

CAPABILITY MATURITY MODEL FOR SOFTWARE

Eduardo Mayer Fagundes
e-mail: eduardo@efagundes.com

1. Introdução

Após décadas de incontáveis promessas sobre como aumentar a produtividade e qualidade de software, através de novas metodologias e tecnologias, as organizações chegaram à conclusão que o problema fundamental é a inabilidade para gerenciar processos de desenvolvimento de software. Em muitas organizações os projetos não cumprem os prazos planejados e os custos ficam acima do budget, afetando os benefícios que o projeto traria para a organização. Uma resposta para esse ambiente caótico foi o desenvolvimento de um modelo de maturidade para software, o CMM.

O CMM – *Capability Maturity Model* para software é um conjunto de processos desenvolvido pela SEI – *Software Engineering Institute* (www.sei.cmu.edu) em 1986 para melhorar o desenvolvimento de aplicações em organizações que trabalham com tecnologias de software. O processo é dividido em 5 níveis de desenvolvimento: Inicial, repetível, definido, gerenciado com métricas e otimizado.

Esses cinco níveis de maturidade provêm uma escala mensurável de maturidade, como também a capacidade de execução, de uma organização que usa tecnologias de software. Esses níveis também ajudam a definir a prioridade dos esforços de uma organização de software. Cada vez mais as organizações dependem de regras formais do que indivíduos que executam projetos sem planejamento. Para tornar os projetos competitivos, dentro do budget e prazo planejados, as organizações devem desenvolver softwares dentro de ambientes “maduros”.

Em uma organização madura, os gerentes monitoram a qualidade dos produtos de software e os processos que são utilizados para produzi-los. O objetivo é baseado em métricas julgar a qualidade dos produtos e analisar os problemas dos produtos e processos. O cronograma de desenvolvimento e o budget devem estar baseados em dados históricos para serem mais realísticos; os custos, funcionalidade e qualidade dos produtos são normalmente arquivados. Em geral, a disciplina de um processo é consistentemente seguida por que todos os participantes entendem o valor do que eles estão fazendo, e a necessidade da infra-estrutura para suportar esses processos.

2. Conceitos fundamentais sobre Processos de Maturidade

Um processo de software pode ser definido como um conjunto de atividades, métodos, práticas e mudanças que as pessoas devem usar para desenvolver e manter os produtos de software. Por exemplo, planejamento, documentos sobre o desenho do software, códigos, resultado de testes e manuais de usuários. Os processos de software mostram a quanto uma organização é madura para desenvolver e manter os produtos de software.

Existem três tipos de análise sobre os processos de software:

- (1) Capacidade. Essa análise descreve a gama de resultados que podem ser atingidos com o uso dos processos de software.
- (2) Desempenho. Analisa o atual estágio dos processos de software e os resultados obtidos pelo seu uso.
- (3) Maturidade. Analisa até que ponto um processo específico está definido, gerenciado, mensurado, controlado e é efetivo. Maturidade implica ter potencial para um crescimento consistente aplicando os processos de software em projetos da organização.

Uma organização ganha maturidade se institucionaliza os processos de software através de políticas, padrões e estrutura organizacional.

3. Características dos Níveis de Maturidade

Os processos devem ser continuamente aperfeiçoados através de pequenas melhorias, ao invés de inovações revolucionárias. O CMM provê uma estrutura – *framework* – para organizar os passos de melhorias dentro de cinco níveis de maturidade em processos de software de uma organização. Esses cinco níveis de maturidade definem uma escala para medir o estágio de maturidade de uma organização.

