

Aula 00

*TRF 2ª Região (Técnico Judiciário -
Informática) Passo Estratégico de
Conhecimentos Específicos*

Autor:

Thiago Rodrigues Cavalcanti

12 de Dezembro de 2023

DESENVOLVIMENTO: QUALIDADE DE SOFTWARE: CONCEITOS BÁSICOS (CMMI VERSÃO 1.3, ABNT NBR ISO/IEC 12207:2009), MPS.BR. ANÁLISE E PROJETO ORIENTADO A OBJETO: UML

Sumário

Apresentação.....	2
O que é o Passo Estratégico?	2
Análise Estatística.....	3
Roteiro de revisão e pontos do assunto que merecem destaque	3
Qualidade de Software	3
Conceitos.....	3
Aspectos Importantes.....	4
Visões de Qualidade.....	4
Garantia da Qualidade de Software	5
Gerenciamento de Qualidade.....	7
Modelos de Qualidade.....	8
Norma ISO/IEC 9126	9
CMMI (versão 2.0)	11
MPS.BR	16
UML – Unified Modeling Language.....	20
Aposta estratégica.....	22
Questões estratégicas	24



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

A análise estatística estará disponível a partir da próxima aula.

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:

Qualidade de Software

Conceitos

De acordo como SWEBOK 3.0¹, a qualidade de software se refere “as características desejadas de produtos de software, a extensão em que um produto de software em particular possui essas características e aos processos, ferramentas e técnicas que são usadas para garantir essas características”. Portanto, ela está diretamente ligada com a qualidade do processo através do qual o software é desenvolvido, portanto, para se

¹ SWEBOK. **Guide to the Software Engineering Body of Knowledge**. Projeto criado pela IEEE para servir como referência em assuntos considerados, de forma geral, como pertinentes a Engenharia de Software. Disponível em: <<https://www.computer.org/web/swebok>>.



ter qualidade em um produto de software é necessário ter um processo de desenvolvimento bem definido, que deve ser documentado e acompanhado.

Nas normas ISO/IEC 9126-1 e ISO/IEC 25010 a definição de qualidade de software é "a capacidade do produto de software em satisfazer as necessidades implícitas e explícitas quando usado em condições específicas".

"Qualidade de software é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais" [Pressman, 1994].

A qualidade de software não pode ser entendida como perfeição. Qualidade é um conceito multidimensional, realizado por um conjunto de atributos, representando vários aspectos relacionados ao produto: desenvolvimento, manutenção e uso. Qualidade é algo factível, relativo, dinâmico e evolutivo, adequando-se ao nível dos objetivos a serem atingidos (SIMÃO, 2002)².

Aspectos Importantes

Diante das definições acima, chegamos a três aspectos importantes. Segundo Crosby³ "A qualidade é a conformidade aos requisitos". Esta definição nos dá direção sobre qual o caminho seguir para julgar a qualidade de um software.

- Os **requisitos de software** são a base a partir da qual a qualidade é medida. A falta de conformidade aos requisitos significa falta de qualidade.
- Padrões especificados definem um conjunto de **critérios de desenvolvimento** que orientam a maneira segundo a qual o software passa pelo trabalho de engenharia. Se os critérios não forem seguidos, o resultado quase que seguramente será a falta de qualidade.
- Existe um conjunto de **requisitos implícitos** que frequentemente não são mencionados na especificação (por exemplo o desejo de uma boa manutenibilidade).

Note que se o software se adequar aos seus requisitos explícitos, mas deixar de cumprir seus requisitos implícitos, a qualidade do software pode ser comprometida.

Visões de Qualidade

Desenvolvedor

Para o desenvolvedor, a qualidade do software está voltada às características internas. Entretanto, a qualidade interna tem que responder às questões externas que o cliente venha a questionar. Por exemplo:

² SIMÃO, R. P. S. **Características de Qualidade para Componentes de Software**. 2002. Dissertação (Mestrado em Informática Aplicada) - Programa de Pós-Graduação em Informática, Fundação Educacional Edson de Queiroz, Universidade de Fortaleza, Fortaleza.

³ Philip Bayard Crosby, uma das mais importantes personalidades da qualidade.



- As funções X estão disponíveis e são executadas eficientemente?
- Funciona corretamente em imprevistos, como, por exemplo, efetuar débito em uma conta com saldo negativo?
- O software é seguro, ou seja, evita que pessoas ou sistemas não autorizados tenham acesso aos dados para modificar?
- É fácil de usar ou requer muito treinamento?
- É fácil de integrar com outros sistemas existentes?
- Aceita trabalhar com arquivos de outros sistemas ou enviar dados para outros sistemas?

Por essas questões, os desenvolvedores não podem menosprezar o papel do usuário/cliente, não podendo se esquecer das necessidades implícitas do uso.

Usuário

Na visão do usuário, seu interesse está no uso do software, na sua funcionalidade, no desempenho e nos efeitos que o uso possa produzir na sua empresa ou organização. Ou seja, o cliente valoriza a qualidade do software de acordo com sua resposta às suas necessidades.

Gerente

Existe ainda a visão do gerente da empresa, onde a qualidade do produto não pode ser desvinculada dos interesses da organização. Os gerentes avaliam aspectos de conformidade em relação aos requisitos dos clientes e desenvolvedores e também aspectos de custo e cronograma.

Garantia da Qualidade de Software

Para se garantir a qualidade de um produto deve ser implementado um conjunto de métodos e técnicas durante seu desenvolvimento. Abaixo vou citar e definir alguns desses métodos e técnicas.

Identificação de Defeitos

Conhecer os conceitos de erro, defeito e falha é o requisito básico para se construir um processo de garantia de qualidade. É preciso saber identificar o problema, antes de investigar sua origem e, desta forma, saber corrigi-lo. Abaixo temos esses conceitos bem definidos:

- **Erro:** Trata-se de um engano de um indivíduo.
- **Defeito:** Deficiência mecânica ou algorítmica que se ativada pode levar a uma falha. Podemos encontrar os seguintes tipos de defeitos:

Informação Estranha	Informação desnecessária.
Fato Incorreto	Informação que não é verdadeira para as condições especificadas.



Inconsistência	Informações conflitantes
Ambiguidade	Informação passível de ter múltiplas interpretações
Omissão	Informação necessária não incluída

- **Falha:** Evento notável onde o sistema viola suas especificações.

Quanto antes um defeito for exposto, menor será o custo para correção e maior a probabilidade de corrigi-lo corretamente e de forma definitiva. Como solução é necessário introduzir atividades de VV&T ao longo de todo o ciclo de desenvolvimento.

Validação, Verificação e Testes (VV&T)

As atividades de VV&T atuam na garantia de qualidade, assegurando que o software cumpre as especificações definidas e atende às necessidades de seus usuários. Abaixo temos as definições:

- **Verificação:** A verificação tem o objetivo de avaliar se o que foi planejado realmente foi realizado. Ou seja, se os requisitos e funcionalidades documentados foram implementados, além disso a verificação também pode ser realizada para especificação de sistemas, para avaliar se os requisitos estão sendo documentados como deveriam e ainda prever falhas ou inconsistências entre requisitos. Normalmente é caracterizada pela pergunta: Fizemos o software corretamente?
- **Validação:** A validação tem o objetivo de avaliar se o que foi entregue atende as expectativas do cliente. Ou seja, se os requisitos, independente do que foi planejado, estão sendo implementados para atender a regra de negócio do cliente, se o sistema é realmente aquilo que o cliente quer e está pagando para ter. A validação final do sistema é realizada pelo próprio cliente ou usuário. Também caracterizada por uma pergunta: Fizemos o software correto?

Os principais métodos de Validação e Verificação são:

- 1) Estática: Revisões de Software (Revisões aos Pares, Inspeções de Software, entre outros). Denomina-se estática, pois não envolve a execução do produto, mas apenas leitura de artefatos, como código.

São processos ou atividades para leitura de um artefato de software visando assegurar que ele cumpre sua especificação e atende às necessidades de seus usuários. Tem por objetivo realizar validação e verificação estática de artefatos de software. Pode ser aplicada a qualquer artefato produzido ao longo do processo de desenvolvimento de software.

Tipos de Revisões:

- Inspeções de Software: mais focadas em encontrar defeitos.
- Walkthroughs: são mais apropriados para atividades de brainstorming, para explorar alternativas de projeto e resolução de problemas.



