



UNIVERSIDADE DE BRASÍLIA

Faculdade de Economia, Administração, Contabilidade e Ciência da Informação e Documentação

Departamento de Ciência da Informação e Documentação

Programa de Pós-Graduação em Ciência da Informação

**ELEMENTOS BÁSICOS PARA A CONSTRUÇÃO DE MODELO DE
GESTÃO DA INFORMAÇÃO PARA ENGENHARIA DE *SOFTWARE*:
ESTUDO DE CASO PARA ATIVIDADE DE REQUISITOS NO SERPRO**

Roberto Duarte Pontual de Lemos

BRASÍLIA, DF

2009



UNIVERSIDADE DE BRASÍLIA
Faculdade de Economia, Administração, Contabilidade e Ciência da Informação e Documentação
Departamento de Ciência da Informação e Documentação
Programa de Pós-Graduação em Ciência da Informação

ELEMENTOS BÁSICOS PARA A CONSTRUÇÃO DE MODELO DE GESTÃO DA INFORMAÇÃO PARA ENGENHARIA DE *SOFTWARE*: ESTUDO DE CASO PARA ATIVIDADE DE REQUISITOS NO SERPRO

Roberto Duarte Pontual de Lemos

Dissertação apresentada ao Departamento de Ciência da Informação e Documentação da Universidade de Brasília como parte dos requisitos para obtenção do Título de Mestre em Ciência da Informação.

Orientador: Prof. Dr. Rogério Henrique de Araújo Júnior

BRASÍLIA, DF

2009



FOLHA DE APROVAÇÃO

Título: “Elementos básicos para a construção de modelo gestão da informação para engenharia de software: estudo de caso para atividade de requisitos no Serpro”

Autor (a): Roberto Duarte Pontual de Lemos.

Área de concentração: Transferência da Informação.

Linha de pesquisa: Gestão da Informação e do Conhecimento.

Dissertação submetida à Comissão Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação do Departamento de Ciência da Informação e Documentação da Universidade de Brasília como requisito parcial para obtenção do título de **Mestre** em Ciência da Informação.

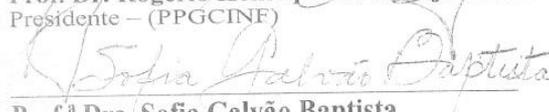
Dissertação aprovada em: 24 de agosto de 2009.

Aprovado por:



Prof. Dr. Rogério Henrique de Araújo Júnior

Presidente – (PPGCINF)



Prof.ª Dra. Sofia Galvão Baptista

Membro Interno – (UnB/PPGCINF)



Dr. Marcilio Mendes de Oliveira

Membro Externo – (OSx Telecom - Visent)

Prof. Dr. Renato Tarciso Barbosa de Sousa
Suplente – (UnB/PPGCINF)

À minha esposa, Carla, e minhas filhas – Marcela e Maria Antônia – pelo amor, carinho, compreensão nas muitas horas de ausência, pelo incentivo nas horas de incerteza, e pela alegria nas conquistas. Vocês são muito importantes em tudo em minha vida.

AGRADECIMENTOS

A realização deste trabalho contou com a colaboração de muitas pessoas, que se dispuseram a conversar e compartilhar suas idéias e a quem agradeço e sempre serei grato. Em especial agradeço a:

Ao meu orientador, Prof. Dr. Rogério Henrique de Araújo Júnior, pelo compartilhamento de seu conhecimento e experiências; dedicação; paciência para esclarecer as incontáveis dúvidas; e incentivo e apoio nos momentos difíceis, pontos chave para a conclusão deste trabalho;

Aos Profs. Drs. Sofia Galvão Baptista e André Porto Ancona Lopez, representando todos os professores do Departamento de Ciência da Informação e Documentação da UnB, pelo apoio durante o curso, mas principalmente pelo desafio e estímulo na busca de novas idéias e produção de conhecimento científico;

Aos meus colegas do Serpro, pelas sugestões, apoio e colaboração na realização do trabalho;

À Jucilene e Marta, em nome dos funcionários do CID, pela paciência, bom humor e apoio na solução dos pedidos, geralmente de última hora;

A meus pais, pelos ensinamentos, e pelo exemplo de comportamento ético, amor e dedicação à família;

A Deus, pelas oportunidades e presença em minha vida.

RESUMO

A ciência da informação tem ampliado suas áreas de atuação, como parte da influência de suas características interdisciplinares. Uma das disciplinas correlatas em que ela tem atuado é na ciência da computação, com o uso de modelos de gestão da informação para engenharia de software. O desenvolvimento de sistemas de informação tornou-se uma atividade crítica para boa parte das organizações, em função de sua dependência dos sistemas para suas atividades operacionais e gerenciais. A dificuldade das empresas de desenvolvimento de software em entregar no prazo sistemas que atendam as exigências dos usuários, levou a adoção de técnicas de engenharia e modelos de maturidade, com objetivo de tornar a atividade mais previsível, em termos de prazos de entrega e qualidade dos sistemas. Entretanto, apenas as soluções técnicas não atenderam as expectativas dos clientes, e uma das alternativas avaliadas foi a associação entre engenharia de software e conceitos e modelos de gestão da informação e conhecimento. Esta pesquisa avaliou modelos de gestão da informação utilizados em processos de desenvolvimento de software, e identificou os elementos chave para a construção de um modelo de gestão da informação para ser integrado ao processo do Serpro de desenvolvimento de software (Processo Serpro de Desenvolvimento de Soluções - PSDS). O objetivo é que a futura utilização destes elementos em um modelo de gestão da informação para o PSDS melhore aspectos relacionados ao reuso de informações em projetos de software, e permita a redução de prazos e melhoria da qualidade dos sistemas de informação desenvolvidos pelo Serpro.

Palavras-chave: ciência da informação, gestão da informação, modelos, engenharia de software, desenvolvimento de sistemas, Serpro

ABSTRACT

The interdisciplinary characteristics of information science have expanded its influence on other disciplines. One of the main correlated disciplines is computer science, specially with the association of information management models and software engineering. Information systems development has become a critical activity in most organizations, due to its reliance on systems for their operational activities and management. The difficulty of software development companies to deliver systems that meet user requirements led to adoption of maturity models and engineering techniques, with the aim of making the activity more predictable in terms of quality and delivery times. However, technical solutions alone were not able to met customers expectations, and one of the alternatives evaluated was the association between software engineering and concepts of knowledge and information management. This research evaluated management information models used in software development processes, and identified key elements to build a model of information management to be integrated into PSDS (Serpro process of software development). The goal is that future use of these elements allow the integration of PSDS and a model of information management, and lead to improve information reuse in systems development and implementation, and allows a reduction of development time, and a quality improvement of information systems developed by Serpro.

Key-words: information science, information management, models software engineering, systems development, Serpro.

LISTA DE FIGURAS

Figura 1 - Modelo conceitual do Serpro	16
Figura 2 - Modelo Ecológico para Gestão da informação.....	32
Figura 3 - Processo de gestão da informação	33
Figura 4 - Ciclo da gestão da informação	34
Figura 5 - Representação de domínios para requisitos	49
Figura 6 - Custo relativo para corrigir um erro	53
Figura 7 - Paradigma da melhoria da qualidade (QIP)	62
Figura 8 - A Fábrica de Experiências (FE).....	63
Figura 9 - Visão geral - fluxo da macroatividade de requisitos do PSDS.....	84
Figura 10 – Esquema – modelo de Informação para o PSDS.....	127
Figura 11 – Representação do Modelo	128
Figura 12 – Modelo e o Ciclo de Gestão da Informação	139

LISTA DE GRÁFICOS

Gráfico 1 – Dados de Formação Acadêmica.....	88
Gráfico 2 – Armazenamento de Informações dos Clientes	90
Gráfico 3 – Formas de Compartilhamento de Informações.....	91
Gráfico 4 – Clientes e a Especificação de RNFs.....	93
Gráfico 5 – Clientes e aproveitamento de RNFs.....	94
Gráfico 6 – Índice de consulta à BS no início de projetos.....	97
Gráfico 7 – Razões para não consulta à BS.....	97
Gráfico 8 – Índice de registro de experiências na BS	98
Gráfico 9 – Razões para não registro de experiências	98
Gráfico 10 – Percepção sobre reuso de informações e o Serpro	101

RELAÇÃO DE ABREVIATURAS E SIGLAS

- ADI** – *American Documentation Institute*
- ASIS** – *American Society of Information Science*
- BE** – Base de Experiências
- BS** – Base de Soluções
- CASE** – *Computer Aided Software Engineering*
- CI** – Ciência da Informação
- CMMI** – *Capability Maturity Model for Integration*
- FE** – Fábrica de Experiências
- GSFC** – *Godard Space Flight Center*
- IBGE** - Fundação Instituto Brasileiro de Geografia e Estatística
- IBM** – *International Business Machine*
- IEC** – *International Electrotechnical Commission*
- IEEE** – *Institute of Electrical and Electronic Engineers*
- ISO** – *International Organization for Standardization*
- KPA** – *Key process area*
- MPS-BR** – Melhoria do Processo de Software Brasileiro
- NASA** – *National Aeronautics and Space Administration*
- NIST** – *National Institute of Standards and Technology*
- PMA** – *PostMortem Analysis* (análise pós-morte)
- PDCA** – *Plan/Do/Check/Act*
- PSDS** - Processo Serpro de Desenvolvimento de Soluções
- QIP** – *Quality Improvement Paradigm*
- RNF** – Requisitos não-funcionais
- RUP** – *Rational Unified Process*
- SEC** – *Software Experience Center*
- SEI** – *Software Engineering Institute*
- SEL** – *Software Engineering Laboratory*
- Serpro** – Serviço Federal de Processamento de Dados
- SWEBOK** – *Software Engineering Body of Knowledge*
- Ti** – Tecnologia da Informação
- UnB** – Universidade de Brasília

SUMÁRIO

1. INTRODUÇÃO	14
2. JUSTIFICATIVA E COLOCAÇÃO DO PROBLEMA	15
2.1. O Serpro	15
2.2. Problema	18
2.3. Objetivos	20
2.3.1. Objetivo Geral	20
2.3.2. Objetivos Específicos	20
2.4. Justificativa	20
3. REVISÃO DE LITERATURA	22
3.1. Trabalhos Correlatos	22
3.2. Ciência da Informação e Informação	24
3.2.1. Ciência da Informação	24
3.2.2. Dados, Informação e Conhecimento	27
3.2.3. Conclusão	30
3.3. Gestão da Informação	31
3.3.1. Ciclo da Gestão da Informação – Davenport	31
3.3.2. Ciclo da Gestão da Informação – Choo	33
3.3.3. Conclusão	41
3.4. Desenvolvimento e Engenharia de <i>Software</i>	42
3.4.1. Desenvolvimento de <i>Software</i>	42
3.4.2. Processos de <i>Software</i>	45
3.4.3. Modelos de Maturidade para Desenvolvimento de <i>Software</i>	46
3.4.4. Requisitos e Engenharia de Requisitos	48
3.4.5. Conclusão	52
3.5. Gestão da Informação e Engenharia de <i>Software</i>	54
3.5.1. Gestão da Informação e Engenharia de <i>Software</i> – Cenário	55
3.5.2. Gestão da Informação em Organizações de <i>Software</i>	57
3.5.3. Um Modelo Geral para Reuso de Experiências em Engenharia de <i>Software</i>	61
3.5.4. A Experiência da NASA – O Laboratório de Engenharia de <i>Software</i>	64
3.5.5. A Experiência da DaimlerChrysler – Centro de Experiência em <i>Software</i>	66
3.5.6. A Experiência da Ericsson Technology – A Máquina de Experiências	67
3.5.7. Conclusões	70
3.6. Conclusões da Revisão de Literatura	72
4. PRESSUPOSTOS	75
4.1. Pressuposto Geral	75
4.2. Pressupostos Específicos	75

4.3. Variáveis	76
4.3.1. Variáveis do 1º Pressuposto Específico	76
4.3.2. Variáveis do 2º Pressuposto Específico	76
4.3.3. Variáveis do 3º Pressuposto Específico	76
5. METODOLOGIA	77
5.1. Delimitação do Estudo	77
5.2. Embasamento Teórico da Metodologia	78
5.3. Caracterização do Universo	79
5.4. Caracterização da Amostra	79
5.5. Delineamento da Pesquisa	80
5.6. Coleta de Dados	80
5.6.1. Técnica Selecionada	80
5.6.2. Instrumento de Coleta de Dados	81
6. AMBIENTE DA PESQUISA – O PSDS	81
7. ANÁLISE DOS DADOS E COMPROVAÇÃO DOS PRESSUPOSTOS	87
7.1. Análise Descritiva dos Dados	87
7.1.1. Análise das Informações Demográficas	87
7.1.1.1. Formação Acadêmica	87
7.1.1.2. Cargo e Experiência	88
7.1.2. Informações do Cliente e Domínio da Aplicação – Bloco I	89
7.1.2.1. Informações sobre clientes	89
7.1.2.2. Disponibilidade e registro de informações de clientes	89
7.1.2.3. Aquisição e armazenamento de informações de clientes	90
7.1.2.4. Compartilhamento de informações dos clientes	90
7.1.3. Avaliação do Primeiro Pressuposto Específico	91
7.1.4. Requisitos Não-funcionais e Produtos e Serviços – Bloco II	92
7.1.4.1. Identificação e registro de requisitos não-funcionais	92
7.1.4.2. Clientes e requisitos não-funcionais	93
7.1.4.3. Utilização de requisitos não-funcionais	93
7.1.4.4. Disponibilidade de produtos e serviços	94
7.1.5. Avaliação do Segundo Pressuposto Específico	95
7.1.6. Reuso de Informações – Bloco III	95
7.1.6.1. PSDS e o registro de experiências	95
7.1.6.2. Consulta a base de soluções	96
7.1.6.3. Registro de informações na base de soluções	97
7.1.6.4. Reunião de avaliação dos projetos	98
7.1.7. Avaliação do Terceiro Pressuposto Específico	99
7.1.8. Aspectos Gerais – Bloco IV	100
7.1.8.1. Conhecimento do conceito de gestão da informação	100
7.1.8.2. Influência do reuso de informações	100
7.1.8.3. Posição do Serpro quanto ao reuso de informações	100
7.2. Análise dos Dados e o Ciclo de Gestão da Informação	102

7.2.1. Necessidades de informação	102
7.2.2. Aquisição da Informação	102
7.2.3. Organização e Armazenamento da Informação	103
7.2.4. Produtos e Serviços de Informação	103
7.2.5. Distribuição da Informação	104
7.2.6. Uso da Informação	104
8. CONCLUSÕES	105
8.1. Sugestões para novas pesquisas	107
REFERÊNCIAS	108
GLOSSÁRIO	114
APÊNDICE A - INSTRUMENTO DE COLETA DE DADOS	116
APÊNDICE B - PROPOSTA DE MODELO DE GESTÃO DA INFORMAÇÃO	126
APÊNDICE C – PROPOSTA PARA DOMÍNIOS DE NEGÓCIOS NO SERPRO	143

1. INTRODUÇÃO

“A informação é uma parte integral de toda a atividade humana.” – a afirmativa de Castells (2001, p. 78) mostra a importância da informação, e de seu uso, na sociedade atual.

O mesmo autor destaca (p. 87) que nas duas últimas décadas assistimos o surgimento de uma economia informacional e global. Ela é informacional porque a produtividade e a competitividade dos agentes dessa economia - empresas, regiões ou nações - dependem basicamente de sua capacidade de gerar, processar e aplicar de forma eficiente a informação.

A capacidade de capturar informações, organizá-las, e utilizá-las de forma ordenada com as estratégias de negócio tornou-se fator crítico de sucesso em um cenário de evolução tecnológica e competição globalizada. O competidor pode estar na próxima esquina ou em outro continente, e uma nova tecnologia pode tornar negócios obsoletos em poucos meses. Segundo Drucker (1988, p. 103), as organizações não tem escolha, seu funcionamento deve estar baseado no uso de informações.

Entretanto, apesar de constatarem a importância do uso de informações, poucas organizações tratam o seu gerenciamento com o mesmo nível de atenção de outras áreas tradicionais, como finanças e logística.

A evolução tecnológica faz com que as organizações dos setores público e privado apóiem suas estratégias de atuação no uso intensivo de sistemas de informação, e enfrentem um desafio cada vez mais importante para seu sucesso: como transformar o conhecimento das pessoas, suas normas, ou as leis do país em sistemas de informação?

Não basta construir sistemas, eles devem retratar a realidade, serem estáveis em seu funcionamento e flexíveis para refletir com rapidez as mudanças do cenário de negócios. Assim, nas últimas duas décadas a atividade de desenvolvimento de sistemas de informação passou a ter relevância para o sucesso das organizações.

Para atender a estas necessidades as organizações de tecnologia da informação (TI) envolvidas com desenvolvimento de *software* vêm adotando soluções focadas em processos, como modelos de maturidade e processos de engenharia de *software*. Mas, mesmo com a adoção de processos e uso de ferramentas, Birk *et al* (2006, p. 59) demonstram que projetos na área de desenvolvimento de *software* ainda apresentam um histórico constante de exceder os custos e prazos previstos.

O objetivo do trabalho é realizar um estudo de caso no Serpro, e avaliar a integração de um modelo de gestão da informação com o processo de desenvolvimento de sistemas do Serpro – PSDS (Processo Serpro de Desenvolvimento de Soluções) – na macroatividade de requisitos. Como resultado, espera-se que o modelo proposto melhore o fluxo de informações no processo de desenvolvimento de *software*, facilite o reuso de informações e melhore a qualidade dos produtos de *software* desenvolvidos pelo Serpro.

2. JUSTIFICATIVA E COLOCAÇÃO DO PROBLEMA

A colocação do problema e justificativa da pesquisa será iniciada com a descrição do Serpro, organização onde está localizado o estudo de caso.

2.1. O Serpro

O Serpro – Serviço Federal de Processamento de Dados - é uma empresa pública do Ministério da Fazenda, criada em 1964 para atender necessidades de modernização dos processos de trabalho da administração fazendária.

Ao longo dos últimos quarenta anos, além da evolução do mercado, visível até nos termos usados em sua área de atuação - “processamento de dados, “informática” e “tecnologia da informação” – a empresa teve oportunidade de ampliar seu mercado de atuação e atender a outros órgãos da administração pública federal.

A partir de 1987, com a implantação do SIAFI (Sistema Integrado de Administração Financeira), o governo federal percebeu que era possível obter benefícios expressivos com a adoção de processos padronizados para suas atividades operacionais.

Ele constatou ainda que a automação destes processos com o uso de sistemas de informação permitia ganhos significativos, com redução dos custos de operação, maior eficiência e padronização no atendimento aos cidadãos. Esta automação permitia maior transparência nas ações de governo e maior agilidade no atendimento às demandas informacionais da sociedade.

Foi então iniciada a construção de novos sistemas de informação para atender processos estruturadores, que afetam boa parte dos órgãos da administração pública federal, ou que afetam parcela expressiva da sociedade, como os sistemas da arrecadação federal, licenciamento de veículos, carteira de habilitação e emissão de passaporte, apenas para citar alguns.

Dentre os principais sistemas de informação desenvolvidos e operados pelo Serpro podemos destacar¹:

- Sistema Integrado de Administração Financeira (SIAFI);
- Sistema Integrado de Administração de Pessoal (SIAPE);
- Sistema Integrado de Administração de Serviços Gerais (SIASG);
- Sistemas da Receita Federal do Brasil (RFB) nas áreas de arrecadação, fiscalização e tributação;
- Sistema Integrado de Comércio Exterior (SISCOMEX);
- Portal de Compras do Governo Federal (ComprasNet);
- Registro Nacional de Veículos Automotores (RENAVAM);
- Registro Nacional de Carteiras de Habilitação (RENACH);

¹ Informações disponíveis na Intranet do Serpro – disponível apenas para acesso interno.

- Sistema Nacional de Passaporte (SINPA);
- Sistema de Tráfego Internacional (STI).

Para atender a estes serviços o Serpro tem uma estrutura composta de Sede e 10 Regionais, onde ficam as atividades operacionais, além de escritórios que cobrem todas as unidades da federação. A Figura 1 mostra o modelo operacional² do Serpro – as unidades comerciais (Unidades de Relacionamento com Clientes) são responsáveis pelas atividades comerciais – assinatura de contratos, atendimento aos clientes e gestão dos serviços.

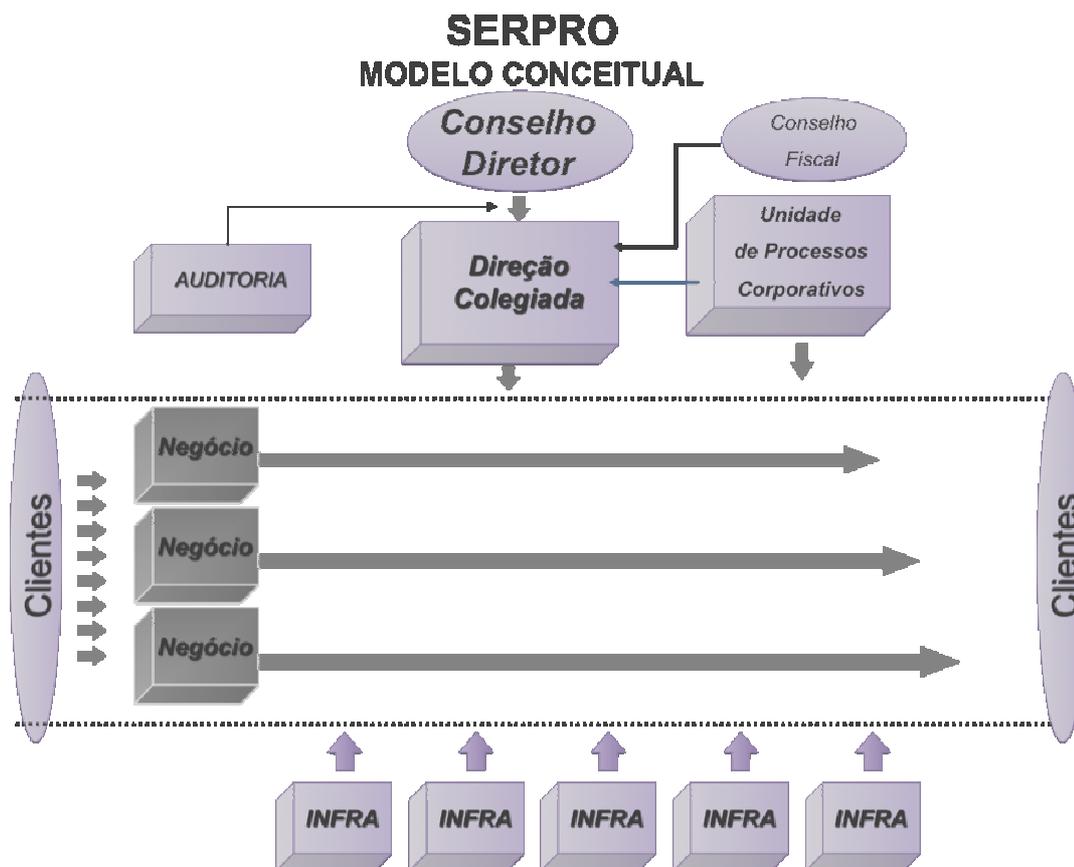


Figura 1 - Modelo conceitual do Serpro

Fonte: Modelo Conceitual do Serpro – Sistema de Normas Organizacionais do Serpro

As unidades técnicas – Unidades de Produtos e Serviços – são responsáveis pela operação e gestão dos sistemas e rede de comunicações. Elas são apoiadas por unidades de gestão empresarial, responsáveis pelos processos de finanças, gestão de pessoas e recursos logísticos.

² O modelo operacional sofreu algumas alterações em julho de 2007, mas elas não afetaram o processo de desenvolvimento, e não serão consideradas neste trabalho.

A Empresa possui cerca de 10.500 empregados³, e a parte principal da infraestrutura tecnológica é composta de três centros de dados e uma rede de comunicações que alcança cerca de 150 municípios brasileiros, com mais de 200.000 usuários cadastrados em seu sistema de controle de acesso. O faturamento da empresa em 2007 foi de cerca de R\$ 1,5 bilhões⁴, sendo que cerca de R\$ 500 milhões referem-se a serviços de desenvolvimento e manutenção de sistemas de informação.

Nos últimos anos, com a disseminação do conceito de governo eletrônico, e a evolução e ampliação do uso da Internet, o governo federal aumentou suas exigências em relação à segurança, disponibilidade e tempo de resposta dos sistemas de informação. A dinâmica da sociedade também exigiu maior qualidade e menor tempo para o seu desenvolvimento e implementação, já que o governo federal precisa acompanhar o ritmo do mercado, para atender suas necessidades nos setores de regulação, arrecadação e fiscalização de atividades econômicas e sociais.

A partir de 2001 foi iniciado programa para reorganizar o processo de desenvolvimento de sistemas do Serpro, com o objetivo de padronizar e documentar as atividades de desenvolvimento de sistemas, e atender as demandas dos clientes por menores prazos e maior qualidade dos sistemas de informação. O principal resultado deste programa foi a implantação do PSDS (Processo Serpro de Desenvolvimento de Soluções), baseado no *Rational Unified Process* (RUP), processo de engenharia de *software* elaborado pela empresa *Rational Software*, hoje uma divisão da IBM.

O termo “soluções” indica que o processo se preocupa com todos os aspectos envolvido na negociação, contratação, planejamento, desenvolvimento e implantação de um sistema de informação. O desenvolvimento de *software* é apenas uma das etapas previstas pelo PSDS.

Desde que o modelo operacional foi implantado, em 1995, as equipes de desenvolvimento experimentaram diferentes formas de organização, na busca de melhores resultados de qualidade e produtividade. Inicialmente elas estavam nas áreas comerciais, especializadas por tipo de cliente ou processo de governo.

Em 2003 foi iniciado processo de concentração, em que parte das equipes de desenvolvimento foi integrada em uma unidade única. Esta experiência durou cerca de dois anos, e em 2005 as equipes foram novamente separadas por unidade comercial.

Em outubro de 2007, com a chegada de nova administração na empresa, foi definida uma nova reorganização das equipes de desenvolvimento. Elas foram agrupadas em duas unidades de desenvolvimento de sistemas, uma para atendimento a Receita Federal do Brasil, maior cliente do Serpro, e outra equipe para atender aos demais clientes.

Estas mudanças foram ocasionadas pela percepção de que as equipes de desenvolvimento, mesmo com a implantação de um processo único de

³ Cerca de 7.000 empregados trabalham no Serpro, e cerca de 3.500 prestam serviços diretamente ao Min. da Fazenda, em atividades não relacionadas a serviços de TI. Dados de novembro de 2008 – Informações de RH do Serpro.

⁴ Dado de 2007

desenvolvimento, ainda apresentavam produtividade abaixo de padrões de mercado, e com baixo índice de reuso de soluções já existentes.

O Serpro possui hoje cerca de 20 equipes de desenvolvimento⁵ – chamadas de Pólos de Desenvolvimento. Estes Pólos têm cerca de 2.000 empregados distribuídos em dez Unidades Regionais – Brasília, Belém, Fortaleza, Recife, Salvador, Belo Horizonte, Rio de Janeiro, São Paulo, Curitiba e Porto Alegre.

Há uma determinação corporativa para que todas as equipes de desenvolvimento utilizem o PSDS, como forma única de trabalho. Entretanto, sua adoção ocorre de forma heterogênea entre as equipes. Um dos objetivos da implantação do PSDS era aproveitar o conhecimento e experiência acumulados pelas equipes de desenvolvimento, e incentivar o reuso de informações e artefatos de desenvolvimento – relatórios, especificações e código de programas, dentre outros.

Em relação a estratégias de reuso de informações e experiências, cabe destacar que o Serpro iniciou diversos projetos que buscavam seu incentivo, como a gestão pelos Critérios de Excelência da Fundação Nacional da Qualidade, o projeto de Gestão do Conhecimento, a realização de cursos de especialização em gestão do conhecimento e a criação de uma Intranet Corporativa, onde seriam registradas as informações consideradas relevantes por cada uma das unidades organizacionais da empresa.

Estes projetos, apesar de êxitos em algumas ações, ou áreas específicas, não tiveram continuidade e não oferecem suporte efetivo ao reuso ou reaproveitamento de informações, soluções e práticas de gestão. O PSDS e o processo corporativo para gestão de serviços implementaram o conceito de base de soluções, com objetivo de implementar ações efetivas de reuso.

Entretanto os resultados esperados não foram alcançados, parte em função do uso de tecnologias distintas e imaturas para os dois projetos, e parte pela falta de incentivo ao registro e reutilização de informações e experiências.

2.2. Problema

O desenvolvimento de *software*, de acordo com Bjornson e Dingsoyr (2008, p. 1056), é uma atividade intensiva no uso de informações e conhecimento. A etapa de levantamento de requisitos é onde ocorre a aquisição de conhecimento sobre os processos ou atividades a serem atendidas pelo sistema a ser desenvolvido.

A importância da gestão de requisitos para a qualidade e custos de um produto de *software* é discutida por diversos autores. Boehm e Basili (2001, p. 135) destacam que identificar e corrigir um problema após a entrega do produto pode ser até 100 vezes mais caro do que identificá-lo e corrigi-lo nas fases de requisitos e análise.

Segundo definição do *Institute of Electrical and Electronics Engineers* (IEEE) - SWEBOK (2004, p. 2-1) - requisito expressa necessidades e restrições de um

⁵ São utilizados dados de novembro de 2008. Fonte: InfoRH – Informações de Recursos Humanos do Serpro.

produto de *software*, que contribui para a solução de um problema real. Entretanto, a formalização (codificação) das necessidades de um grupo de pessoas ou uma organização não é uma tarefa simples.

Oberg *et al.* (2000, p. 3) destacam alguns pontos que mostram a complexidade da atividade de identificação de requisitos para o desenvolvimento de um sistema de informação:

- Requisitos nem sempre são óbvios, e podem ter fontes diversas em uma empresa;
- Requisitos nem sempre são facilmente expressos em palavras;
- Requisitos podem ser de tipos diversos, e ter diferentes níveis de detalhes;
- Requisitos diversos podem ser relacionados entre si, o que exige maior esforço na gestão das mudanças;
- Requisitos mudam ao longo do tempo.

Um dos maiores problemas em desenvolvimento de *software* é capturar, de forma clara, os requisitos que definem o que o produto deve ou não fazer. Prikladnicki *et al.* (2003, p. 2) destacam pesquisa indicando que cerca de 70% dos requisitos são difíceis de identificar, não são claros e nem organizados

Esta situação é ainda mais complexa na administração pública, em que nem todos os processos são definidos de forma clara e algumas áreas têm dificuldade na retenção de pessoas que conheçam seus processos de trabalho.

Assim, uma grande dificuldade do Serpro é a clara identificação dos requisitos necessários aos sistemas solicitados por órgãos da administração pública federal. As equipes de desenvolvimento, geralmente dispersas em mais de uma localidade, têm dificuldades na obtenção de informações junto aos clientes, na divulgação de forma consistente para todos os envolvidos no projeto, e em sua manutenção de forma atualizada, para garantir que os sistemas de informação atendam as expectativas dos clientes.

A implantação de um processo de desenvolvimento buscou formalizar ações de identificação e captura de requisitos, bem como sua divulgação e gestão nas equipes durante a execução dos projetos. Entretanto, após sete anos de uso do PSDS, os resultados para a atividade de gestão de requisitos ainda estão abaixo do esperado, gerando insatisfação dos clientes.

As perguntas geradoras para o trabalho são descritas a seguir:

- Como os processos associados à gestão da informação afetam o desenvolvimento de sistemas no Serpro?
- Como o fluxo de informações afeta a atividade de requisitos no desenvolvimento de sistemas no Serpro?
- Como aproveitar as informações disponíveis para evitar a repetição de erros já conhecidos em projetos de desenvolvimento de sistemas?
- Que elementos afetam o reuso de informações no processo de desenvolvimento do Serpro, e podem ser tratados para melhorar a qualidade dos produtos de *software*?