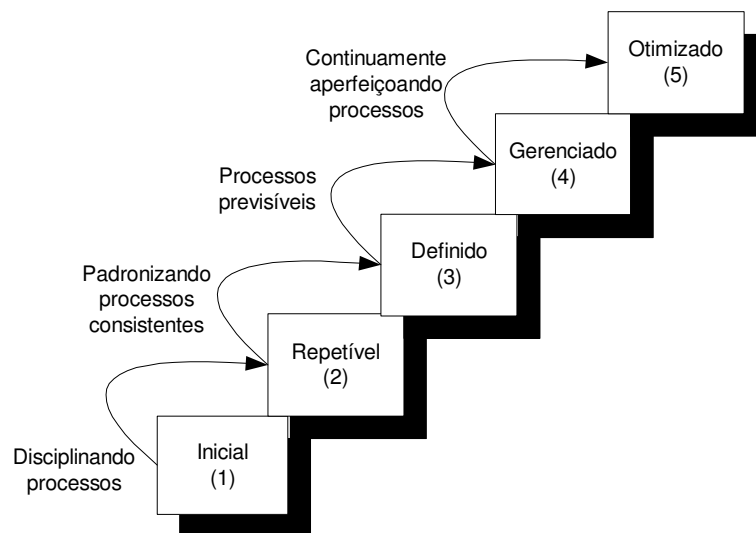


Figura 1. Os cinco níveis do CMM para software.

A Figura 1 apresenta os cinco níveis de maturidade de uma organização que utiliza processos de software.

(1) **Nível 1 – Inicial**

Esse é o nível base, as aplicações são desenvolvidas com métodos e práticas não consistentes. Os processos de desenvolvimento raramente são definidos e as práticas disponíveis são sacrificadas para atender a prazos incorretamente definidos. Os programadores são capazes de realizar suas atribuições

utilizando seus próprios métodos, muitas vezes não consistentes entre os processos da organização. Frequentemente, o gerenciamento dos projetos é fraco e não protege o projeto de rupturas. Essencialmente, as organizações no nível 1 carecem de capacidade de comprometimento consistente.

(2) Nível 2 – Repetível

O primeiro ponto importante nesse nível é estabelecer um ambiente estável para se repetir práticas de sucesso. Então, o nível 2 foca no desenvolvimento de capacidades dos gerentes de projetos para planejar eficazmente os compromissos assumidos e estabelecer um controle dos requerimentos para os produtos de software. Para controlar os projetos deve utilizar diferentes métodos e práticas e o ambiente deve ser estável. As organizações de desenvolvimento de aplicações no nível 2 devem possuir capacitação para liberar seus produtos dentro do cronograma e budget, evitando horas extras e custos além do budget.

(3) Nível 3 – Definição

Após atingir o estágio de repetir as práticas de desenvolvimento com sucesso, as organizações de desenvolvimento de aplicações devem identificar as melhor práticas dos melhores projetos. Subseqüentemente, esses procedimentos são integrados aos padrões de desenvolvimento para toda a organização. Conseqüentemente, uma forte cultura é desenvolvida no nível 3 com a utilização de processos comuns que cobrem os mais importantes elementos do desenvolvimento de aplicações. Uma vez todos os projetos utilizando as melhores práticas e compartilhando lições de aprendizagem, existe o amadurecimento da organização.

(4) Nível 4 – Gerenciamento com métricas

Tendo atingindo o estágio 3 com a utilização de processos comuns nos projetos de desenvolvimento de software, as organizações estão capacitadas a gerar estatísticas que possam caracterizar o desempenho de seus processos. Essas estatísticas provêm informações para se entender a capacidade de desenvolvimento baseado nos processos e as causas das variações de desempenho. Gerenciando o desempenho dos processos de desenvolvimento estatisticamente, uma organização pode prever e controlar os resultados dos projetos. O gerenciamento da quantidade produzida permite um grande envolvimento dos times de projeto e melhorar a previsão dos resultados pelo gerenciamento de projetos.

(5) Nível 5 – Otimização

Melhorias contínuas podem ser desenvolvidas através das lições de aprendizagem de cada projeto podem ser desenvolvidos através de ações pró-ativas durante a avaliação de novos métodos de desenvolvimento, novos processos ou tecnologias. Uma organização com um nível de maturidade 5 estabelece uma infra-estrutura para suportar contínuas mudanças no gerenciamento dos processos de desenvolvimento.