2) Dinâmica: Testes de Software (Testes de unidade, de integração, do software, entre outros).

Processo de executar um programa ou sistema com o objetivo de revelar a presença de erros; ou, falhando nesse objetivo, aumentar a confiança sobre o programa. Envolve todo um processo de planejamento, fases de teste, etc.

Gerenciamento de Qualidade

O Gerenciamento de Qualidade se importa em definir padrões e procedimentos para o desenvolvimento de software com qualidade.

- **Garantia de qualidade**
 - Focada no processo;
 - Estabelece procedimentos que conduzem a um software de alta qualidade
- **Planejamento de qualidade**
 - Seleciona os procedimentos adequados ao projeto de software específico
- **Controle de qualidade**
 - Focado no produto
 - Assegura que os resultados estão dentro do esperado

A garantia de qualidade é focada no **processo** e tem a preocupação em não inserir defeitos (prevenção) durante a construção do produto, já no controle da qualidade a preocupação é no **produto** final, isto é, deve haver uma inspeção visando a detecção de defeitos no produto seja ele o software ou uma documentação para certificar que os resultados obtidos foram os esperados.

Garantia da Qualidade de Software X Controle da Qualidade de Software

Existe uma confusão entre os conceitos dos termos Controle da Qualidade (*Quality Control*) e Garantia da Qualidade (*Quality Assurance*). Embora usados de maneira errônea em muitos lugares, ambos os termos têm propósitos totalmente diferentes.

Garantia da Qualidade (Quality Assurance)	Controle da Qualidade (Quality Control)
Garantia da qualidade garante que o processo é definido e apropriado.	As atividades de controle da qualidade focam na descoberta de defeitos em i específicos.
Metodologia e padrões de desenvolvimento são exemplos de garantia da qualidade.	Um exemplo de controle da qualidade poderia ser: "Os requisitos definidos são os requisitos certos?".
Garantia da qualidade é orientada a processo.	Controle da qualidade é orientado a produto.
Garantia da qualidade é orientada a prevenção.	Controle da qualidade é orientado a detecção.
Foco em monitoração e melhoria de processo.	Inspeções e garantia de que o produto de trabalho atenda aos requisitos especificados.
As atividades são focadas no início das fases no ciclo de vida de desenvolvimento de software.	As atividades são focadas no final das fases no ciclo de vida de desenvolvimento de software.



Garantia da qualidade garante que você está fazendo certo as coisas e da maneira correta.	Controle da qualidade garante que os resultados do seu trabalho são os esperados conforme requisitos.
---	---

É importante observar que o teste de software é uma das atividades de controle da qualidade, ou seja, o teste de software é orientado a produto e está dentro do domínio do controle da qualidade.

Modelos de Qualidade

Modelos de qualidade de software são estruturas ou abordagens sistemáticas que fornecem diretrizes e critérios para avaliar, medir e melhorar a qualidade do software. Esses modelos definem um conjunto de práticas, padrões, processos e métricas que ajudam as organizações a estabelecer e alcançar padrões de qualidade desejados em seus produtos de software.

Esses modelos têm como objetivo fornecer uma estrutura para identificar, analisar e gerenciar os principais aspectos da qualidade do software, como funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade e segurança. Eles são projetados para auxiliar as organizações a implementar práticas de desenvolvimento de software que levem à entrega de produtos de alta qualidade, que atendam às necessidades e expectativas dos usuários.

Os modelos de qualidade de software geralmente incluem etapas para definir objetivos de qualidade, estabelecer processos adequados, medir a qualidade com base em métricas específicas, realizar análises de gaps e implementar melhorias contínuas nos processos e produtos de software. Eles também podem fornecer um conjunto de critérios para avaliar a conformidade com os padrões de qualidade estabelecidos e obter certificações reconhecidas na indústria.

Além disso, os modelos de qualidade de software podem servir como um guia para as organizações no estabelecimento de uma cultura de qualidade, promovendo a adoção de práticas e abordagens que visam melhorar continuamente os processos de desenvolvimento de software, aumentar a eficiência, reduzir erros e defeitos, e melhorar a satisfação do cliente.

É importante ressaltar que diferentes modelos de qualidade de software podem ter abordagens e ênfases diferentes, e podem ser aplicáveis a diferentes contextos e setores da indústria. Alguns modelos são genéricos e aplicáveis a uma ampla gama de organizações, enquanto outros são específicos para setores ou metodologias de desenvolvimento de software. O objetivo geral, no entanto, é sempre alcançar e manter a qualidade do software em conformidade com os requisitos e expectativas dos usuários finais.

Existem vários modelos de avaliação de qualidade. Alguns dos principais modelos são:

- **CMMI (*Capability Maturity Model Integration*):** O CMMI é um modelo de melhoria de processos que fornece diretrizes para o desenvolvimento e aprimoramento de processos organizacionais. Ele abrange várias áreas, como engenharia de sistemas, desenvolvimento de software e gestão de projetos.



- **ISO 9000:** A família de normas ISO 9000 é uma série de normas internacionais que estabelecem critérios para um sistema de gestão da qualidade. Elas são aplicáveis a diversas indústrias, incluindo o desenvolvimento de software.
- **ISO/IEC 9126:** O modelo ISO/IEC 9126 define um conjunto de características de qualidade de software e fornece diretrizes para avaliar e medir a qualidade do software em termos dessas características. Ele abrange aspectos como funcionalidade, confiabilidade, usabilidade e eficiência.
- **TQM (Total Quality Management):** O TQM é um modelo de gestão de qualidade abrangente que enfatiza a participação de todos os membros da organização na melhoria contínua dos processos e produtos. Embora não seja exclusivamente voltado para o software, muitos de seus princípios podem ser aplicados nesse contexto.
- **Six Sigma:** O Six Sigma é uma metodologia de gestão de qualidade que visa reduzir defeitos e variações nos processos. Ele se baseia em técnicas estatísticas para análise e melhoria dos processos, visando atingir um nível de qualidade próximo à perfeição.

Vamos nos aprofundar nos modelos: **ISO/IEC 9126** e **CMMI (Capability Maturity Model Integration)**.

Norma ISO/IEC 9126

O modelo proposto pela ISO/IEC 9126 (NBR 13596) tem como objetivo servir de referência básica na avaliação de produto de software.

A ISO/IEC em sua primeira parte, descreve um modelo de qualidade do produto de software, composto por duas partes: **qualidade interna e externa e qualidade de uso**, como pode ser observado na tabela abaixo.

Tipo de Qualidade	Descrição
Qualidade Interna e Externa	Especifica seis características para qualidade interna e externa, as quais por sua vez são subdivididas em sub características que são manifestadas externamente, quando o software é utilizado como parte de um sistema computacional, e são resultantes de atributos internos do software.
Qualidade de uso	Especifica quatro características de qualidade em uso, mas não apresenta o modelo de qualidade em uso além do nível de característica, que é para o usuário, o efeito combinado das seis características de qualidade do produto de software.

Na tabela abaixo temos o modelo de propósito geral da qualidade interna e externa que define as seis amplas categorias de características de qualidade de software. Estas características estão relacionadas diretamente com os requisitos não funcionais esperados de um software.

Características	Sub Características	Significado
-----------------	---------------------	-------------



Funcionalidade (como as funções e propriedades específicas do produto, satisfazem as necessidades do usuário)	Adequação	Existência de um conjunto de funções apropriadas para as tarefas requeridas;
	Acurácia	Produção de resultados ou efeitos corretos;
	Interoperabilidade	Habilidade de interação do produto de software com outros produtos;
	Segurança de acesso	Aptidão para evitar acessos não autorizados a programas e dados;
	Conformidade	O produto está de acordo com as convenções, as normas ou os regulamentos estabelecidos.
Confiabilidade (como o produto de software é capaz de manter seu nível de desempenho, ao longo do tempo, nas condições estabelecidas)	Maturidade	Estado de maturação do software, detectada por sua baixa frequência de falhas;
	Tolerância a falhas	O nível de desempenho é mantido, quando ocorrem falhas;
	Recuperabilidade	Existem mecanismos que restabelecem e restauram os dados após a ocorrência de falhas.
Usabilidade (o esforço necessário para a utilização do sistema, baseado em um conjunto de implicações e de condições do usuário)	Inteligibilidade	Facilidade de entendimento dos conceitos utilizados no produto de software;
	Apreensibilidade	Facilidade de aprendizado do software;
	Operacionalidade	Faculdade de operar e controlar operações pertinentes ao software.
Eficiência (como os recursos e os tempos envolvidos são compatíveis com o nível de desempenho requerido pelo software)	Comportamento em relação aos recursos	Relaciona-se com a quantidade dos recursos empregados.
	Comportamento em relação ao tempo	Refere-se ao tempo de resposta de processamento.
Manutenibilidade (refere-se ao esforço necessário para a realização de alterações específicas, no produto de software)	Analísibilidade	Característica de ser possível diagnosticar deficiências e causas de falhas;
	Modificabilidade	Característica que o produto deve ter de forma a facilitar modificações e remoções de defeitos;
	Estabilidade	Ausência de riscos ou ocorrências de defeitos inesperados no software;



	Testabilidade	Facilidade de o produto ser testado.
Portabilidade (facilidade do software poder ser transferido de um ambiente para outro)	Adaptabilidade	Faculdade de o produto poder ser adaptado a novos ambientes;
	Capacidade para ser instalado	Facilidade de instalação do produto de Software Conformidade com padrões;
	Capacidade para substituir	O produto de software pode ser substituído por outro, sem grandes esforços.
	Conformidade	O produto está segundo os padrões ou convenções de portabilidade.

