

Questões Gerais e Modelos de Ciclo de Vida

TRE/MT (CESPE 2010)

41 Existem vários modelos de desenvolvimento de software, cada um com suas particularidades. A respeito desse assunto, assinale a opção correta.

- A) No modelo de desenvolvimento em cascata, a primeira etapa é a modelagem do sistema.
- B) No modelo de desenvolvimento incremental, da mesma maneira que no modelo cascata, é possível a solução dos riscos logo no início do projeto.
- C) A metodologia de prototipagem evolutiva é uma abordagem que visualiza o desenvolvimento de concepções do sistema conforme o andamento do projeto, por meio de protótipos visuais.
- D) O modelo de desenvolvimento UML é uma metodologia de desenvolvimento orientada a objetos que gera protótipos muito rapidamente, em função das ferramentas case disponíveis.
- E) O modelo de desenvolvimento em espiral, que tem a codificação como segunda etapa, gera o código do sistema muito mais rapidamente que o modelo de prototipação.

TRE/BA – Análise de Sistemas (CESPE 2010)

Com relação à engenharia de software, julgue os itens a seguir.

51 Um modelo de processo de software consiste em uma representação complexa de um processo de software, apresentada a partir de uma perspectiva genérica.

52 Entre os desafios enfrentados pela engenharia de software estão lidar com sistemas legados, atender à crescente diversidade e atender às exigências quanto a prazos de entrega reduzidos.

TRE/BA – Programação de Sistemas (CESPE 2010)

Em uma visão restritiva, muitas pessoas costumam associar o termo software aos programas de computador. Software não é apenas o programa, mas também todos os dados de documentação e configuração associados, necessários para que o programa opere corretamente. A respeito de engenharia de software, julgue os itens de 61 a 65.

61 A engenharia de software está relacionada com todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até sua manutenção, depois que este entrar em operação. A engenharia de sistemas diz respeito aos aspectos do desenvolvimento e da evolução de sistemas complexos, nos quais o software desempenha um papel importante.

62 Na engenharia de software baseada em componentes, na qual se supõe que partes do sistema já existam, o processo de desenvolvimento concentra-se mais na integração dessas partes que no seu desenvolvimento a partir do início. Essa abordagem é baseada em reuso para o desenvolvimento de sistemas de software.

63 Em estágio inicial do processo de elicitação e análise de requisitos, deve-se decidir a respeito dos limites do sistema. Os requisitos não funcionais descrevem o que o sistema deve fazer, enquanto os requisitos funcionais estão diretamente relacionados às funções específicas fornecidas pelo sistema.

Correios (CESPE 2011)

Acerca de engenharia de software, que permite a criação, de maneira econômica e confiável, de software que trabalhe eficientemente em máquinas reais, julgue os próximos itens.

81 Para que se obtenha sucesso na utilização do Scrum, o cliente deve se

tornar parte da equipe de desenvolvimento do software, participando diretamente do processo.

82 A engenharia da usabilidade é aplicada em qualquer tipo de interface, como, por exemplo, sítios web, software e desktop.

Uma das principais fases da engenharia de usabilidade é a que permite o conhecimento do usuário ao qual o software se destina.

83 A prototipação possibilita que o usuário teste as características do produto final. Dos dois tipos de prototipação, a horizontal é focada nas funcionalidades que permitem ao usuário ver somente detalhes de determinadas partes do sistema.

86 Entre as metodologias ágeis para o desenvolvimento de software, o Scrum permite a criação de equipes autoorganizadas e, conseqüentemente, possibilita o incentivo à comunicação verbal entre todos os membros da equipe. Da mesma forma que as abordagens típicas de Project Management Body of Knowledge ou PRINCE2, o Scrum caracteriza-se por apresentar uma abordagem elementar do gerenciamento de projetos.

SECGE/PE – CESPE 2011

44 Acerca da engenharia de software, assinale a opção correta.

A) A engenharia de software, que tem como alicerce os métodos, é formada de várias camadas, tais como ferramentas, métodos e processo.

B) Apesar de importante no contexto de produtos, a qualidade não é tratada na engenharia de software e, sim, no gerenciamento de projetos.

C) Análise e projeto são duas ações da engenharia de software. A definição de arquitetura e componentes são tarefas da análise, enquanto levantamento e especificação de requisitos são

tarefas do projeto.