4. Definição Operacional de CMM

O CMM possui uma estrutura que indica o caminho recomendado para as organizações melhorarem seus processos de desenvolvimento de software. Essa parte operacional do CMM é desenhada para suportar as várias formas de serem usadas. Existem pelo menos quatro usos do CMM:

- 1) Para identificar pontos fortes e fracos em uma organização;
- 2) Para identificar os riscos dentro de um processo de seleção de fornecedores e monitorar os resultados;
- 3) Para que o alto nível gerencial de uma organização possa entender as atividades necessárias para o lançamento de um processo de software para melhorar o programa.
- 4) Para o pessoal técnico e dos grupos de processos possam melhorar os processos de software em suas organizações.

5. Estrutura Interna dos Níveis de Maturidade

Cada nível de maturidade pode ser decomposto em partes. Com exceção do nível 1, a decomposição de cada nível de maturidade abrange desde os sumários abstratos de cada nível até as definições operacionais das práticas chaves, como é mostrada na figura 2. Cada nível de maturidade é composto por vários processos chaves. Cada processo chave é organizado dentro de 5 secções chamadas de características comuns. As características comuns especificam as práticas chaves que, quando corretamente usadas, atingem os objetivos dos processos chaves das áreas.

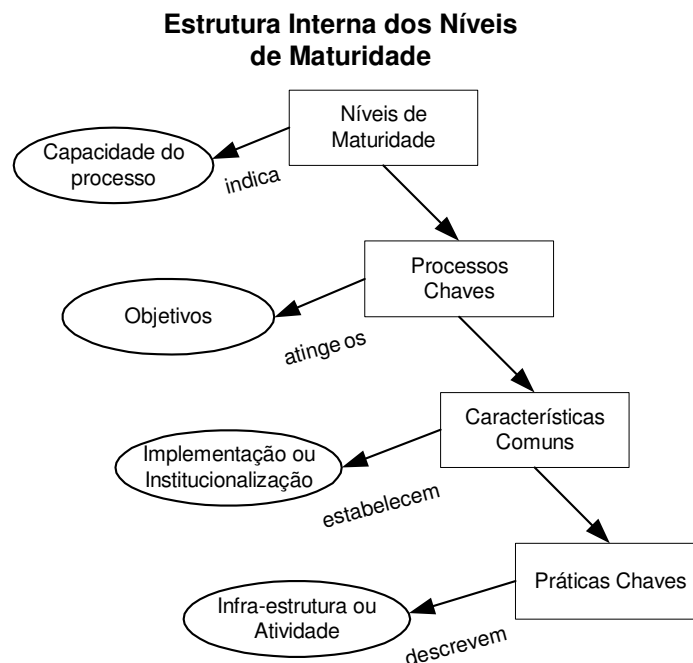


Figura 2. Estrutura Interna dos Níveis de Maturidade

(1) Processo Chave

Cada processo chave identifica um conjunto de atividades relacionadas que, quando executadas coletivamente, cumprem o objetivo do proposto. O processo chave teve ser definido para um único nível de maturidade. O caminho para atingir os objetivos de um processo chave poderá ser diferente para cada projeto, dependendo das diferenças entre os projetos.

O processo é considerado chave, pois ele é crítico para atingir o nível de maturidade. O CMM não descreve todos os processos chaves em detalhes que são requeridos para desenvolver e manter os produtos de software.

Os processos chaves para o nível 2 têm foco nos problemas de controle dos projetos de software:

- Gerenciamento dos requerimentos do projeto.
- Gerenciamento do plano de atividades do projeto.
- Acompanhamento e inspeção do progresso do projeto.
- Gerenciamento dos fornecedores contratados para o projeto.
- Gerenciamento da qualidade do projeto.
- Gerenciamento da configuração de software.

