

O ENVOLVIMENTO DE TODOS NA ORGANIZAÇÃO REPRESENTA UM FATOR CRÍTICO DE SUCESSO NA IMPLANTAÇÃO DO CMM (*CAPABILITY MATURITY MODEL*)

Paulo Henrique N. de Jesus¹

Masterson Oliveira Galvão²

Zilah de Barros Mello³

Marconi Nogueira Placido dos Santos⁴

RESUMO

O intuito desse artigo é mostrar que a implantação do modelo de gestão da qualidade CMM (*Capability Maturity Model*) representa uma mudança em toda a estrutura da organização, sendo que o seu sucesso irá depender da existência de uma sinergia que possa envolver a todos dentro da empresa e esta possa romper paradigmas, não bastando que apenas parte do grupo esteja empenhada, pois não será motivo satisfatório para que o processo evolua. Verificamos que o fator humano é um fator crítico de sucesso na adoção do modelo CMM. Utilizamos como principais fontes para essa pesquisa artigos de periódicos e científicos, livros de autoridades no assunto e como referência um estudo de caso realizado por Gustavo da Gama Torres do Departamento de Ciência da Computação da UFMG-ICEX.

Palavras-Chave: Qualidade. Mudanças. Motivação. Cultura.

ABSTRACT

The intention of this article is to show that the implantation of the Management Quality Model CMM (*Capability Maturity Model*) represent a change in all structure of the Organization but, the success will depend of the existence of a common interest of all members involveds in and, this members will can break paradigms didn't suffice that part of members was involved in because it's not a satisfactory motive for the evolution process. We verified that the human factor is a critical factor of success in the CMM's conception. We used like principals source for this research periodicals articles and scientifics, authority's books in this theme and like reference a case of study realized by Gustavo da Gama Torres of the Computation Science Departament of UFMG-ICEX.

Keywords: Quality. Changes. Motivation. Culture.

1 INTRODUÇÃO

Mudar a forma de trabalhar de um grupo de indivíduos, como é o caso de uma organização, requer a participação de todos na empresa e isto é o que acontece quando da implantação de um novo modelo como o estudado neste artigo, pois o caminho é longo e envolve mudar a maneira de agir das pessoas e nem todas

¹ Bacharel em Administração com ênfase em Análise de Sistemas pela Faculdade Visconde de Cairu (2004). E-mail: pauloj@telemar-ba.com.br

² Pós-graduando em Planejamento Estratégico pela Unopar (2014-atual); Pós-graduando em Gestão de Produção de Bens e Serviços (2015-atual); Especialista em Gerência de Projetos pela Fundação Getúlio Vargas (2007); Especialista em Gestão da Administração Pública pela EsAEx/Faculdade Castelo Branco (2010); Bacharel em Administração com ênfase em Análise de Sistemas pela Faculdade Visconde de Cairu (2004). E-mail: masterson.galrao@gmail.com

³ Bacharel em Administração com ênfase em Análise de Sistemas pela Faculdade Visconde de Cairu (2004). E-mail: zbmello@bol.com.br

⁴ Especialização em Administração pela Universidade Salvador e mestrado em Administração Estratégica pela Universidade Salvador

estão preparadas ou mesmo aceita assumir esse compromisso.

O CMM – Capability Maturity Model for Software -, ou Modelo de Maturidade da Capacitação para Software, propõe um caminho gradual que leva as organizações a se aprimorarem continuamente na busca da sua própria solução dos problemas inerentes ao desenvolvimento sistemático de software (CORDEIRO, 2003).

A implantação do CMM é de longo prazo e exige de todos na empresa, uma compreensão de seus princípios e, especificamente, interesse e apoio efetivo da alta administração...é um processo que envolve aspectos de mudança cultural dentro da empresa que o adota (ROCHA, MALDONADO e WEBER, 2001).

Podemos notar que a empresa precisa se organizar e os diversos setores desta precisam atuar para que a organização como um todo fique ciente das mudanças e que ninguém deixe de participar ativamente. Sendo que esta mudança deve ser feita de forma paulatina de modo que os participantes se adequem pouco a pouco e não se desestimulem com os efeitos do programa modelo que está sendo implantado, bem como a organização deve desenvolver paralelamente ações que mantenham seus colaboradores motivados. (ALBERTIN, 2001) descreve os fatores que uma organização precisa trabalhar para motivar os seus colaboradores:

A perfeita composição dos seguintes fatores produzirão motivação e produtividade no trabalho: variedade de competência - com a interligação de diferentes competências e talentos das pessoas; identidade de tarefa - com a visão de início e fim de uma atividade com seu resultado; significado da tarefa - com a influência que o trabalho tem sobre a vida ou o trabalho de outra pessoa, interna ou externa à organização; resposta (feed back) do trabalho - com informação sobre o desempenho dos empregados.

