



Engenharia de Software Cesgranrio

Prof. Lúcio Camilo
Email: luciocamilo@gmail.com
<http://www.itnerante.com.br/profile/LucioCamilo>



Lúcio Camilo

- Resumo – CV
- Analista de Sistemas do Banco do Brasil – Cedido à PREVI
- Especialidade: Administrador de Servidor de Aplicações
- Pós Graduado em Engenharia de Software
- Autor do Livro “Android para Desenvolvedores”, Editora Brasport
- MBA Gerenciamento de Projetos
- Certificações Profissionais:
 - SCJP, OCWD, OCJA Part I
 - RHSA, Big IP Essentials e Advanced

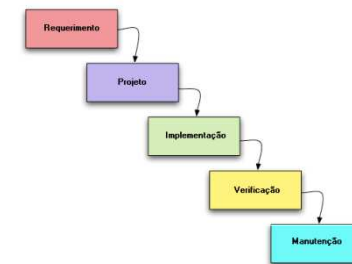
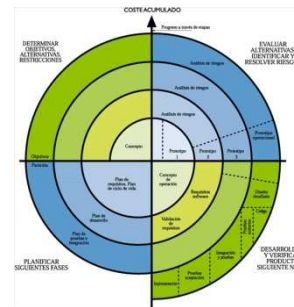
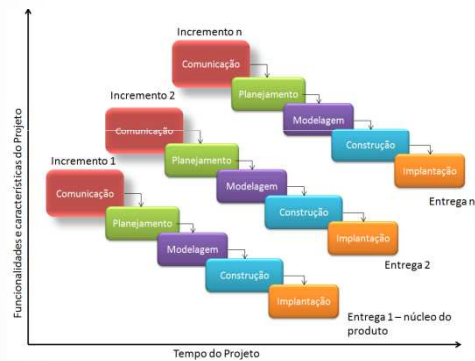
Contatos:

- luciocamilo@gmail.com
- www.itnerante.com.br/profile/luciocamilo

Conteúdo do Módulo

- Fundamentos de Engenharia de Software
- Conceitos
- Modelos de Ciclo de Vida
- Metodologias Ágeis

Conceitos e Ciclo de Vida



Questão 01 – EPE - 2012

Uma das críticas feitas ao modelo do ciclo de vida do desenvolvimento de software em cascata refere-se a

- A) exigência de conhecimento de avaliação e gerenciamento de risco para evitar grandes surpresas no projeto.
- B) comprometimentos na qualidade e nas possibilidades de manutenção a longo prazo, parecendo um protótipo.
- C) pouca flexibilidade para mudanças futuras, exigindo compromissos nas fases iniciais do projeto.
- D) pouca visibilidade das etapas do processo, tornando cara a documentação de todas as versões dos sistemas.
- E) exigências de velocidade as quais levam o engenheiro de software a utilizar linguagens, algoritmos ou ferramentas ineficientes ao longo de todo o projeto.

Modelo Cascata

- Clássico, Linear, Sequencial;
- Requisitos de um problema devem ser estáveis e bem compreendidos;
- Trabalho flui da Comunicação ao Emprego de forma linear;
- Abordagem sequencial e sistemática

Pressman:



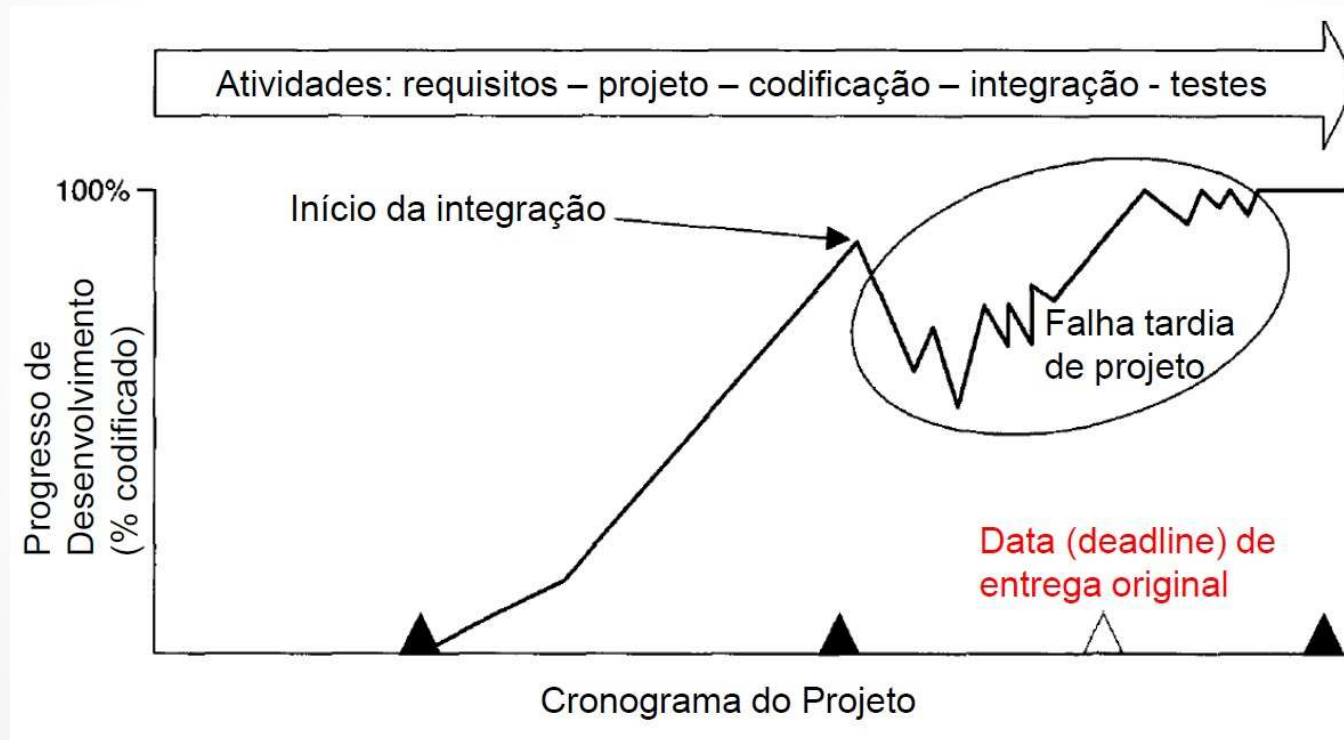
Sommerville:



Modelo Cascata

- Vantagens:
 - Simples e de fácil gerência
 - Adequado para sistemas de baixa complexidade
- Problemas:
 - Projetos reais, geralmente não seguem um fluxo sequencial
 - Geralmente o cliente não sabe definir todos os requisitos no início
 - Uma versão funcional somente estará disponível ao final do projeto
 - Erros são detectados no final
 - Sistema é entregue todo de uma vez
 - Uma etapa depende da outra (Equipes ociosas)
 - Atrasa a redução de riscos

Modelo Cascata



Questão 01 – EPE - 2012

Uma das críticas feitas ao modelo do ciclo de vida do desenvolvimento de software em cascata refere-se a

- A) exigência de conhecimento de avaliação e gerenciamento de risco para evitar grandes surpresas no projeto.
- B) comprometimentos na qualidade e nas possibilidades de manutenção a longo prazo, parecendo um protótipo.
- C) pouca flexibilidade para mudanças futuras, exigindo compromissos nas fases iniciais do projeto.
- D) pouca visibilidade das etapas do processo, tornando cara a documentação de todas as versões dos sistemas.
- E) exigências de velocidade as quais levam o engenheiro de software a utilizar linguagens, algoritmos ou ferramentas ineficientes ao longo de todo o projeto.

Questão 01 – EPE - 2012

Uma das críticas feitas ao modelo do ciclo de vida do desenvolvimento de software em cascata refere-se a

A) exigência de conhecimento de avaliação e gerenciamento de risco para evitar grandes surpresas no projeto.

B) comprometimentos na qualidade e nas possibilidades de manutenção a longo prazo, parecendo um protótipo.

➔ C) pouca flexibilidade para mudanças futuras, exigindo compromissos nas fases iniciais do projeto.

D) pouca visibilidade das etapas do processo, tornando cara a documentação de todas as versões dos sistemas.

E) exigências de velocidade as quais levam o engenheiro de software a utilizar linguagens, algoritmos ou ferramentas ineficientes ao longo de todo o projeto.

Questão 02 – TRANSPETRO - 2012

Na engenharia de software, existem diversos modelos de desenvolvimento de software, e, dentre eles, o modelo em cascata, o qual, no contexto do desenvolvimento de sistemas de software complexos, recomenda

- A) distribuir a elicitação dos requisitos desde o início até o fim do desenvolvimento.
- B) dividir o desenvolvimento do produto de software em fases lineares e sequenciais.
- C) enfatizar a avaliação e mitigação de riscos durante o desenvolvimento.
- D) realizar entregas incrementais do produto de software ao longo do desenvolvimento.
- E) usar prototipagem rápida para estimular o envolvimento do usuário no desenvolvimento.

Questão 02 – TRANSPETRO - 2012

Na engenharia de software, existem diversos modelos de desenvolvimento de software, e, dentre eles, o modelo em cascata, o qual, no contexto do desenvolvimento de sistemas de software complexos, recomenda

A) distribuir a elicitação dos requisitos desde o início até o fim do desenvolvimento.

➡ B) dividir o desenvolvimento do produto de software em fases lineares e sequenciais.

C) enfatizar a avaliação e mitigação de riscos durante o desenvolvimento.

D) realizar entregas incrementais do produto de software ao longo do desenvolvimento.

E) usar prototipagem rápida para estimular o envolvimento do usuário no desenvolvimento.

Questão 03 – FUNASA - 2009

À luz da Engenharia de Software, o ciclo de vida clássico, também chamado de modelo sequencial linear ou modelo em cascata, é um paradigma aplicável ao desenvolvimento de sistemas de informações que

A) enfatiza a análise de riscos, que é feita no início do projeto e revisada a cada fase, incluindo um plano de ataque e ações de mitigação dos riscos, aumentando as chances de sucesso do projeto.

B) prevê uma sequência (ou cascata) de entregas de versões do sistema ao longo do desenvolvimento do mesmo, o que proporciona uma avaliação de progresso baseada em código funcionando, em vez de quantidade de documentação gerada.

C) exige que todos os requisitos sejam conhecidos e corretamente especificados no início do ciclo de vida, dificultando a acomodação de mudanças que surjam nas fases posteriores.