Os processo chaves para o nível 3 têm foco no projeto e na organização:

- Estabelecimento de uma organização responsável pelas atividades que compõem os processos de software.
- Estabelecimento de uma organização para definir, desenvolver e manter um conjunto de processos de software para melhorar a performance dos projetos.
- Estabelecimento de um programa de treinamento para desenvolver os talentos da organização.
- Integração do gerenciamento entre a engenharia e as atividades administrativas dentro de um processo coerente.
- Estabelecimento de uma engenharia de produto que integre todas as atividades de engenharia para o desenvolvimento eficiente do produto.
- Estabelecimento de um método de revisão dos produtos de software para avaliar os defeitos e eficiência.

Os processos chaves para o nível 4 estão ligados a métricas qualitativas dos processos de software e dos produtos de software:

- Gerenciamento dos processos quantitativos para controlar a performance dos projetos.
- Gerenciamento da qualidade dos produtos de software.

Os processos chaves para o nível 5 cobrem os aspectos de como as organizações e os projetos devem implementar continuamente e aferir as melhorias nos processos de software:

- Estabelecimento de processos para prevenir defeitos e evitar a recorrência dos problemas.
- Estabelecimento de processos para identificar mudanças tecnológicas e transferi-las para os processos de software.

(2) Características Comuns

Por conveniência, as práticas que são descritas nos processos chaves são organizadas por características comuns. Essas características comuns são atributos que indicam se a implementação e a institucionalização dos processos chaves são efetivos, repetíveis e duradouros. As cinco características comuns são:

- Compromisso para fazer. Descreve as ações que a organização está tomando para assegurar que o processo está estabelecido e será permanente.
- Habilidade para fazer. Descreve os pré-requisitos que devem existir em um projeto ou organização para implementar um processo de software de forma competente.
- Atividades realizadas. Descreve as funções e procedimentos necessários para implementar um processo chave.
- Aferição e análise. Descreve necessidade de medir um processo e analisar os resultados aferidos.
- Inspeção de implementação. Descreve os passos para assegurar que as atividades são realizadas conforme os processos estabelecidos.

(3) Melhores Práticas

Cada processo chave é descrito em termos de práticas chaves que contribuem para atingir o objetivo do projeto. As práticas chaves descrevem a infraestrutura e as atividades que mais contribuem para uma implementação efetiva e institucionalizada dos processos chaves.

6. Futuras Direções do CMM

As organizações devem trabalhar no longo-prazo para atingir os mais elevados níveis de maturidade. As organizações de software podem levar 10 anos ou mais para construir uma fundação sólida e uma cultura orientada para melhorias contínuas. O CMM não apresenta todas as soluções para se obter sucesso nos projetos. Por exemplo, o CMM não cobre o conhecimento em domínios específicos de negócios, em critérios para definir uma tecnologia específica, em aspectos motivacionais de pessoal e retenção de talentos. Esses aspectos que são cruciais para o sucesso de vários projetos não deverão estar integrados ao CMM.

Durante os próximos anos, o CMM continuará a desenvolvido e extensivamente testado para provar sua eficiência na melhoria dos processos de desenvolvimento de software.

DADOS DO AUTOR



Eduardo Mayer Fagundes estuda os impactos da tecnologia da informação e modelos de gestão de TI nas organizações, focando em tecnologia, técnicas e gestão de pessoas. Seu livro "Como Ingressar nos Negócios Digitais" foi publicado em parceria com o SEBRAE Nacional com o objetivo de ampliar a visão empresarial no comércio eletrônico.

Eduardo é graduado em engenharia elétrica, possui especialização em telecomunicações e é mestre em ciência da computação. Foi professor por mais de 20 anos em conceituadas instituições de ensino. Palestrante em vários seminários e congressos. Foi gerente de infra-estrutura e sistemas da Ford Brasil, responsável pelo desenho da infra-estrutura de TI da moderna fábrica da montadora em Camaçari-Bahia.

Atualmente é diretor de TI (CIO) das empresas do grupo americano AES no Brasil. A AES atua nos mercados de geração e distribuição de energia e na área de telecomunicações. A AES Eletropaulo, maior distribuidora de energia da América do Sul, é uma das empresas do grupo.

Escreve artigos no site www.efagundes.com.