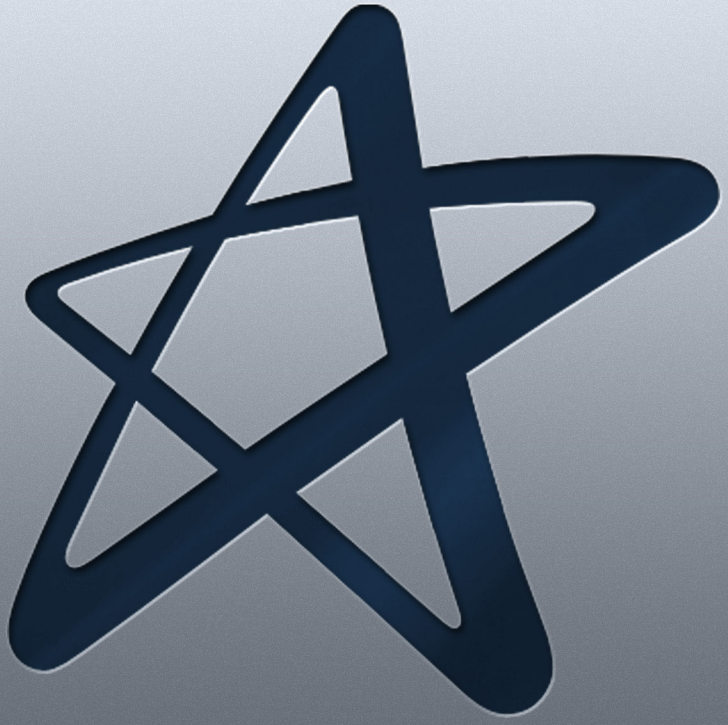
Qualidade de Software



Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

Material Teórico

CMM - Capability Maturity Model (Modelo de Maturidade em Capacitação)

Responsável pelo Conteúdo:
Prof. Ms. Douglas Almendro

Revisão Textual:
Profa. Ms. Selma Aparecida Cesarin.

UNIDADE

CMM - Capability Maturity Model (Modelo de Maturidade em Capacitação)

- Introdução
- CMM
- Níveis do CMM
- Chaves do Processo de Produção
- Entendendo os níveis de maturidade
- Métricas de Qualidade de Software
- Sistemas de Gerenciamento de Qualidade

Objetivo de APRENDIZADO

Nesta Unidade, será explanado o conceito de CMM. Este passo nos levará ao entendimento da certificação dada ao *software* e que é utilizada no mercado. Então, não perca nenhum detalhe dessa Unidade.

Estamos em nossos estudos sobre CMM. A ideia principal é mostrar a descrição deste modelo de maturidade e os principais elementos de um processo de desenvolvimento de *software*.

Veremos os estágios de maturidade por que passam as organizações enquanto evoluem no seu ciclo de desenvolvimento de *software*, por meio de avaliação contínua, identificação de problemas e ações corretivas.

Este caminho de melhoria é definido por cinco níveis de maturidade, que estudaremos em nossas aulas.

Contextualização

O CMM fornece às organizações uma orientação sobre como adquirir controle do processo de desenvolvimento de software e como avançar para uma cultura padrão na gestão de *software*.

O objetivo principal é acompanhar, por meio dos níveis de maturidade, a realização de um processo controlado e mensurado que tem como fundamento a melhoria contínua.

A cada nível de maturidade corresponde um conjunto de práticas de *software* e de gestão específicas, denominadas áreas-chave do processo (KPAs – *Key Process Areas*).

Por este estudo, teremos uma boa visão, principalmente sobre como melhor gerir os processos de desenvolvimento de *software*.

Introdução

Já temos em mente o que vem a ser qualidade, mas, para reforçarmos o conceito, observe como o dicionário Aurélio define qualidade: "propriedade, atributo ou condição das coisas ou das pessoas capaz de distingui-las das outras e de lhes determinar a natureza" (Aurélio, 1986).

Como um atributo de um item, a qualidade se refere àquilo que pode ser medido, ou seja, comparado com padrões conhecidos, tais como tamanho, cor, propriedades elétricas, maleabilidade etc.

Entretanto, é mais difícil categorizarmos qualidade em *software*, que é uma entidade intelectual, do que em objetos físicos.

Vamos começar a nossa Unidade explicando a que nos referimos quando falamos de qualidade de *software*, principalmente no quesito de processos.