D) A engenharia de software visa obter softwares econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais.

E) A engenharia de software aplica abordagem sistemática, disciplinada e não quantificável para desenvolvimento e manutenção de software.

45 A respeito de modelos de desenvolvimento, assinale a opção correta:

A) O modelo de desenvolvimento que se baseia em componentes é um método ágil de desenvolvimento, incompatível com a orientação a objetos.

B) O modelo clássico ou cascata é iterativo, evolutivo e sequencial, tendo como fases requisitos, análise, projeto, implementação, testes e implantação.

C) A prototipagem é um modelo de desenvolvimento que permite usar partes do protótipo construído na solução final do produto, ainda que o protótipo não possibilite obter requisitos do software.

D) O modelo incremental combina elementos do modelo cascata aplicado de maneira iterativa, englobando planejamento, modelagem, construção e implantação.

E) O RAD (rapid application development) é um modelo iterativo que tem como fases a modelagem do projeto, a construção do protótipo, engenharia e implantação.

INMETRO (CESPE 2010)

33 Prototipação é uma técnica utilizada na abordagem evolutiva de desenvolvimento de software, por meio da qual são produzidas versões iniciais (ou protótipos) do sistema com as quais é possível realizar verificações e experimentações antes de se trabalhar na versão final do software.

Acerca desse tema, assinale a opção correta.

- A) Um dos benefícios da prototipação é a documentação normalmente gerada, que facilita a manutenção dos sistemas a longo prazo e a elaboração de casos de teste.
- B) O uso de ferramenta CASE para elaboração de um protótipo garante o sucesso do produto de software desenvolvido.
- C) As versões iniciais de um sistema, denominadas protótipos de baixa fidelidade, permitem a verificação completa de erros e a identificação de requisitos funcionais e não funcionais.
- D) Um dos riscos da prototipação é o usuário confundir o protótipo com o sistema verdadeiro e criar falsas expectativas com relação a prazos e recursos.
- E) Na abordagem evolutiva para desenvolvimento de software, um protótipo do software é produzido e utilizado para identificar possíveis problemas com os requisitos, sendo descartado logo em seguida, e o desenvolvimento do software propriamente dito é, então, iniciado.

41 Um modelo de processo de software, como os modelos cascata, iterativo e rational unified process (RUP), consiste em uma representação abstrata de um processo de software. Abordagens como extreme programming (XP) e Scrum propõem uma forma mais ágil de desenvolver produtos de software. A esse respeito, assinale a opção correta.

- A) O RUP é constituído de quatro fases, nas quais as iterações ocorrem: concepção, elaboração, construção e transição. Na primeira, identificam-se todas as entidades externas que irão interagir com o sistema e definem-se essas interações. Na segunda, são elaborados o modelo de requisitos para o sistema, a descrição da arquitetura e o plano de desenvolvimento de software. No final da fase de construção, que se relaciona ao projeto e programação do sistema, este deve estar em

funcionamento e a documentação associada pronta. A fase de transição envolve os testes e a transferência do sistema para o ambiente real.

- B) O modelo cascata representa as fases do processo separadas e encadeadas, tais como especificação de requisitos, projeto de software, implementação, teste, entre outras. A fase seguinte não pode começar antes que a fase anterior tenha terminado. O maior problema do modelo cascata é a divisão inflexível do projeto em estágios distintos, as iterações são onerosas e envolvem retrabalho.

- C) No modelo em espiral, um exemplo de modelo iterativo, cada loop da espiral representa uma fase do processo de software. Nesse modelo, os riscos não são considerados, pois podem impactar o projeto.

- D) XP engloba princípios como trabalhar com os clientes, utilizar metáforas, manter reuniões curtas, programar por pares, simplicidade, fazer releases em incrementos pequenos e integração contínua. O teste, uma importante atividade da engenharia de software, não é abordado na XP, o que constitui a sua maior limitação.

- E) Embasado nas melhores práticas aceitas pelo mercado, o Scrum não é um processo ou uma técnica para o desenvolvimento de produtos, mas sim um framework que indica diversos processos e técnicas. Ele emprega uma abordagem iterativa e incremental, e trabalha com os seguintes conceitos: backlog do produto, uma lista priorizada de tudo que pode ser necessário no produto; product owner (única pessoa responsável pelo gerenciamento do backlog do produto); sprint (iteração); e times, cujo tamanho ideal, indicado pela abordagem, está entre 15 a 20 pessoas, de forma a facilitar a gestão.