D) foi sempre pouco utilizado na prática, pois se apoia em analogias com práticas de engenharia convencional que não se aplicam bem ao desenvolvimento de sistemas de informação.

E) utiliza a estratégia dividir para conquistar (divide to conquer), prevendo que cada fase do ciclo de vida seja desdobrada em um miniciclo de vida sequencial completo, formando uma cascata de ciclos de vida até o limite adequado para lidar com a complexidade do sistema.

Questão 03 – FUNASA - 2009

À luz da Engenharia de Software, o ciclo de vida clássico, também chamado de modelo sequencial linear ou modelo em cascata, é um paradigma aplicável ao desenvolvimento de sistemas de informações que

A) enfatiza a análise de riscos, que é feita no início do projeto e revisada a cada fase, incluindo um plano de ataque e ações de mitigação dos riscos, aumentando as chances de sucesso do projeto.

B) prevê uma sequência (ou cascata) de entregas de versões do sistema ao longo do desenvolvimento do mesmo, o que proporciona uma avaliação de progresso baseada em código funcionando, em vez de quantidade de documentação gerada.

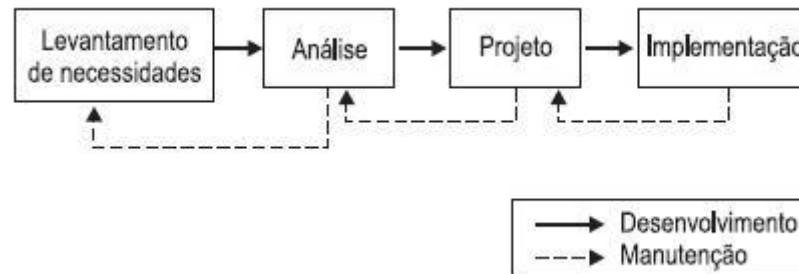
➡ C) exige que todos os requisitos sejam conhecidos e corretamente especificados no início do ciclo de vida, dificultando a acomodação de mudanças que surjam nas fases posteriores.

D) foi sempre pouco utilizado na prática, pois se apoia em analogias com práticas de engenharia convencional que não se aplicam bem ao desenvolvimento de sistemas de informação.

E) utiliza a estratégia dividir para conquistar (divide to conquer), prevendo que cada fase do ciclo de vida seja desdobrada em um miniciclo de vida sequencial completo, formando uma cascata de ciclos de vida até o limite adequado para lidar com a complexidade do sistema.

Questão 04 – ELETROBRÁS - 2010

A figura abaixo representa, simplificada, as fases do Modelo de Ciclo de Vida Cascata.

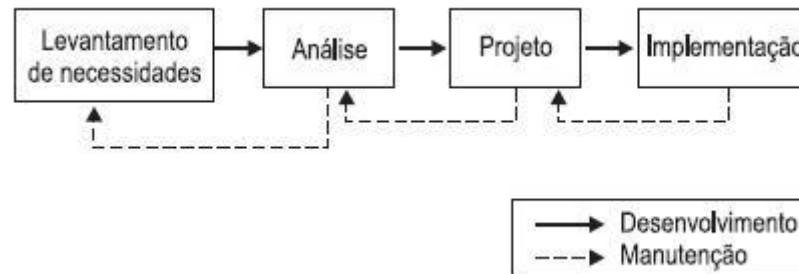


Dentre as diversas características desse modelo, afirma-se que

- a) existe um protótipo do sistema, ao final de cada fase, cada vez mais completo, que permite ao cliente avaliar o produto.
- b) nenhuma fase é terminada até que a sua documentação tenha sido completada e seus produtos aprovados pelo grupo de garantia da qualidade.
- c) o custo de modificação do sistema é praticamente o mesmo, independente da fase em que o projeto esteja.
- d) as fases podem se sobrepor, para acelerar o projeto.
- e) datagramas de fluxo de dados ou diagramas UML são utilizados como técnicas gráficas para se comunicar com seus clientes.

Questão 04 – ELETROBRÁS - 2010

A figura abaixo representa, simplificada, as fases do Modelo de Ciclo de Vida Cascata.



Dentre as diversas características desse modelo, afirma-se que

- a) existe um protótipo do sistema, ao final de cada fase, cada vez mais completo, que permite ao cliente avaliar o produto.
- ➔ b) nenhuma fase é terminada até que a sua documentação tenha sido completada e seus produtos aprovados pelo grupo de garantia da qualidade.
- c) o custo de modificação do sistema é praticamente o mesmo, independente da fase em que o projeto esteja.
- d) as fases podem se sobrepor, para acelerar o projeto.
- e) datagramas de fluxo de dados ou diagramas UML são utilizados como técnicas gráficas para se comunicar com seus clientes.

Questão 05 – PETROBRÁS - 2010

O modelo de ciclo de vida em cascata

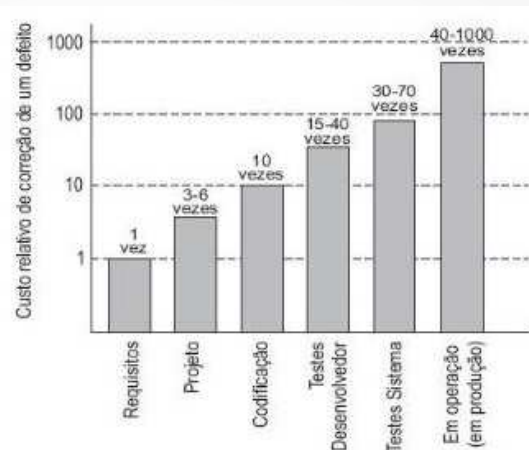
- a) enfatiza a realização sequencial das atividades do desenvolvimento de um produto de software.
- b) enfatiza a comunicação estreita com o cliente durante o desenvolvimento do produto de software.
- c) envolve a ideia principal de criar um protótipo executável e, por meio de transformações sucessivas, chegar ao sistema completamente implementado.
- d) envolve a análise dos riscos envolvidos no desenvolvimento dos requisitos identificados para produto de software.
- e) recomenda a geração de versões incompletas do sistema, que podem ser passadas para o usuário final, o que permite a retroalimentação do processo de desenvolvimento.

Questão 05 – PETROBRÁS - 2010

O modelo de ciclo de vida em cascata

- ➡ a) enfatiza a realização sequencial das atividades do desenvolvimento de um produto de software.
- b) enfatiza a comunicação estreita com o cliente durante o desenvolvimento do produto de software.
- c) envolve a ideia principal de criar um protótipo executável e, por meio de transformações sucessivas, chegar ao sistema completamente implementado.
- d) envolve a análise dos riscos envolvidos no desenvolvimento dos requisitos identificados para produto de software.
- e) recomenda a geração de versões incompletas do sistema, que podem ser passadas para o usuário final, o que permite a retroalimentação do processo de desenvolvimento.

Questão 06 – TJ/RO - 2008



O gráfico acima, adaptado do livro Engenharia de Software, de Roger Pressman, ilustra o custo relativo da correção de um defeito nas diversas fases de um projeto de software, baseado em dados colhidos por Boehm e outros estudiosos. Embora não seja explicitamente informado, os dados se basearam, principalmente, em projetos que utilizaram o modelo de desenvolvimento em cascata. A esse respeito, analise as afirmativas a seguir.

I – O gráfico pode ser utilizado como um argumento a favor do uso de processos de desenvolvimento iterativos.

II – O gráfico deixa claro que o maior investimento na prevenção de defeitos deve acontecer nas fases finais do projeto, preferencialmente depois que o software estiver em uso pelos clientes.

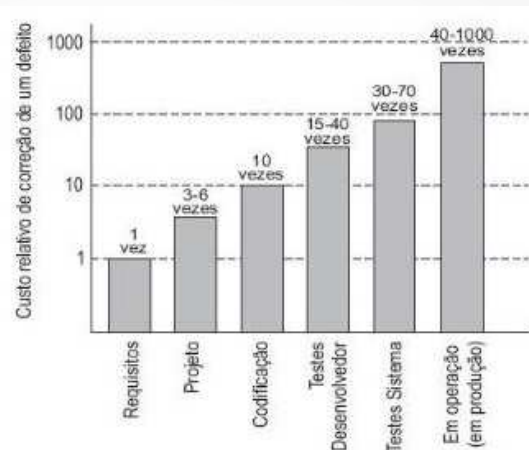
III – O gráfico não é conclusivo a respeito da importância do gerenciamento dos requisitos de um projeto, o que é consistente com a abordagem de muitos processos de desenvolvimento de software atuais, que minimizam este esforço e enfatizam a codificação e os testes unitários de código.

IV – O gráfico sustenta os argumentos de que a qualidade deve ser incorporada ao processo através de técnicas e ações efetivas de detecção, prevenção e controle, garantindo que todas as atividades do projeto resultem em produtos ou subprodutos de qualidade, ao invés de ser uma preocupação secundária ou limitada a um grupo de profissionais de controle de qualidade.

São corretas APENAS as afirmativas

- a) I e II
- b) I e IV
- c) II e III
- d) II e IV
- e) III e IV

Questão 06 – TJ/RO – 2008



O gráfico acima, adaptado do livro Engenharia de Software, de Roger Pressman, ilustra o custo relativo da correção de um defeito nas diversas fases de um projeto de software, baseado em dados colhidos por Boehm e outros estudiosos. Embora não seja explicitamente informado, os dados se basearam, principalmente, em projetos que utilizaram o modelo de desenvolvimento em cascata. A esse respeito, analise as afirmativas a seguir.

I – O gráfico pode ser utilizado como um argumento a favor do uso de processos de desenvolvimento iterativos.

II – O gráfico deixa claro que o maior investimento na prevenção de defeitos deve acontecer nas fases finais do projeto, preferencialmente depois que o software estiver em uso pelos clientes.

III – O gráfico não é conclusivo a respeito da importância do gerenciamento dos requisitos de um projeto, o que é consistente com a abordagem de muitos processos de desenvolvimento de software atuais, que minimizam este esforço e enfatizam a codificação e os testes unitários de código.

IV – O gráfico sustenta os argumentos de que a qualidade deve ser incorporada ao processo através de técnicas e ações efetivas de detecção, prevenção e controle, garantindo que todas as atividades do projeto resultem em produtos ou subprodutos de qualidade, ao invés de ser uma preocupação secundária ou limitada a um grupo de profissionais de controle de qualidade.