Este artigo se resume a mostrar os primeiros passos que uma empresa deve dar para se enquadrar no modelo SEI-CMM que é justamente a passagem da empresa do nível 1 para o nível 2, assim procuraremos descrever nos capítulos seguinte, panorama da engenharia de software com CMM, a situação das organizações que estão no nível 1 do modelo CMM, os pontos essenciais que uma organização deve implementar para ser avaliada como nível 2, as dificuldades encontradas pelas empresas frente a *mudança* organizacional, um estudo de caso realizado por Gustavo da Gama Torres do Departamento de Ciência da Computação da UFMG-ICEX e a conclusão final do estudo.

2 PANORAMA DA ENGENHARIA DE SOFTWARE COM CMM

No ramo da engenharia de software, o fator mais preocupante continua sendo a qualidade no processo de desenvolvimento de software. Esse tema apresenta-se bastante desgastado frente aos clientes, e usuários em geral, que adquirem e utilizam os seus produtos e constataam, mesmo sem conhecimentos técnicos e do seu processo de construção, que nesse ramo há muito o que amadurecer

As indústrias de software estão despreparadas para atender as rápidas necessidades dos mercados simplesmente porque não investiram no aperfeiçoamento de seus processos internos. O que estamos afirmando aqui é que a maioria das empresas que fornecem softwares a sua organização são amadoras, ou seja, desconhecem completamente um processo de engenharia de software (BARTIÉ, 2002).

Esse estigma foi criado e sustentado diante do panorama assustador dos resultados obtidos de diversos projetos mal sucedidos pelas empresas que desenvolvem softwares, resultados que trazem números inconcebíveis no meio empresarial. Os problemas são gerados desde o início do projeto, transferidos e aumentados no processo de desenvolvimento do software, balanceados na sua implementação e corrigidos durante a sua operação, o que é muito grave e oferece riscos à sobrevivência das empresas, tanto a que fornece quanto a que adquire.

Cerca de 55% de todo trabalho realizado pela área de desenvolvimento de software de uma empresa é destinado à correção de erros detectados em sistemas já em funcionamento (ROBERTA, 2003 at SEI)

Na grande maioria das empresas desenvolvedoras de softwares não existe um processo e/ou um modelo definido para a sua engenharia, o que dificulta e muito o cumprimento daquilo que foi negociado com o seu cliente.

Segundo o SGI (Standish Group International), em 2001, somente 9% dos projetos em tecnologia foram realizados dentro do prazo, dentro do orçamento ou dentro das especificações pré-determinadas, enquanto 29% falharam completamente. Tais erros acabam jogando fora algo entre U\$ 80 e U\$ 145 bilhões por ano em investimentos, conforme artigo *sintonizados para o sucesso* da revista Information Week Brasil de janeiro de 2003.

Esses ainda são os reflexos do desenvolvimento de uma ciência a partir do conhecimento empírico e heróico dos seus programadores e sem a devida preocupação a longo prazo, o que tornaram as empresas, de certa forma, escravas desses programadores, que na maioria dos casos não registram o que foi realizado durante a sua criação.

Somado a esses fatores culturais, verifica-se também a inexistência de um planejamento ou um planejamento mal elaborado, inconsistente, o que gera uma série de falhas técnicas e administrativas na honra do prazo, orçamento e especificação acordados, desencadeando os custos intrínsecos em retrabalhos (contabiliza-se aí tanto os custos do investimento nas horas gastas naquilo que já era para estar pronto e entregue quanto na impossibilidade de estar trabalhando em um projeto novo ou na atualização dos sistemas existentes), em queda de conceito frente ao seu cliente e, conseqüentemente, as futuras vendas, assim como nas perdas que normalmente não são mensuradas como queda na moral do grupo que trabalha no projeto, em virtude da sensação de não concluir os projetos com qualidade suficiente, entre outros.

Apenas 45% dos esforços destinam-se ao desenvolvimento de novos sistemas ou à atualizações daqueles já em operação. (ROBERTA, 2003 at SEI).

Como conseqüência dessa verdadeira catástrofe empresarial, verifica-se um excessivo estouro dos orçamento e prazos estimados e a entrega de um produto que não condiz com o que o cliente imaginou receber, o qual deverá ser adequado (retrabalho) para se aproximar daquilo que ele gostaria de adquirir.