FUB (CESPE 2011)

83 Os diversos modelos de processo de software disponíveis permitem a representação abstrata de um processo de software sob diferentes perspectivas. No modelo evolucionário, sob a perspectiva da

arquitetura, a velocidade de desenvolvimento faz que a produção de documentos que reflitam cada versão do sistema seja economicamente inviável, gerando problemas na validação independente de sistemas.

BASA (CESPE 2010)

74 Os princípios de engenharia de software definem a necessidade de formalidades para reduzir inconsistências e a decomposição para lidar com a complexidade.

75 Para garantir o desenvolvimento de qualidade, é suficiente que a equipe tenha as ferramentas mais atuais de engenharia de software e os melhores computadores.

79 No modelo em cascata, o projeto segue uma série de passos ordenados. Ao final de cada projeto, a equipe de projeto finaliza uma revisão. O desenvolvimento continua e, ao final, o cliente avalia a solução proposta.

TRE/MT (CESPE 2010)

31 Um processo de desenvolvimento de software é um conjunto de atividades, parcialmente ordenadas, com a finalidade de se obter um produto de software. Cada processo tem suas particularidades, entretanto, podem-se destacar atividades que são comuns à maioria dos processos. Com relação a processos de desenvolvimento de software, assinale a opção correta.

A) O levantamento de requisitos tem como objetivo compreender o problema a ser resolvido e identificar necessidades. Os requisitos podem ser funcionais, que definem as funcionalidades do sistema, ou não funcionais, que não estão relacionados às funcionalidades.

B A análise tem como foco construir uma estratégia de solução. Os modelos construídos nessa fase devem ser verificados e validados. A verificação tem como objetivo assegurar que as

necessidades do cliente estão sendo atendidas pelo sistema, enquanto a validação tem o objetivo de analisar se os modelos estão em conformidade com os requisitos definidos.

C O projeto produz uma descrição computacional do software sem restrições de tecnologia, ou seja, aspectos físicos e dependentes de implementação não são considerados.

D Na fase de implementação, o sistema é codificado, ou seja, a descrição computacional obtida na fase de projeto é traduzida para código executável, por meio do uso de uma ou mais linguagens de programação.

E Na fase de implantação, o sistema é testado, empacotado, distribuído e instalado no ambiente do cliente.

ANAC (CESPE 2009)

Segundo Ian Sommerville, (Engenharia de software, 2007, p. 5), a engenharia de software é uma disciplina de engenharia relacionada a todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até sua manutenção. Acerca da engenharia de software, julgue os itens a seguir.

61 O termo engenharia pretende indicar que o desenvolvimento de software submete-se a leis similares às que governam a manufatura de produtos industriais em engenharias tradicionais, pois ambos são metodológicos.

64 A fase de projeto define o que o software deve fazer, enquanto a fase de eliciação de requisitos define como o software deve atingir seus requisitos.

65 Requisitos descrevem um acordo ou contrato entre duas partes, especificando, entre outros aspectos, o que o sistema de software deve fazer para ser aprovado em um teste de aceitação.

EMBASA (CESPE 2009)

Acerca de princípios de engenharia de software, julgue os itens a seguir.

67 Um modelo de processo de software descreve os processos que são realizados para atingir o seu desenvolvimento. A notação para as tarefas, os artefatos, os atores e as decisões varia conforme o modelo de processo utilizado.

ANTAQ (CESPE 2009)

101 O modelo em espiral, que descreve o processo de desenvolvimento de um software, apresenta uma espiral em que cada loop representa uma fase distinta desse processo. A ausência de risco nesse modelo o diferencia dos demais modelos de software.

UNIPAMPA (CESPE 2009)

71 O modelo de desenvolvimento espiral foi desenvolvido somente para abranger as melhores características do ciclo de vida clássico.

72 Uma das características do modelo de desenvolvimento em cascata é envolver a sequência de atividades. Nesse modelo, cada fase transcorre completamente e seus produtos são vistos como entrada para uma nova fase.

73 No modelo de desenvolvimento prototipagem, um protótipo é desenvolvido para ajudar no entendimento dos requisitos do sistema.

