

LEANDRO ANDRÉ ASCENÇÃO

**CONTROLE DE QUALIDADE DE SOFTWARE
E GERÊNCIA DE PROJETOS PARA
PEQUENAS EMPRESAS**

Monografia apresentada à Escola
Politécnica da Universidade de São
Paulo, PECE – Programa de Educação
Continuada em Engenharia, para
obtenção do título de MBA –
Engenharia de Software.

**São Paulo
2002**

LEANDRO ANDRÉ ASCENÇÃO

**CONTROLE DE QUALIDADE DE SOFTWARE
E GERÊNCIA DE PROJETOS PARA
PEQUENAS EMPRESAS**

Monografia apresentada à Escola
Politécnica da Universidade de São
Paulo, PECE – Programa de Educação
Continuada em Engenharia, para
obtenção do título de MBA –
Engenharia de Software.

Área de Concentração:
Engenharia de Software.

Orientador:
Prof. Dr.
Keichi Hirama

**São Paulo
2002**

FICHA CATALOGRÁFICA

Ascensão, Leandro André

Controle de qualidade de software e gerência de projetos para pequenas empresas. São Paulo, 2002.

60p.

Monografia – Escola Politécnica da Universidade de São Paulo. PECE - Programa de Educação Continuada em Engenharia.

1. Engenharia de Software 2. Controle de Projetos
1. Universidade de São Paulo. Escola Politécnica.
PECE - Programa de Educação Continuada em Engenharia.

Dedico este trabalho a todas as pessoas que têm um espírito empreendedor e que com muita coragem e força de vontade se dedicam às pequenas empresas.

AGRADECIMENTOS

Agradeço aos professores do curso de MBA em Engenharia de Software, do PECE-USP, pelo conhecimento adquirido ao longo do curso, sem o qual não seria possível realizar este trabalho, em especial ao meu orientador Prof. Dr. Keichi Hirama.

Agradeço aos meus sócios e amigos da Guest Soluções em Informática, pequena empresa de software em que trabalho, pelo apoio e compreensão, do tempo necessário para a realização do curso de MBA e para escrever este trabalho.

Por fim, um agradecimento especial aos meus pais, a Catarina minha noiva, e aos meus amigos, por entenderem a minha falta de tempo.

RESUMO

Este trabalho apresenta as principais características das pequenas empresas de software e como elas podem obter melhores resultados, seguindo os principais métodos e boas práticas de gerência de projeto e qualidade de software. São apresentados de forma sumária, os conceitos de qualidade de software e gerência de projetos, com um foco especial nos modelos CMM e PMBOK. Por fim, é apresentado um exemplo fictício de um projeto de desenvolvimento de software realizado por uma pequena empresa, onde é comentada na prática a importância da utilização dos modelos apresentados.

ABSTRACT

This report shows the concepts of small business organizations and how they can achieve better results, following the best practices and methods for project management and software quality assurance. The concepts of software quality assurance and project management are introduced, with a focus on CMM and PMBOK. Last, there is an example of a software project, developed by an imaginary small business organization, where the practical utilization of the concepts mentioned before are commented.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

RESUMO

ABSTRACT

1	INTRODUÇÃO	01
1.1	Objetivos	01
1.2	Motivações	01
1.3	Estrutura do Trabalho	02
2	A PEQUENA EMPRESA DE SOFTWARE	04
2.1	A definição de pequena empresa	04
2.2	A importância das pequenas empresas	05
2.3	Principais Características	06
3	QUALIDADE DE SOFTWARE	09
3.1	Definições de qualidade de software	09
3.2	Tópicos de Qualidade de Software	11
3.2.1	Garantia da Qualidade de Software (SQA)	11
3.2.2	Verificação e Validação (V&V)	11
3.2.3	Métricas de Software	12
3.2.4	Sistemas Confiáveis (Dependable Systems)	12
3.3	O modelo CMM de qualidade de software	12
3.3.1	Definição	12
3.3.2	Motivação	13
3.3.3	Estrutura	13
3.3.4	O CMM nas empresas brasileiras	16
4	GERÊNCIA DE PROJETOS	17
4.1	Definição de projeto	17
4.1.1	Características do projeto de software	18
4.1.2	Definição de produtos ou serviços únicos	19
4.1.3	Definição de processos de projeto	20

4.2	Gerenciamento de projetos	21
4.3	O PMBOK	21
4.3.1	Grupos de Processos	24
5	GERÊNCIA DE PROJETOS E QUALIDADE DE SOFTWARE PARA PEQUENAS EMPRESAS DE SOFTWARE	27
5.1	Cenário de uma pequena empresa	29
5.2	Início do desenvolvimento de um projeto	29
5.3	Execução e controle do desenvolvimento de um projeto	35
5.4	Finalização e entrega de um projeto	40
5.5	Manutenção do sistema	41
6	CONCLUSÃO	43
6.1	Contribuições do trabalho	43
6.2	Trabalhos futuros	44
7	REFERÊNCIAS BIBLIOGRÁFICAS	45

LISTA DE FIGURAS

Figura 2.1	Gráfico com a porcentagem de fechamento de empresas ao longo dos três primeiros anos de existência.	06
Figura 3.1	Os níveis do modelo CMM.	13
Figura 3.2	Estrutura do CMM.	14
Figura 3.3	Conhecimento do Modelo CMM no Brasil.	16
Figura 4.1	Interligações de processos no PMBOK	25
Figura 4.2	Gráfico de atividades por tempo, nos processos do PMBOK	25
Figura 4.3	Interação dos grupos de processo do PMBOK	26

LISTA DE TABELAS

Tabela 2.1	Classificação das empresas segundo o número de empregados	04
Tabela 2.2	Classificação das empresas segundo o faturamento bruto anual	04
Tabela 2.3	Subclassificação de pequenas empresas de software	05
Tabela 3.1	Modelo de qualidade proposto pela norma ISO/IEC 9126(NBR 13596)	10

LISTA DE ABREVIATURAS E SIGLAS

ADM	Arrow Diagraming Method
CMM	Capability Maturity Model
DEM	Diagrama de Estrutura de Módulos
DER	Diagrama de Entidade Relacionamento
DFD	Diagrama de Fluxo de Dados
EAP	Estrutura Analítica do Projeto
EV	Empresa Virtual
FIPE	Fundação Instituto de Pesquisas Econômicas
IEC	International Electrotechnical Commission
IFPUG	International Function Point Users Group
ISO	International Organization for Standardization
KPA	Área Chave de Processo (Key Process Area)
NASDAQ	National Association of Securities Dealers Automated Quotations
NBR	Normas Técnicas Brasileiras
OV	Organização Virtual
PBQP	Programa Brasileiro da Qualidade e Produtividade
PDM	Precedence Diagraming Method
PF	Pontos de Função
PIP	Programa de Melhoria do Processo (Process Improvement Program)
PMBOK	Corpo de Conhecimento do Gerenciamento de Projetos (Project Management Body of Knowledge)
PME	Pequena e Média Empresa
PMI	Project Management Institute
PMI-SP	Project Management Institute – São Paulo
SEI/SEBK	Corpo de Conhecimento de Engenharia de Software Software Engineering Body of Knowledge
SEBRAE	Serviço de Apoio às Micro e Pequenas Empresas
SEBRAE-SP	Serviço de Apoio às Micro e Pequenas Empresas de São Paulo

SEI	Software Engineering Institute da Carnegie Mellon University
SQA	Garantia da Qualidade de Software (Software Quality Assurance)
UML	Unified Modeling Language
V&V	Verificação e Validação
WBS	Work Breakdown Structure (ver EAP)

1. INTRODUÇÃO

1.1 Objetivos

Este trabalho apresenta as principais características das pequenas empresas de software e como elas podem obter melhores resultados, seguindo os principais métodos e boas práticas de gerência de projeto e qualidade de software.

As soluções apresentadas neste trabalho não têm a finalidade de mostrar como as pequenas organizações devem trabalhar para conseguir certificações formais dos modelos apresentados, e sim apresentar conceitos de como elas podem melhorar seu controle de qualidade e gerência de projetos de software, respeitando as suas características.

1.2 Motivações

O principal motivo para a realização desse trabalho foi verificar uma dificuldade das pequenas empresas na aplicação efetiva de técnicas de gerenciamento de projetos e qualidade de software. Apesar de muitas empresas conhecerem os principais padrões e modelos de qualidade de software e gerência de projeto como ISO, CMM, PMBOK eles não são usados no dia a dia das empresas.

Alguns motivos apresentados para essas dificuldades foram: o tempo necessário para seguir a realização de todos os passos desses modelos; o tempo e custo de treinamento de pessoal no conhecimento e aplicação desses modelos; o tempo e custo para conseguir certificações formais, como o CMM para a empresa, ou pessoais como as concedidas pelo PMI ou pelo IFPUG; entre outros motivos.

Essa motivação levou a uma procura por soluções que levassem em consideração o número reduzido de pessoas nas empresas, o tempo curto de desenvolvimento praticado nos projetos, a sobrecarga de funções realizadas pelos funcionários, as barreiras culturais com relação às metodologias de trabalho e orçamentos reduzidos,

ou seja, que levassem em consideração as características típicas de pequenas empresas.

1.3 Estrutura do Trabalho

A estrutura deste trabalho está dividida em seis capítulos:

O capítulo 1 tem como objetivo apresentar uma visão geral, com os seus objetivos, motivações e estrutura do trabalho.

No capítulo 2 são apresentados os conceitos de pequena empresa de software, as formas de como classificar as pequenas empresas, a importância das pequenas empresas, em especial para o mercado brasileiro, as características típicas de uma pequena empresa, como a forma de trabalho, necessidades, desafios, vantagens e desvantagens em relação as empresas de maior porte.

No capítulo 3 descreve-se o conceito de qualidade de software que será utilizado neste trabalho e a importância da garantia dessa qualidade para as empresas. Em particular será apresentado o modelo de controle e garantia de qualidade de software da SEI/CMM.

No capítulo 4 descreve-se o conceito de gerência de projetos de software que será utilizado neste trabalho e a importância dessa gerência para as empresas. Dentre os modelos de controle e coleções de boas práticas de gerência de projetos atuais, será apresentado o PMBOK.

No capítulo 5 descreve-se um exemplo de projeto numa pequena empresa, utilizando os conceitos vistos nos capítulos anteriores. Este exemplo mostra a importância de utilizar as boas práticas tanto do modelo CMM quanto do PMBOK, para as pequenas empresas de software.

Por fim no capítulo 6 descreve-se a conclusão desse trabalho, com contribuições, dificuldades e pontos que devem ser aprimorados ou que mereçam estudos a parte.

2. A PEQUENA EMPRESA DE SOFTWARE

O objetivo desse capítulo é apresentar o conceito de pequena empresa de software, mostrando quais são as classificações usuais, tanto por número de funcionários quanto por faturamento, qual a importância das pequenas empresas, com dados estatísticos do mercado brasileiro e internacional, além das características típicas de uma pequena empresa, como a forma de trabalho, necessidades, desafios, vantagens e desvantagens.

