

Aula 00

*Passo Estratégico de Tecnologia da
Informação p/ TJ-RJ (Analista de
Sistemas) - Pós-Edital*

Autor:

Thiago Rodrigues Cavalcanti

13 de Março de 2020

CONCEITOS DE ENGENHARIA DE SOFTWARE. PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Sumário

Apresentação.....	1
O que é o Passo Estratégico?	2
Análise Estatística	2
Roteiro de revisão e pontos do assunto que merecem destaque	3
Conceitos de Engenharia de Software	3
Definição de Software	4
Software Livre.....	6
Princípios da Engenharia de Software.....	7
Camadas da Engenharia de Software	8
Processo de Software.....	9
Aposta estratégica	11
Questões estratégicas	13

APRESENTAÇÃO

Olá Senhoras e Senhores,

Eu me chamo Thiago Cavalcanti. Sou funcionário do Banco Central do Brasil, passei no concurso em 2010 para Analista de Tecnologia da Informação (TI). Atualmente estou de licença, cursando doutorado em economia na UnB. Também trabalho como professor de TI no Estratégia e sou o analista do Passo Estratégico de Informática.

Tenho graduação em Ciência da Computação pela UFPE e mestrado em Engenharia de Software. Já fui aprovado em diversos concursos tais como ANAC, BNDES, TCE-RN, INFRAERO e, claro, Banco Central. A minha trajetória como concurseiro durou pouco mais de dois anos. Neste intervalo, aprendi muito e vou tentar passar um pouco desta minha experiência ao longo deste curso.



O QUE É O PASSO ESTRATÉGICO?

O Passo Estratégico é um material escrito e enxuto que possui dois objetivos principais:

- a) orientar revisões eficientes;
- b) destacar os pontos mais importantes e prováveis de serem cobrados em prova.

Assim, o Passo Estratégico pode ser utilizado tanto para **turbinar as revisões dos alunos mais adiantados nas matérias, quanto para maximizar o resultado na reta final de estudos por parte dos alunos que não conseguirão estudar todo o conteúdo do curso regular.**

Em ambas as formas de utilização, como regra, **o aluno precisa utilizar o Passo Estratégico em conjunto com um curso regular completo.**

Isso porque nossa didática é direcionada ao aluno que já possui uma base do conteúdo.

Assim, se você vai utilizar o Passo Estratégico:

- a) **como método de revisão**, você precisará de seu curso completo para realizar as leituras indicadas no próprio Passo Estratégico, em complemento ao conteúdo entregue diretamente em nossos relatórios;
- b) **como material de reta final**, você precisará de seu curso completo para buscar maiores esclarecimentos sobre alguns pontos do conteúdo que, em nosso relatório, foram eventualmente expostos utilizando uma didática mais avançada que a sua capacidade de compreensão, em razão do seu nível de conhecimento do assunto.

Seu cantinho de estudos famoso!

Poste uma foto do seu cantinho de estudos nos stories do Instagram e nos marque:



[@passoestrategico](https://www.instagram.com/passoestrategico)

Vamos repostar sua foto no nosso perfil para que ele fique famoso entre milhares de concurseiros!

ANÁLISE ESTATÍSTICA

Inicialmente, convém destacar os percentuais de incidência de todos os assuntos previstos no nosso curso – quanto maior o percentual de cobrança de um dado assunto, maior sua importância:

Assunto	Grau de incidência em concursos similares
---------	---



Cebraspe	
2.1 CMMI-DEV, MPS.BR, RUP, Scrum e Programação Ágil. 4 Padrões de arquitetura MVC (Model View Controller) e DDD (Domain Driven Design). 5 Ferramentas de desenvolvimento de software (ferramentas CASE) e linguagem UML (versão 2.0).	18,28%
9 Conceitos de linguagens de programação Java (versão 6 ou superior) e PHP (versão 5 ou superior). 10 Web Services. 10.1 Padrões SOAP e REST. 11 Ferramentas de controle de versão. 11.1 SVN e GIT. 12 Bancos de dados transacionais (OLTP) e analíticos (OLAP).	14,11%
6 Arquitetura de software. 6.1 Cliente/servidor, Internet e dispositivos móveis. 7 Engenharia de requisitos. 8 Análise de negócios.	11,69%
14 Linguagem de definição e manipulação de dados SQL.	11,29%
18 ITIL v3.	10,75%
17 COBIT 5.	9,01%
13 Modelagem de dados relacional e dimensional.	6,85%
3 Orientação a objetos. 3.1 Conceitos fundamentais, análise, modelagem e padrões de projeto.	5,91%
1 Conceitos de engenharia de software. 2 Processos de desenvolvimento de software.	5,51%
15 Conceitos sobre sistemas gerenciadores de banco de dados. 15.1 MySQL, Microsoft SQL SERVER, POSTGRESQL, ENTERPRISE DB e ORACLE.	5,11%
16 Mensuração de sistemas em Pontos de Função segundo o Manual de Práticas de Contagem (CPM versão 4.3 do IFPUG) e Roteiro de Métricas de Software do SISP (versão 2.0).	1,48%

ROTEIRO DE REVISÃO E PONTOS DO ASSUNTO QUE MERECEM DESTAQUE

A ideia desta seção é apresentar um roteiro para que você realize uma revisão completa do assunto e, ao mesmo tempo, destacar aspectos do conteúdo que merecem atenção.

Para revisar e ficar bem preparado no assunto, você precisa, basicamente, seguir os passos a seguir:

Conceitos de Engenharia de Software

O termo **Engenharia de Software** foi criado na década de 1960 e utilizado oficialmente em 1968 na *NATO Conference on Software Engineering* (Conferência sobre Engenharia de Software da OTAN). Ele foi criado na tentativa de contornar a dificuldade do desenvolvimento de softwares frente ao crescimento da demanda, conhecida na década de 1970 como crise do software, além de trazer uma abordagem de engenharia para o desenvolvimento de sistemas complexos.



Segundo Friedrich Ludwig Bauer¹, “Engenharia de software é a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais”.

De acordo com o IEEE², “Engenharia de software é a aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do software; isto é, a aplicação de engenharia ao software.”.

Engenharia de software é uma abordagem sistemática e disciplinada para o desenvolvimento de software (PRESSMAN³, 2006).

Em suma, a Engenharia de Software é uma área do conhecimento da Computação que busca estruturar de forma racional e científica, através do uso de modelos matemáticos, a especificação, desenvolvimento e manutenção de sistemas de software aplicando tecnologias e métodos da Ciência da Computação, Gerência de projetos, das Engenharias e outros campos do conhecimento.

A partir destes conceitos, chegamos à conclusão que Engenharia de Software não é simplesmente programação.

Definição de Software

Em uma visão restritiva, o termo software é associado aos programas / aplicativos de computador. Entretanto, seu conceito vai muito mais além. O software são códigos em linguagem natural ou não, entendíveis por um sistema computacional, assim como qualquer dado que faça com que opere corretamente. Ou seja, ele não é apenas o programa, mas sim todos os dados e configurações necessários para que o programa funcione corretamente.

O software é o conjunto de vários artefatos e não apenas o código fonte (SOMMERVILLE, 2003).

De acordo Pressman (2006) os softwares estão divididos nas seguintes categorias:

Software de sistema

São programas que apoiam outros programas, como por exemplo o software que realiza a comunicação com o hardware (sistema operacional) e software que ajuda na construção de outro software (compiladores).

¹ Cientista da computação alemão, autor da primeira definição de engenharia de software.

² Instituto de Engenheiros Eletricistas e Eletrônicos (<http://www.ieee.org.br/>) - sociedade técnico-profissional internacional, dedicada ao avanço da teoria e prática da engenharia nos campos da eletricidade, eletrônica e computação.

