

A MELHORIA DA QUALIDADE DE SOFTWARE: UM ENFOQUE AO MODELO SEI-SW-CMM NÍVEL 2

Angelita Moutin Segoria GASPAROTTO*
Gilberto Alexandre ROCHA**

RESUMO

O presente artigo apresenta a estrutura do Modelo SEI-SW-CMM NÍVEL 2, cujo objetivo é avaliar a maturidade das empresas desenvolvedoras de software e propor melhorias para este processo.

PALAVRAS-CHAVE: Melhoria de software. Qualidade. Processos. CMM.

INTRODUÇÃO

O movimento pela qualidade – dos artesãos ao Desdobramento da Função Qualidade - deve ser entendido como um conjunto de fatos e ações, acumulado ao longo do tempo, o qual levou à preocupação pela qualidade nos dias atuais. A transformação dos mercados desencadeou um processo em que as empresas concentram esforços pela excelência em qualidade. Nesse contexto, as empresas buscam modelos, ferramentas e métodos, com o objetivo de aumentarem suas vantagens competitivas, podendo compartilhar assim os mercados globais.

Com mudanças tecnológicas cada vez mais velozes, crescimento na complexidade dos produtos, competição global, disseminação do uso da tecnologia e evolução nos conceitos da qualidade, torna-se necessário investigar os processos de desenvolvimento de software, com o objetivo de alcançar produtos com qualidade superior. Essa investigação é justificada uma vez que a qualidade do produto de software está diretamente relacionada à qualidade do seu processo de desenvolvimento.

No âmbito de software, até a década de 70, a noção da qualidade estava ligada à parte interna deste, sendo avaliada em termos de linhas de código e tempo de execução. Nos anos 80, havia uma maior qualificação dos usuários. Nos anos 90, a discussão assumiu um caráter mais formal com a apresentação de Normas como a ISO 9000-3 e a ISO 9126. Atualmente, a discussão pela busca da qualidade leva as organizações a se preocuparem com a melhoria de seus produtos desde a sua concepção. Portanto, o enfoque principal dado às atividades de garantia da qualidade de software tem se concentrado na melhoria de seu processo de desenvolvimento.

Em se tratando de avaliação e melhoria do processo de desenvolvimento de software, o CMM (Capability Maturity Model), desenvolvido pelo SEI (Software Engineering Institute) da Universidade Carnegie Mellon (Pittsburgh – EUA), é um modelo utilizado para qualificar e definir os pontos de atuação, além das ações para a melhoria evolutiva desse processo. De acordo com Belloquim (1999), o CMM foi criado para todo tipo de empresa desenvolvedora de software, independentemente do seu tamanho. O sucesso de sua implementação é relativo à capacidade que a empresa tem de absorver os conceitos e adaptá-los à sua realidade.

*Faculdade de Tecnologia de Taquaritinga. Av. Dr. Flávio Henrique Lemos, 585 - Portal Itamaracá. segovia_74@yahoo.com

**Faculdade de Tecnologia de Taquaritinga. Av. Dr. Flávio Henrique Lemos, 585 - Portal Itamaracá. betofatectq@yahoo.com.br

QUALIDADE DE SOFTWARE

Conceitos

A respeito da qualidade de software, há muitos conceitos para defini-la, sob diferentes pontos de vista apresentados na literatura. Abaixo são apresentadas algumas definições dos principais autores:

- Um produto de software apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto (SANDERS & CURRAN, 1994);
- Qualidade de software é o grau em que o software possui uma combinação desejada de atributos (IEEE-STD729, 1983);
- Qualidade de software é o grau em que os atributos do software são capazes de desempenhar sua finalidade especificada (DOD-STD 2168);
- A percepção da qualidade de software é vista principalmente em termos de tempo em que um sistema opera corretamente (SMITH & WOOD, 1989);
- Qualidade é a conformidade aos requisitos estabelecidos explicitamente e aos padrões de desenvolvimento documentados (PRESSMAN, 1994).

