

Ana Karla Chaves Melo

A GESTÃO DA INFORMAÇÃO NO CONTEXTO DO
PROCESSO DE SOFTWARE: UM ESTUDO DE CASO DO
SERPRO - Serviço Federal de Processamento de Dados

Belo Horizonte
Escola de Ciência da Informação
Universidade Federal de Minas Gerais
2004

Ana Karla Chaves Melo

**A GESTÃO DA INFORMAÇÃO NO CONTEXTO DO
PROCESSO DE SOFTWARE: *UM ESTUDO DE CASO DO
SERPRO - Serviço Federal de Processamento de Dados***

Dissertação apresentada ao curso de Pós-Graduação em
Ciência da Informação da Universidade Federal de Minas
Gerais, como requisito parcial à obtenção do título de
Mestre em Ciência da Informação.
Área de Concentração em Informação Gerencial e Tecnológica
Orientador(a): Prof. Dr.º Mauro de Oliveira
Universidade Federal de Minas Gerais

Belo Horizonte
Escola de Ciência da Informação
Universidade Federal de Minas Gerais
2004

Ana Karla Chaves Melo

02:004.4
M 5289
2004
T

OK/13

U.F.M.G. - BIBLIOTECA UNIVERSITÁRIA

OK/06



187880401

NÃO DANIFIQUE ESTA ETIQUETA

OK/20

**A GESTÃO DA INFORMAÇÃO NO CONTEXTO DO
PROCESSO DE SOFTWARE: *UM ESTUDO DE CASO DO
SERPRO - Serviço Federal de Processamento de Dados***

Dissertação apresentada ao curso de Pós-Graduação em
Ciência da Informação da Universidade Federal de Minas
Gerais, como requisito parcial à obtenção do título de
Mestre em Ciência da Informação.

Área de concentração: Informação Gerencial e Tecnológica

Orientadora: Prof^a. Dr^a. Marlene de Oliveira

Universidade Federal de Minas Gerais

Belo Horizonte

Escola de Ciência da Informação

Universidade Federal de Minas Gerais

2004

BIBLIOTECA "PROF^a ETELVINA LIMA"
Escola de Ciência da Informação da UFMG

Ana Karla Chaves Melo

02:004.4
M 528q
2004
T

OK/13

U.F.M.G. - BIBLIOTECA UNIVERSITÁRIA OK/66



187880401

NÃO DANIFIQUE ESTA ETIQUETA

OK/20

**A GESTÃO DA INFORMAÇÃO NO CONTEXTO DO
PROCESSO DE SOFTWARE: *UM ESTUDO DE CASO DO
SERPRO - Serviço Federal de Processamento de Dados***

Dissertação apresentada ao curso de Pós-Graduação em
Ciência da Informação da Universidade Federal de Minas
Gerais, como requisito parcial à obtenção do título de
Mestre em Ciência da Informação.

Área de concentração: Informação Gerencial e Tecnológica

Orientadora: Prof^a. Dr^a. Marlene de Oliveira

Universidade Federal de Minas Gerais

Belo Horizonte

Escola de Ciência da Informação

Universidade Federal de Minas Gerais

2004

BIBLIOTECA "PROF.^a ETELVINA LIMA"
Escola de Ciência da Informação da UFMG



36312

Melo, Ana Karla Chaves

M528g A gestão da informação no contexto do processo de software [manuscrito] : um estudo de caso do SERPRO – Serviço Federal de Processamento / Ana Karla Chaves Melo. – 2004.
153 f. : il., tab.

Orientadora: Marlene de Oliveira
Dissertação (mestrado) – Universidade Federal de Minas Gerais, Escola de Ciência da Informação.
Referência bibliográfica: f. 136 –139.
Anexos: f. 140-153.

1. Ciência da Informação – Teses 2. Engenharia de Software – Teses 3. Gestão da informação – Teses 4. Gerenciamento da informação – Teses 5. Gestão do conhecimento - Teses I. Título II. Oliveira, Marlene de III. Universidade Federal de Minas Gerais. Escola de Ciência da Informação.

CDU: 02:004.4

CDD: 020

Ficha Catalográfica: Biblioteca Etelvina Lima. Escola de Ciência da Informação da UFMG



UFMG

Universidade Federal de Minas Gerais
Escola de Ciência da Informação
Programa de Pós-Graduação em Ciência da Informação

FOLHA DE APROVAÇÃO

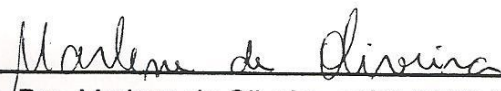
"A GESTÃO DA INFORMAÇÃO NO CONTEXTO DO PROCESSO DE SOFTWARE: UM ESTUDO DE CASO DO SERPRO – SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS".

Ana Karla Chaves Melo

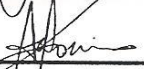
Dissertação submetida à Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação da Universidade Federal de Minas Gerais, como parte dos requisitos à obtenção do título de "Mestre em Ciência da Informação", linha de pesquisa "Organização e Uso da Informação (OUI)".

Dissertação aprovada em 14 de julho de 2004

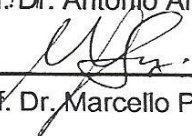
Por:



Profa. Dra. Marlene de Oliveira –ECI/UFMG (Orientadora)

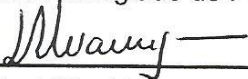


Prof. Dr. Antonio Alfredo Ferreira Loureiro –DCC/UFMG



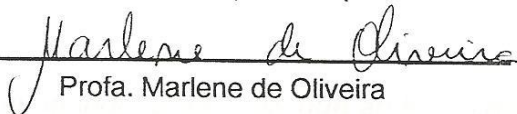
Prof. Dr. Marcello Peixoto Bax - ECI/UFMG

Aprovada pelo Colegiado do PPGCI



Profa. Lídia Alvarenga
Coordenadora

Versão final Aprovada por



Profa. Marlene de Oliveira
Orientadora

AGRADECIMENTOS

A Deus, pela oportunidade de desfrutar da vida.

Aos meus pais, pela compreensão e compreensão da situação que recebi.

A minha irmã, Cária, pelo apoio contínuo e incentivo.

A professora Maria, pela orientação e orientação deste trabalho.

Ao SERPRO, pela oportunidade de desenvolvimento do Mestrado.

Aos gerentes e colaboradores do SERPRO/SUNAT de Belém, pela compreensão e desenvolvimento do trabalho, pela colaboração.

Aos colegas de trabalho, que direta e indiretamente participaram para o desenvolvimento deste trabalho.

Dedico este trabalho a meu marido, José Luiz, que ao longo desta jornada muitas vezes solitária, foi companheiro, compreensivo e um grande incentivador.

AGRADECIMENTOS

A Deus, pela oportunidade de desfrutar de saúde e paz.

Aos meus pais, Chaves e Fransquinha, pelo amor incondicional e pela formação que recebi;

À minha irmã, Ládía Mara, pelos conselhos e incentivos;

À professora Marlene de Oliveira, pela orientação deste trabalho;

Ao SERPRO, pela oportunidade e apoio no desenvolvimento do Mestrado;

Aos gerentes e chefes de projeto dos pólos de desenvolvimento do SERPRO/SUNAT de Belém, Belo Horizonte, Curitiba, Fortaleza, Recife e Salvador, pela colaboração;

Aos colegas do SERPRO Belo Horizonte que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

RESUMO

O desenvolvimento de software consiste em um processo de atividades complexas e carregadas de incertezas, em sua maioria baseia-se no conhecimento tácito dos engenheiros de software. Além disso, o *know-how* e *know-why* adquiridos em cada projeto de desenvolvimento de software, muitas vezes, fica retido apenas em modelos mentais individuais definidos por cada profissional. Sendo assim, é importante que as informações geradas e coletadas nas atividades técnicas e gerenciais de engenharia de software sejam registradas e utilizadas com a finalidade de criar conhecimento sobre o produto e o projeto de software e, também, de dar suporte à aprendizagem e ao aperfeiçoamento do processo de software.

Neste contexto, este trabalho apresenta as práticas e dificuldades no gerenciamento da informação durante o processo de software fundamentado no nível 2 do CMM – Capability Maturity Model.

Para a realização deste trabalho foi utilizada a estratégia de estudo de caso com caráter exploratório e descritivo. Inicialmente, foi realizada a análise documental para mapear a relação entre a gestão da informação e o processo de software. Posteriormente, aplicou-se o questionário visando identificar as práticas e dificuldades na aplicação das diretrizes do CMM nível 2 que fomentam a gestão da informação e a criação de conhecimento.

Como resultados da pesquisa, verificou-se que os gerentes de software praticam a gestão da informação quando executam as diretrizes descritas nas áreas-chaves de processo preconizadas no nível 2 do CMM. Identificou-se também, no processo de criação do conhecimento, que a socialização e a externalização são os dois modos de conversão do conhecimento que ocorrem com maior incidência.

Ao final, são apresentadas recomendações relativas ao comportamento dos gerentes de software e às práticas para a criação de conhecimento, bem como sugestões para estudos futuros.

ABSTRACT

Create software comprehends complex activities full of uncertainty. It usually depends on the implicit knowledge of software engineers. Many times, the know-how and the know-why acquired through each project remain only in personal mental models defined by each engineer. So, it is important to registry and to use the information obtained through technical and management activities to create knowledge about both products and software projects. Either to support the learning and the improvement of software development process.

This paper presents practices and difficulties in information management during the software development process based on the CMM (Capability Maturity Model) level 2.

The strategy used was a close research and descriptive case study. First it was made a documental analysis in order to map the relationship between information management and the software development process based on the CMM level 2. Afterwards a questionnaire was given in order to identify the practices and difficulties in applying the CMM level 2 policies.

The research found out that software managers practise information management when they execute the key process areas policies approved by the CMM level 2. It was also found out that socialization and externalization are the two most used ways of knowledge dissemination.

At last the paper offers recommendations about software managers behaviour and about the practices used to create knowledge. It offers either suggestions for further studies.

LISTA DE FIGURAS

FIGURA 1a – A informação como insumo do processo de software	16
FIGURA 1b – A informação como insumo do processo de software	16
FIGURA 2 – A informação como produto do processo de software	17
FIGURA 3 – Ciclos Simples e Duplo da Aprendizagem Organizacional.....	26
FIGURA 4 – Ciclo de Gerenciamento da Informação.....	27
FIGURA 5 – Ciclo do Conhecimento.....	31
FIGURA 6 – Espiral do conhecimento.....	34
FIGURA 7 – Conteúdo do conhecimento criado pelos quatros modos de conversão	34
FIGURA 8 – A Ciência da Computação e a Engenharia de Software	38
FIGURA 9 – O processo de software	45
FIGURA 10 – Documentos gerados no processo de software	50
FIGURA 11 – Os cinco níveis de maturidade do processo de software	55
FIGURA 12 – Esquema da estrutura do CMM	57
FIGURA 13 – As áreas-chaves de processo por nível de maturidade.....	59
FIGURA 14 – Visibilidade da gerência dentro dos processos de software no nível 2 do CMM.....	63

LISTA DE QUADROS

QUADRO 1 Atividades de engenharia de software em um processo de desenvolvimento de software	46
QUADRO 2 Atividades guarda-chuva no processo de desenvolvimento de software	47
QUADRO 3 Características comuns que organizam as áreas-chaves de processo definidas para um nível de maturidade do CMM.....	58
QUADRO 4 Áreas-chaves de processo por categorias de processo	60
QUADRO 5 Áreas-chaves de processo do CMM nível 2	61
QUADRO 6 Grupos de categorias de análise	65
QUADRO 7 Funcionários com função gerencial por cargo e localização geográfica	69
QUADRO 8 Relação entre as áreas-chaves de processo do CMM nível 2, os processos de gestão da informação e os modos de conversão do conhecimento.....	101
QUADRO 9 Estrutura do questionário.....	102

LISTA DE TABELAS

TABELA 1	Distribuição dos respondentes por grupo de gerentes de software	103
TABELA 2	Distribuição dos gerentes seniores respondentes por projeção regional.....	103
TABELA 3	Distribuição dos líderes de projeto respondentes por projeção regional	103
TABELA 4	Perfil dos respondentes – gerente sênior.....	104
TABELA 5	Perfil dos respondentes – líder de projeto.....	104
TABELA 6	Tamanho do Projeto de Software	105
TABELA 7	Tipos de produto de software	106
TABELA 8	Metodologia de análise utilizado	106
TABELA 9	Fontes de informação utilizadas durante a definição de requisitos	108
TABELA 10	Dificuldades no levantamento e documentação dos requisitos	109
TABELA 11	Revisão dos requisitos levantados e documentados.....	109
TABELA 12	Revisores dos requisitos levantados e documentados.....	110
TABELA 13	Dificuldades para utilização dos requisitos como fonte de informação para as atividades de engenharia e de gestão.....	111
TABELA 14	Uso das medições de requisitos como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe	111
TABELA 15	Fontes de informação utilizadas durante o planejamento	114
TABELA 16	Dificuldades para utilização das estimativas no planejamento das atividades e compromissos do projeto de software.....	116
TABELA 17	Fontes de informação utilizadas durante o acompanhamento e supervisão do projeto de software	118
TABELA 18	Frequência da disseminação das informações de acompanhamento do projeto de software	120
TABELA 19	Avaliação das falhas e acertos do projeto de software	121
TABELA 20	Fontes de informação utilizadas nos processos de garantia da qualidade de software.....	122
TABELA 21	Dificuldades para executar as revisões e auditorias de qualidade do software.....	123
TABELA 22	Frequência da disseminação das medições referentes à qualidade de software.....	124
TABELA 23	Uso das medições de qualidade como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe	124
TABELA 24	Acesso ao repositório de configuração de software	127
TABELA 25	Dificuldades para controlar as mudanças dos produtos de software.....	127

LISTA DE SIGLAS

ACP	- Área-Chave de Processo
CMM	- Capability Maturity Model for Software
CMU	- Carnegie Mellon University
CPMF	- Contribuição Provisória de Movimentação Financeira
DoD	- Departamento de Defesa dos Estados Unidos
IEC	- International Electrothechnical Commission.
IEEE	- Instituto de Engenharia Elétrica e Eletrônica
IRPF	- Imposto de Renda Pessoa Física
ISO	- International Standard Organization
GCS	- Gestão de Configuração de Software
GQS	- Garantia da Qualidade de Software
GR	- Gestão de Requisitos
GSS	- Gestão de Subcontratação de Software
KPA	- Key Process Area
OUUSD (A&T)	- Office of the Under Secretary of Defense for Acquisition and Technology)
PAPS	- Planejamento e Acompanhamento de Software
PSDS	- Processo SERPRO de Desenvolvimento de Software
Receitanet	- Sistema de transmissão de declarações da SRF via Internet da SRF
RUP	- Rational Unified Process
SEI	- Software Engineering Institute
SERPRO	- Serviço Federal de Processamento de Dados
SPICE	- Software Process Improvement and Capability dEetermination
SRF	- Secretaria da Receita Federal
SUNAT	- Superintendência de Administração Tributária
SW – CMM	- Capability Maturity Model for Software
TQM	- Total Quality Management

SUMÁRIO

1. INTRODUÇÃO	12
1.1. Definição do problema e objetivos	14
1.2. Estrutura do trabalho	19
2. A GESTÃO DA INFORMAÇÃO E A CRIAÇÃO DO CONHECIMENTO NAS ORGANIZAÇÕES	20
2.1. A Informação e o seu ciclo de gerenciamento para criar conhecimento organizacional	20
2.2. Os processos de gestão da Informação na criação do conhecimento organizacional	35
2.3. A informação e a criação de conhecimento no contexto das organizações de software	38
3. O PROCESSO DE SOFTWARE E O CMM - MODELO DE MATURIDADE DA CAPACITAÇÃO PARA SOFTWARE	44
3.1. A essência do processo de software	44
3.2. O Modelo de Maturidade da Capacitação para Software (SW-CMM - Capability Maturity Model for Software)	51
4. METODOLOGIA	64
4.1. Estratégia de Pesquisa	64
4.2. Delimitação do Estudo	65
4.3. População do Estudo	68
4.4. Instrumentos de pesquisa	70
4.5. Procedimentos de análise dos dados	71
5. A INFORMAÇÃO E A CRIAÇÃO DE CONHECIMENTO NO PROCESSO DE SOFTWARE FUNDAMENTADO NO CMM NÍVEL 2	73
5.1. Diretrizes comuns	74
5.2. Gestão do projeto de software	75
5.2.1. Planejamento do projeto de software	76
5.2.2. Acompanhamento e supervisão de projeto de software	80
5.3. A Gestão de Requisitos	85
5.4. A Garantia da Qualidade de Software	89
5.5. A Gestão de Configuração de Software	93
5.6. A Gestão de Subcontratação de Software	96
6. RESULTADOS – ANÁLISE E INTERPRETAÇÕES DOS DADOS	102
6.1. Perfil do Respondente	103
6.2. Características do Projeto de Software	105
6.3. Área-Chave de Processo Gestão de Requisitos	107
6.4. Área-Chave de Processo Planejamento de Projeto de Software	113
6.5. Área-Chave de Processo Acompanhamento e Supervisão de Projeto de Software	118
6.6. Área-Chave de Processo Garantia da Qualidade de Software	122
6.7. Área-Chave de Processo Gestão de Configuração de Software	126
7. CONSIDERAÇÕES FINAIS	130
REFERÊNCIAS BIBLIOGRÁFICAS	136

1. INTRODUÇÃO

A década de 90, em especial a sua segunda metade, foi marcada pela evolução do uso de softwares em todos os setores produtivos da sociedade. Hoje, quase tudo que fazemos ou com que interagimos, seja entretenimento, educação, economia, segurança, transportes, saúde, etc passa pelo uso de informação e sistemas de informação que têm como um de seus elementos algum tipo de software de computador.

O software, segundo Pressman (2002), assume um duplo papel, ou seja, ele é um produto e, ao mesmo tempo, o veículo para entrega do produto. Como produto ele disponibiliza o potencial de computação presente no computador ou, mais amplamente, numa rede de computadores local ou na Internet. Por outro lado, o software é um transformador de informação, quer resida em telefone celular, quer opere em um computador de grande porte. Ele produz, gera, adquire, modifica, exibe ou transmite informação. Isto é, o software funciona como um veículo de entrega do produto mais importante da nossa época – a informação.

Em virtude da importância do software na Sociedade da Informação¹ e do dinamismo com que os fatos ocorrem no mundo, a comunidade de Engenharia de Software necessita aprimorar técnicas e metodologias que tornem o desenvolvimento de software mais fácil, rápido, menos oneroso e principalmente de alta qualidade.

No entanto, esta tarefa não é fácil, pois de acordo com Howard Baetjer², citado por Pressman (2002), o software, é um produto especial. Ele agrega um tipo de conhecimento específico, que é tácito, latente, incompleto na sua totalidade e inicialmente encontra-se disperso no grupo. Visto assim, o desenvolvimento de um software é também um processo de aprendizado social que consiste em um diálogo

¹ O termo Sociedade da Informação representa uma profunda mudança na organização da sociedade e da economia, havendo quem a considere um novo paradigma técnico-econômico. É um fenômeno global, com elevado potencial transformador das atividades sociais e econômicas, uma vez que a estrutura e a dinâmica dessas atividades inevitavelmente serão, em alguma medida, afetadas pela infra-estrutura de informações disponível.

² BAETJER, Jr., H., *Software as Capital*, IEEE Computer Society Press, 1998, p.85.

entre usuários e projetistas de software. Trata-se de um processo interativo que a cada nova rodada de diálogo atrai mais conhecimento dos envolvidos, tais conhecimentos são reunidos e embutidos no software.

O processo de desenvolvimento e manutenção de software é denominado simplesmente de processo de software pela comunidade de engenharia de software. Segundo o ponto de vista técnico de Paulk (1993), o processo de software é um conjunto de atividades, métodos, práticas e transformações que as pessoas utilizam para desenvolver e manter um software e seus produtos associados (por exemplo: planos e documentos de desenho³, código, casos de teste e manuais de usuário).

Segundo Pressman (2002), o produto final do processo de software é a informação, que pode ser apresentada na forma de:

1. Programas *de computador* tanto no formato de código-fonte (conjunto de instruções escritas numa determinada linguagem de programação que são interpretadas pelo computador), bem como no de código executável (produto final que é utilizado pelo usuário);
2. *Documentos* que especificam os requisitos e as características do software (tanto para profissionais – documentação técnica e gerencial – como para usuários – manual do usuário);
3. *Dados*⁴ contidos nos programas ou externos a eles e que podem ser fontes de informações estruturadas ou não, por exemplo, banco de dados, arquivos texto, multimídia, etc.

Essas informações são geradas nas atividades técnicas de engenharia de software. A este arcabouço deve-se acrescentar as informações pertinentes às atividades de gestão do processo de software como, por exemplo, gestão do projeto, gestão de configuração e qualidade de software.

³ Consiste na definição de uma estrutura implementável para um produto de software que atenda aos requisitos especificados pelo cliente e/ou usuário.

⁴ Na engenharia de software, dado é entendido como um registro estruturado de transações. E a informação é um dado acrescido de relevância e propósito (valor agregado).

O volume dessas informações pode aumentar exponencialmente com a abrangência e a complexidade do software a ser desenvolvido. Dessa maneira, sem um efetivo gerenciamento das informações coletadas e geradas pode-se chegar a uma situação de caos no ambiente de desenvolvimento de software. Tal situação prejudica a criação de conhecimento e dificulta sua futura utilização para fins de aumento da produtividade das equipes de desenvolvimento de software e da qualidade do produto e do processo de software.

1.1. Definição do problema e objetivos

De acordo com D. J. Flynn, citada por Furnival (1995), relatórios do Reino Unido mostram que 20% do investimento em projetos de sistemas são desperdiçados em softwares nunca entregues ou entregues, mas não usados. O relatório revela ainda que até um bilhão de libras esterlinas extras são gastas em softwares deficientes, isto é, exigem muita manutenção não prevista. Nos Estados Unidos, as agências federais perderam, em 1989, US\$ 17 bilhões em softwares fracos, em comparação com US\$ 9 bilhões em 1982.

Segundo J. A. Ward, citado por Seilheimer (2000), um levantamento feito por uma grande firma de consultoria constatou que 25% dos grandes projetos de software foram cancelados, 60% ultrapassaram os limites de custo, 75% tem problemas de qualidade, e mais importante ainda, menos de 1% de todos os projetos de desenvolvimento de software foram concluídos no prazo previsto e com todos os requisitos de software atendidos.

Quanto à realidade brasileira, de acordo com a 5ª edição da pesquisa Qualidade e Produtividade no Setor de Software Brasileiro⁵ (2001) – realizada pela Secretaria de Política de Informática e Automação do Ministério da Ciência e Tecnologia, cujo trabalho de campo estendeu-se de setembro de 2001 a abril de

⁵ O projeto desta pesquisa foi concebido na época da criação do Subcomitê Setorial da Qualidade e Produtividade em Software, do Programa Brasileiro da Qualidade e Produtividade – SSQP/SW-PBQP. A pesquisa é realizada a cada dois anos com empresas desenvolvedoras de software, visando acompanhar a evolução do setor no aspecto de gestão da qualidade. Em 1999, o escopo da pesquisa passou a incluir questões

2002 - o mercado projetado para o ano 2001 foi de US\$ 6,9 bilhões provenientes da comercialização de software no mercado interno, acrescidos de US\$ 1,021 bilhões de importação de softwares, resultante da remessa em direitos autorais.

De acordo com Paduan (2003), esses números colocaram o Brasil, em 2001, como o sétimo maior mercado de software do mundo. Não há números fechados para 2002, mas os especialistas ainda registram o Brasil em sétimo lugar, empatado com China e Índia. O governo estima que, no ano de 2002, a comercialização de software e serviços relacionados somou US\$ 8,5 bilhões de dólares no país. Para o ano de 2003, a previsão é de US\$ 9,2 bilhões. Tal resultado, quase 1,8% do PIB, colocaria a indústria nacional de software em terceiro lugar entre as 500 maiores empresas brasileiras.

No entanto, a pesquisa do Ministério da Ciência e Tecnologia constatou que as empresas nacionais de software precisam aprimorar sua *produtividade* e se preocuparem com a *qualidade dos produtos* oferecidos ao mercado.

Com relação ao quesito melhoria da capacitação das empresas, a pesquisa constatou, que somente 18% dos entrevistados (80 do total de 446 empresas pesquisadas) admitiu possuir sistemas de qualidade com certificação, sejam eles específicos ou não para a área de desenvolvimento de software. Sendo que, a partir de 1997, observou-se um número crescente no processo de implantação desses sistemas de certificação.

É importante ressaltar que a informação é um dos principais insumos para o desenvolvimento de um software de qualidade. É a partir das informações fornecidas pelo usuário⁶ e/ou cliente⁷ ao analista de requisitos⁸ que o escopo do software⁹ é definido e, por conseguinte, inicia-se todas as demais atividades do processo de desenvolvimento do software (FIG. 1a).

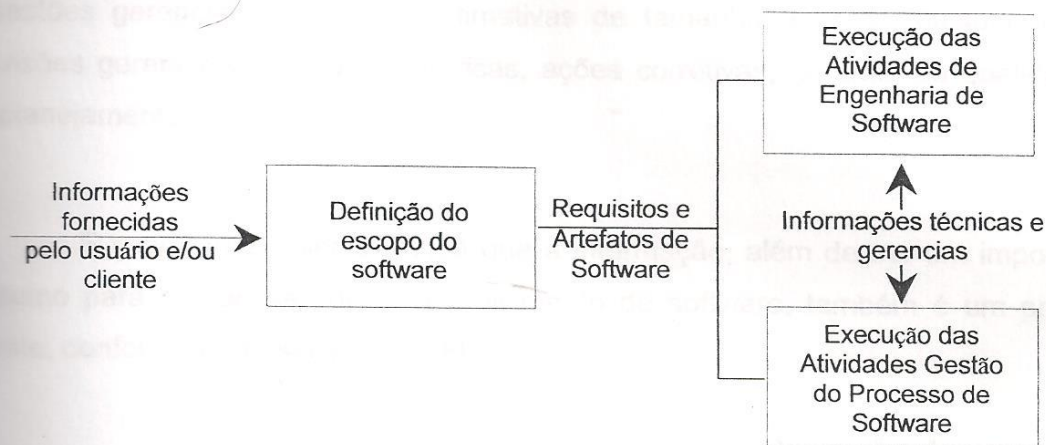


FIGURA 1a – A informação como insumo do processo de software

Além disso, o software, principalmente o gerencial e o que controla transações nas empresas, está inserido em ambientes dinâmicos onde as mudanças são contínuas. Tais mudanças podem afetar os requisitos de software¹⁰ que, portanto, devem ser revistos (FIG. 1b).

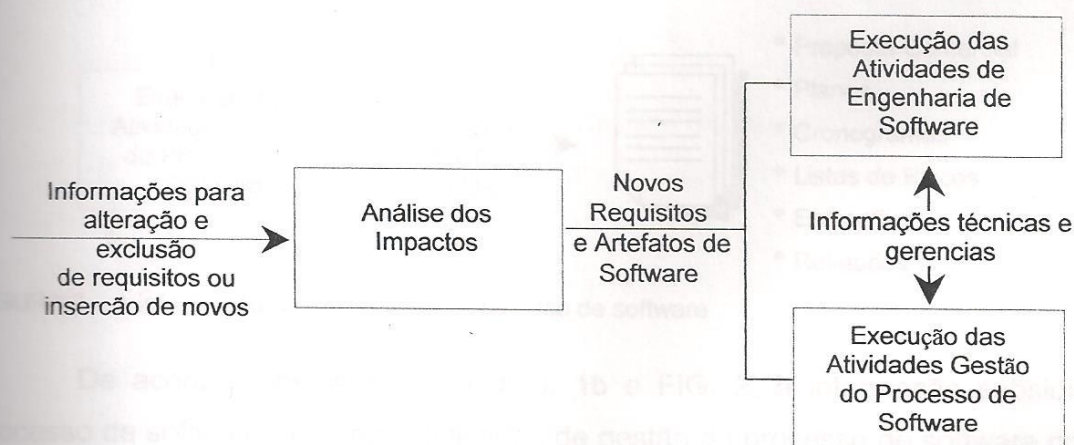


FIGURA 1b – A informação como insumo do processo de software

⁶ Aquele que interage direta ou indiretamente com o software.

⁷ Patrocinador do projeto de desenvolvimento do software, podendo ser ou não usuário do software.

⁸ Profissional responsável pelo levantamento, análise e documentação dos requisitos de software.

⁹ Define as características e funções do software. É medido em relação aos requisitos de software.

¹⁰ Característica intrínseca do software que é necessária para solucionar um problema do usuário ou atender um contrato, padrões, especificações ou outra documentação formal. Os critérios de aceitação do software pelo cliente são definidos com base nos requisitos.

Logo que seja constatada a necessidade de alterar, excluir requisitos ou inserir outros novos, é preciso analisar os possíveis impactos gerados. Estes impactos podem abranger questões técnicas – como, por exemplo, análise, definição ou redefinição de tecnologia, codificação, testes, revisões técnicas, documentação, qualidade, configuração de software. Ademais, os impactos podem envolver questões gerenciais tais como estimativas de tamanho, esforço, custo e prazo, revisões gerenciais, controles, métricas, ações corretivas, garantia da qualidade e replanejamento.

É importante salientar ainda que a informação, além de ser um importante insumo para o processo de desenvolvimento de software, também é um produto deste, conforme pode ser visto na FIG. 2.

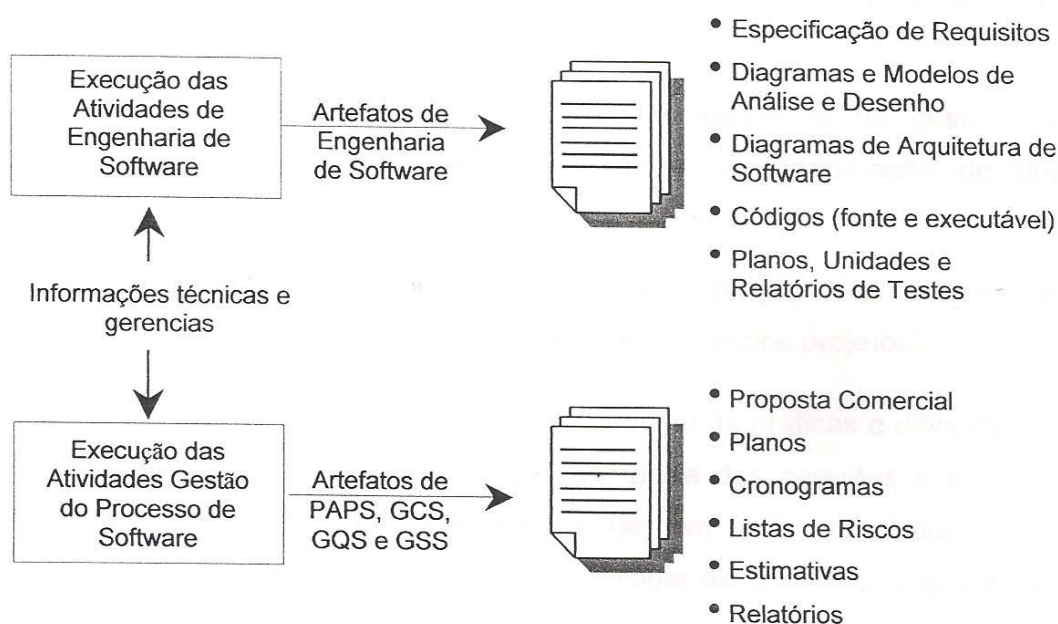


FIGURA 2 – A informação como produto do processo de software

De acordo com as FIG. 1a, FIG. 1b e FIG. 2, a informação subsidia o processo de software tanto nas atividades de gestão do processo de software como nas atividades de engenharia de software. Porém, segundo Humphrey (*apud* Paulk, 1993), é fundamental primeiramente focar os esforços na qualidade da gestão de projeto, antes de se implementar o processo de engenharia, isto porque sem a disciplina gerencial, o processo de engenharia é prejudicado pelas pressões de prazo e de custo.

Sendo assim, este estudo utilizou como referencial o modelo de qualidade SW-CMM Capability Maturity Model for Software (Modelo de Maturidade da Capacitação para Software). Ele aborda os processos de software e está estruturado em cinco níveis de maturidade, sendo que cada nível aborda um conjunto de metas de processo que determina a capacitação do processo de software de uma organização. No entanto, para fins desta pesquisa, focou-se o nível 2 que contempla as questões relativas à gestão do projeto de software.

Portanto, considerando o volume das informações coletadas e geradas durante a execução de um projeto de software e a importância destas para a criação do conhecimento na engenharia de software, surgem as seguintes questões:

- Os gerentes de software e os técnicos de engenharia de software praticam a gestão das informações coletadas e geradas nos projetos de desenvolvimento e/ou manutenção de software?
- Quais são os fatores que dificultam a realização da gestão da informação durante o desenvolvimento e/ou manutenção de um software?
- O conhecimento adquirido durante a execução dos projetos é registrado, disseminado e aproveitado nos próximos projetos?

Neste contexto, o objetivo deste estudo é verificar as práticas e dificuldades no processo de gerenciamento da informação por parte dos gerentes e técnicos durante o desenvolvimento e/ou manutenção de um software. Para tanto, selecionou-se para estudo, uma empresa de tecnologia da informação que tenha como referencial o Modelo de Maturidade da Capacitação para Software - CMM.

Dessa maneira, entende-se que a análise, sob a luz da Ciência da Informação, do processo de software com ênfase no CMM, mais especificamente no nível 2, poderá constituir o início de um trabalho para obtenção de insumos sobre as boas práticas e os obstáculos no processo de construção de conhecimento sobre o projeto, o produto e o processo de software.

1.2. Estrutura do trabalho

Esta pesquisa foi estruturada em sete capítulos. O primeiro introduz o tema e a problemática, o universo de pesquisa, o objetivo geral e as justificativas do estudo.