A engenharia de software, através de órgãos reguladores e instituições particulares sem fins lucrativos, tem focado seus esforços nos últimos anos à busca da qualidade dos produtos gerados pelos desenvolvedores e, para tanto, tem investido em metodologias para minimizar os desvios encontrados nos processos, os quais oneram demasiadamente os projetos, e aumentar a produtividade, o qual apresenta um aspecto desanimador. Muitos têm sido os esforços para mudar esse panorama de descrédito e aumentar a produtividade no setor.

O melhoramento do processo de desenvolvimento são estudos recentes, visando garantir a qualidade do processo, garantir também a qualidade do produto. Estes estudos de qualidade dos processos de software estão intimamente ligados aos estudos da Engenharia de Software, um auxiliando a compreensão do outro,

através de modelos detalhados para desenvolvimento de sistemas ou software, nas suas diversas etapas ou fases. (REZENDE, 2002).

A iniciativa vem de todos os lados, reunião dos papas no assunto para a estruturação e formalização de uma *Linguagem de Modelagem Unificada* (UML) orientada a objetos, de um *processo unificado* (UP) que visa a padronização no processo de desenvolvimento de software, o qual é bastante utilizado pela UML e o *Capability Maturity Model* (CMM) que consiste na certificação de qualidade das empresas que adotam o modelo e galgam degraus de maturidade no seu processo de desenvolvimento dos sistemas. Esses são, dentre outros, os principais movimentos que buscam uma maior eficiência e eficácia no ambiente da engenharia de software e o que vamos focar nessa pesquisa será justamente uma das maiores dificuldades encontrada pelas empresas que não possuem uma metodologia no seu processo de desenvolvimento de software e buscam essa excelência através da CMM, o gerenciamento do fator humano.

Antes de concentrarmos as atenções no problema estudado, vamos fazer uma alusão da definição e um breve histórico de CMM.

CMM quer dizer modelo de maturidade da capacidade, é uma iniciativa da SEI (Software Engineering Institute, Carnegie Mello University, EUA). Este modelo de certificação serve para avaliar e melhorar a capacitação das empresas que produzem software.

De acordo com a jornalista Roberta Paduan da Revista EXAME, trabalhar dentro do padrão CMM significa seguir um guia que estabelece passos que visam aumentar a produtividade e eliminar, ou minimizar a existência dos defeitos no fim do desenvolvimento.

O modelo CMM dá maior ênfase na documentação dos processos para evitar a desorganização e a inexistência de padrões, visando uma organização sistemática dos níveis de maturidade da capacitação empresarial. (REZENDE, 2002).

Os objetivos do padrão CMM são auxiliar as empresas para conhecer melhor seu processo de software para, assim, desenvolver uma melhor manutenção deste; fornecer estrutura conceitual para que as empresas possam gerenciar e melhorar os seus processos de desenvolvimento de software.

O CMM está organizado em cinco níveis distintos e cada nível possui uma área chave do processo, exceto o primeiro nível.

No nível 1 (Inicial), ainda não há nenhuma metodologia adotada e tudo ocorre de forma desorganizada, o sucesso depende dos indivíduos e a introdução de novas tecnologias é arriscada.

No nível 2 (Repetitivo) o sucesso ainda depende das pessoas mas, agora passam a contar com o apoio gerencial. Os compromissos são compreendidos e gerenciados e as atividades bem definidas facilitam a introdução de novas tecnologias.

No nível 3 (Definido), os grupos já trabalham de forma coordenada, o treinamento é planejado de acordo com as necessidades de cada papel e aplicado de forma conveniente, novas tecnologias são avaliadas qualitativamente.

No nível 4 (Gerenciado), já começa a existir um forte sentido de trabalho em equipe, novas tecnologias são avaliadas quantitativamente.

No nível 5 (Otimizado), todos estão engajados em atividades de melhoria contínua, novas tecnologias são planejadas e introduzidas com total controle.

Haja vista que o modelo CMM enfatiza a organização dos processos através de documentação para melhor gerenciar mudanças com custos menores e melhor avaliação do processo, pode-se dizer que ele retrata as ações de planejar, desenvolver, gestionar e manter os processos de desenvolvimento e manutenção do de software. Essas atividades do CMM são similares ao processo de melhoria PDCA (planejar, executar e agir).

O método PDCA que se baseia no controle de processos, foi desenvolvido na década de 30 pelo americano Shewhart, mas foi Deming seu maior divulgador, ficando mundialmente conhecido ao aplicar nos conceitos de qualidade no Japão (MILET, 1993; BARRETO, 1999).