2.1 Definição de pequena empresa

Uma das primeiras dificuldades de um trabalho sobre pequenas empresas é obter uma definição única de pequena empresa.

Segundo o (SEBRAE, 2001) as empresas são classificadas de acordo com o número de funcionários, como se pode observar na tabela 2.1.

Tabela 2.1 - Classificação das empresas segundo o número de empregados

Porte	Empregados
Microempresa	No comércio e serviços até 09 empregados
Empresa de Pequeno Porte	No comércio e serviços de 10 a 49 empregados
Empresa de Médio Porte	No comércio e serviços de 50 a 99 empregados
Empresa de Grande Porte	No comércio e serviços mais de 99 empregados

Fonte: (SEBRAE, 2001)

Segundo (LEI FEDERAL Nº 9841, “Estatuto da Micro e Pequena Empresa”), as micro e pequenas empresas são classificadas de acordo com o faturamento bruto anual, como mostra a tabela 2.2.

Tabela 2.2 - Classificação das empresas segundo o faturamento bruto anual

Porte	Faturamento Bruto Anual
Microempresa	Até R\$ 244.000,00
Empresa de Pequeno Porte	Entre R\$ 244.000,00 e R\$ 1.200.000,00

Fonte: (LEI FEDERAL Nº 9841)

Especificamente para empresas de software, o número de pessoas influencia no modo de trabalho da empresa, isto é, uma empresa com até três funcionários é muito menor e mais fácil de gerenciar do que uma empresa com aproximadamente 50 funcionários com diversos produtos, diversas versões de cada produto ou cliente ao mesmo tempo(LARYD; ORCI, 2000).

Essa percepção resultou em subclassificações do que é considerada pequena empresa, conforme apresentado na tabela 2.3.

Tabela 2.3 – Subclassificação de pequenas empresas de software

Porte	Empregados
Muito muito pequena (XXS)	Até 3 empregados
Muito pequena (XS)	De 3 à 15 empregados
Pequena	De 16 – 50 empregados

Fonte: (LARYD; ORCI, 2000)

Nesse trabalho, será considerada pequena empresa a organização que possuir até 50 empregados, com as ressalvas da tabela 2.3, para análises específicas de cada caso.

2.2 Importância das pequenas empresas

As pequenas empresas têm um papel marcante na economia nacional. “Uma sondagem realizada Serviço Brasileiro de Apoio às Micro e Pequenas Empresas – SEBRAE (2000) demonstrou que estas empresas representam 95% do total dos estabelecimentos industriais, 98% dos comerciais e 99% dos estabelecimentos do setor de serviços. ALVIM (1998) afirma que as empresas de pequeno porte no Brasil são responsáveis por cerca de 4 milhões de empresas constituídas; 60% da oferta total de empregos; 21% da participação no PIB; 96,3% do número de estabelecimentos.” (OTTOBONI; PAMPLONA)

Nos Estados Unidos, de acordo com o US Bureau of the Census 1995 County Business Patterns, aproximadamente 90% das empresas de software e processamento de dados dos Estados Unidos têm menos que 50 empregados.(LAITINEM et al, 2000)

Mas, provavelmente, é no setor tecnológico que a influência das pequenas empresas foi mais sentido nos últimos anos, principalmente levando em consideração a revolução que a Internet realizou na economia global, com as pequenas empresas do vale do silício atuando ativamente na bolsa de valores do setor de tecnologia dos Estados Unidos, a NASDAQ, criando o movimento conhecido como a “bolha da Internet”, incorporando empresas e movimentando bilhões de dólares no mundo.

2.3 Principais Características

O desafio inicial das pequenas empresas é sobreviver, segundo estudos do (SEBRAE-SP, 2001), a taxa de mortalidade das pequenas empresas é muito elevada. Na figura 2.1 podemos perceber que 32% das empresas fecham no primeiro ano e mais de 50% das empresas fecham em até três anos, com uma sobrevivência de apenas 29% das empresas depois de cinco anos.

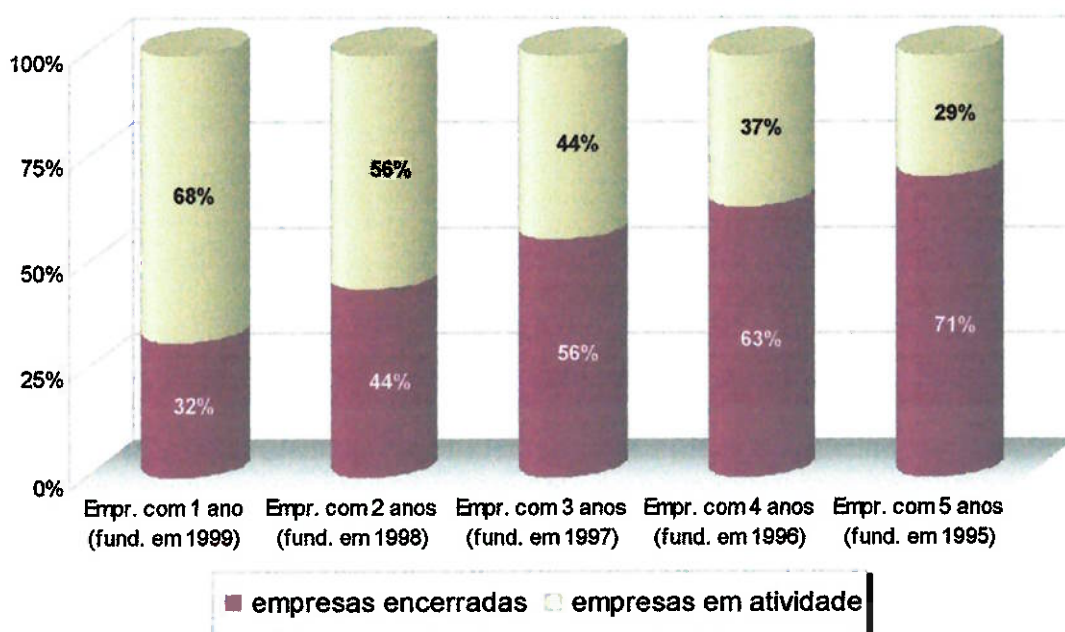


Figura 2.1 – Gráfico com Sobrevivência e mortalidade acumulada das empresas, no Estado de São Paulo (rastreamento realizado em nov/dez 2000).

Fonte: (SEBRAE-SP, 2001)

De acordo com (PAULK, 1998) as pequenas empresas tendem a acreditar que:

- Todos são competentes – As pessoas são contratadas para fazer o trabalho, e não podem realizar treinamentos devido aos custos com investimentos ou pelo tempo gasto.
- Todos se comunicam – Devido a proximidade dos e ao número reduzido de funcionários, a comunicação é praticamente instantânea, ou como define (PAULK, 1998), é praticamente feita por “Osmose”.
- Todos são “heróis” – Todos fazem o que precisar ser feito, vivem em um ambiente livre de regras rígidas, com ciclo de desenvolvimento curto e altas taxas de estresse.

Mesmo assim, as pequenas empresas, assim como as grandes, terão problemas relativos à falta de documentação dos requisitos, alocação de recursos, treinamento de pessoal, revisões e documentações de produtos e a erros devidos à falta de experiência dos gerentes.

Apesar desses desafios, as pequenas empresas podem ser extraordinariamente inovadoras e produtivas. Embora existam inúmeros problemas que requeiram uma grande quantidade de pessoas para resolver, em geral pequenos grupos são mais produtivos do que grandes grupos – principalmente devido à comunicação fácil entre os membros.

Uma outra característica das pequenas empresas é a tendência formação de círculos de parcerias.

De acordo com (BREMER; MUNDIM, 1999) “diversos fatores levam ao crescimento do número de cooperações entre empresas, como o rápido desenvolvimento de novas tecnologias de informação, o processo de globalização e a tendência das empresas concentrarem-se em suas competências essenciais (HAMUEL & PRAHALAD, 1994). O que demonstra que, em vista da crescente

competição internacional, as empresas estão investigando novos processos de integração, a fim de se tornarem competitivas e manterem sua posição competitiva. O foco nas competências essenciais leva as empresas a terem de cooperar para oferecer produtos e serviços completos ao mercado consumidor. Tal cenário abre novas perspectivas de crescimento para Pequenas e Médias Empresas (PMEs), às quais possuem uma estrutura mais ágil e, portanto, mais adequadas às características anteriormente citadas.

Neste ambiente de negócios, PMEs encaram o desafio de encontrar um modelo organizacional que lhes permita explorar oportunidades de negócios, inclusive globais, considerando seus recursos limitados e, sem a exposição ao risco de investimento direto.

Segundo BREMER et al. (1996), Empresas Virtuais (EVs) e sua forma de organização baseada em cooperação e competitividade podem ser assumidas como adequadas para atender estes requisitos, além de suportar a formação de empresas baseadas em alta tecnologia.

Para a formação de EV, pode-se partir da criação de uma Organização Virtual (OV), ou seja, a criação de um grupo de empresas predispostas a cooperarem, que ao surgir uma oportunidade de negócio, se unem e criam uma EV para explorar tal oportunidade (BREMER et al., 1999). Uma ou várias EVs são, portanto, uma consequência ou resultado da OV.”

3. QUALIDADE DE SOFTWARE

O objetivo desse capítulo é apresentar o conceito de qualidade de software que será utilizado nesse trabalho e a importância da qualidade para as empresas.

Existem atualmente diversos modelos e práticas de controle de qualidade, porém, o estudo detalhado de cada modelo ou prática foge do escopo deste trabalho, por isso será mostrada de forma sumária apenas o modelo CMM.

3.1 Definições de qualidade de software

Qualidade de software é um assunto que muito se discute e pouco se pratica.