Mas o que seria processo?

Conjunto de atos por que se realiza uma operação qualquer (química, farmacêutica, industrial etc.): um processo de fabricação de nitroglicerina. / Sequência contínua de fatos que apresentam certa unidade, ou que se reproduzem com certa regularidade; andamento, desenvolvimento: o processo de uma crise econômica (Aurélio, 1986).

Para a engenharia de *software*, seria um conjunto de passos de procedimentos parcialmente ordenados, relacionados a artefatos, pessoas, recursos, estruturas organizacionais e restrições, tendo como objetivo produzir e manter o produto de *software* requisitado.

O CMM ajuda a apontar quais são os procedimentos a serem exigidos para que tenhamos excelência na atividade.

CMM

Vamos falar, agora, sobre o CMM.

O CMM surgiu durante a década de 1980, como um modelo para avaliação de risco na contratação de empresas de *software* pelo Departamento de Defesa dos Estados Unidos, que desejava ser capaz de avaliar os processos de desenvolvimento utilizados pelas empresas que concorriam em licitações como indicação da previsibilidade da qualidade, custos e prazos nos projetos contratados.

Para desenvolver esse processo, o DOD constituiu, junto a Carnegie-Mellon University, o SEI (*Software Engineering Institute*), que, além de ser responsável pela evolução da família CMM, também realiza diversas outras pesquisas em engenharia de *software*.

Observe os dados cronológicos:

- **1986 –** início do desenvolvimento de um modelo de maturidade de processo, para ajudar as organizações a melhorarem seus processos de *software* (por solicitação do governo federal);

- **Junho 1987 –** liberação de breve descrição do modelo de maturidade de processo de *software*;
- **Setembro 1987 –** versão preliminar do questionário de maturidade 1991 – 1ª. versão do CMM (Versão 1.0);
- **1993 –** depois de 5 anos de experiência, o modelo de maturidade evoluiu para um modelo completamente definido, usando conhecimento adquirido das avaliações de processo de *software* e de extensivo retorno das indústrias e do governo CMM – *Capability Maturity Model for Software* (atualmente usada).

Premissa básica que está por baixo do trabalho do SEI sobre maturidade de processo é "A qualidade de um *software* produto é profundamente determinada pela qualidade do processo de desenvolvimento e de manutenção usado para construí-lo".

O SEI desenvolveu um modelo de 5 níveis que orienta uma organização em como "amadurecer" seus processos de *software*.

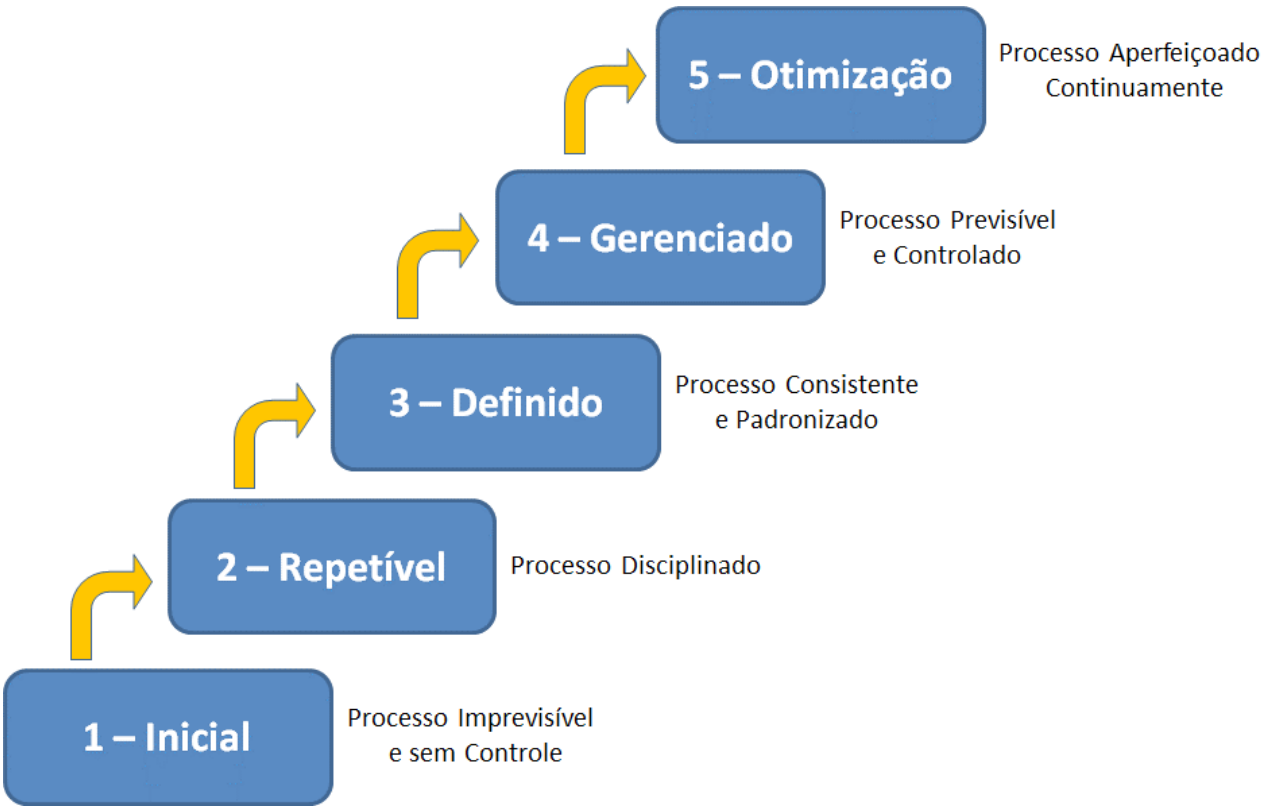
O modelo descreve um caminho evolucionário, que vai de um processo indisciplinado para um processo disciplinado.

