que atua na área, tem resistido por mais de 60 anos de cultura de programação. Durante os seus primórdios, a programação era vista como uma forma de arte. Hábitos e atitudes antigos são difíceis de perder”.	Mito do Profissional.
145. Modelos do novo sistema são usados durante a engenharia de requisitos para ajudar a explicar os requisitos propostos para outros stakeholders do sistema. Os engenheiros usam esses modelos para discutir propostas de projeto e documentar o sistema para a implementação.	um processo de desenvolvimento baseado no modelo de abstração.
146. Muitos de nós sabemos como alguns nomes ou procedimentos ocorrem. Antigamente, alguns eram mais usados do que hoje. Vamos falar do teste de caixa-branca. Quando falamos do teste caixa-branca, iremos analisar o que ocorre em cada momento de uma aplicação, tendo importância o seu código e como foi escrito e desenvolvido seu raciocínio.	a anotação @before lê os dados do programa e prepara o ambiente os futuros testes.
147. Na maioria dos projetos de software, há algum reuso de software. Isso acontece muitas vezes informalmente, quando as pessoas envolvidas no projeto sabem de projetos ou códigos semelhantes ao que é exigido. Elas os buscam, fazem as modificações necessárias e incorporam-nos a seus sistemas.	Projeto do sistema com reuso.

PERGUNTAS	RESPOSTAS
148.No final dos anos de 1960, o desenvolvimento de software era considerado uma arte e não existia um processo formalmente estabelecido. Nesse modelo, os programadores simplesmente codificavam com base em conversas com usuários e depois se reuniam para testar e consertar os erros em conjunto.	o código rígido às modificações.
149.Nos anos 60, a expressão ‘crise de software’ começou a ser utilizada para denominar o conjunto de problemas encontrados no processo de desenvolvimento de software. Entretanto, mesmo com o surgimento da engenharia de software, muitos desses problemas persistiram e são recorrentes até os dias atuais”.	processo de maturação na formação profissional se encontrar em andamento.
150.O conceito de teste de software pode ser visto como roteiro e como controles dos produtos gerados no software. Preferencialmente deve contemplar a verificação de cada fase e a validação do software como um todo, constante na NBR ISO 9000-3	um dos problemas avaliados pela caixa-preta é ver um campo de nascimento gerando idade futura.
151.O grande objetivo da área de controle de qualidade em projetos de desenvolvimento de software é reduzir ao máximo possíveis defeitos nas aplicações criadas. Dentro dessa perspectiva, precisamos pensar mais sobre os indicadores-chave de desempenho (Key Performance Indicator – KPIs) nesse processo.”CRONAPP. Conheça os principais KPIs para controle de qualidade e produtividade no desenvolvimento de software.	pode ser estabelecido e comparado com projetos que se encontram em desenvolvimento presente.
152.O hardware é a parte física do computador, ou seja, todas as peças e componentes usadas pra fazer a máquina funcionar . Os softwares de sistema servem para controlar o sistema interno do computador e para fazer com que tudo funcione lá dentro.	é um produto de características lógicas e não físicas.
153.O modelo clássico foi provavelmente a primeira metodologia publicada de desenvolvimento de software. Este modelo estabelece uma sequencia de etapas. Cada etapa tem associada ao seu término uma documentação que deve ser aprovada para que a etapa posterior possa ter inicio.	Análise e definição de requisitos; Projeto de sistema e software; Implementação e teste unitário; Integração e teste de sistema; e Operação e manutenção.
154.O Objetivo de teste de caixa preta é avaliar a conformidade de um módulo, sem entrar em detalhes quanto à sua implementação. Embora sejam mais simples de realizar, requerem uma massa de dados muito bem estruturadas. Normalmente, os roteiros de testes são preparados antes da construção do sistema.	é feito dentro das interfaces de um software, localiza possíveis erros e é empregado para comprovar que as funções atribuídas são operacionais.
155.O OpenUP é compatível com o RUP em sua estrutura estática pois ambos possuem os elementos: papéis, atividades, artefatos, fluxos e disciplinas. O OpenUP implementa os papéis necessários para times pequenos: Gerente de Projetos, Analista, Arquiteto, Desenvolvedor, Testador e Stakeholders”.	o planejamento do processo e a redução dos riscos por meio do foco na arquitetura.
156.O problema decorre de uma falha na realização de uma verdadeira garantia de qualidade de software, diz Mark S. Kadrich, CISO provisório para o hospital comunitário Martin Luther King Jr., em Los Angeles.‘Estive no setor tempo suficiente para saber que, se eu estou perdendo o sono por falha na tecnologia, estou na indústria errada’, diz ele. ‘Oitenta por cento do software é uma porcaria, enquanto 20 por cento é apenas uma besteira. Há muito pouco que pode ser considerado bem projetado”’.TYNAN, Dan. Seis verdades difíceis de aceitar e aprender.	garantia da qualidade do software (SQA)
157.O processo UP, embora não seja exclusivo, foi desenvolvido no cenário da abordagem Orientada a Objetos, com a proposta da Unified Modeling Language (UML). A UML é uma linguagem visual resultante de numerosos métodos orientados a objeto que existam no início da década de 1990.	2, 4, 3, 1.
158.O PSP (Personal Software Process) tem o objetivo geral de produzir softwares sem defeitos, respeitando prazos e custos planejados. Seu foco principal é a melhoria de processos do indivíduo, tornando sua forma de trabalho mais disciplinada, para o PSP, o indivíduo é responsável pela qualidade de seu trabalho.”	inserir pessoas e equipes de desenvolvimento de software.

PERGUNTAS	RESPOSTAS
159.O RUP é um modelo constituído de fases que identifica quatro fases distintas no processo de software. No entanto, ao contrário do modelo em cascata, no qual as fases são equalizadas com as atividades do processo, as fases do RUP são estreitamente relacionadas ao negócio, e não a assuntos técnicos.	construção.