3 DESCRIÇÃO DA ORGANIZAÇÃO QUE SE ENCONTRA NO NÍVEL 1 DO MODELO CMM.

Descreveremos agora, resumidamente, as principais características da organização que se encontra no Nível 1 do CMM, chamado de Inicial, onde o desenvolvimento é caótico. Não existem procedimentos padronizados, estimativas de custos e planos de projeto. Cada qual desenvolve como quer, não existe documentação e não há mecanismos de controle que permitam ao gerente saber o que está acontecendo, identificar problemas e riscos e agir de acordo. Como

conseqüência, os desvios não são corrigidos e ocorrem os problemas que todos conhecemos muito bem: prazos não cumpridos, orçamentos estourados, software sem qualidade e usuários insatisfeitos. Na verdade, raramente existe um cronograma ou um orçamento. Infelizmente, estima-se que mais de três quartos das empresas norte-americanas encontram-se neste nível, e não há razões para acreditar que a situação seja melhor no Brasil (BELLOQUIM, 1997).

No endereço eletrônico <ftp://ftp.sei.cmu.edu/pub/cmm/cmm-over.pdf> do instituto SEI temos como características do Nível 1 “*The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics*”, o qual define o Nível Inicial como sendo um nível onde os trabalhos vão sendo realizados normalmente e são desenvolvidas medida extras quando do surgimento dos problemas, além do que os processos são caracterizados como *ad-hoc*, muitas vezes caótico e o sucesso depende do individualismo dos desenvolvedores, assim os mesmos são considerados *heróis* pelo seus esforços.

Neste nível, a organização, de um modo geral, não fornece um ambiente estável para desenvolvimento e manutenção de software. Quando uma organização não dispõe de práticas de gestão bem estabelecidas, os benefícios das boas práticas de desenvolvimento de software são minados por um planejamento ineficiente e por um contexto onde os compromissos são sempre reativos. Entende-se por compromissos reativos aqueles que são firmados como reação a algum acontecimento não previsto no planejamento (CMU/SEI-93-TR-24-CMM V1.1).

Em meio a uma crise, os projetos tipicamente abandonam os procedimentos que foram planejados e partem para a codificação e testes. O sucesso depende inteiramente de se ter um gerente excepcional e uma equipe de software madura e eficiente (CMU/SEI-93-TR-24-CMM V1.1).

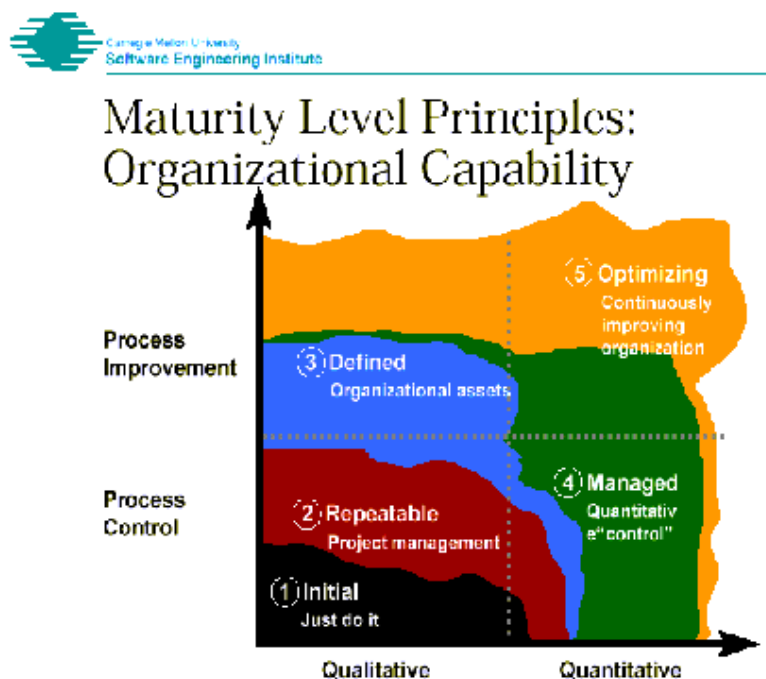
Ocasionalmente, gerentes de software eficientes, que dispõem de poder, podem resistir às pressões de recorrer a atalhos no processo de software, contudo, quando deixam o projeto, a influência estabilizadora se vai com eles. Mesmo um forte processo de desenvolvimento não pode superar a instabilidade criada pela ausência de práticas sólidas de gestão (CMU/SEI-93-TR-24-CMM V1.1).

A *capabilidade* de processo de software em organizações de Nível 1 é imprevisível porque o processo de software é constantemente alterado à medida que o trabalho progride (ou seja, o processo é “*ad hoc*”). Cronogramas, orçamentos,

funcionalidades e qualidade do produto são geralmente imprevisíveis. O desempenho depende da capacidade dos indivíduos e varia com as suas habilidades, conhecimentos e motivações inatas. Raros são os processos de software estáveis em evidência e o desempenho só pode ser previsto através de habilidades individuais e não por meio da capacidade da organização (CMU/SEI-93-TR-24-CMM V1.1).