2.3. Objetivos

2.3.1. Objetivo Geral

- Identificar os elementos chave para a construção de um modelo de gestão da informação para o processo de desenvolvimento de sistemas do Serpro, com um estudo preliminar na atividade de requisitos.

2.3.2. Objetivos Específicos

- Analisar o fluxo de informações na macroatividade de requisitos do processo de desenvolvimento de *software* do Serpro.
- Avaliar como a disponibilidade de informações sobre os clientes, sistemas e requisitos não-funcionais afeta a definição de produtos e serviços de informação para as equipes de requisitos do Serpro.
- Analisar a influência do reuso de informações para a evolução da qualidade da atividade de requisitos para o desenvolvimento de sistemas de informação do Serpro.

2.4. Justificativa

A Ciência da Informação (CI), segundo Saracevic (1996, p. 42), possui três características gerais: ela é interdisciplinar por natureza, está inexoravelmente ligada à tecnologia da informação e é uma participante ativa na evolução da sociedade da informação.

Dentre as relações interdisciplinares avaliadas, é destacada a relação com a tecnologia da informação, que possui áreas com significativo componente informacional, e correntes de pesquisa e desenvolvimento com diferentes visões, modelos e enfoques em relação a ciência da informação. Esta relação interdisciplinar propicia um nível diverso de cooperação intelectual.

O cenário atual, com a importância dos sistemas de informações para as empresas e os cidadãos, o aumento significativo do volume de informações disponíveis e a facilidade de acesso e armazenamento, tem estimulado estudos de modelos de gestão de informações, e de que forma eles podem ser integrados aos processos organizacionais.

Estes estudos se concentram na avaliação dos processos envolvidos na identificação de necessidades, captura, armazenamento e uso de informações. Dois dos modelos mais conhecidos são os propostos por Choo (2003, p. 404) e Davenport (1994, p. 98), que detalham estes processos e descrevem a necessidade de mudar o comportamento das pessoas em relação à informação.

Por outro lado, Macedo (2005, p. 89) destaca que os sistemas de informação têm fundamental importância para a ciência da informação, ao fornecerem meios de captura, processamento e armazenagem, e viabilizam o acesso à informação. Um ponto crítico relacionado aos sistemas de informação é o seu processo de desenvolvimento.

Relatório do *The Standish Group* (1995, p. 2) mostra que apenas 16% dos projetos de *software* atingiram todos os seus objetivos – foram completados no prazo, dentro do orçamento previsto e atenderam as expectativas dos usuários. Cerca de 31% dos projetos iniciados foram cancelados por alguma razão e, dos concluídos, pouco mais de 52% custaram mais do que o previsto. Dados de 2006 mostram que a taxa de sucesso nos projetos de *software* foi de 35%.

Para tentar melhorar estes resultados foram definidos modelos para desenvolvimento de *software*, baseados em processos semelhantes aos de outras disciplinas, resultando na criação da engenharia de *software*. Ela prevê que os resultados de trabalhos já concluídos sejam registrados, estudados e usados no desenvolvimento de novos projetos, evitando assim erros já conhecidos e melhorando o resultado final.

Para apoiar a engenharia de *software* foram desenvolvidos nos últimos vinte anos modelos de processos e ferramentas de tecnologia, mas os resultados ainda são bem abaixo do esperado pelas organizações, em especial com a dependência cada vez maior de sistemas de informação para sua operação e gestão.

Um das causas identificadas para esta incidência de falhas é que equipes de desenvolvimento de *software* são focadas em tecnologias. Lyytinen (1999, p. 86) afirma que a alta incidência de projetos de *software* com falhas permite concluir que avanços em tecnologias não são suficientes para elevar a taxa de sucesso na implementação de sistemas de informação. Projetos continuam suscetíveis a falhas, porque as organizações falham em aprender com suas próprias experiências.

Desouza (2003, p. 2), observa que a engenharia de *software* é um domínio altamente orientado a pessoas e uso de conhecimento. Parreiras (2004, p. 1) destaca que o processo de desenvolvimento de *software* está ligado a gestão da informação e conhecimento nas organizações, uma vez que ele permite mapear, organizar, tratar e disseminar conhecimento. Assim, nos últimos anos uma das áreas de interesse e pesquisa para a melhoria na qualidade dos sistemas de informação é o uso de modelos de gestão da informação e conhecimento em processos de desenvolvimento.

Uma das atividades da engenharia de *software* é a fase de levantamento dos requisitos do sistema, em que são identificadas as funcionalidades que o sistema terá quando concluído. Sua importância para ao desenvolvimento de sistemas pode ser constatada na afirmação de Boehm e Basili (2001, p. 135), de que encontrar e consertar um problema de *software* após sua implantação pode ser até 100 vezes mais caro do que consertá-lo durante a fase de requisitos.

Em função disto, diversos estudos nos últimos dez anos tentam utilizar modelos de gestão e reuso da informação para reduzir os erros gerados no levantamento de requisitos, e melhorar a qualidade dos sistemas de informação entregues aos clientes.

A preocupação com aspectos relacionados à informação é reforçada por Tarapanoff (2006, p. 23), ao afirmar que ela é um fator determinante para a melhoria dos processos, produtos e serviços, tendo valor estratégico nas organizações.

Este trabalho é um estudo de caso, com a proposta de identificação dos elementos chave para a elaboração de um modelo de gestão da informação ao processo de desenvolvimento de *software* do Serpro (PSDS). A escolha da macroatividade de requisitos é porque ela marca o início de todo projeto de desenvolvimento de *software* e, segundo Leffingwell e Widrig (2000, p. 13), é a que apresenta os erros mais comuns, e os de menor custo para reparar.

O estudo e uso de modelos teóricos de gestão da informação está alinhado com as linhas de pesquisa do Departamento de Ciência da Informação e Documentação (CID/UnB), sendo a justificativa para o trabalho apoiada também no caráter interdisciplinar da ciência da informação, bem como na necessidade de sua utilização, seus modelos e referenciais teóricos, para complementar pesquisas e experiências práticas na disciplina de engenharia de *software*.

A pesquisa prática é justificada por Saracevic (1996, p. 41), ao afirmar que as relações entre a ciência da informação e a ciência da computação estão relacionadas aos aspectos práticos da área.

Assim, a pesquisa usará um modelo teórico de gestão da informação, adaptado e aplicado a prática do processo de desenvolvimento de *software* do Serpro, para identificar os pontos que afetam os processos relacionados a informação durante o desenvolvimento de sistemas.

3. REVISÃO DE LITERATURA

A revisão de literatura cobrirá tópicos de interesse da pesquisa: trabalhos correlatos; informação e ciência da informação; gestão da informação; desenvolvimento de sistemas e engenharia de *software*; o processo de desenvolvimento de *software* do Serpro (PSDS); gestão da informação e engenharia de *software*.

3.1. Trabalhos Correlatos

A utilização de recursos informacionais e de conhecimento em empresas de tecnologia da informação tem atraído a atenção de pesquisadores, com especial atenção na implementação de modelos de gestão do conhecimento e seu uso em atividades de desenvolvimento de sistemas de informação, com objetivo de melhorar a qualidade e a taxa de cumprimento de prazos para o desenvolvimento.

Estes trabalhos são concentrados nos cursos de administração e ciência da computação, e sua atenção principal é voltada para construção de modelos organizacionais para a gestão do conhecimento ou a implementação de ferramentas ou sistemas voltados para apoiar sua utilização.

Poucos trabalhos associam diretamente as áreas de desenvolvimento de sistemas e engenharia de *software* à gestão da informação, ou a ciência da informação.

Pesquisa realizada em janeiro de 2008 na base de teses e dissertações da Capes apresentou resultado descrito a seguir, com base nos argumentos de pesquisa citados, sempre no campo de pesquisa “Assunto”:

- Conhecimento e engenharia de *software*
 - Expressão exata - 7 resultados positivos, sendo 2 da área de ciência da informação;
 - Todas as palavras – 272 resultados positivos, maior parte deles da área de engenharia de *software*.
- Gestão do conhecimento e engenharia de *software*
 - Expressão exata – 2 resultados positivos, ambos da engenharia de *software*;
 - Todas as palavras – 39 resultados positivos, sendo apenas dois da área de ciência da informação.
- Gestão de informações e engenharia de *software*
 - Expressão exata – não houve resultados positivos;
 - Todas as palavras – 52 resultados positivos, sendo apenas um da área de ciência da informação.
- Gestão de informações e desenvolvimento de *software*
 - Expressão exata – não houve resultados positivos;
 - Todas as palavras – 169 resultados positivos, sendo a grande maioria na área de administração e ciência da computação.

A análise dos resultados e resumos mostrou que poucas pesquisas associam diretamente o assunto gestão da informação e engenharia de *software*, e a maior parte é da área de ciência da computação e engenharia de *software*. A abordagem do tema tem foco preferencial na avaliação e construção de ferramentas, com menor incidência na revisão dos processos de desenvolvimento de sistemas e na avaliação de resultados.

Uma possível justificativa para este cenário é a dificuldade da obtenção de indicadores e medidas para avaliação de projetos de desenvolvimento, pela sua dependência do contexto de execução – cultura da organização, perfil profissional das equipes, processo e tecnologias utilizadas e área de conhecimento do sistema a ser desenvolvido.

Um bom exemplo da utilização de modelos gerais é a tese desenvolvida por Oliveira (2008, p. 22), cujo objetivo é: “Desenvolver uma proposta de modelo de representação do capital intelectual de organizações que desenvolvem *software*.” O autor propõe a integração de modelo de representação do capital intelectual em empresas de *software*, mas reconhece a dificuldade de avaliar sua utilização, em função da diversidade e complexidade do ambiente.

Dentre as dissertações identificadas com assunto correlato pela busca de assuntos, está a de Queiroz (2001), da área de ciência da computação da

Universidade Federal da Paraíba. Ela tem como objetivo principal a proposição de um framework de gestão do conhecimento e diretrizes para sua implementação em empresas brasileiras de desenvolvimento de *software*. O trabalho tem foco nos processos organizacionais para todas as atividades, e não apenas no desenvolvimento.

Outra investigação identificada na busca é a de Espindola (2006), mas que tem como objetivo estudar o desenvolvimento distribuído de *software*, e propor uma arquitetura de informação para este tipo de projeto. Seu foco é no levantamento das informações necessárias e como estruturá-las, em preocupar-se com seu fluxo na equipes de desenvolvimento.

Este trabalho, de forma complementar propõe uma avaliação do fluxo e da gestão das informações utilizadas no processo de desenvolvimento de *software*, com foco na atividade de engenharia de requisitos. Pretende-se alcançar, com base em estudo de caso do Serpro, uma visão de como o uso de um modelo de gestão da informação pode auxiliar no fluxo de informações nas equipes, e promover a melhoria da qualidade dos sistemas desenvolvidos pela empresa.

3.2. Ciência da Informação e Informação

3.2.1. Ciência da Informação

A ciência da informação, como disciplina científica, tem pouco mais de 50 anos, embora diversos autores façam registros de eventos históricos que já seriam da área - como uso de tábuas de argila na Babilônica, a criação da biblioteca de Alexandria e a invenção da imprensa por Gutenberg em 1439.

Robredo (2003, p. 39) descreve a origem da ciência da informação em cinco períodos distintos, de acordo com acontecimentos relacionados com a disseminação e uso de informações, publicação de trabalhos e realização de conferências relevantes.

- Dos pioneiros da documentação à 1918;
- Período entre 1919 e o final da Segunda Guerra (1945)
- Do pós-guerra até a Conferência Internacional sobre Informação Científica (1958);
- De 1960 à década de 80;
- Da década de 90 aos dias atuais.

Alguns acontecimentos são destacados pelo autor, como os trabalhos realizados por Paul Otlet e Henri La Fontaine, advogados belgas, na evolução das práticas bibliotecárias e documentais, dentre os quais pode ser destacada a Classificação Decimal Universal (DCU), publicada em 1904, e que aprimorou a Classificação Universal de Dewey, publicada em 1876.

Sobre a obra de Otlet, Macedo (2005, p. 60) destaca o *Traité de Documentation*, publicado em 1934, onde é descrito um sistema de organização e disseminação de informações, com a premissa de que todo o conhecimento deveria estar acessível; e criada a idéia de repositórios corporativos com o registro sistemático das informações relevantes de determinada área do conhecimento.

Otlet descreve ainda uma “estação de trabalho”, que permitiria a consulta simultânea a vários registros, a busca de informações, e a possibilidade de fazer relacionamentos entre elas. A idéia é uma visão preliminar do que é hoje a principal rede de informações, a *World Wide Web*.

Em 1945, Vannevar Bush publica o artigo *As We May Think*, em que descreve a necessidade de organizar o grande volume de informações científicas geradas durante a guerra. Uma de suas grandes preocupações era: “[...] a inexistência de pessoal qualificado para conduzir a nova área que surge.”

O texto propõe ainda uma solução tecnológica o “memex”, uma prévia da adoção de ferramentas de tecnologia para acesso e recuperação de informações. Segundo Bush, o “memex”, seria: “[...] um dispositivo para uma pessoa armazenar todos os seus livros, registros e comunicações, e consultá-los com grande velocidade e flexibilidade para a manipulação de informações.”

A partir do final dos anos 50 são realizadas diversas conferências nas áreas de documentação e informações científicas, que impulsionam a discussão sobre a criação de uma nova disciplina. Dentre elas é possível destacar as conferências realizadas em 1961 e 1962 no *Georgia Institute of Technology*, onde surgem as primeiras formulações do que seria a ciência da informação.

Em 1968 o *American Documentation Institute (ADI)*, troca seu nome para *American Society of Information Science (ASIS)* e, no mesmo ano, H. Borko publica um artigo clássico – *Information Science: What is this?* - em que busca uma definição completa para a ciência da informação.

A partir da década de 70 cresce o número de publicações dedicadas a ciência da informação, e são publicados trabalhos relevantes para situá-la como disciplina científica, sendo possível destacar autores como Belkin, Brookes, Jesse Shera e Robert Taylor.

Nos anos 90 ganha força o conceito de interdisciplinaridade. Le Coadic (1994, p. 22) afirma que a interdisciplinaridade se traduz por colaboração entre diversas disciplinas, que leva a interações, i. e., uma certa reciprocidade, de forma que haja enriquecimento mútuo.

Saracevic (1995 e 1996) publicou estudos sobre o tema, destacando que a relação com outras disciplinas, como a ciência da computação e a biblioteconomia, é da natureza da ciência da informação. Ele afirma ainda que a explosão informacional é um problema social, surgido na pesquisa científica, mas presente hoje em todas as atividades da sociedade.

A evolução tecnológica, e o acesso quase ilimitado que a Internet oferece a todo tipo de informação, aumentaram a importância de estudos sobre armazenamento, recuperação e uso da informação, bem como métodos para mensurar seu valor para os usuários.

Ao longo deste período, vários autores divulgaram conceitos do que seria a ciência da informação. Após a realização das conferências do *Georgia Tech Institute*, Shera e Cleveland (1977) elaboraram a seguinte definição:

Ciência da informação é a que investiga as propriedades e comportamento da informação, as forças que regem o fluxo da informação e os meios de processamento da informação para um máximo de acessibilidade e uso. O processo inclui a origem, disseminação, coleta, organização, armazenamento, recuperação,

interpretação e uso da informação. O campo deriva ou relaciona-se com a matemática a lógica, a linguística, a psicologia, a tecnologia computacional, as operações de pesquisa, as artes gráficas, as comunicações, a biblioteconomia, a gestão e alguns outros campos.

A definição de Saracevic (1995, p. 2) inclui fatores sociais e o termo conhecimento:

Ciência da informação é o campo dedicado a pesquisa científica e prática profissional, que trata dos problemas de comunicação e registro do conhecimento entre humanos, em um contexto das necessidades e uso social, individual e/ou institucional da informação. No tratamento destes problemas ela deve aproveitar-se o máximo possível da moderna tecnologia da informação.

Le Coadic (1996, p. 56) define que a ciência da informação tem por objeto o estudo das propriedades gerais da informação (natureza, gênese e efeitos), bem como a concepção de dos produtos e sistemas que permitem sua construção, comunicação e uso.

Finalmente, Borko (1968, p. 1), elabora sua definição para ciência da informação como uma síntese de três definições formuladas por Robert Taylor⁶.

Ciência da informação é a disciplina que investiga as propriedades e o comportamento da informação, as forças que governam seu fluxo, e os meios de processamento da informação para otimizar seu acesso e uso. Está relacionada com corpo do conhecimento relacionado a origem, coleta, organização, armazenamento, recuperação, interpretação, transmissão, transformação e utilização da informação. Isto inclui a investigação da representação da informação em sistemas naturais e artificiais, o uso de códigos para a transmissão eficiente de mensagens, e o estudo de dispositivos e técnicas de processamento de informação, como computadores e seus sistemas.

Ele complementa afirmando que é uma disciplina interdisciplinar, derivada e relacionada a vários campos, como matemática, lógica, linguística, psicologia, tecnologia computacional, pesquisa operacional, comunicações, biblioteconomia, gestão e outros campos similares.

A ciência da informação apresenta componente de ciência pura, que indaga o assunto sem considerar sua aplicação, e um componente de ciência aplicada, que desenvolve produtos e serviços.

O autor destaca que a definição é complicada, mas isto acontece porque o tema é complexo e multidimensional, e a definição foi elaborada de forma abrangente. Ele afirma ainda que, na ciência da informação, a teoria e prática estão inexoravelmente relacionadas.

⁶ *An Introduction to the Vocabulary of Information Technology – Technical Communications*, de Samuel Miles, publicada em 1967. O título do capítulo é “*Professional Aspects of Information Science and Technology*”

A definição de Borko para ciência da informação será utilizada por este trabalho, por tratar os aspectos de fluxo e uso da informação, bem como a questão da interdisciplinaridade, e do desenvolvimento de produtos e serviços, pontos importantes para o estudo do comportamento da informação na atividade de desenvolvimento de *software*.

Cabe ainda destacar que a evolução da sociedade da informação e a evolução tecnológica têm exigido uma mudança de perspectiva da ciência da informação, e em suas relações com outras disciplinas. Saracevic (1996, p. 54) explica que esta evolução tem exercido pressão sobre a ciência da informação.

A criação de redes de informação e de ferramentas e técnicas de armazenamento e recuperação da informação alterou os papéis econômico e social de todas as atividades de informação. Neste cenário um dos aspectos mais importantes para as instituições é a capacidade de fazer a gestão de suas informações, e tornar seu uso ferramenta sustentável de competição em seus mercados.

3.2.2. Dados, Informação e Conhecimento

As expressões - “era da informação”, “sociedade da informação”, “era do conhecimento” - têm sido usadas com frequência para definir a sociedade moderna e o cenário de negócios. A informação é reconhecida hoje como um ativo valioso para as pessoas e para as instituições

Entretanto, ainda não há consenso sobre uma definição comum para o termo “informação”. Wurman (1991, p. 43) destaca que ele é usado de forma ambígua, e liberalmente empregado para definir diversos conceitos.

Segundo Sveiby (1994, p.1), a palavra “informação”, é derivada do latim “*informare*”, que significa “*dar forma a...*”. Ele comenta ainda que o termo é associado a conhecimento e comunicação, como na definição da Universidade de Oxford – “conhecimento comunicado em relação a um fato ou evento; inteligência; notícia”. Ele completa afirmando que ela pode referenciar fatos em si, ou a transmissão de fatos.

Wurman comenta ainda que, após a Segunda Guerra, o termo passou a ser usado para definir qualquer coisa transmitida por um canal de comunicação, e passou a fazer parte do vocabulário, aplicada a algo dito ou comunicado, fizesse ou não sentido para o receptor.

Já Le Coadic (1996, p. 5) define informação como um conhecimento inscrito, ou gravado, sob a forma escrita, oral ou audiovisual. É um significado transmitido a um ser consciente por meio de mensagem inscrita em um suporte.

Barreto (1996, p. 405) define informação como sendo: “Estruturas significantes com a competência de gerar conhecimento no indivíduo, em seu grupo ou na sociedade.”

Ele cria ainda o conceito de agregados de informação, onde são armazenadas as estruturas de informação. Os agregados representam os diferentes estoques que as estruturas significantes de informação podem assumir, como acervos em geral ou bases de dados.

Na tentativa de buscar um significado comum, Robredo (2003, p. 1) apresenta pelo menos uma dezena de definições para “informação”, citando autores, glossários e dicionários. Uma das definições apresentadas é a do *Harrods Librarian Glossary of Terms Used in Librarianship, Documentation and Book Crafts and Reference Book*, que define informação como: “Um conjunto de dados organizado de forma compreensível, registrado em papel ou outro meio e suscetível de ser comunicado”.

Entretanto, sua análise chega a conclusão similar a de Wurman: “[...] fica a impressão de que tudo, e seu contrário pode, de alguma forma, se relacionar com informação.

Para complementar as definições apresentadas Robredo (2003, p. 9) descreve características da informação, ao afirmar que ela é suscetível de ser:

- Registrada (codificada) de várias formas;
- Duplicada e reproduzida;
- Transmitida por diversos meios;
- Conservada e armazenada em suportes diversos;
- Medida e quantificada;
- Adicionada a outras informações;
- Organizada, processada e reorganizada segundo diversos critérios;
- Recuperada quando necessário, de acordo com regras pré-estabelecidas.

As definições citadas usam os termos dados e conhecimento, sendo oportuno diferenciá-los, pois é comum que sejam usados para representar o mesmo fenômeno. Uma solução adotada é tratá-los como uma relação contínua, onde há aumento de valor e da contribuição humana, como Davenport (1997, p. 9), ao usar uma definição e características para cada um dos termos:

- **Dados** – simples observações de estados do mundo real. São estruturados, facilmente capturados por máquinas, geralmente podem ser quantificados e podem ser transferidos facilmente;
- **Informação**⁷ - dados dotados de relevância e propósito. Requer análise e há necessidade de mediação humana;
- **Conhecimento** – informação que pode incluir reflexão, síntese e contexto da mente humana. É difícil de estruturar, difícil de transferir e capturar em máquinas.

O referido autor explica que informação também é o veículo que usamos para expressar e comunicar conhecimento, em negócios e em nossas vidas. Ela tem mais valor que dados e, ao mesmo tempo, mais ambiguidade

Já Roberts (2000, p. 2), faz a seguinte distinção entre os termos. Dados são definidos como uma série de observações, medidas ou fatos, na forma de números, palavras, sons ou imagens. Dados sozinhos não têm significado, mas são matéria prima para produção de informação.

⁷ Davenport usa definição adaptada de Peter Drucker, do artigo “The Coming of the New Organization” – Harvard Business Review, jan-feb 1988, pp. 45-53

Informação é definida como dados organizados em algum padrão, e relacionada a algum contexto para ter significado; e conhecimento é a aplicação e uso produtivo da informação. Envolve a compreensão por experiência ou aprendizagem.

Ainda em relação a informação e conhecimento, Nonaka e Takeuchi (1995, p. 58) ressaltam que ambos são dependentes de contexto, da situação, dizem respeito a significados, e são criados de forma dinâmica em interações sociais entre pessoas. Para eles a informação é um meio necessário (material) para elicitación e representação do conhecimento. O conceito de conhecimento usado por eles é (grifo no original): “[...] processo humano e dinâmico de justificar crenças pessoais com relação à *verdade*.”

Os referidos autores ressaltam as seguintes diferenças do conhecimento em relação à informação: o conhecimento refere-se a crenças e compromissos, é função de atitude, perspectiva e intenção e, mais importante, está essencialmente relacionado a ação.

Wiig (1999, p.6) oferece um interessante ponto de vista quando afirma que, apesar de parecer que há um contínuo de sinais entre dados, informação e conhecimento, quando examina-se os conceitos, e os processos que os criam, encontramos descontinuidades que fazem informação fundamentalmente diferente de conhecimento.

Para ele informação consiste de fatos e dados organizados para caracterizar uma situação particular, condição desafio ou oportunidade.

Por outro lado, ele define que conhecimento é possuído por humanos, como verdades e crenças, perspectivas e conceitos, julgamentos e expectativas. O conhecimento é usado para receber informação – reconhecer e identificar; analisar, interpretar e avaliar; adaptar, planejar e implementar – para agir

Wiig afirma ainda que a descontinuidade entre informação e conhecimento é causada pela forma como o conhecimento é criado a partir da informação recebida.

Finalmente, Wilson (2002, p. 2) ao analisar os três termos, afirma que, tudo o que está fora da mente das pessoas pode ser classificado como dados, se consiste de simples fatos; e de informação, se os dados forem complementados com um contexto de relevância. Para o autor, conhecimento é definido como o que sabemos, e envolve o processo mental de compreensão, entendimento e aprendizagem que ocorre na mente, e somente nela, embora ocorra a interação com o ambiente externo e com outras pessoas. Assim, sempre que uma pessoa expressa algo que sabe, ela o faz por meio de mensagens – oral, escrita, gráfica ou gestual – e estas mensagens não carregam conhecimento, mas constituem informação, que outra pessoa pode assimilar, compreender e incorporar a suas próprias estruturas de conhecimento.

Este conceito oferece um contraponto aos argumentos de Nonaka e Takeuchi (1995, p. VIII), quando classificam o conhecimento em dois tipos: explícito e tácito. O conhecimento explícito pode ser articulado em linguagem formal, expressões matemáticas, especificações e manuais, e pode ser facilmente codificado e transmitido como mensagem. O conhecimento tácito é difícil de ser representado em linguagem formal. É o conhecimento pessoal, embutido na

experiência individual, e envolve fatores intangíveis, como crenças pessoais e sistema de valores.

Embora esta classificação tenha se tornado muito popular nos últimos 10 anos, cabe destacar ressalva feita por Polanyi *apud* Roberts (2000, p. 3), ao argumentar que:

Enquanto o conhecimento tácito pode existir por si próprio, o conhecimento explícito depende da possibilidade de ser tacitamente entendido e aplicado. Dessa forma, todo o conhecimento é tácito, ou baseado em conhecimento tácito. Conhecimento totalmente explícito é impensável.

Para este trabalho serão utilizados os seguintes conceitos:

- **Dados** - observações, medidas ou fatos do mundo real, na forma de números, palavras, sons ou imagens. São facilmente representados, armazenados e transferidos;
- **Informação** – dados, dotados de relevância em um contexto, e conhecimento codificado em algum suporte - escrito, oral ou audiovisual. É um significado transmitido a um ser consciente por meio de mensagem inscrita em um suporte;
- **Conhecimento** – o que sabemos, resultado de processo mental de compreensão, entendimento e aprendizagem. Existe apenas na mente das pessoas e está ligado ao processo de ação.

3.2.3. Conclusão

A evolução do uso da informação permitiu novas abordagens para a ciência da informação, retratadas nos conceitos elaborados a partir dos anos 60, como os de Saracevic, Le Coadic e Borko. Além dos aspectos relacionados a suas propriedades, a atenção dos pesquisadores foi voltada para o estudo e compreensão dos fluxos e os meios de coleta, armazenamento, processamento e uso.

A disciplina ganhou um contexto complexo e multidimensional, com sua característica interdisciplinar e forte relação com outras disciplinas científicas, como a biblioteconomia, ciência da computação e gestão, e onde os aspectos teóricos e práticos devem ser considerados. Ela ganha destaque na mesma proporção em que a informação é tratada hoje como um ativo valioso na sociedade moderna, a ponto de ser referenciada como a “sociedade da informação”. Marchiori (2002, p. 73), destaca que a informação é valorizada como recurso, e define a competitividade de pessoas, grupos, produtos, serviços e atividades.

O termo “informação” é usado de forma ambígua e liberal, englobando os conceitos de dados, informação, mensagens e conhecimento. Embora esta liberalidade não seja relevante para a maior parte das pessoas, ela é importante para as organizações. O entendimento das características e diferenças entre eles

deve ser aplicado no tratamento de seu fluxo e principalmente no processo de gestão e uso das informações.

A característica principal de “dados” é sua natureza de observação ou fato do mundo real, facilmente estruturado, armazenado e transferido. Já informação e conhecimento têm uma diferença fundamental em relação a dados, que é a dependência de contexto e interpretação.

A informação está sempre codificada em algum suporte, diferente do conhecimento, presente apenas na mente das pessoas. Do conhecimento podemos afirmar ainda que é resultado de processo mental de compreensão e aprendizagem e, segundo Nonaka e Takeuchi, está sempre associado a algum tipo de ação.

3.3. Gestão da Informação

Wilson (2002, p. 1) define gestão da informação como:

A aplicação de princípios de gestão para a aquisição, a organização, o controle, a disseminação e o uso de informação relevante para o efetivo funcionamento das organizações de todos os tipos [...]. Gestão da informação está relacionada com o valor, a qualidade, a propriedade, o uso e a segurança da informação no contexto do desempenho da organização.

Ainda em relação a gestão da informação, Tarapanoff (2006, p.22) destaca que seu principal objetivo é identificar e potencializar os recursos informacionais de uma organização e sua capacidade de informação; ensiná-la a aprender e adaptar-se às mudanças.

Segundo ela, a evolução da gestão da informação, de um foco inicial de gestão de documentos e dados para recursos informacionais, gerou melhores resultados em relação a eficiência operacional, evitando desperdícios e automatizando processos. A autora complementa ainda afirmando que a gestão dos recursos informacionais tem como principal finalidade o acompanhamento eficiente de processos, o apoio à tomada de decisões e a obtenção de vantagem competitiva.

Este trabalho considera os modelos para ciclo de desenvolvidos por Davenport e Choo na década de 90. Eles abordam os principais aspectos relacionados a necessidades e uso de informações, e consideram a relação e envolvimento com ferramentas de tecnologia da informação, ponto importante para o desenvolvimento do trabalho.

3.3.1. Ciclo da Gestão da Informação – Davenport

Para Davenport (1997, p. 7): “As organizações não tem idéia do que sabem, e do que precisam saber”, e destaca ainda que elas não podem suportar tamanha incompetência em informações, embora o custo de ter a informação errada, ou não usar a informação correta, seja difícil de medir.

O autor criou a metáfora de “ecologia da informação” ao examinar as principais propriedades e a dinâmica do ambiente de informações de uma organização; e definiu que a ecologia da informação de uma organização é formada por três ambientes interconectados e sobrepostos, mostrados na Figura 2: ambiente de informações, o ambiente organizacional o circundando e, influenciando ambos, o ambiente externo.

O ambiente informacional é o núcleo da abordagem ecológica para a gestão da informação, sendo composto de seis componentes principais: estratégia, política, cultura/comportamento, pessoal, processos e arquitetura de informação.

Devem ser tratadas as questões do papel da informação na estratégia de negócios da organização, seu modelo de governança, pessoal especializado em informações, processos relacionados a informação, a arquitetura informacional e a cultura a ser criada, em especial no compartilhamento de informação dentro da organização.

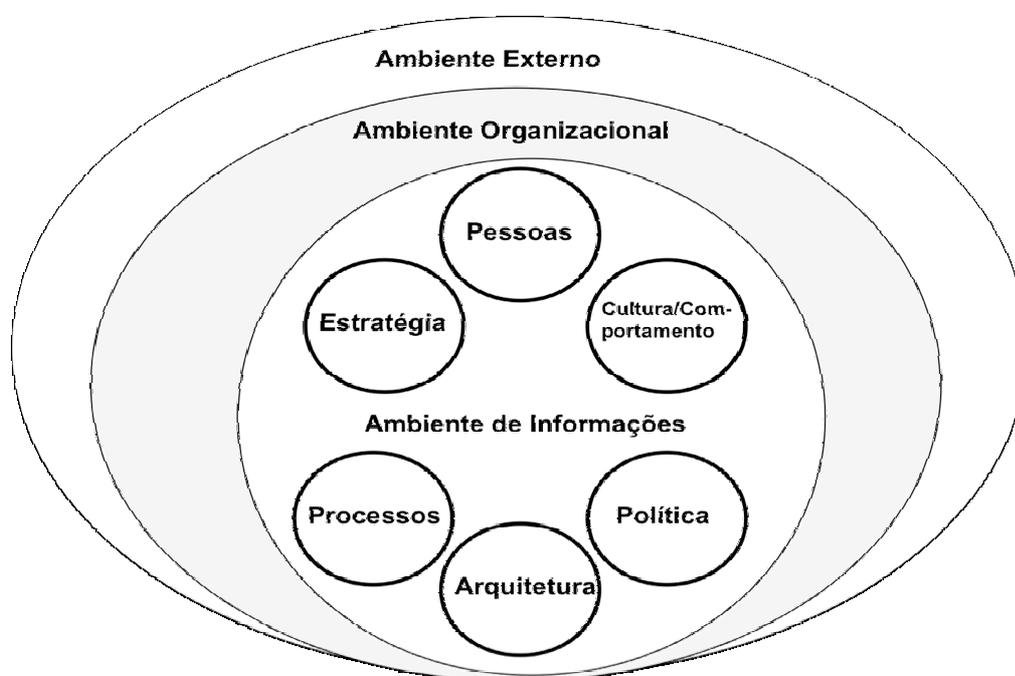


Figura 2 - Modelo Ecológico para Gestão da informação

Fonte: Davenport, 1997, p. 34

Davenport descreve o ciclo de gestão da informação como parte dos processos de informação da organização. Ele define gestão da informação como um conjunto estruturado de atividades que tratam a forma que as organizações capturam, distribuem, e usam informação e conhecimento. O referido autor destaca que a gestão da informação deve incluir toda a cadeia de valor da informação, e apresenta duas versões de um processo de gestão da informação.