74 No modelo de desenvolvimento em espiral, a análise de riscos não impacta na elaboração de um produto ou protótipo.

75 No modelo de desenvolvimento incremental, a cada iteração são realizadas várias tarefas. Na fase de análise, pode ser feito o refinamento de requisitos e o refinamento do modelo conceitual.

SECONT-ES (CESPE 2009)

75 Em um processo iterativo de desenvolvimento de software, a essência é que a especificação seja desenvolvida conjuntamente com o software. A entrega incremental é um modelo de processo projetado para apoiar a iteração de processo, em que a cada incremento é gerado um novo protótipo do sistema, enquanto os requisitos da próxima entrega são analisados. Depois de gerados todos os incrementos, o sistema está completo e, somente então, pode ser colocado em operação.

ANATEL (CESPE 2009)

95 Entre os modelos de ciclo de vida de software, o modelo espiral possui maior proximidade com as práticas da engenharia clássica empregadas, por exemplo, na construção de casas, quando comparado aos modelos cascata e de componentes reusáveis.

INMETRO (CESPE 2009)

85 Em um processo de desenvolvimento em cascata, os testes de software são realizados todos em um mesmo estágio, que acontece após a finalização da fase de implementação.

86 Em uma empresa que tenha adotado um processo de desenvolvimento de software em cascata, falhas no levantamento de requisitos têm maior possibilidade de gerar grandes prejuízos do que naquelas que tenham adotado desenvolvimento evolucionário.

Ministério das Comunicações (CESPE 2009)

67 No modelo de processo de desenvolvimento em cascata, ciclo de vida é dividido em uma série de fases. Na fase de projeto, é definida tipicamente a arquitetura do sistema, quando estilos de arquitetura e padrões de projeto podem ser empregados. A princípio, em cada fase, podem ser produzidos artefatos variados, por exemplo: planos, códigos e modelos.

68 No modelo de processo de desenvolvimento iterativo, o ciclo de vida pode ser dividido em iterações. Em cada interação pode ocorrer análise, projeto, implementação e teste.

O desenvolvimento iterativo tende a reduzir riscos, pois os componentes do software são progressivamente integrados.

SERPRO (CESPE 2008)

63 O modelo em cascata consiste de fases e atividades que devem ser realizadas em sequência, de forma que uma atividade é requisito da outra.

64 O modelo iterativo e o modelo em espiral possuem características semelhantes: ambos permitem que as atividades do processo sejam planejadas e avaliadas ao longo do ciclo de vida.

65 Para a especificação de software e verificação de sistemas, uma alternativa que se fundamenta na matemática discreta e na lógica é o modelo incremental.

67 Os modelos ágeis são muito mais rápidos e eficientes que os modelos incremental e iterativo, não partilhando aqueles, portanto, das visões adotadas por estes.

TST (CESPE 2008)

No que se refere aos ciclos de vida e metodologias de desenvolvimento de software, julgue os itens subseqüentes.

93 No modelo de desenvolvimento seqüencial linear, a fase de codificação é a que gera erros de maior custo de correção.

94 O modelo de desenvolvimento em espiral permite repensar o planejamento diversas vezes durante o desenrolar do projeto.

TJDFT (CESPE 2008)

Acerca da engenharia de software e de metodologias e ciclos de desenvolvimento de software, julgue os itens subseqüentes.

69 O modelo em espiral é um modelo de processos de software que reúne a natureza iterativa da prototipação com os aspectos sistemáticos e controlados do modelo seqüencial linear.

70 O modelo de desenvolvimento por prototipação é caracterizado pela ausência de métricas de controle, dada a natureza experimental do desenvolvimento e do produto obtido.

73 O desenvolvimento com base em componentes é uma abordagem típica da programação estruturada e tem foco na produção de bibliotecas de software reutilizáveis.

DATAPREV (CESPE 2006)

<i>Software requirements</i>
<i>Software design</i>
<i>Software construction</i>
<i>Software testing</i>
<i>Software maintenance</i>
<i>Software configuration management</i>
<i>Software engineering management</i>
<i>Software engineering process</i>
<i>Software engineering tools and methods</i>
<i>Software quality</i>

A figura acima apresenta um sumário das áreas usualmente reconhecidas como sendo praticadas na engenharia de software. Considere as seguintes atribuições profissionais exigidas em uma organização: desenvolver sistemas em ambiente de grande porte, atuando preferencialmente nas fases de implementação de códigos e transição do processo de desenvolvimento adotado, utilizando-se de metodologia e

ferramentas CASE/IDE homologadas na organização, visando à racionalização e à otimização dos processos do cliente.