³ Roger S. Pressman é um engenheiro de software, escritor e consultor, norte-americano. Autoridade reconhecida internacionalmente nas tecnologias em melhoria de processos de software e engenharia de software.



Software de aplicação

São os programas desenvolvidos para auxiliar na execução de funções específicas para o negócio de uma empresa, por exemplo o processamento de dados comerciais ou técnicos.

Software científico e de engenharia

São programas que auxiliam no processamento de números em massa, por exemplo na astronomia, meteorologia, biologia molecular.

Software embutido (embarcado)

São programas desenvolvidos para serem executados dentro de produtos específicos, como por exemplo o sistema de teclas digitais de um forno micro-ondas.

Software para linhas de produtos

São conhecidos como software de prateleira, e projetados para prover capacidade específica de utilização por muitos clientes.

Aplicações Web / aplicativos móveis

São programas voltados para as redes e abrangem uma série de variedades, contemplando aplicativos para dispositivos móveis.

Software de inteligência artificial

São softwares que utilizam algoritmos não numéricos para solucionar problemas complexos, não passíveis de computação ou análise direta. Este tipo software está presente na robótica, no reconhecimento de voz e imagem, nos jogos.

Software legado

São os sistemas de uma organização que, apesar de serem bastante antigos, fornecem serviços essenciais e geralmente utilizam bancos de dados obsoletos. Esses sistemas podem sofrer com



dois extremos, ou são continuamente modificados para se adequar às mudanças, ou nunca são alterados quando atendem às necessidades do cliente. A grande preocupação em relação a eles está na baixa qualidade, pois muitas vezes não existem documentações, ou se existem são pobres de detalhes.

Software Livre

Também existe o conceito de software livre, que remete ao programa que dá liberdade ao utilizador, permitindo que ele o estude, modifique e compartilhe com outras pessoas. Para tanto, é necessário que o utilizador tenha acesso ao código-fonte, para alterá-lo de acordo com as suas necessidades. Por conta da confusão em relação a este termo, onde empresas proprietárias de softwares utilizam para associá-lo a cópia disponibilizada sem nenhum custo, o software livre também é dividido em categorias:

Software com copyleft

É um software livre cujos termos de distribuição não permitem que os redistribuidores incluam restrições adicionais quando redistribuem ou modificam o software. Isso significa que toda cópia, mesmo modificada, precisa ser software livre.

Software livre sem copyleft

É diferente da categoria anterior, pois o autor permite que a redistribuição e modificação possa conter restrições adicionais.

Domínio público

Software não sujeito ao copyright. Este é um caso especial de software livre sem copyleft, onde as cópias ou versões modificadas podem não ser livres.

Software com licença GPL

A licença GPL (*General Public License*) é a definição de licença de software idealizada por Richard Stallman em 1989, onde os softwares derivados de um produto originalmente licenciado pela GPL só podem ser distribuídos se utilizarem a mesma licença.



ESCLARECENDO



Copyright significa o direito de permissão de cópia de uma obra por outros usuários, dando a liberdade de copiar, modificar e redistribuir, exigindo que esse direito seja mantido em todas as versões modificadas.

Princípios da Engenharia de Software

Os princípios da Engenharia de Software constituem a base dos métodos, tecnologias, metodologias e ferramentas adotadas que norteiam a prática de desenvolvimento de software. Os princípios se aplicam ao processo e ao produto de software através da adoção de métodos e técnicas. Os princípios-chave são:

Rigor e Formalidade

Ao considerar que a engenharia de software deve ser realizada de maneira sistemática, o rigor é um complemento necessário a criatividade, que visa aumentar a confiança dos desenvolvedores. A formalidade é o rigor no seu nível mais elevado. Nesse princípio estão incluídas por exemplo a documentação rigorosa dos passos de desenvolvimento e gerenciamento, a análise sistemática dos dados de testes, a avaliação dos prazos de entrega.