São corretas APENAS as afirmativas

a) I e II

➔ b) I e IV

c) II e III

d) II e IV

e) III e IV

Questão 07 – BNDES - 2012

No ciclo de vida em cascata, o custo de correção é menor na fase de

- a) testes.
- b) transição.
- c) implementação.
- d) requisitos.
- e) manutenção.

Questão 07 – BNDES - 2012

No ciclo de vida em cascata, o custo de correção é menor na fase de

- a) testes.
- b) transição.
- c) implementação.
- ➡ d) requisitos.
- e) manutenção.

Questão 08 – PETROBRÁS – PROCESSOS - 2012

No âmbito dos processos de desenvolvimento de software, entre as vantagens do desenvolvimento incremental sobre o modelo em cascata, inclui-se a(o):

- (A) consistência com outros modelos de processo de engenharia, no qual documentação e modelos são produzidos em cada fase.
- (B) redução da interação com o usuário, colocando o foco na programação e entregando mais software útil.
- (C) criação de um modelo matemático do sistema, garantindo um programa consistente com as especificações por meio de transformações corretas.
- (D) diminuição do custo de mudança nos requisitos do usuário, pela diminuição da quantidade de análise e documentação que deve ser refeita.
- (E) congelamento de partes do desenvolvimento para continuação em estágios mais avançados, o que leva a um software bem estruturado e sem truques de programação.

Questão 08 – PETROBRÁS – PROCESSOS - 2012

No âmbito dos processos de desenvolvimento de software, entre as vantagens do desenvolvimento incremental sobre o modelo em cascata, inclui-se a(o):

(A) consistência com outros modelos de processo de engenharia, no qual documentação e modelos são produzidos em cada fase.

(B) redução da interação com o usuário, colocando o foco na programação e entregando mais software útil.

(C) criação de um modelo matemático do sistema, garantindo um programa consistente com as especificações por meio de transformações corretas.

➡ (D) diminuição do custo de mudança nos requisitos do usuário, pela diminuição da quantidade de análise e documentação que deve ser refeita.

(E) congelamento de partes do desenvolvimento para continuação em estágios mais avançados, o que leva a um software bem estruturado e sem truques de programação.

Questão 09 – BACEN - 2010

Determinada superintendência de um órgão público solicita o desenvolvimento de um sistema de informação que deve recolher informações de pessoas físicas de todo o Brasil, pela Internet. O superintendente, que abandonará a gestão em menos de 1 mês, exige que o sistema completo esteja no ar em 2 semanas e aponta que erros podem ser reparados após a implantação do sistema. Com base nesse relato, afirma-se que

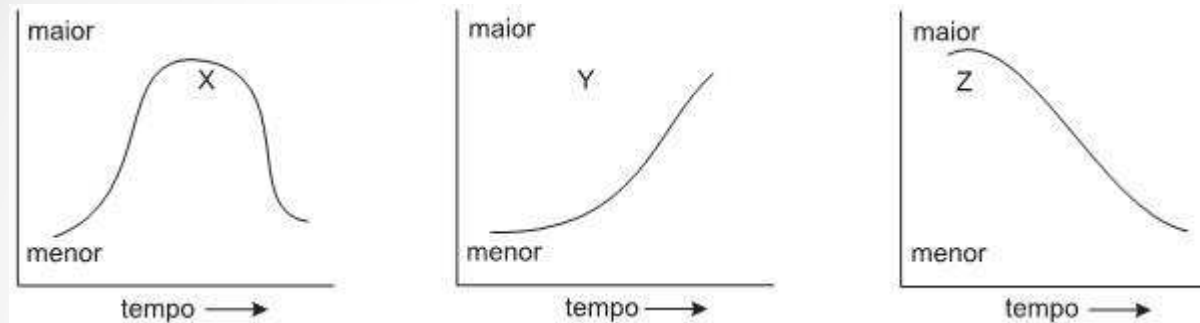
- a) a correção de erros deve ser feita, preferencialmente, na fase de manutenção, já que é mais eficiente corrigir o produto depois de construído.
- b) a construção de sistemas WEB, nesse prazo, é inviável, independente do tamanho do projeto.
- c) esse tipo de solicitação é muito improvável na atualidade, uma vez que há expectativas quanto à redução de custos, mas não de prazo.
- d) o custo esperado de reparo na fase de manutenção é até três vezes superior ao custo de reparo na fase de requisitos.
- e) os erros mais caros são aqueles cometidos na análise de requisitos e descobertos pelo usuário.

Questão 09 – BACEN - 2010

Determinada superintendência de um órgão público solicita o desenvolvimento de um sistema de informação que deve recolher informações de pessoas físicas de todo o Brasil, pela Internet. O superintendente, que abandonará a gestão em menos de 1 mês, exige que o sistema completo esteja no ar em 2 semanas e aponta que erros podem ser reparados após a implantação do sistema. Com base nesse relato, afirma-se que

- a) a correção de erros deve ser feita, preferencialmente, na fase de manutenção, já que é mais eficiente corrigir o produto depois de construído.
- b) a construção de sistemas WEB, nesse prazo, é inviável, independente do tamanho do projeto.
- c) esse tipo de solicitação é muito improvável na atualidade, uma vez que há expectativas quanto à redução de custos, mas não de prazo.
- d) o custo esperado de reparo na fase de manutenção é até três vezes superior ao custo de reparo na fase de requisitos.
- ➡ e) os erros mais caros são aqueles cometidos na análise de requisitos e descobertos pelo usuário.

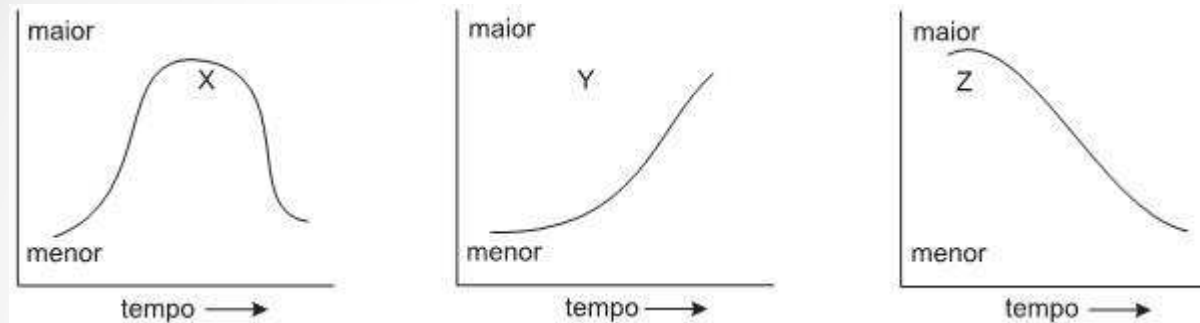
Questão 10 – PETROBRÁS (INFRA) - 2012



O tamanho da equipe de desenvolvimento, os riscos e o custo das mudanças nos requisitos, variam ao longo da realização de um projeto. Essas variações podem ser representadas, geralmente, por gráficos respectivamente semelhantes às curvas

- a) X, Y e Z
- b) X, Z e Y
- c) Y, Z e X
- d) Y, X e Z
- e) Z, X e Y

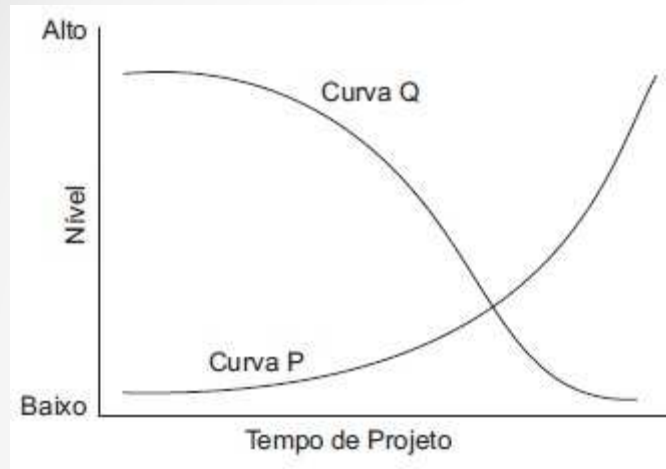
Questão 10 – PETROBRÁS (INFRA) - 2012



O tamanho da equipe de desenvolvimento, os riscos e o custo das mudanças nos requisitos, variam ao longo da realização de um projeto. Essas variações podem ser representadas, geralmente, por gráficos respectivamente semelhantes às curvas

- a) X, Y e Z
- ➡ b) X, Z e Y
- c) Y, Z e X
- d) Y, X e Z
- e) Z, X e Y

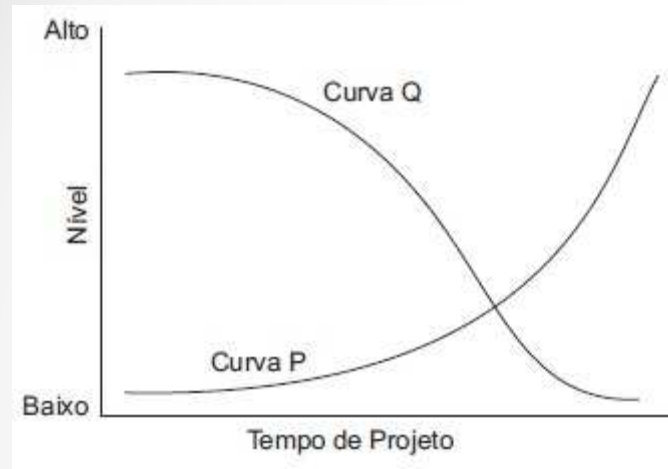
Questão 11 – PETROBRÁS – PROCESSOS - 2012



Considerando o ciclo de vida de um projeto, as curvas Q e P do gráfico apresentado podem representar, respectivamente, as tendências do(a)

- (A) custo de mudanças e do nível de risco
- (B) custo de mudanças e do nível de incerteza
- (C) nível de incerteza e do nível de risco
- (D) nível de incerteza e da influência das partes interessadas
- (E) influência das partes interessadas e do custo de mudanças

Questão 11 – PETROBRÁS – PROCESSOS - 2012



Considerando o ciclo de vida de um projeto, as curvas Q e P do gráfico apresentado podem representar, respectivamente, as tendências do(a)

- (A) custo de mudanças e do nível de risco
- (B) custo de mudanças e do nível de incerteza
- (C) nível de incerteza e do nível de risco
- (D) nível de incerteza e da influência das partes interessadas

➡ (E) influência das partes interessadas e do custo de mudanças

Questão 12 – CHESF - 2012

Um analista desenvolve um software e identifica que os seus requisitos iniciais estão razoavelmente bem definidos, mas o escopo geral do desenvolvimento não permite um processo puramente linear. Ele sabe que precisa, em curtíssimo prazo, prover um conjunto limitado de funcionalidades do software para os usuários, que serão refinadas e expandidas em versões futuras.