Desde o início dos tempos do desenvolvimento de software temos problemas de qualidade que não só ainda não foram resolvidos como ainda parecem que se agravam a cada nova geração tecnológica. Prazos excedidos, baixa produtividade, custos altos e qualidade deficiente são alguns dos problemas constantes no desenvolvimento de software.(BELLOQUIM, 1997)

Para abordar a questão da qualidade de software é fundamental analisar a visão de qualidade de software dos principais participantes de um projeto. Para os desenvolvedores um software de qualidade é aquele em que projeto e os códigos fonte são bem estruturados. Já para os usuários é um produto que tenha bom desempenho. Na visão da gerência é um projeto realizado dentro de suas estimativas. Para o cliente é um produto que atenda os seus negócios.(HIRAMA,2002)

A tabela 3.1 mostra o modelo de referência de características de qualidade proposto pela norma ISO/IEC 9126 (NBR 13596), que tem por objetivo servir de referência básica na avaliação de produto de software.(GOMES)

Tabela 3.1 – Modelo de qualidade proposto ISO/IEC 9126(NBR 13596)

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	SIGNIFICADO
Funcionalidade O conjunto de funções satisfazem as necessidades explícitas e implícitas para a finalidade a que se destina o produto?	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Gera resultados corretos ou conforme acordados?
	Interoperabilidade	É capaz de interagir com os sistemas especificados?
	Segurança de acesso	Evita acesso não autorizado, acidental ou deliberado a programas e dados?
	Conformidade	Está de acordo com normas e convenções previstas em leis e descrições similares?
Confiabilidade O desempenho se mantém ao longo do tempo e em condições estabelecidas?	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas como ele reage?
	Recuperabilidade	É capaz de recuperar dados após uma falha?
Usabilidade É fácil utilizar o software?	Inteligibilidade	É fácil entender os conceitos utilizados?
	Aprendizado	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar a operação?
Eficiência Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto?	Comportamento em relação ao tempo	Qual é o tempo de resposta e de processamento?
	Comportamento em relação aos recursos	Quanto recurso utiliza?
Manutenibilidade Há facilidade para correções, atualizações e alterações?	Analísabilidade	É fácil encontrar uma falha quando ocorre?
	Modificabilidade	É fácil modificar e remover defeitos?
	Estabilidade	Há grandes riscos de <i>bugs</i> quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?	Adaptabilidade	É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado?
	Capacidade para ser instalado	É fácil instalar em outros ambientes?
	Capacidade para substituir	É fácil substituir por outro software?
	Conformidade	Está de acordo com padrões ou convenções de portabilidade?

Nesse trabalho, qualidade de software será considerada uma política de controle, que possibilite desenvolver softwares levando em consideração as características, visões e expectativas apresentadas acima.

3.2 Tópicos de Qualidade de Software

A seguir serão apresentadas de forma sumária as principais áreas de conhecimento relativas à qualidade de software, de acordo com (SEBK, 1999).

3.2.1 Garantia da Qualidade de Software (SQA)

Esta área é responsável por funções de gerenciamento de software e atividades cuja intenção é assegurar que um produto de software está em conformidade com os seus requisitos funcionais e de performance. Ela inclui conhecimento sobre a organização das unidades de garantia de qualidade, planejamento da qualidade, controle de registro, análises, auditoria, e relatos. Esta área inclui também técnicas de garantia de qualidade como análises de Pareto, análise de tendências, estatísticas de controle de qualidade e testes de regressão.

3.2.2 Verificação e Validação (V&V)

Esta área é responsável por conhecimentos sobre verificação e validação (V & V) com conceitos, métodos, atividades e produtos entregues associados com cada fase no ciclo de desenvolvimento do software. Esta área inclui conhecimentos de planejamento e organização de V&V; verificações (walkthroughs), e inspeções; análise de rastreamento; técnicas de verificação formal; e testes de software.

3.2.3 Métricas de Software

Esta área é responsável pela formulação de medidas e métricas de software, a coleção de dados necessária para as métricas formuladas, a computação de métricas, e a análise, interpretação e retorno dos resultados das métricas. Esta área envolve o conhecimento sobre as métricas do produto (requisitos, design e código), métricas de recursos (recursos humanos e técnicos), e métricas de processo (medição de esforço e cronograma).

3.2.4 Sistemas Confiáveis (Dependable Systems)

Esta área é responsável pelo conhecimento sobre o desenvolvimento de sistemas de software que precisam ser confiáveis (i.e., sistemas que possuem requisitos não funcionais críticos como confiabilidade, segurança e proteção). Esta área inclui conhecimentos sobre especificações de softwares confiáveis, métricas para verificar a confiabilidade, testes estatísticos, tolerância a falhas, programação de defesa e tratamento de exceções, e especificações de garantia de segurança.

3.3 O modelo CMM de qualidade de software

3.3.1 Definição

O modelo CMM (Capability Maturity Model) é baseado em uma coleção de boas práticas de gerenciamento e conceitos de qualidade no desenvolvimento e manutenção de software.

Esse modelo serve como um guia para a melhoria organizacional das empresas, classificando a maturidade de desenvolvimento em níveis e estabelecendo metas mensuráveis para avaliar em qual nível a empresa (ou setores da empresa) se encontra.(PAULK et al.);(PAULK, 1998).

3.3.2 Motivação

Segundo os idealizadores do CMM, após duas décadas de promessas não cumpridas sobre ganhos de produtividade e qualidade na aplicação de novas metodologias de software e tecnologia, as organizações estão concluindo que o seu principal problema é a inability de gerenciar o processo de desenvolvimento de software.

Em muitas organizações, os projetos com frequência atrasam e ultrapassam os orçamentos, e os benefícios de melhores métodos e ferramentas não podem ser usufruídos em um ambiente de indisciplina e projetos caóticos.

O CMM foi desenvolvido para guiar as empresas de software na seleção de estratégias de melhoria de processo de desenvolvimento através da determinação do processo de maturidade corrente e na identificação dos pontos críticos para a qualidade de software e melhoria do processo de desenvolvimento.(PAULK et al.)

3.3.3 Estrutura

No CMM foram estabelecidos 5 níveis referentes à maturidade que a organização possui para desenvolver software, que são: nível inicial, repetitivo, definido, gerenciado e otimizado.

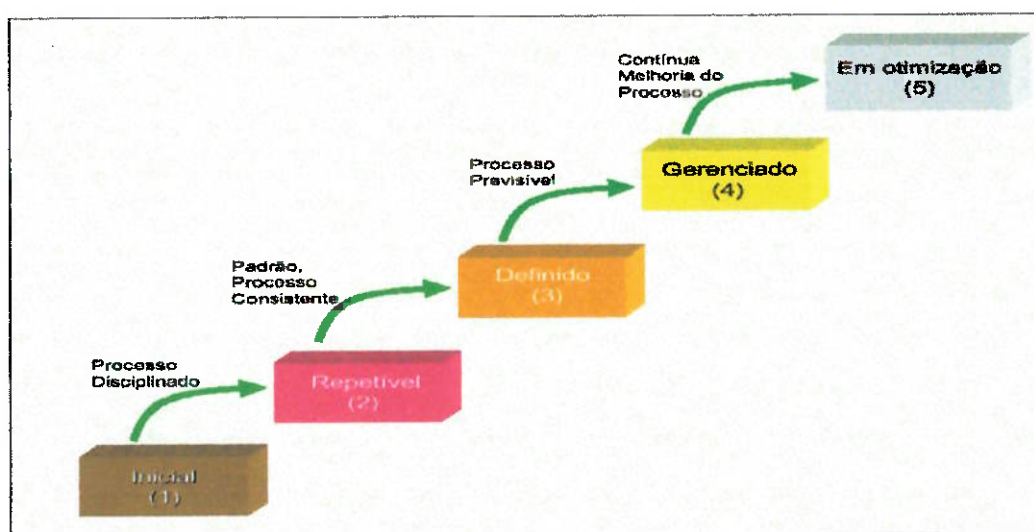


Figura 3.1 – Os níveis do modelo CMM.

Fonte: (PAULK et al.)

Cada nível é dividido em áreas chaves do processo (KPA) que estabelecem grandes temas a serem abordados. Cada área chave é detalhada em práticas chave e objetivos. O cumprimento desses objetivos determina o nível de maturidade corrente da organização.

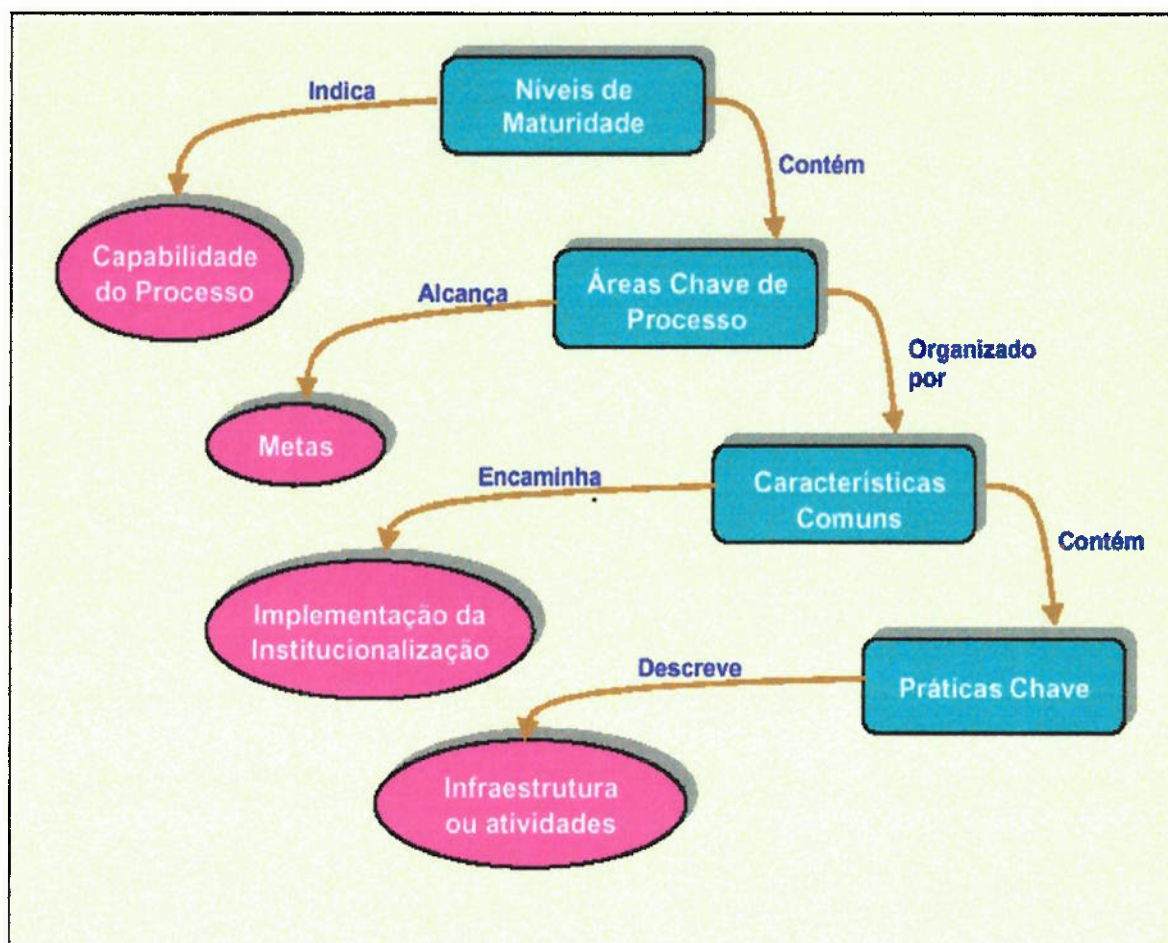


Figura 3.2 – Estrutura do CMM

Fonte: (PAULK et al.)

No **Nível 1** ou **Inicial**, praticamente não existe um processo de desenvolvimento prevalecendo o caos. Poucos processos são definidos e o sucesso depende de esforços individuais e heróicos. Nesse nível não existem áreas chave de processo a serem atendidas. (PAULK et al.)