Na primeira delas – Davenport (1993, p. 98) - define um processo composto de seis etapas:

- Identificação de necessidades e exigências de informação;
- Coleta e aquisição da informação;
- Categorização e armazenamento da informação;
- Compactação e formatação da informação;
- Disseminação e distribuição da informação;
- Análise e uso da informação.

A outra versão de processo de gestão da informação descrita por ele – Davenport (1997, p. 135) - apresenta o processo em quatro etapas, conforme descrito na Figura 3: determinação de requisitos, captura, distribuição e uso.

Entretanto, o detalhamento de cada uma delas mostra que foram consideradas todas as seis etapas anteriormente citadas, e que houve apenas uma reorganização do conteúdo e método de representação.

Segundo o autor, a etapa de levantamento de necessidades é normalmente negligenciada e, quando o intercâmbio de informações é apenas formal, como geralmente acontece, o resultado é uma informação de má qualidade.

Ele explica ainda que o processo de gestão de informação é geralmente tratado como apoio aos processos operacionais básicos da organização, e sujeito a cortes em seus recursos em momentos de crises, com a perda de pessoal especializado.

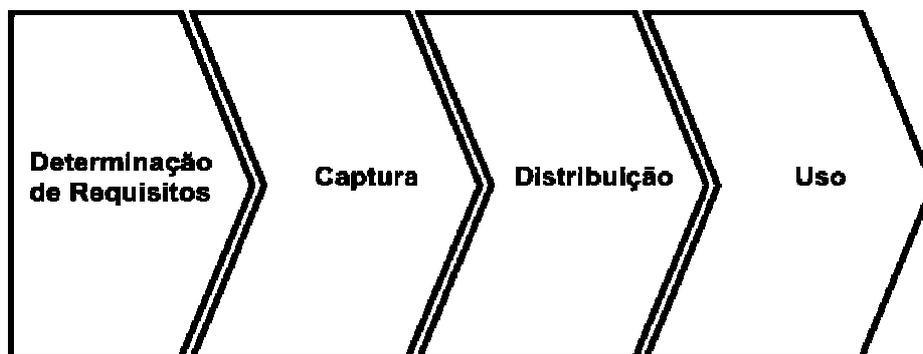


Figura 3 - Processo de gestão da informação

Fonte: Davenport, 1997, p. 135

3.3.2. Ciclo da Gestão da Informação – Choo

Uma outra abordagem da gestão da informação é apresentada por Choo (2002, p. 24), ao descrever um ciclo que tem como objetivo principal o tratamento das fontes de informação e da capacidade informacional da organização, para permitir que ela aprenda e se adapte a mudanças de seu ambiente. Ele afirma que a gestão da informação deve considerar o contexto social do uso da informação, uma vez que ela ganha significado e propósito pelo compartilhamento mental e afetivo.

A gestão da informação pode ser entendida como um conjunto de processos que suporta as atividades de aprendizagem organizacional. Os seis processos

descritos por Choo foram baseados no ciclo proposto por Davenport em 1993, mas com títulos e descrição adaptadas para um novo cenário de negócios.

O ciclo de gestão da informação e os processos propostos por Choo são mostrados na Figura 4 e descritos a seguir: necessidades de informação, aquisição da informação, organização/armazenamento, produtos e serviços informacionais, distribuição e uso da informação.

O ciclo começa com o comportamento adaptativo da organização, ações que criam informações, interagem com outras organizações e alteram o ambiente, gerando novas mensagens que devem ser avaliadas. A medida que novas informações são recebidas e processadas, elas geram necessidades de novas informações, reiniciando o ciclo.

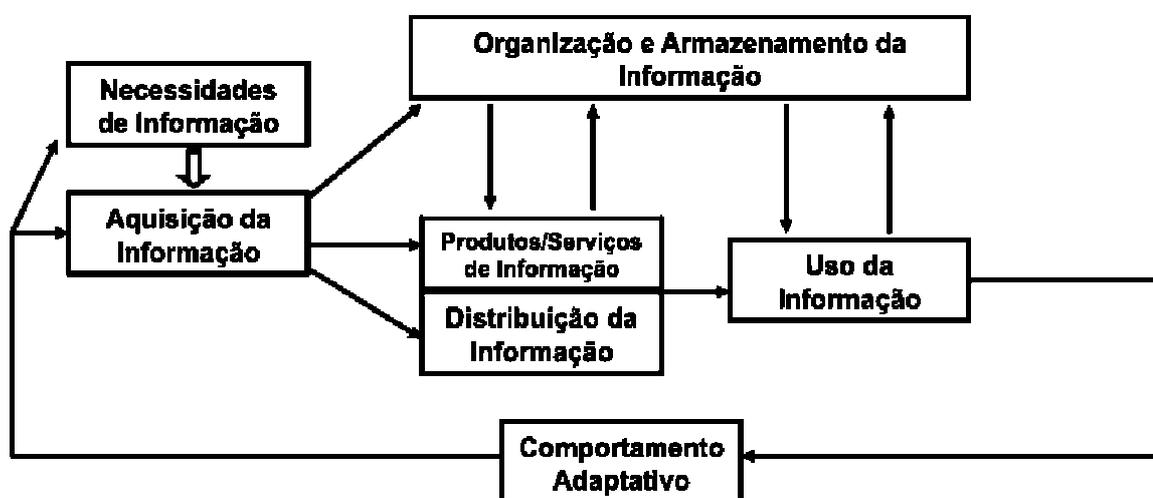


Figura 4 - Ciclo da gestão da informação

Fonte: Choo - 2002, p. 24

Necessidades de Informação

Choo (2006, p. 406) afirma que a avaliação precisa das necessidades de informação dos grupos e indivíduos da organização é o primeiro passo para a criação de uma estratégia de gestão da informação. Segundo ele:

[...] as necessidades de informação são condicionais, dinâmicas e multifacetadas, e uma especificação completa só é possível dentro de uma rica representação do ambiente total em que a informação é usada.

A identificação das necessidades de informação deve ser suficientemente completa na representação das reais necessidades dos usuários, e deverá ser elicitada junto as pessoas, em um processo de comunicação complexo, já que muitas pessoas acham difícil expressar suas necessidades de informação de forma de forma clara.

Martin (1976, p. 488) destaca que necessidade é um conceito escorregadio, e difícil de identificar. Muitas pessoas nunca pararam para refletir nas suas necessidades, outras não conseguem articular o que desejam, ou são tímidas demais para isto. É comum que o profissional da informação tenha que

compensar esta situação, com sugestões de necessidades para o usuários, como o uso de questionários e entrevistas.

Britain *apud* Bettiol (1990, p. 63) refere-se ao termo necessidade como abstrato, e uma combinação das seguintes necessidades: expressas pelos usuários; necessidades que o usuário não consegue expressar; necessidades presentes, expressas ou não; e necessidades futuras e potenciais.

As necessidades individuais de informação surgem de problemas, incertezas e ambiguidades encontradas em situações e experiências específicas. Ela ocorre de forma que a pessoa possa usar a informação para iniciar uma ação.

Le Coadic (1996, p. 41) identifica duas grandes classes de necessidades de informação. A necessidade de informações em função do conhecimento, derivada do desejo de saber; e a necessidade de informações em função da ação, derivada de necessidades exigidas para a realização de atividades pessoais ou profissionais.

Ele destaca que a análise das necessidades de informação deve responder as seguintes perguntas:

Quem necessita da informação? Que tipo de informação? Para qual grupo de pessoas? Por que precisam dela? Quem decide quanto a essa necessidade? Quem seleciona? Que uso é dado ao que é fornecido? Que consequências resultado desse uso para o indivíduo, o grupo, a instituição e a sociedade em seu conjunto?

Wilson (2002, p. 8) identifica aspectos que devem ser tratados no processo de identificação da necessidade de informações. O primeiro deles é que as necessidades mudam de acordo com alterações no ambiente dos usuários, interno e externo à organização.

Por fim, a estrutura proposta por Taylor *apud* Choo (2006, p. 406) sugere uma forma sistemática de analisar as necessidades de informação em uma organização: identificar os grupos de usuários da informação, reconhecer os tipos de problemas que eles enfrentam, examinar seu ambiente profissional e social, e entender as formas pelas quais eles consideram que um problema foi resolvido.

Assim, como o recebimento e processamento gera novas necessidades, é importante que os gestores de informação tratem a identificação de necessidades como uma atividade contínua, avaliando retorno das informações recebidas e monitorando as mudanças de prioridade da organização.

Aquisição da Informação

A aquisição de informações tornou-se uma atividade crítica, mas muito complexa, em função da multiplicidade de fontes e formatos. Ela deve atender a duas demandas: ter grande amplitude para atender as necessidades organizacionais de refletir os eventos e mudanças do ambiente externo; manter uma seleção das mensagens a serem examinadas, em função de limitações da capacidade de atenção e cognitiva.

Devem ser observadas as características relacionadas a ampliação ou restrição das fontes de informação. A ampliação da variedade de fontes, como a busca de conceitos em outras áreas, auxilia na busca de soluções criativas para

os problemas. Por outro lado, a redução da variedade facilita a concentração de esforços no tratamento do assunto.

Choo (2006, p. 411), afirma que uma forma de tratar a variedade de informações é envolver o máximo de pessoas possível na captura de informações, criando uma rede de coleta de informações. As pessoas sempre serão consideradas como fontes mais valiosas de informação, por sua capacidade de filtrar e sumarizar, ressaltar os aspectos importantes e interpretar ambiguidade.

Ainda segundo o autor, vários critérios podem influenciar a seleção e o uso das fontes de informação. Pesquisas mostram que a acessibilidade de uma fonte de informação é mais importante que sua qualidade, i. e., muitos usuários preferem fontes locais, mas que estejam acessíveis, mesmo que elas não sejam as mais confiáveis.

Em relação ao processo de tomada de decisão, as organizações utilizam regras, normas e preferências para atenuar a variedade de informações – por exemplo, designando fontes e tipos de informação a ser coletada – para facilitar o processo de busca e escolha. A seleção e o uso das fontes de informações deve ser planejado e continuamente monitorado e avaliado, como todos os recursos vitais para a organização.

Organização e Armazenamento da Informação

A informação que é capturada ou criada tem que ser organizada e armazenada para facilitar seu compartilhamento e recuperação. Ela representa um componente significativo e frequentemente consultado da memória organizacional, e será recuperada para interpretar situações, solucionar problemas e tomar decisões.

Choo (2006, p. 416) destaca a importância do processo para a organização, ao afirmar que:

A maneira como a informação é armazenada reflete como a organização percebe e representa seu ambiente, inclusive a maneira como denomina suas entidades, especifica os relacionamentos, acompanha transações e avalia desempenhos.

O armazenamento é uma atividade em que a tecnologia da informação tem grande impacto, seja por sua capacidade de absorver grandes quantidades de dados, seja porque hoje a grande maioria dos documentos já é elaborada em computadores.

Os sistemas tornaram-se muito eficientes do ponto de vista de captura e armazenamento de dados estruturados, como registros de clientes e transações efetuadas, geralmente compostos por campos pré-definidos, cujos valores são sempre entre intervalos conhecidos.

Entretanto, estes mesmos sistemas ainda são muito ineficientes para responder novas perguntas dos usuários e oferecer informações para suporte a atividades relevantes da operação, como atendimento aos clientes, ou para o processo de tomada de decisão.

Desta forma, os aspectos de busca e recuperação da informação são relevantes. Araújo Júnior (2007, p. 75), descreve que a função dos sistemas de

recuperação da informação parte da seguinte premissa: “[...] levar ao usuário/demandante o documento certo que irá satisfazer a sua necessidade específica de informação.”

Os ativos de informação de uma organização não estão confinados em dados transacionais, e alguns dos mais valiosos encontram-se em relatórios de projetos, correspondências internas e arquivos de áudio. A necessidade de tratar informações do ambiente externo exige a incorporação de suportes diversos, como filmes, fotografias e material disponível na Internet.

Para todos eles deve haver um bom sistema para recuperação de informações que, segundo Robertson *apud* Araújo Júnior, deve ter as seguintes características:

Efetividade – significa quão bem ele desempenha uma tarefa delegada;

Benefício – o quanto se ganha com sua utilização em determinado contexto; e

Eficiência – custo de toda a operação, isto é, equilíbrio entre custo e benefício.

As consequências de perda de informações, ou a incapacidade de encontrá-las quando necessário, têm levado as organizações a buscar sistemas de armazenamento e recuperação de informações que atendam as características acima citadas, além da flexibilidade para facilitar a utilização. O sistema deve ser capaz de armazenar informações de fontes diversas, suportar múltiplas visões dos usuários e conectar itens que são logicamente relacionados.

Produtos e Serviços de Informação

O objetivo de desenvolver produtos e serviços de informação não é apenas oferecer informação relevante para o usuário, mas oferecer a informação de forma que aumente seu potencial de uso. Os produtos e serviços devem entregar e apresentar a informação de forma que seu conteúdo, formato e outros atributos atendam aos requisitos de determinada situação, e que afetam a resolução de um problema.

Taylor (1986, p 50) destaca que os produtos e serviços, e os sistemas de informação em geral, devem ser desenvolvidos como um conjunto de atividades que adiciona valor a informação sendo processada, e levar os usuários a ter ações mais efetivas e melhores decisões. O autor identifica cerca de 22 atividades que podem adicionar valor a informação, e que são divididas em cinco categorias, ou critérios de seleção dos usuários.

- Facilidade de uso – reduz a dificuldade de uso do produto ou serviço e inclui a capacidade de apresentar e arranjar os dados para facilitar a busca e seleção, auxiliando o usuário em obter respostas e ganhar experiência com o sistema;
- Redução de ruído – é obtida pela exclusão da informação não desejada, e inclui o valor adicionado por tecnologias como indexação ou técnicas de gestão de dados para auxiliar o usuário em reduzir o universo de informações a um grupo de possível utilidade – que o autor chama de acesso intelectual;

- Qualidade – critério dos usuários que avalia excelência do produto ou serviço, com a transferência de informações sem erro e cobrindo todos os aspectos desejados, com um serviço que apresente performance consistente ao longo do tempo;
- Adaptabilidade – capacidade dos serviços de apresentar respostas às necessidades dos usuários em seu ambiente de trabalho. Inclui valor adicionado pela possibilidade de atender demandas específicas e suportando formas diversas de trabalhar os dados de forma interativa;
- Redução de custo e tempo – são valores percebidos pelos usuários, baseados na velocidade de obter respostas, e valor economizado por eles no processo.

Distribuição da Informação

Choo (2006, p. 414) explica que este é o processo pelo qual a organização dissemina e compartilha informações de diversas fontes. Uma ampla distribuição da informação promove maior aprendizado organizacional, auxilia o processo de recuperação de informações relevantes e permite a criação de novas informações ou conhecimentos pela associação de informações de áreas diversas.

Ele destaca que a entrega da informação deve ser feita por meios que estejam de acordo com os hábitos e preferências dos usuários, de acordo com o princípio – “...a informação certa para a pessoa certa, no formato, local e hora certos.”

Um ponto importante da distribuição é o compartilhamento de informações, com objetivo de permitir que informações de múltiplas fontes, e processadas por vários usuários, sejam trabalhadas para gerar novas visões e soluções.

O referido autor destaca ainda que, em processos de solução de problemas, as pessoas trabalham em equipes que combinam experiência e especializações. Elas compartilham informações em diálogos de grupos, extraem informações e conhecimento de fora da organização e descobrem novos conceitos com o uso de analogias. Durante a tomada de decisão o fluxo de informações é regulado por rotinas e regras que determinam quem tem acesso a informação, quem pode solicitá-las ou alterá-las.

É também mostrada a necessidade de acabar com a separação entre fornecedores e usuários – ambos devem colaborar como parceiros na disseminação e adição de valor à informação. Para encorajar os usuários a serem participantes ativos no processo, deve ser facilitada a sua participação como comentarista ou avaliador, com a possibilidade de redirecionar e enviar a outros a informação que ele recebeu.

Uso da Informação

Choo (2006, p. 106) relata que o uso da informação é um conceito difícil de definir satisfatoriamente, e utiliza uma forma pragmática: o indivíduo seleciona mensagens entre um grupo maior de mensagens que recebe. A escolha é feita quando ele percebe uma relação significativa entre o conteúdo da mensagem e a tarefa ou problema que tem em mãos.

Ainda segundo o autor, o uso da informação: "... envolve a seleção e o processamento da informação, de modo a responder uma pergunta, resolver um problema, tomar uma decisão, negociar uma posição ou entender uma situação."

Ele comenta ainda que o processo é social e interativo, para gerar novas informações ou conhecimento, e levar a tomada de decisão. Durante o processo nova informação pode ser recebida e mudar a percepção do usuário sobre o problema, dando lugar a novas incertezas. O resultado do processo é uma mudança no estado do conhecimento do usuário, ou de sua capacidade de agir.

No processo de uso a utilidade e valor da informação serão medidos pela importância do assunto, requisitos e expectativas, que dependem da atividade do usuário e de seu contexto de trabalho. Estudos feitos por Taylor analisam o uso da informação, seus ambientes, fluxos e terminam por definir classes de uso da informação.

Taylor (1986, p. 24) propõe que os contextos de trabalho sejam definidos como ambientes de uso da informação, que:

Podem ser definidos como conjunto de elementos que: a) afetam o fluxo e o uso das mensagens de informação que entram, circulam e saem de qualquer entidade; e b) determinam os critérios pelos quais o valor das mensagens de informação pode ser julgado nestes contextos.

Os elementos destes ambientes podem ser agrupados em quatro categorias: grupos de pessoas, dimensão do problema, ambiente de trabalho e pressupostos para a solução dos problemas. Choo descreve assim as categorias definidas por Taylor:

- Grupos de pessoas - possuem atitudes e pressupostos comuns sobre a natureza de seu trabalho, que afetam seu comportamento na busca e uso da informação. Os pressupostos podem ser resultado de treinamento profissional, ou assimilados informalmente pela participação em grupos de trabalho;
- Dimensões do problema - são características que ocorrem em determinado grupo de pessoas, que costumam mudar com o tempo, e determinam os critérios para julgar a relevância e o valor da informação para uma determinada categoria de problemas. Taylor apud Choo (2006, p.95) afirma que: "... cada ambiente de uso de informação tem um tipo diferente de problema, gerado por seu ambiente particular, e pelas exigências de sua profissão, ocupação ou estilo de vida";
- Ambiente de trabalho – características físicas e sociais da organização, ou unidade em que um grupo de pessoas trabalha, e que influenciam as atitudes em relação à informação, tipo de informação requerida, seu fluxo e disponibilidade. O autor destaca que a percepção das pessoas sobre o papel e a importância da informação é influenciada pelo estilo e cultura da organização, e que o fluxo e a disponibilidade da informação podem ser afetados por aspectos do ambiente de trabalho, como hierarquia e localização das fontes de informação;

- Solução dos problemas – os pressupostos são percepções compartilhadas pelas pessoas sobre o que constitui a solução de seus problemas. Choo afirma que eles orientam a busca e uso da informação, ao criar expectativas sobre especificidades da informação necessária para resolver os problemas. As percepções e previsões das pessoas controlam indiretamente o tempo e esforço gasto na busca, e a quantidade de informação necessária. Um exemplo clássico é o dos executivos, que nem sempre procuram as melhores soluções, mas consideram um problema resolvido quando encontram uma solução suficientemente boa.

Taylor *apud* Choo (2006 p. 108) propõe uma classificação de usos da informação, gerada por necessidades percebidas pelos usuários em determinadas situações. São definidas oito classes:

- Esclarecimento – informação é usada para criar contexto ou dar significado a uma situação. É normalmente usada para responder perguntas;
- Compreensão do problema – a informação é usada de maneira específica, que permitirá melhor compreensão do problema;
- Instrumental – informação que permite o indivíduo saiba o que, e como fazer. Instruções são a forma mais comum desta classe;
- Factual – informação é usada para descrever a realidade, determinar fatos ou acontecimento. Seu uso depende da qualidade (precisão e confiabilidade) da informação disponível;
- Confirmativa – usada para verificar outra informação; seu uso envolve a busca de outra opinião. Se a informação existente não for confirmada, o usuário pode tentar reinterpretar a informação, ou decidir qual a fonte mais confiável;
- Projetiva – usada para prever o que provavelmente vai acontecer no futuro. Seu uso envolve previsões, estimativas e probabilidade;
- Motivacional – informação usada para iniciar ou manter o envolvimento do usuário, para que ele prossiga em um curso de ação;
- Pessoal ou política – usada para criar relacionamentos ou promover melhoria de reputação, satisfação pessoal ou *status*.

Choo complementa observando que a atitude do indivíduo – educação, treinamento e experiência passada, entre outros - em relação à informação é um dos elementos que mais influenciam o processo de uso da informação.

Um desafio da gestão da informação é projetar estruturas e processos de informação que tratem de forma flexível os elementos do ambiente de uso da informação, de forma que eles facilitem a busca e uso no apoio a solução de seus problemas, e no processo de tomada de decisão.

3.3.3. Conclusão

Os conceitos de gestão da informação apresentados destacam a necessidade da integração das informações, e os processos associados, à estratégia organizacional. Ainda que autores como Wilson e Moresi considerem seu entendimento ainda ambíguo, e muitas vezes referenciada como gestão de dados ou de conhecimento, Moresi (2000, p. 15) a considera relevante para a estratégia em todos os setores.

Para este trabalho será considerado o seguinte conceito para gestão da informação: aplicação de princípios de gestão para tratar os processos de aquisição, organização, disseminação e uso da informação, para permitir que a organização aprenda e adapte as mudanças de seu ambiente. Ela deve suportar a aprendizagem organizacional

Wilson (2002) explica ainda que, no contexto da ciência da informação, a gestão da informação está identificada com os trabalhadores da informação, cuja percepção da informação inclui dados, inteligência organizacional, inteligência competitiva e fontes externas, bem como os recursos de tecnologia utilizados. Neste contexto, a gestão da informação tem maior preocupação com o significado da informação para os usuários, e com aspectos de sua recuperação.

Choo (2002, p.225) também destaca a importância dos usuários da informação, ao afirmar que eles devem ser os atores centrais nos processos de gestão, trabalhando com especialistas em informação e especialistas em tecnologia da informação.

O ciclo de gestão proposto por Choo é composto por seis processos: necessidades de informação; aquisição; organização e armazenamento; produtos e serviços; distribuição e uso da informação.

Ele se preocupa com os aspectos relacionados ao fluxo da informação, influências do ambiente informacional, os usuários e suas expectativas. As necessidades de informação são dinâmicas, alteradas pelo próprio uso da informação. Sua identificação deve analisar as pessoas envolvidas, o ambiente em que elas trabalham e como resolvem problemas.

O processo de aquisição da informação deve considerar um número adequado de fontes, bem como a variedade de informação, para que a complexidade dos ambientes seja refletida. As informações capturadas serão guardadas por um sistema de armazenamento, uma extensão da memória organizacional, e que deve possibilitar a sua recuperação de forma flexível e eficiente.

Os serviços de informação devem preocupar-se com a oferta de informações relevantes para os usuários, de forma que aumente seu potencial de uso. A abordagem dos produtos e serviços deve tratar a agregação de valor à informação que será processada, de forma que ela possa ajudar o usuário ter ações mais eficazes e melhorar o processo de tomada de decisões.

A distribuição da informação promove seu compartilhamento, e o processo deve permitir que os usuários participem de forma ativa, facilitando assim o uso da informação pelas equipes. O uso da informação prevê a sua seleção e processamento, para resolver um problema ou tomar uma decisão. A definição de

ambientes de uso da informação trata sua formação por quatro elementos: grupos de pessoas, dimensões de problemas, ambiente de trabalho e solução de problemas.

Este uso pode ainda ser agrupado em oito categorias: esclarecimento, compreensão do problema, instrumental, factual, confirmativa, projetiva, motivacional e pessoal/política.

Este trabalho utilizará o ciclo de gestão da informação proposto por Choo, que apresenta uma visão completa do ciclo de vida da informação e tem maior aderência ao ambiente de informações de organizações de tecnologia, caso do Serpro, ambiente a ser tratado nesta pesquisa. Ele também engloba a visão ecológica e abrangente proposta por Davenport, e reforça a necessidade de foco no comportamento das pessoas, e não no uso de tecnologia como solução principal para os problemas relacionados ao uso da informação.

3.4. Desenvolvimento e Engenharia de *Software*

3.4.1. Desenvolvimento de *Software*

O desenvolvimento de *software* é uma atividade cada vez mais importante para as organizações, na medida em que os sistemas de informação tornaram-se críticos para os negócios. Os sistemas de informações são a principal forma de relacionamento entre muitas empresas e seus clientes.

Alguns exemplos conhecidos são a Gol Linhas Aéreas, Submarino e Amazon; situações de mau funcionamento ou indisponibilidade de seus sistemas podem gerar grandes prejuízos, além de danos em sua imagem, já que elas baseiam sua estratégia de atuação em uma forte presença na Internet, com pouca ou nenhuma estrutura adicional para atender seus clientes.

Para elas, e a maior parte das empresas de médio e grande porte, é fundamental que seus sistemas de informações atendam a todos os requisitos de negócio, tanto do ponto de vista de funções, como da capacidade de atendimento aos clientes com qualidade e rapidez. Eles devem ainda ser flexíveis, para que eventuais mudanças na estratégia de atuação sejam rapidamente transferidas para os sistemas.

A migração das empresas para a Internet representou uma grande evolução nos modelos de relacionamento com os clientes, e permitiu que elas ampliassem seus mercados de atuação, pela facilidade de alcance a clientes em todas as partes do mundo.

Entretanto, esta evolução também gerou novas preocupações. Os clientes podem estar em locais diversos e desejam ter comodidade de acesso a qualquer horário, o que exige que os sistemas estejam disponíveis 24 horas. A grande facilidade de acesso exige que eles atendam a grandes volumes de usuários simultaneamente e tenham mecanismos sofisticados de segurança, pois podem ser alvos de ataques maliciosos.

Neste cenário, apesar da redução dos custos unitários com *hardware*, *software* e serviços de comunicações, gastos com atividades de desenvolvimento e produção de sistemas de informações aumentaram muito. Assim, nos últimos

15 anos as organizações têm dedicado atenção crescente a seus sistemas de informações.

Estudo recente do Instituto Brasileiro de Geografia e Estatística (IBGE 2009, p. 42) sobre o setor de tecnologia da informação e comunicações no Brasil mostra que o mercado de informática no Brasil em 2006 gerou receitas de R\$ 37 bilhões, e que as despesas para a contratação de serviços de desenvolvimento de *software* por encomenda ficaram próximas de R\$ 11 bilhões, além de R\$ 6 bilhões para contratação de licenças de *software* de produtos prontos.

Entretanto, apesar destes investimentos, e da evolução das ferramentas de tecnologia, o resultado entregue pelas áreas de desenvolvimento de sistemas ainda frustra a maior parte dos clientes e empresas. O relatório *Standish Report* (1995, p. 2) ficou famoso na área de sistemas, ao mostrar que apenas 16% dos projetos de *software* foram considerados como sucesso, isto é, foram concluídos no prazo, dentro do orçamento e atenderam aos requisitos dos usuários. Ele afirmava ainda que 31% dos projetos foram simplesmente cancelados.

Os dados do mesmo relatório de 2006 retratam melhora significativa nestes indicadores, com cerca de 35% dos projetos de *software* completados como sucesso. Mas ainda indica que 19% dos projetos foram cancelados antes de sua conclusão. Embora melhores estes números ainda são incompatíveis com uma indústria tão importante.

Relatório da agência do governo dos Estados Unidos para padrões e tecnologia – *National Institute for Standards and Technology* (NIST, 2002, p. ES11) estima que o custo econômico de erros em *software* no país em 2000 foi da ordem de US \$ 59 bilhões, e propõe ações para a melhoria da infraestrutura de testes, o que poderia reduzir este valor em cerca de US\$ 22 bilhões. Uma das justificativas para estes dados é o fato da pouca experiência das pessoas e empresas, já que o desenvolvimento de *software* é uma atividade nova, com pouco mais de cinquenta anos.

Kruchten (2000, p. 4) descreve que é possível destacar as seguintes causas para falhas em projetos de *software*:

- Entendimento inadequado das necessidades dos usuários;
- Falta de habilidade, ou organização, para tratar as mudanças de requisitos;
- Inconsistências não detectadas entre requisitos, projeto e implementação;
- Módulos de *software* que não se encaixam;
- Produtos de *software* com baixa qualidade, ou com problemas de desempenho.

Dingsoyr e Conradi (2002, p. 392) destacam que *software* é um produto imaterial, sendo difícil ter uma visão do produto como um todo – um sistema pode ter milhões de linhas de código, onde é muito difícil identificar um problema.

Rus *et al.* (2001, p. 5) lembram que produtos de *software*, em geral, são diferentes entre si, e exemplificam que desenvolver sistemas para o ônibus espacial é muito diferente de desenvolver para uma máquina de lavar roupas. Os autores mostram ainda que o desenvolvimento de *software* exige o conhecimento de pelo menos dois domínios: o de tecnologia e processos de desenvolvimento; e

o domínio de conhecimento para a área que o produto de *software* irá atender – financeira, pessoal, impostos ou comércio exterior, apenas para citar alguns.

Em função desta diversidade, Basili, Caldiera e Rombach (1994, p. 471) defendem que a disciplina de desenvolvimento de *software* é essencialmente experimental, evolucionária e não-repetitiva, o que dificulta a comparação de experiências e a realização de pesquisas científicas.

Algumas das variáveis que afetam o processo de desenvolvimento de *software* são descritas pelos mesmos autores, como organização das equipes, experiência das pessoas, capacitação das pessoas, tecnologias utilizadas, e conhecimento do domínio de negócio a ser atendido, dentre outros. Estas variáveis dificultam a repetição de contexto e de experiências para comparação.

Em função disto, nos últimos vinte anos buscou-se incentivar o uso de técnicas e sistemáticas, repetíveis, para que a atividade de desenvolvimento de *software* tenha características similares as disciplinas de engenharia, onde o uso de técnicas consagradas permite a elaboração de projetos, definição de metas e obtenção de resultados confiáveis, em termos de custo, prazo e qualidade do produto final.

As definições a seguir mostram a preocupação de agregar à atividade as práticas e modelos desenvolvidos na área de engenharia, em mais de dois mil anos de história.

Para Sommerville (2001, p. 78) a engenharia de *software* é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação. Esta é uma definição complexa, pois considera que a engenharia de *software* não trata apenas do processo de desenvolvimento, mas também dos mecanismos de apoio a produção do *software*.

Estudo do IEEE (SWEBOK, 2001, p. 1.1), define engenharia de *software* como a aplicação de abordagem sistemática, quantificável e disciplinada ao desenvolvimento, operação e manutenção de *software*, i. e., a aplicação da engenharia ao *software*.

Já Pressman (2002, p.17), define engenharia de *software* como “...a criação e a utilização de sólidos princípios de engenharia a fim de obter *software* de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais”.