Qual o modelo de ciclo de vida de desenvolvimento de software mais adequado a esse caso?

- a) Cascata
- b) Espiral
- c) Formal
- d) Incremental
- e) Prototipação

Questão 12 – CHESF - 2012

Um analista desenvolve um software e identifica que os seus requisitos iniciais estão razoavelmente bem definidos, mas o escopo geral do desenvolvimento não permite um processo puramente linear. Ele sabe que precisa, em curtíssimo prazo, prover um conjunto limitado de funcionalidades do software para os usuários, que serão refinadas e expandidas em versões futuras.

Qual o modelo de ciclo de vida de desenvolvimento de software mais adequado a esse caso?

- a) Cascata
- b) Espiral
- c) Formal
- ➡ d) Incremental
- e) Prototipação

Questão 13 – BNDES - 2011

Um projeto estruturado para desenvolvimento de sistemas é dividido em diversas fases e, entre elas, a que é responsável pelos testes de aceitação de um sistema que, passando por ela, estará pronto para ser implantado e posto em produção.

A partir dessa fase, qualquer modificação que aconteça nesse sistema pertencerá à fase denominada

- a) manutenção
- b) implementação
- c) análise
- d) reconhecimento do problema
- e) estudo da viabilidade

Questão 13 – BNDES - 2011

Um projeto estruturado para desenvolvimento de sistemas é dividido em diversas fases e, entre elas, a que é responsável pelos testes de aceitação de um sistema que, passando por ela, estará pronto para ser implantado e posto em produção.

A partir dessa fase, qualquer modificação que aconteça nesse sistema pertencerá à fase denominada

- ➡ a) manutenção
- b) implementação
- c) análise
- d) reconhecimento do problema
- e) estudo da viabilidade

Questão 14 – BNDES - 2012

De um modo geral, o processo de desenvolvimento de um software pode ser organizado partindo de três fases importantes, que são as de definição, de desenvolvimento e de manutenção. Na fase de manutenção, dentre outras atividades, são

- (A) levantados os requisitos dos usuários para a programação das diversas fases do projeto, inclusive as operacionais e as preditivas.
- (B) efetuados os testes de funcionalidade do software, revistos os objetivos para os quais ele foi desenvolvido e redefinidas as funções em desacordo com esses objetivos.
- (C) incluídas novas funções requeridas pelo cliente e feitas adaptações por modificações de hardware.
- (D) reavaliadas as bases operacionais, nas quais o software está sendo executado, e prototipados os novos requisitos de hardware.
- (E) propostas pelos usuários as novas rotinas que eles definiram como fundamentais para a continuidade do produto implantado.

Questão 14 – BNDES - 2012

De um modo geral, o processo de desenvolvimento de um software pode ser organizado partindo de três fases importantes, que são as de definição, de desenvolvimento e de manutenção. Na fase de manutenção, dentre outras atividades, são

- (A) levantados os requisitos dos usuários para a programação das diversas fases do projeto, inclusive as operacionais e as preditivas.
- (B) efetuados os testes de funcionalidade do software, revistos os objetivos para os quais ele foi desenvolvido e redefinidas as funções em desacordo com esses objetivos.
- ➡ (C) incluídas novas funções requeridas pelo cliente e feitas adaptações por modificações de hardware.
- (D) reavaliadas as bases operacionais, nas quais o software está sendo executado, e prototipados os novos requisitos de hardware.
- (E) propostas pelos usuários as novas rotinas que eles definiram como fundamentais para a continuidade do produto implantado.

Questão 15 – TRANSPETRO - 2011

Na Engenharia de Software, há diversos modelos de ciclo de vida, definidos com variados níveis de formalidade. O modelo

- a) cascata (ou clássico) é adequado para controlar riscos e requisitos voláteis durante o desenvolvimento do sistema.
- b) codificação e correção (code and fix) é adequado para alcançar um bom nível de manutenibilidade do sistema.
- c) prototipagem descartável é adequado para descartar a fase de levantamento de requisitos do sistema a ser desenvolvido.
- d) prototipagem evolutiva entrega uma versão inicial do sistema, que considera requisitos já definidos com o cliente.
- e) espiral é inadequado quando são necessários o uso de protótipos durante a validação do sistema e o reúso de software.

Questão 15 – TRANSPETRO - 2011

Na Engenharia de Software, há diversos modelos de ciclo de vida, definidos com variados níveis de formalidade. O modelo

- a) cascata (ou clássico) é adequado para controlar riscos e requisitos voláteis durante o desenvolvimento do sistema.
- b) codificação e correção (code and fix) é adequado para alcançar um bom nível de manutenibilidade do sistema.
- c) prototipagem descartável é adequado para descartar a fase de levantamento de requisitos do sistema a ser desenvolvido.
- ➡ d) prototipagem evolutiva entrega uma versão inicial do sistema, que considera requisitos já definidos com o cliente.
- e) espiral é inadequado quando são necessários o uso de protótipos durante a validação do sistema e o reúso de software.

Questão 16 – ELETROBRÁS - 2010

O termo Modelo de Ciclo de Vida é utilizado para descrever um grupo de atividades e a forma como elas se relacionam.

Considerando o Modelo de Ciclo de Vida de Sistemas por Prototipagem Evolucionária, afirma-se que

- a) os clientes não têm acesso a uma visualização dos progressos do desenvolvimento.
- b) é possível determinar com exatidão o tempo que o projeto irá demorar.
- c) não deve ser utilizado quando os requisitos mudam rapidamente e o cliente está relutante em aceitar um conjunto de requisitos.
- d) não há uma forma de saber de antemão o número de iterações que serão necessárias.
- e) apenas a fase final gera um produto que não é um documento.

Questão 16 – ELETROBRÁS - 2010

O termo Modelo de Ciclo de Vida é utilizado para descrever um grupo de atividades e a forma como elas se relacionam.

Considerando o Modelo de Ciclo de Vida de Sistemas por Prototipagem Evolucionária, afirma-se que

- a) os clientes não têm acesso a uma visualização dos progressos do desenvolvimento.
- b) é possível determinar com exatidão o tempo que o projeto irá demorar.
- c) não deve ser utilizado quando os requisitos mudam rapidamente e o cliente está relutante em aceitar um conjunto de requisitos.
- ➡ d) não há uma forma de saber de antemão o número de iterações que serão necessárias.
- e) apenas a fase final gera um produto que não é um documento.

Questão 17 – EPE - 2007

O ciclo de vida clássico da engenharia de software tem sofrido críticas dos especialistas. Uma delas é o fato de o cliente ter de declarar todos os requisitos explicitamente na primeira parte do projeto, gerando insegurança. Uma técnica utilizada para minimizar esse problema é conhecida como:

- a) análise por ponto de função.
- b) casos de teste.
- c) linha básica ou baseline.
- d) projeto de software.
- e) prototipação.

Questão 17 – EPE - 2007

O ciclo de vida clássico da engenharia de software tem sofrido críticas dos especialistas. Uma delas é o fato de o cliente ter de declarar todos os requisitos explicitamente na primeira parte do projeto, gerando insegurança. Uma técnica utilizada para minimizar esse problema é conhecida como:

- a) análise por ponto de função.
- b) casos de teste.
- c) linha básica ou baseline.
- d) projeto de software.
- ➡ e) prototipação.

Questão 18 – PETROBRÁS - 2010

Muito utilizada pelos engenheiros de software, a prototipação consiste na(o)

- a) análise dos custos e dos riscos de desenvolvimento, com base nos requisitos iniciais.
- b) coleta inicial dos requisitos e no planejamento do projeto, com base no grupo de programadores envolvidos.
- c) decisão de prosseguir ou não na direção de um sistema concluído, com base nos comentários do cliente.
- d) procedimento estruturado que avalia os requisitos finais de um sistema em fase de testes.
- e) processo que capacita o desenvolvedor a criar um modelo do software que será implementado.

Questão 18 – PETROBRÁS - 2010

Muito utilizada pelos engenheiros de software, a prototipação consiste na(o)

- a) análise dos custos e dos riscos de desenvolvimento, com base nos requisitos iniciais.
- b) coleta inicial dos requisitos e no planejamento do projeto, com base no grupo de programadores envolvidos.
- c) decisão de prosseguir ou não na direção de um sistema concluído, com base nos comentários do cliente.
- d) procedimento estruturado que avalia os requisitos finais de um sistema em fase de testes.
- ➡ e) processo que capacita o desenvolvedor a criar um modelo do software que será implementado.

Questão 19 – PETROBRÁS (Técnico) - 2012

Dentre os modelos de Processo de Software, um dos mais conhecidos é o linear sequencial, que pode ser descrito como sendo composto de 4 fases sequenciais:

Análise, Projeto, Codificação e Testes.

A fase de Projeto caracteriza-se por ser onde o(s)

- a) comportamento necessário do software é entendido.
- b) domínio de informação é entendido.
- c) código pode ser gerado automaticamente.
- d) requisitos de sistema e do software são levantados.
- e) requisitos são traduzidos em uma representação do software.

Questão 19 – PETROBRÁS (Técnico) - 2012

Dentre os modelos de Processo de Software, um dos mais conhecidos é o linear sequencial, que pode ser descrito como sendo composto de 4 fases sequenciais:

Análise, Projeto, Codificação e Testes.

A fase de Projeto caracteriza-se por ser onde o(s)

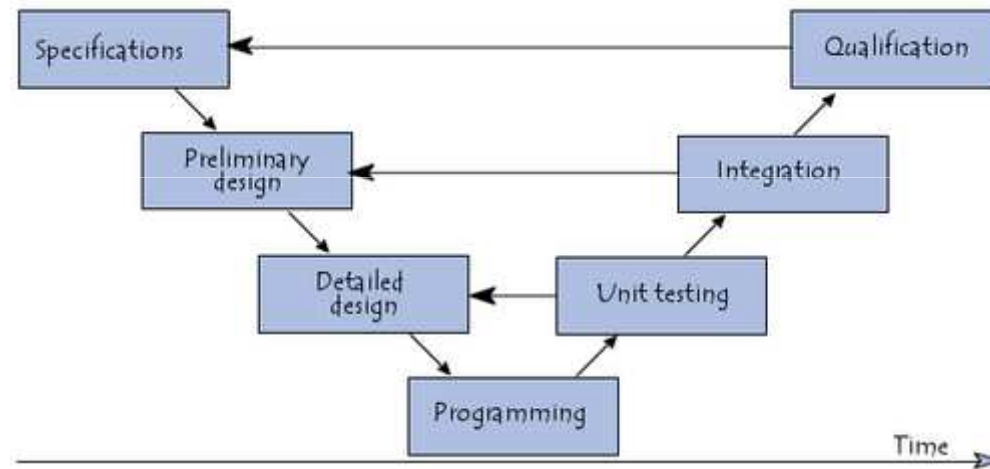
- a) comportamento necessário do software é entendido.
- b) domínio de informação é entendido.
- c) código pode ser gerado automaticamente.
- d) requisitos de sistema e do software são levantados.
- ➡ e) requisitos são traduzidos em uma representação do software.