O segundo e o terceiro capítulos apresentam a fundamentação teórica. O segundo apresenta conceitos sobre a gestão da informação com vistas a criar conhecimento organizacional, assim como a contextualização dessa temática no ambiente das organizações de software. O terceiro capítulo aborda os conceitos relacionados ao processo de software e ao Modelo de Maturidade da Capacitação para Software - SW-CMM. Trata-se de uma revisão sintética da literatura sobre os temas, porém importante para o entendimento da sistemática do desenvolvimento de um software e das diretrizes definidas no CMM.

O quarto capítulo caracteriza o tipo de pesquisa e descreve os procedimentos da coleta e análise dos dados.

O quinto capítulo apresenta o primeiro resultado deste estudo. Ele estabelece relações entre a informação e o processo de software fundamentado no nível 2 do CMM. Nestas relações, é evidenciada a inserção da informação no contexto da gestão de requisitos e das atividades guarda-chuva do processo de desenvolvimento de software. Procurou-se realizar essa leitura a luz dos conceitos de gerenciamento da informação e de conversão do conhecimento postulados por Choo (1998a, 1998b) e Nonaka & Takeuchi (1997), respectivamente.

O sexto capítulo descreve a análise dos dados e a discussão dos resultados obtidos. Por fim, o sétimo capítulo apresenta as principais conclusões e recomendações para estudos futuros.

2. A GESTÃO DA INFORMAÇÃO E A CRIAÇÃO DO CONHECIMENTO NAS ORGANIZAÇÕES

2.1. A Informação e o seu ciclo de gerenciamento para criar conhecimento organizacional

A base teórica deste estudo aborda conceitos e idéias sobre gestão da informação e a criação do conhecimento nas organizações. Destacou-se na literatura sobre a temática alguns autores mais pertinentes às questões propostas para norteá-lo.

De acordo com McGee & Prusask (1994), nas próximas décadas a informação, mais do que terra ou capital, será a força motriz na criação de riquezas e propriedades. Neste tipo de economia, o sucesso é determinado pelo que você sabe e não pelo que você possui.

McGee & Prusask (1994) destacam ainda que em uma economia onde a informação tem papel importante, a concorrência entre as organizações baseia-se em sua capacidade de adquirir, tratar, interpretar e utilizar a informação de forma eficaz. As organizações que liderarem essa competição serão as grandes vencedoras do futuro, enquanto as que não o fizerem poderão ser vencidas pelos concorrentes.

Para Choo (1998a), a informação é o recurso estratégico da organização, o que a coloca em um patamar que vai além de agente de produção. Na realidade, a informação é o recurso que possibilita a efetiva combinação e utilização dos demais agentes de produção. Ela é um meta-recurso que coordena a mobilização dos outros recursos de produção com a finalidade de incrementar o desempenho da organização.

Ainda neste contexto, McGee & Prusask (1994) afirmam que embora a informação seja um ativo que precisa ser administrado da mesma forma que os outros tipos de ativos representados pelos seres humanos, capital, propriedades e bens materiais, ela representa uma classe particular dentre estes. Isso porque a informação é infinitamente reutilizável, não se deteriora nem se deprecia, e seu valor é determinado exclusivamente pelo usuário.

Para Eaton & Bawden, citados por Silveira (2002), existem algumas diferenças-chaves entre a informação e os recursos tangíveis de uma organização. As características da informação que evidenciam sua distinção são as seguintes:

1. *Valor da informação* - não é prontamente quantificável, pois a informação não tem valor intrínseco, seu valor depende de seu contexto e de seu uso;
2. *Consumo da informação* - a informação não é perdida quando é dada a alguém, seu compartilhamento e transmissão podem causar acréscimo;
3. *Dinâmica da Informação* - a informação não pode ser considerada como um recurso estático que se acumula e estoca dentro de um sistema confinado, mas sim como uma força dinâmica dentro do sistema no qual está inserida;
4. *Ciclo de vida da informação* - a idéia de ciclo de vida é uma simplificação imprópria, pois a informação pode possuir múltiplos ciclos de vida e ser extremamente variável;
5. *Individualidade da informação* - a informação aparece de diferentes formas e é expressa de diferentes maneiras, mas somente adquire valor no contexto de uma situação individual.

Para Silveira (2002), a informação, quando considerada como recurso, é importante para a organização e para os elementos da organização. Sua importância pode ser demonstrada através do modo como ela é usada, por quem, e para que, ressaltando o propósito de contribuir para o contexto na qual está inserida. Além disso, é necessário compreender que fatores culturais e organizacionais

influenciam na adoção e implementação de práticas de gerenciamento da informação na organização.

Para Rowley (1998), cada área de conhecimento possui uma interpretação do que seja informação. Por conseguinte, apesar de inúmeros debates ocorridos nas mais variadas áreas (ciência da computação, ciência da informação, administração, economia), sobre o gerenciamento da informação, ainda não há unanimidade no entendimento do objeto informação. Uma vez que cada uma das disciplinas citadas possui uma visão diferenciada deste objeto, não há uma teoria única e compartilhada de informação.

Na perspectiva de Ponjuan Dante (1998), citado por Marchiori, a gestão da informação “inclui em dimensões estratégicas e operacionais, os mecanismos de obtenção e utilização de recursos humanos, tecnológicos, financeiros, materiais e físicos para o gerenciamento da informação. E a partir disto, ela mesma pode ser disponibilizada como insumo útil e estratégico para indivíduos, grupos e organizações” Marchiori (2002, p.74).

Ao considerar o contexto da administração organizacional, Choo (1998a) destaca que muitas das informações que têm impacto sobre a organização são insinuações sutis (conversas informacionais). Isto é, representam mais um potencial do que uma prescrição para a ação. Para se tornar estratégica, a informação precisa ser transformada em conhecimento para assim guiar a ação. Na perspectiva de Davenport & Prusak (1999), para que a informação se transforme em conhecimento, os seres humanos precisam fazer um processamento cognitivo da informação por meio das seguintes atividades:

- *Comparação*: de que forma as informações relativas a esta situação se comparam a outras situações conhecidas?
- *Conseqüências*: que implicações estas informações trazem às decisões e tomadas de ação?
- *Conexões*: quais as relações deste novo conhecimento com o conhecimento já acumulado?

- *Conversação*: o que as *outras* pessoas pensam desta informação?

Para Choo (1998a), essa transformação da informação em aprendizado e *insight*, assim como o compromisso com a ação, muitas vezes equivale a administrar recursos de informação, ferramentas e tecnologias de informação, ou políticas e padrões de informação. Embora todas essas funções sejam necessárias, também é importante definir uma estrutura unificadora que aglutine todas essas funções. Os recursos de informação, as ferramentas tecnológicas e os padrões de política constituem a infra-estrutura tecnológica da administração da informação. Acima dessa infra-estrutura, a geração e transformação da informação são moldadas pela cultura organizacional, pela maneira como a organização interpreta seus propósitos e suas ações, e pela especificação de regras, rotinas e papéis. E como resultado a organização obtém um novo conhecimento que é utilizado para sustentar e incrementar o desempenho da organização em um ambiente instável e competitivo.

Choo (1998a) aponta diferentes tipos de conhecimentos existentes em uma organização:

1. *Conhecimento tácito* – é o conhecimento implícito usado pelos membros da organização para realizar seu trabalho e dar sentido a seu mundo. É difícil de verbalizar porque é expresso por habilidades (técnicas e não técnicas) que são baseadas na ação. Ele não pode se reduzir a regras e receitas. É aprendido durante longos períodos de experiência e de execução de uma tarefa, durante os quais o indivíduo desenvolve uma capacidade para fazer julgamentos intuitivos sobre a realização bem-sucedida da atividade. Apesar de não ser codificado, o conhecimento tácito pode e é normalmente ensinado por meio de exemplos e compartilhado pelo uso de analogias, metáforas, modelos e histórias. É vital para a organização porque as empresas só podem aprender e inovar estimulando de algum modo o conhecimento tácito de seus membros. Ele torna-se substancialmente valioso quando se transforma em novas competências, produtos e serviços.
2. *Conhecimento baseado em normas ou explícito* – é aquele que pode ser expresso formalmente com a utilização de símbolos, podendo

portanto ser facilmente comunicado ou difundido¹¹. O conhecimento explícito pode se basear em objetos ou regras. O conhecimento baseado em objetos pode ser encontrado, por exemplo, em especificações de produtos, patentes, códigos de software, banco de dados automatizados, desenhos técnicos, protótipos e outros. O conhecimento explícito é baseado em regras quando é codificado em normas, rotinas ou procedimentos operacionais ou padrões. Cada organização deve desenvolver seu próprio repertório de rotinas, baseado em sua experiência e no ambiente específico em que ela opera. Além disso, o conhecimento explícito facilita a transferência de aprendizado dentro da organização.

3. *Conhecimento cultural* – faz parte da cultura organizacional e é comunicado oral e verbalmente por meio de histórias, metáforas, analogias, exposição das visões e missões organizacionais. Inclui suposições e crenças que são usadas para descrever e explicar a realidade, bem como convenções e expectativas que são usadas para agregar valor e significado para as novas informações. Essas crenças, regras e valores compartilhados formam a estrutura na qual os membros da organização constroem a realidade, reconhecem a importância de novas informações e avaliam ações e interpretações alternativas.

Os três tipos de conhecimentos organizacionais são interdependentes e podem ser encontrados em qualquer organização, porém as organizações inteligentes¹² se especializam na expansão, renovação e atualização desses conhecimentos. Para isso, elas promovem a acumulação do conhecimento tácito para aumentar a habilidade e a capacidade criativa dos seus empregados. Elas usufruem da vantagem do conhecimento explícito para maximizar a eficiência e a transferência de aprendizado. E, por fim, elas desenvolvem o conhecimento cultural

¹¹ NONAKA, Ikujiro; TAKEUCHI, Hirotaka. Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação. Rio de Janeiro: Campus, 1997.

¹² São organizações especializadas em coordenar seus recursos e competências informacionais para transformar informação em conhecimento. E por conseguinte, utilizar-se deste conhecimento para sustentar e incrementar o seu desempenho em um ambiente instável e competitivo.

que modela as finalidades e os significados dentro da sua comunidade CHOO (1998b).

Essa nova visão de gerenciamento da informação implica em mudanças técnicas e culturais na organização. Tais mudanças resultam em novos aprendizados e adaptações. Para McGee & Prusask (1994), nas organizações inseridas em ambiente estável, onde o ritmo de mudanças é moroso, o aprendizado e a adaptação podem ser lentos. Neste contexto, o método de tentativa e erro torna-se uma estratégia de aprendizado eficaz e provavelmente a mais apropriada. No entanto, em ambientes instáveis, onde as mudanças ocorrem com rapidez, o aprendizado explícito e sistemático é básico para a estratégia de manutenção e adaptação das organizações a seus ambientes.

Na opinião de McGee & Prusask (1994), no ambiente econômico contemporâneo, as organizações que não sejam capazes de adaptação morrem rápida e conscientemente. Encontrar uma solução eficaz para o problema de definição, execução e integração depende do uso e gerenciamento eficazes da informação. O encontro de soluções subseqüentes, à medida que o ambiente é modificado, torna-se dependente do aprendizado, e o aprendizado depende do uso e gerenciamento efetivos da informação.

Neste contexto, Choo (1998a) afirma que o gerenciamento da informação deve ser visto como o gerenciamento de uma rede de processos que adquire, cria, organiza, distribui e usa a informação. Sendo assim, ele propõe um modelo de processos de gerenciamento da informação de natureza cíclica e contínua que é mapeado a partir dos ciclos simples e duplo de aprendizagem organizacional¹³.

¹³ C. ARGYRIS, C.; SCHÖN, D. A.. Organizational Learning: A Theory of Organizational Socialization, Research in Organizational Behavior 1 (1979): 1-37.

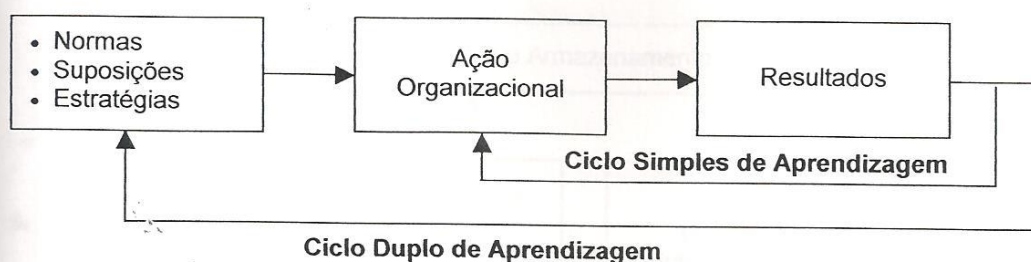


FIGURA 3 – Ciclos Simples e Duplo da Aprendizagem Organizacional

Fonte – Choo, 1998a, p. 15.

A FIG. 3 apresenta os dois ciclos de aprendizagem. O *ciclo simples de aprendizagem* acontece quando as ações organizacionais são suficientes para corrigir as falhas sem ocasionar a contestação das normas organizacionais. O *ciclo duplo de aprendizagem* ocorre quando a correção das falhas requer a modificação das normas organizacionais. Enquanto o ciclo simples é adaptativo e preocupa-se com o controle organizacional, o ciclo duplo é gerador de aprendizagem. Este último relaciona-se com a criação de novos modelos mentais que renovam ou inovam normas, suposições e estratégias que guiam as ações organizacionais.

Os modelos mentais na organização, segundo Kim (1993), fornecem um contexto segundo o qual se deve observar e interpretar materiais novos, e determinam como a informação armazenada é relevante para uma dada situação. Representam mais do que um conjunto de idéias, memórias e vivências – são como o código-fonte¹⁴ de um software, o gerente e o árbitro da aquisição, da retenção do uso e da eliminação de novas informações. Mas são muito mais do que isso, por que também são como o programador do código-fonte de um software que tem o *know-how*¹⁵ para projetar um código-fonte diferente, assim como tem o *know-why*¹⁶ para escolher qual o melhor código-fonte.

¹⁴ Conjunto de instruções escritas numa determinada linguagem de programação que são interpretadas pelo computador.

¹⁵ Habilidades técnicas que implicam a capacidade física de produzir alguma ação.

¹⁶ Capacidade de articular uma compreensão conceitual de uma experiência.

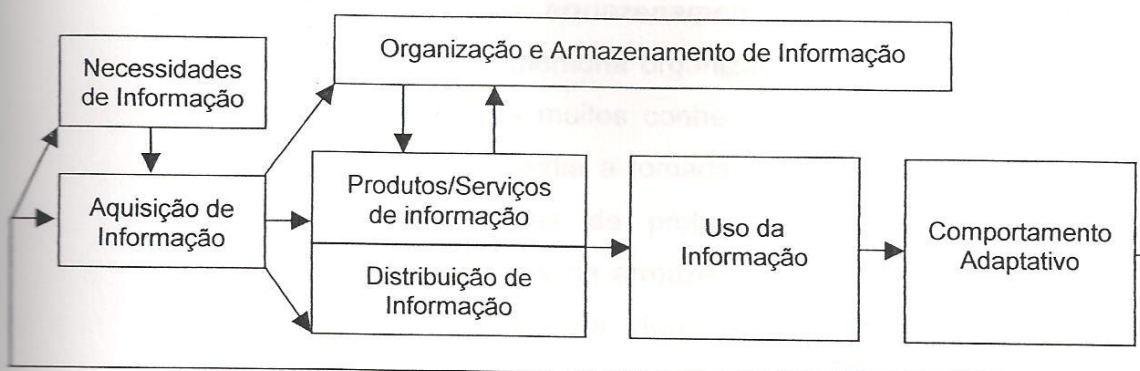


FIGURA 4 – Ciclo de Gerenciamento da Informação

Fonte – Choo, 1998a, p. 24.

A FIG. 4 mostra o ciclo de gerenciamento da informação proposto por Choo (1998a) que tem como base o usuário da informação, bem como as perspectivas situacionais que destacam as dimensões cognitivas e sociais nas atividades de busca e uso da informação. Esse modelo pode ser visto como um ciclo contínuo de atividades correlatas, a saber:

1. As **necessidades de informação** surgem em decorrência de problemas, incertezas e ambigüidades que são típicos nas situações e experiências específicas. Com a finalidade de tornar o ambiente menos obscuro, os membros da organização buscam por informações que melhor atendam a situação problemática enfrentada no momento para então tomar decisões e solucionar problemas.
2. A aquisição **da informação** tornou-se uma função crítica e cada vez mais complexa no gerenciamento da informação. É conduzida de acordo com as necessidades informacionais que refletem a diversidade do ambiente no qual a organização está inserida. Sendo assim, é importante criar uma rede descentralizada, porém controlada e bem administrada, de coleta de informações que envolva os membros da organização. Embora a diversidade de fontes seja aconselhada, de modo a evitar a saturação da informação, é importante que a seleção e o uso das fontes de informação sejam planejados e continuamente monitorados e avaliados como qualquer outro recurso vital para a organização.

3. A **organização e o armazenamento da informação** tem como finalidade criar uma memória organizacional. Esta funciona como um eficaz repositório dos muitos conhecimentos e experiências de uma organização, que subsidia a tomada de decisões, a interpretação de situações, as soluções de problemas, bem como a responder perguntas. Os sistemas de armazenamento da informação são cada vez mais requisitados para oferecer a flexibilidade necessária para captar informações, apoiar as múltiplas visões que os usuários têm dos dados, conectar itens que são funcionais ou logicamente relacionados e permitir que os usuários explorem padrões e conexões. Para melhor estruturar as informações armazenadas no repositório, a organização inteligente utiliza-se de regras e políticas básicas de padronização.
4. Os **produtos e serviços de informação** conseguem agregar valor a organização se a informação fornecida é consistente, relaciona-se com as necessidades e preferências informacionais dos seus usuários. A informação oferecida deve abranger não apenas a área do problema, mas também as circunstâncias específicas que afetam a resolução do problema ou tipo específico de problema. Os produtos e serviços de informação devem apresentar informações resumidas, relacionadas, organizadas e categorizadas, considerando o perfil dos grupos de usuários e suas necessidades de informação. Dessa maneira, aumenta-se a capacidade de busca, seleção e compreensão das informações por parte dos usuários.
5. A **distribuição da informação** é o processo que possibilita a disseminação e o compartilhamento de informações, na organização, oriundas de diversas fontes. Ela tem como principal objetivo propiciar a criação de novas percepções e conhecimento sobre problemas e situações difíceis. É importante ressaltar que segundo Hubber¹⁷ (*apud* Choo, 1998a) uma distribuição ampla pretende que:
 - A aprendizagem organizacional torne-se mais consistente e freqüente;

¹⁷ G. P. Hubber, Organizational Learning: the Contributing Procesess and Literature, em Organizations Science.

- A recuperação da informação seja mais promissora;
 - Novas informações possam ser criadas por meio da reunião de itens díspares.
6. O **uso da informação**, de forma geral, envolve a seleção e o processamento da informação de modo a responder a uma pergunta, resolver um problema, tomar uma decisão, negociar uma posição ou entender uma situação. No ambiente organizacional, tem como objetivo criar conhecimento, não apenas na razão lógica de dados e fatos, mas na forma de representações que fornecem significado e contexto para ações objetivas. Dessa maneira, a organização faz uso da informação para efetuar a construção social da realidade e assim imprimir significado¹⁸ as experiências organizacionais. Além disso, ela também usa a informação para fundamentar as tomadas de decisões, assim como para construir conhecimento.

Em cada caso, o uso da informação é um processo social de investigação espontâneo, recíproco e repetitivo. Nesse processo investigatório, os participantes esclarecem e contestam mutuamente suas representações e crenças. Sendo que as escolhas podem levar mais em consideração a intuição ou negociações políticas do que as análises racionais.

Isto posto, o grande desafio do gerenciamento da informação é projetar e criar estruturas e processos de informação que sejam tão flexíveis, enérgicos e permeáveis quanto os processos de investigação e tomada de decisões que eles tentam dar suporte.

Ainda analisando a FIG. 4, o resultado do uso eficiente da informação é o comportamento adaptativo, ou seja, a seleção e execução de ações dirigidas para objetivos, mas que também reagem às condições do ambiente. As reações da organização interagem com as ações de outras organizações, gerando novos sinais e mensagens significativos e, dessa forma, mantendo novos ciclos de uso da informação.

Conforme a literatura, autores enfatizam três situações nas quais a criação e o uso da informação desempenham um papel estratégico na determinação da competência organizacional de crescer e adaptar-se. A primeira acontece quando a organização usa a informação para compreender as mudanças e acontecimentos presentes no seu ambiente externo com a finalidade de garantir a sua adaptação e assim prosperar. A segunda situação de uso estratégico da informação ocorre quando a organização cria, organiza e processa informação para gerar novos conhecimentos através da aprendizagem organizacional. Os conhecimentos novos possibilitam o desenvolvimento de novas competências, novos produtos e serviços, uma melhor percepção das oportunidades de negócio, bem como aperfeiçoamento dos processos organizacionais. A terceira situação ocorre quando a organização busca por informações para que estas sejam avaliadas e, em seguida, subsidiem o processo de tomada decisão Choo (1998b).

Embora essas linhas sejam abordadas de formas distintas e separadas, o autor acredita que essas três situações de uso da informação – construção de significado, criação do conhecimento e a tomada de decisão – são processos altamente interconectados e que por meio da análise da interação das três atividades, chega-se a uma visão holística do uso da informação organizacional.

Durante a *construção de significado* o principal processo informacional é a interpretação das notícias e mensagens a respeito do ambiente no qual a organização está inserida. No decorrer da *criação do conhecimento* o processo informacional mais importante consiste na conversão do conhecimento. Já na *tomada de decisão* verifica-se que a atividade-chave é o processamento e análise das informações referentes às possíveis alternativas para que sejam avaliados os seus prós e contras. Os três modos de uso da informação – interpretação, conversão e processamento – compreendem processos sociais dinâmicos que continuamente constroem e reconstroem sentido, conhecimento e ação. Sendo assim, Choo (1998b) acredita que a organização que é capaz de integrar efetivamente as atividades de construção de significado, de criação do conhecimento e de tomada de decisão pode ser descrita como uma organização do conhecimento.

¹⁸ O foco da construção do significado (sense making) é compreendido como a atividade humana de observação,

Dessa forma, retomando o conceito de organização inteligente definido por Choo (1998a), podemos considerar que a organização do conhecimento é uma evolução da organização inteligente, ou seja, é aquela que já estruturou suas atividades de gerenciamento da informação. Ela adota um modelo para o uso desta informação que inter-relaciona os processos de construção do significado, criação do conhecimento e tomada de decisão. Esse modelo possibilita construir um ciclo contínuo de aprendizagem e adaptação denominado de “ciclo do conhecimento” (FIG. 5).

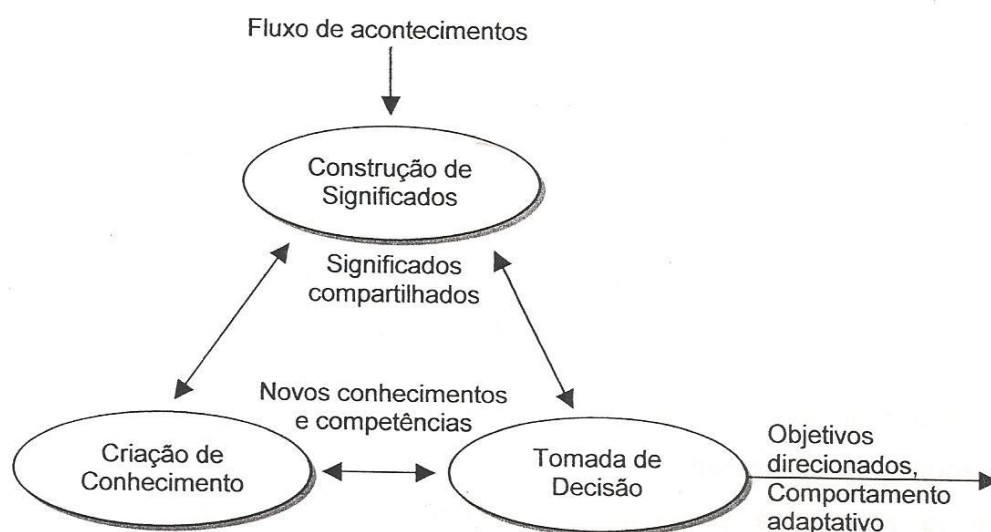


FIGURA 5 – Ciclo do Conhecimento

Fonte – Choo, 1998b, p. 18.

De acordo com Choo (1998b), o gerenciamento dos recursos e processos informacionais faz com que a organização do conhecimento seja capaz de:

- Adaptar-se às mudanças do ambiente de forma ágil e eficaz;
- Envolver-se em processos contínuos de aprendizagem organizacional, incluindo o desprendimento de percepções, normas e modelos mentais que não mais se enquadram a uma nova realidade;
- Mobilizar o conhecimento e as competências dos seus membros de maneira a gerar a inovação e a criatividade;
- Centrar sua inteligência e conhecimento em ações lógicas e decisivas.

interpretação e compreensão do mundo exterior, inferindo-lhe sentidos lógicos advindos do uso e interação com instituições, mídia, mensagens e situações.

Dessa forma, com relação aos modelos de uso da informação organizacional definidos por Choo (1998b), para fins de desenvolvimento dessa pesquisa, nos ateremos mais especificamente à criação do conhecimento sob a luz dos conceitos de Nonaka e Takeuchi sobre o tema.

Esta abordagem foi escolhida porque o desenvolvimento de software consiste em um processo de atividades complexas e carregadas de incertezas, sendo fomentado, na sua maioria, pelo conhecimento tácito dos engenheiros de software. Além disso, o *know-how* e *know-why* adquiridos a cada projeto de desenvolvimento de software são retidos em modelos mentais individuais definidos por cada profissional. Isto, de acordo com Nonaka & Takeuchi (1997), dificulta a inovação e o desenvolvimento de novos produtos.

O conhecimento organizacional, segundo Nonaka & Takeuchi (1997), é analisado sob os aspectos de duas dimensões:

1. A primeira é a epistemológica, onde é abordada a distinção entre os conhecimentos tácito e explícito. Para os autores, os dois tipos de conhecimentos não são entidades totalmente separadas e, sim, mutuamente complementares. Na realidade, um interage com o outro e juntos realizam trocas nas atividades criativas dos seres humanos.
2. A segunda que é a ontológica, trata a questão da amplificação do conhecimento, isto é, a ampliação do nível individual até o nível interorganizacional, significando a difusão do conhecimento do indivíduo até a construção de uma rede entre as organizações. A criação do conhecimento representa, na realidade, a interação de informações, idéias e ações que deve existir entre os indivíduos para que o conhecimento possa ser gerado.

Sob a perspectiva das dimensões epistemológica e ontológica, Nonaka e Takeuchi postulam quatro modos de conversão do conhecimento que são experimentados pelas pessoas para criar conhecimento. Os modos de conversão são:

- *Socialização (tácito em tácito)* – ocorre quando, por meio da observação, da imitação e da prática, um indivíduo compartilha com outro algum conhecimento tácito (experiências, modelos mentais e habilidades técnicas) que possui, sem que sejam produzidos novos *insights*.
- *Externalização (tácito em explícito)* – processo no qual o conhecimento tácito torna-se explícito por meio do compartilhamento de metáforas, analogias, modelos ou histórias. Ele pode ser provocado por meio de diálogos que geralmente acontecem durante as reflexões coletivas. Segundo Nonaka e Takeuchi, a externalização consiste na essência do processo de criação do conhecimento organizacional.
- *Combinação (explícito em explícito)* – processo de sistematização de conceitos em um sistema de conhecimento, envolvendo a combinação de conjuntos diferentes de conhecimento explícito. Neste caso os indivíduos trocam e combinam conhecimentos por meio de documentos, reuniões, conversas telefônicas, troca de mensagens eletrônicas ou redes de comunicação computadorizadas. O conhecimento existente, quando utilizado sob perspectivas diferentes, pode produzir outros conhecimentos explícitos em novos formatos.
- *Internalização (explícito em tácito)* – é o processo onde o conhecimento explícito gerado por meio da socialização, da externalização e da combinação é incorporado na forma de conhecimento tácito. Nonaka e Takeuchi (1997), afirmam que na internalização são necessárias a verbalização e a estruturação do conhecimento sob a forma de documentos, manuais ou histórias orais. Uma vez que documentos ou manuais facilitam a transferência do conhecimento explícito para outras pessoas, ajudando-as a vivenciar indiretamente as experiências dos outros profissionais da organização. Está intimamente relacionado ao “aprender fazendo”.

A FIG. 6 mostra que cada um dos quatro modos corresponde a um fator que provoca a espiral do conhecimento. A *socialização* se inicia dentro de um campo de interação capaz de facilitar o compartilhamento de conhecimentos. A *externalização* é provocada pelo diálogo ou reflexão coletiva, nos quais o emprego de uma metáfora ou analogia ajuda as pessoas a externarem o conhecimento tácito, que, de outra maneira, seria mais difícil de ser transmitido. A *combinação* surge por meio da colocação de algum conhecimento já existente e proveniente de outros locais da organização em uma rede e, finalmente, a *internalização* surge a partir do “aprender fazendo” (Nonaka e Takeuchi, 1997).

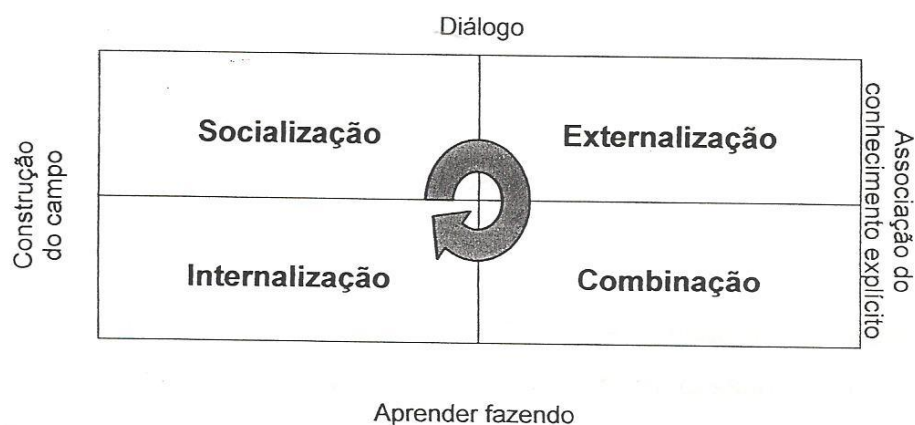


FIGURA 6 – Espiral do conhecimento

Fonte – Nonaka & Takeuchi, 1997, p.80.

De maneira análoga aos fatores acima citados, cada modo de conversão do conhecimento corresponde um tipo de conteúdo.

	Conhecimento Tácito	em	Conhecimento Explícito
Conhecimento Tácito do	Socialização (Conhecimento Compartilhado)		Externalização (Conhecimento Conceitual)
Conhecimento Explícito	Internalização (Conhecimento Operacional)		Combinação (Conhecimento Sistêmico)

FIGURA 7 – Conteúdo do conhecimento criado pelos quatro modos de conversão

Fonte – Nonaka & Takeuchi, 1997, p.81.

Na seção subsequente os processos do ciclo de gerenciamento da informação proposto por Choo (1998a) são analisados sob a luz dos modos de conversão entre os conhecimentos tácito e explícito definidos por Nonaka & Takeuchi (1997).

2.2. Os processos de gestão da Informação na criação do conhecimento organizacional

- **As necessidades de informação**

No processo de criação de conhecimento, segundo Choo (1998b), é preciso elencar as lacunas de conhecimento, de habilidades técnicas e cognitivas existentes na organização. Outrossim, é necessário identificar as fontes informacionais que possam suprir tais lacunas.

O autor também afirma que se a situação for de solução de um problema ou desenvolvimento de um novo produto, informações são necessárias para se obter suficiente clareza de definição e, então, o processo possa ser iniciado. A atividade de definição e estruturação do problema é uma parte necessária do processo de construção do conhecimento. Essa atividade, muitas vezes, exige que os membros da organização dialoguem e reflitam para esclarecer, na medida do possível:

1. As questões referentes à natureza e aos limites da situação problemática;
2. Os objetivos e critérios de desempenho para o desenvolvimento de novos produtos;
3. As tecnologias relevantes e os recursos e competências disponíveis e necessárias.

Ainda na linha de solução de um problema ou desenvolvimento de um novo produto, o autor define dois fatores que determinam os tipos de informação que serão necessárias:

1. A maturidade do projeto tecnológico subjacente ao produto;
2. O grau de alinhamento entre a linha proposta e o mercado consumidor.

Com base nesses tipos de informação, a organização tem condições de definir a estratégia adequada de definição de um novo produto.

- **A aquisição da informação**

Mais do que qualquer fonte impressa ou banco de dados automatizado, as pessoas são as fontes de informação mais valiosas em qualquer organização. As fontes humanas filtram e resumem as informações, enfatizam os elementos importantes, interpretam aspectos ambíguos e em geral oferecem uma comunicação mais rica e satisfatória de uma questão. Ampliar a variedade de informações é quase sempre uma pré-condição para as soluções criativas, enquanto atenuar a variedade de informações ajuda a concentrar esforços de desenvolvimento Choo (1998b).

Durante a construção do conhecimento, a aquisição da informação caracteriza-se por uma intensa pesquisa e partilha de informações, assim como pelo uso de fontes de informação que proporcionem acesso a experiências externas ao grupo ou a organização. Neste contexto, Choo (1998b) aponta dois principais obstáculos nesse processo:

1. Dificuldade de encontrar as fontes certas e transferir as informações disponibilizadas por estas fontes para o grupo e para a organização como um todo;
2. Dificuldade de reter e tornar explícito o conhecimento tácito dos especialistas.

- **A organização e o armazenamento da informação**

Na opinião de Choo (1998b), as duas principais finalidades dos sistemas de armazenamento de informações na criação do conhecimento são:

1. Focalizar fontes de experiência dentro da organização;
2. Recuperar relatórios de trabalho anteriores ou problemas semelhantes.