A tabela abaixo apresenta as quatro características definidas na qualidade de uso.

Características
Eficácia: capacidade do produto de software de permitir que seus usuários atinjam metas especificadas com acurácia e completitude, em um contexto de uso especificado.
Produtividade: capacidade do produto de software de permitir que seus usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida, em um contexto de uso especificado.
Segurança: capacidade do produto de software de apresentar níveis aceitáveis de riscos de danos a pessoas, negócios, software, propriedades ou ao ambiente, em um contexto de uso especificado.
Satisfação: capacidade do produto de satisfazer usuários, em um contexto de uso especificado.

CMMI (versão 2.0)

CMMI (*Capability Maturity Model Integration*) é um modelo de melhoria de processos que fornece diretrizes e práticas para aprimorar a qualidade, eficiência e desempenho dos processos de uma organização. O CMMI 2.0, lançado em 2018, é a versão atualizada e aprimorada do modelo, que traz diversas melhorias em relação à versão anterior.

O CMMI 2.0 foi projetado para ser mais simples, flexível e adaptável às necessidades das organizações modernas. Ele possui um foco maior em resultados de negócios e em atender aos objetivos estratégicos das organizações, além de fornecer orientações mais claras para a implementação e melhoria dos processos.

Uma das principais mudanças do CMMI 2.0 é a estrutura modular, que permite que as organizações escolham os módulos relevantes para suas necessidades específicas. Os módulos abordam diferentes áreas, como desenvolvimento de produtos, serviços, aquisição e gestão de pessoas, proporcionando uma abordagem mais personalizada para a melhoria dos processos.

Além disso, o CMMI 2.0 incorpora conceitos ágeis e práticas de DevOps, reconhecendo a importância dessas abordagens no ambiente de desenvolvimento de software atual. Ele também inclui orientações sobre práticas de segurança cibernética, permitindo que as organizações abordem de forma mais eficaz os desafios relacionados à segurança da informação.



O CMMI 2.0 também introduz uma abordagem de avaliação mais flexível, permitindo que as organizações realizem avaliações de forma mais incremental e frequentemente. Isso ajuda a identificar oportunidades de melhoria de forma mais ágil e a monitorar o progresso ao longo do tempo.

No CMMI 2.0, as áreas de foco são organizadas em duas categorias principais: ***Categorias e Áreas de Capacidade***.

As Categorias representam os grupos mais amplos de áreas de foco no CMMI 2.0. Elas fornecem uma visão geral dos diferentes aspectos da melhoria de processos e qualidade. As categorias no CMMI 2.0 são:

Doing (Executando): Esta categoria engloba as áreas relacionadas à execução e entrega dos produtos e serviços. Ela se concentra nas atividades operacionais e técnicas necessárias para desenvolver e entregar produtos e serviços de alta qualidade.

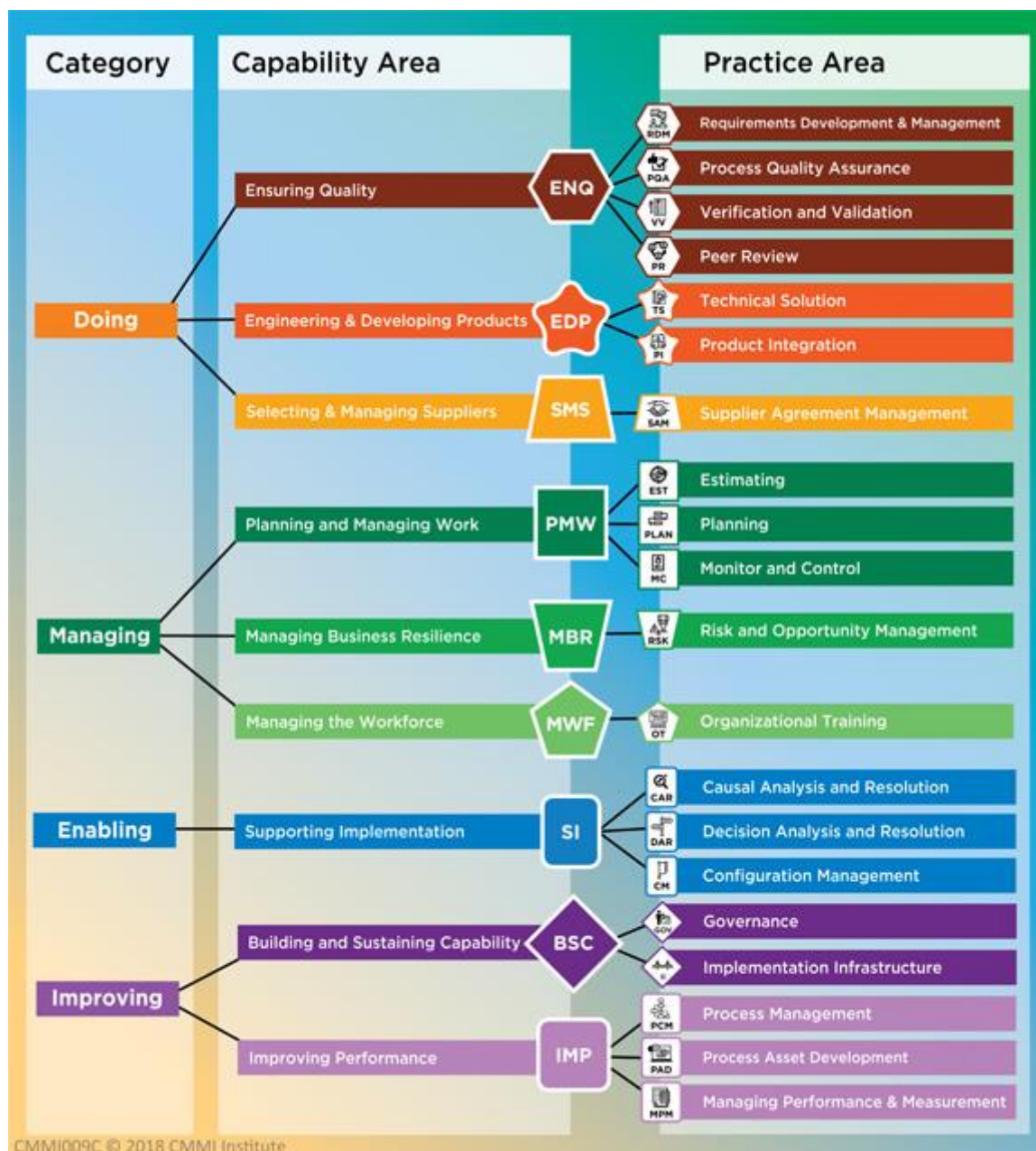
Managing (Gerenciando): Esta categoria abrange as áreas relacionadas à gestão de projetos e processos. Ela se concentra em práticas e técnicas para planejar, executar e monitorar projetos, bem como gerenciar riscos, custos, cronogramas e recursos.

Enabling (Habilitando): Esta categoria trata das áreas relacionadas ao suporte e habilitação das atividades de desenvolvimento e entrega. Ela engloba práticas e processos que fornecem o ambiente e os recursos necessários para facilitar o trabalho eficaz das equipes de projeto.