Entretanto, Kruchten (2004, p.1) destaca que, apesar da evolução dos processos e técnicas de desenvolvimento, quatro características importantes ainda tornam a engenharia de *software* diferente de outras disciplinas tradicionais de engenharia.

A primeira delas é a ausência de uma teoria fundamental, assim como existem as leis da física. Apesar da existência de técnicas de construção, produtos de *software* são essencialmente não estruturados, e uma falha em um ponto pode gerar falhas em outras áreas do produto. Também é difícil definir padrões de engenharia, e a alternativa é o uso de melhores práticas.

Outra diferença importante é a facilidade, pelo menos aparente, de realizar mudanças. *Software* é fácil de ser alterado, e as organizações gostam de

aproveitar esta característica para realizar alterações durante seu desenvolvimento, e mesmo após sua implantação. É muito difícil estimar os custos e riscos das mudanças.

A terceira característica destacada por Kruchten é a rápida evolução da tecnologia. Técnicas de desenvolvimento, ferramentas de desenvolvimento e ambientes operacionais têm rápida evolução, o que dificulta a consolidação do conhecimento das equipes, e cria pressão para a contratação de novos produtos e ampliação de atividades de treinamento para manter as equipes atualizadas.

Finalmente, ele afirma que o custo de construção do *software* é baixo, o que facilita a percepção de que os erros podem ser corrigidos em seguida com facilidade.

Para contornar estes problemas, e permitir maior controle dos resultados a serem entregues, foi adotada a estratégia da definição de processos para o desenvolvimento de *software*.

3.4.2. Processos de *Software*

As tentativas de ordenar a atividade resultaram na definição de processos e modelos de maturidade para o desenvolvimento de *software*, e a criação da disciplina de engenharia de *software*. As organizações têm utilizado estes processos e modelos na expectativa de resolver parte de seus problemas de desenvolvimento de sistemas de informação.

Segundo Borges (2001, p. 2), um processo de desenvolvimento pode ser visto como um conjunto de atividades, métodos, ferramentas e práticas que são utilizadas para construir um produto de *software*. Já Humphrey *apud* Borges (*op. cit.*), define processo de desenvolvimento como o conjunto de tarefas de engenharia de *software* necessárias para transformar os requisitos dos usuários em um sistema.

Kruchten (2000, p. 15) descreve que um processo de desenvolvimento de *software* tem quatro papéis:

- Oferecer um guia para ordenar as atividades das equipes de desenvolvimento;
- Especificar os artefatos que devem ser construídos, e quando eles devem ser construídos;
- Direcionar as tarefas de cada componente e da equipe como um todo;
- Oferecer critérios e métricas para avaliação dos produtos e atividades do projeto de *software*.

O autor menciona ainda que, sem um processo bem definido, a equipe de desenvolvimento irá trabalhar de forma desordenada, e o resultado dependerá de esforços individuais, o que não é uma situação sustentável no médio e longo prazos.

Um dos processos mais conhecidos é o RUP (*Rational Unified Process*), desenvolvido pela empresa *Rational Software*, hoje uma divisão da IBM. De acordo com Kruchten (2000, p. 17), ele oferece uma abordagem disciplinada na

atribuição de responsabilidades e tarefas em uma organização de desenvolvimento de *software*. Seu objetivo é assegurar a produção de *software* de qualidade, que atenda às necessidades dos usuários, com prazo e orçamento previsíveis.

O processo foi construído com base em práticas consideradas de sucesso pelo mercado de desenvolvimento de *software*, e organizadas de forma a atender a diversidade de projetos de uma organização. Ele é composto por seis práticas:

- Desenvolvimento iterativo de *software*;
- Gestão de requisitos de *software*;
- Uso de arquiteturas baseadas em componentes;
- Uso de modelos visuais para representação do sistema, normalmente com uso de uma linguagem específica para este fim;
- Verificação contínua da qualidade do *software*;
- Controle das mudanças no produto de *software*.

O processo RUP possui quatro elementos principais:

- Trabalhadores – quem executa atividades;
- Atividades – como o processo é desenvolvido;
- Artefatos – o que deve ser feito para concluir uma atividade;
- Fluxos de trabalho – quando, e em que ordem, cada atividade deve ser feita.

O processo é utilizado hoje por mais de 1.200 empresas de desenvolvimento de sistemas em todo o mundo (KRUCHTEN, 2000).

3.4.3. Modelos de Maturidade para Desenvolvimento de *Software*

A necessidade de desenvolver *software* de forma gerenciada e confiável propiciou a criação de modelos de maturidade para desenvolvimento de *software*. O mais conhecido e utilizado deles foi desenvolvido pelo *Software Engineering Institute* (SEI), da Universidade Carnegie Mellon – CMMI (*Capability Maturity Model Integration*).

O CMMI, segundo o SEI (2000), é uma abordagem de melhoria de processos, um modelo de referência que contém práticas que levam à maturidade em uma atividade. A primeira versão do modelo foi originalmente criada para a disciplina de desenvolvimento de *software*, mas hoje o CMMI está disponível para outras disciplinas.

A versão atual do CMMI (versão 1.2) apresenta dois modelos: *CMMI for Development* (CMMI-DEV), dirige-se ao processo de desenvolvimento de produtos e serviços; e o *CMMI for Acquisition* (CMMI-ACQ), voltado para processos de aquisição e terceirização de bens e serviços.

O CMMI para Desenvolvimento (CMMI) é um modelo de referência e um conjunto de melhores práticas de mercado, que cobre atividades de desenvolvimento e manutenção de produtos e serviços. Organizações de diversos

setores, como financeiro, hardware, fabricação de automóveis, desenvolvimento de *software* e telecomunicações o utilizam como referência para seus processos de desenvolvimento.

O CMMI descreve práticas para gestão de projetos, gestão de processos, engenharia de *software*, uso de equipes integradas e outros processos de suporte usados em desenvolvimento e manutenção. Ele aborda práticas que cobrem todo o ciclo de vida de um produto, sua concepção, desenvolvimento, entrega e manutenção de *software*.

O modelo utiliza o conceito de níveis para descrever um caminho evolutivo recomendado para as empresas interessadas em melhorar seus processos de desenvolvimento de produtos. Um nível de maturidade é um degrau de evolução para a melhoria dos processos.

Cada nível de maturidade estabelece um patamar importante para os processos organizacionais, e prepara o caminho para o próximo nível de maturidade. Estes níveis são medidos pelo alcance de objetivos associados a um conjunto predefinido de áreas do processo.

O CMMI é estruturado em cinco níveis de maturidade, onde o nível 1 é o menos maduro e o nível 5 é o mais maduro. Cada nível de maturidade, com exceção do nível 1, é composto de várias áreas-chave de processo (*key process areas* - KPAs). Falbo (2005, p.19) detalha os cinco níveis de maturidade.

- Nível 1 – Inicial: O processo de *software* é caracterizado como *ad hoc* e, eventualmente, caótico. Poucos processos são definidos e o sucesso depende de esforços individuais. Organizações deste nível operam, tipicamente, sem formalizar procedimentos, estimativas de custo e planos de projeto;
- Nível 2 – Repetível: Os processos básicos de gestão são estabelecidos para acompanhar custo, cronograma e funcionalidade. Os sucessos em projetos anteriores com aplicações similares podem ser repetidos;
- Nível 3 – Definido: A organização possui um processo padrão definido que é usado como base para todos os projetos. As atividades de engenharia e gerência de *software* são estáveis e há um entendimento comum e amplo das atividades, papéis e responsabilidades no processo;
- Nível 4 – Gerenciado: A organização fixa metas quantitativas de qualidade para produtos e processos e fornece instrumentos para medições consistentes e bem definidas. Tanto o processo de *software* como os produtos são quantitativamente entendidos e controlados;
- Nível 5 – Otimizado: A organização possui uma base para melhoria contínua e otimização do processo. Dados sobre a eficiência de um processo são usados para efetuar análises custo-benefício de novas tecnologias e para propor mudanças no processo.

Um outro modelo de referência para processos é o MPS.BR (Melhoria de Processo de *Software* Brasileiro), modelo de referência desenvolvido no Brasil que define modelo de melhoria e avaliação de processo de *software*. É voltado, preferencialmente, às micro, pequenas e médias empresas brasileiras.

Ele pretende ser reconhecido internacionalmente como um modelo aplicável à indústria de *software*, e foi desenvolvido aderente a modelos e normas

internacionais. A base técnica utilizada para a construção do MPS.BR é composta pelas normas NBR ISO/IEC 12207, e suas emendas 1 e 2, e a ISO/IEC 15504. Além disso, ele também cobre o conteúdo do CMMI.

O MPS.BR define 7 níveis de maturidade, e está dividido em três componentes:

- Modelo de Referência (MR-MPS): contém os requisitos que as organizações deverão atender para estar em conformidade com o MPS.BR. Define, também, os níveis de maturidade e da capacidade de processos e os processos em si.
- Método de Avaliação (MA-MPS): contém o processo de avaliação, os requisitos para os avaliadores e os requisitos para averiguação da conformidade ao modelo MR-MPS. Está descrito de forma detalhada no Guia de Avaliação e foi baseado na norma ISO/IEC 15504.
- Modelo de Negócio (MN-MPS): contém uma descrição das regras para a implementação do MR-MPS pelas empresas de consultoria, de *software* e de avaliação.

3.4.4. Requisitos e Engenharia de Requisitos

Requisitos estão associados a questão central em um projeto de software: o que exatamente o sistema deve fazer?

Leffingwell e Widrig (2000, p.15) apresentam duas definições⁸ para o termo:

- Capacidade de um *software*, necessitada pelo usuário para resolver um problema ou alcançar um objetivo;
- Capacidade de um *software* que deve estar presente em um sistema, ou componente do sistema, para satisfazer um contrato, padrão, especificação ou outra documentação formal.

Pressman (2002, p.580) associa requisitos a qualidade, ao afirmar que qualidade de *software* é:

[...] conformidade a requisitos funcionais e de desempenho explicitamente declarados, normas de desenvolvimento explicitamente documentadas e a características implícitas que são esperadas de todo *software* profissionalmente desenvolvido.

Ainda segundo o autor, a associação de requisitos a qualidade enfatiza três pontos importantes:

- Os requisitos de *software* são a base pela qual a qualidade é medida. A falta de conformidade com os requisitos é falta de qualidade;
- As normas definem um conjunto de critérios de desenvolvimento que guia o modelo pelo qual o *software* é submetido à engenharia. Se os critérios não são seguidos, quase certamente ocorrerá falta de qualidade;

⁸ As definições são adaptadas de Dorfmann e Thayer – Standard, Guidelines and Examples of System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990.

- Um conjunto de requisitos implícitos frequentemente não é mencionado, como por exemplo o desejo de facilidade de uso. Se o *software* satisfaz seus requisitos explícitos, mas deixa de satisfazer os requisitos implícitos, a qualidade do *software* é suspeita.

Já segundo Blaschek (2002, p.1) os requisitos de um sistema de informação constituem uma especificação de suas características e propriedades, o que ele deve fazer, de como ele deve se comportar, bem como suas restrições de operação.

Esta definição explicita a importância do contexto, isto é, restrições ou limitações que o projeto deve atender, além das especificações de funcionalidades.

Leffingwell e Widrig (2000, p.19) propõem uma divisão e representação gráfica da definição de requisitos, conforme a Figura 5. Eles consideram dois domínios – o domínio do problema, onde estão os usuários e demais envolvidos para quem os problemas técnicos e de negócios a serem resolvidos.

O outro é o domínio da solução, onde o sistema será desenvolvido. Eles consideram importante a tradução das necessidades dos usuários em características do sistema, descritas em linguagem do usuário – como, por exemplo “[...] o formulário de cadastramento dos usuários será acessado pela Internet.”

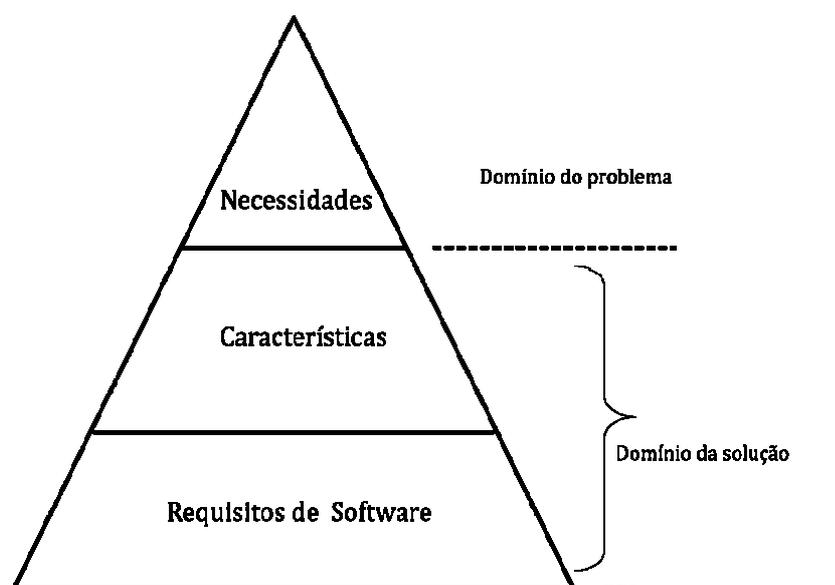


Figura 5 - Representação de domínios para requisitos

Fonte: Leffingwell e Widrig, 2000, p. 21

Estas características, definidas como um serviço que o sistema oferece para atender as necessidades dos usuários, são a camada intermediária no domínio da solução.

Por fim, uma vez que as características principais estão acordadas com os usuários, a equipe do projeto pode iniciar a definição dos requisitos de *software*, tratando de forma detalhada todas as necessidades. Esta definição é feita usando linguagens específicas, gerando a especificação de requisitos.

Segundo Davis (1999, p.43) a especificação de requisitos é um documento que registra comportamentos externamente observáveis do sistema a ser construído. Estes comportamentos incluem as seguintes especificações:

- Entradas do sistema – descreve o conteúdo das entradas, mas também os dispositivos e sua forma. Esta área envolve muitos detalhes e é sujeita a muitas alterações, em função do uso de interfaces gráficas, novas tecnologias de entrada de dados e uso da Internet;
- Saídas do sistema – descrição dos dispositivos de saída – voz, vídeo, ou texto – que devem ser suportados, bem como formatos da informação gerada pelo sistema;
- Funções do sistema – mapeamento entre entradas e saídas, e suas várias combinações;
- Atributos do sistema – requisitos típicos não relacionados a funções, como confiabilidade, disponibilidade e capacidade de atendimento, que os desenvolvedores devem considerar no projeto;
- Atributos do ambiente operacional – requisitos adicionais relacionados ao ambiente de operação do sistema, como restrições de operação, compatibilidade com sistemas operacionais e outros pacotes de *software*.

O mesmo autor afirma que os requisitos devem também exibir as seguintes características: completos, compreensíveis, consistentes, concisos, rastreáveis, modificáveis, não redundantes e organizados.

A evolução da importância dos requisitos para o desenvolvimento de sistemas levou a associação do conceito de engenharia também para esta atividade - a engenharia de requisitos. Espindola *et al.* (2004) destacam que o termo “engenharia” implica o uso de técnicas sistemáticas e repetíveis, para garantir que os requisitos do sistema sejam completos, consistentes e relevantes.

Já Nuseibeh e Easterbrook (2000, p. 1) a conceituam como o ramo da engenharia de *software* que se preocupa com os objetivos do mundo real, funções e restrições de um sistema. Ela se ocupa também das relações destes fatores com a especificação precisa do comportamento do sistema, e sua evolução ao longo do tempo.

Ainda em relação a engenharia de requisitos, Leffingwell e Widrig (2000, p.16) a definem como:

Abordagem sistemática para elicitare, organizar e documentar os requisitos de um sistema e o processo para estabelecer e manter acordo entre o cliente e a equipe de projeto nas alterações dos requisitos do sistema.

Os requisitos podem ser classificados de diversas formas, e a mais relevante para esta pesquisa é separação em requisitos funcionais e requisitos não-funcionais (RNF).

Requisitos funcionais são as declarações das funções que o sistema deve oferecer, como o sistema se comporta em cada situação em sua operação. Em

alguns casos os requisitos funcionais podem também explicitamente definir o que o sistema não deve fazer.

Exemplos de requisitos funcionais são: possibilitar o cálculo dos gastos diários, mensais e anuais com pessoal; emitir relatórios de compras a cada mês; os usuários podem obter o número de aprovações, reprovações e trancamentos em todas as disciplinas por um determinado período de tempo.

Os requisitos não-funcionais são normalmente registrados como restrições, ou condições específicas, para a operação do sistema – de tempo, custo, desempenho ou padrões a serem atendidos. Eles são parte importante na descrição do contexto de desenvolvimento de um sistema. Exemplos de requisitos não-funcionais são: o tempo de resposta do sistema deve ser inferior a 10 segundos; ele deve atender a até 1.000 usuários simultâneos; o sistema deve trabalhar em ambiente Linux.

A complexidade das tecnologias envolvidas, e a importância dos sistemas de informação para a operação das organizações, têm ampliado a quantidade e a importância de aspectos não-funcionais exigidos pelos clientes, como custo, confiabilidade, segurança, manutenibilidade, portabilidade, desempenho, e rastreabilidade de informações, dentre outros.

Este fato é destacado por Chung, Nixon e Yu (1995, p. 31), ao afirmarem que não há uma definição formal de requisitos não-funcionais (RNF), e nem uma lista completa de quais eles devem ser para cada projeto. Mas os autores completam que os RNF são cruciais, e devem ser tratados o mais cedo possível nos projetos de *software*.

Os requisitos sofrem influência dos domínios do sistema a ser desenvolvido. Alguns destes requisitos são relacionados ao domínio de negócio, como por exemplo, restrições de segurança e desempenho. Outros podem ser associados ao domínio da tecnologia, como compatibilidade com ambientes operacionais e suporte a dispositivos específicos.

Assim, a importância dos RNF é associada a decisões que devem ser tomadas para o andamento do projeto, e para as quais o seu entendimento é fundamental. Em empresas de prestação de serviços de TI, caso do Serpro, é comum que requisitos não-funcionais sejam utilizados como métricas ou indicadores de níveis de serviços em contratos, razão pela qual devem ser tratados com relevância desde o início dos projetos.

Finalmente, cabe destacar que a engenharia de requisitos, assim como a ciência da informação, tem natureza multidisciplinar. Nuseibeh e Easterbrook (2000) relatam que ela é uma atividade centrada em pessoas, e afetada pela forma que as pessoas percebem e entendem o mundo a seu redor, como elas interagem e como seu ambiente de trabalho as afeta. A base teórica e as práticas de elicitação e modelagem de requisitos estão associadas às ciências sociais e cognitivas:

- Psicologia cognitiva – facilita o entendimento das dificuldades que as pessoas têm em descrever suas necessidades;
- Antropologia – oferece métodos de observação de atividades humanas, o que auxilia em melhor entendimento de como sistemas em computadores podem ajudar nestas atividades;

- Sociologia – entendimento de mudanças políticas e culturais causadas pela automação de tarefas em sistemas;
- Linguística – a atividade de requisitos é basicamente de comunicação, e análises desta disciplina têm alterado a forma como o idioma é usado para especificação de requisitos.

Para este trabalho, será considerada a seguinte definição - os requisitos de um sistema definem os serviços que ele deve oferecer e as restrições e condições específicas para sua operação. Embora simplificada ela contém os elementos importantes para o processo e resultado final.

3.4.5. Conclusão

O desenvolvimento de *software* tornou-se uma atividade crítica para boa parte das organizações, já que parte expressiva dos negócios depende de sistemas de informação para sua execução. O mercado de desenvolvimento de sistemas no Brasil, segundo o IBGE (2006), gerou receitas de R\$ 11 bilhões mas, apesar dos investimentos em ferramentas, os resultados são ruins.

Dentre as principais causas para os atrasos, cancelamento e custos acima dos previstos em projetos de sistemas, estão o entendimento inadequado das necessidades dos usuários, falta de habilidade das equipes para tratar as mudanças de requisitos e a baixa qualidade dos produtos de *software*.

Além disto, as mudanças constantes de tecnologia e contexto dos projetos dificulta o estabelecimento de disciplina que permita o aproveitamento de experiências de sucesso, e a não repetição de erros já corrigidos. Basili e Rombach afirmam que a disciplina de desenvolvimento de sistemas é essencialmente experimental e não-repetitiva, dificultando a comparação de experiências.

Para tratar este problema foi criado o conceito de engenharia de *software*, que incentiva o uso de técnicas e sistemáticas repetíveis, similares as de outras disciplinas de engenharia, visando a garantia de obtenção de resultados mais previsíveis em termos de qualidade, custos e prazos.

Esta iniciativa foi complementada com a utilização de processos padronizados e de modelos de maturidade para desenvolvimento de *software*. A utilização de processos padronizados busca oferecer um guia para as equipes de desenvolvimento, especificar os tipos de artefatos que devem ser construídos, direcionar as tarefas dos componentes das equipes e oferecer métricas para a avaliação dos produtos e atividades ao longo do projeto.

Dentre os modelos de maturidade utilizados no desenvolvimento de sistemas o mais conhecido é o CMMI, que define cinco níveis de maturidade organizacional, um caminho evolutivo composto de melhores práticas nas áreas de gestão de projetos, gestão de processos, desenvolvimento de sistemas e gestão de equipes.

Entretanto, apesar destes modelos de processos e facilidades para gestão de práticas de projetos, Kruchten (2004) ressalta que a engenharia de *software* possui características que a diferenciam de outras disciplinas científicas, como produtos não estruturados, a rápida evolução da tecnologia e custo baixo para construção e alteração de produtos. Elas dificultam a consolidação de conhecimentos e práticas, e a previsibilidade de resultados.

Neste cenário, uma das áreas mais importantes para o modelo é a engenharia de requisitos, que engloba levantamento de informações e necessidades que devem ser atendidas por um sistema. Os requisitos são normalmente classificados como funcionais e não-funcionais. Os requisitos funcionais são relacionados as funcionalidades do sistema para atender necessidades dos processos de trabalho dos usuários

Já os requisitos não-funcionais referem-se as restrições e condições específicas impostas ao projeto, e tem crescido de importância nos últimos anos, pela complexidade das tecnologias envolvidas, e de características de disponibilidade e desempenho exigidas pelas organizações.

O levantamento de requisitos é uma atividade complexa, de natureza multidisciplinar e com forte dependência das pessoas. Estudos mostram que ela concentra uma parcela significativa dos erros na implementação de sistemas de informação, e que a correção de erros nesta fase pode ser até 1.000 vezes mais barato do que em outras fases do projeto, conforme levantamento feito por Pressman (2006, p.580), e mostrado na Figura 6, a seguir.

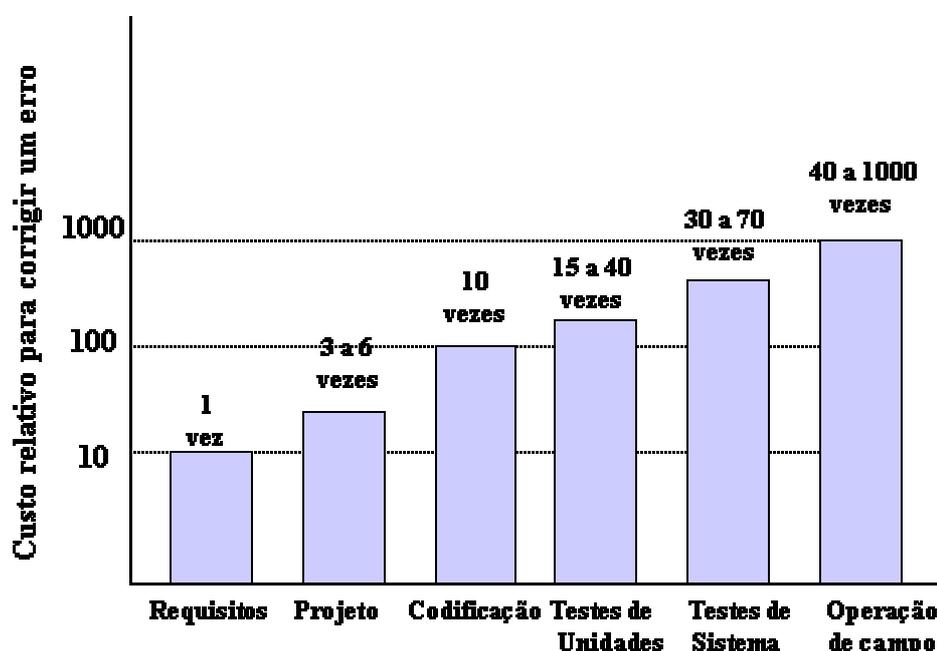


Figura 6 - Custo relativo para corrigir um erro

Fonte: Pressman, 2006, p.580

3.5. Gestão da Informação e Engenharia de Software

Conforme citado em 3.2, os termos informação e conhecimento são usados com frequência de forma intercambiável, e o mesmo tem ocorrido com as expressões “gestão da informação” e “gestão do conhecimento”.

Davenport e Marchand (2001, p. 1) mostram que muitos projetos de gestão do conhecimento possuem elementos significativos de gestão da informação, já que as pessoas precisam de informações de onde o conhecimento reside e, para compartilhar o conhecimento é necessário transformá-lo em algum tipo de informação.

Segundo os autores, pesquisa de 1996 examinou 31 projetos diferentes, nominados por seus gerentes de gestão do conhecimento, e concluiu que o que eles efetivamente gerenciavam parecia ser uma mistura de informação, conhecimento e um pouco de dados.

Cerca de 80% dos projetos envolviam a criação de algum tipo de repositório para armazenar itens considerados relevantes pelos empregados, como melhores práticas, documentação de produtos, apresentações de vendas e mesmo informações sobre cardápios e rotas de ônibus. Os repositórios incluíam o que pode se chamado de “informação sobre conhecimento”, isto é, informação que leva a um tipo de conhecimento, seja na forma de um documento ou de um especialista.

Uma das justificativas encontrada pelos autores para esta forte ligação entre informação e gestão do conhecimento é que as pessoas estão constantemente convertendo conhecimento em várias formas de informação – como por exemplo memorandos, relatórios e apresentações – e adquirindo informações de outros para aprimorar seu conhecimento.

Alvarenga Neto *et al.* (2007, p. 7) explicam que o conhecimento só existe na mente humana e no espaço entre mentes e, fora deste contexto, é visto como informação. Os autores sugerem que não se gerencia conhecimento, mas se estimula o conhecimento pela criação de contextos organizacionais favoráveis. Para eles a gestão do conhecimento assume o significado de gestão, de e para o conhecimento.

Ainda tratando da distinção entre gestão da informação e do conhecimento, Wilson (1999, p. 4) descreve a dificuldade de implementar o conceito de gestão do conhecimento - o conhecimento é um construto individual e social, e há uma natural dificuldade para uma empresa gerenciar o que está na mente das pessoas. Ele defende a criação de uma estratégia de sensibilização para o conhecimento, em que todos os gestores seriam tratados como gestores de informações, e não apenas os profissionais tradicionalmente encarregados do assunto.

O referido autor afirma ainda que o conhecimento não pode ser diretamente gerenciado, e que podemos apenas gerenciar informação sobre o conhecimento possuído pelas pessoas nas organizações. Mesmo assim, a informação a ser gerenciada será necessariamente incompleta, em função das fronteiras do conhecimento pessoal serem difusas, e estarem em permanente mudança a medida que o indivíduo amplia seu conhecimento.

Finalmente, Michaud (2006, p. 233) parte do pressuposto que o conhecimento só reside na cabeça das pessoas e é, por natureza, implícito; e

conclui que sua comunicação é feita sempre por meio de informações. Segundo ele,...

[...] O entendimento e o posicionamento dos processos apresentados pelos professores Nonaka e Takeuchi sofrem um pequeno ajuste de terminologia, atuando mais no domínio da informação e desdobrando o de combinação em combinação da informação, atuando de forma explícita, e combinação de conhecimento, atuando no domínio puramente tácito.

O autor complementa citando que os modelos e tecnologia disponíveis hoje para a sociedade possibilitam muito mais uma gestão da informação do que a gestão do conhecimento, apesar de alguns passos nesta direção, com pesquisas de inteligência artificial e redes neurais.

Assim, apesar da importância do conhecimento, seu compartilhamento e criação de novos conhecimentos, para a sobrevivência das organizações, este trabalho terá foco na utilização do conceito e modelos de gestão da informação.

3.5.1. Gestão da Informação e Engenharia de *Software* – Cenário

A disciplina de engenharia de *software* foi criada com o objetivo de utilizar métodos, técnicas e ferramentas na atividade de desenvolvimento de *software*, assim como é feito nas áreas tradicionais da engenharia, e garantir resultados mais previsíveis para os projetos de sistemas de informação. Assim, a partir da segunda metade da década de 80, vários projetos foram iniciados, em empresas e centros de pesquisa, para identificar as melhores formas de alcançar este resultado.

Glass (1999, p.75) cita algumas das iniciativas do período:

- Técnicas estruturadas – uso de técnicas estruturadas para análise, projeto e programação;
- Linguagens de programação de quarta geração (4GL);
- CASE (*Computer Aided Software Engineering*) – ferramentas para suporte a engenharia de *software*, principalmente nas fases de análise e projeto;
- Métodos formais – fases de especificação e verificação de *software*;
- Método Sala-limpa (Cleanroom) – método para remoção de defeitos de *software*;
- Modelos de processo – descrição de processos apropriados para engenharia de *software*;
- Tecnologia orientada a objeto – geração de *software* com base em problemas a serem resolvidos, e seus objetos.

Ele afirma que, embora muitas destas tecnologias e métodos tenham mostrado alguns resultados, há dificuldade em avaliar sua eficiência, em função

de mudanças de linguagens e bancos de dados, e a falta de estudos e pesquisas sobre como eles realmente influenciam o desenvolvimento de *software*.

Este posicionamento é confirmado, pela dificuldade em melhorar os resultados dos processos de desenvolvimento de *software*. Lyytinen e Robey (1999, p. 86) argumentam que as organizações de desenvolvimento de sistemas de informação falharam no aprendizado de meios efetivos de solucionar problemas como escopo e riscos de projetos.

Eles comentam que, ironicamente, a maior parte das informações necessárias a este aprendizado está disponível na experiência prévia de outros projetos de *software* e que, para evitar estas falhas, as equipes de desenvolvimento devem aprender com experiências – suas e de outras equipes – e usar estas informações para mudar suas práticas de trabalho.

Dentre as iniciativas associadas a modelos de processos, uma das mais relevantes é a que trata o compartilhamento de informações e conhecimento entre as equipes de desenvolvimento, como forma de evitar a repetição de erros já ocorridos, e aproveitar soluções ou práticas de comprovado sucesso.

Esta abordagem reforça a visão de interdisciplinaridade defendida por Saracevic, e confirma a integração entre desenvolvimento de *software* e a ciência da informação. A integração deixa de lado apenas os aspectos de uso de ferramentas e tecnologias, com novos conceitos e modelos sendo incorporados aos processos de desenvolvimento.

A partir dos anos 80, são realizados diversos trabalhos com componentes da ciência da informação integrados ao processo de desenvolvimento. Parreiras (2003, p. 7) afirma que a engenharia de *software* é uma atividade orientada ao conhecimento, que está disperso entre as diversas pessoas da equipe.

Ao longo do projeto, cada pessoa da equipe toma várias decisões relacionadas à sua atividade, com várias opções possíveis, diferente de processos de produção ou manufatura, onde boa parte das tarefas pode ser repetida entre projetos distintos.

Rus *et. al* (2001, p. 1) relatam que desenvolvimento de sistemas é uma atividade intensiva em pessoas e conhecimento, e que, como nas empresas de consultoria, investimento e propaganda, o maior ativo de uma organização de desenvolvimento de sistemas é seu capital intelectual. Segundo eles, o conhecimento em engenharia de *software* é diversificado e cresce constantemente.

Registram ainda que, em todo projeto de desenvolvimento de *software*, pelo menos duas áreas de conhecimento devem ser consideradas. A primeira delas é chamada domínio de negócio, e trata do assunto para o qual o sistema será desenvolvido – financeiro, administrativo e gestão de pessoas, apenas para citar alguns exemplos.