Sem a disciplina descrita no modelo, programas de melhoria podem mostrar-se ineficientes porque os fundamentos necessários para apoiar os melhoramentos sucessivos não foram estabelecidos.

Os 5 níveis de maturidade descrevem fundamentos sucessivos para melhoria contínua do processo e definem uma escala ordinal para medir a maturidade de processo de uma organização.

As vantagens dos níveis de maturidade é que eles fornecem prioridades claras, que orientam a seleção de algumas atividades de melhoramento que serão muito úteis, se implementadas imediatamente.

Isso é importante porque a maioria das organizações podem focalizar somente algumas poucas atividades de melhoramento de cada vez.



Níveis do CMM

O CMM fornece às organizações orientação sobre como ganhar controle do processo de desenvolvimento de *software* e como evoluir para uma cultura de excelência na gestão de *software*.

O objetivo principal nas transições desses níveis de maturidade é a realização de um processo controlado e mensurado como a fundação para melhoria contínua.

Cada nível de maturidade possui um conjunto de práticas de *software* e gestão específicas, denominadas áreas-chave do processo.

Estas devem ser implantadas para a organização atingir o nível de maturidade em questão.

O nível de maturidade é um estágio evolutivo bem definido, em busca de um processo de *software* maduro. Cada nível de maturidade fornece uma gama de fundamentos para a melhoria contínua do processo e um conjunto de práticas de *software* e gestão específicas, denominado áreas-chave do processo, que devem ser implantadas para a organização atingir o nível de maturidade em questão.

Cada nível compreende um conjunto de objetivos de processos que, quando satisfeitos, estabilizam um componente importante do processo de *software*. Alcançando cada nível da estrutura de maturidade, estabelecem-se diferentes componentes no processo, resultando em um crescimento na capacidade processual da organização.

A maturidade do processo de *software* é a extensão para a qual um processo específico é explicitamente definido, gerenciado, medido, controlado e efetivado. A maturidade representa o potencial de crescimento e indica a riqueza do processo de *software* da organização e a consistência com que o mesmo é aplicado em todos os seus projetos.

Em uma organização madura, o processo de *software* é bem compreendido – o que geralmente é feito por meio de documentação e treinamento – e está sendo continuamente monitorado e melhorado pelos seus usuários.

A atividade de um processo de *software* maduro é conhecida e implica que a produtividade e a qualidade resultantes do processo possam ser continuamente melhoradas por meio de ganhos consistentes na disciplina alcançada com a sua utilização.

Quando uma organização obtém ganhos na maturidade de um processo de *software*, ela o institucionaliza por meio de políticas, padrões e estruturas organizacionais.

A institucionalização exige a construção de uma infraestrutura e de uma cultura corporativa que possa dar suporte aos métodos, práticas e procedimentos de negócio, que perdurem após possíveis afastamentos daqueles que originalmente os definiram.