No **Nível 2** ou **Repetitivo**, as políticas de gerenciamento de projeto e os procedimentos de implantação dessas políticas estão estabelecidos. Práticas bem sucedidas em projetos anteriores podem ser repetidas em outros projetos similares.

Um processo efetivo é caracterizado como um processo que seja praticado, documentado, treinado, mensurável e apto para melhorias.

No nível 2 existem seis KPA: gerenciamento de requisitos, planejamento de projeto de software, acompanhamento de projeto de software, gerenciamento de subcontrato de software, garantia da qualidade de software e gerenciamentos da configuração de software. (PAULK et al.)

No **Nível 3 ou Definido**, tanto as atividades de gerenciamento, quanto de engenharia, do processo de desenvolvimento de software estão documentadas, padronizadas e integradas em um padrão de desenvolvimento da empresa formando um processo padronizado e consistente.

No nível 3 existem sete KPA: enfoque no processo da organização, definição do processo da organização, programa de treinamento, gerenciamento integrado de software, engenharia de produto de software, coordenação intergrupos e revisões. (PAULK et al.)

No **Nível 4 ou Gerenciado**, são coletadas medidas detalhadas da qualidade do produto e processo de desenvolvimento de software. Essas medidas estabelecem uma base quantitativa para evolução do processo de software e dos produtos.

No nível 4 existem duas KPA: gerenciamento quantitativo do processo e gerenciamento da qualidade de software. (PAULK et al.)

No **Nível 5 ou Otimizado**, a organização inteira é focada para o melhoramento contínuo do processo. A organização tem capacidade de identificar as forças e fraquezas do processo ativamente, com o objetivo de prevenir a ocorrência de defeitos.

No nível 5 existem 3 KPA: prevenção de defeito, gerenciamento da mudança tecnológica e gerenciamento da mudança no processo. (PAULK et al.)

3.3.4 O CMM nas empresas brasileiras

Segundo a pesquisa "Qualidade e Produtividade no Setor de Software Brasileiro" (PBQP) - realizada entre empresas que desenvolvem, distribuem ou editoram software - existe um significativo aumento no número de empresas que tem algum conhecimento ou que efetivamente utilizam o modelo CMM no Brasil de 1995 à 1999, conforme ilustrada na figura 3.3.

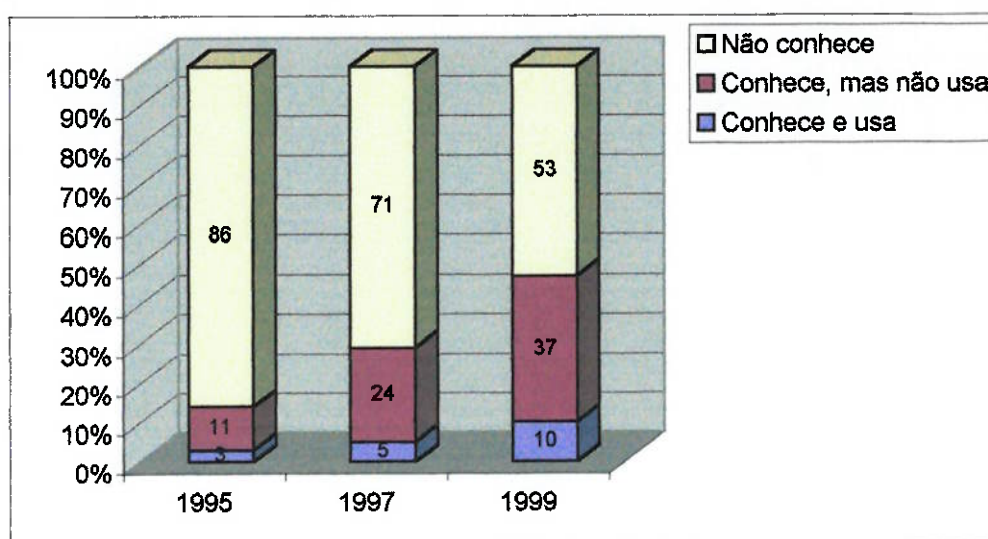


Figura 3.3 - Conhecimento do Modelo CMM no Brasil.

Fonte: (PBQP)

4. GERÊNCIA DE PROJETOS

O objetivo desse capítulo é apresentar o conceito de gerência de projetos de software que será utilizado nesse estudo e a importância dessa gerência para as empresas.

Dentre os modelos e coleção de boas práticas de controle de projetos atuais será apresentado de forma sumária o PMBOK, que servirá de base para toda a conceituação deste capítulo.

4.1 Definição de projeto

Organizações executam trabalhos. Trabalhos geralmente envolvem *Operações* ou *Projetos*, embora os dois possam se sobrepor, Operações e Projetos compartilham muitas características, por exemplo, ambos são:

- Executados por pessoas.
- Restringidos por recursos limitados.
- Planejados, executados e controlados.

As operações e projetos diferem entre si principalmente porque as operações têm um caráter contínuo e repetitivo, ao passo que os projetos têm um caráter *temporário* e *único*. Um projeto pode, portanto, ser definido em termos de suas características distintivas – um projeto é um esforço temporário empreendido para criar um produto ou serviço único. *Temporário* significa que todo projeto tem um início e um término definido. *Único* significa que o produto ou serviço é, de algum modo, diferente de todos os produtos e serviços similares.

Projetos são empreendidos em todos os níveis da organização. Podem envolver uma ou milhares de pessoas. Podem exigir menos de 100 ou mais de 10.000.000 horas para serem concluídos. Os projetos podem envolver uma única unidade da organização ou podem transpor suas fronteiras, como nas parcerias.

Freqüentemente, os projetos representam componentes críticos da estratégia de negócio da empresa executora. Alguns exemplos de projetos incluem:

- Desenvolver um novo produto ou serviço.
- Efetuar mudança na estrutura, no pessoal ou no estilo de uma organização.
- Conceber um novo veículo de transporte.
- Desenvolver ou adquirir um sistema de informações novo ou modificado.
- Construir um edifício ou uma instalação.
- Conduzir uma campanha política.
- Implantar um novo procedimento ou processo de negócio.

4.1.1 Definição de Temporário

Temporário significa que todo projeto tem um início e um término definidos. O projeto é concluído quando seus objetivos são atingidos ou quando se torna claro que estes não serão ou não poderão ser alcançados e o projeto é abortado. Temporário não significa necessariamente ter curta duração, já que muitos projetos duram vários anos. Em qualquer caso, contudo, a duração de um projeto é finita; um projeto não é um esforço contínuo.

Além do mais, temporário geralmente não se aplica ao produto ou serviço criado pelo projeto. Muitos projetos são empreendidos para criar resultados duradouros. Por exemplo, um projeto para construir um monumento nacional produzirá um resultado com expectativa de durar séculos.

Muitos empreendimentos são temporários no sentido de que terminarão em um dado momento. Por exemplo, se a linha de montagem de uma montadora de automóveis for descontinuada e a fábrica for fechada. Projetos são fundamentalmente diferentes, pois um projeto é dado por encerrado quando os seus objetivos propostos forem atingidos, ao passo que esforços não caracterizados como projetos adotam um novo conjunto de objetivos e prosseguem com sua realização.

A natureza temporária dos projetos pode se referir a outros aspectos do empreendimento, tais como:

- A oportunidade de mercado (market window) é geralmente temporária – a maioria dos projetos tem um prazo limitado para produzir seu produto ou serviço.
- A equipe do projeto, como equipe, raramente sobrevive ao seu encerramento – a maioria dos projetos são executados por uma equipe criada com essa única intenção, e que é dissolvida e os seus integrantes são realocados ao seu término.

4.1.2 Definição de produtos ou serviços únicos

Um projeto envolve fazer algo que não foi feito anteriormente e que é, portanto, único.

Um produto ou serviço pode ser único mesmo se a categoria a que pertença for ampla. Por exemplo, milhares de prédios de escritório foram construídos, porém, cada um deles é único – os proprietários são diferentes, as concepções são diferentes, as localizações são diferentes, as construtoras são diferentes, e assim por diante. A presença de elementos repetitivos não altera a singularidade do esforço global. Por exemplo:

- Um projeto para desenvolver um novo avião comercial pode exigir muitos protótipos.
- Um projeto para lançar um novo medicamento no mercado pode exigir milhares de doses de medicamento para realizar testes clínicos.
- Um projeto de um empreendimento imobiliário pode incluir centenas de unidades individuais.

Por ser único o produto de cada projeto, as características que diferenciam o produto ou serviço devem ser progressivamente elaboradas.

Progressivamente significa “prosseguir passo a passo por incrementos constantes” ao passo que elaborado significa “feito com cuidado e detalhadamente, inteiramente desenvolvido”. Essas características diferenciadoras serão definidas ao início do projeto em termos gerais e serão detalhadas e explicitadas à medida que a equipe do projeto desenvolver um entendimento melhor e mais completo do produto.

A elaboração progressiva das características do produto deve ser cuidadosamente coordenada com a definição apropriada do escopo do projeto, especialmente se o projeto for executado sob contrato. Uma vez adequadamente definido, o escopo do projeto – o trabalho a ser realizado – deve permanecer constante, mesmo enquanto as características do produto estiverem sendo progressivamente elaboradas.

4.1.3 Definição de processos de projeto

Os projetos são compostos por processos. Um processo “é uma série de ações que geram um resultado”. Os processos do projeto são executados por pessoas e geralmente recaem em uma das duas principais categorias:

- Processos de gerenciamento de projetos estão relacionados com a descrição e com a organização do trabalho do projeto.
- Os processos voltados ao produto estão relacionados com as especificações e a criação do produto do projeto. Os processos voltados ao produto são tipicamente definidos pelo ciclo de vida do projeto e variam de acordo com o campo de aplicação.

Os processos de gerenciamento de projetos e os processos voltados ao produto sobrepõem-se e interagem por todo o projeto. Por exemplo, o escopo do projeto não pode ser definido sem um entendimento básico de como criar o produto.

4.2 Gerenciamento de projetos

Gerenciamento de Projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender ou superar as necessidades e expectativas que os interessados (stakeholders) possuem no projeto. Atingir ou exceder necessidades e expectativas dos interessados (stakeholders) invariavelmente envolve balancear demandas conflitantes entre:

- Escopo, tempo, custo e qualidade.
- Interessados (stakeholders) com necessidades e expectativas diferentes.
- Requisitos identificados (necessidades) e requisitos não identificados (expectativas).

O termo Gerenciamento de Projetos é às vezes usado para descrever uma abordagem organizacional ao gerenciamento de uma operação. Essa abordagem, mais propriamente denominada, Gerenciamento por Projetos (Management by Projects), trata muitos aspectos da operação rotineira como sendo projetos, a fim de aplicar técnicas de Gerenciamento de Projetos a eles.

Embora um entendimento do Gerenciamento de Projetos seja obviamente crítico para uma organização gerenciada por projetos, uma discussão detalhada dessa abordagem está fora do escopo deste trabalho.