A primeira finalidade refere-se à manutenção de uma lista dos especialistas internos e suas respectivas habilidades individuais como parte do conhecimento armazenado da organização. A segunda finalidade está relacionada com a importância de se elaborar um sistema de classificação seguro, porém flexível no que tange a recuperação de informações de pesquisas e documentos de projetos passados.

É importante ressaltar que um sistema de organização e armazenamento de informações bem estruturado e indexado oferece acesso ao conhecimento explícito acumulado pela organização, o que pode agilizar o processo de criação do conhecimento.

Como definido anteriormente, a utilização dos conhecimentos explícito e tácito são complementares e interdependentes. A prática de know-how tácito quase sempre implica o uso de planos, documentos e instrumentos impressos. A leitura e interpretação do conhecimento explícito, por sua vez, quase sempre requerem a contribuição dos conhecimentos individuais.

- **A distribuição da informação**

Durante a criação do conhecimento, segundo Choo (1998b), em especial no contexto da resolução de problemas ou do desenvolvimento de inovações, as pessoas trabalham em equipes de projeto que combinam várias experiências e especialidades. Elas compartilham intensamente as informações em diálogos de grupo, extraem conhecimentos de fora da organização e descobrem novos conceitos por meio do uso de analogias e metáforas.

- **Uso da informação**

As crenças dos indivíduos e da organização incorporadas nas rotinas e nas normas de avaliação determinam tanto a utilidade da nova informação como o valor do novo conhecimento a ser adquirido. Sendo assim, pessoas, recursos, ferramentas e processos precisam ser reunidos para transformar conhecimento em produtos e em competências essenciais e reais.

2.3. A informação e a criação de conhecimento no contexto das organizações de software

Antes de contextualizar a informação e a criação do conhecimento nas organizações de software, é importante definir o conceito de engenharia de software.

A engenharia de software, ao contrário da ciência da computação, não tem como objetivo final a pesquisa básica em hardware ou teoremas sobre algoritmos¹⁹. Sua função é focalizar o computador como uma ferramenta que auxilia na solução de problemas. Isto é, o engenheiro de software trabalha com as funções do computador para implementar uma solução para um determinado problema.

A relação entre a ciência da computação e a engenharia de software pode ser visualizada na FIG. 8.

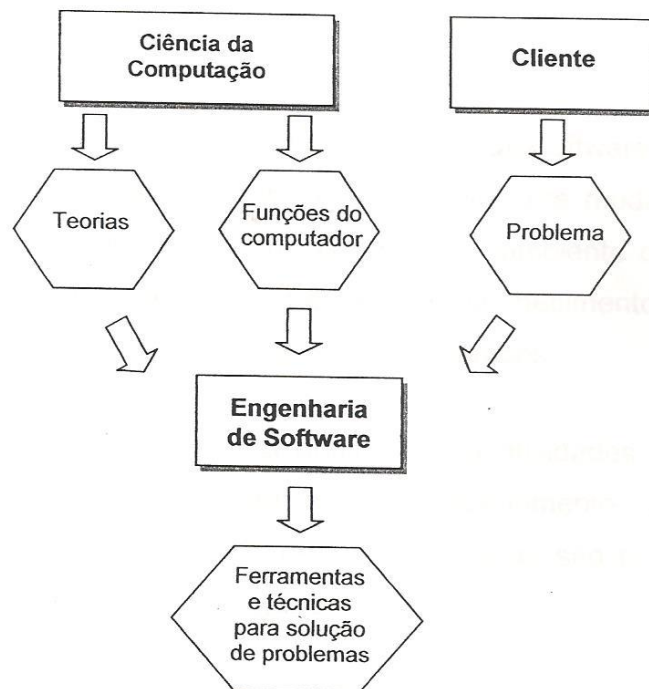


FIGURA 8 – A Ciência da Computação e a Engenharia de Software

Fonte – PFLEEGER, 2001, p. 5.

¹⁹ Conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em número finito de etapas (Houaiss, 2001).
 Conceito central da codificação de software, pois codificar software é basicamente construir algoritmos (Guimarães e Lages, 1985)

De acordo com Paula (2001), a Engenharia de Software não se confunde com a Ciência da Computação, e nem é uma disciplina desta, tal como a Engenharia Metalúrgica não é uma disciplina da Física dos Metais, nem a Engenharia Elétrica é uma disciplina da Física da Eletricidade. Como toda engenharia, a Engenharia de Software usa resultados da ciência, no caso a Ciência da Computação, e fornece problemas para estudo desta. Sendo assim, são vocações profissionais completamente distintas, tão distintas quanto as vocações do engenheiro e do físico, do médico e do biólogo, do político e do cientista político.

Visto assim, pode-se observar que a engenharia de software se preocupa com o software como produto. Dessa forma, utiliza processos, ferramentas, métodos de gestão e técnicas para desenvolver software de alta qualidade da maneira mais eficiente possível. Neste contexto, a comunidade de engenheiros de software cria conhecimento quando desenvolve atividades relativas à aprendizagem, à captura, e ao reuso de experiências relacionadas com o desenvolvimento do produto de software.

A gestão do conhecimento na engenharia de software é importante, pois o desenvolvimento de software é um processo que sofre mudanças muito rápidas. Além disso, os conhecimentos técnicos e sobre o ambiente onde o software será inserido está disperso no grupo. Portanto, o conhecimento na engenharia de software é diverso, volátil e alcança grandes proporções.

De acordo com a literatura analisada, as atividades de identificação das necessidades, aquisição, organização e armazenamento, desenvolvimento de produtos e serviços, distribuição e uso da informação são tarefas imprescindíveis para a criação do conhecimento organizacional.

Sendo assim, para melhor compreensão do ambiente informacional em estudo é preciso identificar quais as necessidades informacionais existentes nas organizações desenvolvedoras de software, mais especificamente, aquelas relacionadas direta e indiretamente com o processo de desenvolvimento de software. Estas necessidades informacionais estão relacionadas com:

- ***Diminuição dos prazos e dos custos e aumento da qualidade*** - refere-se à repetição das boas práticas, oriundas de projetos antigos, em novos projetos de desenvolvimento de software. Isto proporciona a redução do tempo gasto refazendo tarefas e produtos desnecessariamente, o que, por conseguinte, pode viabilizar maior produtividade, menores custos e aumento da qualidade do software;
- ***Tomadas de decisões mais eficazes*** - durante o desenvolvimento de software as pessoas envolvidas no processo tomam decisões, sejam elas técnicas ou gerenciais. Portanto, é importante embasar tais decisões com informações consistentes, confiáveis e não apenas fundamentadas na experiência dos técnicos e dos gerentes alocados no desenvolvimento de um software;
- ***Aquisição de conhecimento sobre novas tecnologias*** - consiste na aquisição rápida, por parte dos técnicos, de proficiência em novas tecnologias;
- ***Aquisição de conhecimento sobre o domínio do problema*** - é uma necessidade importante e está relacionada com o domínio do ambiente ou do negócio para o qual o software é desenvolvido;
- ***Compartilhamento do conhecimento sobre as políticas e práticas locais*** - consiste nas informações referentes às bases de softwares já existentes na organização, aos padrões e convenções de programação;
- ***Colaboração e compartilhamento do conhecimento técnico*** - está relacionada com a disseminação e compartilhamento mútuo de habilidades técnicas²⁰ de engenharia de software entre os membros do grupo de desenvolvedores de software.

Com base nessas necessidades informacionais, observa-se que a engenharia de software envolve vários tipos de conhecimentos, de diferentes níveis, tais como técnicos, gerenciais, corporativos, do ambiente onde o software está inserido, do produto de software e do projeto de desenvolvimento deste.

Por outro lado, criar conhecimento por meio do uso dessas informações é uma atividade que exige trabalho metódico, com métodos e recursos bem aplicados para lapidar estas informações. Para tanto, Rus & Lindvall (2002) afirmam que é necessário às organizações de software, além dos desenvolvedores, envolver outras pessoas nessas atividades, como por exemplo, um grupo estruturado de melhoria do processo de software (SPIN – Software Process Improvement) que proporcionaria suporte aos desenvolvedores nas suas atividades diárias ao invés de sobrecarregá-los com esforço extra.

A principal atividade da engenharia de software é desenvolver software, o que gera documentos como, por exemplo, contratos, planos de projeto, especificações de requisitos e da arquitetura de software. Tais documentos registram informações referentes aos problemas ocorridos durante a execução do projeto, bem como as soluções cabíveis que foram empregadas considerando o contexto, as lições aprendidas e as análises ao término do projeto. Em suma, são documentos que registram os pontos fortes e fracos do processo e do produto de software. Inclui também a análise do custo, esforço e prazo estimado *versus* o que foi realizado, o histórico dos eventos ocorridos e os seus impactos para o projeto.

O acesso a esses documentos pelos membros das equipes propicia a criação de um conhecimento que pode ser aproveitado em projetos subsequentes. Além disso, é possível identificar as pessoas envolvidas e por meio de processos de socialização obter o conhecimento tácito adquirido durante o projeto.

²⁰ Referem-se às técnicas relacionadas às atividades de requisitos, análise, desenho, implementação, testes e implantação. Estas atividades estão detalhadas na próxima seção.

Considerando o ambiente de desenvolvimento de software descrito, Rus & Lindvall (2002) esclarecem que as atividades de gestão do conhecimento auxiliam o desenvolvimento de software nas seguintes áreas:

- **Suporte às atividades fundamentais da engenharia de software** por meio do:
 1. Gerenciamento de documentos - foco na autoria, revisão, edição e uso dos documentos, consistindo no principal meio de captura do conhecimento explícito;
 2. Gerenciamento de competências - identifica os especialistas da organização e é visto como uma forma de rastrear o conhecimento tácito que ainda não é passível de ser documentado;
 3. Reutilização de software (estabelecimento de repositório de reúso de código-fonte²¹ e demais artefatos gerados na engenharia de software).
- **Suporte à memória do produto e do projeto de software** – os desenvolvedores executam práticas de engenharia, como a gestão de mudança e controle de versão, documentação sobre decisões de desenho²², implementação, testes, implantação, análise e rastreabilidade²³ de requisitos. Essas atividades fomentam direta ou indiretamente a criação de uma base de informações sobre os projetos e produtos de software desenvolvidos. O uso dessas informações por parte dos engenheiros de software potencializa a melhoria do processo e do produto de software.
- **Suporte à aprendizagem e ao aperfeiçoamento** - a definição de um conjunto de métricas com metas bem definidas permite ao gerente do projeto estimar custo, esforço, prazo, tamanho e outros parâmetros do projeto. Além das informações quantitativas, podem ser consideradas

²¹ Conjunto de instruções escritas numa determinada linguagem de programação que são interpretadas pelo computador.

²² Consiste na definição de uma estrutura implementável para um produto de software que atenda aos requisitos especificados pelo cliente e/ou usuário.

²³ Relacionamento entre os requisitos de software.

também as informações qualitativas, como casos, lições aprendidas, relatos de sucessos e fracassos, problemas e soluções correspondentes. Por meio da indução, generalização e abstração desse conhecimento é possível criar novo conhecimento aplicável em projetos futuros que sejam similares e tenham o mesmo contexto.

Entende-se que este trabalho deve ser analisado no contexto das áreas de *suporte à memória do produto e do projeto de software e suporte à aprendizagem e ao aperfeiçoamento* dessa atividade.

3. O PROCESSO DE SOFTWARE E O CMM - MODELO DE MATURIDADE DA CAPACITAÇÃO PARA SOFTWARE

Neste capítulo são introduzidos os conceitos relacionados ao processo de software e ao Modelo de Maturidade da Capacitação para Software - SW-CMM. Trata-se de uma revisão sintética da literatura sobre os assuntos, mas importantes para o entendimento da sistemática de desenvolvimento de um software e das diretrizes definidas no CMM.

Este capítulo inicia-se com a apresentação de alguns conceitos que exprimem a essência do processo de desenvolvimento e manutenção de software ou simplesmente processo de software como é mais comumente denominado pela comunidade de engenharia de software.

3.1. A essência do processo de software

Processo é um conjunto de passos parcialmente ordenados e constituídos por métodos, atividades, práticas e transformações usadas para atingir uma meta, cuja execução se dá por meio de um projeto. A arquitetura de um processo define a base de sustentação para a ordenação de seus elementos.

Um processo é definido quando pode ser repetido. Deve ser documentado de forma a detalhar: o produto (o que é feito), os passos (quando), os agentes (quem faz), os insumos (o que usa) e os resultados (o que produz). Os processos podem ser definidos por seu maior ou menor detalhamento e os seus passos podem ter ordenação apenas parcial, o que permite a concomitância entre alguns passos. Um conjunto de passos pode ser definido como um subprocesso. Paula (2001).

Na Engenharia de Software as atividades de desenvolvimento, manutenção e aquisição de software podem ser definidas como processos. As atividades de levantamento e a análise dos requisitos, desenho, implementação, testes e implantação podem ser consideradas subprocessos de um processo de desenvolvimento de software.

O processo de software, segundo Humphrey (1995), é a seqüência de passos necessários para desenvolver e manter o software. Quando corretamente estruturado e apresentado, a definição de um processo de software direciona o trabalho dos profissionais.

O processo de software, segundo Paulk (1993), é um conjunto de atividades, métodos, práticas e transformações que as pessoas utilizam para desenvolver e manter um software e os seus produtos associados (por exemplo: planos e documentos de desenho, código, casos de teste, manuais de usuário, etc).

Na mesma linha de Paulk, Pressman (2002) define o processo de software como um arcabouço para as tarefas que são necessárias para se construir um software de qualidade. O autor afirma ainda que o processo de software é o fundamento da Engenharia de Software e que este pode ser caracterizado de acordo com a FIG. 9.

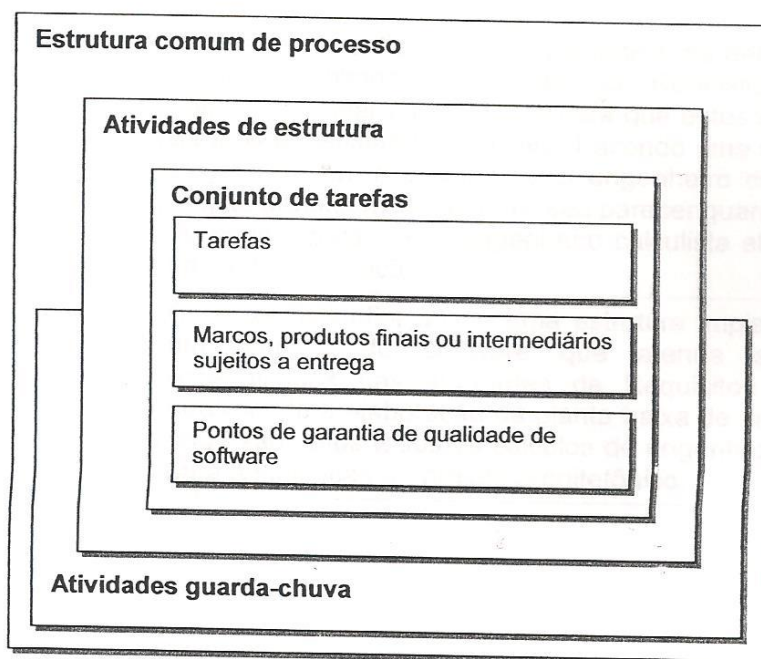


FIGURA 9 – O processo de software

Fonte – PRESSMAN, 2001, p.22.

Como pode ser visto na FIG. 9, uma estrutura comum de processo define um conjunto de atividades de estrutura que são aplicáveis a todos os projetos de software, independentemente de seu tamanho ou complexidade. Cada atividade de

estrutura é composta por conjuntos de tarefas – cada um constituído por marcos de projeto, produtos do trabalho, pontos de garantia de qualidade e por uma coleção de tarefas de engenharia de software conforme descritas no QUADRO 1 – que permitem às atividades de estrutura serem adaptadas de acordo com as características do projeto de software e às necessidades da equipe do projeto.

QUADRO 1
Atividades de engenharia de software em um processo de desenvolvimento de software

(continua)

Atividades	Descrição
Requisitos	Neste momento, os profissionais de engenharia de software começam o processo de captura dos requisitos de software ²⁴ , ou seja, é feito um levantamento das necessidades do usuário e, com base nessas informações, identifica-se o escopo do software. Dessa forma, estabelece-se, entre usuários e desenvolvedores, quais serão as funcionalidades do software a fim de que esse atenda as necessidades apontadas pelos usuários. De maneira análoga, ao término desta atividade tem-se um artefato semelhante ao projeto arquitetônico de uma construção.
Análise	Consiste na verificação da qualidade e no detalhamento dos requisitos obtidos na atividade de Requisitos. Têm como objetivo detalhar os requisitos para que estes sirvam de base para os projetistas de software. Fazendo uma analogia com a Engenharia Civil, é como se o engenheiro civil avaliasse o projeto arquitetônico, dando o seu parecer quanto à viabilidade do projeto para que o engenheiro calculista elabore a planta baixa da construção.
Desenho ²⁵	Consiste na definição de uma estrutura implementável para um produto de software que atenda aos requisitos especificados nas atividades de Requisitos e Análise. É equivalente a elaboração da planta baixa de uma construção, onde são feitos todos os cálculos de engenharia necessários para a execução do projeto arquitetônico.

²⁴ Característica intrínseca do software necessária para solucionar um problema do usuário ou atender um contrato, padrões, especificações ou outra documentação formal. Definem os critérios de aceitação do software.

²⁵ Usada como sinônimo de *design*, e não acepção de desenho pictórico (que equivaleria a *drawing* ou *drafting*). Na Engenharia de Software, muitas vezes o desenho é chamado de projeto. No entanto, o termo projeto será utilizado apenas na acepção de unidade gerencial como uma atividade com início, meio e fim que tem como objetivo desenvolver um produto ou serviço (Paula, 2001).

QUADRO 1
Atividades de engenharia de software em um processo de desenvolvimento de software

Atividades	Descrição
Implementação	É a realização do desenho do software, isto é, consiste na codificação de programas de computador. Seria equivalente ao momento de erguer uma construção.
Testes	É uma prática que visa detectar a maior quantidade possível de defeitos que por ventura não foram detectados na atividade de implementação, considerando os limites de tempo e custo. Assemelha-se aos testes de laboratório realizados na Engenharia Civil.
Implantação	Consiste em disponibilizar o software para o usuário, considerando a configuração do ambiente tecnológico onde o software será inserido, bem como o treinamento dos usuários.

Ainda analisando a FIG. 9, finalmente têm-se as chamadas *atividades guarda-chuva* que são aplicadas ao longo de todo o processo de desenvolvimento de um software. Essas atividades guarda-chuva são complementares, porém independentes a qualquer atividade de estrutura.

QUADRO 2
Atividades guarda-chuva no processo de desenvolvimento de software

Atividades	Descrição
PPS - Planejamento do Projeto de Software	Estabelece planos razoáveis para a execução das atividades de engenharia de software, bem como para a gestão do projeto de software.
APS - Acompanhamento do Projeto de Software	Visa prover visibilidade adequada do progresso real do projeto, de forma que a gerência execute ações corretivas quando o desempenho do projeto desvia significativamente dos planos de software.
GQS - Garantia da Qualidade de Software	Contempla a revisão e a auditoria de produtos de software. Além disso, desenvolve atividades para verificar se os procedimentos e padrões adotados são cumpridos, subsidiando o projeto de software e os gerentes envolvidos com o resultado dessas revisões e auditorias.
GCS - Gestão de Configuração de Software	Consiste em um conjunto de atividades desenvolvidas para administrar as modificações nos itens de configuração que podem ocorrer ao longo de um projeto de desenvolvimento de software. É considerado como item de configuração toda a informação que é criada como parte do processo de desenvolvimento de software.

Os conceitos apresentados sobre processo de software são complementares no que se refere à riqueza de detalhes, entretanto, em todos eles fica evidente que para se construir software de qualidade é necessário seguir um conjunto de atividades bem definidas e estruturadas. Dessa maneira, é preciso estabelecer um modelo de processo de software, ou seja, formalizar uma estratégia que defina de forma clara e objetiva como o arcabouço de atividades e tarefas, descrito na FIG. 9 e nos QUADROS 1 e 2, é organizado. De acordo com Pressman (2002), o modelo de processo de software deve ser escolhido de acordo com:

- A natureza do software e sua aplicação;
- Os métodos e ferramentas a serem utilizados pela equipe desenvolvedora do software;
- Os controles e os produtos intermediários e finais que são requeridos.

O autor afirma ainda que todo o desenvolvimento de software pode ser caracterizado como um ciclo de solução de problema, no qual são encontrados quatro estágios distintos:

- Situação atual – representa a situação atual do ambiente no qual o software será inserido;
- Definição do problema – identifica o problema específico a ser resolvido;
- Desenvolvimento técnico – resolve o problema por intermédio da aplicação de alguma tecnologia;
- *Integração da solução* – entrega os resultados (documentos, programas, dados, nova função de negócio, novo produto, etc) àqueles que solicitaram a solução inicialmente.

Na engenharia de software, esse ciclo de solução de problema é denominado de ciclo de vida do software. Ele é o ponto de partida para definir a arquitetura de um processo de software, pois determina quais as atividades de engenharia e em que ordens devem ser executadas. Existem muitos modelos de ciclo de vida do software, tais como o seqüencial linear, em cascata, em espiral,

prototipagem, incremental, *Unified Process* e o advento de modelos *light*, tais como XP²⁶, no entanto, não entraremos em detalhes pelo fato que esta abordagem extrapolaria o tema deste estudo.

Voltando à FIG. 9, a camada de *atividades de estrutura* constituída por *conjuntos de tarefas* de engenharia de software é estruturada conforme um modelo de ciclo de vida. Assim, é possível verificar que a camada de *atividades de estrutura* é o núcleo técnico do processo de software.

Por outro lado, considerando ainda a FIG. 9, verificamos que a camada de *atividades guarda-chuva* complementa o processo de software inserindo as tarefas gerenciais que visam o controle do projeto de desenvolvimento do software e também o controle dos produtos associados ao software.

Como descrito no capítulo 1, os produtos resultantes do processo de software são informações na forma de programas de computador (código-fonte²⁷ e código executável²⁸), documentos e dados. As informações no formato de documentos podem ter cunho técnico (gerados na camada *Atividades de estrutura*) ou gerencial (gerados na camada *Atividades guarda-chuva*) e constituem a documentação de um software, conforme FIG. 10.

²⁶ XP - EXtreme Programing

²⁷ Conjunto de instruções escritas numa determinada linguagem de programação que são interpretadas pelo computador

²⁸ Produto final que é utilizado pelo usuário.

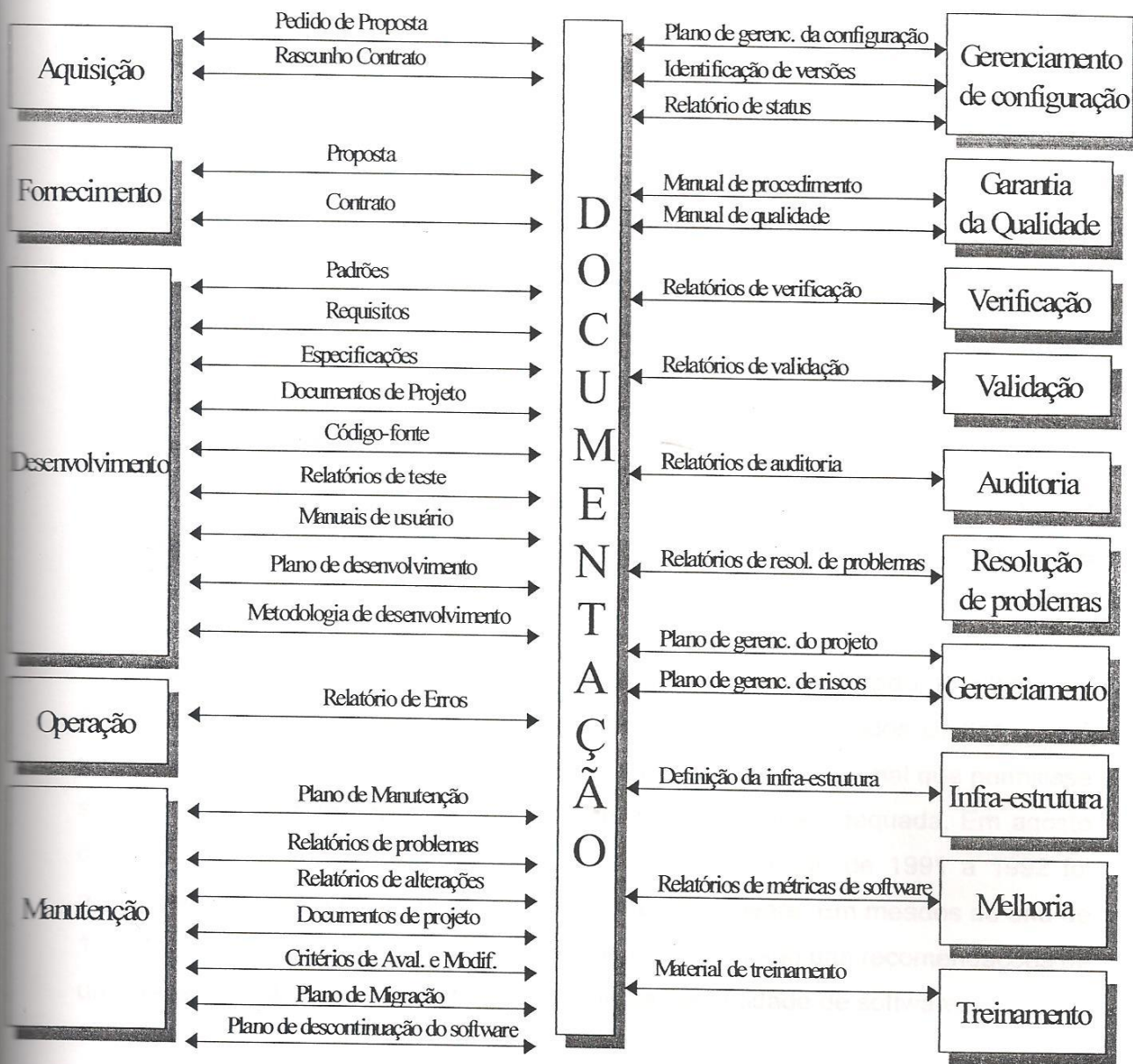


FIGURA 10 – Documentos gerados no processo de software

Fonte - ISO 12207

3.2. O Modelo de Maturidade da Capacitação para Software (SW-CMM - Capability Maturity Model for Software)

O modelo SW-CMM - Capability Maturity Model for Software (Modelo de Maturidade da Capacitação para Software) foi elaborado pelo SEI - Software Engineering Institute (Instituto de Engenharia de Software) que está sediado na CMU - Carnegie Mellon University em Pittsburgh, Pennsylvania, Estados Unidos.

O SEI é um centro de pesquisa e desenvolvimento criado em 1984 pelo Departamento de Defesa dos Estados Unidos (DoD - Department of Defense) e é patrocinado pelo OUSD (A&T) (Office of the Under Secretary of Defense for Acquisition and Technology) que tem como objetivo aprimorar a prática de Engenharia de Software.

O SW-CMM, ou simplesmente CMM como é mais conhecido, foi elaborado para atender ao Departamento de Defesa do Governo dos Estados Unidos, que é um grande consumidor de software e precisava de um modelo formal que permitisse selecionar os seus fornecedores de software de forma mais adequada. Em agosto de 1991 foi lançada a versão 1.0 do CMM que no período de 1991 a 1992 foi amplamente usada e revisada pela comunidade de software. Em meados do ano de 1993 foi lançada a versão 1.1 do CMM refletindo o resultado das recomendações de um *workshop* sobre a versão 1.0 e do retorno da comunidade de software.

Embora não seja uma norma emitida por uma instituição internacional (como a ISO - International Standard Organization ou o IEEE - Instituto de Engenharia Elétrica e Eletrônica), o CMM tornou-se um dos modelos mais prestigiados. Ele tornou-se referência para o modelo Software Process Improvement and Capability Determination (SPICE) que foi elaborado pela ISO em conjunto com a IEC - International Electrotechnical Commission.

O CMM visa avaliar e melhorar a capacitação das empresas produtoras de software em relação aos seus processos de desenvolvimento e manutenção de software. Sendo assim, o CMM classifica as organizações em níveis de maturidade.

Esta estrutura de estágios de maturidade está baseada nos princípios de qualidade de produto. Além disso, o CMM consiste na aplicação dos conceitos de gerência de processos do TQM - Total Quality Management (Gerenciamento da Qualidade Total) no contexto do processo de desenvolvimento e manutenção de software.

É importante ressaltar que o CMM tem uma intensa e extensa estrutura conceitual para guiar as organizações que desejam obter um melhoramento contínuo e disciplinado da gerência e do processo de desenvolvimento e manutenção de software. A definição de uma estratégia de melhoria arquitetada com base em uma estrutura de maturidade do processo de software orienta quanto ao caminho a ser seguido para se obter uma contínua melhoria do processo. Esta estrutura permite perceber os avanços e identificar as deficiências na organização.

No entanto, o CMM não estabelece normas e ações específicas a serem implantadas, apenas determina as boas práticas que devem ser adotadas para se alcançar um determinado nível de maturidade. Cabe à organização desenvolver a sua sistemática de implantação e uso das práticas prescritas no CMM de acordo com as suas características e conforme o tipo de software que desenvolve. Ademais, o CMM não aborda a gestão de pessoas e nem a aquisição de tecnologia, embora também sejam questões importantes para se ter um projeto bem sucedido. E, sobretudo, não objetiva fornecer soluções rápidas para projetos em dificuldades.

Porém, antes de iniciarmos a apresentação de como o CMM é estruturado, a fim de proporcionar o seu melhor entendimento, salientamos que alguns conceitos fundamentais associados à estrutura de maturidade do processo de software precisam ser melhor esclarecidos. Os conceitos são:

- **Capacitação do Processo** descreve o conjunto de resultados esperados que podem ser alcançados com a aplicação de um processo de software. Ela fornece um meio de se presumir os resultados mais prováveis de serem obtidos no próximo projeto de desenvolvimento de software a ser empreendido pela organização (Paulk, 1993).
- **Desempenho do processo de software** representa os resultados reais alcançados seguindo-se o processo de software. Assim, o

desempenho do processo de software concentra-se nos resultados alcançados, diferente da capacitação do processo de software que enfoca os resultados esperados. Com base nos atributos de um projeto específico e no contexto no qual ele é conduzido, o desempenho real de um projeto pode não refletir a capacitação do processo total da organização. Isto porque o ambiente pode limitar a capacitação do projeto. Por exemplo, alterações radicais nas características do software a ser desenvolvido ou na tecnologia utilizada podem colocar a equipe envolvida no projeto na situação de aprendizes. Com isso a capacitação e o desempenho deste projeto podem diminuir em relação à capacitação do processo total da organização (Paulk, 1993).

- **Maturidade do Processo** significa a extensão para a qual um processo específico é explicitamente definido, gerenciado, medido, controlado e efetivado. A maturidade representa o potencial de crescimento de capacitação, indica a riqueza do processo de software da organização e a consistência com que ele é aplicado em todos os seus projetos (Paulk, 1993).

Com base no conceito de maturidade do processo, o CMM caracteriza as organizações desenvolvedoras de software em imaturas e maduras.

Em uma organização imatura, durante a execução de um determinado projeto, os processos de software geralmente são improvisados por experientes desenvolvedores e gerentes de software. Porém, embora o processo de software seja especificado, ele não é obrigatório e, por conseguinte, não é rigorosamente seguido. Outra característica das organizações imaturas é que os cronogramas e os orçamentos são rotineiramente excedidos. Isto ocorre porque os cronogramas não são elaborados com base em estimativas realistas, conseqüentemente a funcionalidade e a qualidade do software são freqüentemente comprometidas para que o cronograma seja cumprido. Dessa forma, não se cria uma base informacional objetiva para a avaliação da qualidade do software e nem para a resolução de problemas associados ao produto e ao processo de software. Sendo assim, é difícil antever a qualidade do produto de software.

Por outro lado, uma organização de software madura possui uma larga habilidade para gerenciar o desenvolvimento de software e manter os seus processos. No caso do processo de software, este é definido e suas regras e responsabilidades são claras em toda a organização, pois o processo é disseminado de forma precisa para a equipe já existente e para os novos funcionários. Além disso, os processos definidos são revisados sempre que necessário e as melhorias são implementadas por meio de testes-piloto e/ou análise de custo-benefício.

Uma organização de software madura também se caracteriza por ter gerentes que monitoram a qualidade dos produtos de software e a satisfação do cliente. Esta tarefa proporciona a criação de uma base informacional objetiva e quantitativa. As informações contidas nessa base podem ser utilizadas como referência para a avaliação da qualidade do produto de software, assim como para análise dos problemas relacionados a ele e ao seu processo de construção. Uma vez que os cronogramas e os orçamentos são elaborados com base em um histórico de desempenhos, eles são realistas, e conseqüentemente os resultados esperados para custo, cronograma, funcionalidade e qualidade do produto são quase sempre alcançados.

Conforme se pode observar, a discrepância da qualidade da gestão e do processo de construção e manutenção de software entre as organizações imaturas e maduras é considerável. Dessa forma, seria praticamente impossível, em um só momento, evoluir do estado de imaturidade para o patamar máximo de maturidade.

Portanto, o CMM é estruturado em cinco níveis (um a cinco) de maturidade, conforme FIG. 11.