Separação de Interesses

Abrange o domínio sobre a complexidade do software, separando os problemas principais e investindo esforços para solução de um de cada vez, dividindo as responsabilidades. Como exemplo temos o desenvolvimento por fases, separando os interesses por atividades e respeitando o tempo estipulado.

Modularidade

Na modularidade um sistema complexo pode ser dividido em peças mais simples, denominadas módulos. Esse princípio depende do suporte da separação de interesses, para que ao lidar com um módulo específico seja possível ignorar os detalhes dos outros módulos. Como característica desse princípio, os módulos devem ter um alto nível de coesão, onde existe uma forte relação entre eles, mas com um baixo acoplamento, onde a interação entre eles seja baixa, possibilitando que eles sejam compreendidos como unidades em separado.

Abstração

Esse conceito visa a identificação de aspectos importantes, ignorando os detalhes. Por exemplo, os botões de um relógio que são sua interface, entretanto eles são a abstração em relação ao ajuste da hora.



Antecipação a Mudanças

Esse princípio está diretamente relacionado ao suporte à evolução de um software, considerando aspectos relacionados ao processo de evolução e compatibilidade com mudanças futuras.

Generalidade

É o princípio que visa durante a resolução de um problema descobrir se ele é apenas uma instância de um problema mais abrangente, onde a solução pode ser reutilizada em outros casos.

Incrementação

É relacionada à evolução de um software através de incrementos estruturados. Pode ser realizado através da entrega de subconjuntos de um sistema desde cedo, visando coletar o feedback dos usuários e adicionar funcionalidades de forma incremental.

Camadas da Engenharia de Software

A Engenharia de Software é uma tecnologia em camadas. Associado ao desenvolvimento, é preciso também aplicar processos, métodos e ferramentas sendo que a pedra fundamental que sustenta a engenharia de software é o foco na qualidade.



Fonte: PRESSMAN (2010).

Processo

O processo de engenharia de software é o adesivo que mantém unidas as camadas de tecnologia e permite o desenvolvimento racional e oportuno de softwares de computador.

Os processos de software formam a base para o controle gerencial de projetos de software (por exemplo: PMBOK) e estabelecem o contexto no qual:



- os métodos técnicos são aplicados;
- os produtos de trabalho (modelos, documentos, dados, relatórios, entre outros) são produzidos;
- os marcos são estabelecidos,
- a qualidade é assegurada;
- as modificações são adequadamente geridas.
-

Métodos

Os métodos de engenharia de software fornecem a técnica de “como fazer” para construir softwares. Eles abrangem um amplo conjunto de tarefas que incluem:

- comunicação;
- análise de requisitos;
- modelagem de projeto;
- construção de programas;
- testes;
- manutenção.

Ferramentas

As ferramentas de engenharia de software fornecem apoio automatizado ou semi-automatizado para o processo e para os métodos. Quando ferramentas são integradas de modo que a informação criada possa ser usada por outra, é estabelecida a chamada engenharia de software apoiada por computador.

Processo de Software

O processo de software é o conjunto de atividades que constituem o desenvolvimento de um sistema computacional. Estas atividades são agrupadas em fases, como: definição de requisitos, análise, projeto, desenvolvimento, teste e implantação. Em cada fase são definidas, além das suas atividades, as funções e responsabilidades de cada membro da equipe, e como produto resultante, os artefatos. O que diferencia um processo de software do outro é a ordem em que as fases vão ocorrer, o tempo e a ênfase dados a cada fase, as atividades presentes, e os produtos entregues.

As atividades do processo de desenvolvimento de software incluem:

Análise de requisitos de software

A extração dos requisitos de um desejado produto de software é a primeira tarefa na sua criação. Embora o cliente, provavelmente, acredite saber o que o software deva fazer, esta tarefa requer habilidade e experiência em Engenharia de Software para reconhecer a incompletude, ambiguidade ou contradição nos requisitos.