4.3 O PMBOK

PMBOK, Project Management Body of Knowledge (Corpo de Conhecimentos do Gerenciamento de Projetos), é um termo que descreve a soma dos conhecimentos intrínsecos à profissão de Gerenciamento de Projetos. Como em outras profissões, tais como Direito, Medicina e Contabilidade, esse corpo de conhecimentos pertence aos que o praticam e aos acadêmicos que o aplicam e o aprimoram.

O PMBOK completo engloba o conhecimento de práticas comprovadas e tradicionais que são amplamente aplicadas, bem como o conhecimento de práticas inovadoras e avançadas que apresentam um uso mais restrito. (PMBOK)

O PMBOK divide esse corpo de conhecimento em nove Áreas de Conhecimento, cada uma possuindo diversos processos, conforme a descrição abaixo:

1. **Gerenciamento da Integração do Projeto.** Descreve os processos requeridos para assegurar que os vários elementos do projeto sejam adequadamente coordenados.

Consiste no Desenvolvimento do Plano Global do Projeto, Execução do Plano Global do Projeto e o Controle Global de Mudanças.

2. **Gerenciamento do Escopo do Projeto.** Descreve os processos requeridos para assegurar que o projeto inclua todo e tão somente o trabalho necessário para ser concluído com sucesso.

Consiste na Iniciação, Planejamento do Escopo, Definição do Escopo, Verificação do Escopo e Controle de Mudanças do Escopo.

3. **Gerenciamento do Tempo do Projeto.** Descreve os processos requeridos para assegurar a conclusão do projeto no prazo previsto.

Consiste na Definição de Atividades, Seqüenciamento de Atividades, Estimativa de Duração de Atividades, Desenvolvimento da Programação e Controle da Programação.

4. **Gerenciamento de Custos do Projeto.** Descreve os processos requeridos para assegurar que o projeto seja concluído no orçamento aprovado.

Consiste no Planejamento de Recursos, Estimativas de Custos, Orçamento e Controle de Custos.

5. **Gerenciamento da Qualidade do Projeto.** Descreve os processos requeridos para assegurar que o projeto atenda as necessidades pelas quais ele foi empreendido.

Consiste no Planejamento da Qualidade, Garantia da Qualidade e Controle da Qualidade.

6. **Gerenciamento de Recursos Humanos do Projeto.** Descreve os processos requeridos para utilizar de maneira mais eficaz o pessoal envolvido no projeto.

Consiste no Planejamento Organizacional, Recrutamento de Pessoal e Desenvolvimento de Equipe.

7. **Gerenciamento de Comunicações do Projeto.** Descreve os processos requeridos para assegurar adequadamente e a tempo a geração, coleta, disseminação, armazenamento e disposição final das informações do projeto.

Consiste no Planejamento de Comunicações, Distribuição de Informação, Relatório de Desempenho e Encerramento Administrativo.

8. **Gerenciamento de Riscos do Projeto.** Descreve os processos relacionados com a identificação, análise e respostas aos riscos do projeto.

Consiste em Identificação de Riscos, Quantificação de Riscos, Desenvolvimento de Respostas a Riscos e Controle de Respostas a Riscos.

9. **Gerenciamento do Suprimento do Projeto.** Descreve os processos requeridos para aquisição de bens e serviços de terceiros.

Consiste no Planejamento de Suprimento, Planejamento do Processo de Solicitação, Processo de Solicitação, Seleção de Fornecedor, Administração de Contrato e Encerramento de Contrato.

4.3.1 Grupos de Processos

Os processos de gerenciamento de projetos podem ser organizados em cinco grupos, cada um deles com um ou mais processos:

- **Processos de iniciação** – reconhecem que um projeto, ou uma fase de um projeto deve começar e geram o comprometimento para que isso aconteça.
- **Processos de planejamento** – elaboram e mantêm um esquema operacional (workable) para atender às necessidades para as quais a realização do projeto foi proposta.
- **Processos de execução** – coordenam pessoas e outros recursos para realizar o plano.
- **Processos de controle** – asseguram que os objetivos do projeto sejam atingidos pelo monitoramento e medição do progresso, tomando medidas corretivas quando necessário.
- **Processos de encerramento** – formalizam a aceitação do projeto ou da fase do projeto, conduzindo-o a um final ordenado.

Os grupos de processos estão ligados pelos resultados que eles produzem – o resultado ou produto de um torna-se um insumo para um outro. Dentro dos grupos de processos essenciais, as ligações são iterativas – o processo de planejamento logo de início fornece ao de execução um Plano Global do Projeto documentado, que, por sua vez, fornece atualizações documentadas ao plano à medida que o projeto progride. Essas ligações estão ilustradas na Figura 4.1.

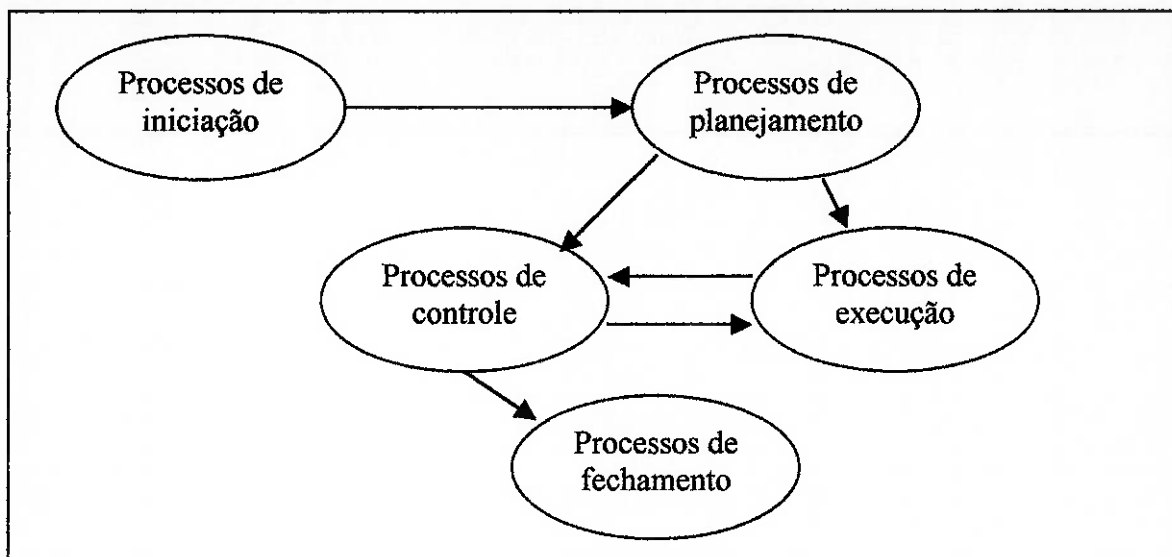


Figura 4.1 – Interligações de processos no PMBOK

Fonte: (FILGUEIRAS, 2002)

Além disso, os grupos de processos de gerenciamento de projetos não são discretos, nem eventos singulares; são atividades que se sobrepõem e que ocorrem em níveis variados de intensidade durante cada uma das fases do projeto.

A Figura 4.2 ilustra como os grupos de processos se sobrepõem e variam em uma fase.

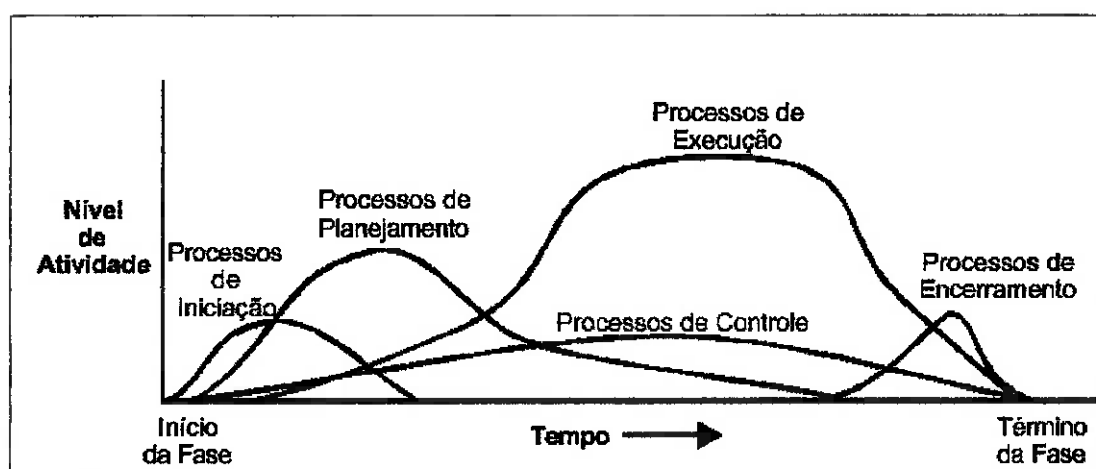


Figura 4.2 – Gráfico de atividades por tempo, nos processos do PMBOK

Fonte: (CAMPOS)

Finalmente, as interações dos grupos de processos também cruzam as fases de tal forma que o encerramento de uma fase fornece os insumos para a iniciação da seguinte. Por exemplo, o encerramento da fase de projeto (design) requer a aceitação, por parte do cliente, dos documentos de projeto (design). Simultaneamente, os documentos de projeto (design) definem a descrição do produto para o desenvolvimento da fase de implementação. Essa interação é ilustrada na Figura 4.3.

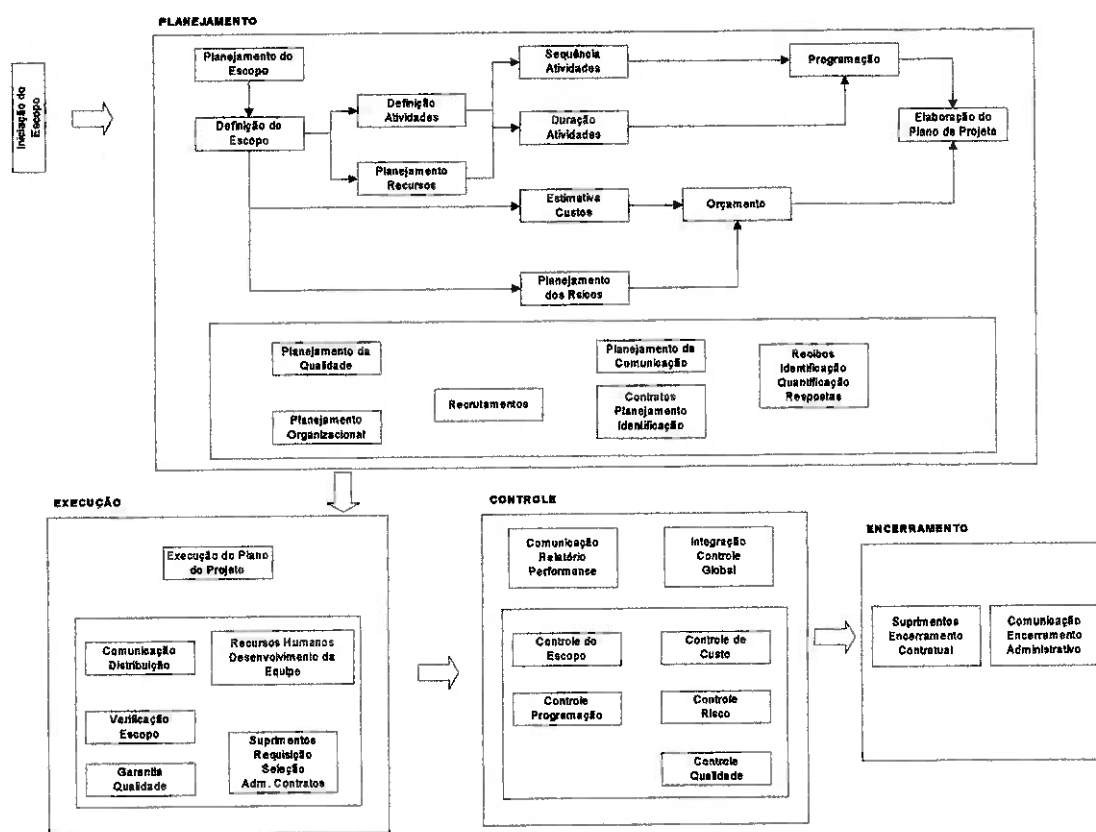


Figura 4.3 – Interação dos grupos de processo do PMBOK

Fonte: (CAMPOS)

A repetição dos processos de iniciação no começo de cada fase ajuda a manter o projeto focado nas necessidades do negócio para o qual ele foi proposto. Isso deve também contribuir para garantir que o projeto seja interrompido, caso não haja mais a necessidade que o gerou, ou quando o projeto provavelmente não satisfará aquela necessidade.