Improving (Melhorando): Esta categoria abrange as áreas relacionadas à melhoria contínua dos processos e da organização. Ela se concentra em práticas e técnicas para avaliar, medir e melhorar os processos, bem como promover uma cultura de aprendizado e inovação.

Essas quatro categorias são projetadas para cobrir uma ampla gama de áreas de foco e fornecer uma estrutura abrangente para a melhoria de processos e qualidade em uma organização. Cada categoria contém várias áreas de capacidade específicas, que são as áreas detalhadas de foco dentro de cada categoria.





Estrutura Modular

A estrutura modular do CMMI 2.0 é uma das principais características que o diferencia das versões anteriores do modelo. Ela permite que as organizações selecionem e implementem os módulos específicos que são relevantes para suas necessidades e objetivos.

O CMMI 2.0 é composto por um conjunto de módulos independentes, cada um abordando uma área específica de foco. Esses módulos representam diferentes disciplinas e práticas de negócio, permitindo que as organizações personalizem sua implementação do modelo de acordo com suas necessidades. Os módulos podem variar de acordo com a edição do CMMI 2.0 utilizada, mas abaixo estão alguns dos módulos mais comuns encontrados no modelo:

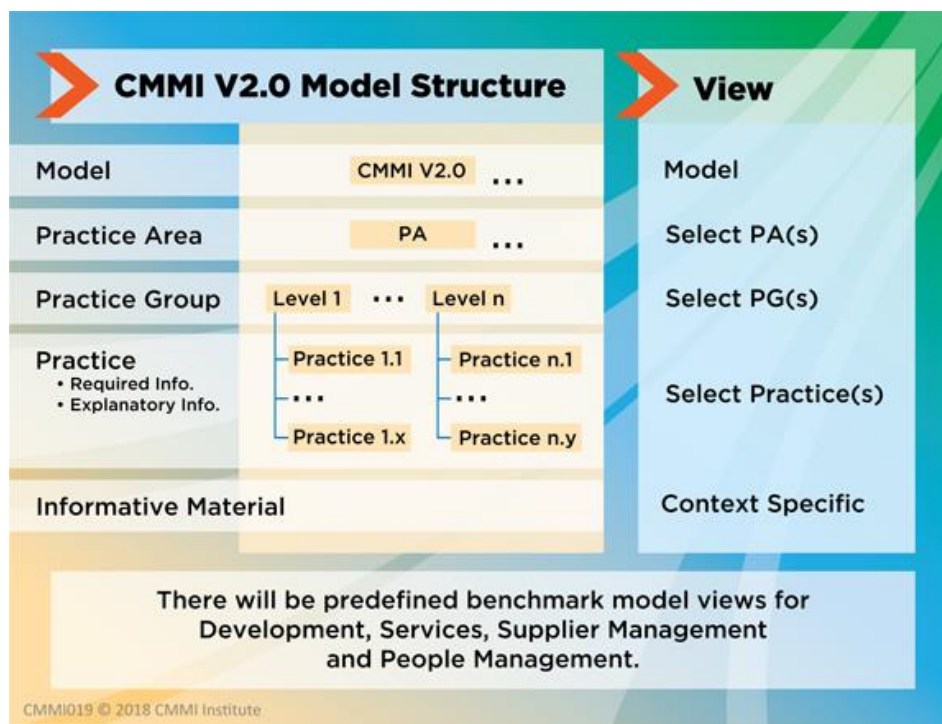


- **Desenvolvimento de Produtos:** Este módulo se concentra nas melhores práticas para o desenvolvimento de produtos, incluindo engenharia de sistemas, engenharia de software, desenvolvimento de hardware, entre outros. Ele abrange áreas como gerenciamento de requisitos, design, codificação, teste, integração e entrega de produtos.
- **Serviços:** O módulo de Serviços aborda as melhores práticas para a gestão de serviços, especialmente na área de Tecnologia da Informação (TI). Ele fornece orientações para a definição, planejamento, entrega e suporte de serviços de qualidade, incluindo gerenciamento de incidentes, gerenciamento de problemas, gerenciamento de mudanças e gerenciamento de níveis de serviço.
- **Aquisição:** Este módulo trata das práticas relacionadas à aquisição de produtos e serviços de fornecedores externos. Ele inclui diretrizes para seleção de fornecedores, contratação, negociação de contratos, gestão de relacionamento com fornecedores e avaliação do desempenho dos fornecedores.
- **Gerenciamento de Processos:** O módulo de Gerenciamento de Processos abrange a gestão de processos organizacionais, incluindo o estabelecimento, documentação, implementação, medição e melhoria contínua dos processos. Ele fornece orientações para a definição de processos eficazes, monitoramento do desempenho dos processos e implementação de melhorias.
- **Gerenciamento de Pessoas:** Este módulo concentra-se na gestão de pessoas e recursos humanos. Ele aborda áreas como definição de papéis e responsabilidades, avaliação de competências, desenvolvimento profissional, retenção de talentos e promoção de um ambiente de trabalho colaborativo.

Cada módulo do CMMI 2.0 possui seus próprios objetivos, práticas e resultados esperados. As organizações podem escolher os módulos que são relevantes para suas atividades e prioridades, permitindo uma implementação mais focada e direcionada às necessidades específicas da organização.

Essa estrutura modular dá às organizações flexibilidade para adaptar o CMMI 2.0 às suas necessidades individuais, evitando a aplicação de todo o modelo de forma rígida. Isso permite que as organizações priorizem as áreas mais críticas para a melhoria dos processos e escolham as práticas mais relevantes para sua indústria e contexto operacional.





Níveis de maturidade

No CMMI 2.0, existem cinco níveis de maturidade que representam diferentes estágios de desenvolvimento e capacidade organizacional. Cada nível é definido por um conjunto específico de práticas e metas que uma organização deve atingir para demonstrar seu nível de maturidade. Aqui estão os cinco níveis de maturidade do CMMI 2.0:

Level 1: *Managed* (Gerenciado)

No nível 1, as práticas básicas de gerenciamento de projetos são estabelecidas para garantir que as atividades sejam planejadas, executadas e controladas. A organização é capaz de atingir metas específicas de custo, cronograma e qualidade.

Level 2: *Defined* (Definido)

No nível 2, a organização estabelece processos padronizados e gerencia suas atividades de acordo com esses processos definidos. Os projetos são executados de forma consistente e são baseados em práticas estabelecidas.

Level 3: *Quantitatively Managed* (Gerenciado Quantitativamente)

No nível 3, a organização utiliza métricas e análises quantitativas para gerenciar seus processos. Os processos são monitorados e controlados de forma proativa, com base em dados e informações coletados.

Level 4: *Continuously Improving* (Melhoria Contínua)



No nível 4, a organização implementa uma abordagem de melhoria contínua. Os processos são analisados regularmente e ajustes são feitos para melhorar sua eficácia e eficiência.

Level 5: *Predictable* (Previsível)

No nível 5, a organização atingiu um alto nível de maturidade, onde os processos são previsíveis e altamente controlados. A organização possui a capacidade de ajustar e otimizar seus processos para atingir metas específicas e se adaptar a mudanças nas necessidades do negócio.



Cada nível de maturidade representa um estágio progressivo de desenvolvimento organizacional, onde as práticas e capacidades são aprimoradas ao longo do tempo. A conquista de cada nível requer um compromisso contínuo com a melhoria de processos e a cultura organizacional voltada para a qualidade.