Vamos ver o que possuem os níveis de maturidade do CMM.

Nível 1: Inicial

No nível 1 de maturidade, os processos normalmente são ad hoc e a organização geralmente não dispõe de ambiente estável. O sucesso nestas organizações depende da competência e do heroísmo dos funcionários e não do uso de processos estruturados. Devido ao imediatismo, um ambiente caótico, o nível 1 de maturidade raramente produz um produto ou serviço que funcione. Assim, frequentemente, eles excedem o orçamento e o prazo em seus projetos.

Nível 2: Repetível

No nível 2 de maturidade, o desenvolvimento do *software* é repetido. O processo pode não se repetir para todos os projetos da organização, que pode usar ferramentas de Gerência de Projetos para mapear os custos e o prazo do projeto.

A adoção de um processo de desenvolvimento ajuda a garantir que práticas existentes são utilizadas em momentos de stress. Quando estas práticas são adotadas, os projetos decorrem e são gerenciados de acordo com o planejamento inicial.

O status do projeto e os serviços entregues são visíveis ao gerenciamento (por exemplo: é possível a visualização de marcos do projeto e o término da maioria das tarefas).

Técnicas de gerenciamento de projetos são estabelecidas para mapear custos, prazos e funcionalidades. Um mínimo de disciplina nos processos é estabelecido para que se possa repetir sucessos anteriores em projetos com escopo e aplicação similar. Ainda há um risco significante de exceder os custos e estimativas de prazo de desenvolvimento.

Nível 3: Definido

A organização possui um conjunto de processos padrão, que são a base do nível 3 e estão estabelecidos e melhorados periodicamente. Estes processos padrão são usados para estabelecer uma consistência dentro da organização e os projetos estabelecem seus processos definidos pelo conjunto de padrões processuais da organização.

O gerenciamento da organização estabelece os objetivos dos processos, baseado no conjunto de padrões pré-definidos e garante que estes objetivos sejam encaminhados de forma apropriada.

Uma crítica distinção entre os níveis 2 e 3 é o escopo dos padrões, descrição dos processos e procedimentos. No nível 2, os padrões, descrições de processos e procedimentos podem ser bem diferentes em cada instância específica do processo (por exemplo, em um projeto particular). No nível 3, os padrões, descrições de processo e procedimentos para o projeto são guiados pelo conjunto padrão de processos da organização.

Nível 4: Gerenciado

Utilizando métricas precisas, o gerenciamento pode efetivamente controlar os esforços para desenvolvimento de *software*. Em particular, o gerenciamento pode identificar caminhos para ajustar e adaptar o processo para projetos particulares, sem perda de métricas de qualidade ou desvios das especificações.

Organizações neste nível conseguem metas quantitativas para o processo de desenvolvimento de *software* e de manutenção.

Nível 5: Otimizado

O nível de maturidade 5 foca no contínuo aumento do desempenho dos processos por meio de melhorias de inovação tecnológica e incremental. Objetivos de melhoria quantitativa dos processos para a organização são estabelecidos, continuamente revisados, refletindo os objetivos da organização e usando critérios de gerência de processos.

Os efeitos da melhoria da revisão dos processos são medidos e acompanhados, utilizando-se processos de melhoria de qualidade. Ambos – os processo definidos e o conjunto de processos padrão da organização – são alvos de melhoria de métricas.

Chaves do Processo de Produção

Cada nível, com exceção do primeiro, é composto por áreas chave de processo. Essas áreas são formadas por sessões, chamadas áreas comuns. As áreas comuns especificam as práticas chave necessárias para se alcançar a principal meta da área chave do processo, que são uma descrição do caminho que uma organização deve tomar para se tornar madura. Este caminho foi baseado em anos de experiência, tanto no processo de produção, quanto no gerenciamento do processo.