A outra área a ser considerada é o domínio da tecnologia, e refere-se aos processos, modelos e ferramentas de desenvolvimento de *software*. O conhecimento relativo a estas áreas será codificado nos artefatos do projeto, como relatórios, documentos e código de programas de computador.

Os autores destacam que cada produto de *software* é diferente dos demais, seja porque as necessidades dos usuários evoluíram e há um novo contexto, seja

porque novas tecnologias serão agregadas a um novo projeto. Em função disto eles defendem que, apesar da importância do aproveitamento de informações e artefatos, uma abordagem única não pode ser usada para todos os projetos.

Dingsoyr (2000, p. 56) relata que, recentemente, há um foco especial no reuso de experiências de projetos anteriores para aumento da qualidade de novos projetos. Isto pode ser feito pelo aproveitamento direto de artefatos, e também pela criação de um ambiente de aprendizagem entre os desenvolvedores de uma equipe ou projeto.

A maior parte dos projetos descritos na literatura utiliza uma das abordagens descrita por Hansen, Nohria e Tierney (1999, p. 107) para a utilização dos conhecimentos na área de engenharia de *software*:

- Codificação – consiste na codificação, registro e armazenamento do conhecimento em bases de dados, de onde pode ser acessado por outras pessoas na organização. Normalmente a codificação é feita em abordagem “pessoa-documento”, isto é, a pessoa que domina o conhecimento é responsável por sua codificação;
- Personalização – conhecimento é sempre associado as pessoas, sendo compartilhado em contatos pessoais diretos. Nestes casos, em geral, há disponibilidade de sistemas para identificação dos especialistas em cada área ou assunto.

Na abordagem de codificação, o conhecimento é transformado em informação, para posterior armazenamento, distribuição e uso.

3.5.2. Gestão da Informação em Organizações de *Software*

Apesar de constatar a importância do uso de técnicas de gestão da informação e compartilhamento de conhecimento, as organizações de desenvolvimento de sistemas de informação ainda têm dificuldade para a aplicação destes conceitos, e seu efetivo acoplamento a seus processos de desenvolvimento.

Desouza (2003, p.100) analisa barreiras ao uso efetivo de gestão da informação e conhecimento em processos de engenharia de *software*, e destaca que, em algumas situações, não é possível capturar e codificar o conhecimento. Em função da especificidade dos projetos ele é contextual em sua natureza. Outra barreira é a dificuldade de ter meios de troca de informações e conhecimento que sejam capazes de oferecer boas experiências aos usuários.

Rus *et. al.* (2001, p. 8) citam a pressão dos prazos para entrega de sistemas como um dos principais obstáculos para a dificuldade na codificação do conhecimento e no uso das informações já registradas sobre projetos anteriores. As equipes simplesmente não têm tempo para registrar as experiências de seus projetos. Eles destacam a ausência da cultura de reuso de experiências em organizações de desenvolvimento de sistemas, sendo a evolução tecnológica uma das explicações para o fato – as equipes e pessoas buscam o uso de novas tecnologias, o que dificulta o reuso de soluções e experiências.

Os referidos autores afirmam que a característica de “invisibilidade” do *software* também deve ser considerada. Seria mais fácil reusar algo que é visível, como partes em projetos de engenharia. No cenário atual, é comum o desenvolvimento de módulos ou sistemas que já existem, mas que a equipe não tem ciência. Eles complementam citando a natureza contextual de projetos de *software*, sua integração com diversas ferramentas de tecnologia, o domínio de negócios atendido e requisitos específicos para o sistema, que tornam difícil o aproveitamento e reuso de experiências em outros projetos, com contextos diversos.

Estudos realizados na área de reuso em *software* mostram aspectos diversos que podem afetar projetos de reuso nas organizações. O conceito de reuso de *software* é definido por Kim e Stohr (1998, p. 119) como:

[...]o uso de recursos de *software* previamente desenvolvidos, em todas as fases do ciclo de desenvolvimento, para uma nova aplicação. Um recurso reusável pode ser um módulo de programa, especificações de requisitos, estruturas de dados, casos de testes e decisões de projetos, dentre outros.

Estes autores definem ainda três tipos principais de reuso de *software*, baseado no estágio do produto: (1) produtos finais, como código de programa, especificações de requisitos, projetos e estudos de testes; (2) informações de processo, como decisões de projeto e (3) conhecimentos do domínio da aplicação. Estruturas de dados, casos de testes e documentação são tipos adicionais de recursos que podem ser reusados. A Tabela 1 mostra alguns dos impedimentos identificados por eles para o reuso de *software*, agrupado por aspectos gerais, técnicos e não-técnicos.

Ainda em relação a reuso de informações e experiências, cabe citar a posição de Glass (1998, p. 57), de que deve haver algo seriamente errado sobre reuso, pois todos acreditam que ele é muito importante para a engenharia de *software*, mas que este potencial nunca é alcançado. Ele destaca ainda que muitos autores apontam deficiências de gestão como principais responsáveis pela situação, mas contra-argumenta citando que a principal razão para o problema é a falta de bons componentes de *software* que possam ser reusados.

O referido autor considera duas categorias principais de reuso: conceitual e de componentes. Na primeira inclui artefatos relacionados a engenharia, como arquitetura, requisitos e testes; padrões de projeto e interface. No reuso de componentes são considerados partes de código, módulos da aplicação ou serviços.

Segundo ele, a evolução do negócio de desenvolvimento de aplicações colocou obstáculos adicionais ao reuso de componentes. Um deles é a constatação de que construir um componente de *software* reusável leva mais tempo e é mais caro do que construir um componente apenas para um projeto.

Categoria	Aspectos da Pesquisa	Impedimento ao reuso de <i>software</i>
Aspectos Gerais	Definição e escopo	<ul style="list-style-type: none"> • Falta de terminologia aceita e entendida para descrever os conceitos
	Aspectos econômicos	<ul style="list-style-type: none"> • Investimento necessário para promover o reuso de <i>software</i>. • Falta de modelo econômico para explicar custos e benefícios do reuso de <i>software</i>
Aspectos Técnicos	Processo de Reuso de <i>Software</i>	<ul style="list-style-type: none"> • Falta de metodologia para criar e implementar reuso de <i>software</i>
	Tecnologias para reuso de <i>software</i>	<ul style="list-style-type: none"> • Ausência de recursos de <i>software</i> confiáveis para reuso. • Ausência de ferramentas e técnicas para o reuso de <i>software</i>.
Aspectos não-técnicos	Comportamentais	<ul style="list-style-type: none"> • Ausência de comprometimento e incentivo e treinamento para reuso de <i>software</i>. • Síndrome NIA (Não Inventado Aqui).
	Organizacionais	<ul style="list-style-type: none"> • Falta de suporte organizacional para institucionalização do reuso de <i>software</i> • Dificuldade em medir os benefícios do reuso.
	Legais e contratuais	<ul style="list-style-type: none"> • Problemas relacionados a direito de propriedade

Tabela 1 – Impedimentos ao reuso de *software*

Fonte: Kim e Stohr, 1998, p. 4

Outros obstáculos, estes mais associados a gestão, são a pressão por prazos de desenvolvimento cada vez menores, e a falta de uma sistemática de reconhecimento que valorize empregados que constroem e utilizam componentes reusáveis.

Já Lyytinen e Robey (1999, p. 92) falam de barreiras para o aprendizado em desenvolvimento de sistemas, focados na questão da aprendizagem organizacional. A primeira delas seria limitação na capacidade de aprendizagem, proporcional a capacidade de processamento de informações e construção de significado. As organizações avaliadas no estudo relataram que estavam tão empenhadas em cumprir os prazos previstos que não era possível relatar e avaliar experiências no âmbito dos projetos.

Os referidos autores citam que outra barreira é a falta de incentivos para o aprendizado. A obsessão pelo sucesso não permite o aproveitamento de aprendizado com as falhas, pois elas tendem a ser apagadas da memória organizacional. Apenas os sucessos merecem registro. Eles comentam ainda sobre a influência da formação excessivamente técnica dos profissionais de TI. O foco de sua formação e treinamento ocorre em ferramentas e tecnologia, o que dificulta a abordagem baseada em processo e no reuso de informações e experiências.

Assim, na maior parte das organizações, as experiências e práticas de trabalho de projetos anteriores não são capturadas e reaproveitadas, as equipes de desenvolvimento não se beneficiam de experiência disponível, e acabam por repetir erros já tratados em projetos anteriores.

Dentre os trabalhos realizados para mudar esta situação destacam-se reuniões de avaliação e a fábrica de experiências. A sistemática de reuniões de avaliação, também conhecidas como *postmortem analysis (PMA)*, tem sido usada como forma de capturar experiências e sugestões de melhoria de projetos concluídos.

Birk, Dingssoyr e Stalhane (2002, p. 43) descrevem que o objetivo é que os membros de uma equipe compartilhem suas experiências, reconheçam e registrem o que eles aprenderam durante o projeto. O modelo descrito é composto de três fases: preparação, coleta de dados e análise.

Eles destacam que, em função das atribuições e compromissos das equipes, é importante que a duração do processo seja ajustada em cada organização, para atender as limitações de tempo existentes.

O resultado é registrado em um relatório de experiências do projeto, contendo:

- Descrição do projeto – os produtos desenvolvidos, métodos usados, tempo e esforço necessários;
- Principais problemas identificados, com suas descrições e causas identificadas;
- Principais fatores de sucesso do projeto, com comentários da equipe;
- Transcrição da reunião de análise, para que a equipe possa avaliar como ela discutiu os acertos e problemas.

É recomendado que este tipo de reunião seja realizada quando o projeto for concluído, ou atingir um marco expressivo, em que é possível obter experiências que possam ser utilizadas em outros projetos.

Os autores não recomendam o uso do método no meio do projeto, ou quando houver conflitos na equipe, em que haverá dificuldade em discutir os problemas, e não será possível ter foco em oportunidades de melhoria.

Outro trabalho desenvolvido nesta área, e uma das principais pesquisas em gestão da informação e engenharia de *software* é a fábrica de experiências (do termo em inglês *Experience Factory*), um modelo de captura e reuso de informações e conhecimento desenvolvido na Universidade de Maryland.

Ela será descrita a seguir, bem como três projetos que a utilizaram. O primeiro deles foi realizado na administração espacial norte-americana (*NASA – National Aeronautics and Space Administration*), em conjunto com a Universidade de Maryland. Os demais foram realizados em duas empresas européias – a Ericsson e a Daimler-Chrysler.

Eles foram escolhidos para mostrar como a idéia pode ter implementações distintas, em função do contexto das organizações, seus valores e cultura.

3.5.3. Um Modelo Geral para Reuso de Experiências em Engenharia de Software

Basili, Caldiera e Rombach (1994, p. 2) destacam que o sucesso em desenvolvimento de *software* depende de alguns requisitos básicos, como entender de processos e produtos de *software*, avaliar as falhas e sucessos nos projetos e aprender com as experiências passadas para ter melhores resultados.

Para eles, é fundamental a melhoria contínua no processo de desenvolvimento, o empacotamento das experiências para reuso e a compreensão de certos conceitos relativos ao desenvolvimento de sistemas. Dentre eles é possível destacar:

- Projetos de *software* apresentam características diversas, e há uma relação direta entre processo e produto. Assim, um modelo de processo único de desenvolvimento não funcionará em todas as situações;
- Há necessidade de métricas, que devem ser baseadas em objetivos e modelos para garantir visibilidade do andamento do projeto;
- O reuso de experiências e informações, na forma de processos e produtos, é importante para alcançar melhoria no processo de desenvolvimento;
- As experiências devem ser avaliadas para potencial reuso – é necessário um processo de análise para verificar de que forma ela pode ser aproveitada;
- As experiências podem ser empacotadas de várias maneiras – equações, histogramas, algoritmos, relatórios e descrição de contexto. Este empacotamento é fundamental para seu aproveitamento em novos projetos.

Para atender as necessidades do processo de desenvolvimento de *software*, foi criada uma solução baseada em paradigma de evolução da qualidade adaptado para *software* – o paradigma de melhoria de qualidade (QIP – *quality improvement paradigm*).

O paradigma é suportado por uma ferramenta para definir metas para os projetos, um mecanismo para aferição destas metas, e por uma abordagem organizacional para a construção de competência em *software* e sua oferta para a organização – a fábrica de experiências.

O QIP foi desenvolvido por Basili, sendo resultado da aplicação de método científico ao problema da melhoria de qualidade do *software*. Ele é relacionado ao ciclo PDCA (*Plan/Do/Check/Act*) da área de gestão da qualidade, e articulado em seis passos:

- Caracterização – entendimento do ambiente, com base em modelos e intuição acerca de dados e informação;
- Definição de Objetivos – estabelecer objetivos para o sucesso do projeto e melhoria da performance organizacional;
- Escolha do Processo – com base na caracterização do ambiente e os objetivos estabelecidos, escolher processo mais adequado ao projeto;

- Execução – construção do produto;
- Análise – analisar dados e informações coletadas ao final de cada projeto para avaliar as práticas, determinar problemas e fazer recomendações para melhorias futuras;
- Empacotamento – consolidação das experiências obtidas na forma de novas informações obtidas dos projetos, e armazenamento em uma base de experiências para projetos futuros.

A Figura 7 mostra uma representação do QIP, com seus seis passos.

A correta aplicação do paradigma depende da caracterização do projeto e sua classificação em relação a um conjunto de características, o que permite o agrupamento dos projetos de acordo com objetivos e características similares. A caracterização, também chamada de domínio de negócio, oferece contexto para definição de objetivos, seleção de processos, avaliação e reuso de experiências.

Também devem ser considerados fatores relacionados ao projeto e seu ambiente, que afetam o processo de desenvolvimento de *software* e o produto final. Eles são relacionados às pessoas (número de pessoas e nível de conhecimento), domínio do sistema e domínio da tecnologia, entre outros.

Neste cenário é importante a definição de objetivos que estejam relacionados a caracterização do ambiente. Eles devem ser mensuráveis, orientados a modelo e definidos a partir de perspectivas diversas, como os usuários, projeto e organização.

A coleta de dados deve ser considerada como parte do processo, e não um acessório, e podem ser coletados dados de recursos, como esforço por atividade, fase e tipo de equipe utilizado; dados de defeitos e mudanças no projeto; dados de medições no processo, com conformidade; e dados sobre as características do produto.

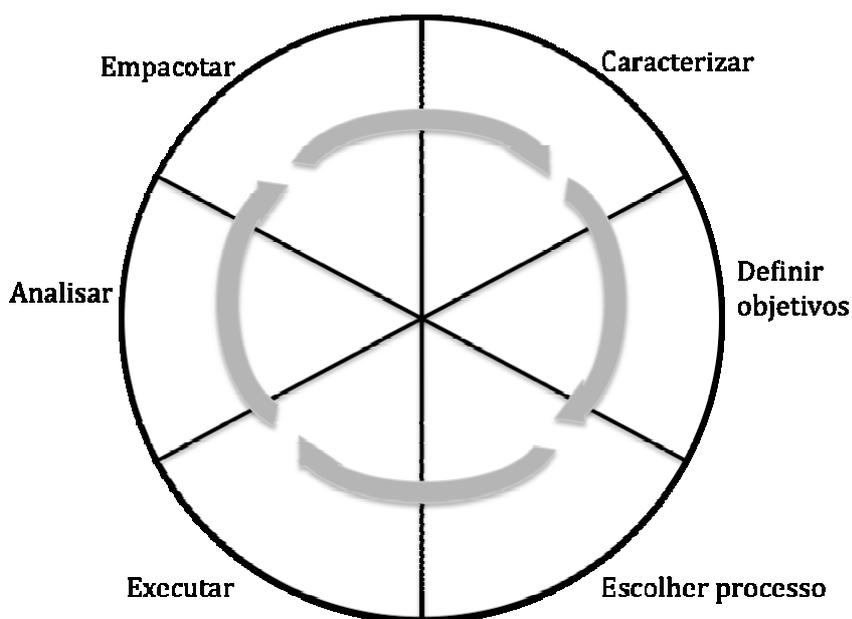


Figura 7 - Paradigma da melhoria da qualidade (QIP)

Fonte: Basili, Caldiera e Rombach, 1994, p. 5

Os autores concluem afirmando que o QIP é baseado na noção de que a melhoria nos produtos e processos de *software* requer a acumulação contínua de experiências (aprendizado), que possam ser entendidas e modificadas (modelos de experiência) e registradas em um repositório de experiências (base de experiências).

As experiências registradas podem ser acessadas e modificadas para atender necessidades de novos projetos, criando a possibilidade de reuso.

O modelo criado por Basili prevê a separação física ou lógica do desenvolvimento do projeto, realizado pela “organização de projeto”, e da atividade de aprendizagem sistemática e empacotamento de experiências reusáveis – realizada pela “fábrica de experiências”.

A fábrica de experiências (FE) - modelo descrito na Figura 8 - é uma infraestrutura para suportar o reuso de experiências, processos e produtos de *software*. As experiências são coletadas de projetos de desenvolvimento, avaliadas, empacotadas e armazenadas em uma base de dados.

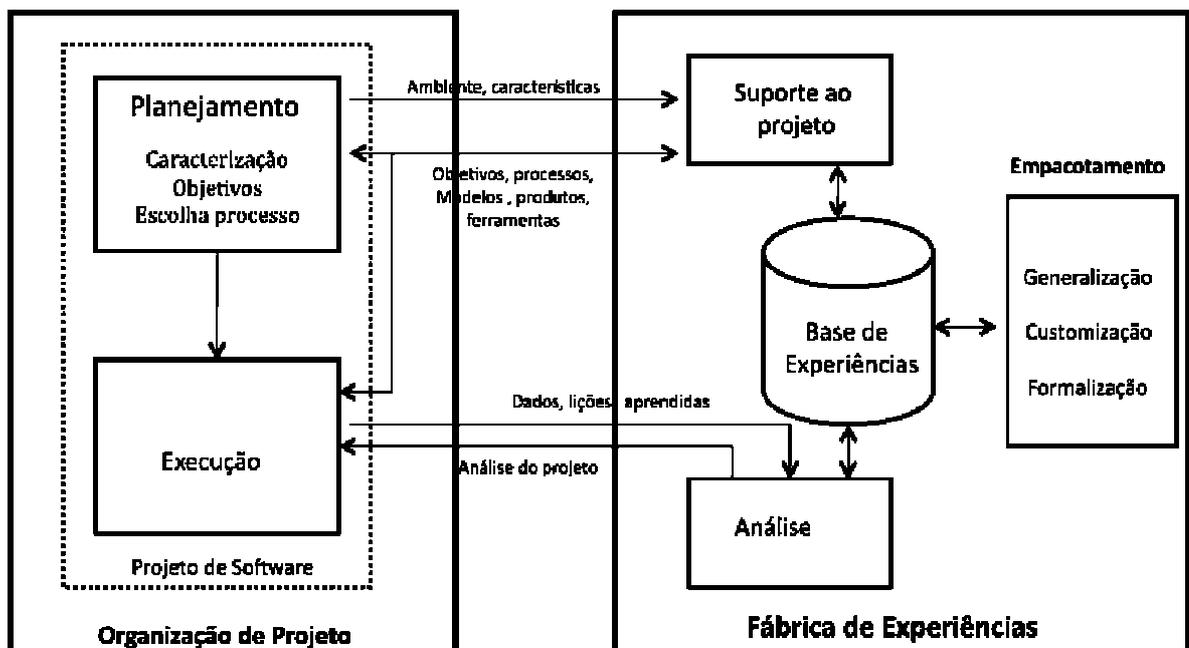


Figura 8 - A Fábrica de Experiências (FE)

Fonte: Basili, Caldiera e Rombach, 1994, p. 9

Segundo Dingsoyr (2000, p. 56), entende-se por “empacotamento” de uma experiência sua generalização, formalização e customização, para que ela seja reusável em outro projeto. Ele descreve os seguintes exemplos de pacotes:

- Pacotes de produtos – informações sobre ciclo de vida de um produto, informações sobre como reusá-lo e lições aprendidas com o reuso;
- Processos – informações sobre como executar um processo de ciclo de vida.;

- Ferramentas – instruções sobre o uso de ferramentas e experiências com elas;
- Gestão – informações de referência para gerentes de projeto;
- Dados – informação de referência para outros projetos ou atividades de *software*.

O empacotamento de experiências é realizado pelo uso de modelos e métricas de processos de *software*, e outras formas de conhecimento, via pessoas e documentos.

A organização de projeto, cujo objetivo é desenvolver e manter produtos de *software*, fornece à fábrica de experiências (FE) informações sobre características e ambiente do projeto, planos, dados de uso de recursos, registros de qualidade e informações do processo de desenvolvimento. Ela também oferece informações de desempenho dos modelos utilizados pelo projeto.

Segundo Basili (1994, p. 472) a FE implementa uma base de experiências com um conjunto integrado e acessível de modelos de experiência, analisado, sintetizado e empacotado, que capturam experiências de projetos passados.

O reuso sistemático nos projetos exige uma estrutura organizacional específica, com um modelo de evolução de *software* que suporte o reuso de experiências, um conjunto de processos de aprendizagem e armazenamento de experiências; e a integração das duas funções. O autor defende que a FE é a unidade organizacional que realiza esta integração.

Ainda segundo Basili, é importante entender que o reuso de artefatos, no domínio de *software*, é um objetivo sempre perseguido, com um histórico de pouco sucesso. Dentre as razões para isto é citado que o reuso não se refere apenas ao código de programas, ele deve englobar também o contexto de desenvolvimento.

Outra razão é que o reuso de experiências é geralmente informal, e não é incorporado aos processos de desenvolvimento de *software*. Também é citado que as experiências não são corretamente avaliadas para seu potencial de reuso, e que elas podem ter necessidade de customização para se adaptarem ao novo projeto.

Finalmente, Basili argumenta que o foco dos projetos é a entrega, e não o reuso, razão pela qual ele defende que as atividades que criam pacotes de experiência reusáveis estejam fora dos projetos. Isto cria a necessidade de organizações separadas, pelo menos logicamente: a organização de projetos para o desenvolvimento de produtos e a fábrica de experiências para a criação dos pacotes de experiências.

3.5.4. A Experiência da NASA – O Laboratório de Engenharia de *Software*

A experiência da NASA teve início em 1976, durante reuniões entre uma de suas unidades, o *Godard Space Flight Center* (GSFC) e a Universidade de Maryland, com objetivo de criar um programa de pesquisa em desenvolvimento de *software*.

Basilli et. al. (2002, p. 1) descrevem que a criação do SEL (*Software Engineering Laboratory*) teve como principais objetivos promover o entendimento de:

- Princípios básicos envolvidos em *software* e seu processo de desenvolvimento;
- Fatores que afetam o desenvolvimento de *software* e seu relacionamento;
- As características de classes de problemas e produtos; os tipos de problemas encontrados e os erros durante o desenvolvimento, bem como o relacionamento entre metodologia e controle gerencial.

Desde o início a metodologia básica do laboratório era realizar experimentos pela observação dos projetos em desenvolvimento na NASA, e coletar dados relevantes para alcançar os objetivos definidos.

A experiência pode ser dividida em três períodos distintos. O período inicial compreende os anos entre 1976 e 1983, cuja ênfase foi na coleta de dados. Neste período identificou-se que a atividade de coleta de dados exigia um processo rigoroso, e pessoal especializado, e que havia necessidade de estabelecer um compromisso entre o volume de dados desejado, e o que podia realmente ser coletado.

Foi observada uma sobrecarga da atividade de coleta e análise de dados, em cerca de 10% do esforço planejado para cada projeto. Este dado exigiu um acordo entre a NASA, a Universidade e o prestador de serviços, para que fosse incorporado ao contrato de desenvolvimento. Por outro lado, verificou-se que não houve alteração significativa no comportamento das equipes de desenvolvimento em função da atividade de coleta de dados.

O segundo período compreende os anos entre 1984 e 1994, em que, já com conhecimento do ambiente, e com a estabilidade da atividade de coleta de dados, foi possível integrar os conceitos de QIP e fábrica de experiências. Esta integração permitiu a associação do conceito da fábrica de experiências a construção de uma organização voltada para a aprendizagem na área de desenvolvimento de *software*.

Nesta fase verificou-se que o uso de boas práticas para desenvolvimento de *software* propiciou uma melhoria significativa nos indicadores de desenvolvimento do SEL. Os resultados mais expressivos foram obtidos entre os anos de 1987 e 1991, como uma redução de 75% na taxa de defeitos, redução de 55% nos custos de desenvolvimento e um aumento de cerca de 300% na taxa de reuso dos projetos.

No período entre 1991 e 1995 as taxas de reuso foram de 35%, 42% e 8%, respectivamente, mostrando a evolução do modelo, embora com taxas de melhoria mais modestas.

A partir de 1995 a NASA passou por processo de reestruturação, motivada por redução de seu orçamento no governo, e uma reorganização do GSFC, com o afastamento da maior parte da equipe de gestão de desenvolvimento, o que começou a afetar o andamento das atividades do SEL.

A nova gestão estabeleceu objetivos e metas de redução de prazos de entrega e custos, que terminaram por reduzir o contato do SEL com os projetos.

Desta forma o laboratório teve dificuldade em encontrar novos projetos, e a própria administração da NASA passou a não enxergar mais valor no trabalho do SEL, que acabou suspenso oficialmente em 2001.

Nos cerca de 25 anos de funcionamento o projeto teve uma história de sucesso na ampliação dos conhecimentos relacionados ao processo de desenvolvimento de *software*. O uso do arcabouço da fábrica de experiências para apoiar uma ciência experimental de engenharia de *software* é considerado a melhor abordagem para o tema.

O estudo do projeto SEL conclui listando lições que, se solucionadas, podem auxiliar atividades de evolução do processo no futuro. As lições foram agrupadas nas seguintes categorias.

- Coleta de dados - a coleta de dados exige um processo rigoroso e pessoal especializado; deve haver um compromisso em solicitar apenas as informações que serão efetivamente usadas; e deve ser estabelecida uma base para os produtos, processos e métricas, para permitir avaliação do progresso ao longo do tempo;
- Gestão - o suporte do nível decisório é fundamental; apesar da importância da coleta de dados, os prazos de entrega do produto serão sempre mais importantes; e a organização deve ter poder de decisão sobre o processo de melhoria do processo;
- Foco na agenda de pesquisa - há uma forte relação entre pesquisa e prática em engenharia de *software*, e as duas atividades se beneficiam disso; e a proximidade entre pesquisadores e desenvolvedores beneficia aos dois grupos;
- Gestão de Pessoas – o treinamento para coleta de dados é essencial; e necessidade de administrar a tensão contínua entre a necessidade de *feedback* rápido para os desenvolvedores, e a necessidade de dedicar tempo à análise e construção de experiências.

3.5.5. A Experiência da DaimlerChrysler – Centro de Experiência em *Software*

O segundo modelo de implementação do conceito de fábrica de experiências é o projeto da fabricante de automóveis DaimlerChrysler, denominado Centro de Experiência em *Software* (*Software Experience Center – SEC*). Ele é descrito por Houdeck, Schneider e Wieser (1998) e Schneider, Hunnius e Basili (2002), e parte da premissa de que um fabricante de automóveis tem alta dependência de *software* de qualidade para suas unidades eletrônicas de controle.

Os objetivos do projeto eram a melhoria da qualidade do *software* desenvolvido, e o reuso de conhecimento, já que a empresa considerava que o reuso de experiências era fundamental para uma melhor performance de seus projetos, em função do uso adequado de um recurso escasso, a competência em desenvolvimento de sistemas.

O ponto de partida para o projeto foi a implementação na NASA do conceito de FE, mas o ambiente e a cultura da DaimlerChrysler exigiram modificações do

conceito original. Ficou claro que, em função do prazo previsto para o projeto, cerca de dois anos, não seria possível adotar o foco intensivo em métricas da FE, que utilizou cerca de três anos apenas para estabelecer padrões e métodos de coleta de dados.

O projeto SEC teria maior atenção em aspectos qualitativos voltados para a melhoria do processo de desenvolvimento, com objetivo de implementar nas equipes de *software* da DaimlerChrysler o conceito de organizações de aprendizagem.

Os artigos descrevem a criação de uma base de experiências, conceituada como o armazenamento constante de experiências relacionadas ao processo de desenvolvimento de *software*. Para a montagem desta base foram designadas pessoas para realização de entrevistas e coleta de formulários de *feedback* com líderes de projeto.

O empacotamento de experiências é descrito como a comparação, análise e combinação de experiências relatadas em projetos diversos, e organizado de acordo com as etapas do processo de desenvolvimento de *software*. A catalogação das experiências foi considerada um grande desafio, pela dificuldade de relatar os diferentes contextos e torná-las relevantes para outros usuários em situações reais.

As experiências catalogadas foram consolidadas na Intranet da empresa, junto a descrição do processo de *software*, associando cada etapa do processo a materiais de treinamento, experiências, listas de verificação (*checklists*), e contatos de pessoas que poderiam auxiliar em diferentes contextos. Os autores destacam que os procedimentos de descrição de contextos e reuso foram mais importantes do que o uso de ferramentas para automação do processo.

Os relatos mostram a dificuldade em coletar dados sobre o reuso das experiências nos diversos projetos, mas concluem que os resultados foram positivos para a organização, e que o projeto teve incentivos para continuar. Eles descrevem algumas recomendações, como a elaboração de técnicas de empacotamento de experiências que privilegiem a simplicidade e a interação com os usuários, para garantir que elas sejam utilizáveis e relevantes para os projetos.

Os dois trabalhos concluem que a implementação de abordagens de aprendizagem organizacional para o desenvolvimento de *software* implica em alterações nos modelos e processos de trabalho, que devem ser adotadas por toda a organização. Caso isto não ocorra a melhoria nos resultados não acontecerá e o investimento em esforço e tempo será desperdiçado.

3.5.6. A Experiência da Ericsson Technology – A Máquina de Experiências

O terceiro modelo de implementação da fábrica de experiências para a área de desenvolvimento de sistemas de informação a ser avaliado é o da Ericsson *Software Technology*, empresa responsável pelo desenvolvimento de sistemas para centrais telefônicas e telefonia celular.

Johansson, Hall e Coquard (1999), descrevem projeto realizado a partir de 1998, e que teve como base o modelo da FE. O problema central era o mesmo de outras organizações envolvidas com engenharia de *software*, isto é, como

aproveitar experiências de projetos já concluídos para acelerar os prazos de desenvolvimento e evitar erros já conhecidos.

Os autores deixam claro que a abordagem usada por Basili ao descrever a FE não poderia ser adotada na Ericsson, pois implicaria em grande mudança organizacional para a criação de grupos separados, e alterações em rotinas de trabalho consideradas adequadas.

Foi criado então o conceito de “máquina de experiências” (*experience engine*), um modelo para trabalhar de forma contínua a transferência e reciclagem de experiências na empresa. Ele tem como base para seu funcionamento a idéia de que a maior parte da transferência de conhecimento em engenharia de *software* é feita em interações face-a-face, e não em consultas a bases de dados.

Foi identificado que a troca de experiências pode ser separada em três grupos principais: gestão, envolvendo planejamento do projeto, relatórios de progresso e avaliação de mudanças de requisitos; experiências técnicas, relacionadas a decisões de projeto e codificação e soluções adotadas para problemas específicos; e negócios, envolvendo negociação e acordos com os clientes.

Partiu-se então para a criação de uma base de dados para armazenamento das experiências. Embora a definição original da FE não tenha foco na criação de uma grande base de dados, os autores destacam que é fácil cair na armadilha de registrar todos os eventos e ocorrências, e utilizar um número excessivo de métricas.

No caso da Ericsson, a abordagem inicial incluiu uma grande base de dados para este fim mas, após um período de avaliação, concluiu-se que esta não era a melhor solução para a organização. O volume de métricas e dados era tão extenso que raramente era avaliado para uso na melhoria do processo.

Decidiu-se então concentrar os esforços em um número menor de métricas. Os autores observam que, embora o tempo usado nesta avaliação tenha sido inicialmente considerado como desperdício, ele foi importante para mostrar o caminho mais adequado para o projeto.