MPS.BR

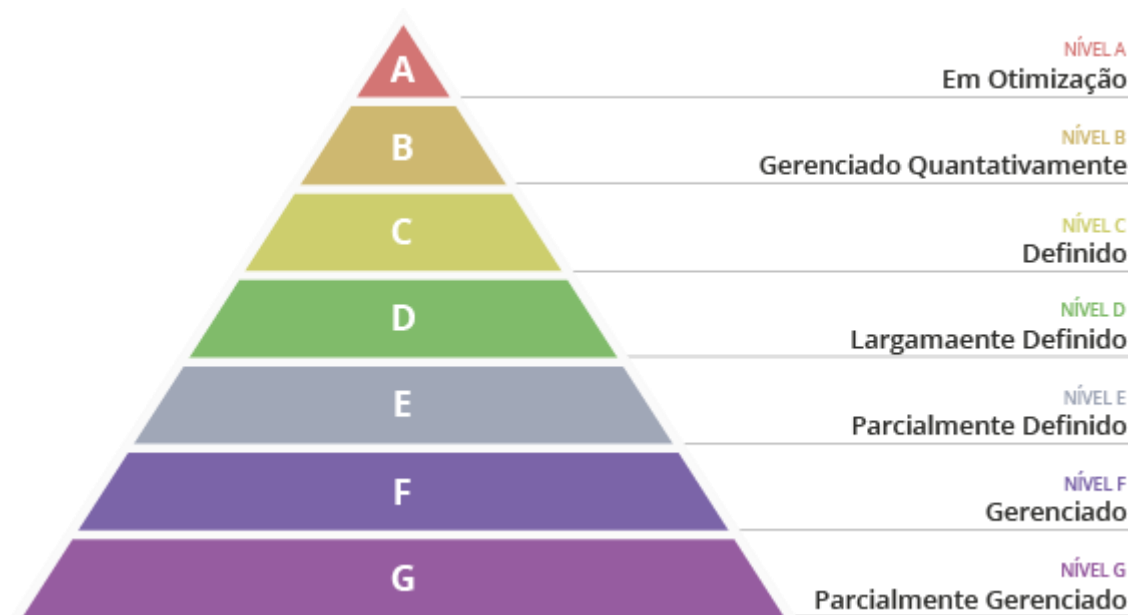
O modelo MPS.BR foi criado em 2003 pela Softex (Associação para Promoção da Excelência do Software Brasileiro), com o objetivo de melhorar a capacidade de desenvolvimento de softwares por empresas brasileiras.

O MPS.BR funciona como um selo que indica o nível de maturidade da empresa em relação às práticas relacionadas ao desenvolvimento de software. Esse selo funciona como uma garantia que a empresa utiliza as boas práticas e possui condições de desenvolver um software de qualidade, dentro dos custos e prazos estabelecidos. O objetivo é ajudar as organizações a aprimorar a qualidade de seus processos de software e, consequentemente, a qualidade de seus produtos e serviços relacionados.



Entre as metas do MPS.BR temos a que visa definir e aprimorar um modelo, reconhecido nacional e internacionalmente, de melhoria e avaliação de processo de software, visando preferencialmente as micro, pequenas e médias empresas, com o objetivo de atender as necessidades do negócio. Também são estabelecidos um processo e um método de avaliação, que dá sustentação e garante que o MPS.BR está sendo empregado de forma coerente às suas definições.

O MPS 2021 é baseado em 7 níveis de maturidade, que vão do nível G (nível inicial) ao nível A (nível mais avançado). Cada nível de maturidade é definido por um conjunto de processos de software que precisam ser implementados e gerenciados pela organização. Além dos níveis de maturidade, ele é baseado em 16 áreas de processo, que abrangem as principais atividades relacionadas ao desenvolvimento de software, como gerenciamento de requisitos, gerenciamento de projetos, garantia da qualidade, entre outras.



- **Nível G: Parcialmente Gerenciado**

O nível de maturidade G é o primeiro nível a ser implementado do modelo. Este nível é composto por dois processos:

Gerência de Projetos (GPR): O propósito é estabelecer e manter planos que determinam as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desalinhamentos significativos no desempenho do projeto. Devem ser implementados 19 resultados esperados, neste nível.

Gerência de Requisitos (GRE): O intuito é gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto. Devem ser implementados 5 resultados esperados.

- **Nível F: Gerenciado**



O nível de maturidade F é o segundo nível do modelo, sendo aderente ao nível 2 do CMMI-DEV. Este nível é composto pelos processos do nível anterior (GPR e GRE) e por mais cinco processos:

Aquisição (AQU): O propósito é gerenciar a aquisição de produtos que satisfaçam às necessidades expressas pelo adquirente. Devem ser implementados 8 resultados esperados.

Gerência de Configuração (GCO): O propósito é estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos. Devem ser implementados 7 resultados esperados.

Garantia da Qualidade (GQA): O propósito é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos, procedimentos e padrões estabelecidos. Devem ser implementados 4 resultados esperados.

Gerência de Portfólio de Projetos (GPP): O propósito é iniciar e manter projetos que sejam necessários, suficientes e sustentáveis, de forma a atender os objetivos estratégicos da organização. Devem ser implementados 8 resultados esperados.

Medição (MED): O propósito é coletar, armazenar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais. Devem ser implementados 7 resultados esperados.

- **Nível E: Parcialmente Definido**

No nível de maturidade E entram os processos organizacionais. É composto pelos processos dos níveis anteriores (GPR, GRE, AQU, GCO, GQA, GPP e MED), pelos resultados evoluídos do Gerência de Projetos (3 resultados esperados) e por mais quatro processos:

Avaliação e Melhoria do Processo Organizacional (AMP): O propósito é determinar o quanto os processos padrão da organização contribuem para alcançar os objetivos de negócio da organização e para apoiar a organização a planejar, realizar e implantar melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos. Devem ser implementados 10 resultados esperados.

Definição do Processo Organizacional (DFP): O propósito é estabelecer e manter um conjunto de ativos de processo organizacional e padrões do ambiente de trabalho usáveis e aplicáveis às necessidades de negócio da organização. Devem ser implementados 8 resultados esperados.

Gerência de Recursos Humanos (GRH): O propósito é prover a organização e os projetos com os recursos humanos necessários e manter suas competências adequadas às necessidades do negócio. Devem ser implementados 11 resultados esperados.

Gerência de Reutilização (GRU): O propósito é gerenciar o ciclo de vida dos ativos reutilizáveis. Devem ser implementados 5 resultados esperados.

- **Nível D: Largamente Definido**



O nível de maturidade D entram os processos da engenharia de software. É composto pelos processos dos níveis anteriores (GPR, GRE, AQU, GCO, GQA, GPP, MED, AMP, DFP, GRH, GRU) e por mais cinco processos:

Desenvolvimento de Requisitos (DRE): O propósito é definir os requisitos do cliente, do produto e dos componentes do produto. Devem ser implementados 7 resultados esperados.

Integração do Produto (ITP): O propósito é compor os componentes do produto, produzindo um produto integrado consistente com seu projeto, e demonstrar que os requisitos funcionais e não-funcionais são satisfeitos para o ambiente alvo ou equivalente. Devem ser implementados 9 resultados esperados.

Projeto e Construção do Produto (PCP): O propósito é projetar, desenvolver e implementar soluções para atender aos requisitos. Devem ser implementados 8 resultados esperados.

Validação (VAL): O propósito é confirmar que um produto ou componente do produto atenderá a seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido. Devem ser implementados 7 resultados esperados.

Verificação (VER): O propósito é confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados. Devem ser implementados 6 resultados esperados.

- **Nível C: Definido**

O nível de maturidade C é composto pelos processos dos níveis de maturidade anteriores (GPR, GRE, AQU, GCO, GQA, GPP, MED, AMP, DFP, GRH, GRU, DRE, ITP, PCP, VAL, VER), por mais três processos:

Desenvolvimento para Reutilização (DRU): O propósito é identificar oportunidades de reutilização sistemática de ativos na organização e, se possível, estabelecer um programa de reutilização para desenvolver ativos a partir de engenharia de domínios de aplicação. Devem ser implementados 9 resultados esperados.

Gerência de Decisões (GDE): O propósito é analisar possíveis decisões críticas usando um processo formal, com critérios estabelecidos, para avaliação das alternativas identificadas. Devem ser implementados 7 resultados esperados.

Gerência de Riscos (GRI): O propósito é identificar, analisar, tratar, monitorar e reduzir continuamente os riscos em nível organizacional e de projeto. Devem ser implementados 9 resultados esperados.

- **Nível B: Gerenciado Quantitativamente**

O nível de maturidade B é composto pelos processos dos níveis de maturidade anteriores (GPR, GRE, AQU, GCO, GQA, GPP, MED, AMP, DFP, GRH, GRU, DRE, ITP, PCP, VAL, VER) e pelos resultados evoluídos do Gerência de Projetos (7 resultados esperados). Neste nível a implementação dos processos deve atender aos atributos de processo dos níveis anteriores.

- **Nível A: Em Otimização**



O nível de maturidade A é composto pelos processos dos níveis de maturidade anteriores (G ao B). Não possui processos específicos do nível. Neste nível a implementação dos processos deve atender aos atributos de processo dos níveis anteriores.