160.O SWEBOK é recomendado para diversos tipos de público, em todo o mundo, com o objetivo de ajudar organizações a terem uma visão consistente da Engenharia de Software. É endereçado a gerentes, engenheiros de software, às sociedades profissionais, estudantes, professores e instrutores desta área de conhecimento.	auxiliar as empresas a observarem de maneira mais ampla a Engenharia de Software como um todo.
161.O teste de cenário pode simular condições específicas, eventos e cenários de risco. Os testes devem ser realizados em situações o mais próximo possível de realidade para efetivamente garantir que, em caso de crise ou eventos de falha, o Plano de continuidade de Serviço de TI possa atender satisfatoriamente aos seus propósitos.	uma abordagem na qual o usuário vai criar cenários que auxiliam no desenvolvimento de casos de teste do sistema.
162.O teste de sistemas de inteligência artificial é um interesse emergente no Brasil. Os fornecedores estão trabalhando em maneiras de verificar se um sistema com IA está respondendo conforme o esperado. No entanto, é necessária experiência humana para criar os métodos de teste.	teste unitário, teste de componentes e teste de sistema.
163.O teste de software faz parte de um aspecto mais amplo de garantia de qualidade no desenvolvimento do Software. Ele deverá ser executado em qualquer projeto de desenvolvimento independente da metodologia de gerenciamento de projetos utilizada.	F, F, V, V.
164.O teste de stress se trata de colocar tantos usuários virtuais até que o nível de serviços (SLM) junto dos nossos usuários seja ‘quebrado’ ou a aplicação ‘pare’. Neste ponto sabemos qual é o momento em que a aplicação irá falhar pela quantidade de acessos/usuários.	I e II.
165.O teste end-to-end é um tipo de teste de Sistemas que visa colocar o sistema à prova de uma forma mais completa a partir da simulação de um ambiente real, tais como: acessar um banco de dados, usar uma rede de comunicação e interagir com outros hardwares e sistemas. Os testes dependentes do ambiente de produção (estresse, carga e performance) devem ser realizados, de preferência, no ambiente final onde o software irá operar.	o teste de stress avalia como o programa se comporta se várias pessoas acessarem ao mesmo tempo.
166.Os modelos são usados durante o processo de engenharia de requisitos para ajudar a extrair os requisitos do sistema; durante o processo de projeto, são usados para descrever o sistema para os engenheiros que o implementam; e, após isso, são usados para documentar a estrutura e a operação do sistema.	implementar os requisitos ao novo sistema.
167.Os primeiro programas a serem codificados devem ser aqueles relacionados ao gerenciamento do sistema e, em seguida, os demais. O ideal é que cada programa seja testado isoladamente e também com os demais, observando o plano de teste previamente definido no projeto do sistema”.	a implementação.
168.Os requisitos expressam as necessidades e restrições colocadas sobre o produto de software que contribuem para a solução de algum problema do mundo real. Esta área envolve elicitação, análise, especificação e validação dos requisitos de software.	Geralmente Aceitas, Pesquisa Avançada e Especializado.
169.Os testes de segurança tenta verificar se todos os mecanismos de proteção embutidos num sistema o protegerão de fato de acesso indevidos. Já o teste de desempenho é idealizado para avaliar o desempenho do software de forma isolada e dentro do contexto de um sistema integrado.	são projetados para garantir que o sistema processe as atividades destinadas a ele e se o mesmo consegue atender aos seus requisitos.

PERGUNTAS	RESPOSTAS
170. Para ajudar nessa questão, a International Organization Standardization – ISO e a International Electrotechnical Commission – IEC, que são organismos normalizadores com importância internacional reconhecida no setor de software, se uniram para editar normas internacionais conjuntas. A norma internacional ISO/IEC, define qualidade de software como a totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas”.	falta de padronização do software.
171. Pode-se ter interação do usuário, que envolve entradas e saídas, interação entre o sistema que está em desenvolvimento e outros sistemas, ou interação entre os componentes do sistema.	Modelagem de caso de uso.
172. São introduzidos testes Beta de todos os produtos e as solicitações dos clientes por modificações são incorporadas no produto entregue. Novas versões de produtos, desenvolvidas com esse processo modificado, são entregues.”SOMMERVILLE, Ian. Engenharia de Software. 9ª ed. Pearson Education, 2011. p. 472. O trecho fala sobre as mudanças no processo de projeto de software. Assim, considerando as informações apresentadas e os conteúdos estudados, é correto afirmar que essas mudanças podem:	atrair clientes novos.
173. São realizados pela equipe testes visando a execução do sistema como um todo ou um subsistema (parte do sistema), dentro de um ambiente operacional controlado, para validar a exatidão e perfeição na execução de suas funções. Neste estágio de teste, a operação normal do sistema deve ser simulada sendo testadas todas as suas funções de forma mais próxima possível do que ocorrerá no ambiente de produção.	descobrir se dois objetos são diferentes é função do assertNotSame.
174. Se a organização não fizer o relatório por nível de teste, mesmo assim deverá fazer o relatório master, que representará, neste caso, um sumário de projeto de teste. Deve haver uma correspondência entre o Plano Master de Teste e o relatório Master de Teste.	disponibiliza uma série de informações relevantes sobre as falhas observadas no sistema, além do seu comportamento.
175. Testes de desenvolvimento incluem todas as atividades de testes que são realizadas pela equipe de desenvolvimento do sistema. O testador do software geralmente é o programador que o desenvolveu, embora nem sempre seja assim”.	Teste de sistema.
176. Um processo representa, dentro da área de software, um conjunto de atividades cujo objetivo é atingir uma meta previamente estipulada. Já por capacidade e maturidade de um processo, deve-se ter a noção do grau de qualidade com o qual um processo atinge um resultado esperado.” DEVMEDIA. Maturidade no desenvolvimento de software: CMMI e MPS.	direcionar a empresa, implementando e melhorando de forma contínua o processo de software.
177. Uma das vantagens do modelo UP/RUP em relação aos projetos apresentados nas seções anteriores é a desvinculação das fases com os fluxos de trabalho que podem ser ativados em todos os estágios do processo. Outra vantagem é o reconhecimento de que a implantação de software no ambiente do usuário é parte do processo.	Perspectiva Dinâmica, Estática e Prática.