Qualidade do Processo de Desenvolvimento de Software

De acordo com Paulk et al. (1993), o processo de desenvolvimento de software é representado por um conjunto seqüencial de atividades, objetivos, transformações e eventos que encapsulam estratégias para o cumprimento da evolução do software.

Xerox (1983) considera processo como sendo uma série de atividades de trabalho correlacionada caracterizada por uma série específica de entradas e tarefas que agregam valor ao produto.

De acordo com Juran (1989) um processo é definido como uma série sistemática de ações direcionadas para a consecução de uma meta. Para estar de acordo com os termos dessa definição, um processo deve ser:

- Direcionado em função de uma meta: não se pode planejar abstratamente, sem saber qual é a meta. Isso significa que, para planejar para a qualidade, deve-se antes estabelecer as metas e padrões da qualidade que se deseja alcançar;
- Sistemático: as ações que constituem um processo estão todas interligadas e são independentes. Elas também são progressivas e seguem uma seqüência estipulada;
- Capaz: o resultado final adequado do planejamento é um processo capaz de atender às metas sob condições de operações normais;
- Legítimo: o processo se desenvolve através de canais autorizados. Ele deve ser aprovado pelos que receberam responsabilidades relacionadas a ele.

Para Fiorini (1998) conhecer os processos significa conhecer como os produtos e serviços são planejados, produzidos e entregues ao cliente. Para que um processo de software seja efetivo para o cumprimento dos seus objetivos, torna-se necessário um planejamento detalhado que revele a realidade do ambiente de desenvolvimento do software. Deve-se considerar aspectos específicos do projeto, como metas e políticas, equipe de desenvolvimento, cronograma, disponibilidade de recursos humanos, técnicos e financeiros.

O Modelo SEI-SW-CMM

Histórico e Conceituação

O SW-CMM foi desenvolvido pelo SEI (Software Engineering Institute) sediado na CMU (Carnegie Mellon University) em Pittsburg, Pennsylvania, Estados Unidos. O SEI é um centro de pesquisa criado em 1984 pelo Departamento de Defesa dos Estados Unidos (DoD – Department of Defense) e é patrocinado pelo OUSD (A&T) (Office of the Under Secretary of Defense for Acquisition and Technology). O SEI tem por missão aprimorar a prática de Engenharia de Software. As áreas de atuação do SEI são: capacitação de gerência de software, tecnologia para a engenharia e aptidão para a transição. O SEI focaliza a transição tecnológica: o desenvolvimento e adoção das melhores práticas de engenharia de software.

O CMM classifica as organizações produtoras de software em cinco grupos (ou cinco níveis) de maturidade. Os cinco níveis de maturidade propostos pelo SW-CMM são: inicial, repetível, definido, gerenciado e em otimização. A Figura 1 a seguir, ilustra os Níveis de Maturidade do CMM:

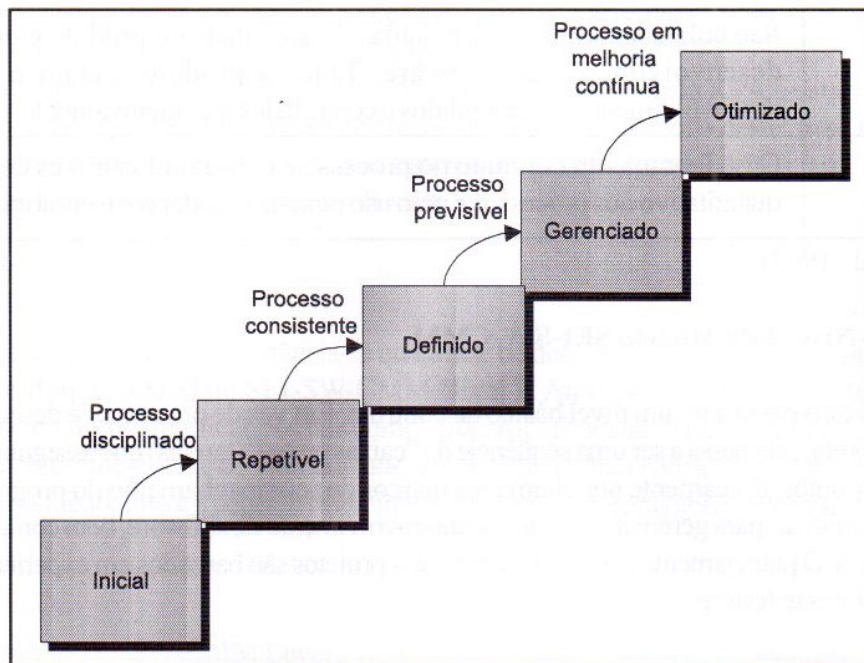


Figura 1 - Níveis de Maturidade do CMM
Fonte: (Paulk et al., 1993 p. 16)

Como apresentado pela Figura 1, o CMM classifica as organizações em cinco níveis evolutivos de maturidade, cada um com suas características próprias. Estas características estão apresentadas no Quadro 1:

Quadro 1 - Visão Geral dos Níveis de Maturidade do CMM

Nível do CMM	Características
1 Inicial	O processo de desenvolvimento é desorganizado e até caótico. Poucos processos são definidos e padronizados e o sucesso depende de esforços individuais e heróicos dos desenvolvedores.
2 Repetível	Os processos básicos de gerenciamento de projeto estão estabelecidos e permitem acompanhar custo, cronograma e funcionalidade e é possível repetir o sucesso de um processo já utilizado anteriormente.
3 Definido	Tanto as atividades de gerenciamento quanto as de engenharia do processo de desenvolvimento de software estão documentadas, padronizadas, e integradas em um padrão de desenvolvimento da organização. Todos os projetos utilizam uma versão aprovada e adaptada do processo padrão de desenvolvimento de software da organização.
4 Gerenciado	São coletadas medidas detalhadas da qualidade do produto e do processo de desenvolvimento de software. Tanto o produto quanto o processo de desenvolvimento são entendidos e controlados quantitativamente.
5 Em otimização	O melhoramento contínuo do processo é conseguido através de um feedback quantitativo dos processos e pelo uso pioneiro de idéias e tecnologias inovadoras.

Fonte: (Paulk et al., 1993)

Apresentando o Nível 2 do Modelo SEI-SW-CMM

No Nível 2, o processo passa a ter um nível básico de controle. Em vez de processo de desenvolvimento ser uma única “caixa preta”, ele passa a ser uma seqüência de “caixas pretas” (tarefas) que asseguram a visibilidade em determinados pontos, tipicamente nos chamados marcos de acompanhamento do progresso do projeto. São estabelecidas políticas para gerenciar projetos de desenvolvimento de software, bem como procedimentos para implementá-las. O planejamento e a gerência de novos projetos são baseados em experiências adquiridas com projetos similares anteriores.

A capacitação do processo é melhorada projeto a projeto, através do estabelecimento de disciplinas básicas de gerência. Para cada projeto são estabelecidos processos que são definidos, documentados, praticados, executados, treinados, medidos, obedecidos e passíveis de melhoria.

Todos os níveis de maturidade do CMM com exceção do Nível 1, são compostos de um certo número de áreas-chave de processo (Key-Process Areas - KPAs). Essas áreas descrevem os objetivos que devem ser atingidos, assim como as questões a serem tratadas para se alcançarem estes objetivos e atingir o nível proposto. As áreas-chave de processos estabelecidas no Nível 2 são: Gerenciamento de Requisitos, Planejamento do Projeto de Software, Acompanhamento e Supervisão do Projeto de Software, Gerenciamento de Sub-Contratos de Software, Garantia da Qualidade de Software, Gerenciamento da Configuração de Software.