UML – Unified Modeling Language

A necessidade de uma técnica-padrão que visa cobrir todas as etapas de desenvolvimento de um sistema de informação. O espectro que vai da análise de requisitos, passando pela modelagem, projeto, implementação até chegar a implantação. Uma destas técnicas é conhecida como UML – *Unified Modeling Language*. Ela oferece um mecanismo na forma de notação diagramática e sintaxe de linguagem associada para cobrir todo o ciclo de vida.

A UML combina conceitos comumente aceitos de muitos métodos e metodologias orientados a objetos (O-O). Ela é genérica independente de linguagem e plataforma. A UML tem muitos tipos de diagramas que podem ser divididos em duas categorias:

Diagramas estruturais. Estes descrevem os relacionamentos estruturais ou estáticos entre objetos de esquema, objetos de dados, componentes de software. Incluem os diagramas de classes, de objetos, de componentes e de implementação.

Diagramas comportamentais. Sua finalidade é descrever o comportamento ou relacionamento dinâmico entre componentes. Incluem diagramas de caso de uso, de sequência, de colaboração, de estados e de atividades.

Pense que um sistema tem aspectos estáticos e dinâmicos, esses dois tipos de diagrama descrevem essas características. Vamos dar um exemplo de cada um dos tipos de diagrama apenas para ajudar você na fixação do conteúdo. O **Diagrama de Classes** oferece um ótimo exemplo do tipo de diagrama estrutural e fornece um conjunto inicial de elementos de notação que todos os outros diagramas de estrutura usam. O propósito do diagrama de classes é mostrar os tipos que estão sendo modelados no sistema.

Uma classe é representada na forma de um retângulo, contendo duas linhas que separam 3 partes. A primeira contém no nome da classe, a segunda os atributos da classe e a última os métodos da mesma. Vejamos uma figura que ajude a entender melhor esses conceitos:



Concurseiro
nome: String
concursoAlvo: String
identidade: Integer
dataDeNascimento: Date
getConcursoAlvo() : String

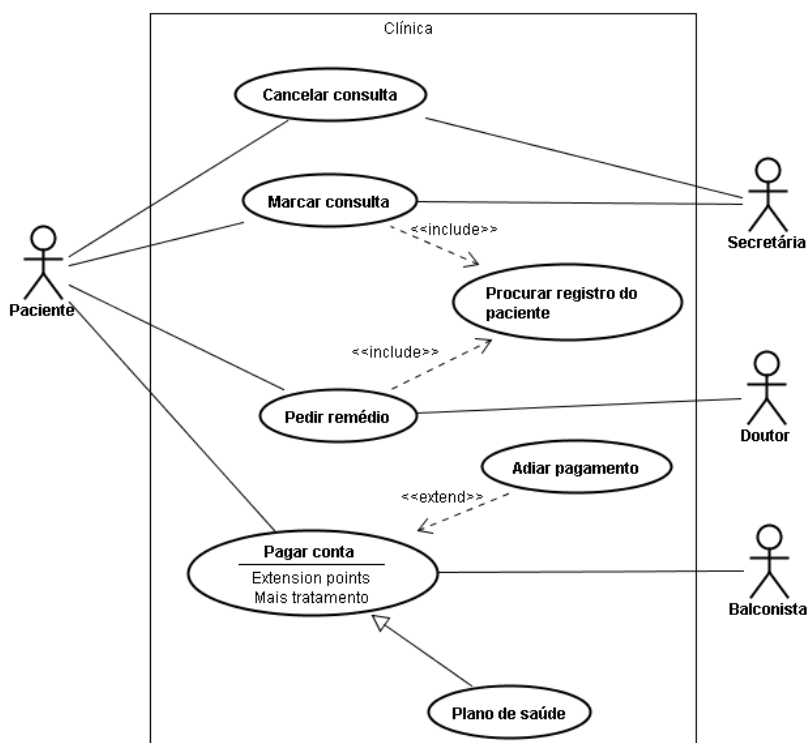


```
getIdade(): Integer
```

Observe a classe acima, temos a classe Concurseiro com os atributos nome, concursoAlvo, indetidade e dataDeNascimento. Veja que cada atributo tem um tipo de dado associado. Por fim, o retângulo mostra as operações que podem ser executadas com os dados desta classe, conhecidos como métodos. Um detalhamento maior sobre o que seria um diagrama de classe pode ser visto neste *artigo*⁴.

O diagrama comportamental que usaremos como exemplo é o diagrama de Casos de Uso. O diagrama de casos de uso tem o objetivo de **auxiliar a comunicação entre os analistas e o cliente**. Ele descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. O cliente deve ver no diagrama de casos de uso as principais funcionalidades de seu sistema.

O diagrama de Caso de Uso é representado por atores, casos de uso e relacionamentos entre estes elementos. Um **ator** é representado por um boneco e um rótulo com o nome do ator. Um ator é um usuário do sistema, que pode ser um usuário humano ou um outro sistema computacional. Um **caso de uso** é representado por uma elipse e um rótulo com o nome do caso de uso. Um caso de uso define uma grande função do sistema. Os relacionamentos ajudam a descrever casos de uso. A figura abaixo descreve um diagrama de caso de uso. Nela temos os atores Paciente, Secretária, Doutor e Balconista que fazem acesso a diferentes macro funcionalidades do sistema, ou seja, os casos de uso.



Antes de você pergunte o que são os termos <<include>> e <<extends>> na figura acima, deixa eu tentar explicar de uma forma bem simples. O <<include>> afirma que um caso de uso depende do outro, tente

⁴ <https://www.ibm.com/developerworks/br/rational/library/content/RationalEdge/sep04/bell/index.html>



observar na figura, para marcar uma consulta é necessário ter os dados do paciente. Já o <<extends>> inclui uma funcionalidade extra que não é obrigatória, por exemplo, adiar pagamento é uma possibilidade na ação de pagar conta.

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 jurisprudenciais⁵.



Características	Sub Características	Significado
Funcionalidade (como as funções e propriedades específicas do produto, satisfazem as necessidades do usuário)	Adequação	Existência de um conjunto de funções apropriadas para as tarefas requeridas;
	Acurácia	Produção de resultados ou efeitos corretos;
	Interoperabilidade	Habilidade de interação do produto de software com outros produtos;
	Segurança de acesso	Aptidão para evitar acessos não autorizados a programas e dados;
	Conformidade	O produto está de acordo com as convenções, as normas ou os regulamentos estabelecidos.
Confiabilidade (como o produto de software é capaz de manter seu nível de desempenho,	Maturidade	Estado de maturação do software, detectada por sua baixa frequência de falhas;
	Tolerância a falhas	O nível de desempenho é mantido, quando ocorrem falhas;

⁵ 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.



ao longo do tempo, nas condições estabelecidas)	Recuperabilidade	Existem mecanismos que restabelecem e restauram os dados após a ocorrência de falhas.
Usabilidade (o esforço necessário para a utilização do sistema, baseado em um conjunto de implicações e de condições do usuário)	Inteligibilidade	Facilidade de entendimento dos conceitos utilizados no produto de software;
	Apreensibilidade	Facilidade de aprendizado do software;
	Operacionalidade	Faculdade de operar e controlar operações pertinentes ao software.
Eficiência (como os recursos e os tempos envolvidos são compatíveis com o nível de desempenho requerido pelo software)	Comportamento em relação aos recursos	Relaciona-se com a quantidade dos recursos empregados.
	Comportamento em relação ao tempo	Refere-se ao tempo de resposta de processamento.
Manutenibilidade (refere-se ao esforço necessário para a realização de alterações específicas, no produto de software)	Analísibilidade	Característica de ser possível diagnosticar deficiências e causas de falhas;
	Modificabilidade	Característica que o produto deve ter de forma a facilitar modificações e remoções de defeitos;
	Estabilidade	Ausência de riscos ou ocorrências de defeitos inesperados no software;
	Testabilidade	Facilidade de o produto ser testado.
Portabilidade (facilidade do software poder ser transferido de um ambiente para outro)	Adaptabilidade	Faculdade de o produto poder ser adaptado a novos ambientes;
	Capacidade para ser instalado	Facilidade de instalação do produto de Software Conformidade com padrões;
	Capacidade para substituir	O produto de software pode ser substituído por outro, sem grandes esforços.
	Conformidade	O produto está segundo os padrões ou convenções de portabilidade.