Identificou-se ainda que as métricas não seriam a única base para a decisão, em especial considerando que os projetos diferiam muito em tamanho, complexidade e experiência das equipes envolvidas.

O objetivo principal do projeto da Ericsson era a transferência de experiências entre equipes e, com base na cultura da empresa, concluiu-se que a comunicação verbal, preferencialmente em reuniões presenciais, seria a melhor forma de comunicar e espalhar estas experiências.

Foram então identificadas três categorias de troca de experiências: *flashes*, situações de aprendizagem e educação/treinamento. Esta última foi identificada como a falta de treinamento formal em alguma área, e não é tratada pelo projeto.

Johansson, Hall e Coquard (1999) explicam que as categorias de troca de experiências não pretendem ser uma regra rígida ou formalização da relação entre as pessoas, e foram criadas apenas para ser um guia prático para o projeto.

Flashes representam a categoria mais comum, e são caracterizados por interações curtas. Podem ser mediadas por um agente – *broker flashes* – ou apenas comunicadas de forma tradicional – *communicator flashes*. Elas

geralmente ocorrem dentro da organização, mas não há limitação que aconteça também com pessoal externo.

O projeto também criou dois papéis novos nas equipes de desenvolvimento de *software*, o agente de experiências (*experience broker*) e o comunicador de experiências (*experience communicator*). O papel do agente é formal, e eles devem estar visíveis e acessíveis para as equipes, para facilitar a busca de experiências. Já o comunicador tem um papel informal na organização.

Os agentes de experiências atuam como pólo central do processo. Devem ter bom conhecimento da empresa – seus produtos, processos e projetos em andamento, e seu papel pode ser caracterizado como:

- Identificar comunicadores de experiências, nas diversas áreas de atuação;
- Facilitar o contato com os comunicadores de experiências;
- Empenhar-se para conectar as pessoas de áreas e equipes diversas;
- Ser percebido como sempre disponível;
- Tornar percebidos os benefícios de situações de *flashes* e aprendizagem.

O papel de agente é formalizado pela organização, e a pessoa indicada deve uma boa rede de contatos e ser respeitado como generalista.

Já os comunicadores são pessoas com experiência, desejo e boa vontade em compartilhá-las, com alguma capacidade de transmitir conhecimentos. Nem toda pessoa com experiência em determinado assunto poderá atuar como comunicador.

A premissa considerada no projeto é que o papel não permite que ele execute a tarefa de outros – ele irá auxiliar a pessoa a resolver seus problemas. Ele não é considerado como parte da equipe regular do projeto, pois embora isto possa auxiliar o trabalho em curto prazo, contraria a idéia da máquina de experiências, pois nesta situação haveria pouca ou nenhuma transferência de conhecimento.

Os autores destacam que o papel dos comunicadores é informal, e deve continuar desta forma, já que pesquisas de comportamento mostravam excesso de papéis formalizados na organização, com baixo índice de aceitação pelos empregados. Foi recomendado ainda que eles devem ter um mecanismo formal de reconhecimento, na forma de remuneração pelo desempenho da atividade.

O relato conclui que o projeto procurou mostrar a viabilidade de gerenciar a experiência organizacional sem o uso intensivo de grandes bases de dados e de um grande número de métricas de controle, pouco afeitas a cultura da Ericsson.

Em contraponto, esta abordagem tem forte dependência de pessoas com grande experiência e credibilidade na organização, e com perfil e atitude para atuar como agentes ou comunicadores. Este ponto deve ser considerado, pois estas pessoas podem ter demanda para atuar em outras funções na empresa, ou mesmo serem alvos dos concorrentes em função de sua experiência.

É destacada recomendação de avaliar os pontos principais da máquina de experiências para a continuidade do projeto, em especial os processos de coleta

e compartilhamento de experiências, pois já foi identificada a necessidade de aumentar a quantidade de pessoas disponíveis para atuar como agentes e comunicadores.

3.5.7. Conclusões

Assim, como informação e conhecimento, os termos gestão da informação e gestão do conhecimento são usados muitas vezes para descrever os mesmos processos. Alvarenga Neto, Wilson e Michaud destacam que o conhecimento está na cabeça das pessoas, razão pela qual não pode ser diretamente gerenciado.

Eles defendem a gestão da informação, mostrando que ela é também mais adequada aos modelos e tecnologias disponíveis hoje. Ela deve incluir as informações sobre os conhecimentos disponíveis em uma organização, uma vez que há um processo contínuo de conversão de conhecimento em informação.

A gestão da informação começa a ser considerada ferramenta importante para a engenharia de *software* ao final dos anos 80, quando se percebe que apenas soluções de tecnologia, como as descritas por Glass (1999) não serão suficientes para a evolução desejada para os sistemas de informação.

Rus *et. al* (2001□□□□) complementam ao relatar que o desenvolvimento de *software* é uma atividade intensiva em pessoas e conhecimento, e que empresas de desenvolvimento de sistemas tem o capital intelectual como um de seus principais ativos.

Esta constatação motivou pesquisas e projetos com foco no reuso de informações e experiências de projetos anteriores, seja pelo aproveitamento de artefatos, seja pela criação de um ambiente de aprendizagem entre as equipes.

Apesar disto, Desouza, Glass, Rus, Lyttne e Robey descrevem barreiras reais para o uso de modelos de gestão da informação em organizações de *software*:

- Dificuldade em capturar e codificar o conhecimento, pois projetos de *software* são específicos e contextuais;
- Pressão de prazos para entrega de sistemas, o que dificulta o registro de experiências;
- Pressão para evolução tecnológica, o que dificulta o reuso de soluções e experiências;
- Ausência de cultura de reuso de artefatos e experiências em organizações de desenvolvimento de sistemas;
- Formação de profissionais com foco em aspectos técnicos, dificultando a aplicação de modelos de processos para reuso de informações;
- Falta de incentivo para a aprendizagem, especialmente em situações de problemas ou falhas, que geralmente não são registradas como parte da memória organizacional.

Neste contexto, foi desenvolvido projeto no final dos anos 80 para captura e reuso de informações e conhecimento em projetos de *software*, que ainda hoje

serve de base para diversas organizações. A Fábrica de Experiências (FE) foi proposta com a premissa de que o sucesso em projetos de *software* depende do entendimento dos processos e produtos, da avaliação das falhas e sucessos de projetos anteriores e do aprendizado com experiências passadas.

Ao defender a necessidade de melhoria contínua no processo de desenvolvimento, Basili, Caldiera e Rombach (1994) propõem a aplicação de um paradigma de evolução da qualidade para o processo de desenvolvimento. Eles concluem afirmando que a melhoria nos produtos e processos de *software* requer a acumulação de experiências, que possam ser entendidas, adaptadas para reuso e registradas em um repositório.

A proposta da FE é criar experiências – o conceito de empacotamento prevê a generalização, formalização e customização - de forma a permitir o reuso em outros projetos. Os pacotes podem estar em forma de produtos, processos, ferramentas, dados e modelos de gestão.

Como o foco das organizações de *software* é a entrega dos produtos no prazo, e não o reuso em si, Basili (1994) defende a criação de uma estrutura separada, para cuidar das tarefas de captura e codificação das experiências.

Foram consideradas as implementações da FE na NASA, DaimlerChrysler e Ericsson, sendo que apenas a primeira delas conseguiu estabelecer um modelo de coleta de dados como parte do processo. Nas demais obteve-se informações qualitativas.

A experiência da NASA – SEL (*Software Engineering Laboratory*) – foi realizada com os idealizadores da FE, e seguiu fielmente o projeto original. Ela constatou que o processo da coleta de dados para avaliação dos projetos implicou em acréscimo de cerca de 10% no esforço previsto para os projetos, compensados pelos resultados obtidos nos primeiros anos, relativos a redução de erros e redução de custos nos projetos.

A experiência da DaimlerChrysler – SEC (*Software Experience Center*) teve como objetivos principais a melhoria da qualidade dos produtos de *software*, e o reuso de experiências. Em função das características do projeto, decidiu-se pela não adoção do modelo de métricas preconizado pela FE. Foram relatados resultados positivos, embora reconhecendo a dificuldade em coletar dados sobre o reuso das experiências.

Na Ericsson Technology o projeto foi denominado “máquina de experiências”, e o modelo da FE passou por adaptações, pois não seria possível criar estruturas separadas para geração de informações para reuso. O objetivo principal do projeto era a transferência de experiências entre as equipes.

Foram definidos três tipos de troca de experiências, e dois papéis nas equipes e desenvolvimento para cuidar da organização e garantir o acesso a pessoas com experiência sempre que necessário. O projeto relatou resultados positivos, que recomendaram sua continuidade, apesar de observarem a forte dependência das pessoas a serem escolhidas para os novos papéis no desenvolvimento.

As experiências de FE avaliadas concluem que a implementação de modelos de aprendizagem organizacional para o desenvolvimento de *software* implica em alterações nos modelos e processos de trabalho, que devem ser adotadas por toda a organização.

3.6. Conclusões da Revisão de Literatura

A pesquisa em bases de teses e dissertações mostrou que pouco trabalhos associam os temas gestão da informação e desenvolvimento de sistemas de informação, com o foco em aspectos da ciência da informação. A maior parte das pesquisas encontradas situa-se na área de ciência da computação, e referem-se a avaliação e implementação de ferramentas. Os profissionais da área de *software* concentram seus estudos em ferramentas e métodos.

Este trabalho busca avaliar os aspectos relacionados ao fluxo e a gestão das informações utilizadas no processo de desenvolvimento de *software*, com foco na atividade de engenharia de requisitos. Ele destaca os aspectos relacionados a identificação, captura, processamento e uso da informação, com base em idéias e propostas da ciência da informação.

O conceito de CI adotado na pesquisa é o elaborado por Borko (1968, p.1), e que destaca a preocupação com as propriedades, o comportamento da informação, as forças que governam seu fluxo, e os meios de processamento da informação para otimizar seu acesso e uso.

A CI como disciplina evoluiu com a importância do uso da informação, e os estudos passam a tratar da compreensão, dos fluxos, meios de coleta, armazenamento, processamento e uso. A partir dos anos 90 ela ganhar um contexto complexo e multidimensional, principalmente em função de sua interdisciplinariedade, e sua relação com outras disciplinas, como a biblioteconomia, ciência da computação e gestão. Saracevic (1996) avalia que a evolução da sociedade da informação e a evolução tecnológica têm exercido pressão sobre a CI, e exigido uma mudança de perspectiva em relação a outras disciplinas.

Neste cenário, de forma crescente, a informação passa a ser valorizada como recurso, e ajuda a definir a competitividade de pessoas, grupos e produtos. Apesar da importância do termo “informação”, autores como Wurman (1991) e Robredo (2003) consideram que ele é utilizado de forma ambígua, por vezes representando dados, informação ou conhecimento, e sem tratar as diferenças existentes entre eles.

A literatura registra o esforço dos cientistas para tornar claras estas diferenças, como é o caso de Davenport (1997), ao afirmar que a informação é o veículo que usamos para expressar e comunicar conhecimentos, e que a informação tem mais valor que dados, e também mais ambiguidade. Já Roberts (2000) explica que dados são matéria prima para a produção da informação, e que sozinhos não têm significado.

Em relação a informação e ao conhecimento, Nonaka e Takeuchi (1995) ressaltam que ambos são dependentes de contexto, da situação, dizem respeito a significados e são criados de forma dinâmica. Eles entendem que a informação é um meio necessário para representação do conhecimento.

Wilson (2002) defende a idéia de que tudo o que está fora da mente das pessoas pode ser classificado como dados, se consiste de simples fatos, e informação, se os dados forem complementados por um contexto relevante. Ele

afirma ainda que a gestão da informação é: “[...] aplicação de princípios de gestão para a aquisição, o controle, a disseminação e o uso da informação [...]”

Alvarenga Neto, Wilson e Michaud destacam que o conhecimento está na cabeça das pessoas, razão pela qual não pode ser diretamente gerenciado. Eles defendem a gestão da informação, mostrando que ela é também mais adequada aos modelos e tecnologias disponíveis hoje. Ela deve incluir as informações sobre os conhecimentos disponíveis em uma organização, uma vez que há um processo contínuo de conversão de conhecimento em informação.

A gestão da informação, segundo Tarapanoff (2006) e Moresi (2000), deve estar associada a estratégia das organizações, via integração das informações e dos processos. Neste cenário, Davenport (1997) e Choo (2006) mostram que o fluxo de informações nas organizações é associado a um ciclo básico, que regula a busca, armazenamento e uso das informações.

O ciclo de vida proposto por Choo, base para esta pesquisa, é composto por seis processos: identificação de necessidades; aquisição; organização e armazenamento; produtos e serviços de informação; distribuição e uso da informação. Ele se preocupa com os aspectos relacionados ao fluxo de informações, influência do ambiente organizacional, os usuários e suas expectativas.

Um ponto de destaque no ciclo proposto é o processo que trata dos serviços de informação, que devem preocupar-se com a oferta de informações relevantes, que aumentem o potencial de uso, e a agregação de valor a informação que será processada, para auxiliar os usuários em seus processos de ação e tomada de decisão.

O desenvolvimento de sistemas é associado a vários processos da gestão da informação, e se tornou um componente crítico na estratégia competitiva das organizações, já que muitos negócios, ou parte de negócios, dependem do uso de sistemas para desempenho de suas atividades e atendimento a seus clientes.

Apesar desta importância, e do gasto do Brasil com desenvolvimento de sistemas ter atingido cerca de R\$ 11 bilhões em 2006, os dados mostram que os usuários ainda estão insatisfeitos com os resultados já alcançados, já que quase 20% dos projetos de *software* foram cancelados em 2006.

Basili, Caldiera e Rombach (1994) afirmam que a disciplina de desenvolvimento de *software* é essencialmente experimental, evolucionária e não-repetitiva. Kruchten (2000), Dingsoyr (2002) e Rus (2001) complementam, ao afirmarem que há causas específicas para este resultado, como entendimento inadequado das necessidades dos usuários, falta de habilidade para tratar mudanças de requisitos, evolução constante de tecnologia e a diferença natural que existe entre projetos de *software* para áreas diversas, o que dificulta o aprendizado e a repetição de práticas de sucesso.

Neste contexto foi criado o conceito de engenharia de *software*, que incentiva o uso de técnicas e sistemática repetíveis, similares as de outras disciplinas de engenharia, visando a garantia de obtenção de resultados mais previsíveis em termos de qualidade, custos e prazos.

Esta iniciativa foi complementada com a utilização de processos padronizados e de modelos de maturidade para desenvolvimento de *software*. A utilização de processos padronizados oferece um guia para as equipes de

desenvolvimento, ao especificar os tipos de artefatos que devem ser construídos, direcionar as tarefas dos componentes das equipes e oferecer métricas para a avaliação dos produtos e atividades ao longo do projeto. O modelo mais conhecido na área de *software* é o CMMI, que define cinco níveis de maturidade e um conjunto de melhores práticas em desenvolvimento de sistemas.

Uma área crítica em projetos de desenvolvimento é a identificação das necessidades que devem ser atendidas pelo sistema – os requisitos. Seu levantamento e registro são atividades complexas, de natureza multidisciplinar e dependente de pessoas. A definição de requisitos explicita a importância do contexto do projeto, as restrições ou características específicas que devem ser atendidas, além das especificações funcionais.

Os requisitos não-funcionais sofrem influência do domínio de negócio a que o sistema pertence, como restrições de segurança e desempenho; ou de tecnologia, como compatibilidade com ambientes operacionais específicos. A engenharia de requisitos, assim como ciência da informação, tem natureza multidisciplinar, sendo uma atividade centrada em pessoas, e afetada pela forma como as pessoas percebem seu ambiente de trabalho e sua relação com outras pessoas.

A escolha desta atividade para este trabalho é que ela trata aspectos ligados a identificação e codificação de conhecimento, e que a identificação e correção de erros nesta fase do projeto de *software* pode custar até 1.000 vezes mais barato do que em outras fases do projeto. (PRESSMAN).

No final dos anos 80 a gestão da informação começa a ser considerada ferramenta relevante para a engenharia de *software*, quando se percebe que apenas soluções de tecnologia não serão suficientes para a evolução desejada nos produtos de *software* (GLASS). A constatação de que o desenvolvimento de *software* é uma atividade intensiva em pessoas e conhecimento motivou pesquisas e projetos com foco no reuso de informações e experiências de projetos anteriores.

Um ponto relevante, conforme Dingsoyr (2000), é o reuso de informações e experiências. A identificação de domínios de negócio e o contexto do projeto auxiliam este reuso, ao agrupar projetos com características similares. Apesar da evolução nesta área, estudos mostram barreiras que devem ser vencidas pelas empresas para o alcance dos objetivos de reuso.

Estas barreiras podem ser classificadas como de natureza técnica, gerais de projeto e aspectos comportamentais e organizacionais, como a pressão para troca de tecnologias e entrega de produtos no prazo, bem como uma ausência da cultura de reuso nas organizações de desenvolvimento de *software*.

Dentre os projetos que tentaram lidar com a questão de informação e conhecimento em projetos de *software* está a fábrica de experiências (FE), proposta por Basili e baseada na premissa de que o sucesso em projetos de *software* depende do entendimento dos processos e produtos, avaliação de falhas e sucesso de projetos anteriores.

A proposta da FE é criar uma equipe separada da equipe do projeto, e que fica responsável pela identificação de experiências candidatas, que podem ser compartilhadas com outros projetos. Estas experiências serão empacotadas, processo que prevê sua generalização, formalização e customização, para

permitir o reuso em outros projetos. Os pacotes podem estar em forma de produtos, processos, ferramentas, dados e modelos de gestão.

A FE trabalha com a premissa de que a melhoria dos processos e produtos de *software* requer a acumulação contínua de experiências, que possam ser entendidas, modificadas e registradas em um repositório de experiências.

A FE foi registrada por três organizações distintas em sua forma de operação – NASA, DaimlerChrysler e Ericson – e houve registro de resultados positivos em todas elas, embora em graus diversos. Foi observado também que o uso de modelos de aprendizagem organizacional para o desenvolvimento de *software* implica em alterações nos modelos e processos de trabalho, que devem ser adaptados para cada tipo de organização.

4. PRESSUPOSTOS

4.1. Pressuposto Geral

A qualidade dos sistemas de informação desenvolvidos pelo Serpro está relacionada a avaliação das necessidades informacionais da equipe de requisitos, ao fluxo e reuso de informações no processo de desenvolvimento de *software* do Serpro. O processo de desenvolvimento do Serpro (PSDS) não trabalha aspectos relacionados ao fluxo de informações e a disponibilidade de produtos e serviços de informação. Assim, a identificação dos elementos chaves que afetam os processos relacionados a gestão da informação pode melhorar a qualidade dos produtos de software desenvolvidos pelo Serpro.

4.2. Pressupostos Específicos

Considerando que a evolução da qualidade em produtos de *software* está relacionada ao fluxo de informações e ao aproveitamento de experiências anteriores, e que a adoção de um modelo de gestão da informação pode facilitar o reuso de informações e melhorar os resultados no desenvolvimento de *software*, os pressupostos específicos são os seguintes.

1º Pressuposto Específico

Um dos aspectos importantes do levantamento de requisitos é conhecer o cliente e o domínio de negócios para o qual o sistema será desenvolvido. A obtenção de informações sobre este domínio e o cliente facilitará a evolução da qualidade dos requisitos; está relacionada a eficiência dos processos de identificação da necessidade de informações da equipe de requisitos, e a aquisição de informações dos usuários.

2º Pressuposto Específico

O aproveitamento de experiências de projetos anteriores depende da identificação e descrição do contexto em que cada projeto foi realizado, já

que ele afeta diversos aspectos do desenvolvimento do sistema. A correta identificação dos requisitos não-funcionais, e sua inclusão em processos de organização, distribuição e uso da informação facilita a identificação e replicação de contextos em projetos de *software*.

3° Pressuposto Específico

O reuso de informações pode melhorar a qualidade dos produtos desenvolvidos pelo Serpro. Entretanto, esta prática depende da disponibilidade de informações para reuso, com o registro de práticas e experiências dos projetos concluídos; e a criação de condições adequadas para a prática de reuso.

4.3. Variáveis

4.3.1. Variáveis do 1° Pressuposto Específico

Fator I - identificação das informações necessárias para equipes de requisitos.

Variáveis:

- Grau de disponibilidade de informações sobre o cliente e o domínio do sistema a ser desenvolvido;
- Disponibilidade de mecanismos de compartilhamento das informações sobre o sistema a ser desenvolvido;

4.3.2. Variáveis do 2° Pressuposto Específico

Fator I – Identificação dos requisitos não-funcionais

Variáveis:

- Existência de sistemática para identificação de requisitos não-funcionais;
- Identificação de requisitos não-funcionais para sistemas de um mesmo cliente.

Fator II – disponibilidade e uso de informações de requisitos não-funcionais nos produtos/serviços de informação.

Variáveis

- Grau de disponibilidade de requisitos não-funcionais nos produtos/serviços usados pela equipe de requisitos;
- Grau de utilização das informações de requisitos não-funcionais.

4.3.3. Variáveis do 3° Pressuposto Específico

Fator I – Disponibilidade de informações para reuso nas atividades de requisitos.

Variáveis:

- Grau de registro de informações sobre projetos já concluídos;
- Grau de utilização de informações de projetos já concluídos.

Fator II – Condições de trabalho para o reuso de informações nas atividades de requisitos.

Variável:

- Disponibilidade de ferramenta e ambiente de trabalho que incentive ao reuso de informações nos projetos *software*.

5. METODOLOGIA

A metodologia a ser usada para a pesquisa é dividida em duas etapas. Na primeira é apresentada discussão teórica sobre modelos de gestão da informação, com a comparação de três modelos já utilizados na área de engenharia de *software*. A segunda etapa será uma pesquisa de campo com profissionais da equipe de requisitos do Serpro em Brasília, para buscar a identificação de elementos chave para utilização de processos de gestão da informação em conjunto com o processo do Serpro de desenvolvimento de soluções. A pesquisa de campo será baseada em coleta de dados com questionário aplicado aos analistas de requisitos do Serpro em Brasília.

5.1. Delimitação do Estudo

A pesquisa avaliou o processo de desenvolvimento de soluções do Serpro (PSDS), em sua atividade de levantamento de requisitos, nos aspectos relacionados aos fluxos de informações, e produtos e serviços informacionais utilizados pelas equipes de requisitos dos projetos.

Apesar da formalização de papéis a serem cumpridos pelas equipes, artefatos e fluxos de documentos, o desconhecimento pelas pessoas dos conceitos relacionados a gestão da informação dificulta os processos de aquisição e uso da informação, em especial o reuso de informações obtidas em projetos anteriores.

A avaliação foi realizada com base em pesquisa teórica sobre modelos de gestão da informação, desenvolvimento e engenharia de *software*, e a utilização de métodos e ferramentas de gestão da informação e conhecimento para o desenvolvimento de sistemas de informação.

Foi verificado que há dificuldade na replicação de experiências e coleta de indicadores em projetos de *software*, principalmente em função de mudanças de tecnologia e dependência de contexto do domínio de negócio do sistema a ser desenvolvido.

Por outro lado, as experiências da NASA e Ericsson relataram graus de sucesso, onde foi possível aliar o uso de modelos de gestão da informação à

desenvolvimento de sistemas, com resultados positivos para os projetos de *software*.

Estas experiências foram relevantes para a avaliação do caso do Serpro e identificação de elementos chave que permitam a elaboração de proposta para a adoção de ciclo de gestão da informação para o processo de desenvolvimento de soluções do Serpro, na macroatividade de requisitos.

5.2. Embasamento Teórico da Metodologia

O propósito da pesquisa foi a avaliação da atividade de levantamento de requisitos do processo de desenvolvimento de soluções do Serpro, do ponto de vista dos fluxos de informações necessários a evolução da qualidade dos sistemas de informação desenvolvidos.

A partir dos anos 90 houve um grande esforço na evolução das ferramentas de tecnologia utilizadas no desenvolvimento de sistemas, bem como a adoção de conceitos da engenharia e de modelos de maturidade de processos. Entretanto, estes esforços estão concentrados nas áreas de tecnologia e processos, e apesar de alguma evolução, as organizações ainda reportam um elevado número de problemas nos produtos de *software* desenvolvidos por prestadores de serviços como o Serpro.

Pressman (2002, p.37) considera que a saída do processo de *software* é a informação, que pode ser dividida em três categorias: programas de computador, documentos que descrevem programas de computador e dados. Esta informação é armazenada em agregados de informação, segundo o conceito de Barreto (1996), que afirma ainda que os agregados têm duas funções básicas. A primeira delas é a produção da informação, por meio da qual os estoques são formados por unidades estáticas de informação, armazenadas em acervos ou bases de dados. Ele destaca que os estoques estáticos de informação são indispensáveis ao processo de geração de conhecimento, mas não efetivam este processo por si só.

A outra função dos agregados é a transferência da informação, onde ocorre a migração, assimilação da informação e a produção ou criação de conhecimento, que poderá alterar seu contexto e percepção.

Assim, é oportuno destacar que são citadas como atividades de gestão da engenharia de *software* – projeto, riscos, requisitos e configuração – e não aparece a gestão dos agregados de informação. Sua função de transferência da informação será fundamental para geração de conhecimento e evolução da qualidade dos requisitos.

Autores, como Basili (1992), destacam que o desenvolvimento de sistemas é uma atividade essencialmente experimental, e que há uma dificuldade natural na replicação de experiências em razão do contexto de cada projeto – tecnologias, domínio de negócio experiência dos clientes e desenvolvedores, dentre outros.

Apesar disto, os esforços realizados para inclusão de modelos de gestão da informação em processos de desenvolvimento de *software* – chamados por vezes de gestão de conteúdos ou gestão do conhecimento – mostram que é possível obter resultados, em especial com o aproveitamento de experiências de projetos

anteriores, e o reuso de artefatos – programas, requisitos ou documentos sobre programas.

Estes resultados consideram a utilização do conceito de domínio do sistema, e de que eles podem ser descritos, pelo menos parcialmente, pelos requisitos não-funcionais a ele associados. Por outro lado, a atividade de engenharia de requisitos, como demonstrado por Nuseibeh e Easterbrooks (2000), é multidisciplinar como a Ciência da Informação, e sua integração deve ser utilizada para melhorar os resultados no levantamento e gestão dos requisitos.

5.3. Caracterização do Universo

A pesquisa teórica tem seu escopo em propostas de modelos para gestão da informação – em especial os modelos de Davenport e Choo – e trabalhos realizados em organizações de desenvolvimento de *software*. O modelo de reuso de informações em desenvolvimento de *software* foi idealizado na Universidade de Maryland, e implementado de forma diversa em três instituições, associando conceitos de engenharia de *software* com gestão da informação e conhecimento em processos de desenvolvimento de sistemas.

O estudo de caso, com a identificação dos elementos básicos para a construção de um modelo de gestão da informação, será realizado no Serpro, empresa pública do governo federal do Brasil. A atividade de desenvolvimento de *software* é realizada em 20 equipes, com cerca de 2.000 empregados distribuídos em 10 Unidades Regionais da empresa.

A atividade de levantamento de requisitos é realizada por cerca de 200 empregados, também distribuídos em 10 Unidades. Estes empregados receberam treinamento básico – grade de treinamento do PSDS – e utilizam o mesmo processo e as mesmas ferramentas de desenvolvimento de *software*.

A escolha da atividade de requisitos ocorreu por sua importância para o processo de desenvolvimento. As informações que ela gera serão a base para todas as demais atividades do processo de desenvolvimento de sistemas e, os erros gerados nesta fase terão custo mais alto de correção, razão pela qual há uma forte preocupação com a qualidade e adequação dos requisitos definidos.

5.4. Caracterização da Amostra

O Serpro possui cerca de 2.000 empregados que atuam em desenvolvimento de sistemas de informação, distribuídos em dez Unidades diferentes. Destes, cerca de 200 trabalham em alguma atividade relacionada ao levantamento e gestão de requisitos. A equipe de Brasília possui 120 empregados, 22 dos quais atuam como analistas de requisitos, isto é, executam atividades relacionadas à engenharia de requisitos no processo de desenvolvimento de *software*. Estas pessoas têm cargo de analista, e são chamados analistas de requisitos. A formação mínima exigida é graduação, além de treinamentos específicos para atuar na atividade de requisitos.

A pesquisa foi realizada na Unidade do Serpro em Brasília, na modalidade de censo, com os 22 empregados da equipe que atuam na atividade de requisitos. A equipe de requisitos de Brasília é a que atua em mais clientes diferentes, já que a maior parte dos clientes é baseada em Brasília. Em função disto eles possuem a visão mais diversificada, em relação a clientes distintos, com necessidades diversas em termos de funcionalidades e complexidade dos sistemas.

5.5. Delineamento da Pesquisa

A parte inicial da pesquisa consiste na pesquisa teórica, que trata da ciência da informação, modelos de gestão da informação, bem como trabalhos já realizados utilizando modelos de gestão da informação e conhecimento e sua combinação com engenharia de *software*.

Três trabalhos foram identificados como adequados a esta pesquisa, e descritos de forma mais detalhada na revisão de literatura: o modelo da NASA, conforme descrito por Basili (2002, p. 5); o esquema usado na Ericsson Europa, e descrito por Johansson, Hall e Coquard (1999, p. 171); e finalmente experiência da DaimlerChrysler, e descrito por Schneider, Hunnlus e Basili (2002, p. 46).

Os três trabalhos têm como base a proposta usada na NASA, mas as diferenças na adaptação para as outras duas organizações são relevantes para este estudo de caso, razão pela qual também serão descritas. O estudo de campo foi realizado com a seleção de analistas das equipes de desenvolvimento do Serpro lotados em Brasília, e envolvidos em atividades de levantamento e gestão de requisitos. A técnica usada foi de aplicação de questionário semi-estruturado.

5.6. Coleta de Dados

5.6.1. Técnica Selecionada

A técnica selecionada para a coleta de dados foi a de aplicação de questionário semi-estruturado a um grupo de 22 pessoas da equipe de desenvolvimento de sistemas do Serpro em Brasília, que atuam na macroatividade de requisitos.

O questionário foi pré-testado com duas pessoas da equipe de requisitos, para avaliar a compreensão dos termos usados e possíveis inconsistências entre as informações solicitadas. A coleta preliminar possibilitou alguns ajustes no instrumento, mas sua estrutura básica foi preservada. Por esta razão os dois participantes que participaram do teste de avaliação do questionário foram considerados na amostra.

O questionário foi aplicado de forma conjunta, ocasião em que 13 pessoas o responderam. As nove pessoas restantes foram contactadas por email, uma vez que não puderam comparecer a reunião

5.6.2. Instrumento de Coleta de Dados

O questionário respondido pela equipe de requisitos foi estruturado em cinco blocos e 31 questões. Um bloco de questões foi reservado para cada pressuposto, com foco nos aspectos de identificação e compartilhamento de informações sobre clientes e requisitos não-funcionais; e a abordagem que o processo oferece ao reuso de informações.

Um bloco adicional foi utilizado para avaliar a percepção que as pessoas têm da atenção que o Serpro dedica ao tema “reuso de informações para desenvolvimento de *software*”, e que pode subsidiar estratégias para implantação do modelo proposto.

O último bloco coletou informações demográficas sobre os analistas de requisitos, como formação acadêmica e tempo de experiência com a atividade.

Para a elaboração do questionário a principal fonte de consulta foi o instrumento de coleta de dados de Araújo Júnior (1998, p. 167).

6. AMBIENTE DA PESQUISA – O PSDS

O objetivo deste capítulo é descrever o Processo Serpro de Desenvolvimento de Soluções (PSDS), e em especial a macroatividade de requisitos, que será tratada na pesquisa. A versão avaliada é a 6.3, disponível em fevereiro de 2008.