As áreas chaves de processo do nível repetível são definidas como :

» **Gerenciamento de requisitos:** Os requisitos do sistema são controlados de forma a estabelecer um perfil mínimo a ser utilizado pela engenharia de *software* e pela administração. Os planos, produtos e atividades do *software* são sempre consistentes com os requisitos de sistema;

» **Planejamento do Projeto:** Estimativas relativas ao *software* são documentadas para uso no planejamento e acompanhamento do projeto do *software*. As atividades de projeto de *software* e compromissos assumidos são planejadas e documentadas;

» **Visão geral e acompanhamento do projeto:** Resultados reais são acompanhados de acordo com o planejamento do *software*. Quando os resultados apresentam um significativo desvio do planejamento do *software*, são tomadas ações corretivas, que são acompanhadas até o final do projeto. Mudanças nos compromissos assumidos são feitas em comum acordo com os grupos e indivíduos afetados;

» **Gerenciamento de subcontratados:** O contratante seleciona subcontratos qualificados. Ele os subcontratados estão de acordo no que diz respeito aos compromissos assumidos um com o outro e mantém uma comunicação constante. O contratante acompanha os resultados reais do subcontratado, de acordo com os compromissos assumidos;

» **Garantia da qualidade do *software*:** As atividades de garantia de qualidade de *software* são planejadas. A conformidade dos produtos de *software* e atividades com os padrões, procedimentos e requisitos é verificada objetivamente. Os grupos e indivíduos afetados são informados das atividades de garantia de qualidade de *software* e de seus resultados. Questões relacionadas a não conformidade que não são resolvidas dentro do projeto de *software* são encaminhadas à gerência geral;

» **Gerenciamento de configuração:** As atividades de gerenciamento de configuração são planejadas. Os produtos de trabalho de *software* são identificados, controlados e estão disponíveis. Mudanças nos produtos de trabalho identificados são controlados. Os grupos e pessoas afetadas são informados da situação atual e projetada dos produtos de trabalho de *software*.

As áreas chaves de processo do nível definido são definidas como:

» **Foco do processo organizacional:** São coordenadas atividades de desenvolvimento e melhoramento do processo de *software* em toda a organização. Os pontos fortes e fracos do processo de desenvolvimento de *software* utilizado são identificados, de acordo com um padrão de processo. São planejadas atividades de desenvolvimento e melhoramento do processo em nível de organização;

» **Definição do processo organizacional:** O processo padrão de desenvolvimento de *software* da organização é desenvolvido e mantido. A informação relacionada ao uso do processo padrão de desenvolvimento de *software* é coletada, revisada e disponibilizada;

» **Programa de treinamento:** As atividades de treinamento são planejadas.

» É fornecido treinamento para o desenvolvimento de habilidades e conhecimentos necessários para realizar o gerenciamento do *software* e as funções técnicas, tanto para o grupo de engenharia de *software*, quanto para outros relacionados ao desenvolvimento;

» **Gerenciamento de *software* integrado:** O processo de *software* definido para o projeto é uma versão adaptada do processo padrão de desenvolvimento de *software* da organização. O projeto é planejado e gerenciado de acordo com este padrão;

» **Engenharia de produto de *software*:** As atividades de engenharia de *software* são definidas, integradas e consistentemente realizadas para produzir o *software*;

» **Coordenação intergrupos:** Os grupos de engenharia identificam, acompanham e resolvem todas as questões intergrupos. Todos os grupos de trabalho afetados concordam com os requisitos do cliente e com os acordos entre os grupos de engenharia;

» **Revisão conjunta:** Atividades de revisão conjunta são planejadas. Defeitos nos produtos de trabalho são identificados e removidos.

As áreas chave de processo do nível gerenciado são definidas como:

» **Gerenciamento quantitativo dos processos:** As atividades de gerenciamento quantitativo dos processos são planejadas. O desempenho do processo de desenvolvimento de *software* definido é controlada quantitativamente. A capacidade do processo de desenvolvimento de *software* padrão da organização é conhecida em termos quantitativos;

» **Gerenciamento da qualidade de *software*:** As atividades de gerenciamento da qualidade de *software* do projeto são planejadas. Objetivos mensuráveis da qualidade do produto de *software* e suas prioridades são definidos.

» O progresso real em direção à realização dos objetivos de qualidade para os produtos de *software* é quantificado e gerenciado.

As áreas chave de processo do nível otimizado são definidas como:

» **Prevenção de defeitos:** As atividades de prevenção de defeitos são planejadas. As causas comuns de defeitos são procuradas, identificadas e sistematicamente eliminadas;

» **Gerenciamento de mudanças tecnológicas:** A incorporação de mudanças tecnológicas é planejada. Novas tecnologias são avaliadas para determinar seu efeito na qualidade e na produtividade e as adequadas são incorporadas na prática normal de toda a organização;

» **Gerenciamento de mudanças no processo:** O melhoramento contínuo do processo é planejado. Toda a organização participa das atividades de melhoramento do processo de *software*. O padrão de processo de *software* da organização e os processos de *software* de cada projeto definido são melhorados continuamente.