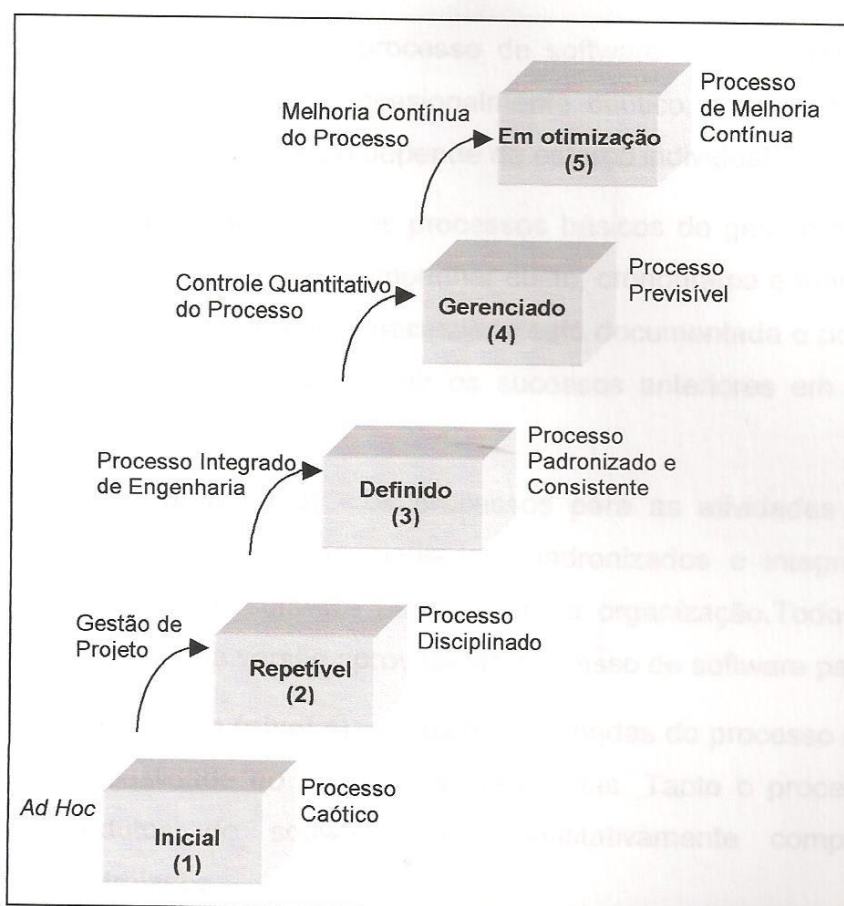


FIGURA 11 – Os cinco níveis de maturidade do processo de software

Fonte – Paulk, 1993, p.8.

Nesta estrutura é descrito um caminho evolutivo para o qual é estabelecida uma escala crescente para medir a maturidade e avaliar a capacitação do processo de software, assim como ajudar a organização a priorizar seus esforços para melhorias.

A organização do CMM mostrada na FIG. 11 permite que as ações de melhoria para o crescimento da maturidade do processo de software sejam implementadas gradativamente. As setas rotuladas na FIG. 11 indicam o tipo de capacitação de processo que está sendo institucionalizado pela organização em

cada etapa da estrutura de maturidade. Descreve-se a seguir os níveis de maturidade destacando suas principais características (Paulk, 1993):

- **Inicial (nível 1)** - o processo de software é caracterizado como *ad hoc*²⁹ e até mesmo ocasionalmente caótico. Poucos processos são definidos e o sucesso depende de esforço individual.
- **Repetível (nível 2)** - os processos básicos de gestão de projeto são estabelecidos para acompanhar custo, cronograma e funcionalidade. A disciplina de processo necessária está documentada e pode ser usada para que se possa repetir os sucessos anteriores em projetos com aplicações similares.
- **Definido (nível 3)** - os processos para as atividades de gestão e engenharia são documentados, padronizados e integrados em um processo de software padrão para a organização. Todos os projetos utilizam uma versão aprovada do processo de software padrão.
- **Gerenciado (nível 4)** - medidas detalhadas do processo de software e da qualidade do produto são realizadas. Tanto o processo como os produtos de software são quantitativamente compreendidos e controlados.
- **Em otimização (nível 5)** – tem como objetivo prevenir a ocorrência de falhas. Aborda a melhoria contínua do processo, através da identificação das oportunidades de melhoria e fortalecimento do processo de maneira pró-ativa.

²⁹ De acordo com (Houaiss, 2001), significa feito exclusivamente para explicar o fenômeno que descreve e que não serve para outros casos, não dando margem a qualquer generalização (diz-se de regra, argumento, definição etc.)

De acordo com a FIG. 12, cada nível, exceto o número um, é composto por um conjunto de Áreas-Chaves de Processo (ACP ou KPA – Key Process Area).

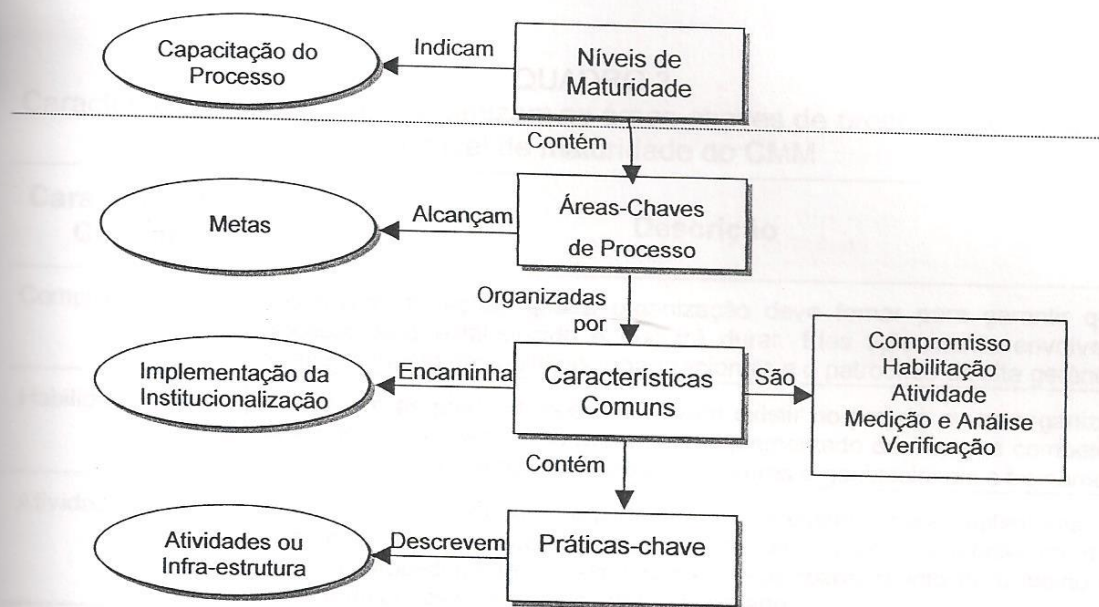


FIGURA 12 – Esquema da estrutura do CMM

Fonte – FIORINI, 1998, p.18.

Área-Chave de Processo é aquela que é determinante para a capacitação do processo, por conseguinte, é imprescindível para a obtenção de um nível de maturidade do CMM. Embora outros assuntos, como por exemplo, a gestão de pessoas e a aquisição de tecnologia, afetem o desempenho de um processo, as áreas-chaves de processo descritas no CMM foram selecionadas devido à sua eficácia e eficiência na melhoria da capacitação do processo de uma organização.

As áreas-chaves conduzem a organização ao alcance de metas de melhoria do processo para um nível de maturidade. Sendo que, para atender uma ACP, cada uma das metas para aquela área-chave de processo deve ser satisfeita. As metas resumem as práticas-chave, bem como significam o escopo, os limites e a intenção de cada área-chave de processo. As metas também podem ser utilizadas para determinar se uma organização ou projeto efetivamente implementou a área-chave de processo.

A FIG. 12 mostra também que cada ACP está organizada em cinco seções denominadas coletivamente de Características Comuns que são, na verdade, atributos que indicam se a implementação e a institucionalização de uma área-chave de processo é eficaz, repetível e duradoura. As cinco características comuns são descritas no QUADRO 3.

QUADRO 3
Características comuns que organizam as áreas-chaves de processo definidas para um nível de maturidade do CMM

Característica Comum	Descrição
Compromissos	Descrevem as ações que a organização deve tomar para garantir que o processo seja estabelecido e que irá durar. Eles tipicamente envolvem o estabelecimento das políticas organizacionais e o patrocínio da alta gerência.
Habilidades	Descrevem as precondições que devem existir no projeto ou na organização para que o processo de software seja implementado de maneira competente. Elas tipicamente envolvem recursos, estruturas organizacionais e treinamento.
Atividades	Descrevem os papéis e os procedimentos necessários para implementar uma área-chave de processo. Elas tipicamente envolvem o estabelecimento de planos e procedimentos, a execução e o acompanhamento do trabalho e da <i>tomada de ações corretivas quando necessário</i> .
Medições e Análises	Descrevem a necessidade de medir o processo e analisar as medições. Elas tipicamente incluem exemplos das medições que poderiam ser feitas para determinar a situação e a eficácia das atividades.
Verificação da Implementação	Descreve os passos para garantir que as atividades sejam executadas em conformidade com o processo que foi estabelecido. Ela tipicamente engloba revisões e auditorias pela gerência e pela garantia da qualidade de software.

FONTE – Paulk, 1993, p.60.

As práticas definidas no item característica comum como *Atividades* descrevem o que deve ser implementado para estabelecer a capacitação do processo. As práticas estabelecidas nos demais itens, em geral, compõem a infraestrutura necessária para que a organização possa institucionalizar o que está definido em *Atividades*.

De acordo com Paulk (1993), é importante ressaltar que as práticas específicas a serem executadas em cada área-chave de processo evoluem de acordo com a obtenção, por parte da organização, de níveis mais elevados de maturidade de processo. Por exemplo, a maioria das capacitações de estimativas de projeto descritas na área-chave de processo Planejamento de Projeto de Software, do Nível 2, deve evoluir para tratar os dados adicionais de projeto disponíveis nos

Níveis 3, 4 e 5. Neste contexto, a ACP Gestão Integrada de Projeto, no Nível 3, é uma evolução das ACPs Planejamento de Projeto de Software e Acompanhamento e Supervisão de Projeto de Software, do Nível 2, à medida que o projeto é gerido usando um processo de software definido.

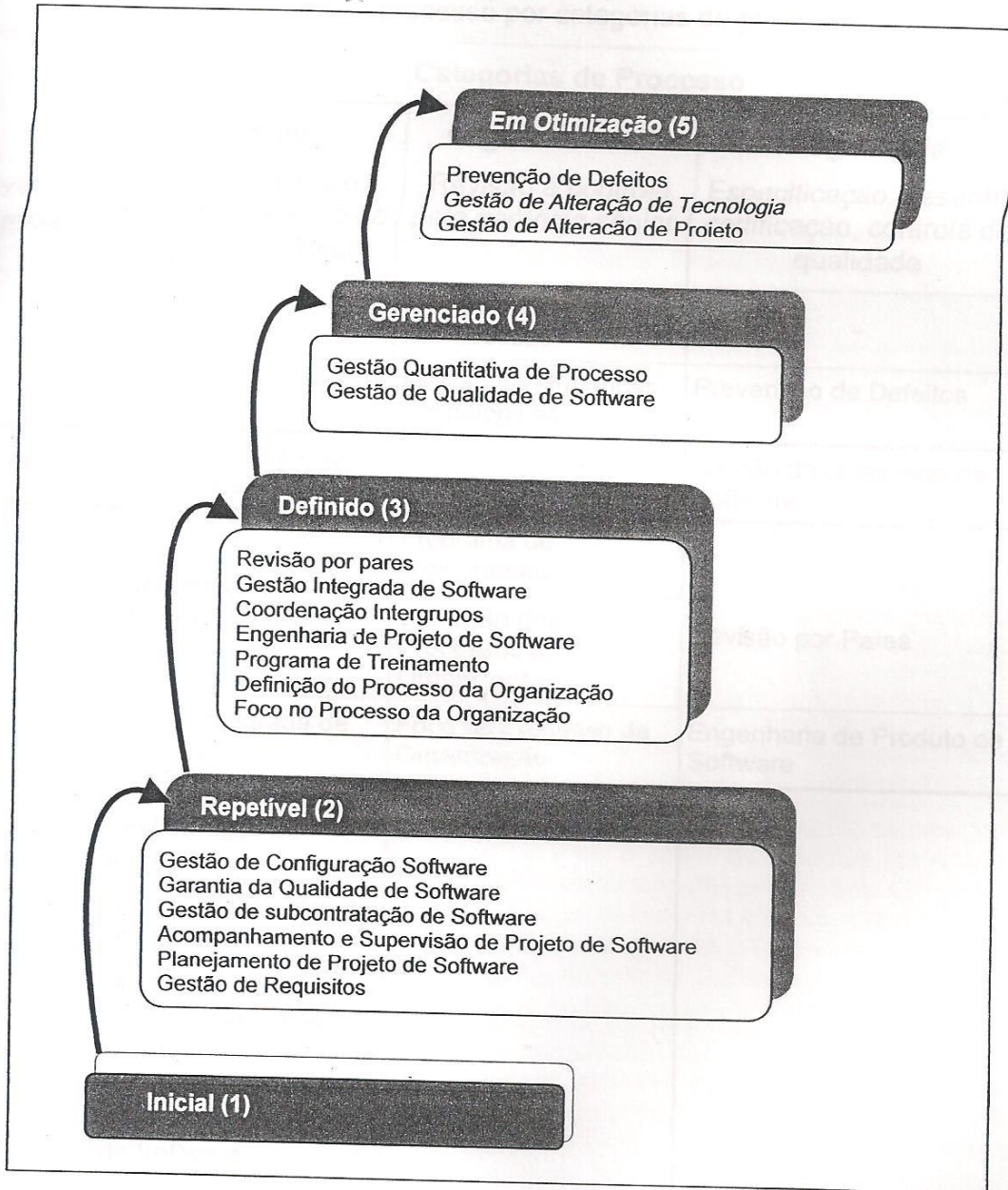


FIGURA 13 – As áreas-chaves de processo por nível de maturidade

Fonte – Paulk, 1993, p.36.

Neste contexto, é possível classificar as ACP's em três categorias de processos, conforme o QUADRO 4.

QUADRO 4
Áreas-chaves de processo por categorias de processo

Nível de Maturidade	Categorias de Processo		
	Gestão Planejamento e acompanhamento de projeto de software	Organizacional Revisão e controle pela gerência sênior	Engenharia Especificação, desenho, codificação, controle de qualidade
5	-	Gestão de Mudanças no Processo	-
		Gestão de Mudanças Tecnológicas	Prevenção de Defeitos
4	Gestão Quantitativa dos Processos	-	Gestão da Qualidade de Software
3	Coordenação Intergrupos	Programa de Treinamento	Revisão por Pares
		Definição dos Processos da Organização	
	Gestão integrada de Software	Foco no Processo da Organização	Engenharia de Produto de Software
2	Gestão de Configuração de Software	-	-
	Garantia da Qualidade de Software		
	Gestão da Subcontratação de Software		
	Acompanhamento e Supervisão do Projeto de Software		
	Planejamento do Projeto de Software		
	Gestão de Requisitos		

Fonte – Adaptado de (Neto, 2003).

QUADRO 5
Áreas-chaves de processo do CMM nível 2

Área-Chave de Processo	Objetivo
Gestão de Requisitos	Estabelecer um entendimento comum, entre o contratante e a equipe do projeto de software, sobre os requisitos de software ³⁰ que deverão ser atendidos. Envolve o estabelecimento de compromissos, tanto técnicos, quanto relativos a datas de entrega ou outras restrições, de modo que a equipe do projeto possa garantir a documentação, planos e atividades consistentes com os requisitos.
Planejamento do Projeto de Software	Estabelecer planos razoáveis para a execução das atividades de engenharia de software, bem como para a gestão do projeto de software. Esta área envolve a elaboração de estimativa para o trabalho a ser executado, a definição dos compromissos necessários e do plano de execução do trabalho. Esses itens constituem a base necessária para gerir o projeto de software, sem eles, uma gestão efetiva de projeto não pode ser implementada.
Acompanhamento e Supervisão do Projeto de Software	Prover visibilidade adequada sobre o progresso real do projeto, de forma que a gerência possa executar ações corretivas quando o desempenho do projeto de software sofre um desvio significativo com relação ao planejado. Ações corretivas podem incluir revisões do plano de desenvolvimento para refletir as reais realizações, replanejamento do trabalho restante ou realização de ações para melhorar o desempenho. Esta área envolve o acompanhamento e revisão das realizações e resultados em comparação as estimativas, compromissos e planos documentados; envolve o ajuste dos planos de acordo com base nos resultados da análise do estimado <i>versus</i> executado.
Gestão da Subcontratação de Software	Selecionar subcontratados qualificados e gerenciá-los de modo efetivo. Ela combina os assuntos relativos as demais áreas-chaves de processo que são aplicadas apropriadamente aos subcontratados.
Garantia da Qualidade de Software	Fornecer uma gestão com visibilidade apropriada sobre os processos em uso pelo projeto de software e sobre os produtos construídos. Esta área envolve a revisão e a auditoria dos produtos de software, as atividades para verificar se os procedimentos e padrões são aplicados e o fornecimento dos resultados ao gerente do projeto e outras áreas envolvidas. A Garantia de Qualidade de Software é parte integrante da maioria dos processos de gestão e de engenharia de software.
Gestão da Configuração de Software	Estabelecer e manter a integridade dos produtos do projeto de software ao longo de todo o ciclo de vida do software. Esta área envolve a identificação da configuração do software em determinadas fases durante o seu desenvolvimento (p.e. produtos intermediários de software selecionados e suas respectivas descrições), o controle sistemático das mudanças na configuração e a manutenção da integridade e rastreabilidade da configuração. A Gestão de Configuração de Software é parte integrante da maioria dos processos de gestão e de engenharia de software.

Fonte – Paulk, 1993, p.37.

³⁰ Característica intrínseca do software necessária para solucionar um problema do usuário ou atender um contrato, padrões, especificações ou outra documentação formal. Definem os critérios de aceitação do software.

A análise dos QUADROS 4 e 5 aponta para quatro das ACP's do nível 2 do CMM, a saber:

- Planejamento de Projeto de Software;
- Acompanhamento e Supervisão de Projeto de Software;
- Garantia da Qualidade de Software;
- Gestão da Configuração de Software.

Elas referem-se às atividades guarda-chuva do processo de software descritas por Pressman (ver FIG. 9 e QUADRO 2). As *práticas referentes à Gestão de Subcontratação* e a *Gestão de Requisitos* foram acrescentadas ao grupo de atividades guarda-chuva. A gestão de requisitos infere diretamente sobre as atividades de engenharia de software, uma vez que as atividades descritas nesta ACP estão relacionadas com as atividades de levantamento, análise e documentação das necessidades dos usuários do software a ser desenvolvido, daí a sua importância no nível 2 do CMM.

Observando ainda os QUADRO 4 e 5, fica claro que as áreas-chaves de processo do nível 2 do CMM (Repetível) tem como foco exclusivamente os processos de gestão. Isto enfatiza a importância que o CMM confere a duas etapas:

- Primeiro, obtém-se a capacidade de planejar, executar e controlar as atividades dos projetos;
- Segundo, estabelece-se um processo de engenharia (metodologias e práticas de software) para a organização que desenvolve software. Porém, nada impede que uma organização no nível 2 adote boas práticas de engenharia.

Para Humphrey (*apud* Paulk, 1993), a razão fundamental para se focar em gestão de projeto, antes de se implementar o processo de engenharia, é o fato de que sem a disciplina gerencial, o processo de engenharia é prejudicado pelas pressões de prazo e de custo.

Os controles de gestão estabelecidos no nível 2 possibilitam a visibilidade interna do projeto em momentos definidos. Neste contexto, as atividades técnicas³¹ de engenharia de software podem ser visualizadas como uma sucessão de caixas pretas, permitindo a visibilidade da gestão nos pontos de transição entre os fluxos de atividades definidos em cada caixa (marcos do projeto). Sendo assim, mesmo que a gerência não conheça detalhes do que está acontecendo dentro de cada caixa, os produtos e os pontos de verificação dos processos são identificados e conhecidos. Tais produtos e pontos de verificação são instrumentos que propiciam à gerência verificar se o processo está funcionando e, assim, reagir aos problemas quando os mesmos ocorrem.

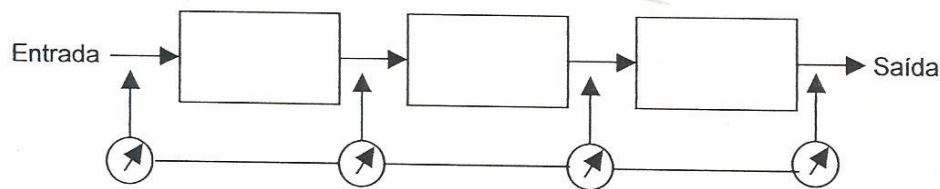


FIGURA 14 – Visibilidade da gerência dentro dos processos de software no nível 2 do CMM
Fonte – Paulk, 1993, p.27.

Resumindo, a capacitação do processo de software obtida pelas organizações que alcançam o nível 2 de maturidade do CMM pode ser intitulada de disciplinada. Isto porque a organização institucionalizou as atividades básicas de gestão de projeto, as quais proporcionam o controle efetivo dos processos.

Esta capacitação permite o registro e o uso das informações pertinentes à memória do projeto, do processo e do produto de software. Isso possibilita que os sucessos mais recentes possam ser repetidos e as falhas evitadas em projetos subseqüentes que tenham aplicações similares. Além do mais, contribui para a aprendizagem e o aperfeiçoamento da atividade de desenvolvimento e manutenção de software.

³¹ Consistem nas *atividades de estrutura* descritas por Pressman, conforme FIG. 9 e QUADRO 3

4. METODOLOGIA

4.1. Estratégia de Pesquisa

A realização da presente pesquisa envolveu um estudo descritivo com estratégias qualitativas que, segundo Lüdke & André (1996), desenvolvido em uma situação natural, tem um plano aberto e flexível, assim como focaliza a realidade de forma complexa e contextualizada.

A técnica escolhida foi a de estudo de caso que terá como unidade de análise o SERPRO – Serviço Federal de Processamento de Dados que é uma empresa pública vinculada ao Ministério da Fazenda fundada em 1964. O SERPRO tem por objetivo a comercialização e prestação de serviços de informática.

Como já foi descrito, este estudo tem como objetivo verificar as práticas e dificuldades no processo de gerenciamento da informação por parte dos gerentes e técnicos durante o desenvolvimento e/ou manutenção de um software. O SERPRO tem como referencial o Modelo de Maturidade de Capacitação para Software – CMM, daí a sua escolha para este estudo.

Para alcançar o objetivo principal desta pesquisa, os seguintes objetivos específicos foram determinados:

- Analisar a relação entre as práticas-chaves especificadas no nível 2 do CMM e as atividades que constituem a gestão da informação.
- Identificar as práticas e dificuldades na aplicação das diretrizes especificadas no nível 2 do CMM que estejam relacionadas com a gestão da informação.

Com o intuito de orientar as atividades de coleta e análise dos dados, selecionou-se os dois grupos de categorias descritos no QUADRO 6.

QUADRO 6
Grupos de categorias de análise

Grupos de Categorias	Categorias
ACP CMM N2 Áreas-Chaves de Processo do CMM Nível 2	<ul style="list-style-type: none"> ▪ Planejamento do Projeto de Software ▪ Acompanhamento e Supervisão do Projeto de Software ▪ Gestão de Requisitos ▪ Gestão de Configuração de Software ▪ Garantia da Qualidade de Software ▪ Gestão de Subcontratação
PGI Processos de Gestão da Informação*	<ul style="list-style-type: none"> ▪ Necessidades de Informação ▪ Aquisição de Informação ▪ Organização e Armazenamento da Informação ▪ Produtos/Serviços de Informação ▪ Distribuição da Informação ▪ Uso da Informação

(*) Ciclo de gerenciamento da Informação Choo (1998a).

4.2. Delimitação do Estudo

Ainda conforme Lüdke & André (1986), em um estudo de caso é necessário que o caso seja sempre bem delimitado, devendo ter seus contornos claramente definidos no desenrolar do estudo. Portanto, nesta seção delimitaremos o universo do estudo que será desenvolvido nesta pesquisa.

Segundo Goode & Hatt (1968), citado por Lüdke & André (1986), o caso se destaca por se constituir em uma unidade dentro de um sistema mais amplo. O interesse, portanto, incide naquilo que ele tem de único, de particular, mesmo que posteriormente venham ficar evidentes certas semelhanças com outros casos ou situações.

Como já foi mencionado anteriormente, o SERPRO – Serviço Federal de Processamento de Dados tem como atividade fim a prestação de serviços em

tecnologia da informação com vistas a modernizar e dar agilidade aos setores estratégicos da Administração Pública Federal. Sua sede está localizada em Brasília, mas possui projeções regionais nas dez regiões fiscais da SRF - Secretaria da Receita Federal (Brasília, Belém, Fortaleza, Recife, Salvador, Belo Horizonte, Rio de Janeiro, São Paulo, Curitiba e Porto Alegre).

Em 1999, foi criado o Processo Corporativo Conhecimento e Aprendizagem Organizacional que mais tarde se transformaria no Processo e Política SERPRO de Gestão do Conhecimento e da Aprendizagem Organizacional. Em linhas gerais, a Política evidencia a importância das pessoas na geração e na aplicação de conhecimentos, habilidades e experiências com a finalidade de oferecer aos clientes do SERPRO soluções que atendam as suas expectativas, obtendo-se assim melhor desempenho empresarial.

Dentro desta nova linha de atuação, o SERPRO deu início a um processo de modernização de uma das suas principais atividades, o desenvolvimento de software. Tal processo de modernização do desenvolvimento de software adota como modelo de qualidade o CMM – Capability Maturity Model.

Como um dos resultados desse processo, o SERPRO obteve a certificação dos pólos de desenvolvimento de software da Superintendência de Administração Tributária das projeções regionais de Recife e Salvador. As demais regionais encontram-se no processo de implantação com vistas à certificação até o final do ano de 2004.

Atualmente, o SERPRO é a única empresa pública da América Latina a obter certificação CMM nível 2. Esta certificação é conferida pelo SEI - Software Engineering Institute (Instituto de Engenharia de Software) que está sediado na CMU – Carnegie Mellon University em Pittsburgh, Pennsylvania, Estados Unidos.

O SERPRO atua no segmento das ações estruturadas e integradoras da Administração Pública Federal, que é constituída pelo Ministério do Planejamento, Orçamento e Gestão. Além desses Ministérios, a atuação do SERPRO estende-se a outros órgãos governamentais que venham a constituir ações correlatas e que

demandem serviços característicos da Empresa. No entanto, o seu principal cliente é o Ministério da Fazenda com suas secretarias e demais órgãos, correspondendo a 85,2% do volume de negócios da Empresa.

Lüdke & André (1986), apontam como um dos problemas fundamentais para o investigador de um estudo de caso o acesso à informação necessária para responder às suas questões.

Dessa forma, optou-se por investigar as atividades dos profissionais da Superintendência de Negócios Administração Tributária – SUNAT do SERPRO, devido à representatividade significativa do segmento das finanças públicas no universo dos negócios da empresa.

A SUNAT – Superintendência de Negócios Administração Tributária é uma das oito unidades de negócio da empresa. Ela é composta por uma unidade central, situada em Brasília e por 10 pólos de desenvolvimento de sistemas, distribuídos nas 10 regiões fiscais da SRF; um Departamento de Segurança e Engenharia de Software e uma Coordenação Nacional de Desenvolvimento do Sistema de Informações Econômico-Financeiras - SIEF.

Na unidade central estão localizados os órgãos de abrangência nacional que trabalham em forma de coordenações de negócio e controladoria (área administrativa). Nesses órgãos são monitorados os aspectos comerciais, de produção dos serviços, relativos aos resultados, à qualidade e ao planejamento. Nos Pólos de Desenvolvimento existem 41 equipes, com cerca de 600 desenvolvedores, com dedicação exclusiva aos serviços da Secretaria da Receita Federal. Tais equipes têm a atribuição de desenvolver novas soluções de software e dar manutenção às já existentes. (CARVALHO, 2002).

A SUNAT atende a Secretaria da Receita Federal, desenvolvendo tecnologia, ferramentas e principalmente softwares pertinentes à Arrecadação, Tributação, Fiscalização e demais funções do cliente, de forma integrada. Tais produtos têm como foco facilitar o processo de arrecadação, gestão e a tomada de

decisões da SRF, de forma a facilitar e tornar mais amigável o relacionamento fisco-contribuinte.

Em cada projeção regional, a SUNAT atua em um conjunto de produtos e serviços de acordo com as habilidades e as especialidades dos seus profissionais. Dentre os diversos serviços prestados podemos citar alguns sistemas, tais como IRPF, Receitanet, Site da Secretaria da Receita Federal, Declaração de Isentos, Certificação Digital, CPMF e outros já inseridos nas esferas econômicas e sociais da vida do Governo, das empresas e dos cidadãos brasileiros.

4.3. População do Estudo

A população específica da pesquisa é composta por empregados do SERPRO com função de gerência de nível médio, pois grande parte das práticas preconizadas no CMM nível 2 abrange as atividades gerenciais desempenhadas por este grupo de gerentes. Os gerentes que compõem a população selecionada estão lotados na Superintendência de Negócios Administração Tributária – SUNAT localizadas em Belém, Belo Horizonte, Curitiba, Fortaleza, Recife e Salvador.

Das 09 projeções regionais existentes, as 06 acima citadas foram selecionadas considerando o quantitativo do corpo gerencial e das equipes, a diversidade das características dos projetos de software executados por cada uma delas e a diferença de estágio de implantação das práticas do nível 2 do CMM em que cada uma delas encontrava-se. Essas características proporcionaram a diversidade da amostra necessária para se obter uma melhor análise dos dados.

Com relação estágio de implantação das práticas do nível 2 do CMM, segundo os relatórios de resultados de garantia da qualidade de software do SERPRO, as regionais de Fortaleza, Recife e Salvador utilizam plenamente o PSDS – Processo Serpro de Desenvolvimento de Soluções (processo de software padrão da organização fundamentado no CMM nível 2), as regionais de Belém e Curitiba encontram-se em adiantado processo de melhoria em relação ao uso do PSDS e a regional de Belo Horizonte necessita intensificar o uso do PSDS.

Conforme citado anteriormente, durante o período desta pesquisa, a certificação foi conferida ao pólo de desenvolvimento de software da Superintendência de Administração Tributária das projeções regionais de Recife e Salvador. As demais regionais encontram-se em estágios diferentes no processo de implantação do CMM, porém todas com vistas à certificação até o final do ano de 2004.

A população específica do estudo foi dividida em dois grupos. O primeiro é o grupo gerente sênior, denominado na linguagem da SUNAT de 'liderança de 1º nível' e constituído pelos Chefes de Pólo. O segundo que é o grupo líder de projeto, denominado na SUNAT de 'liderança intermediária', é constituído pelos Chefes de Projeto. O objetivo dessa classificação é enriquecer a análise dos dados por meio da identificação de possíveis semelhanças e diferenças na atuação dos grupos.

De acordo com a hierarquia da empresa, esta população de empregados com função gerencial de nível tático e operacional conduz o planejamento, execução e controle dos projetos de software da unidade de negócios SUNAT. Dessa forma, a amostra é representativa em relação às 06 projeções regionais selecionadas e é constituída de 06 gerentes seniores e 18 líderes de projeto, conforme o QUADRO 7.

QUADRO 7
Funcionários com função gerencial por cargo e localização geográfica

Projeções Regionais	Função	
	Gerente Sênior	Líder de Projeto
Regional Belém	1	1
Regional Belo Horizonte	1	3
Regional Curitiba	1	3
Regional Fortaleza	1	3
Regional Recife	1	5
Regional Salvador	1	3
	6	18

4.4. Instrumentos de pesquisa

Os instrumentos utilizados na pesquisa para coleta de dados contemplaram a análise documental, observação na projeção regional de Belo Horizonte, e um questionário (ANEXO B) que foi distribuído por meio do correio eletrônico.

A análise documental referente ao CMM nível 2 e ao processo de software padrão da empresa, PSDS – Processo SERPRO de Desenvolvimento de Soluções, contribuiu no sentido de apontar relações entre a gestão da informação e o processo de software fundamentado no nível 2 do CMM. A identificação dessas relações foi o primeiro resultado do estudo realizado. Tais relações foram detalhadas no capítulo 5 deste trabalho.

Com relação à origem da documentação analisada, destaca-se que aquela referente ao processo de software padrão do SERPRO foi levantada no *site* do PSDS que está disponível na *intranet* da empresa. Neste *site* estão armazenadas todas as informações referentes ao PSDS, assim como artigos, normas, padrões e políticas organizacionais correiatas. No que diz respeito ao CMM nível 2, utilizou-se a documentação oficial que se encontra disponível no *site*³² do SEI – Software Engineering Institute of Carnegie Mellon University.

Para identificar as práticas e dificuldades na aplicação das diretrizes especificadas no nível 2 do CMM, utilizou-se o questionário e os relatórios mensais de análise e acompanhamento dos resultados da GQS - Garantia da Qualidade de Software. Tais relatórios são elaborados pelos grupos de GQS corporativo e das respectivas regionais e estão disponíveis no REVISA, ferramenta institucional e corporativa de suporte à garantia da qualidade de software.

O questionário foi remetido por correio eletrônico da *intranet* da empresa. O envio foi realizado no dia 09 de fevereiro de 2004 com prazo de retorno para o dia 05 de março de 2004, porém em virtude da demora no atendimento, foi necessário

³² <http://www.sei.cmu.edu>

estender o prazo até o dia 31 de março de 2004. É importante ressaltar que a última parte do questionário referente à área-chave de processo gestão de subcontratação não foi preenchida por nenhum dos participantes da pesquisa. Os respondentes do questionário justificaram o ocorrido afirmando que, apesar do processo de software padrão definido pela organização abordar as diretrizes desta ACP, eles não praticavam a subcontratação de software e, por isso, não teriam conhecimento de causa para responder as questões contidas na parte 8 do questionário.

Dessa forma, nesta pesquisa não foi possível levantar dados referentes às práticas e dificuldades na aplicação das diretrizes especificadas para a área-chave de processo gestão de subcontratação.

É importante salientar também que o uso do correio eletrônico permitiu a obtenção de informações complementares referentes às dúvidas e à análise documental. Além disso, com as informações e dados coletados foi possível constituir um banco de dados para facilitar as análises quantitativas e qualitativas da pesquisa. O banco de dados foi estruturado na forma de planilhas interligadas. Tais planilhas foram elaboradas na ferramenta de processamento de planilhas eletrônicas Microsoft Excel.