Imprima o capítulo Aposta Estratégica separadamente e dedique um tempo para absolver tudo o que está destacado nessas duas páginas. Caso tenha alguma dúvida, volte ao Roteiro de Revisão e Pontos do Assunto que Merecem Destaque. Se ainda assim restar alguma dúvida, não hesite em me perguntar no fórum.



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.

Um mecanismo comum de garantia da qualidade adotado por muitas organizações é o de revisões de software, que podem ser aplicadas ao longo do ciclo de vida de desenvolvimento de software. Sobre revisões de software é correto afirmar que

- A) podem ser aplicadas apenas nas etapas iniciais do ciclo de desenvolvimento visando identificar falhas não percebidas pelo gerente no planejamento do projeto.
- B) as revisões são capazes de detectar e eliminar grande percentual de erros, reduzindo substancialmente o custo das atividades seguintes do ciclo de desenvolvimento.
- C) tem um custo muito elevado, o que, em geral, pode não compensar o investimento realizado.
- D) devem ser realizadas após o software ter sido construído, porque o software pronto é o produto mais importante de um projeto de desenvolvimento.
- E) podem ser aplicadas em diferentes estágios do ciclo de desenvolvimento, porém com benefício menor quanto mais cedo se estiver neste ciclo de desenvolvimento.

Comentários

As revisões de software são utilizadas para encontrar erros durante o processo de desenvolvimento, de modo a não se tornarem defeitos depois da liberação do software. A descoberta precoce de erros, evita que sejam propagados para a próxima etapa na gestão da qualidade. Segundo Pressman, diversos estudos e análise sobre o tema indicam que as atividades de projeto introduzem de 50 a 60% de todos os erros durante a gestão da qualidade. Entretanto, técnicas de revisão demonstraram ser até 75% eficazes na descoberta de falhas no projeto.

Portanto, a alternativa correta é a letra B.

Gabarito: alternativa B.



2.

Considere que uma organização está no nível de maturidade G do modelo MR-MPS-SW, do guia MPS.BR. Este nível é composto por 2 processos. Esta organização

- a) encontra-se no nível Parcialmente Definido, o mais alto da escala, compatível com o nível de maturidade 5 do CMMI-DEV versão 1.3.
- b) deve cumprir o atributo de processo cujo resultado da sua implementação completa é: os processos que estão alinhados a objetivos quantitativos de negócio são identificados.
- c) deve cumprir o atributo de processo cujo resultado da sua implementação completa é: técnicas para análise dos dados coletados são selecionadas.
- d) deve implementar o processo Garantia da Qualidade, cujo propósito é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos e padrões estabelecidos.
- e) deve implementar o processo Gerência de Requisitos, cujo propósito inclui gerenciar os requisitos do produto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.

Comentários

No MPS.BR, os níveis de maturidade são avaliados a partir de atributos de processo (AP). Cada atributo de processo é detalhado por um ou mais resultados esperados (RAP). O atributo de processo AP 1.1 (o processo é executado) evidencia o quanto o processo atinge seu propósito. Este atributo possui apenas a descrição do resultado esperado RAP 1, que traz o texto:

- a) existe uma política organizacional estabelecida e mantida para o processo.
- b) a execução do processo é planejada.
- c) a execução do processo é monitorada e ajustes são realizados.
- d) o processo atinge seus resultados definidos.
- e) as informações e os recursos necessários para a execução do processo são identificados e disponibilizados.

Portanto, a alternativa correta é a letra E.

Gabarito: alternativa E.

3.

Um técnico trabalha em uma organização que atingiu o nível de maturidade 4 do CMMI (quantitativamente gerenciado). Para atingir este nível, todas as áreas de processo dos níveis de



maturidade anteriores e as áreas de processo do nível de maturidade atual precisam atingir o nível de capacidade

- a) 5 (avançado).
- b) 2 (gerenciado).
- c) 3 (realizado).
- d) 3 (definido).
- e) 4 (otimizado).

Comentários

Relembrando que os Níveis de Maturidade (CMMI v. 1.3 - Abordagem por estágios) são:

Nível 1: Inicial

Nível 2: Gerenciado

Nível 3: Definido

Nível 4. Quantitativamente gerenciado

Nível 5: Em otimização

Portanto, alternativa correta, letra D.

Gabarito: alternativa D.

4.

O nível G de maturidade do MR-MPS-SV é composto pelos processos Entrega de Serviços, Gerência de Incidentes, Gerência de Nível de Serviço, Gerência de Requisitos e Gerência de Trabalhos. Neste nível, a implementação dos processos deve satisfazer os atributos de processo:

- a) AP 1.1 (o processo é executado) e AP 2.1 (o processo é gerenciado).
- b) AP 1.1 (o processo é medido) e AP 2.1 (o processo é definido).
- c) AP 1.1 (o processo está implementado) e AP 2.1 (o processo é otimizado continuamente).
- d) AP 1.1 (o processo é objeto de melhorias incrementais e inovações) e AP 2.1 (o processo é otimizado continuamente).
- e) AP 1.1 (o processo é definido) e AP 2.1 (o processo está implementado).

Comentários



Na aula vimos os atributos de processos (AP) relacionados com os níveis de maturidade. Entre eles temos os que estão relacionados ao nível G de maturidade:

AP 1.1 - O processo é executado (Nível G) - o processo realiza o que foi proposto para ele.

AP 2.1 - O processo é gerenciado (Nível G e F) - a execução do processo é gerenciada.

AP 2.2 - Os produtos de trabalho do processo são gerenciados (Nível G e F) - os produtos de trabalho realizados pelo processo são gerenciados de forma adequada.

Entre as alternativas, a única que satisfaz essa relação é a letra A.

Gabarito: alternativa A.

5.

O modelo de qualidade de software CMMI, versão 1.3,

- a) é largamente utilizado pelos Tribunais que utilizam o Scrum, pois o modelo MPS.BR conflita com as metodologias ágeis de software.
- b) adota a ITIL v3 como padrão de gerenciamento da qualidade de serviços de TI.
- c) utiliza a representação contínua para permitir que a organização atinja níveis de capacidade.
- d) não atende às necessidades das grandes empresas brasileiras, por isso foi criado o modelo MPS.BR.
- e) foi modificado em 2010 para manter compatibilidade com a Norma NBR ISO/IEC 12207:2009.

Comentários

O modelo de qualidade de software CMMI, em sua versão 1.3, **utiliza as representações contínua e por estágio para permitir que a organização atinja níveis de capacidade** e maturidade respectivamente. Ademais, o modelo MPS.BR **NÃO conflita com as metodologias ágeis de software**, sobretudo com o **SCRUM**, devendo a organização realizar algumas adaptações nas práticas do Scrum para serem completamente aderentes ao MPS.BR. O CMMI **NÃO** adota a ITIL v3 como padrão de gerenciamento da qualidade de serviços de TI. Além disso, **atende às necessidades das grandes empresas brasileiras**, apesar de ser mais caro que o modelo MPS.BR. Por fim, vale lembrar que o modelo modificado em 2010 para manter compatibilidade com a Norma NBR ISO/IEC 12207:2009 foi o **MPS.BR**, e não o CMMI.

Portanto, a alternativa que responde corretamente a questão é a letra C.

Gabarito: alternativa C.



6.

O modelo de qualidade de software MPS.BR

- a) não é compatível com a Norma NBR ISO/IEC 12207:2009.
- b) adota a ITIL v3 como padrão de gerenciamento da qualidade de serviços de TI.
- c) não pode ser aplicado em nenhum Tribunal Regional do Trabalho.
- d) não atende às necessidades das empresas brasileiras, que preferem a adoção do CMMI.
- e) não conflita com as metodologias ágeis de software, podendo ser utilizado junto com o Scrum.

Comentários

Analisando as alternativas temos:

- a) Vimos na aula que uma das bases é esta norma. ERRADA.
- b) Não adota nenhum padrão desta natureza. ERRADA.
- c) Pode ser aplicado em qualquer organização. ERRADA.
- d) Foi criado pensando nas empresas brasileiras, tanto que o MPS.BR é acrônimo de “Melhoria de Processos do Software Brasileiro”. ERRADA.
- e) CORRETA.

Gabarito: alternativa E.

7.