Especificação

A especificação é a tarefa de descrever precisamente o software que será escrito, preferencialmente de uma forma matematicamente rigorosa. Na prática, somente especificações mais bem-sucedidas foram escritas para aplicações bem compreendidas e afinadas que já estavam bem desenvolvidas, embora sistemas de software de missão crítica sejam frequentemente bem especificados antes do desenvolvimento da aplicação. Especificações são mais importantes para interfaces externas que devem permanecer estáveis.

Arquitetura de Software

A arquitetura de um sistema de software remete a uma representação abstrata daquele sistema. Arquitetura é concernente à garantia de que o sistema de software irá ao encontro de requisitos do produto, como também assegurar que futuros requisitos possam ser atendidos. A etapa da arquitetura também direciona as interfaces entre os sistemas de software e outros produtos de software, como também com o hardware básico ou com o sistema operacional.

Implementação (ou codificação)

A transformação de um projeto para um código deve ser a parte mais evidente do trabalho da engenharia de software, mas não necessariamente a sua maior porção.

▪

Teste

Teste de partes do software, especialmente onde tenha sido codificado por dois ou mais engenheiros trabalhando juntos, é um papel da engenharia de software.

Documentação

Uma importante tarefa é a documentação do projeto interno do software para propósitos de futuras manutenções e aprimoramentos. As documentações mais importantes são das interfaces externas.

Suporte e Treinamento de Software

Uma grande porcentagem dos projetos de software falha pelo fato de o desenvolvedor não perceber que não importa quanto tempo a equipe de planejamento e desenvolvimento irá gastar na criação do software se ninguém da organização irá usá-lo. As pessoas ocasionalmente resistem à mudança e evitam aventurar-se em áreas pouco familiares. Então, como parte da fase de desenvolvimento, é muito importante o treinamento para os usuários de software mais entusiasmados, alternando o



treinamento entre usuários neutros e usuários favoráveis ao software. Usuários irão ter muitas questões e problemas de software os quais conduzirão para a próxima fase.

Manutenção

A manutenção e melhoria de software lidam com a descoberta de novos problemas e requisitos. Ela pode tomar mais tempo que o gasto no desenvolvimento inicial do mesmo. Não somente pode ser necessário adicionar códigos que combinem com o projeto original, mas determinar como o software trabalhará em algum ponto depois da manutenção estar completa, pode requerer um significativo esforço por parte de um engenheiro de software. Cerca de 2/3 de todos os engenheiros de software trabalham com a manutenção, mas estas estatísticas podem estar enganadas. Uma pequena parte destes trabalha na correção de erros. A maioria das manutenções são para ampliar os sistemas para novas funcionalidades, as quais, de diversas formas, podem ser consideradas um novo trabalho. Analogamente, cerca de 2/3 de todos os engenheiros civis, arquitetos e construtores trabalham com manutenção de uma forma similar.

APOSTA ESTRATÉGICA

A ideia desta seção é apresentar os pontos do conteúdo que mais possuem chances de serem cobrados em prova, considerando o histórico de questões da banca em provas de nível semelhante à nossa, bem como as inovações no conteúdo, na legislação e nos entendimentos doutrinários e jurisprudenciais⁴.



Os princípios da Engenharia de Software constituem a base dos métodos, tecnologias, metodologias e ferramentas adotadas que norteiam a prática de desenvolvimento de software. Os princípios se aplicam ao processo e ao produto de software através da adoção de métodos e técnicas. Os princípios-chave são:

Rigor e Formalidade

Ao considerar que a engenharia de software deve ser realizada de maneira sistemática, o rigor é um complemento necessário a criatividade, que visa aumentar a confiança dos desenvolvedores.