Julgue os itens de 86 a 90, acerca dos conceitos e das áreas de engenharia de software listadas e seus relacionamentos com as atribuições profissionais acima descritas.

86 “desenvolver sistemas em ambiente de grande porte”, que constitui uma das atribuições sob consideração, está, com relação às áreas praticadas na engenharia de software, mais associado a “Software configuration management” que a “Software construction”.

87 A “implementação de códigos” está mais associada, como área de prática de engenharia de software, a “Software design” que a “Software maintenance”.

88 Atuar na fase de “transição do processo de desenvolvimento adotado”, caso corresponda a colocar em estado operacional o sistema desenvolvido, está mais relacionado, como prática de engenharia de software, a “Software engineering process” que a “Software testing”.

89 Visar à “racionalização e à otimização dos processos do cliente” está mais associado a “Software requirements” que a “Software engineering process”.

90 Os termos ferramenta “CASE” e ferramenta de “IDE” são sinônimos.

Metodologias Ágeis

TCU (CESPE 2010)

114 O processo XP (extreme programming) envolve a realização das atividades de planejamento, de projeto, de codificação e de teste.

115 A atividade de planejamento XP inclui a criação das denominadas histórias de usuário, nas quais devem ser descritas as

características e as funcionalidades requeridas para o software em desenvolvimento.

116 A atividade de projeto é uma desvantagem do processo XP, pelo fato de requerer uma quantidade de produtos de trabalho considerada excessiva pela comunidade de desenvolvimento de software.

STM (CESPE 2011)

Julgue os itens seguintes, a respeito de diferentes abordagens para o processo de desenvolvimento de software.

70 O extreme programming (XP), que se inclui entre os métodos ágeis, apresenta, entre outras, as seguintes características: pequenos releases, projeto simples, refactoring, programação em pares e propriedade coletiva.

INMETRO (CESPE 2010)

40 Com relação ao processo Scrum, assinale a opção correta.

A) A Time Box (caixa de tempo) dentro da qual um conjunto de atividades deve ser executado é chamada de Product Backlog.

B) Metodologias ágeis de desenvolvimento de software são iterativas e dirigidas por risco, ou seja, o trabalho é dividido em iterações, que são chamadas de Sprints no caso do Scrum. Antes de iniciar cada Sprint, os riscos de insucesso na atividade são reavaliados e geridos. O gerente do projeto deve ser capaz de reagir ou agir proativamente ante tais riscos.

C) Caso um gerente de projetos deseje verificar quais são as funcionalidades a serem implementadas durante um projeto de software, ele deve obtê-las em uma lista conhecida como Sprint Backlog.

D) Se um gerente de projeto deseja alterar alguns dos requisitos do software durante o desenvolvimento de um Sprint, é recomendável que ele altere o planejamento do Sprint.

E) Se, durante um Sprint Planning Meeting, um profissional da equipe é destacado como responsável por descrever os itens do Product Backlog, então se diz que este profissional assumiu o papel de Scrum Master.

MPU – Desenvolvimento (CESPE 2010)

117 Extreme programming (XP) é embasado em requisitos conhecidos, definidos de antemão, que não sofram muitas alterações, devendo ser usado por equipes de pequeno porte, formadas por representantes de todos os stakeholders.

118 Produto da metodologia Scrum, o documento product backlog contém os requisitos definidos a partir da visão do cliente e é utilizado novamente no final do sprint para revisão ou modificações dos requisitos inicialmente definidos.

BASA (CESPE 2010)

77 Desenvolvimento ágil de software (agile software development) ou método ágil é aplicado, principalmente, a grandes corporações, uma vez que permite produzir grandes sistemas de forma ágil.

78 O Scrum é utilizado, como função primária, para o gerenciamento de projetos de desenvolvimento de software, mas também tem sido usado como extreme programming e outras metodologias de desenvolvimento. Teoricamente, o Scrum pode ser aplicado em qualquer contexto no qual um grupo de pessoas necessite trabalhar juntas para atingir um objetivo comum.