O PSDS⁹ foi criado com objetivo de documentar e padronizar as atividades de desenvolvimento de soluções de *software* do Serpro para seus clientes. Ele utiliza alguns conceitos para facilitar seu entendimento por todos na empresa, e alguns deles são relevantes para este trabalho, descritos a seguir:

- Processo – conjunto parcialmente ordenado de passos com objetivo de alcançar uma meta;
- Atividade – é uma unidade de trabalho que um indivíduo deve executar quando atua em desenvolvimento de *software*. A atividade possui um propósito bastante claro, geralmente expresso em criar ou atualizar um artefato;
- Artefato – produtos de trabalho gerados no desenvolvimento de *software*, como modelos, relatórios, programas e planos. As atividades utilizam artefatos como insumo e produzem artefatos como resultado de sua execução;

⁹ As referências ao Processo Serpro de Desenvolvimento de Soluções (PSDS) estão disponíveis no site do processo (<http://psds.portalcorporativo.serpro>). Ele é acessado apenas na Intranet do Serpro.

- Papel – um papel define a responsabilidade de um indivíduo, ou equipe, trabalhando no desenvolvimento de *software*. Um único papel é responsável por cada artefato, ainda que ele possa ser utilizado por muitas pessoas.

O PSDS define os seguintes aspectos do desenvolvimento de sistemas:

- Como deve ser feito – o processo descreve as atividades e subatividades a serem executadas no desenvolvimento de *software*;
- O que deve ser feito – o processo descreve os artefatos gerados pela execução das atividades;
- Quando deve ser feito – o processo define fluxos de atividades, caminhos que mostram a ordem para a execução das atividades e sub-atividades;
- Por quem deve ser feito – o processo define papéis, responsabilidades atribuídas a uma ou mais pessoas durante o desenvolvimento. Uma pessoa pode executar diversos papéis em um mesmo projeto;

É possível definir ainda ferramentas, instrumentos que possibilitam a execução do processo.

As atividades do PSDS foram agrupadas em macroatividades, de acordo com o modelo CMMI, em seus níveis 2 e 3. A implantação das macroatividades e a maturidade de sua prática garantem a aderência aos níveis do CMMI, conforme abaixo.

As macroatividades do Nível 2 são: Requisitos, Gestão de Projeto de *Software*, Gestão de Configuração de *Software*, Medição e Análise, Garantia da Qualidade de *Software* e Gestão de Aquisição com Fornecedor (opcional).

Já para o Nível 3 de maturidade, o PSDS define todas as macroatividades do Nível 2, e mais as seguintes: Análise e Projeto, Implementação, Testes, Homologação, Implantação, Revisão por Pares, Gestão do Processo da Organização, Programa de Treinamento e Análise da Decisão e Resolução.

As macroatividades são agrupadas de acordo com sua função no processo de desenvolvimento.

- Gestão de Processo: Gestão do Processo da Organização e Programa de Treinamento;
- Gestão de Projeto: Gestão do Projeto de *Software* e Gestão de Aquisição com Fornecedor;
- Engenharia de *Software*: Requisitos, Análise e Projeto, Implementação, Testes, Homologação, Implantação e Revisão por Pares;
- Suporte: Gestão da Configuração de *Software*, Garantia da Qualidade de *Software*, Medição e Análise e Análise de Decisão e Resolução.

Este trabalho terá foco no Nível 2 do CMMI, e na macroatividade de requisitos. Será feita uma descrição desta macroatividade, para melhor entendimento da situação atual e da proposta a ser apresentada.

Segundo a versão 6.3 do PSDS, a macroatividade de requisitos tem como objetivo principal nortear o desenvolvimento de uma especificação de requisitos que atenda às necessidades do cliente, e permitir o entendimento do que se deseja construir antes do início do projeto.

O principal responsável pelas atividades de requisitos é o analista de requisitos, que irá interagir com os clientes e demais papéis do PSDS para garantir a continuidade do processo de desenvolvimento de *software*.

Os seguintes objetivos específicos estão descritos no manual do PSDS:

- Coletar, analisar e documentar os requisitos do cliente, derivando conjuntos mais refinados de requisitos para suporte ao desenvolvimento do produto;
- Gerenciar a evolução dos requisitos, garantindo que ações corretivas sejam tomadas em resposta a eventos que afetem as especificações já existentes;
- Manter entendimento comum entre as partes interessadas quanto a satisfação de suas necessidades originais por meio da especificação de requisitos gerada.

A macroatividade é composta de seis atividades, e divididas em subatividades, que detalham as ações que devem ser executadas para a obtenção dos resultados esperados. As subatividades não serão detalhadas, pois não são necessárias ao desenvolvimento deste trabalho. As atividades são mostradas na Figura 9, e descritas a seguir.

- Elicitar Requisitos

A elicitação é a primeira atividade no ciclo de vida de engenharia de requisitos, e seu propósito é capturar conhecimento relevante sobre o problema a ser resolvido e sobre o domínio do sistema. Ela acontece via comunicação, usando as técnicas de elicitação disponíveis, com os clientes e demais partes interessadas no sistema a ser desenvolvido. É composta de três subatividades.

- Documentar e Analisar Requisitos

Os requisitos identificados na atividade anterior devem ser documentados, para criar uma linguagem comum entre as partes interessadas e facilitar o processo de análise, que examina os requisitos na busca de inconsistências. Após a análise, em geral feita por pessoas com mais experiência, os requisitos devem ser documentados em nível apropriado de detalhe. Esta atividade é composta por quatro subatividades e gera o artefato Modelo de Requisitos.

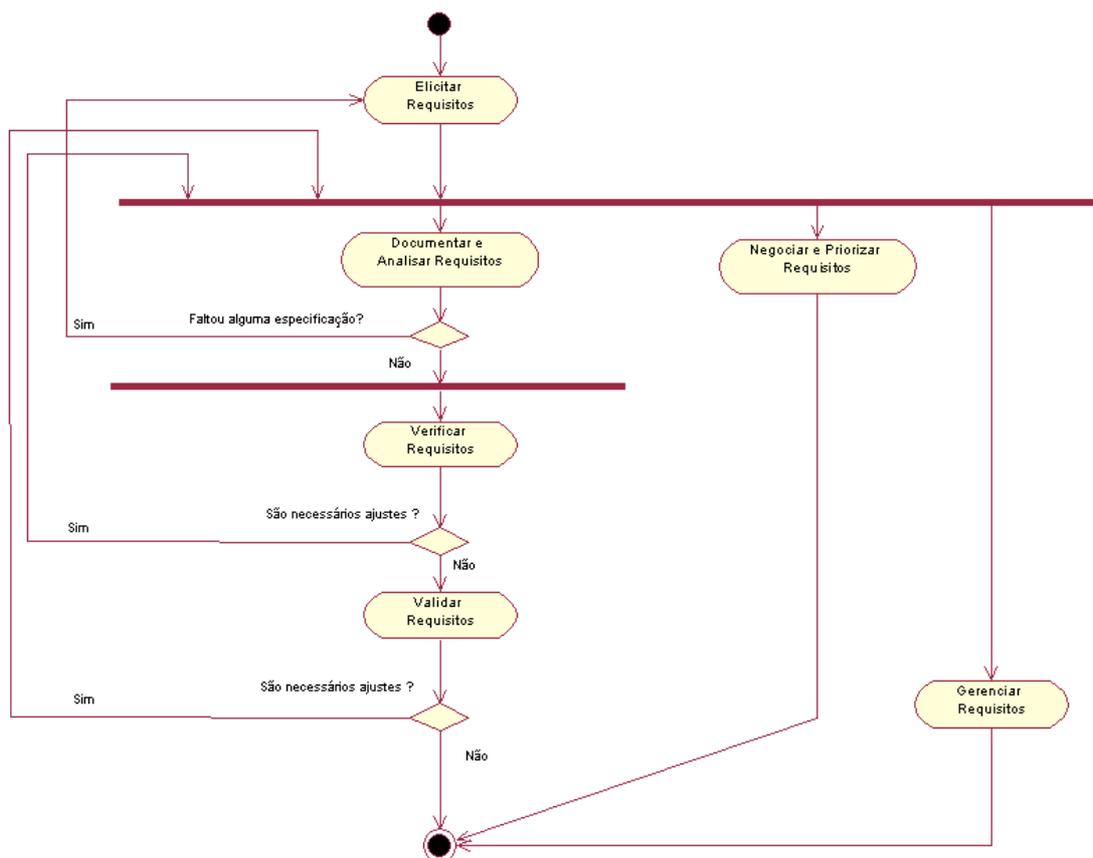


Figura 9 - Visão geral - fluxo da macroatividade de requisitos do PSDS

Fonte: PSDS – sítio de descrição do processo – Intranet do Serpro

- **Verificar Requisitos**

A verificação de requisitos engloba tarefas para analisar o modelo de requisitos consolidado, já que pode haver mais de um analista trabalhando na coleta de requisitos, para uma nova verificação de inconsistências ou defeitos na especificação, e para certificar que o conteúdo do modelo reflita todos os requisitos levantados junto ao cliente. Ela é composta de três subatividades, e gera o artefato Modelo de Requisitos, em sua versão completa, revisada e verificada. Nesta atividade devem ser identificados artefatos que sejam candidatos a reuso em outros sistemas.

- **Negociar e Priorizar Requisitos**

Esta atividade tem como objetivo revisar e negociar os requisitos com todas as partes interessadas no projeto, para resolver conflitos surgidos no processo de análise e verificação dos requisitos dos clientes, ou impactos gerados por mudanças solicitadas pelo cliente no modelo de requisitos. Após a resolução dos conflitos identificados, os requisitos são priorizados, e definidos os marcos de acompanhamento, definidos com o planejamento global do projeto. É composta por duas subatividades, e

gera o artefato ata de reunião, com a versão final dos compromissos assumidos entre o cliente e a equipe.

- Validar Requisitos

A validação busca garantir que os requisitos definidos, documentados, verificados e priorizados atendam às necessidades do cliente, e possam ser efetivamente utilizadas nas demais etapas do processo. A validação, em geral, inclui a formalização do compromisso, via um termo de aceite assinado entre as partes. Ela é formada por três subatividades, e tem como principal resultado o artefato com o modelo de requisitos e o termo de aceite do cliente.

- Gerenciar Requisitos

A atividade de gestão é realizada ao longo de todo o processo de desenvolvimento de *software*, e atualiza constantemente os requisitos, avaliando o impacto de alterações e definindo eventuais necessidades de renegociação dos compromissos assumidos. É composta de três subatividades.

Para o desenvolvimento destas atividades, duas tarefas podem ser destacadas por sua importância para o processo. A primeira delas é a escolha da técnica de elicitação dos requisitos. O PSDS propõe o uso das seguintes técnicas de elicitação de requisitos:

- Técnicas tradicionais – incluem o uso de questionários, entrevistas e análise de documentação existente, como manuais, leis e normas. Questionários são mais úteis quando há uma grande quantidade de partes interessadas, e as entrevistas quando as partes interessadas possuem conhecimentos subjetivos, e estão dispostas a passar pela entrevista;
- Técnicas de elicitação em grupo – englobam técnicas de dinâmica de grupo, para entender de forma detalhada as necessidades. Elas incluem sessões tipo *brainstorming*, quando grupos são reunidos em ambiente e estimulados a trocar idéias; e sessões de JAD (*Joint Application Design*), reuniões mediadas entre usuários e os analistas para a explicitação dos requisitos;
- Prototipação – é a implementação parcial de um sistema, de forma rápida, para obter apreciação dos usuários quanto aos requisitos, após o que o protótipo é descartado. Pode ser útil quando existe um grau elevado de incerteza, ou quando é necessária uma resposta rápida dos usuários;
- Técnicas cognitivas – agrupamento de técnicas originalmente desenvolvidas para aquisição de conhecimento em sistemas baseados em conhecimento. São menos usadas, e dentre elas podemos destacar a análise de protocolo, *laddering*, *card sorting* e *repertory grids*;
- Técnicas contextuais – uso de técnicas de etnografia e análise social, usadas como alternativas para técnicas tradicionais e cognitivas. Consistem na observação de potenciais usuários em seu ambiente de trabalho. São úteis para funções pouco automatizadas, e exigem observadores treinados, sem noções preconcebidas do problema.

A segunda tarefa a ser destacada é a classificação dos requisitos, para a qual o PSDS define um artefato chamado “Guia de Classificação de Requisitos”. Ele descreve cerca de 12 tipos de requisitos, e dois deles serão destacados para este trabalho: os requisitos funcionais e os não-funcionais.

Os requisitos funcionais definem as funções e tarefas que o sistema irá executar, e podem ser descritos como funcionalidades e regras de negócio. Os requisitos não-funcionais referem-se a características importantes do sistema, como segurança, desempenho, tipo de ambiente e tipo de público que acessará o sistema.

Um ponto importante de processos de desenvolvimento de *software*, e também do PSDS, é a determinação de registros das decisões e evolução do processo em artefatos específicos, que devem ser versionados (mantidos em diversas versões, para manter um registro da evolução do sistema) e armazenados para posterior revisão ou utilização.

A preocupação com o registro de decisões e a codificação de conhecimentos sobre os sistemas, além de atender a premissa de geração de evidências dos modelos de maturidade, pretende tratar dois outros aspectos relevantes para o negócio do Serpro. O primeiro é a longa duração do relacionamento com os clientes, já que a maior parte dos clientes tem negócios com a empresa há mais de 20 anos, e boa parte dos sistemas desenvolvidos terá pedidos de manutenção por este prazo.

Neste cenário é importante manter registro do conhecimento sobre os domínios de negócio, e das decisões tomadas quando do desenvolvimento e implementação dos sistemas, pois elas podem ser importantes no relacionamento comercial e operacional no médio e longo prazos.

O segundo aspecto relacionado ao registro de informações é a rotatividade de pessoas nas equipes dos clientes e do Serpro. Em boa parte dos órgãos da administração pública federal há políticas de rodízio de pessoas entre as funções e não há garantia de continuidade de pessoas que especificaram os sistemas. No Serpro, além da rotatividade de pessoal técnico, cerca de 4.000 empregados estarão aptos a se aposentar nos próximos dez anos, e o registro e manutenção do conhecimento sobre o negócio dos clientes é fundamental para a continuidade do negócio.

A implementação¹⁰ do PSDS no Serpro foi realizada com o auxílio das ferramentas da *Rational Software*, hoje uma divisão da IBM. Um dos principais produtos usados é o *RequisitePro*, que auxilia o processo de coleta e gestão dos requisitos. Assim, todos os artefatos de um determinado projeto são armazenados em uma base, onde podem ser consultados pela equipe do projeto. A ferramenta controla as versões e alterações dos artefatos, que podem assim ser utilizados por mais de uma pessoa simultaneamente, sem o risco de perda das alterações feitas por cada pessoa.

¹⁰ Em 2007 foi iniciado projeto para substituição das ferramentas IBM por ferramentas de código aberto. Este processo ainda está em andamento, e as equipes utilizam hoje duas ferramentas para versionamento de artefatos.

Por características de implementação, o *RequisitePro* foi distribuído em diversas instalações, e o acesso as bases foi separado por projeto ou área funcional. O acesso dos analistas de requisitos é restrito aos projetos em que estão cadastrados, ou se sua área funcional. A implementação atual não permite a pesquisa de artefatos em todas as bases simultaneamente, o que é considerado uma limitação pelas equipes para a busca de informações sobre outros projetos.

O processo prevê ainda o uso de um artefato chamado base de soluções (BS), onde soluções, decisões e aspectos considerados relevantes para compartilhamento futuro devem ser registrados pelos analistas ou líderes de projeto, e armazenados para posterior reuso em outros projetos.

7. ANÁLISE DOS DADOS E COMPROVAÇÃO DOS PRESSUPOSTOS

Este capítulo trata da análise dos dados coletados com os questionários aplicados. A análise será feita inicialmente de forma descritiva, com base nos blocos de questões definidos para o instrumento de coleta de dados. Em seguida é feita uma análise complementar, confrontando os resultados com os processos do ciclo de gestão da informação.

7.1. Análise Descritiva dos Dados

A análise descritiva será feita com bases nos pressupostos estabelecidos, e nos blocos de questões associados a cada um deles. As questões foram organizadas de forma a manter um nível de correlação com as atividades do PSDS e facilitar o entendimento pela equipe de requisitos.

7.1.1. Análise das Informações Demográficas

As informações sobre os analistas de requisitos que responderam ao questionário da pesquisa – Questões 28 a 31 – teve como objetivo definir o grupo de pessoas, em relação aos aspectos de qualificação profissional, posição na organização e tempo de experiência.

Cabe destacar que o grupo, formado por 22 pessoas, é bastante homogêneo, com 11 homens e 11 mulheres e todos ocupam o cargo de analista¹¹ de requisitos, têm formação acadêmica similar e passaram por treinamentos do PSDS.

7.1.1.1. Formação Acadêmica

A formação acadêmica dos participantes é avaliada nas Questões 28 e 29, conforme dados da Figura 10.

¹¹ A empresa possui ainda os cargos de Auxiliar e Técnico, que executam funções de menor complexidade.

Todos os empregados são graduados, sendo que 21 deles em cursos de Ciência da Computação¹², e apenas um concluiu o curso de Ciências Contábeis. Esta pessoa é que possui maior experiência na área, o que é considerado natural, já que nesta época (cerca de 15 anos atrás) os cursos de ciência de computação ficaram mais populares, e concentraram profissionais que de alguma forma trabalhavam com TI.

Foi ainda identificado que quatro pessoas concluíram cursos de Pós-Graduação, enquanto três outras já concluíram Mestrado. Todos estes cursos foram em áreas relacionadas a desenvolvimentos de sistemas, como Engenharia de Software e Sistemas de Informação.

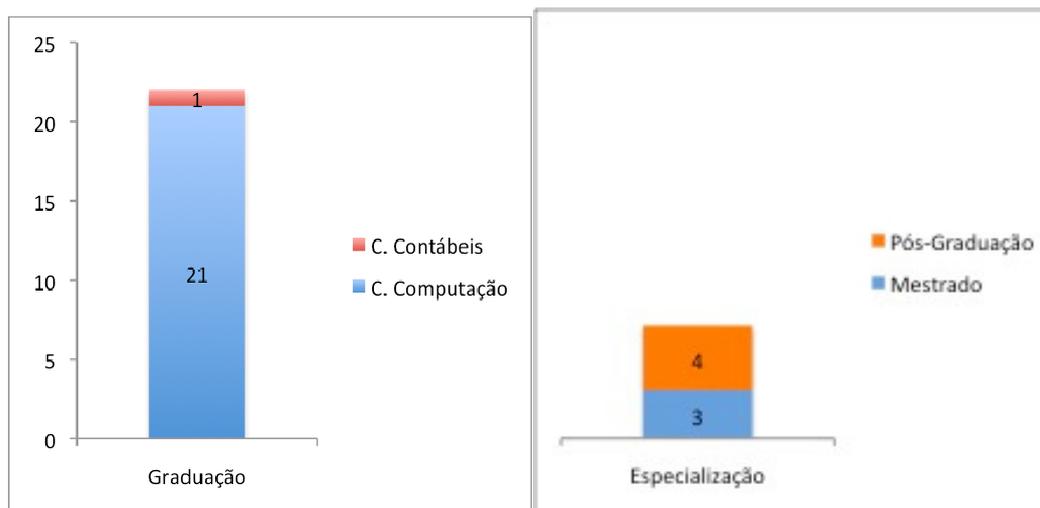


Gráfico 1 – Dados de Formação Acadêmica

7.1.1.2. Cargo e Experiência

A Questão 30 identifica os empregados que ocupam algum cargo gerencial. Das pessoas que responderam ao questionário, apenas três empregados, ocupam cargo gerencial. Os demais 19 são empregados alocados na equipe de requisitos.

Já a Questão 30 considera o tempo de experiência dos respondentes com atividades de requisitos. Em função das repostas obtidas foram consideradas três categorias de experiência atuando em requisitos:

- Experiência igual ou menor que quatro anos – 12 pessoas;
- Experiência maior que quatro e menor que oito anos – sete pessoas;
- Experiência igual ou maior que oito anos – três pessoas.

Cabe destacar que o Serpro retomou a realização de concursos para contratação de pessoal nos últimos 10 anos, e uma das áreas que mais contratou foi a de desenvolvimento. Os dados mostram que 86,4% das pessoas têm tempo de experiência inferior a oito anos, o que significa que

¹² Três pessoas concluíram cursos denominados Sistemas de Informação, mas por simplificação e similaridade de currículo, todos foram tratados como Ciência da Computação.

sua formação já considerou os conceitos de processo e engenharia de software.

7.1.2. Informações do Cliente e Domínio da Aplicação – Bloco I

O Bloco I agrupa as questões de 1 a 8, que buscam mostrar de que forma são identificadas as necessidades de informação sobre os clientes e domínio do sistema a ser desenvolvido; e como estas informações são coletadas e disseminadas na equipe.

Elas estão associadas ao Fator I do primeiro pressuposto específico – identificação das informações necessárias para as equipes de requisitos

7.1.2.1. Informações sobre clientes

As Questões 1 e 2 referem-se a identificação de informações sobre os clientes. A Questão 1 verifica a percepção da equipe em relação a necessidade de dispor de informações sobre o cliente e seu domínio de negócio/sistemas, antes do de iniciar o levantamento de requisitos. Todos os 22 empregados que responderam ao questionário afirmaram que é importante dispor destas informações antes do início do projeto.

A Questão 2 avalia se há identificação prévia das informações que devem ser levantadas sobre o cliente e os sistemas. Os itens de frequência utilizados foram: 1.sempre – mostra a identificação sistemática de informações sobre o cliente; 2.algumas vezes – indica que as informações são identificadas para alguns projetos; 3.nunca – a equipe não identifica as informações. As respostas mostram que sete pessoas (31,8%) sempre se preocupam em identificar as informações antes do início do projeto. Apenas 1 pessoa nunca se preocupa com este assunto, e 14 pessoas (63,6%) identificam algumas vezes as informações antes do projeto. Assim, é possível concluir que, para cerca de 68% dos entrevistados, a identificação de informações antes do início dos projetos não é sistematizada.

7.1.2.2. Disponibilidade e registro de informações de clientes

As Questões 3 e 4 referem-se a identificação e registro de informações de clientes dos projetos de software. A Questão 3 verifica se as equipes dispõem de informações sobre os clientes antes do início do projeto. Os itens disponíveis são: 1. Sim – as informações estão disponíveis; 2. Não – as informações não estão disponíveis. Os dados mostram que, dos 22 analistas que participaram da pesquisa, 19 (86,4%) afirmaram que algum tipo de informação sobre os clientes e domínio da aplicação está disponível.

Por sua vez, a Questão 4 identifica onde as informações sobre os clientes podem ser encontradas. Foram listados como opções sete itens de frequência, e poderiam ser marcados todos os adequados. Cabe destacar os dois itens de maior frequência nas respostas – 19 pessoas (86,4%) para a opção “Site do cliente”- que é uma fonte externa ao Serpro; e 15 respostas (68,2%), para a opção “Conversa com colega”. Estas duas fontes não estão consideradas pelo PSDS. Dois outros itens possuem frequências próximas a metade dos entrevistados – “Relatório de projeto anterior”, com nove respostas (40,9%); e “Requisitos de projeto anterior”, com 11 respostas.

Estes dados mostram, de forma indireta, que metade dos analistas de requisitos reconhece que os artefatos de projetos anteriores podem ser aproveitados como fonte de informação.

7.1.2.3. Aquisição e armazenamento de informações de clientes

As Questões 5 e 6 avaliam a forma de aquisição e armazenamento de informações dos clientes. A Questão 5 registra as formas de aquisição de informações utilizadas junto aos clientes. Quatro itens de frequência estavam disponíveis para resposta: 1.Entrevistas – item apontado por 17 analistas (77,3%); 2.Análise de documentos do cliente (inclui legislação, relatório e normas) – resposta de 16 pessoas (72,7%); 3.Reuniões – item assinalado pelos 22 analistas (100%) e 4.Outros – com campo para descrição, que teve apenas três respostas, que citaram itens das outras opções, razão pela qual não foram listados.

A Questão 6 verifica onde as informações sobre os sistemas obtidas com os clientes são armazenadas. Três opções foram listadas: 1.Artefatos do PSDS – apontada por 18 analistas (81,8%); 2. Relatórios do projeto – escolhida por 3 pessoas (13,6%); e 3.Não são armazenadas – opção que teve apenas uma resposta, 4,6% do total. Esta questão mostra, conforme indicado no Gráfico 2, que o PSDS conseguiu implantar procedimentos homogêneos, pelo menos para algumas atividades dos projetos. As informações deveriam mesmo estar em artefatos do projeto, embora a ferramenta de armazenamento de artefato tenha diversas limitações, sendo enxergada como um simples repositório de documentos.



Gráfico 2 – Armazenamento de Informações dos Clientes

7.1.2.4. Compartilhamento de informações dos clientes

As Questões 7 e 8 mostram se há o compartilhamento das informações obtidas com os clientes, e como ele ocorre. Todos os entrevistados responderam que fazem o compartilhamento de informações com outras pessoas da equipe. Em relação a forma do compartilhamento, a Questão 8, representada no Gráfico 3, apresenta quatro opções para resposta: 1. Alerta

da base do PSDS – opção que não foi assinalada; 2. Email para equipe – opção escolhida por sete pessoas (31,8%); 3. Reuniões da equipe – assinalada por 15 analistas (68,2%); 4. Não há mecanismo de compartilhamento – com sete respostas (31,8%).

Além de quase um terço dos entrevistados relatar que não há mecanismo específico para compartilhamento de informações, cabe destacar que o item com maior frequência – 68,2% - são reuniões, oportunidade em que as informações são compartilhadas, mas nem sempre registradas para uso posterior. O compartilhamento por email também possui limitações, como não garantia de recebimento, retenção e acesso posterior.

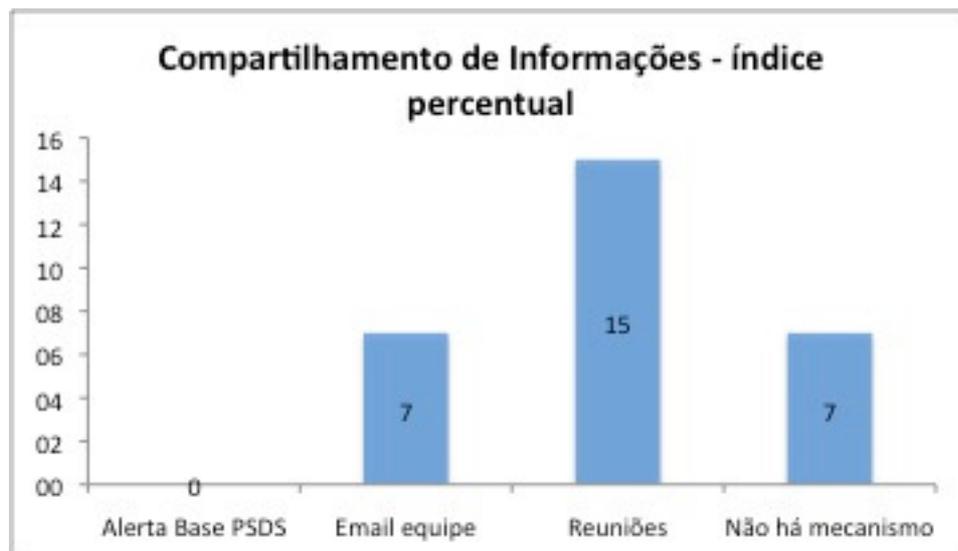


Gráfico 3 – Formas de Compartilhamento de Informações

7.1.3. Avaliação do Primeiro Pressuposto Específico

Um dos aspectos importantes para a qualidade do levantamento de requisitos é o conhecimento que a equipe possui sobre o cliente e sua área de atuação (domínio de negócio). Muitos clientes tem dificuldade em expor os requisitos do sistema de forma clara, e quanto mais informações os analistas de requisitos tiverem sobre o cliente e sua organização, mais completa será a aquisição de informações sobre o sistema.

Os dados coletados mostram que todos os entrevistados consideram importante a disponibilidade de informações, mas que não há uma sistemática para identificar quais seriam as informações necessárias, e nem quais seriam as fontes para sua obtenção. Embora os participantes da pesquisa reconheçam que algumas informações estão disponíveis, as fontes indicadas como as mais importantes para a obtenção de informações sobre os clientes são a Internet (*sites* do cliente) e conversa com os colegas de equipe, fontes nem previstas pelo PSDS.

Em relação ao compartilhamento de informações, foi declarado que as informações obtidas junto aos clientes são preferencialmente armazenadas em artefatos do PSDS, e todos os entrevistados informaram que compartilham as informações. Entretanto, cerca de um terço dos entrevistados responderam que

não há mecanismo específico de compartilhamento, e os demais apontaram como principal forma de compartilhamento reuniões da equipe e envio de email.

Os dados obtidos nas Questões 4 e 8, sobre forma de compartilhamento e aquisição de informações, confirmam pesquisa de Carvalho e Filho (2008, p.16), em que questionário respondido por 868 empregados mostrou que as principais formas de compartilhamento de conhecimento no Serpro são conversa com colegas de trabalho (91,9% de respostas) e discussão em reuniões presenciais (73,4% de respostas positivas).

Assim, o processo de desenvolvimento não identifica as informações que devem ser obtidas junto aos clientes e, embora estabeleça ações para a aquisição de informações com os clientes, ele não trata as principais fontes, e os aspectos de seu compartilhamento entre a equipe do projeto. Desta forma o fluxo das informações para as equipes de requisitos é incompleto.

7.1.4. Requisitos Não-funcionais e Produtos e Serviços – Bloco II

Este bloco é formado pelas Questões 9 a 15, e tenta identificar aspectos do processo e ações das equipes relacionadas a forma de tratamento dos requisitos não-funcionais pelas equipes de projeto; como eles são identificados, armazenados e a percepção do uso de produtos e serviços de informação.

Elas estão associadas aos fatores I e II do segundo pressuposto específico: forma de identificação dos RNFs e disponibilidade de produtos e serviços de informações associadas aos RNFs.

7.1.4.1. Identificação e registro de requisitos não-funcionais

As Questões 9 e 10 tentam determinar se os requisitos não-funcionais (RNFs) que devem ser levantados com os clientes estão identificadas no início do projeto, e de que forma são armazenadas. A questão 9 trata da possível identificação dos RNFs no início do desenvolvimento, e possui três opções possíveis: 1. A relação de RNFs sempre está disponível – com dez respostas (45,5%); 2. A relação de RNFs está disponível algumas vezes, também com dez respostas (45,5%), e; 3. A relação de RNFs nunca está disponível, que foi assinalada por apenas 2 pessoas (9,0%).

Estes números demonstram que, embora exista uma determinação de que os requisitos não-funcionais devem ser apurados com os clientes, não há um modelo sistematizado, o que pode dificultar a identificação de contexto para o projeto

Já a questão 10 pergunta, caso exista a relação de requisitos não-funcionais, onde ela é encontrada. Todas as respostas identificaram algum artefato do PSDS, sendo 16 pessoas (72,7%) em artefatos de requisitos, dois analistas (9,1%) em relatórios de projetos e quatro pessoas (18,2%) informaram que elas são armazenadas em orientações do PSDS.

Cabe destacar que o PSDS possui apenas uma orientação técnica sobre requisitos não-funcionais, que os classifica e mostra alguns exemplos. A indicação de 72,7% de respostas que apontaram artefatos de requisitos

possivelmente refere-se a alguma menção feita aos requisitos não-funcionais nestes artefatos.

7.1.4.2. Clientes e requisitos não-funcionais

As Questões 11 e 12 referem-se aos clientes e a especificação de RNFs. A Questão 11 trata a capacidade dos clientes em realizar a especificação dos requisitos não-funcionais de seus sistemas. As três possíveis opções de resposta eram: 1. Eles sempre fazem boas especificações de RNFs – esta opção não foi marcada por nenhum dos entrevistados; 2. Eles fazem boas especificações algumas vezes – opção escolhida por 19 analistas (86,4%); e 3. Eles nunca fazem boas especificações – eleita por três pessoas (13,6%).

As respostas, no Gráfico 4, confirmam a dificuldade de clientes da administração pública federal em definir contextos de funcionamento para seus sistemas, o que confirma a necessidade de investimento nesta área, em função da importância dos requisitos não-funcionais para o desenvolvimento e implantação de sistemas.