Questão 20 – FINEP - 2011

O modelo de processo de desenvolvimento de software que combina as atividades de desenvolvimento com o gerenciamento de risco é o Modelo

- a) Transformacional
- b) de Especificação Operacional
- c) em V
- d) em Cascata
- e) em Espiral

- Modelo em V



Questão 20 – FINEP - 2011

O modelo de processo de desenvolvimento de software que combina as atividades de desenvolvimento com o gerenciamento de risco é o Modelo

- a) Transformacional
- b) de Especificação Operacional
- c) em V
- d) em Cascata
- ➡ e) em Espiral

Questão 21 – ELETROBRÁS - 2010

Uma fábrica de software utiliza um ciclo de vida de desenvolvimento de sistemas que contempla um conjunto sequencial de ações de desenvolvimento, desde o diagnóstico do problema até os testes necessários à implementação. Além disso, nada está terminado até que todas as fases estejam completas. Esse ciclo de vida é conhecido como

- a) XP.
- b) Cascata.
- c) SCRUM.
- d) Continuum.
- e) Espiral.

Questão 21 – ELETROBRÁS - 2010

Uma fábrica de software utiliza um ciclo de vida de desenvolvimento de sistemas que contempla um conjunto sequencial de ações de desenvolvimento, desde o diagnóstico do problema até os testes necessários à implementação. Além disso, nada está terminado até que todas as fases estejam completas. Esse ciclo de vida é conhecido como

- a) XP.
- ➡ b) Cascata.
- c) SCRUM.
- d) Continuum.
- e) Espiral.

Questão 22 – BR - 2010

Um sistema de informação, direcionado a usuários internos de uma empresa, está sendo desenvolvido segundo o ciclo de vida em cascata. O gerente desse projeto foi informado que, no momento, estão ocorrendo verificações formais sobre a integridade do sistema. Considerando-se que há também a checagem de conformidade das saídas desse sistema, conclui-se que essa é a etapa de

- (A) testes.
- (B) codificação.
- (C) requisitos.
- (D) segurança.
- (E) implantação.

Questão 22 – BR - 2010

Um sistema de informação, direcionado a usuários internos de uma empresa, está sendo desenvolvido segundo o ciclo de vida em cascata. O gerente desse projeto foi informado que, no momento, estão ocorrendo verificações formais sobre a integridade do sistema. Considerando-se que há também a checagem de conformidade das saídas desse sistema, conclui-se que essa é a etapa de

- ➡ (A) testes.
(B) codificação.
(C) requisitos.
(D) segurança.
(E) implantação.

Questão 23 – DECEA - 2009

A fase que deve acontecer primeiro no desenvolvimento de sistemas de uma empresa que utiliza o modelo de ciclo de vida em cascata é a de

- A) testes.
- B) análise e projeto.
- C) especificação de requisitos.
- D) implantação.
- E) implementação.

Questão 23 – DECEA - 2009

A fase que deve acontecer primeiro no desenvolvimento de sistemas de uma empresa que utiliza o modelo de ciclo de vida em cascata é a de

A) testes.

B) análise e projeto.

➡ C) especificação de requisitos.

D) implantação.

E) implementação.

Questão 24 – ELETROBRÁS - 2010

O gerenciamento de grande quantidade de informação na construção de sistemas pode ser contornada usando-se a técnica de refinamentos sucessivos, utilizada no modelo de Ciclo de Vida Iterativo e Incremental. A construção de sistemas, com base nesse modelo de ciclo de vida,

- a) é dividida em, no máximo, 7 incrementos, com 7 iterações cada, devido à restrição da Lei de Miller.
- b) tem seus incrementos trabalhados simultaneamente, acelerando o desenvolvimento do sistema.
- c) contém atividades que podem exigir trabalho, em maior ou menor grau, em todos os incrementos planejados.
- d) define que as atividades de testes sejam realizadas no último incremento, que é planejado exclusivamente para tal propósito.
- e) deve ter a mesma quantidade de iterações em todos os incrementos planejados.

Questão 24 – ELETROBRÁS - 2010

O gerenciamento de grande quantidade de informação na construção de sistemas pode ser contornada usando-se a técnica de refinamentos sucessivos, utilizada no modelo de Ciclo de Vida Iterativo e Incremental. A construção de sistemas, com base nesse modelo de ciclo de vida,

- a) é dividida em, no máximo, 7 incrementos, com 7 iterações cada, devido à restrição da Lei de Miller.
- b) tem seus incrementos trabalhados simultaneamente, acelerando o desenvolvimento do sistema.
- ➡ c) contém atividades que podem exigir trabalho, em maior ou menor grau, em todos os incrementos planejados.
- d) define que as atividades de testes sejam realizadas no último incremento, que é planejado exclusivamente para tal propósito.
- e) deve ter a mesma quantidade de iterações em todos os incrementos planejados.

Questão 25 – PETROBRÁS - 2008

Uma diferença entre o modelo incremental e o modelo evolucionário por prototipagem é que somente o modelo incremental

- A) é iterativo.
- B) é recomendado quando os requisitos do sistemas estão totalmente definidos a priori.
- C) pode gerar um incremento com objetivo de minimizar o risco de operação do sistema final.
- D) objetiva entregar um sistema que seja operacional em cada incremento.
- E) projeta testes antes da implementação do sistema.

Questão 25 – PETROBRÁS - 2008

Uma diferença entre o modelo incremental e o modelo evolucionário por prototipagem é que somente o modelo incremental

- A) é iterativo.
- B) é recomendado quando os requisitos do sistemas estão totalmente definidos a priori.
- C) pode gerar um incremento com objetivo de minimizar o risco de operação do sistema final.
- ➡ D) objetiva entregar um sistema que seja operacional em cada incremento.
- E) projeta testes antes da implementação do sistema.

Questão 26 – IBGE - 2010

Os processos de desenvolvimento de software utilizam, muitas vezes, procedimentos estatísticos para, por exemplo, apoiar a tomada de decisão. Dentro desse contexto, o Diagrama de Pareto é baseado na clássica regra de que

- a) 20% das ocorrências causam 80% dos problemas.
- b) 60% das amostras de um processo normal encontram-se nos limites do desvio padrão.
- c) pontos fora dos limites de um desvio padrão revelam a ocorrência de problemas aleatórios.
- d) três pontos consecutivos abaixo da média indicam um processo em melhoria contínua.
- e) um índice de erro acima dos cinco sigmas indica um processo que alcançou a qualidade.

Questão 26 – IBGE - 2010

Os processos de desenvolvimento de software utilizam, muitas vezes, procedimentos estatísticos para, por exemplo, apoiar a tomada de decisão. Dentro desse contexto, o Diagrama de Pareto é baseado na clássica regra de que

- ➡ a) 20% das ocorrências causam 80% dos problemas.
- b) 60% das amostras de um processo normal encontram-se nos limites do desvio padrão.
- c) pontos fora dos limites de um desvio padrão revelam a ocorrência de problemas aleatórios.
- d) três pontos consecutivos abaixo da média indicam um processo em melhoria contínua.
- e) um índice de erro acima dos cinco sigmas indica um processo que alcançou a qualidade.

Questão 27 – PETROBRÁS - 2008

São estratégias possíveis de serem adotadas por uma organização para evolução de seus sistemas legados as apresentadas a seguir, **EXCETO**:

- a) descartar o sistema completamente.
- b) substituir todo ou parte do sistema por um novo sistema.
- c) deixar o sistema sem alterações e continuar com a manutenção regular.
- d) fazer reengenharia do sistema para aprimorar sua facilidade de manutenção.
- e) realizar engenharia reversa do código do sistema com o objetivo de aumentar sua complexidade ciclomática e torná-lo crítico à empresa.

Questão 27 – PETROBRÁS - 2008

São estratégias possíveis de serem adotadas por uma organização para evolução de seus sistemas legados as apresentadas a seguir,EXCETO:

- a) descartar o sistema completamente.
- b) substituir todo ou parte do sistema por um novo sistema.
- c) deixar o sistema sem alterações e continuar com a manutenção regular.
- d) fazer reengenharia do sistema para aprimorar sua facilidade de manutenção.
- ➡ e) realizar engenharia reversa do código do sistema com o objetivo de aumentar sua complexidade ciclomática e torná-lo crítico à empresa.

Questão 28 – DECEA - 2006

Que paradigma da Engenharia de Software é seqüencial e sistemático, iniciando no nível de sistemas e se estendendo pela análise, projeto, codificação, teste e manutenção?

- A) Espiral
- B) Clássico
- C) Prototipação
- D) Técnicas de quarta geração
- E) Iterativo e incremental

Questão 28 – DECEA - 2006

Que paradigma da Engenharia de Software é seqüencial e sistemático, iniciando no nível de sistemas e se estendendo pela análise, projeto, codificação, teste e manutenção?