Uma vez que a empresa está decidida a iniciar um PIP, é necessário dar o primeiro passo, que é escolher um modelo ou padrão a seguir. Atualmente existe uma variedade de modelos para controle de processo, padrões de garantia de qualidade, técnicas de gerenciamento de projetos e coleções de boas práticas para o desenvolvimento de software, além de a empresa poder optar por desenvolver o seu próprio PIP desde o início.

Como a maior barreira para adoção de um PIP é o iniciar o programa de melhoria, neste estudo será apresentada uma maneira de seguir o modelo CMM nas pequenas empresas, em particular o nível 2 do CMM. Atingindo o nível 2, a empresa terá maturidade suficiente para escolher entre seguir para o nível 3 do CMM ou desenvolver um modelo próprio.(LARYD, ORCI)

Além do CMM, serão mostradas como algumas técnicas e boas práticas de outros modelos, principalmente o PMBOK, podem ser utilizadas no cotidiano de uma pequena empresa, para a melhoria do seu processo de desenvolvimento e gerência de projetos.

Como foi visto no capítulo 3, o CMM nível 2 possui seis KPA: gerenciamento de requisitos, planejamento de projeto de software, acompanhamento de projeto de software, gerenciamento de subcontrato de software, garantia da qualidade de software e gerenciamento da configuração de software.

Para facilitar o entendimento da necessidade destas KPA e exemplificar a sua utilização, neste trabalho será apresentada uma empresa fictícia, que será contratada para um projeto de desenvolvimento de software. Este projeto será dividido em etapas e para cada etapa serão apresentadas as visões e expectativas do cliente, da empresa que irá realizar o projeto e dos parceiros envolvidos.

5.1 Cenário de uma pequena empresa

Uma empresa fictícia possui dez funcionários e alguns anos de experiência em desenvolvimento. Esta empresa recebe um convite para apresentar um orçamento para a realização de um projeto de desenvolvimento de um sistema de software, com um prazo estimado de seis meses de duração.

Este projeto precisará da participação de todos os seus funcionários, porém nem todos os funcionários serão alocados exclusivamente para este projeto.

Será necessária também a participação de empresas parceiras para a realização dos serviços e após a entrega do projeto será contratada a manutenção do sistema desenvolvido.

5.2 Início do desenvolvimento de um projeto

A primeira etapa deste projeto é a apresentação da proposta de orçamento para o cliente. Esta etapa é a iniciação do projeto.

Na visão do cliente, ele espera nesta primeira etapa receber as seguintes informações:

- A empresa contactada entendeu o que é para ser feito e consegue realizar o serviço;
- Os prazos e cronograma para a realização dos serviços;
- O custo por módulo entregue, caso seja possível modular o projeto;
- O custo total do projeto;

5. GERÊNCIA DE PROJETOS E QUALIDADE DE SOFTWARE PARA PEQUENAS EMPRESAS DE SOFTWARE

Toda empresa precisa ser suficientemente organizada para conseguir se manter e evoluir. Falta de processos, documentação escassa, orçamentos irreais, projetos que não são realizados no prazo ou com a qualidade prometida, são características de empresas inexperientes, que podem sobreviver durante um período curto de existência, mas que precisam dar um salto qualitativo, se quiserem permanecer no mercado.

Essas características negativas têm um peso ainda maior nas pequenas empresas, uma vez que o número reduzido de funcionários leva a uma sobrecarga de funções por funcionário, e a redução do tempo que esse funcionário teria para a realização de todas as tarefas necessárias a empresa.

Uma alternativa viável para essas empresas é a criação de um círculo de parceria.

Neste modelo de negócios, a empresa se cerca de parceiros estratégicos para a terceirização de algumas tarefas como programação, marketing, vendas, traduções, etc., a fim de que ela possa atuar mais focada no seu ramo de atividade principal.

Entretanto, a parceria estratégica exige que a empresa tenha um controle efetivo sobre os projetos realizados por seus parceiros. Controle de contratos, controle de prazos de execução e gerenciamento de risco são fundamentais para o sucesso desse modelo.

Quando a empresa percebe que o seu processo de desenvolvimento de software é inadequado para o seu crescimento e eventualmente até para a sua sobrevivência, ela possui a experiência necessária para iniciar um programa de melhoria do processo (na abreviação em inglês **Process Improvement Program**, PIP), uma vez que ela já conhece quais são as suas dificuldades e quais melhorias deve realizar. E principalmente está disposta a mudar, dado que a situação atual é insustentável.

Para a empresa conseguir suprir estas informações, serão necessários os seguintes dados:

- Descrição do produto a ser desenvolvido;
- Decompor o projeto;
- Modelar o sistema;
- Levantamento dos requisitos deste produto;
- Levantamento inicial dos Stakeholders;
- Plano estratégico;
- Critério para a seleção de projetos;
- Informações históricas de projetos;

A descrição do produto a ser desenvolvido é o item básico para o início de um novo projeto de desenvolvimento, neste item o cliente informa à empresa o que é para ser realizado. Com base nesta descrição a empresa tem que coletar quais são os requisitos para este produto, como prazo de entrega, tempo de execução, plataforma, sistema operacional, etc.

A gerência de requisitos é um dos itens de KPA do nível 2 do CMM, seu objetivo é estabelecer uma compreensão comum entre o cliente e a empresa, a cerca dos requisitos do projeto a ser desenvolvido. Este acordo com o cliente é a base para o planejamento e gerenciamento do projeto.(PAULK et al.)

É importante ter registrado uma tabela de requisitos com informações como quem solicitou aquele requisito, em qual data, motivos e impactos em outros requisitos, esta tabela é a base para a negociação inicial da proposta de orçamento e poderá ser usada posteriormente no projeto para a negociação de novas funcionalidades ou mudanças de escopo, sem esses registros a empresa perde poder de negociação e poder de gerenciamento do projeto.(MATOS)

O levantamento inicial dos Stakeholders serve para ter um conhecimento sobre quais as pessoas envolvidas na aprovação do projeto e como deverá ser realizada a

comunicação com estas pessoas, ou seja, conhecendo o perfil dos Stakeholders é possível prever se o diálogo com eles deverá ser mais técnico ou mais didático, mais formal ou informal.

Em especial para pequenas empresas o formalismo exigido pode acarretar em custos extras como a subcontratação de um escritório de advocacia para a realização de contratos, uma vez que raramente uma pequena empresa com dez funcionários, como a prevista nesta simulação, possui um departamento jurídico próprio.

O plano estratégico e os critérios para seleção de projetos são visões da empresa sobre quais os objetivos a serem alcançados pela empresa e os tipos de projeto que podem ser aceitos para a realização e o não comprometimento desses objetivos. O projeto proposto deve ser analisado em termos de retorno financeiro, parcela de mercado, percepção pública, entre outros, para ver se está enquadrado nos objetivos da empresa.(CAMPOS)

As informações históricas sobre ambos os resultados das decisões referentes à seleção e ao desempenho de projetos anteriores servem de apoio para a tomada de decisão sobre a execução do projeto proposto.

Com as informação coletadas sobre o que deve ser feito e com a confirmação de que o projeto é interessante para ser desenvolvido, a empresa contratada deve decompor o projeto em componentes menores e mais fáceis de serem gerenciados, a fim de aprimorar a precisão de estimativas de custos, tempo e recursos e definir uma base de referência (baseline) que será usada para medir e controlar o desempenho, facilitando a clara designação de responsabilidades.

Existem técnicas e ferramentas para auxiliar a compreensão, decomposição e funcionamento do projeto. Na engenharia de software esta área é conhecida como modelagem do sistema (system modeling), onde as técnicas e ferramentas mais comuns atualmente são Diagrama de Fluxo de Dados (DFD), Diagrama de Estrutura

de Módulos (DEM), Diagramas de Entidade Relacionamento (DER) e a Unified Modeling Language (UML). (FILGUEIRAS, 2002)

Além de auxiliar a compreensão do sistema, os documentos produzidos durante esta fase servirão como documentação do projeto e auxiliarão tanto na comunicação com o cliente quanto nas etapas de execução e manutenção do sistema.

É possível supor que a empresa fictícia possua dados relativos a informações históricas, uma vez que ela tem alguns anos de experiência em desenvolvimento.

Porém, muitas vezes estes dados não estão devidamente documentados, como por exemplo, o número de horas planejado e efetivo para realizar um serviço, as falhas e acertos do projeto, as alterações de escopo durante o projeto, o número de reuniões necessárias, etc. Essa falta de dados para a tomada de decisão é um dos itens que podem levar ao fracasso de um projeto, pois o orçamento previsto poderá estar longe da realidade, contendo inclusive falhas de projetos anteriores que não foram melhoradas porque não estavam documentadas.

Entretanto, mesmo com dados históricos, a tarefa de estimar prazos e custos de um projeto é uma das mais complicadas de ser realizada, uma vez que um projeto, por possuir características de unicidade, pode não ter semelhança com os projetos passados.