Considere que um Analista de TI foi designado para atuar em duas áreas de processo do CMMI-DEV v1.3 definidas no nível de maturidade 3, que visam demonstrar que um produto ou componente de um produto satisfaz seu uso pretendido quando colocado no ambiente alvo e garantir que produtos selecionados satisfaçam aos requisitos especificados para ele. O Analista atuará nas áreas de processo

- a) Desenvolvimento de Requisitos e Medição e Análise.
- b) Verificação e Gerenciamento de Configuração.
- c) Medição e Análise e Validação.
- d) Validação e Verificação.
- e) Garantia de Qualidade e Validação.

Comentários

Na aula vimos que as áreas do nível 3 de maturidade do CMMI são:



- Desenvolvimento de Requisitos
- Solução Técnica
- Integração de Produto
- Verificação e Validação
- Foco nos Processos da Organização
- Definição dos Processos da Organização
- Treinamento na Organização
- Gestão Integrada de Projeto
- Gestão de Riscos
- Análise e Tomada de Decisões

Portanto, a alternativa correta é a letra D.

Gabarito: alternativa D.

8.

De acordo com o Modelo de Referência MPS para Software (MR-MPS-SW), alguns processos podem ser excluídos do escopo de uma avaliação MPS, total ou parcialmente, por não serem pertinentes ao negócio da unidade organizacional que está sendo avaliada. Cada exclusão deve ser justificada no Plano de Avaliação. A aceitação das exclusões e suas justificativas é responsabilidade do Avaliador Líder. É permitida a exclusão completa do seguinte processo do nível de maturidade F, desde que não executado pela organização,

- a) Desenvolvimento para Reuso – DRE.
- b) Aquisição – AQU.
- c) Integração do Produto – ITP.
- d) Gerência de Reúso – GRE.
- e) Medição de Riscos – MRI.

Comentários

De acordo com o Guia Geral MPS.BR, os processos que podem ser excluídos são:

Aquisição (AQU) nível F - Desde que não executados pela organização.

Gerência de Portfólio de Projetos (GPP) nível F - Desde que a única atividade da unidade organizacional seja evolução de produto.

Desenvolvimento para Reutilização (DRU) nível C - Desde que seguidas as orientações da tabela de oportunidade x capacidade do guia geral.

Portanto, alternativa correta, letra B.



Gabarito: alternativa B.

9.

Considere, por hipótese, que o Tribunal Regional Eleitoral está adotando as práticas do Modelo de Referência MPS para Software (MR-MPS-SW) do guia MPS.BR. O Tribunal já atingiu um nível de maturidade no qual os processos Gerência de Projetos e Gerência de Requisitos foram implantados, bem como os atributos de processo indicados. Em busca de mais um nível de maturidade, o Tribunal está implantando o processo Garantia da Qualidade, visando assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos, procedimentos e padrões estabelecidos pela equipe de analistas. Desta forma, o Tribunal pretende atingir o nível de maturidade

- a) D – Parcialmente Definido.
- b) E – Gerenciado.
- c) E – Parcialmente Definido.
- d) F – Parcialmente Gerenciado.
- e) F – Gerenciado.

Comentários

Para responder essa questão vamos relembrar os níveis de maturidade do MPS.BR.

- A (Em Otimização);
- B (Gerenciado Quantitativamente);
- C (Definido);
- D (Largamente Definido);
- E (Parcialmente Definido);
- F (Gerenciado); e
- G (Parcialmente Gerenciado).

Ou seja, a única alternativa que encaixa perfeitamente com um dos níveis é a letra E.

Gabarito: alternativa E.

10.

O Modelo de Referência MPS para Software (MR-MPS-SW) define níveis de maturidade que são uma combinação entre processos e sua capacidade. Considere:

I. é a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros.

II. está relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade.



III. expressa o grau de refinamento e institucionalização com que o processo é executado na organização ou na unidade organizacional.

IV. é representada por um conjunto de atributos de processo, dentre os quais se encontra "O processo é objeto de análise quantitativa".

Refere-se à capacidade do processo o que se afirma em

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

Comentários

Note que essa questão traz 4 itens, e todos são relacionados aos termos e definições do MPS.BR.

Na aula vimos que a capacidade do processo no MPS possui os seguintes atributos de processos (AP):

- 1) **AP 1.1 - O processo é executado** (Nível G) - o processo realiza o que foi proposto para ele;
- 2) **AP 2.1 - O processo é gerenciado** (Nível G e F) - a execução do processo é gerenciada;
- 3) **AP 2.2 - Os produtos de trabalho do processo são gerenciados** (Nível G e F) - os produtos de trabalho realizados pelo processo são gerenciados de forma adequada;
- 4) **AP 3.1 - O processo é definido** (Níveis E, D e C) - existe um padrão e que o mesmo apoia a implementação do processo;
- 5) **AP 3.2 - O processo está implementado** (Níveis E, D e C) - o processo, padronizado, é de fato implementado para atingir os seus objetivos;
- 6) **AP 4.1 - O processo é medido** (Nível B) - algumas medições são usadas para assegurar que o desempenho do processo ajuda a alcançar os objetivos para o qual este processo foi proposto;
- 7) **AP 4.2 - O processo é controlado** (Nível B) - o processo é controlado estatisticamente, permitindo se ter previsibilidade, estabilidade e capacidade de execução;
- 8) **AP 5.1 - O processo é o objeto de inovações** (Nível A) - as mudanças no processo são identificadas a partir da análise dos seus indicadores e da investigação de possíveis inovações;
- 9) **AP 5.2 - O processo é otimizado continuamente** (Nível A) - as mudanças no processo têm, de fato, impacto no alcance dos objetivos, no que tange os aspectos relevantes de melhoria do mesmo.

Portanto, a alternativa correta é a letra E.

Gabarito: alternativa E.



Considere que o TRT decidiu implantar as melhores práticas de qualidade de software de acordo com o modelo de referência MRMPS-SW do guia MPS.BR. Um Técnico indicou, corretamente, que o Tribunal deveria iniciar pelo nível

- a) 0 e, à medida que evoluir nos níveis de capacidade, um maior nível de maturidade para desempenhar os processos devem ser atingidos.
- b) de maturidade A e este nível não possui processos específicos.
- c) de maturidade G, que é composto pelos processos Gerência de Projetos e Gerência de Requisitos.
- d) de maturidade G e este nível não possui processos específicos.
- e) 1 e, à medida que evoluir nos níveis de maturidade, um maior nível de capacidade para desempenhar os processos devem ser atingidos.

Comentários

O nível de maturidade G é composto pelos processos gerência de projetos e gerência de requisitos e tem, entre outros propósitos, estabelecer e manter planos que definam as atividades, os recursos e as responsabilidades do projeto. Portanto, a alternativa C é a resposta para nossa questão.

Gabarito: alternativa C.

12.

Considere, por hipótese, que o TRE-SP esteja em busca de uma certificação de qualidade de software. Antes da escolha, uma equipe de Analistas realizou uma comparação relativa ao tratamento da Garantia da Qualidade entre o MR-MPS-SW e o CMMIDEV 1.3. Este comparativo indica, corretamente, que o

- a) CMMI-DEV exige a identificação, registro e comunicação dos problemas e das não conformidades, relacionados à avaliação de processos e produtos.
- b) MR-MPS-SW só exige o estabelecimento de registros das atividades de Garantia de Qualidade, que é parte do exigido no CMMI-DEV.
- c) escalonamento das ações corretivas para níveis superiores exigido pelo CMMI-DEV, não é exigido no MR-MPS-SW, o que pode deixar a solução das não conformidades fragilizadas.
- d) MR-MPS-SW exige que a avaliação da aderência dos produtos de trabalho seja realizada sempre antes da entrega ao cliente externo, bem como em marcos do projeto.
- e) CMMI-DEV exige que a avaliação da aderência dos produtos de trabalho seja realizada apenas antes da entrega a um cliente interno.

Comentários

Essa questão é um dos absurdos cobrados pela banca. Ela exige que você decore os 4 pontos da gerência de qualidade. Relembrando, eles são:



GQA 1. A aderência dos produtos de trabalho aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente, antes dos produtos serem entregues e em marcos predefinidos ao longo do ciclo de vida do projeto;

GQA 2. A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente;

GQA 3. Os problemas e as não-conformidades são identificados, registrados e comunicados;

GQA 4. Ações corretivas para as não-conformidades são estabelecidas e acompanhadas até as suas efetivas conclusões. Quando necessário, o escalamento das ações corretivas para níveis superiores é realizado, de forma a garantir sua solução;

Portanto a alternativa correta é letra D.

Gabarito: alternativa D.

...

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.