A) Espiral

➡ B) Clássico

C) Prototipação

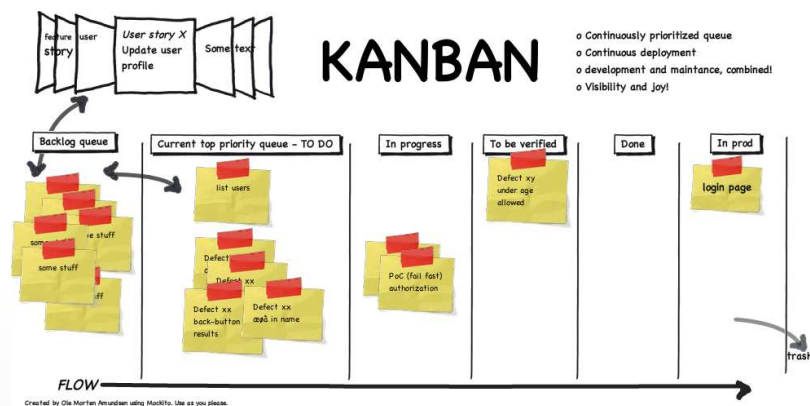
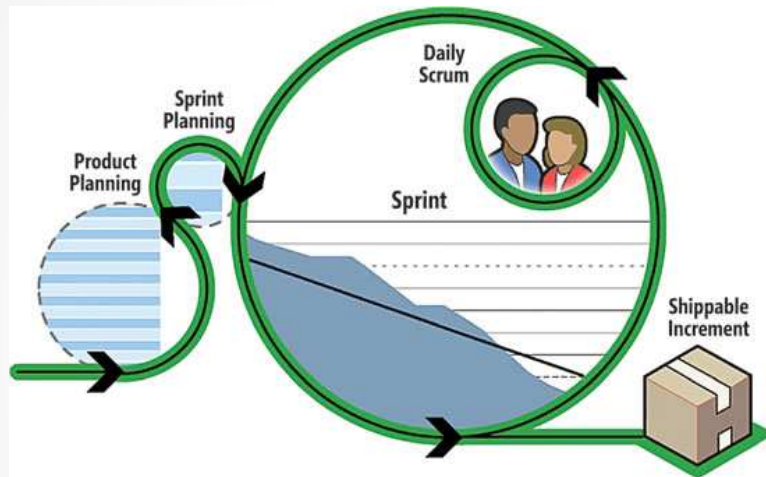
D) Técnicas de quarta geração

E) Iterativo e incremental

Gabarito – Ciclo de Vida

1 – C	10 – B	19 – E
2 – B	11 – E	20 – E
3 – C	12 – D	21 – B
4 – B	13 – A	22 – A
5 – A	14 – C	23 – C
6 – B	15 – D	24 – C
7 – D	16 – D	25 – D
8 – C	17 – E	26 – A
9 – E	18 – E	27 – E
		28 – B

Met. Ageis



Questão 01 – FINEP - 2011

São práticas recomendadas pelo processo ágil de desenvolvimento de software Extreme Programming (XP), **EXCETO** a

- a) Programação em Pares
- b) Integração Contínua
- c) Documentação Abundante e Detalhada
- d) Refatoração Frequente
- e) Padronização de Código

Práticas do XP

- **Jogo do Planejamento (Planning Game)**
 - Desenvolvedores e clientes se reúnem para priorizar as funcionalidades daquela iteração;
- **Fases Pequenas (Small Releases)**
 - Liberação de pequenas versões funcionais;
- **Metáfora**
 - Utilização de metáforas do mundo real para que todos entendam como funciona o sistema.
- **Design Simples**
 - O código será o mais simples possível de acordo com o que está sendo solicitado.
- **Time Coeso (Whole Team)**
 - A equipe é formada por pessoas engajadas e multidisciplinares

Práticas do XP

- Testes de Aceitação (Customer Tests)
 - Testes para aceitar uma determinada funcionalidade do sistema
- Ritmo Sustentável
 - Cada programador trabalha 40 horas por semana. Trabalhar com qualidade.
- Reuniões em Pé (Stand-up Meeting)
 - Produzir reuniões rápidas e diárias, com foco e em até 15 minutos.
- Propriedade Coletiva do Código
 - Todos são responsáveis por todo o código e não é preciso autorização para alterar qualquer mudança no código.
- Padronização do Código (Coding Standards)
 - Regras devem ser estabelecidas para programas e todos devem segui-las.

Práticas do XP

- **Programação em Pares (Pair Programming)**
 - Desenvolvedores trabalham em pares, um colaborando com o outro. Dividem o mesmo computador.
- **Desenvolvimento Orientado a Testes (Test Driven)**
 - Primeiro são criados os testes unitários e depois o código para que os testes funcionem
- **Cliente sempre Presente**
 - Cliente com conhecimento deve estar disponível em tempo integral para a equipe
- **Refatoração**
 - Permite a alteração da estrutura do código sem modificar sua funcionalidade;
- **Integração Contínua**
 - Diversos módulos são integrados diversas vezes por dia.

Questão 01 – FINEP - 2011

São práticas recomendadas pelo processo ágil de desenvolvimento de software Extreme Programming (XP), **EXCETO** a

- a) Programação em Pares
- b) Integração Contínua
- c) Documentação Abundante e Detalhada
- d) Refatoração Frequente
- e) Padronização de Código

Questão 01 – FINEP - 2011

São práticas recomendadas pelo processo ágil de desenvolvimento de software Extreme Programming (XP), **EXCETO** a

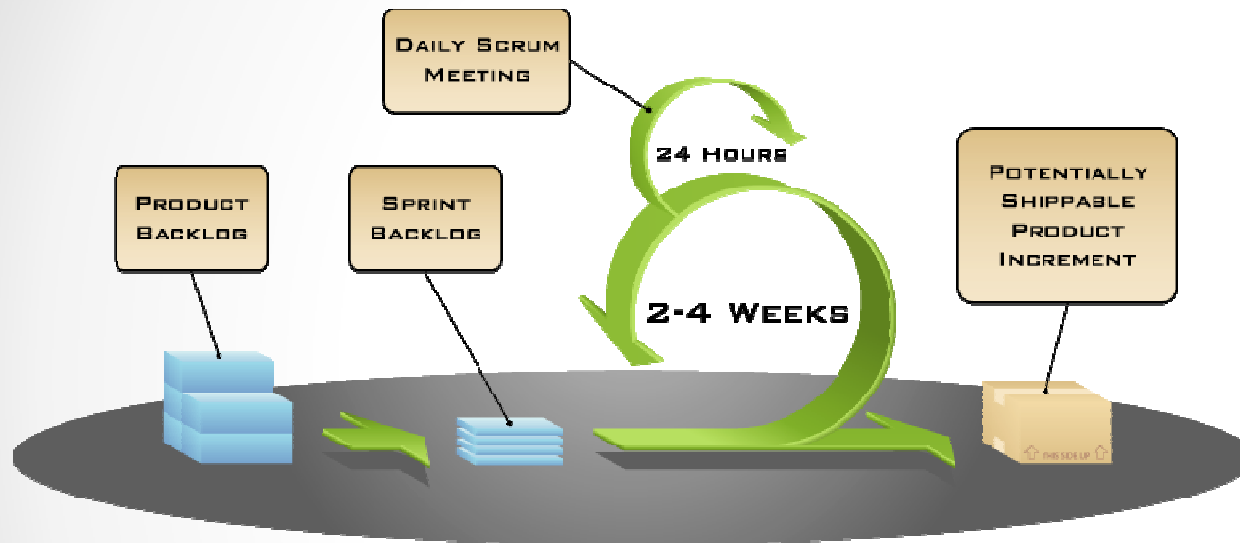
- a) Programação em Pares
- b) Integração Contínua
- ➡ c) Documentação Abundante e Detalhada
- d) Refatoração Frequente
- e) Padronização de Código

Questão 02 – ELETROBRÁS - 2010

No âmbito do desenvolvimento ágil de sistemas de informação, é **INCORRETO** afirmar que, no SCRUM,

- a) as atividades são definidas com uma duração fixa.
- b) o foco é nas tarefas e não nos objetivos e resultados.
- c) o desenvolvimento é iterativo e incremental.
- d) cada iteração foca nas necessidades mais prioritárias.
- e) cada iteração é finalizada com funcionalidades completas.

Características - Scrum



COPYRIGHT © 2009, MOUNTAIN GOAT SOFTWARE


Product Owner


Scrum Master


Equipe de Desenvolvimento

Questão 02 – ELETROBRÁS - 2010

No âmbito do desenvolvimento ágil de sistemas de informação, é **INCORRETO** afirmar que, no SCRUM,

- a) as atividades são definidas com uma duração fixa.
- b) o foco é nas tarefas e não nos objetivos e resultados.
- c) o desenvolvimento é iterativo e incremental.
- d) cada iteração foca nas necessidades mais prioritárias.
- e) cada iteração é finalizada com funcionalidades completas.

Questão 02 – ELETROBRÁS - 2010

No âmbito do desenvolvimento ágil de sistemas de informação, é **INCORRETO** afirmar que, no SCRUM,

- a) as atividades são definidas com uma duração fixa.
- ➡ b) o foco é nas tarefas e não nos objetivos e resultados.
- c) o desenvolvimento é iterativo e incremental.
- d) cada iteração foca nas necessidades mais prioritárias.
- e) cada iteração é finalizada com funcionalidades completas.

Questão 03 – ELETROBRÁS - 2010

No SCRUM, que papel é responsável pela visão do produto e pelo retorno do investimento?

- a) Scrum Master.
- b) Product Owner.
- c) Sprint Planner.
- d) Gerente do Projeto.
- e) Analista de Sistemas Sênior.

Questão 03 – ELETROBRÁS - 2010

No SCRUM, que papel é responsável pela visão do produto e pelo retorno do investimento?

- a) Scrum Master.
- ➡ b) Product Owner.
- c) Sprint Planner.
- d) Gerente do Projeto.
- e) Analista de Sistemas Sênior.

Questão 04 – ELETROBRÁS - 2010

No âmbito de desenvolvimento de sistemas, o XP tem como característica a programação em par, na qual o(a)

- a) programador codifica em companhia do analista de requisitos.
- b) testador realiza os casos de teste em companhia do programador.
- c) código é escrito em par, reduzindo a inserção de bugs.
- d) programador codifica em companhia do gestor do sistema.
- e) disseminação do conhecimento não é priorizada, mas, sim, a individualidade.

Questão 04 – ELETROBRÁS - 2010

No âmbito de desenvolvimento de sistemas, o XP tem como característica a programação em par, na qual o(a)

- a) programador codifica em companhia do analista de requisitos.
- b) testador realiza os casos de teste em companhia do programador.
- ➡ c) código é escrito em par, reduzindo a inserção de bugs.
- d) programador codifica em companhia do gestor do sistema.
- e) disseminação do conhecimento não é priorizada, mas, sim, a individualidade.

Questão 04 – BNDES - 2012

Sendo atualmente conhecida por just-in-time, a produção enxuta contém princípios que compõem a base dos processos ágeis de desenvolvimento de software, como o Extreme Programming (XP).