178. Utilizando o GitLab, uma equipe de engenharia de software pode colaborar com a equipe de operações para criar e entregar softwares de forma mais eficiente. As diferentes fases dos DevOps são: gerenciamento, planejamento, criação, verificação, pacotes, release, configuração, monitoração e segurança.	Exigência da aquisição de softwares.
179. Vem o teste de validação que valida se o software implementado corresponde aos requisitos especificados. Este teste fornece a garantia final de que o software satisfaz a todos os requisitos funcionais, comportamentais e de desempenho.	da abordagem sistemática direcionada aos projetos de casos de teste, em que o usuário vai considerar cada requisito e desenvolver uma série de testes.

PERGUNTAS	RESPOSTAS
180. As fases do ciclo de vida AUP são sequenciais ao longo de todo o projeto. Em muitos casos a Inception Phase se torna um pré-projeto. Após o trabalho de identificação de detalhamento inicial do escopo do projeto principal, é possível estimar o tempo, o custo e os recursos.	observar o escopo inicial do projeto, além da arquitetura sistêmica disponível.
181. Cada modelo representa cada abordagem usada para a criação do software. No modelo praxis o desenvolvimento do software é feito por forma orientada a objetos com processos unificados e fases formadas por iterações.	a etapa em que as demandas dos usuários e os conceitos da aplicação são avaliados.
182. O modelo comportamental indica como o software vai responder aos estímulos ou eventos externos. Para criá-lo, devemos executar etapas: Avaliar todos os casos de uso para entender completamente a sequência de interação dentro do sistema é uma destas etapas.	demonstra como o sistema se comporta a eventos de caráter externo e interno, simultaneamente.
183. Leia o trecho abaixo: “Gerentes com responsabilidades sobre software, assim como gerentes da maioria das áreas, frequentemente estão sob pressão para manter os orçamentos evitar deslizos nos cronogramas e elevar a qualidade. Como uma pessoa que está se afogando e se agarra à crença em um mito do software para aliviar a pressão (mesmo que temporariamente)”.	Mito de Gerenciamento.
184. A norma ISO/IEC 12207 estabelece uma arquitetura de alto nível do ciclo de vida de software que é construída a partir de um conjunto de processos e seus inter-relacionamentos”. MICREIROS.COM. Norma NBR ISO/IEC 12207. Disponível em: < http://micreiros.com/norma-nbr-isoiec-12207/ >. Acesso em: 13 abr. 2019. Esse trecho trata de uma das funções da ISO IEC 12207 no ciclo de vida do software. Assim, considerando as informações apresentadas e os conteúdos estudados, analise os processos primários realizados pela ISO/IEC 12207 e associe-os com suas respectivas características: 1. Fundamentais 2. Suporte 3. Organizacionais () Esse processo ocorre através do contrato existente entre o fornecedor e adquirente do software. () São utilizadas por uma empresa para definir e aplicar uma estrutura formada pelos processos. () Refere-se ao auxílio e à contribuição que o processo possibilita para a qualidade. Agora, assinale a alternativa que apresenta a sequência correta:	1, 3, 2.
185. A revisão de software é um conjunto de atividades sistemáticas na área da engenharia de software que visa garantir a alta qualidade em todo o ciclo do desenvolvimento”. XAVIER, Wagner. Muito além do desenvolvimento de software. Disponível em: < https://computerworld.com.br/2012/08/13/muito-alem-do-desenvolvimento-de-software/ >. Acesso em: 12 abr. 2019. O trecho aborda a definição da revisão de software. Com base nessas informações e no conteúdo estudado, um fator que analisa o esforço dispendido para encontrar e solucionar erros em um programa é classificado como:	manutenibilidade.
186. Métricas ajudam a medir e quantificar os resultados dos processos e oferecem o embasamento necessário para avaliarmos e revisarmos o trabalho em andamento. Elas têm sido uma ferramenta essencial na produção de um software. Diante dessas informações e do conteúdo estudado sobre métricas, é correto afirmar que a métrica de codificação:	analisa a quantidade de comandos utilizados em cada linha ou avalia fisicamente o tamanho do código fonte do software e mede o tamanho através da quantidade de linhas.
187. Muitos profissionais da área de sistemas possuem dificuldades na hora de definir qualidade de software devido à quantidade de aspectos que precisam ser considerados, por esse motivo, a qualidade de software é dividida, em categorias, uma delas é “Usabilidade”, que se preocupa com como o usuário conseguirá utilizar o software com facilidade de operação e sem ocorrência de falhas. Considerando essas informações e o conteúdo estudado em relação ao atributo de Atratividade, pode-se afirmar que:	o atributo de atratividade demonstra como o layout pode tornar o sistema intuitivo para o usuário, auxiliando na compreensão e na operação das funcionalidades.
188. Muitos profissionais da área de sistemas possuem dificuldades na hora de definir qualidade de software, devido à quantidade de aspectos que precisam ser considerados, para este motivo a qualidade de software é dividida em categorias, que têm como objetivo servir base na avaliação de produto de software, na qual o gestor de projetos deve utilizar de mecanismos para	essa categoria se preocupa como o usuário conseguirá utilizar o software e se a interface tem um layout atraente, fácil usar e bem intuitivo.

PERGUNTAS	RESPOSTAS
verificar seus atributos no software que será entregue. Diante dessas informações e do conteúdo estudado sobre qualidade de software, é correto afirmar em relação à categoria de usabilidade:	
189. Na década de 1990, devido à abertura do mercado mundial e com surgimento da internet, os softwares começaram a fazer parte das atividades diárias de uma empresa e, com isso, a preocupação por entregar softwares com qualidade começou a ser discutida. Considerando essas informações e o conteúdo estudado sobre os motivos que levaram as empresas a se importarem com a qualidade do software, pode-se afirmar que as empresas passaram a se importar porque:	estavam desperdiçando bilhões por ano em softwares que não apresentavam as características e funcionalidades prometidas ao cliente.