4.5. Procedimentos de análise dos dados

A análise dos dados qualitativos levantados na análise documental referente às atividades de gestão da informação realizou-se da seguinte maneira.

Inicialmente, procedeu-se a análise documental sobre a documentação do nível 2 do CMM, considerando as categorias apontadas no item 4.1. Posteriormente, essas classificações foram revisadas e os dados reagrupados de acordo com cada área-chave de processo do nível 2 do CMM que são:

- Planejamento de Projeto de Software;
- Acompanhamento e Supervisão de Projeto de Software;
- Gestão de Requisitos;

- Gestão de Configuração de Software;
- Garantia da Qualidade de Software;
- Gestão de Subcontratação de Software.

Por fim, foi apresentada no capítulo 5, na forma de texto descritivo e também de um quadro resumo (QUADRO 6), a análise da relação entre a gestão da informação e as práticas preconizadas em cada área-chave de processo do CMM nível 2.

Os dados quantitativos coletados por questionários foram examinados considerando as categorias apontadas no item 4.1. A análise dos dados coletados é tratada no capítulo 6.

5. A INFORMAÇÃO E A CRIAÇÃO DE CONHECIMENTO NO PROCESSO DE SOFTWARE FUNDAMENTADO NO CMM NÍVEL 2

Este capítulo consiste no primeiro resultado deste estudo. Ele tem como objetivo estabelecer relações entre a informação e o processo de software fundamentado no nível 2 do CMM. As relações são estabelecidas sob a perspectiva de evidenciar a inserção da informação no contexto da gestão de requisitos e das atividades guarda-chuva do processo de desenvolvimento de software. Para tanto, são considerados os conceitos de gerenciamento da informação e de conversão do conhecimento postulados por Choo (1998) e Nonaka & Takeuchi (1997), respectivamente. Sua apresentação antes dos resultados da análise dos dados tem a intenção de clarear o propósito do estudo.

Sendo assim, é desenvolvida uma análise, com base nas categorias selecionadas, da relação dos processos de gerenciamento da informação e construção de conhecimento que ocorrem na gestão de requisitos e em cada uma das atividades guarda-chuva definidas por Pressman (2002) e que se encontram descritas no CMM nível 2.

Inicialmente, são abordadas as diretrizes comuns a todas as áreas-chaves do nível 2 do CMM, em seguida a apresentação da análise é estruturada conforme segue:

1. Gestão do Projeto de Software
 - Planejamento do Projeto de Software;
 - Acompanhamento e Supervisão do Projeto de Software;
2. Gestão de Requisitos;
3. Garantia da Qualidade de Software;
4. Gestão de Configuração de Software;
5. Gestão da Subcontratação de Software.

5.1. Diretrizes comuns

Conforme descrito no capítulo 2, na perspectiva de Ponjuan Dante, a gestão da informação inclui em dimensões estratégicas e operacionais, os mecanismos de obtenção e utilização de recursos humanos, tecnológicos, financeiros, materiais e físicos para o gerenciamento da informação. A partir disto, ela é disponibilizada como insumo útil e estratégico para indivíduos, grupos e organizações.

Na mesma linha, Choo (1998a) afirma que transformar a informação em aprendizado e *insight* muitas vezes equivale a administrar de forma aglutinadora os recursos, ferramentas e tecnologias, políticas e padrões de informação que constituem a infra-estrutura tecnológica da administração da informação.

A importância da infra-estrutura tecnológica definida por Choo e Ponjuan Dante pode ser evidenciada em todas as áreas-chaves de processo do nível 2 do CMM, quando nestas são definidos:

- Compromissos que tipicamente envolvem o estabelecimento de políticas organizacionais e o patrocínio da alta gerência para cada área-chave de processo;
- Habilidades que tipicamente envolvem recursos, orçamentos, treinamento e estruturas organizacionais necessárias para implementação das atividades de cada área-chave de processo;

Conforme visto no capítulo 3, Choo afirma que as organizações devem usufruir da vantagem do conhecimento explícito para maximizar a eficiência e a transferência de aprendizado. Visto assim, o CMM é uma metodologia que orienta as organizações de software a estabelecer procedimentos documentados para a realização das atividades descritas nas ACPs e, também, a explicitar o conhecimento tácito adquirido por meio da documentação do software.

O ambiente das organizações de software é dinâmico e instável, neste cenário, McGee & Prusask (1994) consideram que o aprendizado explícito e

sistemático deve ser a pedra fundamental das estratégias para manutenção e adaptação das organizações a seus ambientes.

Sendo assim, verifica-se uma conformidade das idéias de McGee & Prusak e o CMM, em especial o nível 2. Isto porque o nível 2, também denominado de repetível, estimula o aprendizado explícito e sistemático, pois determina a necessidade de registrar as lições aprendidas, de sucesso e insucesso, durante a execução do projeto de software.

5.2. Gestão do projeto de software

De um modo geral, o processo de desenvolvimento de um software inicia quando o cliente³³ solicita um serviço a uma organização de software. Nesta solicitação estão definidos os primeiros requisitos funcionais³⁴ e não funcionais³⁵ do software. Com base nessas informações iniciais, uma equipe é designada para planejar e executar o projeto que implementará o produto de software solicitado.

A gestão de projeto de software, segundo Pressman (2002) e conforme detalhado no capítulo 3, é uma atividade guarda-chuva dentro da engenharia de software. Ela inicia antes de qualquer atividade técnica e continua ao longo da definição, desenvolvimento e entrega do produto de software. O espectro de gestão de um projeto de software tem quatro focos: pessoas, produto, processo e projeto.

As pessoas devem ser organizadas em equipes efetivas e motivadas para gerar produtos de software de alta qualidade e coordenadas para alcançar comunicação efetiva.

Com relação ao produto final (o software), os requisitos devem ser definidos pelo cliente ao analista de requisitos, classificados, detalhados e disponibilizados para serem implementados pela equipe de desenvolvimento.

³³ Patrocinador do projeto de desenvolvimento do software, podendo ser ou não usuário do software.

³⁴ Descrição das diversas operações que clientes e usuários esperam que sejam realizadas pelo software.

³⁵ Determinam as características desejáveis do software quanto à usabilidade, confiabilidade, desempenho, suportabilidade, bem como descrevem restrições de projeto e conformidades legais.

Por outro lado, o processo a ser seguido, segundo Pressman (2002), deve ser adaptado ao perfil da equipe e ao problema. Neste momento, uma estrutura comum de processo deve ser selecionada, um ciclo de vida do software mais adequado ao problema é aplicado e um conjunto de tarefas é escolhido para executar o desenvolvimento do software.

Finalmente, o projeto de software deve ser estruturado de forma a guiar as atividades da equipe de software e esta a obter sucesso.

De modo geral, a gestão de projetos de software envolve o planejamento, o acompanhamento, a monitoração e o controle do pessoal, processo e eventos que ocorrem à medida que o software evolui de um conceito preliminar para uma implementação operacional.

A responsabilidade da gestão direta sob a equipe do projeto fica a cargo do gerente/líder de projeto. Por outro lado, cabe ao gerente sênior coordenar a interface entre o negócio e os profissionais.

5.2.1. Planejamento do projeto de software

A gestão de projetos de software inicia com um conjunto de atividades denominado de planejamento de projeto de software. Este, conforme descrito no capítulo 3, visa estabelecer planos razoáveis à correta execução das atividades de gestão e engenharia de software.

O planejamento do projeto de software envolve a elaboração de estimativas para o trabalho a ser executado, a definição dos recursos, dos compromissos e do tempo necessário, assim como a definição de um plano de execução do trabalho. Tais itens constituem a base necessária para gerir o projeto de software, sem eles, a implementação de uma gestão efetiva é dificultada.

Neste contexto, o líder de projeto depara-se com um universo de incertezas e ambigüidades referentes à definição das características-chaves do projeto e do

plano do projeto. Tais características, segundo Boehm (1996), podem ser identificadas por meio das respostas das questões abaixo:

1. Por que o sistema está sendo desenvolvido?
(visa identificar se as razões comerciais justificam os custos, o tempo e a alocação da equipe)
2. O que vai ser feito e quando?
(visa estabelecer um cronograma do projeto pela identificação de tarefas-chaves de projeto e dos prazos que são exigidos pelo cliente)
3. Quem é responsável pela função?
(visa identificar o papel e a responsabilidade de cada membro da equipe do projeto)
4. Onde os responsáveis estão localizados na organização?
(visa identificar todos que direta e indiretamente estão envolvidos no projeto)
5. Como o trabalho será desenvolvido técnica e gerencialmente?
(visa estabelecer, com base no escopo do produto, uma estratégia gerencial e técnica para o projeto)
6. Quanto é necessário de cada recurso?
(a resposta a esta pergunta é obtida pela elaboração das estimativas do projeto)

Essas questões-chaves, de certa forma, auxiliam o líder de projeto a identificar as suas necessidades informacionais. Sendo que as informações necessárias, na sua grande maioria, referem-se a escopo, ao tempo, ao custo, à qualidade, aos recursos humanos, às comunicações, ao risco e às aquisições.

Sob a perspectiva de Choo de que as pessoas são as fontes de informação mais valiosas de uma organização, o líder de projeto deve iniciar o processo de busca de informações consultando os clientes e a equipe do projeto. Durante os

diálogos e reuniões formais e/ou informais entre o líder e suas fontes, é possível identificar o processo de socialização.

Ainda no que se refere ao processo de aquisição de informação, o líder deve também consultar dados de métricas³⁶ de software coletados de projetos anteriores. A especificação de requisitos do software também tem que ser consultada, pois define o escopo do produto a ser desenvolvido. No caso de projetos grandes, uma especificação completa dos requisitos não é disponibilizada a tempo para ser usada como base para elaborar o plano do projeto. Sendo assim, é importante pelo menos ter acesso aos requisitos preliminares que definem o escopo do produto.

Com relação aos processos, descritos acima, de identificar as necessidades informacionais e de aquisição da informação no contexto do planejamento do projeto de software, o nível 2 do CMM preconiza a necessidade de definir ou identificar e documentar:

1. Um acordo de serviço para o projeto de software que, em linhas gerais, abrange:
 - a. O escopo do projeto;
 - b. Os objetivos e metas técnicas;
 - c. A identificação dos clientes e usuários finais;
 - d. Os padrões impostos;
 - e. As responsabilidades atribuídas;
 - f. Os vínculos de custos e prazos, com as respectivas metas;
 - g. As dependências entre o projeto de software e outras organizações;
 - h. Os vínculos e metas relativas aos recursos, ao desenvolvimento e/ou manutenção.
2. Como as responsabilidades pelos produtos e atividades são distribuídas e designadas de modo a permitir seu acompanhamento;

³⁶ Métrica de software relaciona as medidas individuais de alguma forma, como por exemplo, o número médio de erros encontrado por revisão.

3. O ciclo de vida do software;
4. Os produtos de software que serão gerados (código-fonte, documentos, relatórios, planos e outros)
5. Os riscos de software relacionados aos custos, recursos, cronograma e aspectos técnicos do projeto;

As informações obtidas pelo líder de projeto são insumos para a definição das estimativas de escopo do projeto, de tempo, de custo, de qualidade e de recursos humanos. Essas estimativas orientam e fundamentam o trabalho de análise de riscos, de elaboração dos artefatos do plano do projeto, da lista de riscos, cronograma, da proposta comercial, da lista de recursos críticos (pessoal, ferramentas, hardware).

Esses artefatos constituem os produtos de informação resultantes das atividades de planejamento de projeto de software. O nível 2 do CMM preconiza que esses artefatos sejam elaborados conforme um procedimento documentado. Sobre a confiabilidade das informações oferecidas nesses produtos, o CMM estabelece que tais artefatos devem ser revisados e/ou auditados pelo grupo da garantia da qualidade de software.

Com relação aos modos de conversão do conhecimento, a definição das estimativas consiste em um bom exemplo de externalização. Isto porque, neste caso, ocorre a elaboração do conhecimento explícito baseado no conhecimento tácito compartilhado nas reuniões com clientes e equipe de especificação de requisitos. Porém, com relação às estimativas, é possível verificar também a combinação quando os artefatos de especificação de requisitos e métricas de software, que são conhecimentos explícitos no formato de documentos e estatísticas são utilizados como fonte de informação. Também é factível de identificar um outro exemplo de combinação, quando se dá a elaboração do plano do projeto com base em métricas de software, especificação de requisitos, cronograma, listas de riscos, de recursos que também são conhecimento explícito.

Ainda sobre os modos de conversão do conhecimento, é possível identificar que o nível 2 do CMM preconiza atividades que promovem o processo de socialização quando estabelece que:

1. O grupo de engenharia de software deve participar da equipe, em conjunto com os demais grupos envolvidos, no planejamento global do projeto ao longo de toda a vida do projeto.
2. Os compromissos do projeto de software que forem assumidos com indivíduos e grupos externos devem ser revistos com o gerente sênior.

5.2.2. Acompanhamento e supervisão de projeto de software

É importante ressaltar que esta área-chave trata de duas atividades distintas. A primeira refere-se ao cuidado diário e sistemático dispensado pelo líder de projeto à verificação do planejado *versus* o realizado. Assim como a implementação dos ajustes nos planos em decorrência dos desvios detectados. A segunda é a supervisão de projeto de software pelo gerente sênior por meio de observações periódicas despendidas ao projeto.

- **Acompanhamento do projeto de software**

No acompanhamento de projeto, para executar a análise do estimado *versus* realizado, o líder de projeto necessita de informações referentes:

- a. Ao tempo despendido às atividades;
- b. À situação das atividades de engenharia, de garantia da qualidade e gestão de configuração (concluídas, em andamento, percentual de conclusão);
- c. Ao esforço despendido no projeto;
- d. Ao desempenho da equipe;
- e. À utilização dos recursos computacionais;
- f. Aos riscos do projeto;
- g. À criação com qualidade dos produtos de software.

A complexidade do processo de aquisição destas informações varia de acordo com a infra-estrutura de ferramentas de gestão de projetos disponível na organização. Quanto maior a integração das ferramentas, mais fácil será o trabalho do líder de projeto, além disso, a consistência das informações também pode aumentar proporcionalmente.

Neste momento, as principais fontes informacionais utilizadas pelo líder de projeto são os artefatos produzidos na atividade de planejamento de projeto que são:

- a. Plano do projeto;
- b. Lista de riscos;
- c. Cronograma;
- d. Proposta comercial;
- e. Lista de recursos críticos (pessoal, ferramentas, hardware).

Além desses artefatos, conta-se com a planilha de acompanhamento de esforço despendido pela equipe, com os artefatos que registram as mudanças de requisitos e seus impactos e, claro, com as reuniões com a equipe do projeto.

É interessante ressaltar a importância da confiabilidade dos produtos de informação gerados nas atividades de gestão de requisitos, em especial os que registram as mudanças de requisitos e seus impactos e daqueles produtos gerados nas atividades de planejamento de projeto. Isto se deve ao fato de que todo o acompanhamento gira em torno dessas informações.

Ainda com relação aos processos de identificação das necessidades informacionais, aquisição e uso da informação no contexto do acompanhamento do projeto de software, o nível 2 do CMM preconiza que:

1. Para o acompanhar as atividades de software deve ser utilizado um plano de projeto de software documentado;
2. O plano do projeto de software deve ser revisado conforme procedimento documentado;

3. As atividades técnicas de engenharia de software, o tamanho dos artefatos de software (ou tamanho das respectivas modificações), o cronograma, os esforços, os custos, os recursos críticos de computação relativos ao projeto devem ser acompanhados, devendo ser tomadas medidas corretivas sempre que se fizerem necessárias;
4. Os riscos de software associados aos custos, aos recursos, ao cronograma e aos aspectos técnicos do projeto devem ser acompanhados;
5. As revisões formais voltadas para a avaliação das realizações e resultados do projeto de software devem ser conduzidas à medida que são atingidos marcos selecionados do projeto.

As informações adquiridas pelo líder de projeto guiam suas ações. Essas ações podem ser desde efetuar pequenos ajustes no cronograma até mesmo rever as estimativas do projeto ou fazer o replanejamento de todo o projeto de software.

Neste momento, quando ocorrem as reuniões periódicas de acompanhamento entre o líder e a equipe do projeto, há o compartilhamento do conhecimento, ou seja, acontece a socialização. Outrossim, quando o líder analisa as informações contidas nos artefatos e toma as decisões cabíveis, é possível identificar internalização por meio da conversão do conhecimento explícito contido nos artefatos em conhecimento tácito que guia as ações do líder de projeto. Por outro lado, quando o líder de projeto altera os artefatos e registra nos relatórios de acompanhamento as mudanças ocorridas e os motivos que as gerou, acontece a externalização.

Neste contexto, o CMM fomenta a criação e confiabilidade dos produtos de informação e o processo de externalização, pois preconiza que:

1. Os dados relativos ao valor corrente dos indicadores de medida e ao replanejamento do projeto de software devem ser registrados;
2. Devem ser estabelecidas medições apropriadas, de modo a determinar o estado das atividades de acompanhamento e supervisão do software;

3. O grupo de garantia da qualidade de software deve revisar e/ou auditar as atividades e artefatos relativos ao acompanhamento e supervisão do projeto de software, relatando os respectivos resultados.

Com relação à distribuição da informação no contexto do acompanhamento, o CMM estabelece que as alterações de compromissos aprovadas, que afetam o projeto de software, devem ser comunicadas aos membros da equipe de desenvolvimento e às outras equipes de software relacionadas.

Das atividades de encerramento de projeto, a mais importante, no contexto da criação do conhecimento, é a elaboração do relatório final do projeto onde o líder registra os principais problemas ocorridos, as soluções empregadas, as lições aprendidas, a reavaliação de estimativas e outras informações. Nesta atividade, o líder de projeto procura explicitar o conhecimento tácito adquirido durante a execução do projeto.

Sob essa perspectiva, o líder de projeto em uma empresa de software assume vital importância nos processos de gestão da informação, seja no fomento a inovação (criação de conhecimento), ou na organização dos processos e dos produtos (conhecimento) que são gerados ao longo do ciclo de vida do software.

Porém, uma dificuldade é o fato de que, ocorrendo problemas de atraso, a maioria dos líderes de projeto é levada a apoiar a equipe do projeto em atividades operacionais. Sendo assim, no momento em que o projeto mais necessita das habilidades gerenciais do líder de projeto, ele não está disponível. Isto é, ao invés de identificar as causas do atraso e agir gerencialmente para corrigi-las, o líder faz com que o projeto tenha um programador a mais e nenhum gerente.

- **Supervisão de Projeto de Software**

O líder de projeto de software tem como tarefa principal entender o problema a ser resolvido, gerenciar o fluxo de idéias e, ao mesmo tempo, fazer com que toda a equipe se preocupe com a qualidade do que está sendo gerado. No entanto, há eventos internos ou externos que interferem no projeto e geram problemas cuja solução extrapola as fronteiras da gerência imediata do líder de projeto. Neste caso, surge a figura do gerente sênior que tem o poder de decisão sobre problemas que ultrapassam o limite do poder de ação do líder de projeto.

Situações onde haja a necessidade de aprovação de aumentos no orçamento, de alocação de recursos para projeto, de alterações significativas em datas ou escopo do projeto e até mesmo a análise do desempenho do gerente de projeto exigem a supervisão de um gerente sênior.

Na supervisão fica claro que o principal usuário de informação é o gerente sênior. Suas necessidades informacionais estão relacionadas ao progresso do projeto no nível macro e não em detalhes técnicos e operacionais. Neste caso, a principal fonte de informação do gerente sênior está nas reuniões periódicas com o líder de projeto.

O gerente sênior usa essas informações para analisar os principais problemas do projeto e o desempenho do líder de projeto. Com base nessas informações, ele sugere alterações e aprova ou não as solicitações do líder. Com relação a esses processos o CMM define que:

1. Os compromissos do projeto de software, as mudanças e os acordos estabelecidos com indivíduos e grupos externos à organização devem ser revistos com o gerente sênior, conforme procedimento documentado;
2. As atividades de acompanhamento e supervisão do projeto de software necessitam ser revisadas com o gerente sênior, em bases periódicas. As principais revisões são:

- a. O desempenho relativo às atividades técnicas, aos custos, ao pessoal e ao cronograma;
- b. Os conflitos e tópicos não definidos nos níveis inferiores;
- c. Os riscos do projeto de software;
- d. Os itens de ação devem ser designados, revisados e acompanhados até as respectivas conclusões;
- e. Deve ser elaborado e distribuído aos grupos envolvidos um relatório resumido de cada uma das reuniões do líder de projeto com o gerente sênior.

Durante a supervisão do projeto de software, verifica-se a incidência de compartilhamento de conhecimento tácito entre o líder de projeto e o gerente sênior.

De forma complementar, ocorre a internalização quando são efetuadas análises comparativas entre o estimado e o realizado.

5.3. A Gestão de Requisitos

Como foi visto no capítulo 3, a gestão de requisitos tem como objetivo manter um controle sobre os requisitos de software que deverão ser atendidos. Tem a finalidade também de estabelecer um entendimento comum sobre estes requisitos, ou seja, entre o contratante e a equipe do projeto de software. Sendo assim, a gestão de requisitos envolve o estabelecimento de compromissos, tanto técnicos quanto aqueles relativos a datas de entrega ou outras restrições³⁷, de modo que a equipe do projeto possa garantir a documentação, planos e atividades consistentes com os requisitos.

No início do projeto de software, surgem incertezas e ambigüidades referentes às funcionalidades que o software deve ter para atender às necessidades do usuário e/ou cliente. Sob esta perspectiva e com a finalidade de tornar o

³⁷ Referem-se às restrições funcionais, de negócio, operacionais, tecnológicas.

ambiente do projeto menos obscuro, as atividades de engenharia³⁸ de requisitos são iniciadas. A equipe do projeto, mais especificamente, o analista de requisitos, começa a identificar as informações necessárias para fundamentar a definição dos requisitos do software e o escopo do produto a ser desenvolvido.

De acordo com as necessidades informacionais definidas, inicia-se o processo de aquisição das informações que demanda acesso a fontes diversas, tais como o cliente, os usuários, documentos, normas, políticas, leis e outros.

Com relação aos processos, descritos acima, de identificar as necessidades informacionais e de aquisição da informação no contexto do processo de desenvolvimento de software, a área-chave de processo gestão de requisitos preconiza a necessidade de definir e documentar:

1. Os requisitos não técnicos³⁹ que influenciam e determinam as atividades do projeto de software;
2. Os requisitos técnicos funcionais⁴⁰ e não-funcionais⁴¹ para o software;
3. Os critérios de aceitação que serão utilizados para validar a concordância do software com os requisitos definidos.

Ainda sobre os processos de identificação das necessidades informacionais e de aquisição da informação, é possível identificar claramente a conversão de conhecimento tácito em tácito, ou seja, a socialização definida por Nonaka & Takeuchi. Isto ocorre quando as fontes informacionais utilizadas pelo analista de requisitos são os clientes, os usuários do software ou o especialista de negócio⁴². Neste caso, o compartilhamento do conhecimento tácito destas fontes pode ser realizado por meio de reuniões/entrevistas presenciais ou à distância e observação do ambiente no qual o software será implantado.

³⁸ Engenharia de requisitos é o conjunto das técnicas de levantamento, documentação e análise de requisitos.

³⁹ Prazo, custo, pessoal, fatores políticos e organizacionais.

⁴⁰ Descrição das diversas operações que clientes e usuários esperam que sejam realizadas pelo software.

⁴¹ Determinam as características desejáveis do software quanto à usabilidade, confiabilidade, desempenho, suportabilidade, bem como descrevem restrições de projeto e conformidades legais.

⁴² Profissional que detém conhecimento especializado sobre uma determinada área, tema ou assunto.

Uma vez que os requisitos de software estão definidos e validados pelo cliente e/ou pelos usuários, o analista de requisitos inicia a análise e documentação destes, conforme preconiza o nível 2 do CMM. Essas atividades têm como produtos finais os artefatos que, na maioria das vezes, são diagramas, documentos e/ou relatórios que definem o escopo do produto de software⁴³ e detalham os requisitos técnicos ou não técnicos a serem implementados pela equipe de desenvolvimento.

Sob a perspectiva do gerenciamento da informação, esses artefatos constituem os produtos de informação resultantes das atividades de gestão de requisitos. Sobre a confiabilidade das informações oferecidas nesses produtos, o CMM estabelece que:

1. Para cada projeto de software, é necessário definir os responsáveis pelo levantamento, análise e documentação dos requisitos de software;
2. É necessário manter os requisitos consistentes com os planos, produtos e demais atividades de software;
3. Antes de serem incorporados ao projeto de software, os requisitos definidos devem ser revisados pela equipe de desenvolvimento para que esta possa:
 - a. Identificar requisitos incompletos e/ou ausentes;
 - b. Determinar se os requisitos são factíveis e apropriados de serem implementados por software, se estão claros e apropriadamente enunciados, consistentes entre si e passíveis de testes;

Os requisitos com problemas em potencial são revisados em conjunto com o analista de requisitos e as alterações necessárias devem ser realizadas.

4. Medições que devem ser realizadas para determinar a qualidade dos requisitos definidos;
5. Os produtos de gestão de requisitos devem ser revisados e/ou auditados pela equipe de garantia da qualidade de software.

⁴³ Define as características e funções do software. É medido em relação aos requisitos de software.

Quanto aos modos de conversão de conhecimento, a formalização dos requisitos no formato de documentos e relatórios de engenharia e gestão de requisitos constitui um bom exemplo de externalização.

Neste momento, os requisitos definidos devem estar devidamente validados, revisados, documentados e disponíveis para os gerentes responsáveis pelo planejamento do projeto e para a equipe desenvolvimento do produto de software.

Por outro lado existem, conforme será mais bem detalhada na seção seguinte, modificações durante o desenvolvimento de um software. Aqui ressaltaremos as atividades e produtos oriundos das mudanças de requisitos.

Uma vez detectada uma mudança de requisitos, seja alteração, inclusão ou exclusão, é necessário analisar os seus impactos. Dessa forma, o analista de requisitos deve rever a especificação de requisitos, em especial as matrizes de interação e de rastreabilidade. Caso a mudança solicitada ocasione impactos, a documentação de requisitos deve ser atualizada para refletir as alterações. Tais alterações são comunicadas as partes interessadas (líder e equipe do projeto, cliente, usuários e outros), por meio de reuniões ou relatórios. Além disso, são efetuadas medições para acompanhar a situação dos requisitos de software quanto à incidência de alterações.

Durante esse processo, verificam-se três modos de conversão do conhecimento. O primeiro é a internalização que ocorre quando o analista de requisitos analisa a documentação de requisitos para avaliar os impactos da solicitação de mudança. O segundo, que é a socialização, acontece quando o analista de requisitos compartilha os impactos com as partes interessadas envolvidas com a alteração. O terceiro é a externalização e ocorre quando as soluções plausíveis e acordadas com as partes interessadas são formalizadas por meio da alteração da documentação de requisitos e dos demais produtos de software relacionados.

5.4. A Garantia da Qualidade de Software

Antes de iniciarmos a descrição sobre garantia da qualidade de software, é pertinente esclarecer os conceitos de *qualidade de software* e *controle da qualidade de software*.

Para Pressman (2002), qualidade de software é a conformidade com requisitos funcionais ou não funcionais formalmente declarado, com padrões de desenvolvimento explicitamente documentados e com características implícitas⁴⁴. Isso é o que se espera em todo software desenvolvido profissionalmente.

O controle da qualidade de software envolve um conjunto de inspeções, revisões e testes aplicados durante o processo de software. Tais revisões e inspeções visam garantir que os produtos gerados satisfaçam aos requisitos especificados para eles. Por outro lado, o controle da qualidade também inclui um ciclo de realimentação no processo de software que cria o produto. A coalizão das características de medição e realimentação faz com que o controle da qualidade seja parte integrante do processo de software.

Sendo assim, a garantia da qualidade de software, conforme descrita no capítulo 1, é uma atividade guarda-chuva que é aplicada ao longo do processo de software. Ela agrega as funções gerenciais de revisar e auditar o processo e o produto de software para verificar se os mesmos estão cumprindo os procedimentos e padrões adotados pela organização. Além disso, envolve o relato dos desvios ocorridos, cujo propósito é prover às gerências de software dos dados necessários para que elas fiquem informadas sobre a qualidade do produto. Caso sejam identificados problemas, cabe a gerência aplicar os recursos necessários para solucioná-los.

⁴⁴ São as expectativas dos clientes e usuários, que são cobradas por estes, embora não estejam documentadas.

A garantia da qualidade de software é realizada por dois grupos distintos, a saber:

- Engenheiros de software que buscam a qualidade desenvolvendo atividades de garantia e controle da qualidade por meio do uso de métodos e medidas técnicas, da condução de revisões técnicas formais e da realização de testes de software;
- Grupo de garantia da qualidade de software busca a qualidade efetuando o planejamento, a supervisão, o registro e a análise das atividades de auditoria.

A área-chave de processo garantia da qualidade de software do nível 2 do CMM abrange apenas as atividades do grupo de GQS. Portanto, a análise desenvolvida nesta pesquisa restringe-se a este universo.

O grupo de GQS é acionado logo no início do projeto de software para ajudar a equipe do projeto, em especial o líder, a estabelecer os planos, os padrões e os procedimentos que serão utilizados. Neste caso, as necessidades informacionais são semelhantes às identificadas na atividade de planejamento do projeto de software e, por conseguinte, o processo de aquisição da informação também. Neste momento, verifica-se a ocorrência de dois modos de conversão: socialização e externalização.

O primeiro se dá quando o líder do projeto e o grupo de GQS debatem sobre o melhor modelo de processo de software a ser abordado no projeto, considerando principalmente as características do produto, da equipe e da tecnologia. O segundo ocorre no momento em que a estratégia a ser adotada é explicitada na forma de plano de projeto de software.

Adicionalmente, o grupo de GQS elabora o seu primeiro produto de informação, que é o plano de GQS do projeto de software. Durante a elaboração do plano, as principais fontes informacionais são as reuniões com o líder de projeto, o plano do projeto de software e o cronograma. O plano de GQS rege as atividades de

garantia da qualidade realizadas pela equipe de engenharia e pelo grupo de GQS. Nele estão contidas informações sobre:

- As avaliações, auditorias e revisões a serem realizadas;
- Os padrões que são aplicáveis ao projeto;
- Os procedimentos para relato e acompanhamento de erros;
- Os documentos a serem produzidos pelo grupo de GQS;
- Quantidade de realimentação fornecida à equipe de projeto do software.

As informações contidas no plano de GQS são utilizadas ao longo de todo o processo de software pela equipe de projeto e pelo grupo de GQS. No caso do grupo de GQS, essas informações em conjunto com o processo de software definido são insumos para a realização das revisões das atividades e auditoria dos produtos de software. Com relação ao plano de GQS, é pertinente identificá-lo como um exemplo de externalização da estratégia de garantia da qualidade de software a ser utilizada no projeto.

Durante as revisões e auditorias, o grupo de GQS revê as atividades e produtos, identifica, documenta e acompanha os desvios e verifica se as correções são feitas a contento. Os resultados das revisões e auditorias são repassados para o líder de projeto, em intervalos periódicos. Caso os desvios não sejam tratados internamente pela equipe do projeto, eles são acompanhados pelo gerente sênior e pelo grupo de GQS da organização, necessariamente nesta ordem, até que sejam resolvidos.

O nível 2 do CMM fomenta, no contexto da garantia da qualidade de software, os processos de identificação das necessidades informacionais, aquisição, distribuição e uso da informação e geração de produtos de informação quando preconiza que:

1. Deve ser preparado um plano de GQS para o projeto de software, conforme procedimento documentado;

2. A equipe de GQS deve participar da preparação e revisão do plano, padrões e procedimentos do projeto de desenvolvimento de software;
3. A equipe de GQS deve relatar, em bases periódicas, os resultados de suas atividades para a equipe de engenharia de software;
4. Os desvios identificados nas atividades e artefatos de software devem ser documentados e tratados, conforme procedimento documentado;
5. As atividades de GQS devem ser revistas com o gerente de projeto tanto em bases periódicas quando se atingir marcos predefinidos;
6. As atividades e os artefatos de software da equipe de GQS do projeto devem ser revistos, em intervalos regulares, por peritos independentes da equipe de GQS;
7. As atividades de GQS devem ser revistas, em bases periódicas, pelo gerente sênior;

A área-chave de processo garantia da qualidade de software, representada pelo grupo de GQS, é o agente impulsionador do comportamento adaptativo do ciclo de gerenciamento da informação definido por Choo (1998a). Isto se dá porque o grupo de GQS, por meio das revisões e auditorias, identifica os desvios na implementação do processo por parte das equipes de projeto. Além disso, ele relata os desvios para as gerências de software e, principalmente, envia sugestões para o grupo da organização responsável pela melhoria do processo de software.

Neste contexto, de uma forma geral, o trabalho da garantia da qualidade de software supervisiona a internalização do processo de software pelos engenheiros e gerentes de software. Ademais, ela também inspeciona a qualidade do conhecimento explícito, ou seja, dos produtos do trabalho de software realizado nos projetos.

5.5. A Gestão de Configuração de Software

Conforme descrito no capítulo 1, Pressman (2002) afirma que a saída do processo de software é informação que pode ser, no contexto da engenharia de software, dividida em três grandes categorias:

- *Programas de computador* (tanto código fonte quanto executável);
- *Documentos* que especificam os requisitos e as características do software (tanto para profissionais – documentação técnica e gerencial - como para usuários – manual do usuário);
- *Dados* contidos nos programas ou externos a eles e que podem ser fontes de informações estruturadas ou não, por exemplo, banco de dados, arquivos texto, multimídia, etc.