Podemos ver claramente no gráfico da fig.01 que o nível de maturidade 1 significa dizer que a organização possui o mais baixo índice de qualidade e controle dos processos. É uma fase em que a empresa realiza muita das tarefas à medida que o desenvolvimento do produto vai requerendo “*just do it*”. Esta é a fase em que se encontra a maioria das organizações e o objeto de estudo deste artigo é justamente o fato da empresa tirar o “*pe*” deste nível para atingir a fase 2 do modelo CMM, chamada de “*repetable*” em que os processos serão mapeados e coordenados de forma sistêmica, pois o problema está justamente em mudar a forma de trabalhar de toda a organização, eliminando praticamente o trabalho individual pelo trabalho em grupo e controlado.

Figura 01 – Maturity Level Principles



Fonte - <ftp://ftp.sei.cmu.edu/pub/>

4 O CAMINHO PARA CHEGAR NO NÍVEL 2 DE MATURIDADE

Para passar ao nível 2, a organização deve instituir controles básicos de projeto, incluindo o Gerenciamento de Projetos (técnicas para planejar e estimar o esforço em projetos, e controlar o progresso), Controle Gerencial (verificação pela Gerência do progresso do projeto em momentos pré-determinados, incluindo a qualidade dos produtos), a instituição de um Grupo de Controle de Qualidade e de procedimentos básicos de Controle de Mudanças (para garantir que mudanças no projeto e manutenções solicitadas não destruam o que já foi feito, garantindo um mínimo de estabilidade no desenvolvimento; nada é mais deletério para um projeto do que requerimentos que mudam constantemente e sem controle) (BELLOQUIM, 1997).

E para atingir o segundo nível do modelo CMM, a empresa deve atuar nas *áreas chaves do processo – KPAs* relacionadas com este nível a fim de garantir total controle do desenvolvimento de projetos e este processo possa ser repetido nos outros projetos que virão, daí o nome do nível ser *Repetível*.

Cada KPA, portanto, identifica um grupo de atividades que tomadas em conjunto, permitem alcançar uma série de objetivos importantes para se obter uma melhoria de processo e como vimos no desenvolvimento do conceito do CMM as áreas chaves do nível 2 são seis a saber: gerenciamento de requisitos; planejamento de projetos; acompanhamento e supervisão de projetos; gerenciamento de sub contratação; garantia de qualidade de software e gerenciamento de configuração. E o objetivo desta gestão é estabelecer um entendimento comum entre o cliente e o projeto de software no qual os requisitos do cliente são usados como referência (CMU/SEI-93-TR-25-CMM-Nível 2 – V1.1).

Vemos aqui que os empregados da organização passarão a trabalhar de forma diferente, que cada passo deverá ser mapeado e registrado, que ninguém trabalhará de maneira própria, pois existirá um planejamento para cada ação, que o cliente deverá atuar de forma mais participativa, enfim são esses e outros efeitos que o modelo CMM influenciará no início de sua implantação “Nível 2” dentro da organização, levando esta a entrar numa fase de mudança organizacional.

5 MUDANÇA ORGANIZACIONAL

Em tempos modernos as empresas têm que mudar de paradigma, assim uma empresa que se situa no nível 1 representada por uma visão orientada por produto e pensa atingir o nível 2 que representa o início de uma visão orientada por processo no modelo CMM, significa que as empresas têm que trabalhar na elaboração de métodos de desenvolvimentos e avaliação, dando prioridade para essa nova visão

Na história da administração existe a *Teoria do Campo de Forças* que parte do princípio de que quando uma mudança é introduzida na organização, algumas forças a impulsionam, enquanto outras forças levam a resistência. E que para implementar a mudança, torna-se necessário analisar ambas as forças que atuam nesta mudança a fim de se remover ou neutralizar as forças restritivas que a impedem de evoluir e incentivar as forças impulsionadoras, sendo que mudanças somente ocorrem quando as forças impulsionadoras e favoráveis forem maiores do que as forças restritivas e impeditivas (CHIAVENATO, 1999).

Já a *Psicologia Social* diz que o indivíduo como ser humano pode modificar suas atitudes a partir de novas informações, novos afetos, novos comportamentos ou situações. Afirmando que o ser humano muda quando descobre que as mudanças lhes farão bem (BOCK, FURTADO, TEIXEIRA, 1999).

Pode-se afirmar que o indivíduo tem uma tendência a não mudar, principalmente pela falta de certeza do que está por vim, do que pela mudança em si, mas o ambiente deixa claro que as coisas precisam se ajustar. Existe um velho provérbio de autor desconhecido que diz “*o homem não quer mudar, mas quer mudanças*”, na realidade o homem não quer mudar, mas quer que as coisas melhorem.