Existem diversas técnicas para ajudar a tarefa de estimar prazos e custos, entre as mais conhecidas estão a Estrutura Analítica de Projeto (EAP / Work Breakdown Structure) para a decomposição do projeto, Precedence Diagramming Method (PDM) e Arrow Diagramming Method (ADM) para sequenciamento de atividades, julgamento de especialistas no assunto (ex. Método Delphi), além de técnicas voltadas especificamente para desenvolvimento de software, como estimativas de número de linhas de código (LOC), o modelo COCOMO ou pontos de função (PF). (FILGUEIRAS, 2002), (PRESSMAN)

Com base nas informações levantadas é traçado um plano de projeto inicial contendo:

- O gerente do projeto;
- Os serviços a serem realizados;
- O perfil das pessoas que irão realizar os serviços;
- O perfil do cliente e o modelo inicial de comunicação com o mesmo;
- A estimativa de tempo de desenvolvimento do projeto;
- A estimativa de custo de desenvolvimento;
- A necessidade ou não de subcontratação;
- A elaboração de um plano de entrega do projeto com divisão por fases;

Nesta simulação, está sendo considerada a necessidade de subcontratação de serviços.

Para esta subcontratação, é feito um levantamento inicial do que deve ser subcontratado, qual o prazo de entrega, quais os requisitos funcionais e restrições dos produtos a serem entregues, ou seja, é iniciado um processo similar a apresentação da proposta da empresa fictícia junto ao cliente, porém os dados relativos a esta subcontratação tem influencia direta no planejamento da proposta e no planejamento da execução do projeto.

Com a definição do plano de projeto inicial é realizada uma proposta para o cliente e inicia-se uma etapa de negociação da proposta.

De acordo com (FILGUEIRAS, 2002) um modelo de proposta para o cliente deveria conter:

- Carta de apresentação;
 - Do gerente do projeto ao tomador da decisão do cliente;
 - Preço e data de entrega ou custo-hora;
 - Pressão para resposta rápida;

- Página de abertura;
- Índice comentado;
- Objetivo;
- Vantagens
 - Por que o processo da empresa é melhor;
 - Vantagem da solução/desvantagens dos concorrentes;
- Custo
 - Preço e data de entrega;
 - Preço separado do hardware e sistema operacional;
 - Custo final do desenvolvimento ou custo-hora;
 - Payback; benefícios não-financeiros;
- Planejamento
 - Metodologia de desenvolvimento; fases; milestones;
 - Equipe de desenvolvimento;
 - Responsabilidades do usuário;
- Produtos
 - Hardware, sistema operacional, pacotes de software;
 - Software desenvolvido;
 - Garantias;
 - Documentos;
 - Treinamento;
 - Como entregar, onde entregar, quando entregar;
- Aceitação
 - Advertência ao usuário que deverá assinar o plano de aceitação;
- Alternativa
 - Comparação com o concorrente;
- Termos, condições e restrições;
- Terminologia;

O modelo proposto é bem abrangente e detalhado, sendo muito útil para propostas mais formais. Proposta menos formais, como por exemplo realizar uma proposta para a terceirização de um serviço para um parceiro, podem excluir alguns itens como a

carta de apresentação por exemplo, uma vez que o parceiro já conhece a empresa e provavelmente só usará a proposta apresentada como base para uma proposta própria junto ao cliente final.

A etapa de negociação pode ser aproveitada para a captação de dados estatísticos importantes, como o número de reuniões necessárias para o término da negociação, diferenças do valor de custo-hora orçadas e aceitos, diferenças no tempo de desenvolvimento orçado e aceito, entre outros.

Particularmente para pequenas empresas, uma fase de negociações longa pode ser prejudicial, pois, como é alta a probabilidade de existir um acúmulo de funções devido ao número reduzido de funcionários, o tempo gasto para a realização do plano de projeto, propostas, reuniões, etc., pode ter um impacto sobre os projetos que estão sendo executados. Por isso é importante que esse tempo seja medido para que ele possa ser previsto em futuros projetos e minimizar os conflitos entre os projetos.

Após a aceitação da proposta e o término da etapa de negociação, tem início a etapa de desenvolvimento do projeto, já com o escopo inicial do projeto definido e aprovado por ambas as partes, muitas vezes firmando um contrato formal de desenvolvimento.

5.3 Execução e controle do desenvolvimento de um projeto

No desenvolvimento do projeto as expectativas do cliente são basicamente:

- Verificar o andamento do projeto;
- Aprimorar os conceitos e requisitos do escopo do projeto;
- Receber os módulos ou produtos parciais, quando for possível modular o projeto;
- Receber treinamento inicial ou explicações de funcionamento dos produtos entregues;

É fundamental para o cliente conseguir acompanhar o andamento do projeto, para que ele possa se planejar caso haja algum atraso, imprevisto ou mudanças no escopo do projeto, que podem inclusive inviabilizar a continuidade e andamento do mesmo.

A empresa contratada deverá definir a forma como irá fornecer estas informações de acompanhamento ao cliente, que pode ser formalmente através de reuniões ou relatórios periódicos previstos em contrato, ou informalmente, através de telefonemas, e-mails, etc.

As comunicações formais e periódicas são mais aconselháveis por registrarem em ata de reunião ou relatório o que foi comunicado, esta informação poderá ser útil em negociações de mudanças de escopo como veremos a seguir.

Um outro ponto importante para o cliente e para a empresa contratada é a revisão dos conceitos e do escopo do projeto a ser realizado. Muitas vezes no início de um projeto existem requisitos que são incógnitas para ambas as partes e que no desenrolar do desenvolvimento vão sendo incluídos no projeto, além de requisitos mudarem por motivos tecnológicos, legislativos, mudanças de cenário político ou econômico, etc. Estas mudanças devem ser analisadas e revistas na etapa de desenvolvimento do projeto, uma vez que podem ter um impacto significativo no orçamento, no prazo ou no risco do desenvolvimento do projeto.

O formalismo na condução do projeto ajuda também a resolver estes problemas de mudança de escopo, uma vez que ele obriga a empresa a registrar um histórico de dados relativos ao projeto. Esse histórico deve possuir a tabela de requisitos do sistema constantemente atualizada, a definição inicial e os registros de mudança no escopo, o contrato de desenvolvimento, atas de reunião, etc. Com estes dados, a empresa consegue conduzir mais facilmente uma negociação com o cliente, apoiada em termos e compromissos assinados por ambas as partes.

Por fim, o cliente tem a expectativa de receber produtos ou módulos do projeto, tanto para conseguir utilizar algumas funcionalidades do projeto como poder cadastrar

informações, treinar pessoal, realizar testes, quanto para garantir a sua gerência ou administração que o projeto está tendo um andamento aceitável. Esta entrega de produtos geralmente é acompanhada com algum tipo de treinamento ou manual de usuário do produto entregue.

Para a empresa contratada, esta entrega de produtos parciais tem que ser analisada por diversos ângulos, por um lado ela representa uma forma de receber pagamentos por desenvolvimentos parciais, é uma maneira também de certificar para a empresa que o produto final está sendo desenvolvido corretamente, além de satisfazer a vontade do cliente. Por outro lado, o cliente pode querer realizar muitas mudanças nestes subprodutos entregues, o que pode gerar atrasos no cronograma geral do projeto e isso para uma empresa com poucos funcionários tem um impacto muito grande, já que a gerência de projetos tem que analisar constantemente a disponibilidade de pessoal para a realização de todos os projetos.

Para a empresa contratada esta é etapa mais crítica do projeto, pois envolve o maior número de atividades e tarefas, nesta etapa a empresa deve:

- Montar a equipe de desenvolvimento interno;
- Medir e controlar o desempenho da equipe;
- Controlar o desenvolvimento dos serviços terceirizados;
- Controlar a sobrecarga de funções e paralelismo com outros projetos;
- Desenvolver e testar o sistema;
- Controlar a comunicação interna, com o cliente e com os parceiros;
- Controlar mudanças de requisitos e de escopo;
- Controlar prazos, custos e riscos;
- Garantir a qualidade do trabalho;
- Controlar a configuração dos produtos;
- Controlar a entrega dos subprodutos;

Com o escopo inicial definido e aprovado do projeto, a empresa tem condições de definir os papéis e funções da equipe de desenvolvimento interno, bem como planejar a interação com os seus parceiros nas terceirizações de serviços.

Este grupo formado com funcionários internos e parceiros tem que caminhar em paralelo e em sincronia, caso contrário, o projeto poderá sair de controle, provocando atrasos e impactos negativos em outros projetos, lembrando que todos os funcionários internos participarão em algum momento deste projeto, portanto quanto mais tempo eles passarem desenvolvendo este projeto, menos tempo eles terão para realizar os outros projetos da empresa.

Cabe ao gerente do projeto verificar este andamento e dar o suporte necessário para o cumprimento das tarefas do projeto, além de alinhar este desenvolvimento com os outros projetos que estão sendo desenvolvidos pela empresa. Consequentemente, o gerente precisa estar constantemente informado, tanto do desenvolvimento deste projeto quanto dos outros projetos da empresa.

O gerente então deve organizar toda a comunicação do projeto, tanto internamente quanto junto aos parceiros e ao cliente, esta comunicação inclui entre outras atividades:

- Realizar reuniões periódicas para verificações;
- Cobrar resultados ou previsões de término, da equipe e dos parceiros;
- Comunicar ao cliente a evolução do projeto;
- Realizar com o cliente reuniões para validação de produtos entregues;
- Realizar com o cliente reuniões para verificação de mudanças de requisitos;
- Negociar com o cliente alterações de prazos e custos;
- Negociar com os parceiros alterações de prazos e custos;

Além destas funções, muitas vezes em uma pequena empresa, o gerente, devido a sobrecarga de funções, pode exercer além das funções organizacionais, funções

operacionais como desenvolvimento, suporte, atendimento, etc. Com isso, quanto maior o número de projetos paralelos, maior a carga de trabalho sobre o gerente.

Voltando ao capítulo 2, este é o cenário dos “heróis”, citado por (PAULK,1998), onde as regras de desenvolvimento não são sempre aplicadas, o ciclo de desenvolvimento é curto, e a taxa de estresse é alta.

Entretanto, para uma empresa evoluir e crescer ela não pode depender tanto dos “heróis”, uma vez que a concentração de poder inviabiliza a substituição ou folga destas pessoas, além do que, trabalhar constantemente com taxas de estresse alta pode diminuir a qualidade do produto entregue e reduzir a produtividade do funcionário.

As práticas de gerência e qualidade, ajudarão a descentralizar o poder difundindo o conhecimento pela empresa, uma vez que elas documentam as tarefas e processos realizados, com isso em caso de substituição de pessoal ou manutenção do produto, o histórico de alterações, procedimentos, negociações estará disponível, facilitando a tomada de decisão.

Em uma pequena empresa, o uso destas práticas não reduzirá o número de atividades de um gerente, porém, ele terá informações mais precisas para trabalhar, melhorando a produtividade de seu trabalho.

Além do papel do gerente, um outro papel fundamental no desenvolvimento de um projeto de sistema é o do desenvolvedor.

O desenvolvedor, ou a equipe de desenvolvimento, deverá ao longo do projeto executar o gerenciamento da configuração de software, ou seja, documentar o sistema, atualizando os diagramas efetuados no início do planejamento do escopo, documentando a versão dos softwares, compiladores, componentes, etc. utilizada no desenvolvimento do sistema, além de testar o sistema, especificando quais testes foram efetuados e quais os resultados obtidos.