Os itens que compreendem toda a informação produzida como parte do processo de software são chamados coletivamente de configuração de software. À medida que o desenvolvimento avança, o número de *itens de configuração de software* aumenta e, na maioria das vezes, muito rapidamente. Isto ocorre porque, conforme visto no capítulo 3 – FIG. 10, a especificação de um sistema abrange um conjunto de documentos, o que, de certa forma, cria uma hierarquia complexa de informação. Além disso, há uma outra variável no processo que afeta diretamente os itens de configuração - a modificação. Esta, inevitavelmente, sempre está presente em qualquer que seja a fase do processo de software. Isto é, a mudança é um fato, por isso é necessário gerenciá-la para que se possa garantir a consistência das informações produzidas no processo de software.

Neste contexto, a gestão de configuração de software - GCS é uma atividade guarda-chuva que é aplicada ao longo de todo o processo de software. Ela consiste em um conjunto de atividades estruturadas para organizar e armazenar os itens de configuração, identificar e controlar mudanças, assegurar que essas mudanças sejam adequadamente implementadas. Além disso, a GCS reporta essas mudanças a todos os indivíduos e/ou grupos que sejam parte interessada.

Com os processos de gestão da informação postulados por Choo, verifica-se uma relação entre o processo de organização e armazenamento da informação e a gestão de configuração de software. É importante ressaltar que a GCS estabelece a infra-estrutura da organização, armazenamento e controle de todas as informações produzidas no processo de software, inclusive nas atividades de configuração de software.

No processo de desenvolvimento de software, durante o planejamento do projeto de software, é necessário especificar um plano de gestão de configuração de software que defina a estratégia do projeto para execução das atividades de gestão de configuração de software.

Para elaborar esta estratégia, o líder de projeto reúne-se com o membro da equipe responsável pela configuração do projeto. Neste caso, eles necessitam das informações contidas no plano do projeto de software, cronograma, lista de recursos e outros. Além do plano de gestão de configuração, as atividades de gestão de configuração de software geram os seguintes produtos de informação: o repositório de armazenamento dos itens de configuração do projeto, as solicitações de modificação de software, os relatórios e ordens de modificação de engenharia.

Com relação a esses produtos, o CMM fomenta a sua criação e a garantia de sua confiabilidade, pois preconiza que:

1. Deve existir ou estar estabelecida, uma comissão com autoridade para gerenciar as *baselines*⁴⁵ do projeto de software;
2. Deve existir uma equipe responsável pela coordenação e implementação de gestão de configuração para o projeto, ou seja, a equipe de GCS do projeto;
3. Deve ser preparado um plano de GCS para cada projeto de software, conforme procedimento documentado;

⁴⁵ Uma especificação ou produto que foi formalmente revisado e aprovado, o qual daí em diante serve como base para o desenvolvimento futuro e que pode ser modificado apenas através de procedimentos formais de controle de modificação (IEEE Std No.610.12-1990).

4. Deve ser estabelecido um repositório para a gestão de configuração que contemple as *baselines* de software e forneça:
 - a. Mecanismos de armazenamento e recuperação de itens/unidades de configuração;
 - b. Mecanismos de compartilhamento e transferência de itens/unidades de configuração entre os grupos afetados e entre níveis de controle dentro do repositório;
 - c. Mecanismos que ajude na utilização de padrões para itens/unidades de configuração;
 - d. Mecanismos de armazenamento e recuperação de versões de arquivos de itens/unidades de configuração;
 - e. Mecanismos que ajudem a garantir a correta geração de produtos a partir do repositório de *baselines* do software;
 - f. Mecanismos de manutenção da estrutura do repositório e seu conteúdo.
5. Os produtos de trabalho de software a serem colocados sob gestão de configuração devem ser identificados;
6. As solicitações de alterações e relatórios de problemas para todos os itens/unidades de configuração são iniciados, revisados, aprovados e encaminhados, de acordo com um procedimento documentado;
7. Auditorias nas *baselines* do software devem ser conduzidas de acordo com um procedimento documentado;
8. A equipe de GCS faz, em intervalos periódicos, a auditoria das *baselines* para verificar se elas estão de acordo com a documentação que as define;
9. A equipe de garantia da qualidade de software (GQS) deve revisar e/ou inspecionar as atividades e produtos de trabalho relativos à gestão de configuração, relatando os respectivos resultados.

Com relação aos modos de conversão, verifica-se a presença da combinação. Isto ocorre quando se elabora a estratégia de configuração de

software, ou seja, o plano de GCS, tomando como base o conhecimento explícito contido no plano de projeto de software, na lista de recursos computacionais (hardware e software), no plano de desenvolvimento de software e no cronograma.

Por fim, sob a perspectiva da criação de conhecimento, a gestão de configuração de software é importante porque ela é uma atividade que fomenta a infra-estrutura necessária para organizar e armazenar o conhecimento explícito gerado durante a execução do projeto de software.

5.6. A Gestão de Subcontratação de Software

Como foi visto no capítulo 3, a gestão de subcontratação de software tem como objetivo selecionar subcontratados qualificados e gerenciá-los de modo efetivo. Isto pode ser feito por meio do estabelecimento de compromissos, acompanhamento e revisão do desempenho e resultados do subcontratado.

Antes porém, é preciso definir e planejar o trabalho a ser subcontratado, ou seja, estabelecer as funções ou subsistemas, padrões, procedimentos, produtos e as atividades de software a serem subcontratados. Para tanto, o gerente de subcontratação necessita de informações referentes aos requisitos técnicos e não técnicos do trabalho a ser subcontratado. Sendo assim, as principais fontes de informação são produtos de informação oriundos das atividades de planejamento de projeto de software e gestão de requisitos, que são:

- Requisitos de software;
- Plano de desenvolvimento de software;
- Padrões e procedimentos de software.

Muitos fatores contribuem para a decisão de subcontratar uma parte do trabalho do contratante principal, tais como falta de recurso disponível ou especializado. Neste contexto, os subcontratados podem ser selecionados com base em considerações técnicas e também em alianças estratégicas de negócio. Sendo assim, o gerente de subcontratação necessita de informações referentes à

capacidade dos participantes da licitação para realizar o trabalho. O processo de aquisição dessas informações abrange o acesso às propostas comerciais apresentadas para a subcontratação planejada, ao registro de trabalhos similares realizados anteriormente, aos certificados da empresa e do seu corpo técnico disponível.

Com relação aos processos de identificação das necessidades informacionais e aquisição da informação, o CMM nível 2 preconiza que:

1. O trabalho a ser subcontratado é definido e planejado de acordo com um procedimento documentado que estabelece:
 - a. Os produtos e as atividades de software a serem subcontratados são selecionados com base em uma avaliação equilibrada das características técnicas e não técnicas do projeto;
 - b. Que a especificação do trabalho a ser subcontratado e dos padrões e procedimentos a serem seguidos sejam derivados dos requisitos do sistema alocados ao software, dos requisitos de software, do plano de desenvolvimento de software, dos padrões e procedimentos de software;
2. O subcontratado de software é selecionado de acordo com uma avaliação da capacidade do participante da licitação em realizar o trabalho, conforme um procedimento documentado.

Uma vez selecionado o subcontratado, a organização estabelece um acordo documentado ou contrato comercial, onde são definidos os termos contratuais entre contratante e subcontratado. Estes termos referem-se aos requisitos, aos produtos e/ou serviços a serem entregues ao contratante, as revisões dos produtos, os procedimentos e critérios de aceitação do produto. O acordo ou contrato é um dos principais produtos de informação gerado nesta área-chave, pois ele é utilizado como base para gerenciar o subcontratado.

As atividades de software são acompanhadas pelo contratante por meio do plano de desenvolvimento de software do subcontratado. Este plano consiste em outro importante produto de informação desta área-chave.

Com relação à confiabilidade dos produtos de informação e a distribuição e uso da informação, o nível 2 do CMM preconiza que:

1. O plano de desenvolvimento de software do subcontratado deve ser revisado e aprovado pelo contratante;
2. As alterações no contrato do subcontratado de software, nos termos e condições da subcontratação e em outros compromissos devem ser decididas de acordo com um procedimento documentado;
3. As medições devem ser realizadas e utilizadas para determinar a situação das atividades da gestão de subcontratação de software;
4. A equipe de garantia da qualidade de software do contratante deve revisar e/ou auditar as atividades e os produtos de trabalho para gerenciar o subcontrato e reportar os resultados;
5. A gerência do contratante deve conduzir revisões periódicas da situação/coordenação com a gerência do subcontratado de software;
6. Devem ser mantidas revisões e intercâmbios técnicos com o subcontratado de software que:
 - a. Fornecem ao subcontratado a visibilidade das necessidades e dos desejos dos clientes e dos usuários finais, quando apropriado;
 - b. Monitoram as atividades técnicas do subcontratado;
 - c. Verificam se a interpretação e a implementação dos requisitos técnicos pelo subcontratado estão em conformidade com os requisitos do contratante;
 - d. Verificam se os compromissos estão sendo cumpridos;
 - e. Verificam se as questões técnicas estão sendo resolvidas no momento.

7. As revisões formais para tratar as realizações e os resultados do desenvolvimento de software do subcontratado devem ser conduzidas em marcos selecionados do cronograma de acordo com um procedimento documentado;
8. As atividades de gestão de subcontratação de software devem ser revisadas com o líder de projeto, em períodos regulares ou motivadas por um evento;
9. As atividades de gestão de subcontratação de software devem ser revisadas sistematicamente com o gerente sênior.

No momento da entrega do produto e/ou serviço subcontratado, é importante verificar a conformidade destes com os procedimentos e critérios de aceitação que foram definidos no contrato estabelecido entre contratante e subcontratado. Neste caso o CMM nível 2 estabelece que:

1. O contratante deve conduzir os testes de aceitação como parte da entrega dos produtos de software de acordo com um procedimento documentado que especifique que:
 - a. Os procedimentos e os critérios de aceitação para cada produto são definidos, revisados e aprovados pelo contratante e pelo contratado antes do teste;
 - b. Os resultados dos testes de aceitação devem ser documentados;
 - c. Um plano de ação deve ser estabelecido para qualquer produto de software que não passe pelo seu teste de aceitação.
2. O desempenho do subcontratado de software deve ser avaliado em intervalos periódicos e a avaliação é revisada com o subcontratado.

Com relação aos modos de conversão de conhecimento, verifica-se que o CMM nível 2 fomenta a socialização quando preconiza as reuniões de revisões e intercâmbios técnicos entre contratante e subcontratado. A externalização é estimulada por meio do registro dos parâmetros de seleção dos subcontratados e dos resultados das avaliações periódicas dos subcontratados.

Relação entre as áreas-chaves de processo do CMM nível 2, os processos de gestão da informação e os modos de conversão do conhecimento

QUADRO 8

Áreas-chaves de Processo do Nível 2 do CMM	Processos de Gestão da Informação ⁴⁶						Modos de Conversão do Conhecimento ⁴⁷			
	Necessidade de Informação	Aquisição da Informação	Organização e Armazenamento da Informação	Produtos e Serviços de Informação	Distribuição da Informação	Uso da Informação	Socialização	Externalização	Combinação	Internalização
Planejamento de Projeto de Software	X	X		X		X	X	X	X	
Acompanhamento e Supervisão de Projeto de software	X	X		X	X	X	X	X		X
Gestão de Requisitos	X	X		X	X	X	X	X		X
Gestão de Configuração de Software	X	X	X	X					X	
Garantia da Qualidade de Software	X	X		X	X	X	X	X		
Gestão de Subcontratação de Software	X	X		X	X	X	X	X		

Fonte – A autora, 2004

⁴⁶ CHOO, Chun Wei. Information Management for Intelligent Organization: the art of scanning the environment. Medford, NJ: Asis, 1995.

⁴⁷ NONAKA, Ikujiro; TAKEUCHI, Hirotaka. Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação. Rio de Janeiro: Campus, 1997.

6. RESULTADOS – ANÁLISE E INTERPRETAÇÕES DOS DADOS

As análises dos dados coletados e a discussão dos resultados são tratadas a seguir. Essa parte é descrita conforme a estrutura do questionário e as categorias selecionadas.

QUADRO 9
Estrutura do questionário

Partes	Objetivos Específicos	Questões
Parte 1- Perfil do respondente	Caracterizar o Gerente Sênior, o Líder do Projeto e a equipe de desenvolvimento	1-5
Parte 2 - Características dos projetos de software executados na área	Identificar os tipos de projetos de software executados pelas equipes de desenvolvimento	6-12
Parte 3 – Área-chave de processo Gestão de Requisitos	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>gestão de requisitos</i> relacionadas com a gestão da informação	13-22
Parte 4 - Área-chave de processo Planejamento do Projeto de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>planejamento do projeto de software</i> relacionadas com a gestão da informação.	23-29
Parte 5 - Área-chave de processo Acompanhamento e Supervisão do Projeto de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>acompanhamento e supervisão do projeto de software</i> relacionadas com a gestão da informação	30-37
Parte 6 Área-chave de processo Garantia da Qualidade de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>garantia da qualidade software</i> relacionadas com a gestão da informação	38-44
Parte 7 - Área-chave de processo Gestão de Configuração de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>gestão de configuração de software</i> relacionadas com a gestão da informação	45-49
Parte 8 - Área-chave de processo Subcontratação de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>subcontratação de software</i> relacionadas com a gestão da informação	50-58

6.1. Perfil do Respondente

Os dados referentes ao perfil do respondente, obtidos por meio da 1ª parte do questionário e complementados pela análise de documentos, caracterizam a população deste estudo de caso. Esta é composta de 19 respondentes do total de 24 indivíduos para os quais foram enviados questionários. O percentual de participação representa 79,1%, conforme TAB. 1.

TABELA 1
Distribuição dos respondentes por grupo de gerentes de software

Gerentes de Software	Nº de Indivíduos	Nº de Respondentes	% de Respostas
Gerente Sênior	6	5	83,3
Líder de Projeto	18	14	77,7
Total	24	19	79,1

TABELA 2
Distribuição dos gerentes seniores respondentes por projeção regional

Gerentes de Software	Projeções Regionais					
	Belém	Belo Horizonte	Curitiba	Fortaleza	Recife	Salvador
Gerente Sênior	1	1	1	1	1	1
Nº Respondentes	1	1	1	1	1	0
% de Respostas	100	100	100	100	100	0

TABELA 3
Distribuição dos líderes de projeto respondentes por projeção regional

Gerentes de Software	Projeções Regionais					
	Belém	Belo Horizonte	Curitiba	Fortaleza	Recife	Salvador
Líder de Projeto	1	3	3	3	5	3
Nº Respondentes	1	3	1	1	5	2
% de Respostas	100	100	33,3	33,3	100	66,7

TABELA 4
Perfil dos respondentes – gerente sênior

Grau Acadêmico				
Técnico	Graduação	Especialização	Mestrado	Doutorado
-	-	80%	20%	-
Formação Acadêmica*				
Ciência da Computação	Engenharia	Matemática	Administração	Ciências Econômicas
20%	20%	-	60%	20%
Tempo de Atuação no Desenvolvimento de Software				
até 3 anos	de 4 a 8 anos	de 9 a 13 anos	de 14 a 18 anos	mais de 20 anos
-	20%	-	20%	60%

Resposta múltipla, não soma 100%. (n=5)

Com relação ao perfil dos respondentes, a partir dos percentuais mais elevados para as características investigadas, verificou-se que os gerentes seniores em sua maioria são administradores com treinamento em cursos de especialização. Este foi o grau acadêmico mais alto detectado naqueles que atuam na área de desenvolvimento de software há mais de 20 anos.

TABELA 5
Perfil dos respondentes – líder de projeto

Grau Acadêmico				
Técnico	Graduação	Especialização	Mestrado	Doutorado
-	33,3%	44,4%	22,2%	-
Formação Acadêmica				
Ciência da Computação	Engenharia	Matemática	Administração	Outros *
55,5%	-	11,1%	11,1%	22,2%
Tempo de Atuação no Desenvolvimento de Software				
até 3 anos	de 4 a 8 anos	de 9 a 13 anos	de 14 a 18 anos	mais de 20 anos
-	22,2%	33,3%	11,1%	33,3%

Resposta múltipla, não soma 100%. (n=9)

* Ciências Contábeis e Economia

Para os líderes de projeto, verificou-se que a maioria tem formação em ciência da computação e também apresenta certificação de especialização como

grau acadêmico mais alto. São profissionais que atuam na área de desenvolvimento de software há mais de 13 anos.

A partir desses dados, constatou-se que a predominância da formação em Administração dos gerentes seniores está em harmonia com as suas atividades de supervisão do projeto de software. Essa harmonia também é identificada em relação aos líderes de projeto que, em sua maioria, tem formação em ciência da computação e, portanto, possuem as habilidades e conhecimentos técnicos necessários para acompanhar a execução de um projeto de software.

6.2. Características do Projeto de Software

Os dados referentes ao perfil do projeto de software, obtidos por meio da 2ª parte do questionário e complementados pela análise de documentos, caracterizam os projetos de software. Metade dos respondentes afirmaram que os projetos de software executados em suas áreas são de grande porte (TAB. 6). Com relação a este dado, é pertinente salientar que 71% desses respondentes estão lotados na mesma regional.

TABELA 6
Tamanho do Projeto de Software

Tamanho do Projeto de Software		
Grande Porte (acima de 150 homem-dia ou 100 pontos de função)	Médio Porte (até 150 homem-dia ou 100 pontos de função)	Pequeno Porte (até 30 homem-dia ou 20 pontos de função)
50%	36%	21%

Resposta múltipla, não soma 100%.

Com relação aos produtos de software, verificou-se que 52% dos respondentes afirmaram que o tipo de software gerado nos projetos executados em suas áreas encaixa-se na categoria de sistemas de informações gerenciais e operacionais. Este grupo de software compreende as aplicações de controle e acompanhamento de transações. O grupo de sistemas de informações estratégicas compreende 8,6%. Por outro lado, 30% dos projetos geram aplicativos para Web

e/ou internet e outros 8,6% produzem PGD - programas geradores de declaração (TAB. 7). Os PGDs são disponibilizados para os contribuintes para que eles possam gerar declarações em meio magnético e posteriormente transmiti-las via internet para a SRF – Secretaria da Receita Federal

TABELA 7
Tipos de produto de software

Produto de Software						
Software Básico	Software de Prateleira	Sistema de Informações Estratégicas	Sistema de Informações Gerenciais e Operacionais	Aplicativo para Web e/ou Internet	Sites Estáticos	Outros*
-	-	8,6%	52%	30%	-	8,6%

Resposta múltipla, não soma 100%

(*) Programa Gerador de Declarações

Os dados da TAB. 8, mostram que a maioria (71,4%) dos respondentes afirmaram que utilizam a metodologia de análise estruturada, 21,4% trabalham com metodologia orientada a objetos e 7,2% com a análise essencial.

TABELA 8
Metodologia de análise utilizado

Metodologia		
Estruturada	Orientada a Objetos	Essencial
71,4%	21,4%	7,2%

Os projetos executados tem uma média de duração que varia entre 1 semana a 5,5 meses e a equipe alocada é composta em média por 3 pessoas. Com relação à definição do papel a ser desempenhado pelos membros da equipe do projeto, 93% dos respondentes afirmaram que os papéis são bem definidos, porém há acúmulo de atividades para uma equipe pequena, o que a sobrecarrega. Os outros 10% afirmaram que os papéis são bem definidos e que o acúmulo de atividades existe, porém não sobrecarrega a equipe.

Com relação à disponibilidade das pessoas alocadas para cada projeto, em um total de 13 respondentes, (46%) afirmaram que a equipe trabalha em mais de um projeto paralelamente.

Assim, a análise dos dados evidenciou que a organização em estudo trabalha com projetos de software de tamanho, características e complexidades diversas. Com relação ao tipo de produto gerado nos projetos de software, um ponto a destacar é a predominância do grupo de sistema de informações gerenciais e operacionais, assim como do grupo de aplicativos para *Web* e/ou internet. Esses grupos de software abrangem aplicações críticas para o cliente, que são utilizadas por um universo de usuários muito grande, além disso, são produtos cuja data de entrega é definida por instruções normativas. Isto significa que, na maioria das vezes, o prazo de entrega é um dos principais requisitos dos projetos de software.

Enfim, com relação ao perfil da equipe do projeto, verificou-se que seus membros possuem papéis bem definidos no projeto, porém são equipes bastante enxutas, ou seja, contém um número mínimo de profissionais. Além disso, os profissionais trabalham em mais de um projeto, o que acarreta a sobrecarga da equipe.

6.3. Área-Chave de Processo Gestão de Requisitos

Os dados referentes à área-chave de processo gestão de requisitos, obtidos por meio da 4ª parte do questionário e complementados pela análise de documentos, identificam as práticas e dificuldades na aplicação das diretrizes desta ACP relacionadas com a gestão da informação.

Quanto aos padrões e procedimentos definidos pela organização para gerenciar requisitos, 100% dos gerentes seniores e 57% dos líderes de projeto afirmaram que são flexíveis de acordo com as características do projeto, o que facilita a execução do projeto. Por outro lado, 43% dos líderes de projeto responderam que os padrões e procedimentos são rígidos, porém não são considerados como agentes que dificultam a execução do projeto.

Dessa forma, concluiu-se que a política de gestão de requisitos da empresa é bem aceita pelos gerentes de software. Na visão dos respondentes isso não é um obstáculo para os projetos de software.

TABELA 9
Fontes de informação utilizadas durante a definição de requisitos

Fontes de Informação	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Reunião presencial com o cliente /usuário	5	100	14	100
Reuniões via áudio ou vídeo-conferência com o cliente/usuário	3	60	5	36
Questionários aplicados aos usuários	-	-	-	-
Pesquisa documental (leis, políticas, normas, procedimentos etc)	-	-	3	21
Prototipagem	1	20	5	36
Os requisitos documentados de outros projetos similares, ou seja, com as mesmas características e contexto	2	40	2	14
Observação do ambiente no qual o software será inserido	-	-	-	-
Os requisitos são previamente definidos pelo cliente/usuário	3	60	5	36
Outros (correio eletrônico)	-	-	1	7

Resposta múltipla, não soma 100%

Com base nos dados apresentados na TAB. 9, constatou-se que, tanto sob a perspectiva dos gerentes seniores como dos líderes de projeto, as fontes de informação mais utilizadas são aquelas que apresentam uma interação direta ou indireta com o cliente/usuário. Neste caso, as fontes são as reuniões presenciais ou a distância via áudio/vídeo-conferência, a prototipagem e a documentação dos requisitos previamente especificada pelo cliente/usuário. Esses resultados vão de encontro com as idéias de (Choo, 1998b) que afirma que as pessoas são as principais fontes de informação. Tais resultados evidenciam a ocorrência da socialização, um dos modos de conversão do conhecimento postulados por Nonaka & Takeuchi (1997).

Ainda analisando a TAB. 9, é importante evidenciar o pouco uso da base de conhecimento organizacional referente a requisitos de software, quando verificou-se um baixo índice na sua utilização. Isto é, os requisitos documentados de outros projetos similares que possuem as mesmas características e o mesmo contexto são pouco utilizados pelos líderes de projeto.

TABELA 10
Dificuldades no levantamento e documentação dos requisitos

Dificuldades	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Cliente/usuário não comprometidos com o projeto	-	-	2	14
Conflitos entre o cliente/usuário e a organização desenvolvedora de software	-	-	2	14
Cliente/usuário não consciente da importância da qualidade dos requisitos e o seu registro	2	40	5	36
Falha na comunicação entre cliente/usuário e equipe desenvolvedora	2	40	2	14
Pressão excessiva de prazo	5	100	13	93
Membros da equipe de desenvolvimento treinados inadequadamente	1	20	2	14
Membros da equipe de desenvolvimento inexperientes	-	-	1	7
Ausência ou número insuficiente de membros da equipe de desenvolvimento com perfil de analista de requisitos	3	60	5	36
Outros (Mudança freqüente de requisitos)	-	-	1	7

Resposta múltipla, não soma 100%

Com relação às dificuldades para levantar e documentar os requisitos de software, identificou-se, em ambos os grupos, que o principal empecilho é a pressão excessiva sobre o prazo de entrega do produto final conforme pode ser visto na TAB. 10. Em segundo lugar, observa-se a problemática do número insuficiente de membros da equipe com perfil de analista de requisitos. Isto significa que há a necessidade de profissionais com habilidades para levantar, documentar, analisar requisitos e também gerenciar as mudanças de definição. É importante ressaltar que, em terceiro lugar, aparece a questão do cliente/usuário não estar consciente da importância da qualidade e do registro dos requisitos de software.

TABELA 11
Revisão dos requisitos levantados e documentados

Requisitos Revisados	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Sim	4	80	11	79
Não (Por quê?)	-	-	-	-
Às vezes (Por quê?)	1	20	3	21

A revisão dos requisitos está diretamente relacionada com a consistência e confiabilidade dos produtos de informação gerados durante as atividades de

engenharia de requisitos. De acordo com os dados da TAB. 11, verificou-se que a maioria (80%) dos respondentes afirmaram que os requisitos são revisados. Porém, os outros 20% argumentou que às vezes a revisão não é possível em virtude das razões abaixo relatadas:

- A descrição da solicitação de serviço é clara e não se faz necessária a revisão;
- Falta pessoa disponível para a atividade;
- Pressão sobre o prazo de entrega do produto;
- A maioria dos requisitos já é conhecida pela equipe, o que dispensa a revisão.

Ainda com relação à revisão dos requisitos, constatou-se que, de acordo com a TAB. 12, 75% dos respondentes afirmaram que os requisitos são revisados pelo cliente/usuário e pela equipe de desenvolvimento.

TABELA 12
Revisores dos requisitos levantados e documentados

Requisitos Revisados	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
O cliente/usuário	4	80	10	71
A equipe de desenvolvimento	4	80	8	57
Outros (especifique)				
Algumas vezes pares que participam de outros projetos	-	-	1	7
Resposta múltipla, não soma 100%				

Com relação as dificuldade na utilização dos requisitos como fonte de informação para as atividades de engenharia e de gestão, os dados da TAB. 13 mostram que a maioria, cerca de 80% dos grupos de gerentes seniores e de líderes de projeto, considera que o principal obstáculo é a mudança contínua de requisitos. Por outro lado, 29% dos líderes de projeto apontam como dificuldade a questão dos requisitos serem definidos de forma inadequada, e outros 21% indicam a falta de clareza dos requisitos como um obstáculo.

TABELA 13
Dificuldades para utilização dos requisitos como fonte de informação para as atividades de engenharia e de gestão

Dificuldades	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Requisitos são definidos de forma inadequada	-	-	4	29
Requisitos não estão claros	-	-	3	21
Requisitos incorretos	-	-	-	-
Requisitos conflitantes	-	-	-	-
Mudanças contínuas dos requisitos	4	80	12	86
Requisitos são definidos de forma inadequada	-	-	-	-
Outros (Especifique)	1	20	-	-
Resposta múltipla, não soma 100%				

De acordo com os dados coletados, 89% dos respondentes afirmaram que, quando ocorrem mudanças, a especificação de requisitos e demais artefatos relacionados são atualizados. Porém, 11% dos respondentes, mais especificamente 2 líderes de projeto, afirmaram que os artefatos às vezes não são atualizados em virtude da indisponibilidade de tempo. É importante descrever a fala de um gerente sênior quanto a esta questão: *"Falta de conhecimento da equipe no uso de técnicas e ferramentas para levantamento adequado dos requisitos, dificulta inclusive a validação dos mesmos, posteriormente junto aos usuários."*

TABELA 14
Uso das medições de requisitos como suporte à melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe

Uso das medições	Gerente Sênior (n=4)		Líder de Projeto (n=13)	
	N	%	n	%
Sim (Como?)	3	75	12	93
Não (Por quê?)	1	25	1	7

Com relação ao uso das medições referentes aos requisitos, conforme TAB. 14, constatou-se que 75% dos gerentes seniores e 93% dos líderes de projeto afirmaram que as mesmas são utilizadas como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe.

Segue abaixo a especificação, sob a perspectiva dos líderes de projeto e gerentes seniores, de como essas informações são utilizadas:

- Alimentam a base histórica da organização;
- Nas reuniões da equipe para ajudar a entender o andamento do projeto através da análise e acompanhamento das medições dos requisitos;
- Para verificar se a alteração do requisito foi em decorrência de desconhecimento do negócio do cliente, por parte da equipe de desenvolvimento, e se, for o caso, definir se a equipe deve ser treinada outra vez;
- A cada relatório periódico de acompanhamento do projeto são levantadas e analisadas as métricas. As lições aprendidas são registradas no relatório de avaliação final do projeto. Este relatório é eventualmente utilizado como referência para outros projetos ou para a tomada de decisões no decorrer do projeto em andamento.
- Fonte de informação para o replanejamento do projeto ou no planejamento de outros novos;
- Feedback para as partes interessadas;

É importante ressaltar que um dos gerentes seniores destacou que as medições não são utilizadas em virtude do processo de software ainda estar incipiente.

De modo geral, a análise dos dados mostrou que, durante o processo de gestão de requisitos, os profissionais de software praticam a gestão da informação, principalmente porque eles realizam atividades de levantamento, análise e documentação de requisitos. Porém, para exercer tais atividades, esses profissionais apontaram como principais dificuldades:

- A pressão excessiva sobre o prazo de entrega do produto;
- O número insuficiente de membros da equipe com perfil de analista de requisitos;

- Os clientes/usuários não terem a consciência da importância em formalizar os requisitos de software;
- As mudanças contínuas de requisitos.

Enfim, sobre o conhecimento envolvido no processo de gestão de requisitos, verificou-se que a criação do conhecimento se dá principalmente por meio da socialização e da externalização. Outro ponto a ser observado é o baixo índice de uso da base histórica de requisitos de software, o que dificulta a criação de novos conhecimentos. Porém, isso pode ser explicado em virtude do processo ser incipiente em algumas regionais pesquisadas e, por conseguinte, a base histórica ainda não contém informações suficientes.

6.4. Área-Chave de Processo Planejamento de Projeto de Software

A análise dos dados referentes à área-chave de processo de planejamento de projeto de software identificam as práticas e dificuldades na aplicação das diretrizes desta ACP relacionadas com a gestão da informação.

Quanto aos padrões e procedimentos definidos pela organização para planejamento de projeto de software, 100% dos gerentes seniores e 86% dos líderes de projeto afirmaram que são flexíveis de acordo com as características do projeto, o que facilita a execução do projeto. Por outro lado, 14% dos líderes de projeto responderam que os padrões e procedimentos são rígidos, porém não são agentes que dificultam a execução do projeto.

Dessa forma, concluiu-se que a política de planejamento de projeto de software da empresa é bem aceita pelos gerentes de software. Na visão dos respondentes isso não é um empecilho para os projetos de software.

TABELA 15
Fontes de informação utilizadas durante o planejamento

Fontes de Informação	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Os requisitos não técnicos ⁴⁸	5	100	10	71
Os requisitos técnicos (requisitos funcionais ⁴⁹ e não-funcionais ⁵⁰)	4	80	11	79
O plano, as estimativas documentadas ⁵¹ e os riscos dos projetos anteriores	5	100	11	79
Relatórios de acompanhamento e encerramento dos projetos anteriores	4	80	10	71
A experiência do líder do projeto e da equipe de desenvolvimento	5	100	13	93
Outras (especifique)	-	-	1	7
Resposta múltipla, não soma 100%				

Com base nos dados apresentados na TAB. 15, identificou-se que, sob a perspectiva dos gerentes seniores, as fontes de informação mais utilizadas durante o planejamento de software são:

- Em primeiro lugar, englobando 100% das respostas, estão os requisitos não técnicos, a experiência do líder do projeto e da equipe de desenvolvimento e as informações referentes a projetos anteriores, tais como o plano e as estimativas documentadas;
- Em segundo lugar, representando 80% das respostas, estão os requisitos técnicos e os relatórios de acompanhamento e encerramento dos projetos anteriores.

Por outro lado, identificou-se que a maioria (93%) dos líderes de projeto utilizam sua experiência e também da equipe de desenvolvimento como fonte de informação durante o planejamento do projeto de software. Em segundo lugar, com percentuais iguais (79%), aparecem os requisitos técnicos e as informações referentes a projetos anteriores, tais como plano e as estimativas documentadas. Em

⁴⁸ Prazo, custo, pessoal, fatores políticos e organizacionais.

⁴⁹ Descrição das diversas operações que clientes e usuários esperam que sejam realizadas pelo software.

⁵⁰ Determinam as características desejáveis do software quanto à usabilidade, confiabilidade, desempenho, suportabilidade, bem como descrevem restrições de projeto e conformidades legais.

⁵¹ Tamanho, esforço, custo e cronograma.

seguida, surgem os requisitos não técnicos e os relatórios de acompanhamento e encerramento dos projetos anteriores.

Com base nesses dados, constatou-se que as fontes de informações são selecionadas de acordo com as necessidades dos grupos pesquisados.

Os gerentes seniores trabalham mais com fontes de informações estratégicas, como por exemplo, prazo, custo, pessoal, fatores políticos e organizacionais. Além disso, observou-se que eles preferem o formato mais sintético, isto é usam a experiência do líder do projeto e as estimativas documentadas. Com relação aos líderes de projeto, eles buscam mais as fontes de informações técnicas e operacionais.

É importante ressaltar o alto índice na utilização dos planos, estimativas, riscos e relatórios de acompanhamento e encerramento dos projetos anteriores. Isto significa que o repositório de conhecimento organizacional referente ao planejamento de projeto é importante para os líderes de projeto e gerentes seniores.

Com relação às estimativas, aos riscos do projeto, às atividades e aos compromissos firmados, 100% dos respondentes afirmaram que disseminam essas informações para as partes interessadas. Tais informações são formalizadas em documentos de planejamento do projeto de software, como por exemplo, estimativas documentadas, cronograma, plano do projeto e lista de riscos.

Com base nesses dados de formalização das informações, constatou-se a ocorrência da externalização, ou seja, elaboração do conhecimento explícito baseado no conhecimento tácito compartilhado nas reuniões entre cliente e equipe de especificação de requisitos.

Com relação às dificuldades na utilização das estimativas como fonte de informação (TAB. 16) para planejar as atividades e compromissos, um total de 80% dos respondentes consideram que a principal dificuldade é a pressão excessiva sobre o prazo de entrega do produto. Por outro lado, 29% dos líderes de projeto apontam como dificuldade o fato do projeto envolver novas tecnologias e 21% consideram a influência política (influência do cliente, dos níveis hierárquicos superiores) no projeto como uma dificuldade.