Um dos princípios básicos do XP, a eliminação de desperdícios, busca

- (A) evitar o efeito negativo que uma definição de risco, na fase inicial do projeto, possa causar na performance do software como um todo, tendo, como saída, informações não relevantes para o processo.
- (B) produzir requisitos bem definidos e completos de forma a abranger todos os processos e rotinas administrativas, funcionais e produtivas almejadas pelos stakeholders envolvidos no projeto.
- (C) reduzir, o máximo possível, o volume de trabalho executado e os subprodutos envolvidos nesse trabalho, concentrando os esforços apenas no que pode produzir um resultado objetivo e palpável ao cliente final.
- (D) descrever os processos que garantam a inclusão, no projeto, de todo o serviço necessário, e somente o serviço necessário, para que esse projeto seja finalizado com sucesso.
- (E) descrever os processos envolvidos no planejamento, no monitoramento e na garantia de que o projeto será realizado dentro dos prazos definidos no escopo, mantendo a qualidade definida e o enxugamento dos custos inicialmente programados.

Questão 04 – BNDES - 2012

Sendo atualmente conhecida por just-in-time, a produção enxuta contém princípios que compõem a base dos processos ágeis de desenvolvimento de software, como o Extreme Programming (XP).

Um dos princípios básicos do XP, a eliminação de desperdícios, busca

(A) evitar o efeito negativo que uma definição de risco, na fase inicial do projeto, possa causar na performance do software como um todo, tendo, como saída, informações não relevantes para o processo.

(B) produzir requisitos bem definidos e completos de forma a abranger todos os processos e rotinas administrativas, funcionais e produtivas almejadas pelos stakeholders envolvidos no projeto.

➡ (C) reduzir, o máximo possível, o volume de trabalho executado e os subprodutos envolvidos nesse trabalho, concentrando os esforços apenas no que pode produzir um resultado objetivo e palpável ao cliente final.

(D) descrever os processos que garantam a inclusão, no projeto, de todo o serviço necessário, e somente o serviço necessário, para que esse projeto seja finalizado com sucesso.

(E) descrever os processos envolvidos no planejamento, no monitoramento e na garantia de que o projeto será realizado dentro dos prazos definidos no escopo, mantendo a qualidade definida e o enxugamento dos custos inicialmente programados.

Questão 05 – IBGE - 2010

O XP (Extreme Programming) usa uma abordagem orientada a objetos como seu paradigma de desenvolvimento predileto. Nessa perspectiva, analise as afirmativas abaixo.

I – A atividade de Codificação começa com a criação de um conjunto de histórias que descreve as características e as funcionalidades requeridas para o software a ser construído.

II – O XP encoraja o uso de cartões CRC (Class– Responsibility–Colaborator) como um mecanismo efetivo para raciocinar sobre o software no contexto orientado a objetos.

III – O XP emprega a técnica de refactoring na codificação, mas desaconselha a utilização da programação por pares.

IV – A criação de testes unitários antes da codificação começar é uma prática do XP.

V – Se um difícil problema de projeto é encontrado como parte do projeto de uma história, o XP recomenda a criação imediata de um protótipo operacional daquela parte do projeto.

Estão corretas APENAS as afirmativas

- a) I, II e IV.
- b) I, III e IV.
- c) I, IV e V.
- d) II, III e V.
- e) II, IV e V.

Questão 05 – IBGE - 2010

O XP (Extreme Programming) usa uma abordagem orientada a objetos como seu paradigma de desenvolvimento predileto. Nessa perspectiva, analise as afirmativas abaixo.

I – A atividade de Codificação começa com a criação de um conjunto de histórias que descreve as características e as funcionalidades requeridas para o software a ser construído.

II – O XP encoraja o uso de cartões CRC (Class– Responsibility–Colaborator) como um mecanismo efetivo para raciocinar sobre o software no contexto orientado a objetos.

III – O XP emprega a técnica de refactoring na codificação, mas desaconselha a utilização da programação por pares.

IV – A criação de testes unitários antes da codificação começar é uma prática do XP.

V – Se um difícil problema de projeto é encontrado como parte do projeto de uma história, o XP recomenda a criação imediata de um protótipo operacional daquela parte do projeto.

Estão corretas APENAS as afirmativas

- a) I, II e IV.
- b) I, III e IV.
- c) I, IV e V.
- d) II, III e V.
- e) II, IV e V.



Questão 06 – BNDES - 2009

Determinado projeto de software utiliza XP (*eXtreme Programming*) como metodologia de desenvolvimento. A esse respeito, é **INCORRETO** afirmar que

- a) o cliente participa ativamente e acompanha os passos dos desenvolvedores diariamente.
- b) os integrantes da equipe se reúnem rapidamente no início do dia, de preferência em pé.
- c) a equipe de desenvolvimento concentra esforços naquilo que gera maior valor para o cliente.
- d) a programação em pares dispensa o desenvolvimento orientado a testes no projeto.
- e) as funcionalidades do software são descritas em histórias, da forma mais simples possível.

Questão 06 – BNDES - 2009

Determinado projeto de software utiliza XP (*eXtreme Programming*) como metodologia de desenvolvimento. A esse respeito, é **INCORRETO** afirmar que

- a) o cliente participa ativamente e acompanha os passos dos desenvolvedores diariamente.
- b) os integrantes da equipe se reúnem rapidamente no início do dia, de preferência em pé.
- c) a equipe de desenvolvimento concentra esforços naquilo que gera maior valor para o cliente.
- ➡ d) a programação em pares dispensa o desenvolvimento orientado a testes no projeto.
- e) as funcionalidades do software são descritas em histórias, da forma mais simples possível.

Questão 07 – BNDES - 2009

Que situação favorece a escolha do uso de XP para um projeto de desenvolvimento de software, em oposição à escolha do RUP ou do modelo Cascata?

- a) Equipe do projeto localizada em diferentes cidades e com poucos recursos de colaboração.
- b) Equipe do projeto formada por pessoas com alto grau de competitividade.
- c) Cliente do projeto trabalhando em parceria com a equipe do projeto e sempre disponível para retirar dúvidas.
- d) Requisitos do software com pequena probabilidade de mudanças.
- e) Presença de um processo organizacional que exige a elaboração de vários documentos específicos para cada projeto.

Questão 07 – BNDES - 2009

Que situação favorece a escolha do uso de XP para um projeto de desenvolvimento de software, em oposição à escolha do RUP ou do modelo Cascata?

- a) Equipe do projeto localizada em diferentes cidades e com poucos recursos de colaboração.
- b) Equipe do projeto formada por pessoas com alto grau de competitividade.
- ➡ c) Cliente do projeto trabalhando em parceria com a equipe do projeto e sempre disponível para retirar dúvidas.
- d) Requisitos do software com pequena probabilidade de mudanças.
- e) Presença de um processo organizacional que exige a elaboração de vários documentos específicos para cada projeto.

Questão 08 – REFAP/SA - 2007

NÃO é uma característica da Extreme Programming (XP):

- a) simplicidade.
- b) agilidade.
- c) desenvolvimento orientado a testes.
- d) programação em par.
- e) documentação extensa e abundante em artefatos.

Questão 08 – REFAP/SA - 2007

NÃO é uma característica da Extreme Programming (XP):

- a) simplicidade.
- b) agilidade.
- c) desenvolvimento orientado a testes.
- d) programação em par.
- ➡ e) documentação extensa e abundante em artefatos.

Questão 9 – DECEA - 2009

O gerente de desenvolvimento de sistemas de uma empresa adotou o XP (eXtreme Programming) como metodologia de desenvolvimento de software. Nessa perspectiva, observe as afirmativas a seguir.

- I – A programação é realizada por duplas em um único computador, com o objetivo de melhorar a qualidade do código-fonte.
- II – Reuniões ocorrem em pé para que sejam rápidas e não se perca o foco nos assuntos.
- III – A documentação produzida não deve ser detalhada.

Está(ão) correta(s) a(s) afirmativa(s)

- A) I, apenas.
- B) II, apenas.
- C) III, apenas.
- D) I e II, apenas.
- E) I, II e III.

Questão 9 – DECEA - 2009

O gerente de desenvolvimento de sistemas de uma empresa adotou o XP (eXtreme Programming) como metodologia de desenvolvimento de software. Nessa perspectiva, observe as afirmativas a seguir.

- I – A programação é realizada por duplas em um único computador, com o objetivo de melhorar a qualidade do código-fonte.
- II – Reuniões ocorrem em pé para que sejam rápidas e não se perca o foco nos assuntos.
- III – A documentação produzida não deve ser detalhada.

Está(ão) correta(s) a(s) afirmativa(s)

- A) I, apenas.
- B) II, apenas.
- C) III, apenas.
- D) I e II, apenas.

➡ E) I, II e III.

Questão 10 – BR - 2011

Dentre as metodologias de desenvolvimento ágil, a eXtreme Programming (XP) é uma das mais conhecidas. Nessa metodologia, são usados os conceitos de teste de aceitação (acceptance test) e de história do usuário (user story). Com relação às práticas recomendadas pela XP, analise as afirmações a seguir.

I – Uma história do usuário deve refletir corretamente as necessidades do cliente com relação a certa funcionalidade do sistema esperada por esse cliente.

II – É uma prática recomendada pela XP que um desenvolvedor especifique os cenários para os testes de aceitação de cada história de usuário que ele implemente.

III – Uma história de usuário é exatamente igual a um caso de uso, devendo ser utilizada em conjunto com um documento de requisitos e ter apenas um teste de aceitação escrito para ela.

É correto **APENAS** o que se afirma em

- (A) I
- (B) II
- (C) III
- (D) I e II
- (E) II e III

Questão 10 – BR - 2011

Dentre as metodologias de desenvolvimento ágil, a eXtreme Programming (XP) é uma das mais conhecidas. Nessa metodologia, são usados os conceitos de teste de aceitação (acceptance test) e de história do usuário (user story). Com relação às práticas recomendadas pela XP, analise as afirmações a seguir.

I – Uma história do usuário deve refletir corretamente as necessidades do cliente com relação a certa funcionalidade do sistema esperada por esse cliente.

II – É uma prática recomendada pela XP que um desenvolvedor especifique os cenários para os testes de aceitação de cada história de usuário que ele implemente.

III – Uma história de usuário é exatamente igual a um caso de uso, devendo ser utilizada em conjunto com um documento de requisitos e ter apenas um teste de aceitação escrito para ela.