⁴ Vale deixar claro que nem sempre será possível realizar uma aposta estratégica para um determinado assunto, considerando que às vezes não é viável identificar os pontos mais prováveis de serem cobrados a partir de critérios objetivos ou minimamente razoáveis.



A formalidade e o rigor no seu nível mais elevado. Nesse princípio estão incluídas por exemplo a documentação rigorosa dos passos de desenvolvimento e gerenciamento, a análise sistemática dos dados de testes, a avaliação dos prazos de entrega.

..... Separação de Interesses

Abrange o domínio sobre a complexidade do software, separando os problemas principais e investindo esforços para solução de um de cada vez, dividindo as responsabilidades. Como exemplo temos o desenvolvimento por fases, separando os interesses por atividades e respeitando o tempo estipulado.

..... Modularidade

Na modularidade um sistema complexo pode ser dividido em peças mais simples, denominadas módulos. Esse princípio depende do suporte da separação de interesses, para que ao lidar com um módulo específico seja possível ignorar os detalhes dos outros módulos. Como característica desse princípio, os módulos devem ter um alto nível de coesão, onde existe uma forte relação entre eles, mas com um baixo acoplamento, onde a interação entre eles seja baixa, possibilitando que eles sejam compreendidos como unidades em separado.

..... Abstração

Esse conceito visa a identificação de aspectos importantes, ignorando os detalhes. Por exemplo, os botões de um relógio que são sua interface, entretanto eles são a abstração em relação ao ajuste da hora.

..... Antecipação a Mudanças

Esse princípio está diretamente relacionado ao suporte a evolução de um software, considerando aspectos relacionados ao processo de evolução e compatibilidade com mudanças futuras.

..... Generalidade

É o princípio que visa durante a resolução de um problema descobrir se ele é apenas uma instância de um problema mais abrangente, onde a solução pode ser reutilizada em outros casos.

..... Incrementação

É relacionada a evolução de um software através de incrementos estruturados. Pode ser realizado através da entrega de subconjuntos de um sistema desde cedo, visando coletar o feedback dos usuários e adicionar funcionalidades de forma incremental.



QUESTÕES ESTRATÉGICAS

Nesta seção, apresentamos e comentamos uma amostra de questões objetivas selecionadas estrategicamente: são questões com nível de dificuldade semelhante ao que você deve esperar para a sua prova e que, em conjunto, abordam os principais pontos do assunto.

A ideia, aqui, não é que você fixe o conteúdo por meio de uma bateria extensa de questões, mas que você faça uma boa revisão global do assunto a partir de, relativamente, poucas questões.



1. CESPE - SLU-DF/2019

Com relação a desenvolvimento de software, julgue o item a seguir.

Refactoring (refatoração) é o processo utilizado para reescrever aplicações desatualizadas, com a finalidade de incrementar e melhorar suas funcionalidades; o uso dessa técnica normalmente aprimora aplicações para disponibilizá-las na Internet.

Comentários

Refatoração é o processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo e sem implementar novas funcionalidades.

A reengenharia de softwares consiste na modificação de um sistema de software (um software que já exista), a fim de reconstruí-lo, com funcionalidade semelhante a um já existente, mas de forma que se tenha melhor performance e maiores e melhores recursos nesse software, assim como no modo como ele é construído.

Portanto, a assertiva descreve o conceito de reengenharia e não de refatoração.

Gabarito: errado.

2. CESPE - SLU-DF/2019

Acerca de conceitos e disciplinas da engenharia de software, julgue o item que se segue.

O processo de desenvolvimento de software, independentemente do seu tamanho e da sua destinação, pode envolver atividades genéricas como comunicação, planejamento, modelagem, construção e uso.



Comentários

Pressman, 7ª ed, página 60, utiliza o termo "emprego" para definir a última etapa do modelo cascata (ciclo de vida clássico). Houve apenas a substituição do termo "emprego", por "uso".

O termo "emprego" no sentido de aplicação prática é sinônimo de "aplicação, uso, utilização, utilidade, serventia, aproveitamento, proveito".