TABELA 16
Dificuldades para utilização das estimativas no planejamento das atividades e compromissos do projeto de software

Dificuldades	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Projeto envolvendo o uso de novas tecnologias	3	60	4	29
Inadequada transferência de tecnologia para o projeto	1	20	0	-
Padrões, políticas e metodologias de engenharia de software inadequados	1	20	0	-
Métodos e ferramentas de processamento das informações inadequados	2	40	0	-
Influência política no projeto	2	20	3	21
Pressão excessiva de prazo	4	80	12	86
Resposta múltipla, não soma 100%				

Ao contrário dos gerentes seniores, nenhum dos líderes de projeto considerou como dificuldades os itens abaixo:

- Inadequada transferência de tecnologia para o projeto;
- Padrões, políticas e metodologias de engenharia de software inadequadas;
- Métodos e ferramentas de processamento das informações inadequados;

Com isto constatou-se que, sob a perspectiva dos líderes de projeto, as principais dificuldades não são de ordem técnica e sim de administração dos compromissos de prazos estabelecidos com o cliente.

É importante salientar ainda que essas dificuldades incidem também sobre a combinação, conversão do conhecimento explícito em implícito. Isto porque a combinação ocorre no planejamento do projeto, conforme descrito no capítulo 5, quando as estimativas são utilizadas na elaboração do plano do projeto.

Isto posto, a análise dos dados evidenciou que, durante o planejamento do projeto de software, os líderes de projeto praticam a gestão da informação, principalmente porque tem necessidades de informações relacionadas ao projeto e ao produto a ser desenvolvido. Além de utilizar tais informações, eles geram outras no momento em que elaboram artefatos como o plano do projeto, o cronograma e as planilhas de estimativas de esforço, tamanho e recursos. Vale a pena ressaltar que os gerentes seniores trabalham mais com as fontes de informações estratégicas. Em contra partida, os líderes de projeto buscam mais as fontes de informações técnicas e operacionais.

Porém, para exercer tais atividades, esses profissionais apontaram como principais dificuldades:

- A pressão excessiva sobre o prazo de entrega do produto;
- A influência política (influência do cliente, dos níveis hierárquicos superiores) sobre o projeto;
- A falta de informações consistentes sobre a aplicação de novas tecnologias.

Enfim, sobre o conhecimento envolvido no planejamento do projeto de software, constatou-se que a criação do conhecimento se dá principalmente por meio da socialização e da externalização. Outro ponto a ser observado é o índice considerável no uso da base de conhecimento explícito sobre os projetos anteriores.

6.5. Área-Chave de Processo Acompanhamento e Supervisão de Projeto de Software

Os dados referentes à área-chave de processo acompanhamento e supervisão de projeto de software, obtidos por meio da 5ª parte do questionário e complementados pela análise de documentos, identificam as práticas e dificuldades na aplicação das diretrizes desta ACP relacionadas com a gestão da informação.

Quanto aos padrões e procedimentos definidos pela organização para acompanhamento e supervisão do projeto de software, 100% dos gerentes seniores e 79% dos líderes de projeto afirmaram que são flexíveis de acordo com as características do projeto, o que facilita a execução do projeto. Por outro lado identificou-se divergência de idéias, quando 14% dos líderes de projeto responderam que os padrões e procedimentos são rígidos, porém não são considerados obstáculos para execução do projeto. Os outros 7% dos líderes de projeto afirmaram que os padrões e procedimentos são rígidos e dificultam a execução do projeto de software.

Dessa forma, concluiu-se que a política de acompanhamento e supervisão de projeto de software da empresa é aceita com restrições pelos gerentes de software. Além disso, ela é considerada por 21% dos líderes de projetos respondentes como uma dificuldade para a execução dos projetos de software.

TABELA 17
Fontes de informação utilizadas durante o acompanhamento e supervisão do projeto de software

Fontes de Informação	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Reunião presencial ou via áudio/vídeo-conferência com a equipe do projeto	5	100	13	93
As análises comparativas entre os resultados reais do projeto e as estimativas geradas no planejamento	4	80	14	100
Os riscos do projeto de software relacionados	4	80	10	71
Informações de acompanhamento e supervisão de projetos anteriores	1	20	3	21
Outros (especifique)	-	-	-	-

Resposta múltipla, não soma 100%

Os dados apresentados na TAB. 17, mostram que, sob a perspectiva dos gerentes seniores, as fontes de informação mais utilizadas durante a supervisão do projeto de software são:

- Em primeiro lugar, a reunião presencial ou via áudio/vídeo-conferência com a equipe do projeto, englobando 100% das respostas;
- Em segundo lugar, representando 80% das respostas, estão as análises comparativas entre o estimado e o realizado e os riscos relacionados à execução do projeto.

Por outro lado, identificou-se que todos os líderes de projeto utilizam as análises comparativas entre os resultados reais do projeto e as estimativas geradas no planejamento como fonte de informação. Em segundo lugar, com percentual de 93%, são utilizadas as reuniões presenciais ou via áudio/vídeo-conferência com a equipe do projeto. Em seguida, com um percentual de 71% são utilizados os riscos relacionados na fase de planejamento do projeto de software.

Com relação aos modos de conversão do conhecimento postulados por Nonaka & Takeuchi (1997), os dados coletados evidenciam, nos dois grupos pesquisados, a ocorrência da socialização, compartilhamento do conhecimento tácito, e da internalização, elaboração do conhecimento tácito a partir do conhecimento explícito.

No caso dos gerentes seniores, a incidência maior é da socialização. Isto ocorre porque as fontes de informação mais utilizadas por este grupo são as reuniões presenciais ou à distância via áudio/vídeo-conferência com a equipe do projeto. Em seguida, são utilizadas as análises comparativas entre o estimado e o realizado e os riscos do projeto, necessariamente nesta ordem.

Porém, no caso dos líderes de projeto, a incidência maior é da internalização. Isto ocorre porque as fontes informacionais mais utilizadas são as análises comparativas entre os resultados reais do projeto e as estimativas geradas no planejamento.

É importante ressaltar ainda que os grupos de gerentes seniores e líderes de projetos utilizam como fonte de informação a base de conhecimento organizacional sobre acompanhamento e supervisão de projetos, em percentuais quase idênticos, 20% e 21% respectivamente. Com base nesses dados, constatou-se que, embora os níveis de atuação dos gerentes seniores e líderes de projeto serem diferenciados, as fontes de informações selecionadas por eles são praticamente as mesmas.

Com relação às informações de acompanhamento do projeto de software, os dados mostram que 100% dos respondentes, gerentes seniores e líderes de projeto, afirmaram que essas informações são documentadas, inclusive aquelas relacionadas às ações corretivas tomadas quando os resultados reais diferem significativamente do planejamento. Sendo que a única variante é a frequência com que as informações são disseminadas para as partes interessadas. De acordo com esse percentual, verificou-se que a externalização, conversão do conhecimento tácito em explícito, ocorre de forma intensa nos dois grupos pesquisados.

TABELA 18
Frequência da disseminação das informações de acompanhamento do projeto de software

Frequência da Disseminação da Informação	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Semanalmente	3	60	3	21
Quinzenalmente	1	20	8	57
Mensalmente	1	20	3	21
Outras (especifique)	2	40	3	21

Resposta múltipla, não soma 100%

De acordo com a TAB. 18, identificou-se que 40% dos gerentes seniores e 21% dos líderes de projeto disseminam as informações de acompanhamento e supervisão do projeto conforme descrito abaixo:

- A cada reunião de acompanhamento, geralmente quinzenalmente;
- Eventualmente nas reuniões de acompanhamento e nos marcos predeterminados do projeto;
- Quando ocorrem desvios;

- Sempre que necessário;
- Eventual, se necessário.

Os dados da TAB. 19 mostram que 100% dos líderes de projeto realizam avaliações formais em conjunto com suas equipes ao final do projeto. Nessas avaliações, as informações referentes às falhas e aos acertos ocorridos durante a execução do projeto são registradas. Esse resultado indica que a incidência de externalização no encerramento do projeto é alta.

TABELA 19
Avaliação das falhas e acertos do projeto de software

Avaliação da falhas e acertos	Gerente Sênior (n=5)		Líder de Projeto (n=12)	
	n	%	n	%
Sim, por meio de reuniões formais (as informações são registradas)	4	80	100	-
Sim, por meio de reuniões informais (as informações NÃO são registradas)	-	-	-	-
Não	1	20	-	-

De modo geral, a análise dos dados mostrou que, durante o processo de acompanhamento e supervisão do projeto de software, os gerentes de software praticam a gestão da informação, principalmente porque eles necessitam de informações relacionadas ao andamento das atividades. É importante ressaltar que os dois grupos de gerentes trabalham praticamente com as mesmas fontes de informações, embora possuam níveis de atuação diferenciados.

Porém, para exercer tais atividades, esses profissionais apontaram como principal dificuldade a rigidez da política organizacional de acompanhamento e supervisão de projeto de software.

Por fim, com relação ao conhecimento envolvido no planejamento do projeto de software, constatou-se que a criação do conhecimento se dá principalmente por meio da socialização, externalização e a internalização.

6.6. Área-Chave de Processo Garantia da Qualidade de Software

Os dados referentes à área-chave de processo garantia da qualidade de software foram coletados pelo questionário e complementados pela análise de documentos. A análise desses dados identificam as práticas e dificuldades na aplicação das diretrizes desta ACP relacionadas com a gestão da informação.

Quanto aos padrões e procedimentos definidos pela organização para garantia da qualidade de software, 60% dos gerentes seniores e 57% dos líderes de projeto afirmaram que são flexíveis de acordo com as características do projeto. Por outro lado, 40% dos gerentes seniores e 43% dos líderes de projeto responderam que os padrões e procedimentos são rígidos, porém eles não dificultam a execução do projeto.

Dessa forma, concluiu-se que a política de garantia da qualidade de software da empresa é bem aceita nos dois grupos pesquisados.

TABELA 20
Fontes de informação utilizadas nos processos de garantia da qualidade de software

Fontes de Informação	Gerente Sênior (n=5)		Líder de Projeto (n=13)	
	n	%	n	%
A revisão do plano, padrões e procedimentos e das atividades de engenharia de software do projeto de desenvolvimento de software	4	80	11	85
A auditoria sobre os artefatos de software	5	100	10	77
Outras (especifique)	-	-	-	-

Resposta múltipla, não soma 100%

Com base na análise dos dados apresentados na TAB. 20, identificou-se que a revisão e a auditoria são fontes de informação com similar importância para ambos os grupos pesquisados.

TABELA 21

Dificuldades para executar as revisões e auditorias de qualidade do software

Dificuldades	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Equipe de desenvolvimento não consciente da importância da qualidade do software	-	-	1	7
Falha na comunicação entre os consultores de qualidade e os membros da equipe de desenvolvimento	1	20	1	7
Membros da equipe de desenvolvimento e/ou consultores de qualidade treinados inadequadamente	-	-	1	7
Ausência ou número insuficiente de membros da equipe com perfil de consultor de qualidade de software	3	60	5	36
Pressão excessiva de prazo	4	80	10	71
Outros (especifique)	1	20	-	-

Resposta múltipla, não soma 100%

Com relação às dificuldades para executar as revisões e auditorias de qualidade do software, identificou-se, em ambos os grupos, que a principal dificuldade é a pressão excessiva sobre o prazo de entrega do produto final, conforme pode ser visto na TAB. 21. Outro problema detectado foi o número insuficiente de membros da equipe com perfil de consultor de GQS. Tal perfil, conforme descrito no capítulo 5, compreende profissionais com habilidades para desenvolver o planejamento, supervisão, registro e análise das atividades de auditoria. Vale a pena ressaltar ainda que um dos gerentes seniores explicitou o problema da dedicação parcial dos consultores de GQS, o que prejudica as atividades de qualidade.

A respeito das medições da qualidade de software, a análise dos dados mostrou que 100% dos respondentes, gerentes seniores e líderes de projeto, afirmaram que as medições são efetuadas. Com relação à disseminação dessas informações, 100% dos gerentes seniores e 79% dos líderes de projeto responderam que elas são disseminadas mensalmente para as partes interessadas, conforme TAB. 22.

TABELA 22

Frequência da disseminação das medições referentes à qualidade de software

Frequência da Disseminação das Medições	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Semanalmente	1	20	1	7
Quinzenalmente	-	-	-	-
Mensalmente	5	100	11	79
Outras	-	-	2	14

Resposta múltipla, não soma 100%

TABELA 23

Uso das medições de qualidade como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe

As informações referentes às medições realizadas são usadas como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe?	Gerente Sênior (n=4)		Líder de Projeto (n=13)	
	n	%	n	%
Sim (Como?)	4	100	12	92
Não	-	-	1	8

De acordo com a TAB. 23, identificou-se que 100% dos gerentes seniores e 92% dos líderes de projeto responderam que as medições de qualidade são utilizadas como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe. Segue abaixo a especificação de como essas informações são utilizadas:

- No replanejamento do projeto ou planejamento de novo projeto;
- Para fornecer um feedback para as partes interessadas;
- Em reuniões formais com a equipe e a gerência sênior como subsídio para se estabelecer as ações corretivas;
- Em reuniões com o gerente sênior, com os líderes de projeto e com as equipes de desenvolvimento.
- Para alimentar a base histórica.
- No acompanhamento das medições e análise dos artefatos/práticas que apresentam maior frequência de problemas. De acordo com os

resultados, ações são tomadas para melhoria dos indicadores localizados;

- Na análise crítica dos resultados e como aprendizado para novos projetos;
- Em reuniões de análise com participação da equipe;
- Para indicar melhorias no relatório mensal de qualidade;
- Para identificar os pontos que apresentam um maior número de ocorrências, alertando a equipe para o fato.

Com base nas descrições de como as medições são utilizadas, constatou-se que a área-chave de processo garantia da qualidade de software é o agente impulsionador do comportamento adaptativo do ciclo de gerenciamento da informação (FIG. 4) e o supervisor da internalização do processo de software.

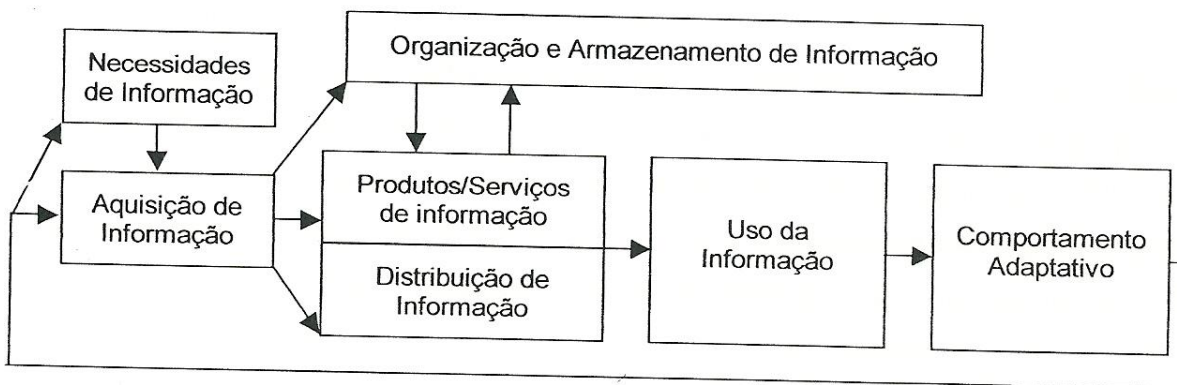


FIGURA 4 – Ciclo de Gerenciamento da Informação

Fonte – Choo , 1998a, p. 24

Assim, a análise dos dados mostrou que, durante o processo de garantia da qualidade de software, a gestão das informações sobre esse processo é praticada. Sendo que isso ocorre principalmente durante as auditorias e revisões, quando essas informações são registradas e disseminadas de forma sistemática para os vários níveis de gerência da empresa. Essas informações são utilizadas de maneiras diversificadas, mas sempre como suporte à melhoria da gestão do projeto de software e ao aperfeiçoamento e à aprendizagem da equipe.

No entanto, para gerenciar essas informações foram apontadas como principais dificuldades:

- A pressão excessiva sobre o prazo de entrega do produto;
- Número insuficiente de profissionais com perfil de consultores de GQS;
- A dedicação parcial dos consultores de GQS.

6.7. Área-Chave de Processo Gestão de Configuração de Software

Os dados referentes à área-chave de processo gestão de configuração de software, obtidos por meio da 7ª parte do questionário e complementados pela análise de documentos, identificam as práticas e dificuldades na aplicação das diretrizes desta ACP relacionadas com a gestão da informação.

Quanto aos padrões e procedimentos definidos pela organização para gestão de configuração de software, 60% dos gerentes seniores e 57% dos líderes de projeto afirmaram que são flexíveis de acordo com as características do projeto, o que facilita a sua execução. Por outro lado, 20 % dos gerentes seniores e 36% dos líderes de projeto responderam que os padrões e procedimentos são rígidos, porém não prejudicam a execução do projeto. Ademais, 20% dos gerentes seniores e 7% dos líderes de projeto afirmaram que os padrões e procedimentos são rígidos e dificultam a execução do projeto de software.

Dessa forma, concluiu-se que a política de gestão de configuração de software da empresa é aceita com restrições pelos gerentes de software, uma vez que é considerada por 27% dos respondentes como um obstáculo para os projetos de software.

Sobre a organização e armazenamento das informações, 100% dos respondentes afirmaram que há um sistema de repositório de gerenciamento de configurações onde são armazenados os produtos (artefatos técnicos ou de gestão) do projeto de software.

TABELA 24
Acesso ao repositório de configuração de software

Tipo de acesso	Líder de Projeto (n=14)	
	n	%
O acesso é liberado com restrições para todas as equipes de projeto do pólo de desenvolvimento	6	57
O acesso é concedido sem restrições para toda a equipe do projeto	2	14
O acesso é concedido para os membros da equipe de acordo com as atividades que eles executam	4	29
Outros (especifique)	-	-

Quanto ao acesso às informações contidas no repositório de gerenciamento de configuração de software, os dados da TAB. 24 mostram que 57% dos respondentes afirmaram que todas as equipes de projeto do pólo de desenvolvimento acessam ao repositório, porém as permissões⁵² ao conteúdo são concedidas com restrições. Do total, 43% dos respondentes afirmaram que o acesso é restrito à equipe do projeto, sendo que, desse percentual, 29% responderam que o acesso dentro da equipe do projeto é diferenciado de acordo com as atividades executadas.

TABELA 25
Dificuldades para controlar as mudanças dos produtos de software

Dificuldades	Gerente Sênior (n=5)		Líder de Projeto (n=14)	
	n	%	n	%
Padrões, políticas e metodologias de gestão de configuração de software adequados	-	-	1	7
Infra-estrutura e ferramentas de gestão de configuração de software inadequados	3	60	5	36
Influência política no projeto	-	-	3	21
Mudanças de requisitos	2	40	3	21
Pressão excessiva de prazo	4	80	9	64
Membros da equipe de desenvolvimento treinados inadequadamente	-	-	2	14
Membros da equipe de desenvolvimento inexperientes	1	20	3	21
Ausência ou número insuficiente de membros da equipe de desenvolvimento com perfil de gestor de configuração de software	1	20	4	29
Outros (especifique)	-	-	-	-

Resposta múltipla, não soma 100%

De acordo com a TAB. 25, verificou-se que, para ambos os grupos, o principal empecilho para controlar as mudanças de configuração é a pressão excessiva sobre o prazo de entrega do produto. Em seguida, aparece a inadequação da infra-estrutura e das ferramentas de gestão de configuração de software. Esses dados apontam uma problemática de infra-estrutura tecnológica.

Outra dificuldade detectada relaciona-se com as habilidades dos membros da equipe. Essa dificuldade tem relevância semelhante à problemática da pressão excessiva do prazo de entrega do produto. Isto porque se somados os percentuais para os itens abaixo descritos, tem-se um índice de (64%):

- Membros da equipe de desenvolvimento de software treinados inadequadamente;
- Membros da equipe de desenvolvimento de software inexperientes;
- Número insuficiente de membros da equipe de desenvolvimento com perfil de gestor de configuração de software.

Além do plano de gestão de configuração, as atividades de gestão de configuração de software geram os seguintes produtos de informação: o repositório de armazenamento dos itens de configuração do projeto, as solicitações de modificação de software, os relatórios e ordens de modificação de engenharia.

Quanto aos produtos de informação gerados pelas atividades de configuração de software, os resultados do estudo indicam que para todo projeto de software, além do repositório, é desenvolvido um plano de gestão de configuração de software, fundamentado no plano do projeto e no cronograma. Sendo assim, a elaboração deste documento indica a ocorrência da combinação, que é a conversão de conhecimento explícito em um novo conhecimento explícito.

Para manter a organização e a confiabilidade das informações do repositório, são realizadas auditorias de configuração. No caso de mudanças, são geradas solicitações de mudanças que registram as alterações necessárias.

⁵² Privilégios de leitura, inclusão, alteração, exclusão sobre as informações contidas no repositório.

De modo geral, a análise dos dados mostrou que, durante a gestão de configuração de software, a gestão da informação é praticada quando são realizadas atividades de organização e armazenamento das informações do projeto. Sendo que as principais dificuldades para gerenciar tais informações são:

- A inadequação da infra-estrutura de ferramentas de gestão de configuração de software;
- A pressão excessiva sobre o prazo de entrega do produto.

Por fim, com relação ao conhecimento envolvido na gestão de configuração de software, constatou-se que a criação do conhecimento se dá principalmente por meio da externalização e combinação.

7. CONSIDERAÇÕES FINAIS

As empresas de software inserem-se em um mercado dinâmico, marcado por uma constante evolução das tecnologias, técnicas e métodos de desenvolvimento. Além disso, enfrentam a concorrência em novos produtos e serviços e a pressão por qualidade e prazos de entrega cada vez mais reduzidos.

Sob esta perspectiva, a gestão da informação é um fator importante no processo de maturação de empresas de software, pois o uso da informação gerencial e/ou tecnológica é determinante para criar conhecimento sobre o produto, o processo e o projeto de software.

O conhecimento adquirido com as informações provenientes de projetos torna-se fundamental para aumentar a produtividade, reduzir custos, aumentar a qualidade do produto e a satisfação dos clientes.

Além disso, o gerenciamento do processo de desenvolvimento de software faz uso freqüente de parâmetros quantitativos provenientes da análise das atividades e da realização de medições de performance. Dessa forma, o histórico destas informações também representa conhecimento explícito que é útil para a realização de estimativas e dimensionamento de esforços para o desenvolvimento de novos produtos.

No entanto, embora a importância da informação no contexto do processo de desenvolvimento de software seja reconhecida, a prática da sua gestão, principalmente com fins de criar conhecimento, ainda apresenta dificuldades operacionais e culturais. Portanto, este trabalho consiste em um passo inicial na investigação das práticas de gestão da informação realizadas pelos profissionais de software, bem como das principais dificuldades encontradas durante o exercício dessas.

Neste contexto, inicialmente este estudo procurou estabelecer uma correlação entre o processo de software fundamentado no nível 2 do CMM, a gestão

da informação (Choo, 1998) e os modos de conversão do conhecimento postulados por Choo (1998a) e Nonaka & Takeuchi (1997), respectivamente. Para isso, foi apresentado no capítulo 5 como as atividades de gestão da informação ocorrem durante o processo de software, sob a perspectiva da gestão de requisitos, da gestão de projeto, da gestão de configuração de software, da garantia da qualidade de software e da gestão de subcontratação.

Em seguida, procurou-se identificar as práticas e dificuldades na aplicação das diretrizes especificadas no nível 2 do CMM que se relacionem com a gestão da informação. Para isso, utilizou-se os dados advindos da pesquisa de campo que propiciaram elaborar algumas conclusões que são apresentadas, de forma sintética, a seguir.

Na gestão de requisitos, caracterizada pela definição, documentação e controle dos requisitos de software, constatou-se que os profissionais de software praticam a gestão da informação, principalmente porque realizam atividades de levantamento, análise e documentação de requisitos. Porém, para exercer tais atividades, esses profissionais apontaram como principais dificuldades: número insuficiente de membros da equipe com perfil de analista de requisitos; os clientes/usuários não terem a consciência da importância em formalizar os requisitos de software; as mudanças contínuas de requisitos. Outro ponto a ser observado é o baixo índice de uso da base histórica de requisitos de software, o que dificulta a criação de novos conhecimentos. Porém, isso pode ser explicado em virtude do processo ser incipiente em algumas unidades pesquisadas e, por conseguinte, a base histórica ainda não contém informações suficientes.

No que diz respeito ao planejamento do projeto de software, apurou-se que os líderes de projeto praticam a gestão da informação, principalmente porque eles tem necessidades de informações relacionadas ao projeto e ao produto a ser desenvolvido. Além de utilizar tais informações, eles também geram outras no momento em que elaboram artefatos como o plano do projeto, o cronograma e as planilhas de estimativas de esforço, tamanho e recursos. Porém, para exercer a gestão dessas informações, esses profissionais apontaram como principais dificuldades: a influência política (influência do cliente, dos níveis hierárquicos

superiores) sobre o projeto; a falta de informações consistente sobre a aplicação de novas tecnologias.

No acompanhamento e supervisão do projeto de software, apurou-se que os gerentes de software praticam a gestão da informação, principalmente porque eles necessitam de informações relacionadas ao andamento das atividades. Sendo que a principal dificuldade apontada pelos profissionais é a rigidez da política organizacional de acompanhamento e supervisão de projeto de software.

No que tange ao processo de garantia da qualidade de software, verificou-se que a gestão das informações referentes a esse processo é praticada, principalmente durante as auditorias e revisões, quando as informações são registradas e disseminadas, de forma sistemática, para os vários níveis de gerência da empresa. Tais informações são utilizadas de forma diversificada, mas sempre como suporte à melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe. No entanto, para gerenciar essas informações foram apontadas como principais dificuldades: o número insuficiente de profissionais com perfil de consultores de GQS; a dedicação parcial dos consultores de GQS.

Na gestão de configuração de software, caracterizada pela organização, armazenamento e controle de mudanças dos itens contidos no repositório de configuração, verificou-se que a gestão da informação é praticada quando são realizadas atividades relacionadas à organização e armazenamento das informações do projeto. Sendo que a inadequação da infra-estrutura de ferramentas de gestão de configuração de software é uma dificuldade para se desenvolver as atividades de GCS;

Em síntese, identificou-se dois modos de conversão do conhecimento que ocorrem com maior incidência durante o processo de software com ênfase no CMM nível 2:

- A socialização que é decorrente da própria natureza da atividade de desenvolvimento de software que envolve principalmente conhecimento tácito, tanto sobre o negócio como sobre a tecnologia envolvida.

- A externalização, que é a transformação do conhecimento tácito em explícito, é proveniente da ênfase que o CMM dá ao registro formal das informações relacionadas ao processo de software, sejam essas gerenciais ou técnicas.

Com relação as dificuldades identificadas, verificou-se que a pressão excessiva sobre o prazo de entrega do produto é uma constante em todas as áreas-chaves de processo pesquisadas. Esta pressão, uma vez não considerada no planejamento do projeto de software, dificulta o processo de externalização do conhecimento tácito adquirido. Este fato nos remete a seguinte afirmativa de Humphrey (*apud* Paulk, 1993): "a razão fundamental para se focar em gestão de projeto, antes de se implementar o processo de engenharia, é o fato de que sem a disciplina gerencial o processo de engenharia é prejudicado pelas pressões de prazo e custo." No entanto, para se obter a disciplina gerencial é necessário mudanças de comportamento por parte dos gerentes de software e, sobretudo, da alta gerência que precisa arcar com o ônus inicial da implantação de um processo de software disciplinado por um modelo de qualidade. Isto porque, se os condutores da organização não legitimam as idéias a serem implementadas, na prática o que ocorre é o descrédito dos desenvolvedores de software que não vão desperdiçar tempo e forças em atividades que não agregam valor ao processo e ao produto de software que precisa ser desenvolvido.

Também constatou-se que em todas as atividades guarda-chuva há a incidência de pelo menos um dos processos de gestão da informação definido por Choo (1998a). Dessa forma, o modelo de gerenciamento de Choo pode ser utilizado para que seja possível realizar uma interpretação das diretrizes preconizadas no CMM nível 2 sob a perspectiva do uso estratégico das informações geradas e registradas pelos profissionais de software em cada uma das áreas-chaves de processo. Dessa forma, as idéias postuladas por Choo podem ser utilizadas para potencializar as atividades de gestão do projeto de software.

Outro fator importante é que para se obter ganhos de produtividade e qualidade, é necessário dispor de profissionais com habilidades para gerenciar informações do produto (analista de requisitos) e do projeto de software (líder de

projeto), bem como de profissionais que dêem apoio à equipe de desenvolvimento no que tange as questões de melhoria do processo de software (consultores de GQS e Grupos de Processo e Engenharia de Software). Isto porque uma vez que esses profissionais trabalhem em conjunto e em sintonia com os demais membros da equipe de desenvolvimento, a internalização do processo de desenvolvimento de software ocorre de forma mais rápida e eficiente.

Neste contexto, entende-se que, para se obter sucesso na condução dos processos de gestão da informação, é necessário que aqueles que conduzem o projeto de software firmem compromissos de forma que o processo de criação de conhecimento não fique prejudicado. Além disso, é necessário que a alta gerência reconheça a importância de se adotar mecanismos que fomentem a inovação (criação de conhecimento), a organização dos processos e dos produtos (conhecimento) que são gerados ao longo do ciclo de vida do software.

Espera-se que a partir dos resultados deste trabalho, surjam alguns caminhos para o desenvolvimento de novas pesquisas que venham ampliar este estudo. São apresentadas a seguir algumas sugestões de projetos que poderão contribuir para a condução de trabalhos futuros:

1. Identificar mecanismos que impulsionem os processos de socialização, externalização, combinação e internalização, sob a perspectiva de cada uma das atividades guarda-chuva do processo de software;
2. Aprofundar o estudo da gestão das informações referentes às estimativas de custo e prazo para a execução de um projeto de software;
3. Detalhar uma proposta de arquitetura tecnológica que integre as inúmeras informações de planejamento e acompanhamento do projeto de software, de forma a facilitar o trabalho dos líderes de projeto;
4. Identificar as razões da pouca utilização da base de conhecimento de requisitos como fonte de informação em novos projetos de software;

5. Ampliar o estudo para os demais níveis de maturidade desenvolvidos no modelo CMM, de forma a abordar problemáticas de gestão das informações relacionadas ao produto de software.

Finalmente, espera-se que os resultados alcançados nesta pesquisa possam servir de ponto de partida para o estudo e desenvolvimento de novos mecanismos que impulsionem os modos de conversão do conhecimento identificados em cada uma das atividades guarda-chuva do processo de software. Espera-se também que as práticas de gestão da informação identificadas possam ser utilizadas como instrumentos operacionais de orientação para as empresas brasileiras de software, afim de que essas ampliem a sua competitividade.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALMEIDA, Napoleão Mendes de. **Gramática Metódica da Língua Portuguesa**. 40ed. São Paulo: Editora Saraiva, 1995.
- BASKERVILLE, Richard. PRIES-HEJE, Jan. **Knowledge capability and maturity in software management**. The DATA BASE for Advances in Information Systems, v. 30, n.2, p. 26-43, spring 2002.
- BELLOQUIM, Átila. MUNCK, Glauco. **Processo de Desenvolvimento na Perspectiva do Projeto de Modernização**. Boletim Técnico 3 publicado no PmoD, 2003. Disponível em <http://www.intra.serpro.publicacao/pmod/index.htm>
- _____, Átila. **Acompanhamento e Supervisão de Projetos**. Boletim Técnico 6 publicado no PmoD, 2003. Disponível em <http://www.intra.serpro.publicacao/pmod/index.htm>
- BENTO, A. M. **Systems analysis: A decision approach**. Information & Management, n.27, p.185-194, 1994.
- BIRK, Andreas. DINGSØYR, Torgeir. STÁLHANE, Tor. **Postmortem: never leave a project without it**. IEEE Software, p. 43-45, may/june 2002.
- BOEHM, B., **Anchoring the Software Process**. IEEE Software, vol. 3, n.4, p. 73-82, July 1996.
- BORGES, Lígia Motta Silveira. FALBO, Ricardo de Almeida. **Gerência de Conhecimento sobre Processos de Software**. Anais do VIII WQS – Workshop de Qualidade de Software, Rio de Janeiro, 2001.
- BRASIL, Ministério da Ciência e Tecnologia. **Pesquisa de Qualidade e Produtividade no Setor de Software Brasileiro**. Disponível em: <http://www.mct.gov.br/sep/insi/Quali2001/Public2001.htm> Acesso em: 27/06/2003.
- CARVALHO, Isamir Machado de. **Liderança e Criação de Conhecimento: Aspectos relevantes da liderança para o processo de criação de conhecimento numa organização de alta tecnologia**. Dissertação de Mestrado. Universidade de Brasília, Brasília, 2002.
- CHOO, Chun Wei, **Information Management for Intelligent Organization: the art of scanning the environment**. Medford, NJ: Asis, 1995.