190. Nas empresas com baixo orçamento ou prazo reduzido para avaliar e desenvolver testes, constitui-se indispensável estabelecer métricas para analisar o funcionamento eficiente e a utilidade dos itens testados. Considerando essas informações e de acordo com o conteúdo estudado sobre métricas básicas e profundas, analise as afirmativas a seguir. I. São os testadores ou analistas de testes que fazem as métricas básicas. II. O gestor de testes é responsável pela realização das métricas profundas. III. Métricas básicas identificam botões de alerta nos aplicativos. IV. “Métricas profundas” é o teste que cria a caixa-preta. Está correto apenas o que se afirma em:	I e II.
191. No CMM, as organizações de software são enquadradas em um dos cinco níveis de maturidade definidos pelo modelo, os quais possuem influência nos princípios de qualidade de produto, cujo objetivo é definir qual o nível de maturidade da instituição e traçar os planos de ação para o desenvolvimento da organização. Considerando essas informações e o conteúdo estudado sobre as características do modelo CMM, pode-se afirmar que entre as limitações da aplicação do modelo estão:	as dificuldades de aplicação em pequenas organizações.
192. No controle de qualidade é importante salientar que os defeitos que não são removidos precocemente de um sistema acabam sendo detectados depois da entrega do software. Quanto mais tarde um defeito é corrigido, mais cara é a sua manutenção. Em relação ao controle de qualidade de um software assinale a alternativa correta:	o controle de qualidade busca identificar esses defeitos através de testes, inspeções e checagem dos artefatos produzidos.
193. No processo de construção de um software, a maturidade dos processos de uma organização visa a qualidade do produto gerado e a consequente satisfação dos seus clientes, através dos modelos de referência. Em relação aos níveis de capacidade e maturidade, sabe-se que, quando o processo suporta e atinge os objetivos específicos da área de processos mediante transformação das entradas identificadas em saídas, ele se encaixa em um determinado nível. Considerando essas informações e os conteúdos estudados sobre o CMMI, pode-se afirmar que o texto está se referindo ao nível de capacidade e maturidade do CMMI:	1.
194. No processo de uma revisão, um grupo de pessoas examina o software e a documentação associada em busca de prováveis problemas e não conformidade com padrões. Revisões e inspeções são atividades que têm o intuito de controlar o nível de qualidade dos entregáveis de projeto. Considerando essas informações e o conteúdo estudado sobre processo de revisão de software, analise as afirmativas a seguir: I. Existem muitas variações na modelagem do processo de revisão, mas que é normalmente composto de três fases: atividades pré-revisão, reunião de revisão, atividades pós-revisão. II. As atividades preparatórias são essenciais para a eficácia da revisão. Em geral, as atividades de pré-revisão estão relacionadas com o planejamento e a preparação da revisão. O planejamento de revisão envolve a definição de uma equipe de revisão, a organização de um tempo e de um lugar para sua ocorrência e a distribuição de documentos a serem revistos. III.	I, II e IV.

PERGUNTAS	RESPOSTAS
195. Normas servem para ajudar um setor. A norma referente aos oito estágios de teste foi criada para melhorar a ordem dos métodos de teste. O objetivo da IEEE é criar conhecimentos, premiações, inovações e normas nas áreas de tecnologia e eletrônica. Considerando essas informações e o conteúdo estudado em métricas relacionadas ao software, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s). I. () O estágio Especificação dos Procedimentos cria uma tabela com uma linha chamada “status”. II. () O estágio Especificação dos Procedimentos cria uma tabela com uma linha chamada “entradas”. III. () O estágio Especificação dos Procedimentos cria uma tabela com uma linha chamada “pré-condições”. IV. () O estágio Especificação dos Procedimentos cria uma tabela com uma linha chamada “data e nome” de quem corrigiu. Agora, assinale a alternativa que apresenta a sequência correta:	F, V, V, F.
196. Normatização é uma atividade regulatória que estabelece o processo de formulação e aplicação de regras para solucionar ou prevenir eventuais problemas, com o objetivo de tornar mais seguro, limpo e eficiente o processo de desenvolvimento e fabricação. Considerando essas informações e os conteúdos estudados sobre os grandes beneficiados da estrutura de normas, analise as afirmativas a seguir: I. As empresas são o único setor que tem liberação para criar suas próprias normas, não sendo necessário que sigam um órgão de normatização para competir em mercados por todo o globo. II. Ao garantir que produtos e serviços estejam de acordo com a conformidade com o solicitado, é garantida a segurança, qualidade e confiabilidade dos mesmos. III.:	II, III e IV.
197. O aplicativo Apache JMeter é um software de código aberto que foi criado para medir desempenhos de programas. Ele simula programas com muitos usuários e cargas pesadas e mede a performance de tais programas com esses níveis de carga. De acordo com o texto e com o conteúdo estudado sobre teste de Stress ou de Carga, pode-se afirmar que programas como o JMeter são importantes, pois:	exibem a simulação da capacidade de carga que o sistema suporta.
198. O CMM é um modelo para medição da maturidade de uma organização no que diz respeito ao processo de desenvolvimento e administração de software, cujo objetivo é definir qual o nível de maturidade da instituição e traçar os planos de ação para o desenvolvimento da organização. Considerando essas informações e o conteúdo estudado sobre as características do modelo CMM, pode-se afirmar que o objetivo:	determina a capacitação da organização e apoia a sua evolução de acordo com os níveis estabelecidos.
199. O CMM tem como objetivo promover o aprimoramento dos processos de software utilizados pelas organizações de desenvolvimento e manutenção de sistemas, evitando a desorganização. Ele veio para minimizar os erros das empresas com relação ao desenvolvimento, planejamento e aperfeiçoamento dos programas computacionais. Considerando essas informações e o conteúdo estudado sobre as características do modelo CMM, a respeito das definições dos processos, pode-se afirmar que:	foram estabelecidas 18 áreas de processos organizados em níveis de maturidade.