Portanto, cuidado nessas pegadinhas que a CESPE/CEBRASPE pode fazer usando sinônimos.

Gabarito: certo.

3. CESPE - SLU-DF/2019

Acerca de conceitos e disciplinas da engenharia de software, julgue o item que se segue.

No processo unificado, requisitos é a disciplina que demanda maior esforço nas fases de elaboração e construção de software.

Comentários

As disciplinas que possuem maior participação na fase de Construção são: Implementação, Teste, Implantação e Gerenciamento de Configuração e Mudança.

De acordo com o gráfico das baleias, as disciplinas que têm a maior na participação na fase de Iniciação são Modelagem de Negócios e Requisitos.

Gabarito: errado.

4. CEBRASPE (CESPE) - Técnico Judiciário (STJ)/Apoio Especializado/Desenvolvimento de Sistemas/2018

Julgue o item, a respeito de engenharia de software e análise de requisitos.

Software é um conjunto de instruções (algoritmos) que, quando executadas, fornecem as características, as funções e o desempenho desejados para um sistema, para um programa ou para uma aplicação.

Comentários



O elaborador da CESPE/CEBRASPE demonstra que gosta bastante do autor Roger S. Pressman. A assertiva nos apresenta parte da definição dada por Pressman em seu livro " Engenharia de software: uma abordagem profissional".

Software consiste em: (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados; (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente; e (3) informação descritiva, tanto na forma impressa como na virtual, descrevendo a operação e o uso dos programas.

Portanto, a assertiva está correta.

Gabarito: certo.

5. CEBRASPE (CESPE) - Técnico (EBSERH)/Informática/2018

Em relação aos conceitos de engenharia de software, julgue o item subsecutivo.

A engenharia de software se concentra nos aspectos práticos da produção de um sistema de software.

Comentários

Segundo Sommerville, "a engenharia de software não se relaciona apenas com os processos técnicos de desenvolvimento de software, mas também com as atividades de gerenciamento de projeto de software e com o desenvolvimento de ferramentas, métodos e teorias que apoiem a produção de software com qualidade. Logo, a engenharia de software se concentra nos aspectos práticos da produção de um sistema de software".

Gabarito: certo.

6. CEBRASPE (CESPE) - Técnico Judiciário (TRT 8ª Região)/Apoio Especializado/Tecnologia da Informação/2016

A respeito dos processos e métodos em engenharia de software, assinale a opção correta.

- a) A qualidade de software avalia se os métodos e processos empregados devem ser aperfeiçoados, após a entrega final do produto de software.
- b) Na engenharia de software, o processo define uma metodologia, ao passo que os métodos determinam os procedimentos técnicos.
- c) Os métodos são elementos que proporcionam suporte para o controle do gerenciamento de projeto e estabelecem o marco do projeto.



- d) Em cada processo e método, as ferramentas da engenharia de software, denominadas CASE, devem ser utilizadas de forma específica, de maneira que as informações geradas não sejam integradas.
- e) O processo fornece um conjunto de informações técnicas que definem as tarefas do desenvolvimento de software.

Comentários

Processos são parte dos métodos, etapas de desenvolvimento do produto final. Os processos são definidos pelo método, que traz uma sequência de operações visando alcançar um objetivo final. Tudo isso é englobado pela metodologia, que se porta como um “caminho a trilhar”, uma diretriz de atuação, estabelecendo uma maneira de pensar e agir com relação ao desenvolvimento. Portanto, os processos são características fundamentais da metodologia empregada, enquanto os métodos são traços característicos da operação.

Portanto, a alternativa correta é a letra B.

Gabarito: alternativa B.

7. CEBRASPE (CESPE) - Técnico Judiciário (TRT 8ª Região)/Apoio Especializado/Tecnologia da Informação/2016

No que diz respeito ao processo de software, que é conceituado como uma metodologia para as atividades, ações e tarefas necessárias para se desenvolver um software de alta qualidade, assinale a opção correta.