- CHOO, Chun Wei, **Knowing Organization**: how organizations use information to construct meaning, create knowledge and make decisions. New York, Oxford: Oxford University Press, 1998.
- COSTA, G. M.; Miranda, J.P.B. de. **Implantando Processos de Desenvolvimento de Software**. Developers Magazine, p. 28-30, set. 1999.
- DAVENPORT, Thomas H.; PRUSAK, Laurence. **Conhecimento Empresarial**: como as organizações gerenciam seu capital intelectual. Rio de Janeiro: Editora Campus, 1999.
- DINGSØYR, Torgeir. CONRADI, Reidar. **A survey of case studies of the use of knowledge management in software engineering**. International Journal of Software Engineering and Knowledge Engineering, vol. 12, n.4, p. 391-414, 2002.
- FIORINI, Soei T.; STAA, Arndt von; BAPTISTA, Renan Martins. **Engenharia de Software com CMM**. Rio de Janeiro: Brasport, 1998.
- FURNIVAL, Ariadne Cholê. **A participação dos usuários no desenvolvimento de sistemas de informação**. Ciência da Informação, v.25, n.2, p.1-13, 1995.
- GILB, Tom. **Principles of Software Engineering Management**, Addison-Wesley, Reading, MA, 1988.
- GUIMARÃES, Ângelo de Moura. LAGES, Newton Alberto de Castilho. **Algoritmos e Estrutura de Dados**. Rio de Janeiro: LTC, 1985.
- HERRING, James E. **Information management: the convergence of professions**. International Journal of Information Management, v.11, n.2, p.144-155, June 1991.
- HOUAISS, Antônio & Et al. **Dicionário Houaiss da Língua Portuguesa**. Rio de Janeiro: Editora Objetiva, 2001.
- HUMPHREY, Watts S. **A Discipline for Software Engineering**, Addison-Wesley Publishing Company, Reading, MA, 1995.
- HUMPHREY, Watts S. **Managing the Software Process**, Addison-Wesley, Reading, MA, 1989.
- KIM, Daniel H. **The Link between Individual and Organizational Learning**. Sloan Management Review Association, 1993.
- KOMI-SIRVIÖ, Seija. MÄNTYNIEMI, Annuka. **Toward a practical solution for capturing knowledge for software projects**. IEEE Software, p. 60-62, may/june 2002.

- LEONARD-BARTON, D. **Wellsprings of knowledge: Building and sustaining the sources of innovation**. Boston: Harvard Business School Press, 1995.
- LÜDKE, Menga. ANDRÉ, Marli E. D. A. **Pesquisa em Educação: abordagens qualitativas**. São Paulo: EPU, 1986.
- MARCHIORI, Patrícia Zeni. **A ciência e a gestão da informação: compatibilidades no espaço profissional**. Ciência da Informação, Brasília, v.31, n.2, p. 72-79, maio/ago. 2002.
- McGEE, James. PRUSAK, Laurence, **Gerenciamento estratégico da informação: aumente a competitividade e a eficiência da sua empresa utilizando a informação como ferramenta estratégica**. 8. ed. Rio de Janeiro: Campus, 1994.
- MARCONI, Mariana de Andrade; LAKATOS, Eva Maria. **Técnicas de Pesquisa**. 5. ed. São Paulo: Atlas, 2002.
- NETO, José Inácio Jaeger. BOCOLI, Fernanda Schimidt. **Successo^{SW} = CMM₂ + PMBOK**. PMI-RS Journal, n.5, p.2-11, maio 2003
- NEVIS, Edwin C.; DIBELLA, Anthony J.; GOULD, Janet M.. **Understanding Organizations as Learning Systems**. Sloan Management Review Association, 1995.
- NONAKA, Ikujiro; TAKEUCHI, Hirotaka. **Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação**. Rio de Janeiro: Campus, 1997.
- PADUAN, Roberta. **Tesouro escondido - a indústria brasileira de software vale ouro**. Revista Exame, São Paulo, 795. ed., 20/06/2003, Disponível em: http://exame.abril.com.br/edicoes/795/tecnologia/conteudo_18280.shtml. Acesso em: 23/06/2003.
- PAULA, Wilson de Pádua Filho. **Engenharia de Software: Fundamentos, Métodos e Padrões**, Rio de Janeiro, LTC, 2001.
- PAULK, M. C.; Curtis, Bill; Chrissis; Mary Beth; Weber, Charles V. **Capability Maturity Model for Software, version 1.1**; Software Engineering Institute, Carnegie Mellon University; CMU/SEI-93-TR-24, ESC-TR-93-177; February 1993. Disponível em: <<http://www.sei.cmu.edu>> Acesso em: 12/04/2003.
- PFLEEGER, Shari Lawrence. **Software Engineering : theory and practice**. 2. ed. Upper Saddle River, New Jersey: Prentice Hall, 2001.

- PRESSMAN, Roger S. **Engenharia de Software**. 5. ed. Rio de Janeiro: McGraw-Hill, 2002. 843p.
- QUEIROZ, Carlos Otávio de Alexandre. **Modelo de Gestão do Conhecimento para Empresas de Desenvolvimento de Software**. Dissertação de Mestrado do Curso de Pós-Graduação em Informática da UFPB. Campina Grande, 2001. 153p.
- ROWLEY, Jennifer. **Towards a framework of information management**. International Journal of Information Management. v.18, n. 5, p. 359-369, 1998.
- ROWLEY, Jeniffer. **What is information ?** Information Services & Use, v.18, n.3, p.243-254, 1998
- RALASUBRAMANIAM, Shivram. JAGADEESAN, Gokulakrishnan. **Knowledge Managemtn at Infosys**. IEEE Software, p. 53-55, may/june 2002.
- RAMESH, Balasubramaniam. **Process knowledge management with traceability**. IEEE Software, p. 50-52, may/june 2002.
- RUS, Iona. LINDVALL, Mikael. **Knowledge management in software engineering**. IEEE Software, p. 26-38, may/june 2002.
- SANTOS, Antonio Raimundo dos. **Metodologia Científica: a construção do conhecimento**. 5. ed. Revisada, Rio de Janeiro: DP&A, 2002.
- SEILHEIMER, Steven D. **Information during systems development: a model for improvement in productivity**. International Journal of Information Management. V.20, p. 287-295, 2000.
- SILVEIRA, Paola Gomes. **O gerenciamento da informação no contexto de projeto: estudo de práticas em empresas brasileiras**. Dissertação de Mestrado do Curso de Pós-Graduação em Ciência da Informação da UFMG. Belo Horizonte, 2002. 184p.
- TARGGAT, W. M. THARP, M. O. **A survey of information requirements analysis techniques**. ACM Computing Surveys, 9(4), 273-290, 1994.
- TAYLOR, Allan. FARREL, Stephen. **Information management in context**. Aslib Proceedings. V44, n.9, p. 319-322, september 1992.
- WEBER, Sérgio. WANGENHEIM, Christiane Gresse von. **Modelagem de processos de software: um pequeno guia**. Relatório Técnico QS002.03P, 26/02/2003. GrupoQS, Departamento de Informática e Estatística da UFSC.

ANEXO A – MENSAGENS AOS RESPONDENTES PARA PARTICIPAÇÃO

Prezado Senhor(a),

Meu nome é Ana Karla Chaves Melo, sou analista de desenvolvimento da SUNAT/ATBHE e mestranda em Ciência da Informação pela Universidade Federal de Minas Gerais - UFMG.

Minha pesquisa para dissertação de mestrado refere-se às práticas de gerenciamento da informação (aquisição, criação, organização, distribuição e uso da informação) com vistas a prover suporte à memória do produto e do projeto de software, bem como à aprendizagem e ao aperfeiçoamento dos gerentes e técnicos em uma empresa de tecnologia da informação que tenha como referencial o Modelo de Maturidade de Capacitação para Software - CMM.

Para fins de coleta de dados desta pesquisa, necessito da vossa colaboração no sentido de preencher o questionário anexo e retornar as respostas até o dia 05 de março de 2004. O questionário é de fácil compreensão, é estruturado de acordo com as áreas-chave do nível 2 do CMM, sendo que as questões são todas objetivas e o tempo médio previsto para respondê-las é de 20 minutos.

Gostaria de salientar que a coleta de dados não inclui informações sigilosas nem de cunho pessoal e que serão utilizadas tão somente para fins acadêmicos.

Sem mais para o momento, antecipo meus agradecimentos esperando poder contar com sua colaboração para o desenvolvimento e valorização deste tema.

Atenciosamente,

Ana Karla Chaves Melo

(31) 3489-6206

mestrado.ufmg@terra.com.br

ana-karla.melo@serpro.gov.br

ANEXO B – QUESTIONÁRIO

1. Informações Gerais

Este questionário foi desenvolvido com o objetivo de verificar quais as práticas e dificuldades no processo de gerenciamento da informação (aquisição, criação, organização, distribuição e uso da informação) com vistas a prover suporte à memória do produto e do projeto de software, bem como à aprendizagem e ao aperfeiçoamento dos gerentes e técnicos em uma empresa de tecnologia da informação que tenha como referencial o Modelo de Maturidade de Capacitação para Software - CMM.

O questionário está estruturado conforme descrito a seguir:

Partes	Objetivos Específicos	Questões
Parte 1- Perfil do respondente	Caracterizar o Gerente Sênior, o Líder do Projeto e a equipe de desenvolvimento	1-5
Parte 2 - Características dos projetos de software executados na área	Identificar os tipos de projetos de software executados pelas equipes de desenvolvimento	6-12
Parte 3 – Área-chave de processo Gestão de Requisitos	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>gestão de requisitos</i> relacionadas com a gestão da informação	13-22
Parte 4 - Área-chave de processo Planejamento do Projeto de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>planejamento do projeto de software</i> relacionadas com a gestão da informação.	23-29
Parte 5 - Área-chave de processo Acompanhamento e Supervisão do Projeto de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>acompanhamento e supervisão do projeto de software</i> relacionadas com a gestão da informação	30-37
Parte 6 Área-chave de processo Garantia da Qualidade de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>garantia da qualidade software</i> relacionadas com a gestão da informação	38-44
Parte 7 - Área-chave de processo Gestão de Configuração de Software	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>gestão de configuração de software</i> relacionadas com a gestão da informação	45-49
Parte 8 - Área-chave de processo Subcontratação de Software ¹	Verificar as práticas e dificuldades na aplicação das diretrizes da ACP <i>subcontratação de software</i> relacionadas com a gestão da informação	50-58

O público alvo deste questionário são os gerentes sênior, os líderes de projeto e os desenvolvedores que trabalham em projetos que implementam as práticas preconizadas nas áreas-chave de processo no nível 2 do CMM. Estas áreas-chave são: planejamento e acompanhamento do projeto de software, gestão de requisitos, garantia da qualidade de software, gestão de configuração de software e gestão de subcontratação de software.

Favor retornar as respostas até o dia 05 de março de 2004.

Desde já, obrigada pela atenção.

¹ Caso os projetos desenvolvidos pelo respondente não envolvam contratos com terceiros, as questões referentes a área-chave Subcontratação de Software não precisam ser respondidas.

2. Instruções de Preenchimento

1. A identificação do respondente é opcional.
2. Para algumas questões podem ser selecionadas mais de uma resposta.
3. As notas explicativas contidas nos enunciados das questões ou nas opções de resposta podem ser lidas posicionando o mouse sobre elas.
4. Para facilitar a navegação pelo questionário, selecione com o *mouse* o(s) campo(s) de resposta que estão sombreados ou utilize as teclas **<Page Down>** e **<Page Up>** para *avançar* e para *voltar*, respectivamente.
5. O tempo médio previsto para preencher este questionário é de 20 minutos.
6. As classificações das funções gerenciais e técnicas utilizadas estão de acordo com o modelo CMM e devem ser consideradas da seguinte forma:
 - Gerente Sênior – refere-se ao gerente geral do pólo de desenvolvimento, responsável pelo acompanhamento de todos os projetos de desenvolvimento e manutenção de software em execução.
 - Líder de Projeto – refere-se ao chefe de uma área funcional do pólo de desenvolvimento ou o profissional por ele delegado, que é o responsável pelo planejamento e acompanhamento da execução dos projetos de software sob sua gestão.
 - Equipe de Desenvolvimento – refere-se aos desenvolvedores de software, ou seja, os analistas ou técnicos de desenvolvimento.

3. Questões

Parte 1 – Perfil do Respondente

1. Empresa/Regional/Área funcional ou setor: []
Nome: [] (opcional)
E-mail: [] (opcional)
Sexo: () Feminino () Masculino
Idade: [] anos

2. Qual o seu grau acadêmico mais elevado?
() Técnico () Mestrado
() Graduação () Doutorado
() Especialização

3. Qual a sua formação acadêmica?
() Ciência da computação () Administração
() Engenharia () Outra (especifique) []
() Matemática () Curso técnico (especifique) []

4. Há quanto tempo executa atividades de desenvolvimento de software (inclusive atividade de apoio ao desenvolvimento, ou seja, planejamento e acompanhamento de projetos, configuração de software, melhoria do processo de software etc)
() até 3 anos () de 14 a 18 anos
() de 4 a 8 anos () mais de 20 anos
() de 9 a 13 anos

5. Qual o item abaixo descreve melhor o seu papel nos projetos em que você já participou?
() Gerente sênior () Analista de requisitos
() Líder de projeto () Membro da equipe de desenvolvimento
() Analista de negócios () Outro (especifique) []

Parte 2 – Características dos Projetos de Software Executados na Área

6. Quanto ao tamanho, a maioria dos projetos pode ser caracterizada como:
- Grande porte – acima de 150 homem-dia ou 100 pontos de função
 - Médio porte – até 150 homem-dia ou 100 pontos de função
 - Pequeno porte – até 30 homem-dia ou 20 pontos de função
7. Quais tipos de produtos de software são gerados na maioria dos projetos?
- Software básico (infra-estrutura de comunicação, sistema operacional etc).
 - Software de prateleira
 - Software aplicativo de gestão de informações estratégicas
 - Software aplicativo de gestão de informações gerenciais e operacionais
 - Software aplicativo para Web e/ou Internet
 - Sites estáticos onde NÃO há troca de informações com o usuário
 - Outros (especifique) []
8. Qual o modelo de análise mais utilizado?
- Estruturada
 - Essencial
 - Orientada a Objeto
 - Outros (especifique) []
9. Qual é a média de duração da maioria dos projetos?
- [] Semanas [] Meses
10. Qual a quantidade média de pessoas nas equipes alocadas aos projetos?
- [] Pessoas
11. Com relação à definição do papel a ser desempenhado pelos membros da equipe do projeto, pode-se afirmar que:
- Os papéis são bem definidos SEM acúmulo de atividades.
 - Os papéis são bem definidos, porém há acúmulo de atividades que NÃO sobrecarrega a equipe.
 - Os papéis são bem definidos, porém há acúmulo de atividades que sobrecarrega a equipe.
12. Qual a disponibilidade das pessoas alocadas para cada projeto em relação à carga horária delas?
- Tempo compartilhado
 - Tempo integral

Parte 3 – Área-Chave de Processo Gestão de Requisitos

Visa estabelecer um entendimento comum, entre o contratante e a equipe do projeto de software, sobre os requisitos de software² que deverão ser atendidos. Envolve o estabelecimento de compromissos, tanto técnicos quanto relativos a datas de entrega ou outras restrições, de modo que a equipe do projeto possa garantir a documentação, planos e atividades de acordo com os requisitos de software a serem implementados.

13. Os projetos seguem uma política organizacional escrita (normas), padrões e procedimentos para controlar os requisitos?
 Sim Não As vezes
 Caso a resposta seja não, vá para a questão 15.
14. Quanto aos padrões e procedimentos definidos pela organização para controlar requisitos, pode-se afirmar que são?
 Rígidos e entram a execução do projeto
 Rígidos, porém NÃO entram a execução do projeto
 Flexíveis de acordo com as características do projeto³, facilitando a execução do projeto
15. No levantamento de requisitos do projeto de software quais são as principais fontes de informações utilizadas?
 Reunião presencial com o cliente⁴/usuário⁵
 Reuniões via áudio ou vídeo-conferência com o cliente/usuário
 Questionários aplicados aos usuários
 Pesquisa documental (leis, políticas, normas, procedimentos etc)
 Prototipagem
 Os requisitos documentados de outros projetos similares, ou seja, com as mesmas características e contexto
 Observação do ambiente no qual o software será inserido
 Os requisitos são previamente definidos pelo cliente/usuário
 Outros (especifique) []
16. Quais são as maiores dificuldades encontradas durante o levantamento e documentação dos requisitos técnicos (requisitos funcionais⁶ e não-funcionais⁷) ou não técnicos⁸, bem como os critérios de aceitação que são utilizados para validar o software?
 Cliente/usuário não comprometidos com o projeto
 Conflitos entre o cliente/usuário e a organização desenvolvedora de software
 Cliente/usuário não consciente da importância da qualidade dos requisitos e do seu registro
 Falha na comunicação entre cliente/usuário e equipe desenvolvedora
 Pressão excessiva de prazo
 Membros da equipe de desenvolvimento treinados inadequadamente
 Membros da equipe de desenvolvimento inexperientes
 Ausência ou número insuficiente de membros da equipe de desenvolvimento com perfil de analista de requisitos
 Outros (especifique) []

² Característica intrínseca do software necessária para solucionar um problema do usuário ou atender um contrato, padrões, especificações ou outra documentação formal.

³ Natureza do produto de software, tecnologia utilizada, requisitos de prazos e custos.

⁴ Patrocinador do software.

⁵ Aquele que interage direta ou indiretamente com o software.

⁶ Descrição das diversas operações que clientes e usuários esperam que sejam realizadas pelo software.

⁷ Determinam as características desejáveis do software quanto à usabilidade, confiabilidade, desempenho, suportabilidade, bem como descrevem restrições de projeto e conformidades legais.

⁸ Prazo, custo, pessoal, fatores políticos e organizacionais.

17. Os requisitos são revisados, isto é, verifica-se se os requisitos técnicos são factíveis e apropriados para serem implementados por software, consistentes entre si e testáveis antes de serem incorporados ao projeto de software?
- Sim
 - Não (por quê?) []
 - As vezes (por quê?) []
- Caso a resposta seja não, vá para a questão 19.
18. Quem revisa os requisitos antes deles serem incorporados ao projeto de software?
- O cliente/usuário
 - A equipe de desenvolvimento
 - Outros (especifique) []
19. Quais são as maiores dificuldades encontradas para se ter os requisitos documentados como uma referência para uso nas atividades de engenharia e de gestão?
- Requisitos são definidos de forma inadequada
 - Requisitos não estão claros
 - Requisitos incorretos
 - Requisitos conflitantes
 - Mudanças contínuas dos requisitos
 - Outros (especifique) []
20. Se os requisitos mudam, são feitos os ajustes necessários nos planos de projeto, produtos e atividades?
- Sim
 - Não (por quê?) []
 - As vezes (por quê?) []
21. As atividades de gestão de requisitos são acompanhadas por meio de medições? (por exemplo: número total das mudanças dos requisitos que são originais, incluídos, alterados e excluídos).
- Sim Não As vezes
- Caso a resposta seja não, vá para a questão 23
22. As informações referentes às medições realizadas são usadas como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe?
- Sim (como?) []
 - Não (por quê?) []

Parte 4 – Área-Chave de Processo Planejamento do Projeto de Software

Seu objetivo é estabelecer planos razoáveis à correta execução das atividades de engenharia de software e para a gestão do projeto de software. Esta área-chave envolve a elaboração de estimativa para o trabalho a ser executado, a definição dos compromissos necessários e a definição de um plano de execução do trabalho. Estes itens são a base necessária para gerir o projeto de software, sem eles, uma gestão efetiva de projeto não pode ser implementada.

23. Os projetos seguem uma política organizacional escrita (normas), padrões e procedimentos para a execução do seu planejamento?
 Sim Não As vezes
 Caso a resposta seja não, vá para a questão 25.
24. Quanto aos padrões e procedimentos definidos pela organização para planejar o projeto de software, pode-se afirmar que são?
 Rígidos e entram a execução do projeto
 Rígidos, porém NÃO entram a execução do projeto
 Flexíveis de acordo com as características do projeto⁹, facilitando a execução do projeto
25. No planejamento do projeto de software quais são as principais fontes de informações utilizadas?
 Os requisitos não técnicos¹⁰
 Os requisitos técnicos (requisitos funcionais¹¹ e não-funcionais¹²)
 O plano, as estimativas documentadas¹³ e os riscos dos projetos anteriores
 Relatórios de acompanhamento e encerramento dos projetos anteriores
 A experiência do líder do projeto e da equipe de desenvolvimento
 Outras (especifique) []
26. As estimativas, riscos, atividades e compromissos são definidos e documentados no planejamento do projeto de software?
 Sim
 Não (por quê?) []
 As vezes (por quê?) []
 Caso a resposta seja não, vá para a questão 28.
27. As estimativas, atividades e compromissos definidos e documentados no planejamento do projeto de software são disponibilizados para as partes interessadas a fim de que elas conheçam-nas e aprove-nas ou não?
 Sim
 Não (por quê?) []
 As vezes (por quê?) []

⁹ Natureza do produto de software, tecnologia utilizada, requisitos de prazos e custo.

¹⁰ Prazo, custo, pessoal, fatores políticos e organizacionais.

¹¹ Descrição das diversas operações que clientes e usuários esperam que sejam realizadas pelo software.

¹² Determinam as características desejáveis do software quanto à usabilidade, confiabilidade, desempenho, suportabilidade, bem como descrevem restrições de projeto e conformidades legais.

¹³ Tamanho, esforço, custo e cronograma.

28. Quais são as principais informações utilizadas pela gerência sênior para aprovar o planejamento do projeto?
- Os requisitos técnicos (requisitos funcionais¹⁴ e não-funcionais¹⁵)
 - Os requisitos não técnicos¹⁶
 - Os riscos do projeto de software relacionados
 - Outros (especifique) []
29. Quais são os principais fatores que dificultam o uso de estimativas para planejar as atividades e compromissos do projeto de software?
- Projeto envolvendo o uso de novas tecnologias
 - Inadequada transferência de tecnologia para o projeto
 - Padrões, políticas e metodologias de engenharia de software inadequados
 - Métodos e ferramentas de processamento das informações inadequados
 - Influência política no projeto
 - Pressão excessiva de prazo

Parte 5 – Área-Chave de Processo Acompanhamento e Supervisão do Projeto de Software

Tem como objetivo estabelecer visibilidade adequada sobre o progresso real, de forma que a gerência possa tomar ações corretivas quando o desempenho do projeto de software sofre um desvio significativo com relação ao planejado. Ações corretivas podem incluir revisões do plano de desenvolvimento para refletir as reais realizações, replanejamento do trabalho restante ou realização de ações para melhorar o desempenho. Esta área-chave envolve o acompanhamento e revisão das realizações e resultados em comparação as estimativas, compromissos e planos documentados; envolve o ajuste dos planos de acordo com os resultados da análise do estimado versus realizado.

30. Os projetos seguem uma política organizacional escrita (normas), padrões e procedimentos para acompanhar e supervisionar os projetos de software?
- Sim Não As vezes
- Caso a resposta seja não, vá para a questão 32.
31. Quanto aos padrões e procedimentos definidos pela organização para acompanhar e supervisionar os projetos de software, pode-se afirmar que são?
- Rígidos e entram a execução do projeto
 - Rígidos, porém NÃO entram a execução do projeto
 - Flexíveis de acordo com as características do projeto¹⁷, facilitando a execução do projeto
32. No acompanhamento e supervisão do projeto de software quais são as principais fontes de informações utilizadas pelo líder de projeto?
- Reunião presencial ou via áudio/vídeo-conferência com a equipe do projeto
 - As análises comparativas entre os resultados reais do projeto e as estimativas geradas no planejamento
 - Os riscos do projeto de software relacionados
 - Informações de acompanhamento e supervisão de projetos anteriores
 - Outros (especifique) []

¹⁴ Descrição das diversas operações que clientes e usuários esperam que sejam realizadas pelo software.

¹⁵ Determinam as características desejáveis do software quanto à usabilidade, confiabilidade, desempenho, suportabilidade, bem como descrevem restrições de projeto e conformidades legais.

¹⁶ Prazo, custo, pessoal, fatores políticos e organizacionais.

¹⁷ Natureza do produto de software, tecnologia utilizada, requisitos de prazos e custos.

33. Quais são as principais informações utilizadas pela gerência sênior para acompanhar o projeto de software?
- Reunião com o líder de projeto
 - Indicadores de custos, de pessoal e de aderência ao cronograma
 - Conflitos e os tópicos não solucionados pelo líder do projeto
 - As análises dos riscos do projeto de software levantadas pela equipe
 - Outros (especifique) []
34. As informações referentes ao acompanhamento do projeto de software, inclusive as ações corretivas tomadas quando os resultados reais diferem significativamente do planejamento, são documentadas e disseminadas para as partes interessadas?
- Sim
 - Não []
 - As vezes (por quê?) []
- Caso a resposta seja não, vá para a questão 36.
35. Com que frequência as informações referentes ao acompanhamento do projeto de software são disseminadas para as partes interessadas no projeto?
- Semanalmente
 - Quinzenalmente
 - Mensalmente
 - Outras (especifique) []
36. Quais são as principais dificuldades encontradas para registrar, disseminar usar as informações referentes ao acompanhamento e supervisão do projeto, inclusive as ações corretivas quando os resultados reais diferem significativamente do planejamento?
- Pressão excessiva por prazo
 - Infra-estrutura inadequada de ferramentas de suporte
 - Ausência de conscientiza
 - Equipe do projeto de software não consciente da importância do registro das falhas a fim de não repeti-las em projetos futuros
 - Outras (especifique) []
37. Ao final do projeto de software, o líder e a equipe do projeto realizam conjuntamente avaliações a respeito das falhas e acertos ocorridos durante a execução do projeto?
- Sim, por meio de reuniões formais (as informações são registradas)
 - Sim, por meio de reuniões informais (as informações NÃO são registradas)
 - Não

Parte 6 – Área-Chave de Processo Garantia da Qualidade de Software

Tem como objetivo fornecer uma gestão com visibilidade apropriada sobre os processos em uso pelo projeto de software e sobre os produtos construídos. Esta área-chave envolve a revisão e a auditoria dos produtos de software, atividades para verificar o quanto eles atendem aos procedimentos e padrões aplicáveis e o fornecimento dos resultados destas revisões e auditorias ao gerente do projeto e outras área envolvidas. A Garantia de Qualidade de Software é parte integrante da maioria dos processos de gestão e de engenharia de software.

38. Os projetos seguem uma política organizacional escrita (normas), padrões e procedimentos para garantir a qualidade dos produtos de software?
 Sim Não As vezes
 Caso a resposta seja não, vá para a questão 40.
39. Quanto aos padrões e procedimentos definidos pela organização para garantir a qualidade dos produtos de software, pode-se afirmar que são?
 Rígidos e entram a execução do projeto
 Rígidos, porém NÃO entram a execução do projeto
 Flexíveis de acordo com as características do projeto¹⁸, facilitando a execução do projeto
40. Para realizar a garantia da qualidade de software quais são as principais fontes de informações utilizadas pelos consultores de qualidade?
 A revisão do plano, padrões e procedimentos e das atividades de engenharia de software do projeto de desenvolvimento de software
 A auditoria sobre os artefatos de software
 Outras (especifique) []
41. Quais são as maiores dificuldades encontradas para executar as revisões e auditorias de qualidade do software?
 Equipe de desenvolvimento não consciente da importância da qualidade do software
 Falha na comunicação entre os consultores de qualidade e os membros da equipe de desenvolvimento
 Membros da equipe de desenvolvimento e/ou consultores de qualidade treinados inadequadamente
 Ausência ou número insuficiente de membros da equipe com perfil de consultor de qualidade de software
 Pressão excessiva de prazo
 Outros (especifique) []
42. A qualidade do projeto de software é acompanhada por meio de medições? (por exemplo: número total de ocorrências de inadequações geradas nas revisões de qualidade).
 Sim
 Não (por quê?) []
 As vezes (por quê?) []

¹⁸ Natureza do produto de software, tecnologia utilizada, requisitos de prazos e custos.

43. Com que freqüência as informações referentes a qualidade do projeto de software são disseminadas para o gerente sênior e o líder do projeto?
- () Semanalmente
 () Quinzenalmente
 () Mensalmente
 () Outras (especifique) []
44. As informações referentes às medições realizadas são usadas como suporte a melhoria da gestão do projeto de software e ao aperfeiçoamento e aprendizagem da equipe?
- () Sim (como?) []
 () Não

Parte 7 – Área-Chave de Processo Gestão da Configuração de Software

Estabelecer e manter a integridade dos produtos (artefatos técnicos ou de gestão) do projeto de software ao longo de todo o ciclo de vida do software do projeto. Esta área-chave envolve a identificação da configuração do software em determinadas fases durante o seu desenvolvimento (p.e. produtos intermediários de software selecionados e suas respectivas descrições), o controle sistemático das mudanças na configuração e a manutenção da integridade e rastreabilidade da configuração ao longo do ciclo de vida do software. A Gestão de Configuração de Software é parte integrante da maioria dos processos de gestão e de engenharia de software.

45. Os projetos seguem uma política organizacional escrita (normas), padrões e procedimentos para controlar as mudanças dos produtos de software que estejam sob gestão de configuração?
- () Sim () Não () As vezes
 Caso a resposta seja não, vá para a questão 47.
46. Quanto aos padrões e procedimentos definidos pela organização para controlar as mudanças dos produtos de software que estejam sob gestão de configuração, pode-se afirmar que são?
- () Rígidos e entram a execução do projeto
 () Rígidos, porém NÃO entram a execução do projeto
 () Flexíveis de acordo com as características do projeto¹⁹, facilitando a execução do projeto
47. É estabelecido um sistema de repositório de gerenciamento de configurações²⁰ onde são armazenados os produtos (artefatos técnicos ou de gestão) do projeto de software?
- () Sim () Não () As vezes
 Caso a resposta seja não, vá para a questão 49.
48. Como é feito o gerenciamento do acesso as informações (artefatos técnicos ou de gestão) contidas no repositório de gerenciamento de configurações do projeto de software?
- () O acesso é liberado com restrições para todas as equipes de projeto do pólo de desenvolvimento
 () O acesso é concedido sem restrições para toda a equipe do projeto
 () O acesso é concedido para os membros da equipe de acordo com as atividades que eles executam
 () Outros (especifique) []

¹⁹ Natureza do produto de software, tecnologia utilizada, requisitos de prazos e custos.

²⁰ Local onde são armazenados todos artefatos gerados no projeto de software

49. Quais são os principais fatores que dificultam controlar as mudanças dos produtos que estejam sob gestão de configuração?
- Padrões, políticas e metodologias de gestão de configuração de software inadequados
 - Infra-estrutura e ferramentas de gestão de configuração de software inadequados
 - Influência política no projeto
 - Mudanças de requisitos
 - Pressão excessiva de prazo
 - Membros da equipe de desenvolvimento treinados inadequadamente
 - Membros da equipe de desenvolvimento inexperientes
 - Ausência ou número insuficiente de membros da equipe de desenvolvimento com perfil de gestor de configuração de software
 - Outros (especifique) []

Parte 8 – Área-Chave de Processo Gestão de Subcontratação de Software

Visa selecionar subcontratados qualificados e gerenciá-los de modo efetivo. Esta área-chave combina os assuntos relativos à Gestão de Requisitos, Planejamento de Projeto de Software e Acompanhamento de Projeto de Software, para controle básico de gestão, junto com a necessária coordenação da Garantia de Qualidade de Software e Gestão de Configuração de Software, as quais são aplicadas apropriadamente aos subcontratados.

Observação: *Caso os projetos desenvolvidos pelo respondente não envolvam contratos com terceiros, as questões referentes a área-chave Subcontratação de Software não precisam ser respondidas.*

50. Os projetos seguem uma política organizacional escrita (normas), padrões e procedimentos para gerenciar a subcontratação de software?
- Sim Não As vezes
- Caso a resposta seja não, vá para a questão 52.
51. Quanto aos padrões e procedimentos definidos pela organização para gerenciar a subcontratação de software, pode-se afirmar que são?
- Rígidos e entram a execução do projeto
 - Rígidos, porém NÃO entram a execução do projeto
 - Flexíveis de acordo com as características do projeto²¹, facilitando a execução do projeto
52. Quais são as principais fontes de informações utilizadas na elaboração do contrato de subcontratação que especifica os produtos e atividades de software a serem subcontratados?
- Contrato padrão
 - Os requisitos não técnicos²²
 - Os requisitos técnicos (requisitos funcionais²³ e não-funcionais²⁴)
 - O plano do projeto de software
 - A especificação de outras subcontratações similares, ou seja, com as mesmas características e contexto
 - A experiência do líder do projeto
 - Outras (especifique) []

²¹ Natureza do produto de software, tecnologia utilizada, requisitos de prazos e custos.

²² Prazo, custo, pessoal, fatores políticos e organizacionais.

²³ Descrição das diversas operações que clientes e usuários esperam que sejam realizadas pelo software.

²⁴ Determinam as características desejáveis do software quanto à usabilidade, confiabilidade, desempenho, suportabilidade, bem como descrevem restrições de projeto e conformidades legais.

53. Quais são as principais fontes de informações utilizadas na seleção do subcontratado?
- Propostas apresentadas para a subcontratação planejada
 - Registros de desempenho anterior em trabalho similar
 - Registros de experiências e desempenho anteriores em aplicações similares
 - Capacitação em engenharia de software e em gestão de projetos
 - Indicadores de pessoal disponível para realizar o trabalho
 - Outros (especifique) []
54. Para acompanhar as atividades de software do subcontratado a gerência contratante utiliza:
- Plano de desenvolvimento de software do subcontratado
 - Revisões periódicas da situação/coordenação com a gerência do subcontratado
 - Indicadores de desempenho técnico, de custos, de pessoal e de aderência ao cronograma
 - Conflitos e questões não solucionados internamente pelo subcontratado
 - Os riscos do projeto de software relacionados
 - Outros (especifique) []
55. Quais são os principais fatores que dificultam a obtenção de informações que proporcione o acompanhamento das atividades de software do subcontratado?
- Padrões, políticas e metodologias de gestão e engenharia de software adotados pelo subcontratado difere daqueles utilizados pelo contratante.
 - Problemas com a liberação do acesso ao repositório de configurações base de software do subcontratado.
 - Outros (especifique) []
56. O desempenho do subcontrato de software é documentado para ser utilizado como fontes de informações para selecionar futuros subcontratados?
- Sim Não
57. As atividades de gestão de subcontratação de software são acompanhadas por meio de medições? (por exemplo: custo das atividades de gestão de subcontratação comparadas com o plano do projeto, datas reais de entrega dos produtos subcontratados comparadas com o plano do projeto).
- Sim Não As vezes
58. As informações referentes às medições realizadas são usadas como suporte a melhoria da gestão de subcontratação e ao aperfeiçoamento e aprendizagem da equipe?
- Sim (como?) []
- Não (por quê?) []