200. O CMMI (Capability Maturity Model Integration) é um modelo de referência que contém práticas Genéricas ou Específicas necessárias à maturidade em disciplinas. O CMMI é uma evolução do CMM e procura estabelecer um modelo único para o processo de melhoria corporativo, integrando diferentes modelos e disciplinas. A versão 1.2 do CMMI sugere que suas 22 áreas de processo sejam agrupadas em quatro categorias de afinidade. Considerando essas informações e os conteúdos estudados sobre CMMI, pode-se afirmar que quatro as categorias são:	gestão do processo, gestão do projeto, engenharia e suporte.

PERGUNTAS	RESPOSTAS
201.O CMMI (Capability Maturity Model Integration) foi criado pelo SEI (Software Engineering Institute), que é um órgão integrante da universidade norte-americana Carnegie Mellon. Trata-se de um modelo com um enfoque voltado para a capacidade de maturidade de processos de software. No modelo de referência CMML, existem dois níveis associados ao caminho de melhoria de processos: níveis de capacidade e níveis de maturidade. O foco em gerenciar e melhorar o desempenho organizacional é uma distinção crítica entre determinados níveis de maturidade.Considerando essas informações e o conteúdo estudado sobre o CMMI, pode-se afirmar que os níveis descritos no texto-base são denominados:	quantitativamente gerenciado e em otimização.
202.O CMMI é um modelo desenvolvido para aplicação específica em software dentro de um contexto de qualidade total no âmbito de uma organização e se baseia principalmente na experiência das empresas de software e em conceitos de gerenciamento de processos e em aplicação de processos de qualidade em outras áreas.Considerando essas informações e o conteúdo estudado do modelo CMMI, pode-se afirmar que:	a metodologia CMMI visa auxiliar a organização a estabelecerem prioridades para melhoria.
203.O CMMI procura nortear a organização no sentido de implementar a melhoriacontínua do processo de software e o faz através de um modelo que contempla duas representações, divididas em níveis, priorizando de forma lógica as ações a serem realizadas. Quanto maior o nível, maior a maturidade da organização, o que pode se traduzir em maior qualidade do produto final, com maior previsibilidade em cronogramas e orçamentos. O CMMI 1.2 tem como foco a melhoria de processos de desenvolvimento de produtos e serviços.Considerando essas informações sobre a melhoria de processos de desenvolvimento de produtos e serviços do CMMI 1.2, pode-se afirmar que esse modelo tem como característica:	o agrupamento de seus componentes estruturais em três categorias: requeridos, esperados e informativos.
204.O CMMI visa fornecer às organizações um conjunto de melhores práticas em desenvolvimento e manutenção de produtos e serviços tecnológicos. Ele pode ser usado como um guia na priorização de ações e esforços de melhoria em seus processos. Quando uma organização atinge um nível de maturidade, considera-se que seus processos alcançaram uma determinada capacidade. Considerando essas informações e o conteúdo estudado sobre os aspectos do CMMI, analise as afirmativas a seguir.I. Um dos objetivos do CMMI é guiar a melhoria de processos na organização.II. No CMMI, a representação contínua começa com práticas básicas de gerenciamento e progride por um caminho predefinido de níveis de sucesso, cada um servindo de base para o próximo.III. No CMMI existem quatro disciplinas de conhecimento.IV. No CMMI as disciplinas são: engenharia de sistemas, engenharia de software, desenvolvimento e integração de produtos e processos e fontes de aquisição.Está correto apenas o que se afirma em:	I, III e IV.
205.O conceito da qualidade foi iniciado e introduzido no Japão e, a partir daí, esses conceitos de qualidade vêm agregando valor para inúmeras empresas em todo o mundo. Inicialmente, o termo qualidade estava relacionado apenas aos conceitos técnicos da produção. Posteriormente, o conceito evoluiu para a visão de satisfazer o cliente. Considerando essas informações e o conteúdo estudado sobre os aspectos gerais do conceito de qualidade, analise as afirmativas a seguir.I. Pode-se considerar que o conceito de qualidade é a adequação dentro das normas para atender às exigências do cliente interessado num produto.II. O conceito de qualidade é universal e absoluto e, em relação a esse conceito, a qualidade pode ser caracterizada como algo definitivo.III. O conceito de qualidade significa perfeição, no geral as empresas estabelecem medidas de qualidade, para assim verificar se essas medidas foram alcançadas.IV. O conceito de qualidade é um conceito subjetivo que está relacionado com as percepções, necessidades e resultados em cada área de produção de um produto.Está correto apenas o que se afirma em:	I e IV.

PERGUNTAS	RESPOSTAS
206.O controle de qualidade engloba um conjunto de ações da engenharia de software que ajudam a garantir que cada produto resultante de um processo atinja suas metas de qualidade. PRESSMAN (2011) afirma que, para fazer o controle da qualidade, ele deve envolver algumas atividades.	uma atividade é realizar teste de software para verificar se os dados de entradas correspondem aos resultados esperados pelos usuários finais.
207.O desempenho de um software está ligado diretamente com tempo de processamento, isto é, o prazo consumido para executar funções simples ou complexas, tal como enviar e-mails, arquivos, fazer postagens nas redes sociais ou transferências. Tempo é qualidade quando se refere a aplicativos, e os usuários preferem programas ágeis e descomplicados.Considerando o texto e o conteúdo estudado sobre teste de desempenho, pode-se dizer que:	o teste de desempenho e o teste de stress podem ser realizados conjuntamente.
208.O IEEE é o instituto responsável por criar padrões de software, ele incluiu o Processo de garantia da qualidade, a Software Quality Assurance (SQA - Garantia de Qualidade de Software) que verifica se os processos estão sendo seguidos e feitos de acordo com o que foi especificado. Para que o SQA aconteça, pode existir uma pessoa responsável, um analista de qualidade ou um auditor de qualidade que vai checar se esses processos estão sendo realizados.Considerando essas informações e o conteúdo estudado sobre garantia de software, pode-se afirmar que:	o foco da garantia da qualidade do software é no processo, ela verifica se os processos estão sendo executados e melhorados.