Como vimos o meio ambiente está em constante mutação e as organizações não podem ficar estáticas observando as coisas acontecerem. No mundo de modo geral, a tecnologia encurtou as distâncias e fomentou a concorrência de maneira que a sobrevivência das empresas está no fato de serem mais competitivas. Fazendo analogia podemos dizer que nesta visão a lei da natureza é transportada para o mundo dos negócios, onde sobrevive quem tiver condições de atrair o mercado oferecendo um *mix* de produtos e serviços que sejam mais vantajosos para a própria empresa e para o mercado, sobrevive quem for melhor, quem for mais forte.

Este é o fato das empresas estarem sempre mudando em busca de algo melhor para si e para seus clientes e este ambiente organizacional em constante mutação tem que ser bem planejado internamente para que a organização funcione como um todo, uma engrenagem avançando para o futuro.

Para que tudo isso funcione, os dirigentes das organizações devem estar cientes de que a empresa é composta por um conjunto de colaboradores e que a *cooperação* é o elemento essencial da organização...como a *cooperação* é essencial para a sobrevivência da organização, o executivo deve tomar decisões que incentivem a coordenação das atividades organizadas (CHIAVENATO, 1999).

Assim quando uma empresa resolve implantar um modelo de qualidade do gerenciamento do processo de desenvolvimento de software, ela deve considerar que a mudança irá atingir toda organização e que as ações devem levar em conta que todos devem trabalhar em prol de um objetivo comum e como considerado em (ALBETIN, 2001),

O primeiro fator crítico de sucesso – FCS identificado na função organização é o estabelecimento de uma estrutura organizacional bem definida em nível de responsabilidades, atribuições, canais de comunicação e participação ativa de todos os níveis.

E para que a empresa consiga atingir seus objetivos ela deve saber trabalhar a motivação e envolvimento de seus empregados como mencionado na introdução deste artigo.

6 ESTUDO DE CASO

Originalmente, o estudo de caso foi projetado para que, através de um experimento controlado, fossem feitas averiguações sobre a efetividade de um processo de software. Em termos gerais, a proposta visava a obtenção de um **critério de decisão** voltado à implementação de um processo de software gerenciável. Os objetivos específicos seriam três, a saber:

- o teste de um processo de software voltado ao controle estatístico e melhorias contínuas, em um ambiente orientado aos modelos tradicionais e estáticos;

- a definição de um critério de medição da produtividade que permitisse comparações antes e depois da implantação do processo de software;
- a medição do efeito da implantação do processo de software sobre a produtividade e qualidade software.

O experimento definiu-se pela aplicação do nível 2 do Modelo CMM (PAULK, 1995). Um pré teste, em duas equipes consideradas de bom nível e uma entrevista com os responsáveis pelo gerenciamento dos métodos de Engenharia de Software, permitiram o enquadramento da organização no nível 1. Definiu-se, também, as equipes, das quais seria escolhida uma, para formar o Grupo de Engenharia de Software - GESw a ser capacitado nas práticas do nível 2 e os grupos de controle.

O plano de trabalho era bastante simples. O fato de ser apenas uma equipe a ser capacitada e abstrair as relações com o restante da organização, possibilitava trabalhar com os materiais originais do Modelo CMM e dispensar a elaboração de um manual de normas. Sinteticamente, a programação começou a ser cumprida visando cinco passos:

- apresentação do plano de trabalho para a aprovação da direção da organização;
- aferição da capacitação dos grupos de software;
- treinamento do GESw para ganhar proficiência nos modelos de qualidade e de maturidade em software;
- acompanhamento das equipes;
- medições finais e análise dos resultados.

O trabalho deveria ser concluído em 4 meses e meio. Duas questões externas ao experimento inviabilizaram as condições de sua realização. A primeira refere-se à preservação de vínculos sistêmicos com funções da organização, capazes de neutralizar as melhorias aplicadas às práticas do processo de software. A organização estava promovendo um grande esforço de nivelamento de todas as suas áreas, para implantação de uma sistemática de planejamento e controle de projetos. Nas equipes de software, o processo de estimativa não passava de uma

referência com o qual havia pouco ou nenhum comprometimento. Ademais, não havia um processo sistemático de realimentação dos dados históricos.

O ciclo de vida dos projetos não estavam completamente delineados, de modo que as fases não eram visíveis ou a ordem da especificação antecedendo o código, pouco clara. As negociações das alterações nos projetos com os grupos afetados e os riscos, não eram documentados, dificultando o comprometimento dos envolvidos na implementação do projeto. Havia um clima organizacional de baixo envolvimento com a necessidade do controle e da prestação de contas. Este ambiente, de ausência de disciplina de projeto, estava refletido nos novos procedimentos de planejamento em implantação, que registravam uma parcela significativa de atrasos, quase sempre justificados por fatores externos aos projetos. O GESw e os grupos de controle precisariam ter sido insulados em relação a este ambiente e não foram.