Entendendo os níveis de maturidade

O CMM é um modelo descritivo, no sentido de descrever atributos essenciais (ou chave) que seriam esperados para caracterizar uma organização em um nível particular de maturidade.

É um modelo normativo, no sentido de que as práticas detalhadas caracterizam os tipos normais de comportamento que seriam esperados em uma organização que desenvolve projetos em larga escala num contexto de contratação governamental.

A intenção é que o CMM tenha um nível suficiente de abstração que não restrinja desnecessariamente a maneira que o processo de *software* é implementado pela organização; ele simplesmente descreve o que normalmente seria esperado dos atributos essenciais do processo de *software*.

Em qualquer contexto em que o CMM for aplicado, deveria ser utilizada uma interpretação razoável das práticas. O CMM deve ser interpretado apropriadamente, utilizando-se o conhecimento de peritos quando o ambiente comercial da organização difere significativamente de uma grande organização fornecedora.

O CMM não é prescritivo; ele não diz à organização como melhorar. O CMM descreve a organização em cada nível de maturidade sem prescrever os meios específicos para consegui-lo. Pode levar vários anos para se passar do Nível 1 para o Nível 2; a movimentação entre os outros níveis geralmente levará cerca de dois anos.

A melhoria de processo de *software* ocorre dentro do contexto dos planos estratégicos e dos objetivos de negócio da organização, da sua estrutura organizacional, das tecnologias em uso, da sua cultura social e sistema de gestão.

O CMM está voltado para os aspectos de processo da Gestão da Qualidade Total; a melhoria de processo bem sucedida implica que os aspectos fora do escopo de processo de *software* também sejam encaminhados (por exemplo: as questões pessoais envolvidas nas mudanças da cultura organizacional que possibilitem a implementação e a institucionalização das melhorias de processo).

O que o CMM não cobre

O CMM não é uma bala de prata [BROOKS, 1987] e não trata todos os assuntos que são importantes para o sucesso dos projetos. Por exemplo, o CMM não indica peritos em domínios específicos de aplicações, não defende tecnologias específicas de *software*, nem sugere como selecionar, contratar, motivar e reter pessoas competentes.

Apesar de esses assuntos serem cruciais para o sucesso do projeto, alguns deles têm sido analisados em outros contextos [CURTIS, 1990]. Entretanto, eles não têm sido integrados ao CMM.

O CMM foi especificamente desenvolvido para prover uma estrutura ordenada e disciplinada dentro da qual possa se encaminhar assuntos de processo de gestão e desenvolvimento de *software*.

Métricas de Qualidade de Software

Um elemento chave de qualquer processo de engenharia é a medição. Nós usamos medidas para melhor entendermos os atributos dos modelos que criamos e, o mais importante, é que nós usamos medidas para avaliarmos a qualidade dos produtos de engenharia ou sistemas que nós construímos.

Ao contrário de outras engenharias, a engenharia de *software* não é baseada em leis quantitativas básicas, medidas absolutas não são comuns no mundo do *software*.

Ao invés disso, nós tentamos derivar um conjunto de medidas indiretas que levam a métricas que fornecem uma indicação de qualidade de alguma representação do *software*.

Embora as métricas para *software* não sejam absolutas, elas fornecem uma maneira de avaliar qualidade por meio de um conjunto de regras definidas.

Garantia de qualidade de software (SQA)

Um dos desafios críticos para qualquer programa de qualidade é tornar possível que qualquer pessoa possa fazer revisões no trabalho de pessoas experientes.

Gerentes querem os melhores projetistas para projetar o produto, então, em geral, SQA não pode tê-los. A necessidade é concentrar esforços em métodos de SQA que permitam que o desenvolvimento possa ser revisado por pessoas que não são desenvolvedores.

O papel de SQA é monitorar os métodos e os padrões que os engenheiros de *software* usam e verificar se eles estão usando apropriadamente seus conhecimentos. Pessoas podem ser experientes em SQA sem, no entanto, serem experientes em projeto de *software*.

Atividades de garantia de qualidade de Software

Em SQA, temos uma variedade de tarefas, que podem se dividir em dois grandes grupos: os engenheiros de *software*, que fazem o trabalho técnico, e o grupo de SQA, que tem responsabilidades no plano de qualidade, na inspeção, na conservação de registros históricos, na análise e no reporting das atividades de SQA para o gerente do projeto.