209.O modelo CMM teve tal repercussão que diversos outros passaram a se basear diretamente nele. Em um dos seus níveis, são realizadas as seguintes atividades: análise causal e resolução, inovação organizacional e implantação.Considerando essas informações e o conteúdo estudado sobre o modelo CMM, pode-se afirmar que as características mencionadas são referentes ao nível:	otimização.
210.O modelo CMMI, na sua terceira versão, está voltado à maturidade das etapas e processos do desenvolvimento de um software. Por processo, devemos entender um conjunto ou bloco de atividades que buscam atingir um objetivo pré-estipulado pelos gestores. O modelo CMMI oferece duas possibilidades de representação: contínua e por estágios. Considerando essas informações sobre elas, pode-se afirmar que:	para atingir um determinado nível, a organização deve satisfazer todos os objetivos da área de processo ou conjunto de áreas de processo que se quer melhorar, não importando se é um nível de capacidade ou de maturidade.
211.O objetivo do CMMI é servir de guia para a melhoria de processos na organização, assim como para auxiliar a habilidade dos profissionais em gerenciar o desenvolvimento de aquisição e manutenção de produtos ou serviços de software, além de proporcionar a visibilidade apropriada do processo de desenvolvimento para todos os envolvidos no projeto. O CMMI 1.3 apresenta três modelos. Considerando essas informações e o conteúdo estudado sobre os três modelos do CMMI, pode-se afirmar que eles são:	desenvolvimento, aquisição e serviços.
212.O objetivo do teste de usabilidade é definir um sistema útil, que supra as necessidades do usuário, que seja prático e de fácil acesso, com funções com aspectos estéticos nítidos, organizados e agradáveis. Dos elementos que o software precisa apresentar, são importantes os atalhos e as ações de retorno ou desistência. Considerando o texto e o conteúdo estudado sobre teste de usabilidade, pode-se dizer que:	muitas empresas colocam o usuário para testar um programa e avaliam através de câmeras e de observação como os usuários se comportam ao usar um programa novo.
213.O PMBOK (Project Management Body of Knowledge) é um guia utilizado no mundo todo chamado para a gestão de projetos, ele foi criado pelo PMI (Project Management Institute), que agrupou os praticantes da área de gestão de projetos que resultou nesse guia que sistematiza os conceitos e as ferramentas da gestão de projetos. Considerando essas informações e o conteúdo estudado sobre gestão de qualidade no PMBOK, pode-se afirmar que:	a gestão de qualidade é dividida em três fases: planejamento, garantia e controle.

PERGUNTAS	RESPOSTAS
214.O Processo Unificado foi proposto por Booch, Jacobson e Rumbaugh em 1999. Essas três pessoas juntaram o conhecimento que adquiriram durante 30 anos no desenvolvimento de softwares e criaram o Processo Unificado. O seu desenvolvimento foi apoiado nas áreas de conhecimento e fases do PMBOK (Project Management Body of Knowledge). Fonte: GRADY, B. UML: guia do usuário. 1 e 2 ed. Rio de Janeiro: Elsevier, 2005.Considerando essas informações e o conteúdo estudado sobre a qualidade de software no Processo Unificado, pode-se afirmar que:	a qualidade é tratada dentro da disciplina gerenciamento de projetos e configuração e mudanças.
215.O programa Selenium é simples, rápido, prático, fácil de aprender, leve e útil. Ele possui um método de instação bem simplificado, porém sofre atualizações anualmente. Caso você queira baixar uma versão antiga, terá que mudar a forma de instalação, importante o plugin no navegador.Considerando essas informações e o conteúdo estudado sobre o Selenium IDE, é correto afirmar que ele:	é instalado por extensão.
216.O termo “métricas” refere-se a estatísticas gerais e medidas de desempenho. Ele pode ser entendido como um meio de mensuração de desempenho por meio de uma coleção de dados específicos ou gerais. A utilização de métricas tem sido uma ferramenta essencial na produção de um sistema.Diante dessas informações e do conteúdo estudado sobre métricas, é correto afirmar que a Métrica de Projeto UML:	verifica o número de diagramas gerados tanto comportamentais e estruturais que delineiam o software, baseado no histórico de outros projetos.
217.O usuário de um software busca segurança com intuito de garantir que uma aplicação funcione plenamente conforme suas especificações, sendo adequadamente imprescindível atestar, ter planejamento e alcançar um ambiente operacional seguro.De acordo com o texto apresentado e com o conteúdo estudado sobre o teste de segurança, é correto afirmar que entre suas funções está:	avaliar se o sistema operacional apresenta brechas, vazamentos de dados e ausência de criptografia.
218.Os defeitos normalmente são introduzidos na transformação de informações entre as diferentes fases do ciclo de desenvolvimento de um software, assim o planejamento dos testes deve ocorrer em diferentes níveis e em paralelo ao desenvolvimento do software.Considerando essas informações e o conteúdo estudado sobre teste de unidade, pode-se afirmar que:	analisam-se partes menores de um software, para verificar se o funcionamento está adequado.
219.Os gestores de projetos de software acreditam que a utilização de métricas tem sido uma ferramenta essencial na produção de um sistema, pois se consegue medir as características do produto com objetivo de descobrir se os requisitos foram atendidos e se possuem um grau de qualidade satisfatório.Diante dessas informações e do conteúdo estudado sobre métricas de software, em relação às vantagens das aplicações de métricas, é correto afirmar que:	com base nos resultados das métricas, é possível fazer estimativas de tempo, custo e escopo.