A outra questão foi de natureza motivacional. Dois episódios relativos à administração de pessoal, aparentemente sem relação de causa e efeito entre eles, tornaram desiguais as condições entre os grupos de software, o que impediria a comparação dos resultados das equipes. Houve uma avaliação de desempenho funcional e uma greve, no período do experimento.

Poder-se-ia argumentar, uma vez que o episódio teria atingido a todos, que as condições de igualdade foram mantidas. Ocorre que experimentos envolvendo pessoas trazem sempre uma carga de subjetividade. Em verdade, todo experimento, por causa da presença do observador.

(...) o corte cartesiano entre o observador e o observado, entre realidade externa e interna, (...) está baseado na ilusão que o mundo físico não tem componente subjetivo. Esta ilusão surge do alto grau de confiabilidade quantitativa das afirmações científicas sobre o mundo exterior, físico. Sua confiabilidade quantitativa nos faz esquecer que estas afirmações estão tão relacionadas com experiências subjetivas como as afirmações sobre o mundo interior, mental. (DELBRUK APUD SOARES, 1993, p. 133)

O argumento para sustentar a decisão de abandonar o experimento é de que as condições tornaram-se desiguais porque os indivíduos componentes dos grupos de software respondem às modificações do ambiente de forma diferente, como sujeitos, com diferentes experiências subjetivas. As condições somente teriam sido iguais se o ambiente externo permanecesse sem alteração, mantendo a possibilidade de controle do experimento, não porque as experiências subjetivas

fossem eliminadas, mas pela existência de uma intersubjetividade nos grupos formada no passado, que o novo processo de software desenvolveria. A intersubjetividade foi desfeita (ou reconstruída), como uma espécie de quebra de acordo, pela visão (subjetiva) de ruptura dos compromissos, emanada dos episódios que provocaram muita tensão: a avaliação de desempenho, com a exclusão da maioria das pessoas dos seus benefícios, e a greve, com a suspensão temporária dos contratos.

Não obstante sua interrupção, o experimento não deixa de apresentar um resultado. Não é possível afirmar, baseado nessa experiência, que este tipo de experimento não possa ser empregado, sob condições adequadas de isolamento. Por outro lado, a situação confirma a afirmação de HUMPHREY [1989], de que a capacitação deve ocorrer para toda a organização, sob pena de não ser mantida.

Esse estudo de caso foi realizado por Gustavo da Gama Torres do curso de pós graduação do Departamento de Ciência da Computação da UFMG-ICEX usado para análise de alguns métodos de processo de software em um ambiente do setor público e que apesar de ter sido abortado servirá de subsídio para análise e referência como prova da necessidade do envolvimento de todos os funcionários da organização e necessariamente o pessoal da alta administração, bem como o fator motivação do grupo e variáveis externas devem ser analisadas para que um plano de ação estabeleça um equilíbrio no desenvolvimento do processo que é o objetivo deste artigo.

7 CONCLUSÃO

O início do caminho para o nível 2 parte de um grande esforço para chamar a atenção para os problemas de software internamente. A partir de uma avaliação formal ou informal é necessário fazer com que todos dentro da empresa comprem a idéia da melhoria, de cima a baixo. Isto inclui a alta administração (diretoria), profissionais de informática (gerentes, analistas programadores, AD, DBA, produção, suporte etc. etc. etc.) e usuários (gerência e pessoal operacional). Isto implica na necessidade de se constituir uma disciplina de comprometimento (LEAL, LUZ, 2003).

Toda organização possui uma metodologia própria de trabalho e quando há necessidade de desenvolvimento organizacional no seu processo, o que implica em mudanças, tem que haver a preocupação em toda sua estrutura e nos aspectos

subjetivos que estão envolvidos, não bastando apenas a vontade de mudar aliada ao esforço de parte da empresa.

Como vimos no estudo de caso, tentar mudar a filosofia de trabalho de apenas parte do grupo sem considerar fatores que envolvem a empresa como um todo, fatalmente direcionará o projeto a uma probabilidade maior de insucesso no seu objetivo final.

Não dá para fazer com que dentro de um grupo, todos aceitem uma mudança de imediato, por isso é necessário realizar um plano de ação que possa abranger desde os aspectos motivacionais individuais até o envolvimento e comprometimento de todos.