- a) Um dos fluxos do processo de software é denominado de paralelo, pois repete uma ou mais atividades antes de prosseguir para a seguinte.
- b) Apesar de os projetos serem diferentes, os processos de software apresentam um conjunto de tarefas iguais.
- c) A aplicação de um processo de software garante que o produto de software a ser entregue estará dentro do prazo acordado e conforme as necessidades do cliente.
- d) O modelo de processo incremental combina elementos dos fluxos de processos lineares e paralelos.
- e) O processo de gerência de projeto é constituído por artefatos de software que serão produzidos, por fatores de garantia e por marcos indicadores do progresso.

Comentários

O processo incremental consiste em, a cada iteração do programa, adicionar uma nova funcionalidade, incrementando o programa. Esse processo, obviamente, segue uma linha progressiva de incrementos. Associado a isso, podem ser implementadas paralelamente novas funcionalidades que não possuem relação uma com a outra.



Portanto, a alternativa correta é a letra D.

Gabarito: alternativa D.

8. CEBRASPE (CESPE) - Analista de Controle (TCE-PR)/Tecnologia da Informação/2016

Assinale a opção correta, a respeito de engenharia de software.

- a) A engenharia de software está relacionada aos diversos aspectos de produção de software e inclui as atividades de especificação, desenvolvimento, validação e evolução de software.
- b) A engenharia de software refere-se ao estudo das teorias e fundamentos da computação, ficando o desenvolvimento de software a cargo da ciência da computação.
- c) No modelo iterativo de desenvolvimento de software, as atividades são dispostas em estágios sequenciais.
- d) Um processo de software é composto por quatro atividades fundamentais: iniciação, desenvolvimento, entrega e encerramento.
- e) O conceito de software se restringe ao desenvolvimento do código em determinada linguagem e seu armazenamento em arquivos.

Comentários

Na alternativa A temos a definição perfeita e sucinta para Engenharia de Software. Trata-se de um ramo específico para todo o contexto de um software: a necessidade do cliente que gera especificação dos requisitos, o desenvolvimento do software (programação), a validação que confere o cumprimento dos requisitos e a evolução que é o aperfeiçoamento e novas possíveis versões do software.

Gabarito: alternativa A.

9. CEBRASPE (CESPE) - Analista de Controle (TCE-PR)/Tecnologia da Informação/2016

No que diz respeito à engenharia de software, assinale a opção correta.

- a) Um dos princípios de agilidade da Agile Alliance dispõe que a entrega completa de um software garante a satisfação do cliente.
- b) Um projeto desenvolvido mediante XP (extreme programming) segue princípios opostos aos de um projeto implementado com base em KIS (Keep It Simple).
- c) O modelo de desenvolvimento em cascata é utilizado em caso de divergência nos requisitos de um software, para permitir a evolução gradual do entendimento dos requisitos durante a implementação do software.



- d) Durante a fase de levantamento de requisitos para a construção de um software, compete aos desenvolvedores organizar as necessidades em ordem de prioridade.
- e) O QFD (quality function deployment) identifica como requisitos fascinantes os recursos que extrapolam as expectativas dos clientes.

Comentários

O QFD é um método que ajuda a transformar requisitos em funcionalidades que agregam valor ao produto final. Tudo aquilo que excede as expectativas do cliente é considerado requisito fascinante.

A alternativa que responde à questão é a letra E.

Gabarito: alternativa E.

...

Forte abraço e bons estudos!

"Hoje, o 'Eu não sei', se tornou o 'Eu ainda não sei'"

(Bill Gates)

Thiago Cavalcanti



Face: www.facebook.com/profthiagocavalcanti
Insta: www.instagram.com/prof.thiago.cavalcanti
YouTube: youtube.com/profthiagocavalcanti



ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.