TRE/BA (CESPE 2010)

A respeito das metodologias eXtreme programming (XP) e Scrum, julgue os itens a seguir.

67 A metodologia XP prevê valores e princípios básicos para serem considerados durante o desenvolvimento de software.

Feedback, coragem e respeito são exemplos de valores; mudanças incrementais, abraçar mudanças e trabalho de qualidade são exemplos de princípios básicos.

68 Um princípio chave do Scrum é o reconhecimento de que desafios fundamentalmente empíricos não podem ser resolvidos com sucesso utilizando-se uma abordagem tradicional de controle. O Scrum adota uma abordagem empírica, aceitando que o problema não pode ser totalmente entendido ou definido, focando na maximização da habilidade da equipe de responder de forma ágil aos desafios emergentes.

101 A metodologia Scrum é facilitada por um *scrum master*, que atua como um mediador entre a equipe e qualquer influência desestabilizadora, além de assegurar que a equipe esteja utilizando corretamente as práticas de Scrum, motivando e mantendo o foco na meta da sprint.

TRE/MT (CESPE 2010)

29 Assinale a opção correta acerca de modelos de processo de software.

A) O modelo em cascata é apropriado para software em que os requisitos ainda não foram bem compreendidos, pois é focado na criação de incrementos.

B) Os processos de desenvolvimento ágil de software valorizam mais: indivíduos e interações do que processos e ferramentas;

software funcionando do que documentação compreensível; colaboração do cliente do que negociação contratual; e respostas a mudanças do que planejamento seguido.

C) Desenvolvimento incremental é sinônimo de desenvolvimento iterativo.

D) A programação extrema (eXtreme Programming, ou XP) é um processo de desenvolvimento ágil que permite a ausência do cliente em fases iniciais, como

nas reuniões de planejamento de versão e planejamento de iteração.

E) A programação em pares é uma característica presente em todos os processos de desenvolvimento ágil, como XP e Scrum.

ANTAQ (CESPE 2009)

102 O extreme programming (XP) constitui método ágil de desenvolvimento de software. Uma das práticas que se enquadram nos princípios dos métodos ágeis é a programação em pares, que promove o compartilhamento da autoria do código do sistema. Além dessa vantagem, a programação em pares atua como processo informal de revisão porque cada linha de código é vista por pelo menos duas pessoas

ANAC (CESPE 2009)

63 Extreme Programming é um modelo de processo de desenvolvimento de software para equipes com grande número de pessoas, que desenvolvem software com base em requisitos vagos e que são modificados rapidamente.

ANATEL (CESPE 2009)

96 Entre as metodologias de desenvolvimento de software atualmente empregadas destacam-se as abordagens embasadas no modelo unificado e as abordagens ágeis. O uso das técnicas de test-driven design, refactoring, design patterns e pair programming é, entre os modelos acima, maior nas abordagens do modelo unificado. Por outro lado, o uso de ferramentas CASE-UML é mais comum nas abordagens ágeis.

INMETRO (CESPE 2009)

66 São práticas ou princípios recomendados no modelo de desenvolvimento de software XP (eXtreme Programming) proposto por Kent Beck: programação em pares; semana de trabalho de 40 horas; refatoração sem

pieidade; desenvolvimento orientado a testes TDD (Test Driven Development); e desenvolvimento de metáforas arquiteturas.

Governo do ES (CESPE 2009)

74 Métodos ágeis de desenvolvimento de sistemas foram propostos principalmente para apoiar o desenvolvimento de aplicações de negócios nas quais os requisitos de sistema mudam rapidamente durante o processo de desenvolvimento. Entre esses métodos está o extreme programming, que envolve um número de práticas, como o planejamento incremental, a definição de um ritmo de trabalho sustentável e a divisão das equipes de trabalho por meio da especialização de seus membros.

Min. das Comunicações (CESPE 2009)

69 Os processos ágeis (agile) de desenvolvimento são tipicamente processos iterativos, em que softwares são desenvolvidos em uma série de incrementos. Nesses processos, os usuários finais tipicamente participam na validação dos incrementos, podendo propor alterações e novos requisitos.

74 As seguintes práticas são recomendadas pelo XP: dividir o projeto em iterações; iniciar cada iteração com o seu planejamento; empregar cartões CRC no projeto simplificar o projeto seguir padrões de codificação; freqüentemente, testar e integrar; executar testes de unidade; programar em pares; incentivar participantes a trabalharem em diferentes partes do projeto.