220.Os métodos de desenvolvimento e complexidade dos softwares são altamente passíveis de erros. Esses erros podem ocorrer devido a problemas na especificação dos requisitos, na modelagem de negócio, no modo que a funcionalidade deve ser desempenhada, na complexidade do sistema e na mudança de requisitos.Considerando essas informações e o conteúdo estudado sobre teste de software, analise as afirmativas a seguir:I. A atividade de teste de software consiste em avaliar a possibilidade e a existência de erros no sistema para que possam ser solucionados ou evitados futuramente.II. É recomendada a execução da atividade de teste de software quando o sistema estiver em produção, para assim resolver as falhas do sistema.III. O teste de software é o processo de execução de um produto para determinar se ele alcançou as especificações e se está funcionando corretamente para o ambiente para o qual foi projetado.IV. O teste de software tem como objetivo revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e corrigidas pela equipe de desenvolvimento antes da entrega final.Está correto apenas o que se afirma em:	I, III e IV.

PERGUNTAS	RESPOSTAS
221.Os principais processos prescritos pelo CMMI para gerenciar riscos são denominados áreas de processos. Assim, sobre processos e categorias de processos, sabe-se que as áreas de processo do CMMI podem ser agrupadas em determinadas categorias.Considerando essas informações e o conteúdo estudado sobre o CMMI, pode-se afirmar que as categorias são:	gerenciamento de processo; gerenciamento de projeto; engenharia e suporte.
222.Os testes de caixa-preta são realizados na interface externa e analisam os erros que ocorrem no programa em suas ações por fora, abrangendo o funcionamento do programa e comportamento de botões e formulários na área que é comandada por usuários do software.De acordo com o texto e o conteúdo estudado sobre técnicas de teste de software caixa-preta, analise as afirmativas a seguir.I. O Nível Unidade é o primeiro teste da técnica caixa-preta a ser feito. Ele testa os códigos das unidades.II. Botões de mudança de tela que não direcionam para a tela correta são um exemplo de erro avaliado pelo teste da caixa-preta.III.	II e III.
223.Para garantir o funcionamento do programa, se fazem necessárias medidas de segurança. É preciso, nesse sentido, checar e garantir que as ferramentas de acesso e suas funções estejam em perfeito funcionamento, recebendo os comandos e executando as aplicações básicas.De acordo com o texto apresentado e com o conteúdo estudado, é correto afirmar que o teste de funcionalidade:	é responsável por verificar o funcionamento de um programa.
224.Para obter melhor desempenho nas funcionalidades de um software são necessárias ferramentas de suporte boas e confiáveis, que é o caso da Selenium, uma das melhores ferramentas de testes funcionais para aplicações que estão no ar. Porém, não é profunda em testes para códigos Java.Considerando essas informações e de acordo com o conteúdo estudado, é correto afirmar que a ferramenta Selenium:	tem a finalidade de realizar testes em navegadores e sites.
225.Pensando no sistema vantajoso em relação aos testes automatizados, para os engenheiros e programadores de software, em como a diminuição de erros e agilidade nos processos são vantagens no desenvolvimento de software, é importante estudar bons métodos, métricas e boas práticas para o funcionamento da aplicação. Considerando essas informações e de acordo com o conteúdo estudado, é correto afirmar que as três abordagens de teste de software são:	Big Bang, Top-down e Bottom-up.
226.Qualidade de Software é um processo que focaliza todas as etapas do desenvolvimento de um sistema e os componentes produzidos, com o objetivo de garantir a conformidade dos processos e dos produtos, prevenindo e eliminando futuros defeitos. A manutenibilidade é uma categoria que a ISO define atributos que devem ser seguidos para se manter a qualidade de software. Diante dessas informações e o conteúdo estudado sobre qualidade de software, em relação à categoria de manutenibilidade é correto afirmar que:	esta categoria analisa se há facilidade para se fazer correções, nas atualizações e alterações do produto.
227.Qualidade de software refere-se às características que os desenvolvedores especificam para um produto, que engloba o grau de atendimento das funcionalidades e características especificadas no modelo de requisitos. A qualidade de um sistema deve atender as seis amplas categorias para avaliação de qualidade do software. Diante dessas informações e o conteúdo estudado sobre qualidade de software. em relação à categoria de portabilidade, é correto afirmar que:	essa categoria mede como o sistema pode ser transferido, adaptado em outros ambientes e infraestruturas.
228.Quando as empresas adotam as metodologias ágeis, as métricas adquirem um papel importante. Elas ajudam a equipe a obter um entendimento mais amplo do processo de desenvolvimento do software, auxiliam a melhorar os acertos e medem o desempenho e a eficiência das soluções. Com as métricas, é possível acompanhar, monitorar e avaliar o sucesso ou a falha de vários processos de negócios.Diante dessas informações e do conteúdo estudado sobre métricas, é correto afirmar que o Release Burndown:	é uma ferramenta de medição para acompanhamento do escopo do trabalho num todo, com objetivo de mostrar uma visão geral do projeto de software.

PERGUNTAS	RESPOSTAS
229. Revisões e inspeções são atividades que têm o intuito de controlar o nível de qualidade dos entregáveis de projeto. Envolvendo um exame do software, de sua documentação e de seus registros de processos, a fim de descobrir erros e omissões, e verificar se foram seguidos os padrões de qualidade. Considerando essas informações e o conteúdo estudado sobre Revisões e inspeções, analise as afirmativas a seguir: I. Revisões e inspeções são normalmente utilizadas junto com testes de programa, pois é uma parte do processo geral de validação e verificação de software. II. No processo de uma revisão, um grupo de pessoas examina o software e a documentação associada em busca de prováveis problemas e não conformidades com padrões. III. A equipe gera documentos de avaliação da entrega conforme o nível de qualidade solicitado e, com isso, toma decisões sobre a entrega. IV. Revisões e inspeções não devem ser utilizadas junto com testes de programa, pois são parte do processo de software que deve ser tratada separadamente. Está correto apenas o que se afirma em:	I, II e III.
230. Sabe-se que durante a elaboração do software os defeitos que aparecem podem advir das atividades do próprio programador que o desenvolve, mesmo que sejam utilizados os métodos e as ferramentas apropriadas, os erros técnicos podem continuar presentes, por isso faz-se necessário que haja a realização dos testes de softwares também nas suas etapas de elaboração. Considerando essas informações e o conteúdo estudado sobre teste de integração, pode-se afirmar que:	este teste é utilizado após aplicar-se o teste de unidade.