Em uma pequena empresa, é muito provável que, devido a falta de pessoal específico, estes desenvolvedores também escrevam os manuais do sistema e realizem treinamentos de uso do sistema junto ao cliente.

Por fim, cabe aos desenvolvedores suprir o gerente do projeto com informações relativas aos seus desempenhos, suas atividades, relatos de finalização de tarefas, ocorrências, etc.

5.4 Finalização e entrega de um projeto

Como visto no capítulo 4, o projeto é concluído quando seus objetivos são atingidos ou quando se torna claro que estes não serão ou não poderão ser alcançados e o projeto é abortado.

Porém, a finalização bem sucedida de um projeto não significa apenas entregar o produto final ao cliente e receber o pagamento, existem alguns processos importantes nesta etapa de encerramento, que serão úteis para novos projetos.

É preciso formalizar com o cliente a entrega e aceitação do produto, para tanto o primeiro passo é realizar o processo de implantação do sistema no seu ambiente de utilização real e verificar as últimas consistências.

Após o período final de testes de implantação, o produto está válido e deverá passar por uma aceitação formal, como um termo de aceitação assinado pelo cliente. Com esta aceitação formalizada, o projeto está formalmente encerrado junto ao cliente.

Para a empresa no entanto, é interessante fazer um levantamento dos conhecimentos obtidos com o projeto, relacionando os erros, acertos, novas tecnologias utilizadas, estatísticas de desempenho, fatos importantes, comparativos entre os prazos e custos estimados com os reais, contendo comentários e explicações para as eventuais

diferenças, enfim, documentar o conhecimento proveniente deste trabalho para uso futuro.

5.5 Manutenção do sistema

Muitas vezes os clientes pedem uma manutenção no sistema desenvolvido, após a conclusão de um projeto. Este tópico descreverá os problemas enfrentados pelas pequenas empresas quando elas não seguem os processo de controle de qualidade descritos nos tópico anteriores.

É relativamente simples gerenciar a manutenção sistemática e periódica de um sistema, basicamente aloca-se um número determinado de funcionários que irão executar tarefas relativas a manutenção, periodicamente. Para a gerência é fácil estimar o tempo utilizado destes funcionários e, como são tarefas de rotina, o ambiente de desenvolvimento para este sistema provavelmente estará sempre disponível e atualizado.

Os problemas surgem quando esta manutenção é feita de forma esporádica, com urgência de prazos.

O primeiro obstáculo é preparar a equipe que irá realizar a manutenção, cabe a gerência saber quem estará disponível para realizar a tarefa e quando estará disponível.

O próximo obstáculo é preparar o ambiente de desenvolvimento. Realizar a manutenção em um sistema depois de alguns meses, ou anos, pode ser inviável, caso a empresa não possua uma documentação do sistema. Informações como quais as versões utilizadas do sistema operacional, do banco de dados, do compilador, dos componentes de terceiros, entre outros, são essenciais para a criação do ambiente de desenvolvimento.

Além do ambiente, a compreensão do funcionamento do sistema também é essencial. Quais são os requisitos funcionais do sistema? Qual o impacto da alteração de um cálculo? Qual a utilidade de um campo específico em uma tabela de banco de dados? Qual o resultado esperado da execução de um teste e, caso já tenha sido realizada a manutenção, qual era o resultado anterior obtido? Estes são alguns exemplos de perguntas que inviabilizam a manutenção em um sistema, caso não consigam ser respondidas.

Em um mundo de ciclo de desenvolvimentos curtos e acúmulos de projetos, as pequenas empresas desperdiçarão um tempo crucial para entender e recriar o ambiente de desenvolvimento dos sistemas que elas mesmas produziram, caso a documentação necessária não tenha sido produzida, enquanto o projeto estava sendo executado.

6. CONCLUSÃO

6.1 Contribuições do trabalho

Neste trabalho foram apresentadas as características das pequenas empresas de software, gerenciamento de projetos e qualidade de software.

Em relação as pequenas empresas, foi apresentada uma visão geral do seu funcionamento com um enfoque especial sobre a atuação das pequenas empresas no Brasil e sobre o modelo de círculo de parcerias.

Sobre qualidade de software, foram apresentados as principais áreas de atuação com os seus conceitos básicos, além da descrição do modelo CMM.

Em relação a gerência de projetos, foram apresentadas as boas práticas do PMBOK com os seus conceitos e características.

A união destas três áreas no exemplo proposto, onde uma pequena empresa realiza um projeto de desenvolvimento de sistema de software, com os requisitos do CMM e conceitos do PMBOK, é provavelmente a maior contribuição deste trabalho.

Com este exemplo foi possível apresentar tanto a união dos conceitos das três áreas anteriores, como também alguns exemplos práticos de ferramentas e técnicas utilizadas, bem como da importância para as pequenas empresa de software de seguirem as boas práticas de gerenciamento de projetos e qualidade de software.

Com os tópicos apresentados, foi possível apresentar a novos empreendedores, gerentes software ou de projetos de pequenas empresas, conceitos para introduzir ou aprimorar a qualidade e eficiência de suas empresas, bem como ajudar a levantar no meio acadêmico uma discussão maior sobre as necessidades e características das pequenas empresas de software.

6.2 Trabalhos futuros

Uma das maiores dificuldades encontradas para a realização deste trabalho foi encontrar estudos específicos sobre o modo de trabalho das pequenas empresas de software, principalmente trabalhos acadêmicos que levassem em consideração a realidade brasileira. Um estudo mais aprofundado e específico sobre este tema seria muito interessante.

Outro ponto a ser explorado é apresentar outros métodos, práticas ou normas sobre qualidade de software e gerenciamento de projeto, uma vez que este projeto está baseado principalmente no CMM e PMBOK.

7. REFERÊNCIAS BIBLIOGRÁFICAS

ALVIM, P.C.R. de C. **O papel da informação no processo de capacitação tecnológica das micro e pequenas empresas.** Brasília: Revista Ciência da Informação, v.27, n.1, p.28-35, jan./abr. 1998.

BELLOQUIN, Á. **Qualidade de Software: Um Compromisso da Empresa Inteira.** São Paulo: Developers Magazine, jun. 1997.

BREMER, C. **Uma Análise da Aplicação do Modelo de Empresas Virtuais na Indústria Brasileira.** Piracicaba: XVI Encontro Nacional de Engenharia de Produção - Brazil Second Congress of Industrial Engineering, Universidade Metodista de Piracicaba, Piracicaba. 1996.

BREMER, C. F.; MUNDIM, A. P. F. **Proposta de um ambiente cooperativo suportado por computador para participação de pequenas e médias empresas em organizações virtuais.** São Paulo: XXI Simpósio de Gestão da Inovação Tecnológica. 2000.

BREMER, C.; WALZ, M.; MOLINA, A.; EVERSHEIM, W. **Global Virtual Business - A Systematic Approach For Exploiting Business Opportunities in Dynamic Markets.** International Journal of Agile Manufacturing. Vol. 2, Issue 1. 1999.

CAMPOS, J. C. F. Apostilas relativas a disciplina **Gestão de Escopo.** São Paulo: Escola de Administração de Empresas de São Paulo, Fundação Getulio Vargas, GVPEC – Programa de Educação Continuada. 2002.

FILGUEIRAS, L. – Apostilas relativas a disciplina **Gerência de Projetos de Desenvolvimento e Aquisição de Software.** São Paulo: Escola Politécnica da Universidade de São Paulo, PECE – Programa de Educação Continuada em Engenharia, curso MBA – Engenharia de Software. 2001/2002.

GOMES, N. S. **Qualidade de Software** - Uma necessidade. Disponível em: http://www.esaf.fazenda.gov.br/cst/arquivos/Qualidade_de_Soft.pdf. Acessado em 17 dez. 2002.

HAMUEL, G.; PRAHALAD, C. K. **Competing for the Future**. E.U.A. Boston. 1994.

HIRAMA, K. – Apostilas relativas a disciplina **Qualidade de Software**. São Paulo: Escola Politécnica da Universidade de São Paulo, PECE – Programa de Educação Continuada em Engenharia, curso MBA – Engenharia de Software. 2001/2002

LAITINEN, M et al. **Software Engineering in the Small**. IEEE Software, p. 75-77, set./out. 2000.

LARYD, A; ORCI, T. **Dynamic CMM for Small Organizations**. Suécia: Umea University, Department of Computer Science, 2000.

LEI FEDERAL Nº 9841 **Estatuto da Microempresa e da Empresa de Pequeno Porte**. DOU/ 6.10.99, Out/1999.

MATOS, J. P. – Apostilas relativas a disciplina **Engenharia de Requisitos**. São Paulo: Escola Politécnica da Universidade de São Paulo, PECE – Programa de Educação Continuada em Engenharia, curso MBA – Engenharia de Software. 2001/2002

OTTOBONI, C; PAMPLONA, E. de O. **Proposta de pesquisa para avaliar a necessidade de se medir o desempenho financeiro das micro e pequenas empresas**. Disponível em: www.iem.efei.br/edson/download/Artceliaenegep01.doc. Acesso em: 17 dez. 2002.

PAULK, M. C. **Using the Software CMM in Small Organizations**. The Joint 1998 Proceedings of the Pacific Northwest Software Quality Conference and the Eighth International Conference on Software Quality. 1998.

PAULK, M.C.; CURTIS, B.; CHRISSIS, M.B.; WEBER, C.V. **The Capability Maturity Model for Software, Version 1.1.** Software Engineering Institute, CMU/SEI-93-TR-24, DTIC Number ADA263403, fev. 1993.

PBQP. Programa Brasileiro da Qualidade e Produtividade. **Qualidade e Produtividade no Setor de Software**, Resultados da Pesquisa 2001.

PMBOK **Project Management Body Of Knowledge.** Project Management Institute - PMI 1996, Versão traduzida pelo PMI-SP.1996.

PRESSMAN, R. S. **Software Engineering – A Practioner’s Approach.** McGraw-Hill, 5 edição. 2001.

SEBK **Software Engineering Body of Knowledge.** v.1.0. Estados Unidos: Carnegie Mellon Software Engineering Institute. 1999. Disponível em:
<http://www.sei.cmu.edu/publications/documents/99.reports/99tr004/99tr004sm3-3.html>. Acessado em 02 dez. 2002.

SEBRAE **Avaliação de 1999 e perspectivas para o ano 2000.** Relatório de Sondagem. Mar./2000.

SEBRAE **Fatores Condicionantes e Taxa de Mortalidade de Empresas.** Relatório de Pesquisa. Out./1999.

SEBRAE-SP **Sobrevivência e mortalidade das empresas paulistas de 1 a 5 anos.** Relatório Final. Out./2001.

SEBRAE **Definição de Micro e Pequenas Empresas (MPEs).** Pesquisas. Out/2001. Disponível em:
http://www.sebraesp.com.br/novo/pesquisa/download/informacoes_basicas.doc
Acesso em 17 dez. 2002.