75 No XP, não é necessário detalhar o planejamento para todo o projeto; tal planejamento pode ser atualizado a cada iteração; histórias menos relevantes são realizadas primeiro; as histórias são divididas em tarefas de programação; cada tarefa deve ser extensa de modo a ser executada ao longo de várias iterações.

PRODEST (CESPE 2008)

99 Projetar detalhadamente todo o software antes de iniciar a sua implementação é uma prática recomendada pelo XP. O software deve ser projetado para atender tanto aos requisitos atuais quanto aos potenciais requisitos futuros. Para atingir esse objetivo, são analisados os possíveis cenários de evolução futura e são empregados padrões de projeto para facilitar a manutenção.

100 Constituem práticas recomendadas pelo XP a colocação rápida de uma versão simples em produção, a liberação das novas versões em curtos intervalos de tempo, a programação em duplas, a refatoração (refactor) dos códigos produzidos, a adoção de padrões para a codificação; a integração e o teste contínuos de códigos; a limitação em 40 horas da carga de trabalho semanal.

101 O XP é um processo que visa a um desenvolvimento ágil e portanto não recomenda os testes de unidade, pois eles consomem muitos recursos. Durante o desenvolvimento, o primeiro teste recomendado é o smoke test que foca os detalhes de funcionamento. O smoke test é realizado após as unidades serem integradas. Após o smoke test, é realizado o teste de sistema.

TJ/PE (FCC 2007)

Considere:

- I. Desenvolvimento de um modelo geral.
- II. Construção da lista de funcionalidades.
- III. Plano de liberações com base nas funcionalidades a implementar.
- IV. Projetar com base nas funcionalidades.
- V. Implementar com base nas funcionalidades.

São fases de projetos que seguem o processo projetado por Peter Coad, Erich Lefebvre e Jeff De Luca chamado de

A) MDA

- B) XP
- C) FDD
- D) RUP
- E) MVC

CASE

TRT/RN – Técnico (CESPE 2010)

Com relação às ferramentas CASE, julgue os itens subsequentes.

76 AnalystDesigner, Software through Pictures e System Architect são exemplos de ferramentas upper CASE disponíveis no mercado.

77 As ferramentas CASE que auxiliam o desenvolvedor durante o processo de levantamento de necessidades, de análise e de elaboração de projeto são denominadas lower CASE ou ferramentas back-end.

TRT/5 (CESPE 2009)

Existem diversas ferramentas para o desenvolvimento de software, entre elas as ferramentas CASE. Acerca dessas ferramentas, julgue os itens subsequentes.

61 Com a evolução das técnicas de desenvolvimento de software, tornou-se necessário o uso de técnicas e metodologias que fossem capazes de representar um software em sua fase de projeto.

62 Software criado com suporte por ferramentas CASE devem ser independentes e isoladas, pois é desejável isolar o problema do software a ser desenvolvido.

63 Ferramentas CASE que utilizam metodologia de desenvolvimento de sistemas em banco de dados baseado no modelo entidade-relacionamento são capazes de gerar código SQL a partir de um modelo físico de dados.

64 Ferramentas CASE que suportam o desenvolvimento orientado a objeto não se integram com bancos relacionais porque o mecanismo de geração de código SQL não é orientado a objeto.

65 Um projeto de desenvolvimento de software pode usar mais de uma ferramenta CASE. O principal requisito é que todas sejam orientadas a objeto e proporcionem integração com a Internet como requisito mínimo.

DETRAN (CESPE 2009)

98 As ferramentas CASE têm por objetivo apoiar atividades do processo de software, tais como análise de requisitos, modelagem de sistema, depuração e teste. Essas ferramentas podem incluir geradores de código que, a partir do modelo do sistema, gerem automaticamente os códigos-fonte.

MDS (CESPE 2009)

118 Uma ferramenta CASE (computer-aided software/system engineering) integrada, também chamada de I-CASE, permite a transferência de informação, como modelos, programas e documentos, de uma ferramenta para outra. Entretanto, uma I-CASE não permite a mudança de um estágio do processo de engenharia de software para outro

STF (CESPE 2008)

104 A manipulação de atributos de requisitos demanda maior suporte de ferramentas CASE verticais que de ferramentas CASE horizontais, sendo contrária a situação no caso de manipulação de glossários.