Os engenheiros de software buscam qualidade de *software*, aplicando sólidos métodos e medidas, conduzindo revisões técnicas formais e realizando testes de *software* bem planejados.

O papel do grupo de SQA é ajudar o time de engenharia de *software* a alcançar um produto de alta qualidade.

O Instituto de Engenharia de *Software* (SEI) recomenda as seguintes atividades a serem realizadas por um grupo de SQA:

- Preparar um plano de SQA para o projeto. O plano é desenvolvido durante o planejamento do projeto e é revisado por todas as partes interessadas;
- Participar do desenvolvimento da descrição do processo do projeto do *software*. O time de engenharia de *software* seleciona um processo para o trabalho a ser realizado. O grupo de SQA revisa a descrição do processo, verificando se ele está de acordo com a política organizacional, os padrões internos para o *software*, os padrões impostos externamente e outras partes do plano de projeto de *software*;
- Revisar as atividades dos engenheiros de *software* para verificar se o processo de *software* definido está sendo seguido. O grupo de SQA identifica, documenta e trilha os desvios do processo e verifica quais correções devem ser realizadas;
- Garantir que os desvios no trabalho do *software* e nos produtos do trabalho são documentados e mantidos de acordo com o procedimento documentado;
- Registrar qualquer discordância e reportar para o gerente.

Além dessas atividades, o grupo de SQA coordena o controle e o gerenciamento de mudanças e ajuda a coletar e analisar métricas de *software*.

Sistemas de Gerenciamento de Qualidade

Personal Software Process (PSP)

O estímulo original para desenvolver o PSP surgiu de questões sobre o Capability Maturity Model (CMM). Muitos viam o CMM como projetado para grandes organizações e não entendiam como ele poderia ser aplicado a trabalhos individuais e em times pequenos de projeto.

Apesar do CMM poder ser aplicado para ambas, pequenas e grandes organizações, uma orientação mais específica se tornava claramente necessária.

Após alguns anos de pesquisa, 12 das 18 áreas chave de processo do CMM foram adaptadas para o trabalho de engenheiros de *software* individuais.

O principal objetivo do PSP é fazer com que os engenheiros de *software* fiquem atentos no processo que eles usam para fazer seus trabalhos e estejam sempre verificando suas performances no processo.

Engenheiros de *software* são treinados individualmente para um conjunto de objetivos pessoais, definindo os métodos a serem usados, medindo seus trabalhos, analisando os resultados, e ajustando os métodos para atingir seus objetivos.

O PSP é uma estratégia para o desenvolvimento pessoal com o objetivo de aumento de produtividade.

Trabalhos experimentais foram iniciados em algumas corporações para verificar como engenheiros experientes poderiam reagir ao PSP e explorar a introdução de seus métodos. Foi verificado que os engenheiros de *software* geralmente são atraídos pela estratégia do PSP e encontram métodos que ajudam em seus trabalhos. Nas palavras de um engenheiro: "Isto não é para a empresa, é para mim". O leitor interessado pode obter informações adicionais em [SEI] .

O PSP aplica princípios de processo para o trabalho do engenheiro de *software* por:

- Fornecer um padrão de processo pessoal definido;
- Introduzir uma família de medidas de processo;
- Usar essas medidas para trilhar e avaliar o desempenho;
- Se esforçar para obter critérios de qualidade e melhorar os objetivos.

Usando o PSP, engenheiros:

- Desenvolvem um plano para todo o projeto;
- Registram seu tempo de desenvolvimento;
- Trilham seus defeitos;
- Mantêm dados de um projeto em relatórios resumidos;
- Usam esses dados para planos de projetos futuros;
- Analisam dados que envolvem seus processos a fim de aumentar suas performances.

Visão crítica sobre controle de qualidade de software

O grande problema no controle de qualidade de *software* é ainda a falta de consciência de muitas empresas e profissionais que lidam com sistemas complexos em adotarem uma política de qualidade nos trabalhos a serem desenvolvidos.