231. São características de um dos estágios do modelo CMM: referenciamento de requisitos, planejamento de projeto, monitoramento e controle de projeto, gerenciamento de fornecedores, medição e análise, garantia da qualidade do processo e do produto, gerenciamento de configuração. Considerando essas informações e o conteúdo estudado sobre o modelo CMM, pode-se afirmar que as características descritas são parte do estágio:	gerenciado.
232. São sete tipos de teste de software apresentados na unidade de estudo: funcionalidade, desempenho, usabilidade, segurança, configuração, falhas e recuperação e, por último, carga ou stress. Cada teste pertencente às técnicas, podendo ser: caixa-preta, caixa-branca ou ambas. Considerando essas informações e o conteúdo estudado sobre testes de software, pode-se dizer que:	o teste alfa é feito para avaliar a opinião de um pequeno grupo de pessoas, contendo nele pessoas conhecidas.
233. Se um dos principais objetivos de uma corporação, na qual há um forte comprometimento organizacional, é implementar processos robustos de desenvolvimento de sistemas que envolvam lidar com requisitos, adotar o CMMI-DEV, versão 1.3, é uma boa opção. Nesse modelo, na abordagem por estágios, as atividades de elicitar, analisar e estabelecer os requisitos do cliente e dos componentes do software caracterizam um determinado processo. Considerando essas informações e o conteúdo estudado do modelo CMMI, pode-se afirmar que o processo descrito é denominado:	desenvolvimento de requisitos.
234. Segundo Pressman (2011), na qualidade do software, há as atividades de: inspeções, revisões e testes utilizados ao longo do processo de software, para verificar que cada produto satisfaz os requisitos estabelecidos, são atividades que compõem qual conceito seguir.	controle de qualidade.
235. Testar se o software instalado funciona como programado, se executa suas funções como o planejado, em diferentes hardwares, sob diversas condições e prováveis instabilidades, se está adaptável a qualquer tamanho de memória ou possíveis interrupções de rede são condições fundamentais para avaliar a qualidade e confiabilidade de um programa. De acordo com o texto apresentado e o conteúdo estudado, é correto afirmar que a meta do teste de configuração é:	ser útil para revelar a funcionalidade do software em diferentes sistemas operacionais, pares e máquinas.

PERGUNTAS	RESPOSTAS
236. TestMaster é uma ferramenta para guardar testes, criar relatórios e é ainda dedicado à automação de teste. Para abrir, é necessário ter o Apache como servidor web, aceita importações de Word e arquivos SQL e CSV. No TestMaster pode-se criar gavetas com status: aprovado ou reprovado. Considerando essas informações e o conteúdo estudado sobre TestMaster, é correto afirmar que o TestMaster:	é uma ferramenta usada para o teste de projetos.
237. Todos os desenvolvedores estão suscetíveis a erros de programação, já que esses sistemas possuem alta complexidade. Para solucionar e evitar tal problemática existe uma atividade em que se pode avaliar, testar e corrigir tais problemas denominada como teste de software, a qual é feita de diversas maneiras e usando diversas metodologias. Considerando essas informações e o conteúdo estudado sobre teste nos softwares, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s): I. () A execução do teste nos softwares antigamente era feita pelos próprios desenvolvedores dos sistemas.	V, V, V, F.
238. Top-down é utilizada em vários processos e programas feitos em Java, que possui orientação a objetos. Assim, seus códigos usam herança, polimorfismo e sobrecarga de método; logo, para aplicações Java, top-down pode ser uma abordagem bem-vinda. Considerando essas informações e de acordo com o conteúdo estudado sobre estratégias e abordagem de teste “top-down”, é correto afirmar que:	a burocratização de ser um teste em ordem decrescente o torna lento para testar níveis inferiores.
239. Um software de qualidade deve atender seis categorias definidas pela ISO/IEC 9126 (NBR13596). Eficiência é a que se preocupa em medir se o consumo dos recursos e o tempo utilizado são compatíveis com o nível de desempenho requerido para o produto.	esse atributo mede quanto leva para processar um recurso do sistema.
240. Um software de qualidade é fácil de usar, funciona corretamente, é de fácil manutenção e mantém a integridade dos dados para evitar possíveis falhas. Por este motivo, a ISO dividiu a qualidade de software em seis categorias que têm como objetivo servir base na avaliação de produto de software. Diante dessas informações e o conteúdo estudado sobre qualidade de software, é correto afirmar em relação à categoria de confiabilidade que:	nela é descrita a capacidade de tolerância à falhas e se o desempenho do sistema se manterá dentro do esperado.
241. Uma estratégia de teste de software integra métodos de projeto de casos de teste em uma série planejada de passos que resulta na construção bem-sucedida de um software. Sabe-se que durante a elaboração do software os defeitos que aparecem podem advir das atividades do próprio programador que o desenvolve, mesmo que sejam utilizados os métodos e as ferramentas apropriadas. Considerando essas informações e o conteúdo estudado sobre teste de software, analise os itens a seguir sobre as estratégias de teste para softwares convencionais: I. Uma estratégia de teste adota uma visão incremental do teste, começando com o teste de unidades individuais de programa, assim por diante. II.	I, III e IV.
242. Teste de software é o processo de execução de um produto para determinar se ele alcançou as especificações e se está funcionando corretamente para o ambiente para o qual foi projetado. Tem como objetivo revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e corrigidas pela equipe de desenvolvimento.	os métodos de desenvolvimento, devido à complexidade dos softwares, são passíveis de erros.
243. Uma estratégia de teste de software integra métodos de projeto de casos de teste em uma série planejada de passos que resulta na construção bem-sucedida de um software. Sabe-se que durante a elaboração do software os defeitos que aparecem podem advir das atividades do próprio programador que o desenvolve, mesmo que sejam utilizados os métodos e as ferramentas apropriadas.	I, III e IV.

Boas Provas!