BASA (CESPE 2010)

Com relação às ferramentas CASE, julgue os itens a seguir.

80 A auditoria do código-fonte garante o cumprimento de padrões de linguagem que as ferramentas CASE promovem para alcançar a qualidade do software.

81 As técnicas de modelagem que utilizam métodos de análise estruturada e projeto estruturado (SA/SD) são técnicas CASE para projetos pequenos de software de baixa complexidade.

82 As ferramentas CASE permitem melhor gerenciamento de projetos por meio da criação de estruturas analíticas de projeto, (EAP); definem a estrutura de divisão de trabalho; planejam a programação do projeto e permitem o acompanhamento de projetos de software.

83 As linguagens de consulta a bancos de dados, os compiladores, as linguagens de programação convencionais e os ambientes de programação orientada a objetos são exemplos típicos de ferramentas CASE de apoio e manutenção.

84 A prototipação é apoiada por meio de ferramentas CASE que permitem a criação de projetos de dados e a integração com layouts de relatórios e com a interface e apresentação de tela.

85 As ferramentas CASE não possuem subdivisão de acordo com a finalidade de manutenção

TCE/TO (CESPE 2008)

46 Ferramentas CASE que oferecem serviços utilizados durante todas as etapas de desenvolvimento de um software são classificadas como

- A) vertical.
- B) diagonal.
- C) transversal.
- D) meridional.
- E) horizontal.

Gabaritos:

Questões Gerais e Modelos de Ciclo de Vida

TRE/MT (CESPE 2010) - 41 C

TRE/BA – Análise de Sistemas (CESPE 2010)
51 E, 52 C

TRE/BA – Programação (CESPE 2010)
61 C, 62 C, 63 E

Correios (CESPE 2011)
81 C, 82 C, 83 E, 86 E

SECGE/PE (CESPE 2011)
44 D, 45 D

INMETRO (CESPE 2010)
33 D, 41 B

FUB (CESPE 2011)
83 C

BASA (CESPE 2010) - 74 C, 75 E, 79 E

TRE/MT - 31 D

ANAC (CESPE 2009) - 61 C, 64 E, 65 C

EMBASA (CESPE 2009) - 67 C

ANTAQ (CESPE 2009) - 101 E

UNIPAMPA (CESPE 2009)
71 E, 72 C, 73 C, 74 E, 75 C

SECONT-ES (CESPE 2009) - 75 E

ANATEL (CESPE 2009) - 95 E

INMETRO (CESPE 2009) - 85 E, 86 C

Ministério das Comunicações (CESPE 2009)
67 C, 68 C

SERPRO (CESPE 2008) –
63 C, 64 C, 65 E, 67 E

TST (CESPE 2008) - 93 E, 94 C

TJDFT (CESPE 2008)
69 C 70 E 73 E

DATAPREV (CESPE 2006)
86 E, 87 E, 88 E, 89 C, 90 E

Metodologias Ágeis

TCU (CESPE 2010)
114 C, 115 C, 116 E

STM (CESPE 2011)
70 C

INMETRO (CESPE 2010)
40 B

MPU – Desenvolvimento (CESPE 2010)
117 E, 118 C

BASA (CESPE 2010) - 77 E, 78 C

TRE/BA (CESPE 2010) - 67 C, 68 C, 101 C

TRT/MT (CESPE 2010) - 29 B

ANTAQ (CESPE 2009) - 102 C

ANAC (CESPE 2009) - 63 E

ANATEL (CESPE 2009) - 96 E

INMETRO (CESPE 2009) - 66 C

Governo do ES (CESPE 2009) - 74 E

Min. Comunicações (CESPE 2009) - 69 C, 74 C, 75 E

PRODEST (CESPE 2006) - 99 E, 100 C, 101 E

TJ/PE (FCC 2007)
C

CASE

TRT/RN - Técnico (CESPE 2010)
76 C, 77 E

TRT/5 (CESPE 2009) - 61 C, 62 E, 63 C, 64 E, 65 E

DETRAN (CESPE 2009) - 98 C

MDS (CESPE 2009) - 118 E

STF (CESPE 2008) - 104 C

BASA (CESPE 2010)

80 C, 81 E, 82 C, 83 E, 84 C, 85 E

TCE/TO (CESPE 2008) - 46 E