Para tanto, deverá ser feito um trabalho de conscientização da importância dessas mudanças, através de uma comunicação aberta, treinamentos, eventos e etc., para que todos enxerguem que os benefícios serão tanto para a organização, em forma de produtividade e competitividade, quanto para o indivíduo, no que tange ao fator subjetivo de realizar um trabalho bem feito com menor índice de retrabalho.

Consideramos que implantar o modelo CMM representa mexer com certos valores na organização que não são mensuráveis, mas que influenciam substancialmente no sucesso da implementação do mesmo. Por isso, ao pesquisarmos a saída de uma organização que trabalhava de forma caótica para o primeiro nível dentro do modelo CMM, o qual o seu processo passa a ter um controle mais efetivo, percebemos que, além dos diversos fatores como o custo de implementação e ferramentas necessárias para controle, é primordial que haja uma preocupação com o aspecto que envolve o fator humano inerente em qualquer processo de mudança organizacional.

Sugerimos estudos dos impactos do ambiente externo e sua influência na organização durante essas mudanças e os reflexos dos custos de implementação do modelo CMM, como fator de retração de mercado para sua adoção.

REFERÊNCIA

BARTIÉ, Alexandre. **Garantia da Qualidade de Software**, Rio de Janeiro: Campus, 2002, 291p.

FIORINI, Soeli T. et al. **Engenharia de Software com CMM**, Rio de Janeiro: Brasport, 1998.

ROCHA, Ana Regina Cavalcanti da, MALDONADO, José Carlos e WEBER, Kival Chaves. **Qualidade de Software – Tema e Prática**, São Paulo: Prentice Hall, 2001.

ALBERTIN, Alberto Luiz. **Administração de Informática**, São Paulo: Atlas, 2001.

LAUDON, Kenneth C. e LAUDON, Jane Price. **Sistema de Informação com Internet**, Rio de Janeiro: LTC, 1999, 4ªed.

INTHURN, Cândida. **Qualidade e Teste de Software**, Santa Catarina: Visual Book, 2001.

CHIAVENATO, Idalberto. **Introdução à Teoria Geral da Administração**, Rio de Janeiro: Campus, 1999 – 3ªed. compacta.

BOCK, Ana Mercês Bahia, FURTADO, Odair e TEIXEIRA, Maria de Lourdes Trassi. **Uma Introdução ao Estudo de Psicologia**, São Paulo: Saraiva, 1999 – 13ªed.

REZENDE, Denis Alcides. **Engenharia de Software e sistemas de informações**, Rio de Janeiro: Brasport, 2002 – 2ª ed, 358p.

PADUAN, Roberta. **A Volta da Qualidade**, Exame, Ano 37, n.º 01, Janeiro de 2003, pag 64-66.

CORDEIRO, Marco Aurélio. **Modelo de Qualidade de Desenvolvimento de Software**, Artigo de Direitos Autorais da Companhia de Informática do Paraná - CELEPAR em:

<http://www.pr.gov.br/celepar/celepar/batebyte/edicoes/2000/bb99/modelos.htm - 08/04/2003>.

BELLOQUIM, Átila. Artigo: **O Modelo SEI/CMM e o Processo de Desenvolvimento de Software**, Revista: Developers Magazine, 03/1997.

CMU/SEI-93-TR-24-CMM V1.1, **documento oficial do instituto SEI sobre o modelo CMM V1.1** no site <ftp://ftp.sei.cmu.edu/pub/cmm/cmm-over.pdf> em 13/04/2003.

CMU/SEI-93-TR-25-CMM-Nível 2 – V1.1, **documento oficial do instituto SEI sobre o modelo CMM Nível 2 - V1.1** no site <ftp://ftp.sei.cmu.edu/pub/cmm/cmm-over.pdf> em 13/04/2003.

TORRES, Gustavo da Gama. **O Problema de Instabilidade da Organização, no Estudo de Caso Sobre a Transição para Processos de Software Orientados à Melhoria Contínua, em Ambiente do Setor Público**, em: <http://www.dcc.ufmg.br/pos/html/spg2000/anais/gustavo/gustavo.html> - 18/04/2003.

LEAL, Luciana Nunes e LUZ, Cláudio de Miranda. Artigo: **Qualidade de Software – CMM (Capability Maturity Model)**, em http://www.dcc.ufrj.br/~schneide/es/2000/1/trab/al11_19.htm - 13/04/2003.

HUMPHREY, W. S. (1989) **Managing the Software Process**. Addison-Wesley.

PAULK, M. C. org. (1995) **The Capability Maturity Model: Guidelines for Improving the Software Process**. Addison-Wesley.

SOARES, J. O. P. (1993) **Fundamentos Conceituais Para Projeto de Aplicações de Computadores**. USP (Tese de Doutorado).