É correto **APENAS** o que se afirma em



- (A) I
- (B) II
- (C) III
- (D) I e II
- (E) II e III

Questão 11 – ELETROBRÁS - 2010

Em um projeto de desenvolvimento de software, os membros da equipe do projeto conversam, diariamente, numa rápida reunião, para verificar o andamento das tarefas e expor eventuais dificuldades. Essa equipe é multidisciplinar, composta predominantemente de profissionais experientes que trabalham em conjunto com, pelo menos, um representante do cliente. As iterações de trabalho são curtas e, ao final de cada uma delas, o produto ganha novas funcionalidades. Nesse momento, a versão atual é apresentada funcionando ao cliente, visto que ter o software funcionando é mais importante do que ter uma documentação detalhada. O modelo de desenvolvimento de sistemas que se encaixa nesse cenário é o

- a) em espiral.
- b) de software aberto.
- c) de prototipagem rápida.
- d) scrum.
- e) cascata.

Questão 11 – ELETROBRÁS - 2010

Em um projeto de desenvolvimento de software, os membros da equipe do projeto conversam, diariamente, numa rápida reunião, para verificar o andamento das tarefas e expor eventuais dificuldades. Essa equipe é multidisciplinar, composta predominantemente de profissionais experientes que trabalham em conjunto com, pelo menos, um representante do cliente. As iterações de trabalho são curtas e, ao final de cada uma delas, o produto ganha novas funcionalidades. Nesse momento, a versão atual é apresentada funcionando ao cliente, visto que ter o software funcionando é mais importante do que ter uma documentação detalhada. O modelo de desenvolvimento de sistemas que se encaixa nesse cenário é o

- a) em espiral.
- b) de software aberto.
- c) de prototipagem rápida.
- ➡ d) scrum.
- e) cascata.

Questão 12 – PETROBRÁS - 2008

Diversos processos de software emergiram nos últimos anos, principalmente aqueles de natureza iterativa. Podemos citar o Processo Unificado, XP (Extreme Programming) e, mais recentemente, o Scrum, como alguns dos mais difundidos. Correlacione, a seguir, esses processos de software e suas características.

Processos	Características
I - Processo Unificado	P - Reuniões em pé, diárias, com perguntas específicas
II - Extreme Programming	Q - Refatoração freqüente
III - Scrum	R - Priorização dos maiores riscos
	S - Eliminação do desperdício

A relação correta é

- a) I – P, II – Q, III – R.
- b) I – R, II – Q, III – P.
- c) I – R, II – S, III – Q.
- d) I – S, II – P, III – R.
- e) I – S, II – R, III – P.

Questão 12 – PETROBRÁS - 2008

Diversos processos de software emergiram nos últimos anos, principalmente aqueles de natureza iterativa. Podemos citar o Processo Unificado, XP (Extreme Programming) e, mais recentemente, o Scrum, como alguns dos mais difundidos. Correlacione, a seguir, esses processos de software e suas características.

Processos	Características
I - Processo Unificado	P - Reuniões em pé, diárias, com perguntas específicas
II - Extreme Programming	Q - Refatoração freqüente
III - Scrum	R - Priorização dos maiores riscos
	S - Eliminação do desperdício

A relação correta é

- a) I – P, II – Q, III – R.
- ➡ b) I – R, II – Q, III – P.
- c) I – R, II – S, III – Q.
- d) I – S, II – P, III – R.
- e) I – S, II – R, III – P.

Questão 13 – PETROBRÁS - 2006

Há um considerável debate sobre os benefícios e a aplicabilidade do desenvolvimento ágil de software em contraposição aos processos mais convencionais de engenharia de software. Relacione o modelo ágil de software com a sua respectiva característica.

Modelo

I – DAS II – DSDM III – FDD IV – XP

Característica

(P)

Define um ciclo de vida que incorpora três fases: especulação, colaboração e aprendizado. Durante a fase de aprendizado, à medida que os membros de uma equipe começam a desenvolver os componentes que fazem parte de um ciclo adaptativo, a ênfase está tanto no aprendizado quanto no progresso em direção a um ciclo completo.

(Q)

O conceito característica é uma função valorizada pelo cliente, que pode ser implementada em duas semanas ou menos. Este modelo define seis marcos de referência durante o projeto e implementação de uma característica: travessia do projeto, projeto, inspeção de projeto, código, inspeção de código, promoção para construção.

(R)

Fornece um arcabouço para construir e manter sistemas que satisfazem às restrições de prazo apertadas por meio do uso de prototipagem incremental em ambiente controlado de projeto. Essa abordagem sugere uma filosofia que é emprestada de uma versão modificada do princípio de Pareto.

A relação correta é:

- a) I – P, II – Q, III – R.
- b) I – P, II – R, III – Q.
- c) I – Q, III – R, IV – P.
- d) II – P, III – R, IV – Q.
- e) II – Q, III – P, IV – R.

ASD (Adaptative Software Development)

- Técnica para construção de software complexos;
- Bases se concentram na colaboração humana e na auto-organização das equipes;
- Ciclo de vida com 3 fases: especulação, colaboração e aprendizagem;
- Enfatiza o aprendizado e o progresso para um ciclo completo;
- Adaptação contínua do processo de acordo com o trabalho em questão;

DSDM – Dynamic Systems Development Method

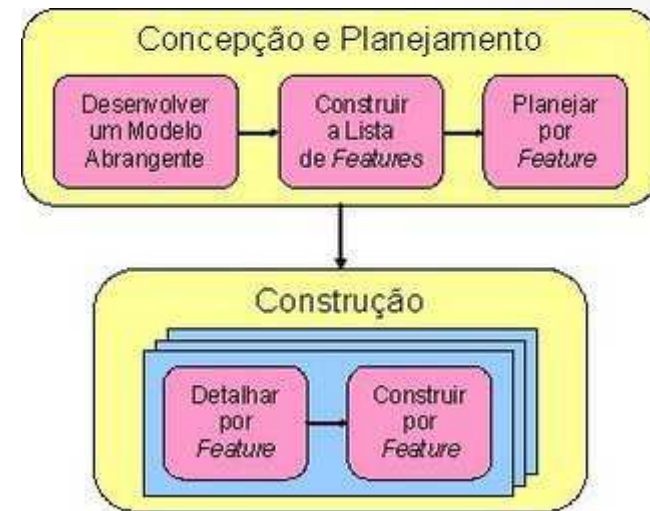
- Método de Desenvolvimento Dinâmico de Sistemas;
- Oferece uma metodologia para se trabalhar com restrições de prazo;
- Utiliza prototipagem incremental em um ambiente controlado;
- Baseia-se no princípio de Pareto (80/20);
- Cada iteração segue a regra dos 80%;
- Contempla 3 fases: pré-projeto, ciclo de vida e pós-projeto;

FDD – Feature Driven Development

- Desenvolvimento Dirigido por Funcionalidade
- Focado na entrega regular de funcionalidades para o cliente;
- Equipes podem variar de 10 a 250 programadores;
- Menos detalhado que outras metodologias XP;
- Funcionalidade = “uma função valorizada pelo cliente passível de ser implementada em duas semana ou menos”;
- Relatórios de acompanhamento;

FDD – Feature Driven Development

- Define 5 processos:
 - Desenvolver um Modelo Geral,
 - Construir uma Lista de Funcionalidades,
 - Planejar por Funcionalidades,
 - Projetar por Funcionalidade,
 - Desenvolver por Funcionalidade.
- Define seis marcos:
 - Travessia do Projeto,
 - Projeto,
 - Inspeção do Projeto,
 - Código,
 - Inspeção do Código,
 - Promoção para Construção.



Questão 13 – PETROBRÁS - 2006

Há um considerável debate sobre os benefícios e a aplicabilidade do desenvolvimento ágil de software em contraposição aos processos mais convencionais de engenharia de software. Relacione o modelo ágil de software com a sua respectiva característica.

Modelo

I – DAS II – DSDM III – FDD IV – XP

Característica

(P)

Define um ciclo de vida que incorpora três fases: especulação, colaboração e aprendizado. Durante a fase de aprendizado, à medida que os membros de uma equipe começam a desenvolver os componentes que fazem parte de um ciclo adaptativo, a ênfase está tanto no aprendizado quanto no progresso em direção a um ciclo completo.

(Q)

O conceito característica é uma função valorizada pelo cliente, que pode ser implementada em duas semanas ou menos. Este modelo define seis marcos de referência durante o projeto e implementação de uma característica: travessia do projeto, projeto, inspeção de projeto, código, inspeção de código, promoção para construção.

(R)

Fornece um arcabouço para construir e manter sistemas que satisfazem às restrições de prazo apertadas por meio do uso de prototipagem incremental em ambiente controlado de projeto. Essa abordagem sugere uma filosofia que é emprestada de uma versão modificada do princípio de Pareto.

A relação correta é:

- a) I – P, II – Q, III – R.
- b) I – P, II – R, III – Q.
- c) I – Q, III – R, IV – P.
- d) II – P, III – R, IV – Q.
- e) II – Q, III – P, IV – R.

Questão 13 – PETROBRÁS - 2006

Há um considerável debate sobre os benefícios e a aplicabilidade do desenvolvimento ágil de software em contraposição aos processos mais convencionais de engenharia de software. Relacione o modelo ágil de software com a sua respectiva característica.

Modelo

I – DAS II – DSDM III – FDD IV – XP

Característica

(P)

Define um ciclo de vida que incorpora três fases: especulação, colaboração e aprendizado. Durante a fase de aprendizado, à medida que os membros de uma equipe começam a desenvolver os componentes que fazem parte de um ciclo adaptativo, a ênfase está tanto no aprendizado quanto no progresso em direção a um ciclo completo.

(Q)

O conceito característica é uma função valorizada pelo cliente, que pode ser implementada em duas semanas ou menos. Este modelo define seis marcos de referência durante o projeto e implementação de uma característica: travessia do projeto, projeto, inspeção de projeto, código, inspeção de código, promoção para construção.

(R)

Fornece um arcabouço para construir e manter sistemas que satisfazem às restrições de prazo apertadas por meio do uso de prototipagem incremental em ambiente controlado de projeto. Essa abordagem sugere uma filosofia que é emprestada de uma versão modificada do princípio de Pareto.

A relação correta é:

- a) I – P, II – Q, III – R.
- ➡ b) I – P, II – R, III – Q.
- c) I – Q, III – R, IV – P.
- d) II – P, III – R, IV – Q.
- e) II – Q, III – P, IV – R.

Gabarito – Met. Agéis

1 – C	5 – E	10 – A
2 – B	6 – D	11 – D
3 – B	7 – C	12 – B
4 – C	8 – E	13 – B
4* – C	9 – E	