- KPA 1 - Gerenciamento de requisitos: de acordo com Paulk et al. (1993), a proposta desta KPA é estabelecer um entendimento comum entre o cliente e a equipe do projeto sobre os requisitos alocados ao software. Envolve o estabelecimento e a manutenção dos requisitos do software de acordo com as necessidades do cliente;
- KPA 2 - Planejamento do projeto do software: a finalidade desta KPA é estabelecer planos para o desempenho e manutenção do projeto. Este planejamento envolve o desenvolvimento de estimativas para o trabalho, o estabelecimento de compromissos e de planos para desempenhar o trabalho;
- KPA 3 - Acompanhamento e supervisão de projeto de software: A finalidade desta KPA é promover uma visão adequada do progresso real do projeto, de modo que o gerenciamento possa tomar medidas efetivas quando o desempenho se desvia do plano proposto;
- KPA 4 - Gerenciamento de sub-contratos de software: a finalidade desta KPA é selecionar fornecedores qualificados e gerenciá-los eficazmente. Esse processo envolve selecionar o fornecedor, estabelecer compromissos, acompanhar e revisar o desempenho e os resultados obtidos;
- KPA 5 - Garantia de Qualidade de Software: a finalidade desta KPA é promover o gerenciamento com visibilidade do processo que está sendo utilizado e dos produtos que estão sendo desenvolvidos. Isso envolve revisões e auditorias nos produtos de software e nas atividades para assegurar que estão em conformidade com os padrões e procedimentos aplicados;
- KPA 6 - Gerenciamento da configuração de software: a proposta desta KPA é estabelecer e manter a integridade dos produtos do projeto de software ao longo do ciclo de vida deste. Esse processo envolve identificar os itens de configuração, controlar sistematicamente as alterações e manter a integridade da configuração ao longo do ciclo de vida do software.

CONCLUSÕES

Este artigo procurou apresentar a importância da qualidade de desenvolvimento de software nos dias atuais, tomando como enfoque, o Modelo SEI-SW-CMM Nível 2. Através da implementação deste Modelo, poder-se-á verificar efeitos significativos tais como: alta produtividade, baixo custo de desenvolvimento, envolvimento e comprometimento do fator humano, comprometimento com a qualidade e planejamento das atividades. Todos os efeitos citados poderão trazer inúmeras vantagens estratégicas e competitivas de mercado.

ABSTRACT

This article shows the SEI-SW-CMM Level 2 framework, whose objective is to evaluate the maturity of software houses and propose improvement for this process.

KEYWORDS: Software improvement. Quality. Processes. CMM.

REFERÊNCIAS

- BELLOQUIM, Átila. CMM em Pequenas Organizações: Seria Mesmo Possível? **Developers Magazine**. Fev. 1999.
- FIORINI, S.T. *Engenharia de Software com CMM*. Rio de Janeiro, Ed. Brasport. 1998
- IEEE (1983). *IEEE Standard Glossary of Software Engineering Terminology*. New York: IEEE, ANSI/IEEE Std 729. 1983.
- JURAN, J. M. *Planejando para a Qualidade*. 2. ed. São Paulo: Ed. Pioneira. 1989, p. 195-255
- PAULK, M. C. et. al. *Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute,

Carnegie Mellon University, CMU/SEI-91-TR-24. ESC-TR-93-177. 1993.

PRESSMAN, R. S. *Software Engineering - A Practitioner's Approach*, Edição Européia, Editora McGraw-Hill. 1994.

SANDERS, J. & CURRAN, E. *Software Quality - A Framework for Success in Software Development and Support*, Addison-Wesley. 1994.

SMITH, D. J. & WOOD, K. B. (1989). *Engineering Quality Software - A Review of Current Practices Standards and Guidelines Including New Methods and Development Tool*, Elsevier Science Publishers Ltd. 1989.

XEROX. *Liderança através da Qualidade*. Leesburg. Virgínia, Agosto. 1983.