Como prova disso, uma pesquisa realizada em 1995 pelo Subprograma Setorial da Qualidade e Produtividade em *Software* (SSQP/SW), no qual foi indicado que [CESAR97]:

- Apenas 38,9% das empresas incluem sistematicamente metas ou diretrizes para qualidade em seus planos;
- 25,1% delas coletam sistematicamente indicadores de qualidade de seus produtos e serviços;
- Apenas 11,5% têm um programa de qualidade total implantado;
- A grande maioria (85,9%) não conhece o modelo CMM;
- 66,6% não adota nenhum procedimento específico de garantia da qualidade do produto de *software*;
- A avaliação de desempenho dos funcionários é feita periodicamente em apenas 16,4% das empresas; 54,1% delas disseram fazê-la informalmente.

O importante não é o modelo a ser seguido. Quando o objetivo é otimizar a qualidade do *software*, deve-se ter cuidado, porém, em definir certos limites para os resultados a serem alcançados.

A criação de *software* com qualidade requer um esforço tanto no nível financeiro quanto no nível de conscientização das pessoas envolvidas no processo, sem, no entanto, sairmos da órbita em que o cliente é o termômetro da qualidade de um determinado produto.

Mesmo se o produto (*software*) alcançar grande parte dos requisitos de qualidade, mas ultrapassar uma data que seja de vital importância para o cliente, o produto não terá qualquer serventia e, por isso, deve-se ter compromisso com prazos e outras coisas mais e, em certos casos, é desejável restringir o escopo da qualidade (partindo do máximo da qualidade para uma qualidade muito boa) a ser atingida, a fim de satisfazer as necessidades do cliente.

Esta é uma visão realística de encarar a qualidade de *software* nos sistemas a serem desenvolvidos.

Material Complementar

Para aprofundar seus estudos sobre CMM, consulte os sites e as seguintes referências:

- ✓ http://www.cin.ufpe.br/~in953/olds/relatorios/fabrica1.pdf
- ✓ http://www.spinsp.org.br/
- ✓ http://ibpi.org/standard/isoiec-15504/

GAMMA, Erich. **Padrões de projeto:** soluções reutilizáveis de software orientado a objetivos. Porto Alegre: Bookman, 2000.

SOMMERVILLE, Ian. **Engenharia de software.** 8.ed. São Paulo: Pearson Addison-Wesley, 2007.

Referências

FERNANDES, Aguinaldo Aragon. http://www.redepro.rs.gov.br/docs/11177116862Seminario_redepro_palestra_1.pdf. Acesso em: 22 jul. 2010.

GAMMA, Erich. **Padrões de projeto:** soluções reutilizáveis de software orientado a objetivos. Porto Alegre: Bookman, 2000.

GOMES, Nelma S. **Qualidade de Software - Uma Necessidade.** Artigo obtido em fev. 2003 em: www.esaf.fazenda.gov.br/cst/arquivos/Qualidade_de_Soft.pdf.

MCCALL J.; RICHARDS P.; WALTERS, G. **Factors in Software Quality** (3 vols.), NTIS AD-AO49-014, 015, 055, Nov. 1977.

PAULA FILHO, Wilson de Pádua. **Engenharia de software:** fundamentos, métodos e padrões. 3.ed. Rio de Janeiro: LTC, 2005.

PRESSMAN, Roger S. **Engenharia de software.** 6.ed. Porto Alegre: Bookman, 2006.

_____. Software Engineering, **A Practicioners Approach**, McGraw-Hill.

RAKITIN, Steven R. **Software Verification and Validation: a Practitioner's Guide**. Artech House, 1997.

RESENDE, Denis Alcides. **Engenharia de software e sistemas de informação.** 3.ed. Rio de Janeiro: Brasport, 2005.

SOMMERVILLE, Ian. **Engenharia de software.** 8.ed. São Paulo: Pearson Addison-Wesley, 2007.

[SEI2000] Sei, An **Overview of Capability Maturity Model Integration (CMMI)** – Version 1.0, Tutorial presented at SIMPROS 2000 [23], 2000.

[SEI2002a] Sei, **Web Site do software Engineering Institute** – SEI, http://www.sei.cmu.edu/ (CMMI Models available at www.sei.cmu.edu/cmmm).

[ISO9001:2000] International Standard Organization Certification for IMS Company.

[ISO12207:2000] International Standard Organization. ISO/IEC 12207 **Amendement: Information Technology** – Amendement to ISO/IEC 12207, versão PDAM 3, novembro 2000.

Anotações


Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo SP Brasil
Tel: (55 11) 3385-3000

Universidade Cruzeiro do Sul

UNICID Universidade Cidade de S. Paulo

UNIFRAN Universidade de Franca

UDF Centro Universitário

Módulo Centro Universitário