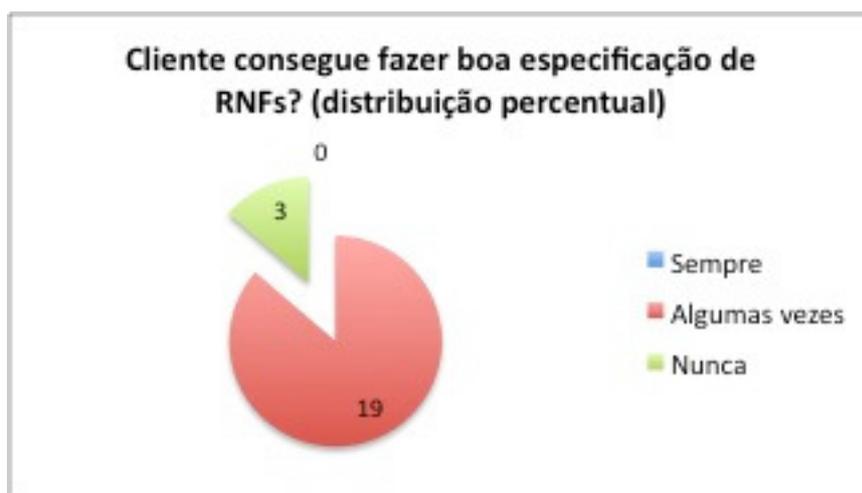


Gráfico 4 – Clientes e a Especificação de RNFs

7.1.4.3. Utilização de requisitos não-funcionais

A forma de utilização dos RNFs nos projetos é avaliada pelas Questões 12 e 13. A primeira delas identifica a possibilidade de aproveitamento de requisitos não-funcionais para sistemas de mesmos clientes. Havia três opções de resposta: 1. Os requisitos são sempre semelhantes – obteve duas respostas (9,1%); 2. Os requisitos para um mesmo cliente são semelhantes algumas vezes – opção de 18 entrevistados (81,8%); 3. Os requisitos para um mesmo cliente nunca são semelhantes – opção apontada por duas pessoas (9,1%). O resultado, mostrado no Gráfico 5, indica que há uma boa possibilidade de reuso dos requisitos não-funcionais, especialmente para um mesmo cliente.

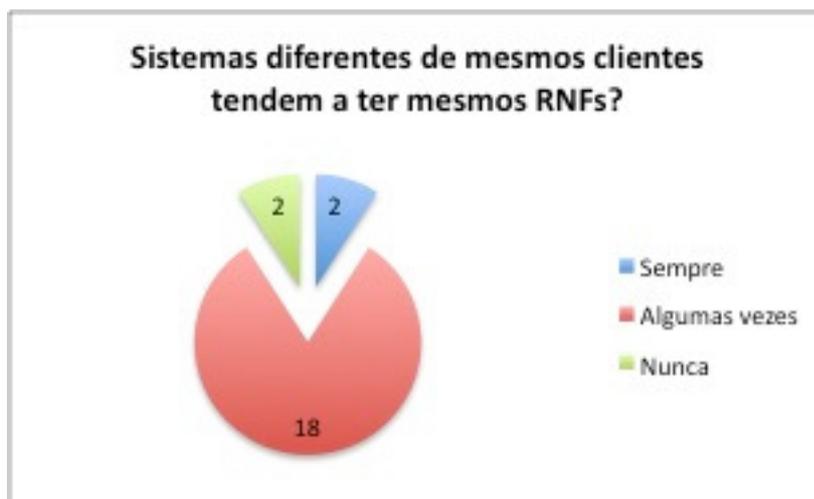


Gráfico 5 – Clientes e aproveitamento de RNFs

A Questão 13 visa determinar a intensidade em que as equipes de requisitos utilizam informações de RNFs de projetos anteriores. As opções disponíveis eram: 1. Sempre utilizo informações de RNFs de projetos anteriores, escolhida por quatro pessoas (18,2%); 2. Algumas vezes utilizo informações sobre projetos anteriores – opção escolhida por 16 entrevistados (72,7 %); e 3. Nunca utilizo informações sobre projetos anteriores, opção assinalada por dois analistas (9,1%).

7.1.4.4. Disponibilidade de produtos e serviços

A percepção da equipe de requisitos sobre a disponibilidade de produtos e serviços de informação é avaliada pela pelas Questões 14 e 15. Não foi utilizada uma definição única de produtos e serviços de informação, razão pela qual considera-se que a resposta é uma percepção dos entrevistados sobre o tema.

A Questão 14 questiona quanto a existência de produtos e serviços de informação disponíveis para execução da atividade de requisitos. A maior parte dos entrevistados, 15 pessoas (68,2%), afirma que há produtos e serviços disponíveis, enquanto os sete restantes (31,8%) afirmam que não.

A Questão 15 pede que sejam indicados os produtos/serviços identificados no item anterior. O item anterior foi respondido “Sim” por 15 pessoas, mas uma delas não identificou nenhum produto ou serviço. Assim, 14 pessoas apontaram os seguintes produtos/serviços (por número de citações):

- Artefatos PSDS (forma genérica) – 7;
- RequisitePro¹³ (ferramenta de registro de requisitos) – 6;
- Orientações do PSDS – 3;
- Grupos de especialistas¹⁴ – 2;

¹³ Ferramenta da Rational Software utilizada para armazenamento dos requisitos funcionais e não-funcionais.

- Clear Case¹⁵ – 1.

Em relação a Questão 14, é oportuno comentar quase um terço dos entrevistados entendem que não há produtos e serviços de informação disponíveis. Por outro lado, os demais entendem que os artefatos do processo tem esta função, embora com limitações da ferramenta.

7.1.5. Avaliação do Segundo Pressuposto Específico

A identificação de domínio de negócio e contexto de projetos de *software* é importante para o aproveitamento de experiências de projetos anteriores, o que facilita o processo de desenvolvimento e alcance dos resultados esperados.

Os dados obtidos nas Questões 9 e 10 mostram que o PSDS não possui sistemática para a identificação de RNFs para os domínios de negócio tratados pelo Serpro, embora uma parte da equipe identifique que algumas orientações estão disponíveis.

Este fator é relevante, até mesmo em função da percepção dos analistas de requisitos quanto a capacidade dos clientes em definir os RNFs para seus sistemas. As repostas para a Questão 11 mostram que, na grande maioria das vezes, os clientes têm dificuldade em definir os RNFs para seus sistemas. Este trabalho deve ser orientado pelos analistas de requisitos do Serpro.

As respostas à Questão 12 reforçam a percepção de que é muito provável que sistemas de mesmos clientes tenham RNFs, e contexto, semelhantes, o que facilitaria o reuso de RNFs e experiências para projetos que atendam a estes clientes. As equipes percebem a possibilidade de aproveitamento, e até a utilizam de forma não sistemática.

7.1.6. Reuso de Informações – Bloco III

Este bloco de questões, itens 16 a 22, tem como objetivo avaliar a visão da equipe de requisitos em relação a reuso de informações – previsão de uso no desenvolvimento de *software*, práticas em uso e estágio da base de soluções, local onde as experiências das equipes em desenvolvimento de *software* devem ser registradas para uso posterior.

As perguntas estão associadas ao terceiro pressuposto específico, e a seus fatores I e II: disponibilidade de informações para reuso, sua utilização e as condições existentes no ambiente para o processo de reuso.

7.1.6.1. PSDS e o registro de experiências

As Questões 16 e 17 tratam aspectos do PSDS relacionados a orientações e registros de informações e práticas durante os projetos de *software*. A

¹⁴ Grupos de trabalho definidos pelo PSDS, que oferecem suporte a algumas atividades.

¹⁵ Produto da *Rational Software* utilizado para versionamento de artefatos.

primeira delas avalia a visão geral sobre orientações do PSDS relacionadas ao registro de experiências e práticas. As respostas de 14 pessoas (63,6%) indicam que há orientação do processo para o registro de práticas. Os oito analistas restantes (36,4%) responderam que não percebem estas orientações.

A Questão 17 verifica se os analistas de requisitos registram informações relativas ao projeto de *software*, que podem ser de interesse para outros projetos. As 3 opções para resposta foram: 1. Sempre registro informações que possam interessar a outros projetos – opção escolhida por quatro pessoas (18,2%); 2. Os registros para outros projetos são feitos algumas vezes – item assinalado por 11 analistas (50,0%); 3. Nunca registro informações que possam ser aproveitadas por outros projetos – opção de sete pessoas (31,8%).

As respostas mostram que, apesar de mais de 60% dos analistas reconhecerem que o PSDS orienta para o registro de experiências, mais de 30% das pessoas nunca registra experiências, e apenas 18,2% o fazem com frequência. Os dados mostram divergência entre as orientações e o comportamento real da equipe ao longo do projeto.

7.1.6.2. Consulta a base de soluções

As Questões 18 e 19 mostram a percepção da equipe de requisitos em relação ao uso da base de soluções, e determinam o grau de consulta a base de soluções (BS) antes do início de um projeto, e eventuais razões para que ela não seja utilizada.

A Questão 18, ver Gráfico 6, é sobre o uso da BS como fonte de informações sobre cliente e o domínio da aplicação. As opções de resposta eram: 1. Sempre consulto a BS – com duas respostas (9,1%); 2. Algumas vezes consulto a BS - com seis respostas (27,3%); 3. Nunca consulto a BS no início de projetos – item que obteve 14 respostas (63,6%).

As pessoas que responderam aos itens 2 (Algumas vezes)/3 (Nunca), em número de 20, também deveriam responder a questão 19, para determinar as razões pela qual a BS não é usada como fonte de consulta. Entretanto, apenas 15 pessoas responderam a esta questão. Havia opções para marcar até cinco itens de frequência, com o seguinte resultado, em ordem decrescente de relevância:

- ferramenta não oferece facilidades de busca e recuperação – nove respostas (60,0%);
- as informações que ela oferece não são relevantes – seis (40,0%);
- projeto possui contexto diferente dos demais – quatro (26,7%);
- nova tecnologia – experiências antigas não ajudam – quatro (26,7%);
- não acho relevante para o projeto – uma pessoa (6,7%).

Os resultados são representados no Gráfico 7, e 60% das pessoas não acessam a BS porque ela não oferece facilidades para a recuperação das informações, e 40% das pessoas acha simplesmente que ela não possui informações relevantes.



Gráfico 6 – Índice de consulta à BS no início de projetos



Gráfico 7 – Razões para não consulta à BS

7.1.6.3. Registro de informações na base de soluções

As Questões 20 e 21 complementam as anteriores, e tentam identificar se as pessoas registram informações relevantes na BS e, caso não o façam, qual a razão para este comportamento.

A Questão 20, ver Gráfico 8, refere-se ao uso da BS para registro de decisões de projetos ou outras informações consideradas relevantes. As opções oferecidas foram: 1. Sempre registro informações na BS – nenhum empregado atendeu a este item; 2. Registro informações algumas vezes – com três respostas (13,6%) positivas; 3. Nunca uso a BS – item assinalado por 19 empregados (86,4%).

A questão 21 devia ser respondida apenas pelas pessoas que marcaram os itens 2/3 na questão 20, mas houve apenas 18 registros de respostas, com a distribuição representada no Gráfico 9:

- O registro não é relevante para o processo e produto – não houve respostas neste item;
- Não acho que outras pessoas vão consultar a BS – sete respostas (39%);

- Não há tempo – os prazos de entrega são muito curtos – 11 respostas (61%);
- A ferramenta não é adequada para registro e pesquisa – seis respostas (33%).

É oportuno observar que as razões apresentadas para não gerar registros de experiências nos projetos encontram-se em três categorias distintas: cultural (os outros não consultarão a base); projeto (os prazos são curtos e não permitem atividades não voltadas a produção do produto) e técnica (a ferramenta não é adequada para registro e pesquisa dos dados).



Gráfico 8 – Índice de registro de experiências na BS



Gráfico 9 – Razões para não registro de experiências

7.1.6.4. Reunião de avaliação dos projetos

As práticas de engenharia de *software* e gestão de projeto recomendam a realização de reunião de finalização dos projetos de *software*. Embora esta atividade não faça parte da macroatividade de requisitos, sua inclusão na pesquisa foi considerada pertinente, por tratar-se de oportunidade própria para registro de práticas e experiências consideradas relevantes ao longo do projeto. As Questões 22 e 23 avaliam se as reuniões foram realizadas e, caso positivo, que tipo de registro foi realizado para uso em projetos futuros.

A Questão 22 apenas verifica se as reuniões de avaliação final de projeto foram realizadas. As opções de resposta eram: 1. Sim – com dez respostas (45,5%); e 2. Não – 12 respostas (54,5%). Houve um equilíbrio entre os itens, embora exista recomendação formal do PSDS para que esta reunião seja realizada. O fato é que, para os projetos em que esta equipe participou, em mais da metade das ocasiões a reunião de avaliação final do projeto não foi realizada.

A pergunta 23 identifica, para os projetos em que houve a reunião, quais as informações registradas. Os principais temas considerados nestas reuniões são os pontos fracos e fortes do projeto (oito e nove respostas, respectivamente). Outros aspectos relevantes, como registro das decisões (três respostas) e erros cometidos (seis respostas) aparecem com percentuais bem mais baixos.

7.1.7. Avaliação do Terceiro Pressuposto Específico

A revisão de literatura mostrou que a qualidade dos produtos de *software* pode ser melhorada com o reuso de informações de projetos já concluídos, desde que existam condições adequadas que permitam o reuso pelas equipes de desenvolvimento de *software*.

As respostas ao questionário mostram que parte da equipe percebe que o PSDS orienta a prática de registro de experiências, como comprovado pelas respostas à Questão 16. Entretanto, em seguida é mostrado que apenas cerca de 16% dos entrevistados utilizam esta prática com frequência e os demais a utilizam apenas esporadicamente, ou nunca a utilizam (cerca de 26% dos entrevistados).

Este cenário é reforçado com os dados das questões seguintes, onde é possível observar que cerca de 63% dos participantes nunca consulta a base de soluções antes de novos projetos, e apenas 9,1% o fazem com frequência.

Da mesma forma, quando perguntados se registram informações de projeto na base de soluções, quase 90% das respostas indicaram que a base de soluções não é alimentada com informações relevantes – possível causa para seu baixo uso em outros projetos.

As razões apresentadas para a não utilização da BS confirmam outras pesquisas sobre o assunto, sendo a principal razão a falta de tempo, em função dos prazos curtos para entrega dos projetos de *software*. Cabe destaque para a idéia comum de que as pessoas não terão interesse em consultar a BS, razão apontada por quase 44% das respostas para não fazer registro das experiências .

A falta de ferramenta adequada para registro e recuperação também foi apontada como fator relevante que não estimula o registro de informações sobre os projetos concluídos.

Finalmente, as Questões 22 e 23 indicam que em menos da metade dos projetos é realizada reunião formal de avaliação, embora esta prática seja considerada importante para registro de pontos negativos e positivos do projeto, estimulando o compartilhamento de experiências e uma situação de aprendizagem para outras equipes.

Assim, embora o PSDS oriente o uso de algumas práticas de registro e compartilhamento de informações, fica claro que o processo e o fluxo de trabalho

no Serpro não favorece a implementação de práticas de reuso. Por outro lado, os dados da Questão 25 confirmam a percepção de que o reuso pode melhorar a qualidade dos sistemas desenvolvidos pelo Serpro.

7.1.8. Aspectos Gerais – Bloco IV

Este bloco de questões – itens 24 a 27 - trata da percepção que a equipe de requisitos tem do processo de gestão da informação, usando o processo de reuso de informações como base para o questionamento. O objetivo é oferecer subsídios a implementação do modelo para as equipes de desenvolvimento.

7.1.8.1. Conhecimento do conceito de gestão da informação

A Questão 24 avalia o grau de conhecimento que a equipe de requisitos demonstra do conceito de gestão da informação, lembrando que não foi pedida verificação de conhecimento do conceito. Cerca de 90% das pessoas, 20 respostas, consultadas informaram que conhecem o conceito de gestão da informação.

O dado mostra que o ciclo de gestão da informação já possui algum nível de conhecimento na equipe, e que a implantação de um modelo de gestão da informação junto ao PSDS não deve apresentar resistências por parte da equipe.

7.1.8.2. Influência do reuso de informações

As Questões 25 e 26 medem a percepção da equipe quanto a influência que o reuso de informações pode gerar nos projetos de sistemas de informação. A Questão 25 mostra que os empregados percebem valor no reuso de informações para a evolução da qualidade dos projetos de *software*. Todos os entrevistados responderam que o reuso é importante para a evolução da qualidade de *software*.

A Questão 26 identifica aspectos em que o reuso de informações pode auxiliar projetos de *software*. Ela apresenta quatro itens para resposta, sendo que todos obtiveram altos índices de respostas, o que comprova o potencial de melhoria que a prática de reuso de *software* pode levar aos projetos de *software*. O item com menor valor (16 respostas, ou 72,7%) – refere-se a maior facilidade para tratamento de requisitos não-funcionais, e os de maior valor (21 e 20 respostas) tratam de requisitos não-funcionais e evitar erros de projetos anteriores.

7.1.8.3. Posição do Serpro quanto ao reuso de informações

A Questão 27 retrata, na visão dos empregados, a forma que o Serpro trata a questão de reuso de informações na área de desenvolvimento de *software*. Foi incluída uma opção na qual o Serpro não considera o assunto como relevante, mas ela não recebeu respostas, i. e., os empregados julgam que o Serpro considera o assunto relevante.

As outras quatro opções julgam que o Serpro considera o assunto relevante, mas com as seguintes ressalvas, representadas também no Gráfico 10:

- Não há ferramentas adequadas para tratar reuso – 12 respostas (54,5%);
- Processo (PSDS) não oferece suporte ao reuso de informações – seis respostas (27,3%);
- Não há incentivo aos empregados – oito respostas (36,4%);
- Não há uma estratégia geral para implantação do conceito – 16 respostas (72,7%).

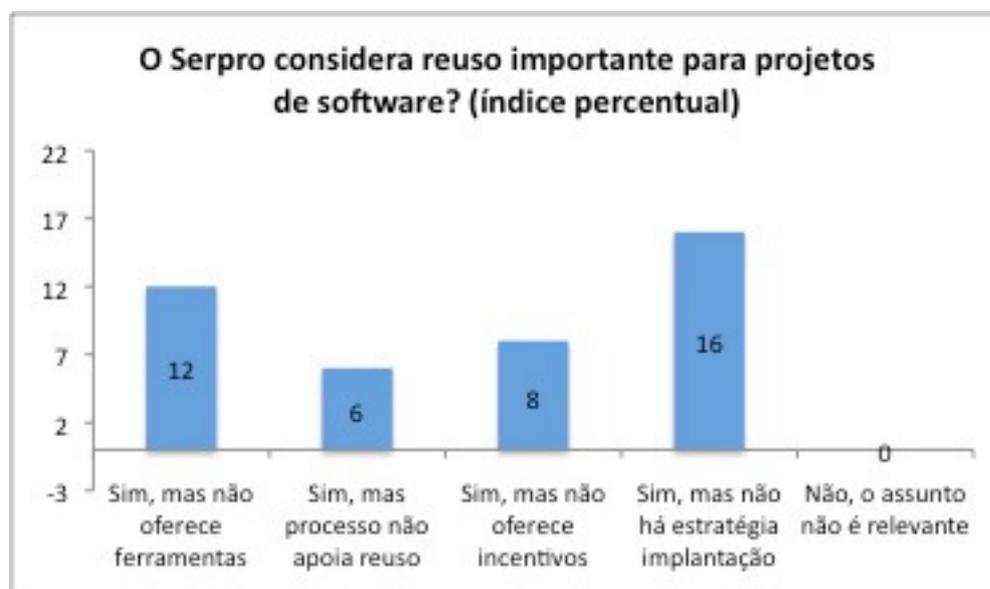


Gráfico 10 – Percepção sobre reuso de informações e o Serpro

Aspectos Gerais – Conclusão

As respostas das quatro últimas questões mostram que os analistas de requisitos entendem que o reuso de informações pode efetivamente melhorar a qualidade dos produtos de *software* desenvolvidos pelo Serpro.

Eles destacaram que o reuso de informações pode auxiliar no tratamento dos RNFs, evitar a repetição de erros e reduzir o tempo dos projetos. Consideram também que a empresa reconhece o assunto como relevante, mas não consegue traduzir esta importância na elaboração de uma estratégia global para o assunto, e ao não adequar o PSDS a este propósito e não ajustar as ferramentas para atender a esta finalidade.

Estes itens devem ser avaliados e tratados para a elaboração de ações de reuso de informações em projetos de *software*.

7.2. Análise dos Dados e o Ciclo de Gestão da Informação

Este item analisa os dados coletados, de acordo com os processos previstos por Choo para o ciclo de gestão da informação. O objetivo é oferecer subsídios para a identificação dos elementos chave, e futura construção de um modelo para aplicação dos conceitos de gestão da informação ao PSDS.

7.2.1. Necessidades de informação

Le Coadic destaca que a análise das necessidades de informação abrange cinco etapas: identificar os usuários e usos da informação; descrever a população alvo e o ambiente; identificar as necessidades desta população; avaliar as necessidades; e descrever, comunicar e implementar decisões.

Ele explica ainda que as necessidades dos usuários devem ser entendidas no contexto do mundo real, como base para a tomada de ação.

A análise das respostas da Questão 2 mostra que não há um processo sistematizado de identificar as informações necessários para o início dos projetos. Os artefatos definidos pelo PSDS têm foco em funcionalidades do sistema e formalidades para geração de evidências do processo.

Por outro lado, os dados da Questão 9 indicam que, para 54,6% dos participantes da pesquisa, não há uma identificação sistemática dos requisitos não-funcionais que devem ser usados para os sistemas.

7.2.2. Aquisição da Informação

O processo de aquisição de informação deve atender a duas demandas principais: possuir amplitude suficiente para atender as necessidades organizacionais de refletir as mudanças do ambiente; e selecionar as mensagens a serem examinadas, em função da limitação da capacidade de atenção e tratamento.

No caso da macroatividade de requisitos, a descrição define o uso de diversas fontes de informação a serem pesquisadas no cliente e outras partes interessadas no projeto, externas ao Serpro. Fontes internas não são citadas, o que não favorece o aproveitamento de informações já existentes na empresa, e que podem ser relevantes para o projeto.

As respostas da Questão 4 mostram que a fonte mais relevante de informações – sites do cliente na Internet – não é citada pelo processo. O segundo item mais frequente – “conversa com colegas” – também não é mapeado no PSDS, mostrando que há fluxos de informação não tratados.

Os dados da Questão 5 confirmam a importância de fluxos externos para aquisição de informações, já que os três itens citados são tratados pelo processo de *software*.

A Questão 11 evidencia a necessidade de melhoria no processo de aquisição de informações internas, ao apontar a dificuldade dos clientes em especificar os requisitos não-funcionais para suas aplicações.

7.2.3. Organização e Armazenamento da Informação

A importância deste processo está nas facilidades que ele pode gerar para a recuperação e compartilhamento da informação. Os ativos de informação no processo de desenvolvimento de *software*, não estão em dados transacionais, mas sim em relatórios, atas de reunião, documentos de requisitos e na cabeça das pessoas.

As respostas das Questões 6 e 10 destacam que a principal forma de armazenamento de informações sobre os projetos é feita em artefatos definidos pelo processo – artefatos de requisitos, relatórios de projeto e orientações do processo. Como a organização e armazenamento dos artefatos na ferramenta é por projeto, e as pessoas são habilitadas apenas nos projetos em que participam diretamente, este modelo de uso dificulta o acesso às informações.

As informações e experiências de interesse comum deveriam estar na BS, mas as repostas as Questões 18 e 20 mostram que 86,4% da amostra não registra estas informações. Assim, a regra geral é que pessoas do desenvolvimento tenham acesso apenas aos artefatos dos projetos em que participam.

De forma complementar, conforme respostas às Questões 19 e 21, um percentual elevado de entrevistados afirma que a ferramenta utilizada não oferece facilidades de registro e pesquisa.

Desta forma, a estratégia usada pelo PSDS não atende premissa básica de Choo para este processo, quando afirma que a informação, capturada ou criada, tem que ser organizada e armazenada para facilitar seu compartilhamento e recuperação.

7.2.4. Produtos e Serviços de Informação

O desenvolvimento de produtos e serviços de informação tem como objetivo oferecer informações relevantes para os usuários, e aumentar seu potencial de uso. Taylor destaca ainda que eles devem adicionar valor a informação processada, e levar seus usuários a melhores decisões.

As respostas registradas para as Questões 14 e 15 indicam que a maior parte dos entrevistados entende que o processo de desenvolvimento oferece produtos e serviços relacionados a requisitos. Para a Questão 14, cerca de dois terços apontaram sua disponibilidade, e indicaram na Questão 15 que estes produtos/serviços seriam os artefatos do processo.

Em relação aos artefatos, o PSDS define 18 artefatos para a macroatividade de requisitos, com modelo de formato e descrição resumida do conteúdo de cada campo. Uma vez produzidos os artefatos são armazenados em um repositório, sem preocupação com aspectos de busca e recuperação.

Assim, é possível afirmar que os artefatos do PSDS não atendem aos seguintes critérios básicos definidos por Taylor para produtos e serviços de informação: facilidade de uso, redução de ruído, qualidade e redução de custo e tempo.

7.2.5. Distribuição da Informação

O principal resultado deste processo é a disseminação e compartilhamento de informações. Além de favorecer a aprendizagem organizacional, ele favorece a criação de novas informações, ponto importante para uma atividade tão dinâmica como o desenvolvimento de *software*.

A Questão 8 mostra que 31,8% das pessoas entendem que não há mecanismo para compartilhar informações sobre os clientes. Entre os demais, 68,2 entendem que o compartilhamento em reuniões atende a esta função, enquanto 31,8% informaram que também o fazem por email.

Já o resultado da Questão 17 diz que apenas 18,2% das pessoas se preocupam em sempre distribuir informações de seus projetos. Os demais, mesmo considerando que a informação é relevante nunca o fazem (31,8%) ou o fazem apenas algumas vezes (50 %).

A Questão 20, que trata especificamente da base de soluções, mostra que 86,4% das pessoas nunca fazem registros na base de soluções, mostrando que é baixa a preocupação com o compartilhamento e o fluxo de informações dentro da equipe, e com outras equipes.

Os dados confirmam que o PSDS não se preocupa com a distribuição da informação. O processo descreve o formato dos artefatos, e indica a equipe onde utilizá-los, mas sem tratar aspectos relacionados a distribuição e compartilhamento de informação. A distribuição é abordada de forma indireta em algumas atividades, pela circulação de artefatos entre os diversos papéis.

7.2.6. Uso da Informação

Choo descreve o uso da informação como um processo iterativo, onde geração de novas informações e conhecimento leva a tomada de decisão. Este deveria ser um ponto de grande relevância em projetos de desenvolvimento de sistemas, já que as pessoas envolvidas são, ao longo do projeto, responsáveis por muitas decisões, que devem ser tomadas em tempo de execução das atividades.

O PSDS descreve o uso de informações registradas nos artefatos de requisitos, mas sua amplitude é restrita aos passos previstos para cada artefato, e mesmo assim de forma limitada.

Os dados obtidos como respostas nas Questões 17 a 23 confirmam esta afirmação, ao mostrar que há um baixo índice de registro de informações relevantes dos projetos já concluídos, o que facilita a repetição de erros, dificulta sua correção de erros e atrapalha a evolução da qualidade dos produtos de *software*.

8. CONCLUSÕES

O cenário de desenvolvimento de *software* é complexo e dinâmico, consequência da dependência cada vez maior que os negócios têm do bom funcionamento de seus sistemas de informação, da evolução tecnológica e das exigências cada vez mais severas das organizações por maior qualidade dos sistemas e menor prazo para seu desenvolvimento. Eles devem ter capacidade de atender grandes volumes de usuários, com graus elevados de performance e segurança, além de flexibilidade para atender novas necessidades.

Historicamente, as empresas de desenvolvimento de sistemas se valeram de evoluções de tecnologias e métodos para atender estas demandas. No caso de tecnologias é possível citar a evolução dos processadores, memória, capacidade de armazenamento, linguagens de programação e bancos de dados. Na área de métodos e modelos de trabalho, os mais conhecidos são o RUP e o CMMI, tentativas de tratar a disciplina como a engenharia, com maior previsibilidade de execução e de resultados.

A revisão de literatura mostrou que esta abordagem não é mais suficiente para atender as demandas das organizações para a construção e manutenção de seus sistemas de informação. Uma das alternativas para atender a estas demandas é a associação com outras disciplinas. Características da atividade de desenvolvimento de *software*, como dependência de pessoas, troca de experiências e multidisciplinaridade, a aproximaram da ciência da informação. Como resultado, estudos na área de engenharia de software são direcionados para avaliação de mecanismos de captura, organização e utilização de informações e conhecimento.

Foi constatado que a tecnologia é relevante e deve ser considerada, mas fica claro que as organizações de desenvolvimento de *software*, empresas de excelência técnica por natureza, têm dificuldade no tratamento dos aspectos organizacionais e culturais voltados para os processos associados à gestão da informação.

A revisão indicou ainda que o aproveitamento de informações já disponíveis nas empresas, e de técnicas e modelos de gestão da informação, em conjunto com processos de desenvolvimento de *software*, produziu resultados positivos. Um ponto comum entre os projetos de sucesso é tratar os outros aspectos relevantes para a disciplina, como os culturais e os voltados para modelos de processos e uso de informação.

A pesquisa buscou confirmar estes pontos em relação ao processo de desenvolvimento de *software* do Serpro, e estabeleceu um pressuposto geral de que a identificação dos elementos que afetam os processos relacionados a gestão da informação pode melhorar os aspectos relacionados ao levantamento de informações dos clientes, permitir a criação de produtos e serviços de informação e levar ao aumento do reuso de informações de projetos anteriores.

Os dados coletados mostram que a equipe de requisitos considera importante a disponibilidade de informações sobre os clientes e o domínio das aplicações, mas não há uma sistemática para identificá-las, identificar suas fontes

e capturá-las. As fontes de informação mais utilizadas pelos analistas não são tratadas pelo PSDS, e não há consenso sobre a disponibilidade de mecanismos de compartilhamento das informações coletadas. O armazenamento em artefatos não oferece facilidades de busca e recuperação, o que dificulta o fluxo de informações entre as equipes.

Foi percebido também que os requisitos não-funcionais (RNF) são importantes para a identificação de domínio e contexto de negócios dos sistemas, mas os clientes têm dificuldades em sua especificação e o processo de desenvolvimento não possui sistemática de identificação e proposição de RNF para os domínios de negócio atendidos pelo Serpro. Este aspecto diminui as oportunidades de reuso de informações para projetos de mesmos clientes, ou de mesmo domínio de negócio.

Em relação ao registro de informações e seu aproveitamento em outros projetos, foi verificado que, em sua maior parte, os analistas de requisitos não registram informações ou decisões que consideram relevantes nos projetos em que participam. Eles também não consultam a base de soluções, para avaliar o que poderia ser aproveitado em novos projetos.

A prática de reuso de informações não é sistematizada e não utiliza o compartilhamento de experiências entre as equipes. Há uma dependência apenas das experiências anteriores dos envolvidos. As razões apresentadas pelos analistas do Serpro para este comportamento seguem o padrão apresentado na literatura, e incluem aspectos técnicos, culturais e organizacionais.

Finalmente, os dados coletados na pesquisa confirmam a percepção de que o reuso de informações pode melhorar a qualidade dos produtos de software desenvolvidos pelo Serpro, em especial para evitar a repetição de erros de projetos anteriores e melhorar o tratamento de requisitos não-funcionais.

Desta forma, os pressupostos estabelecidos foram confirmados, e as perguntas formuladas na discussão do problema – associadas aos processos de gestão da informação; ao fluxo de informações no processo de desenvolvimento; e ao aproveitamento das informações de projetos anteriores – foram respondidas.

É possível concluir que foram identificados os elementos chave para a construção de um modelo de gestão da informação para a atividade de requisitos dos processo de desenvolvimento de *software* do Serpro: identificação das informações sobre clientes e domínio de negócios; requisitos não-funcionais; e reuso de informações já disponíveis na empresa.

Com base nas conclusões, são feitas as seguintes recomendações para a adoção de conceitos e modelos de gestão da informação no PSDS:

- Revisar as atividades e artefatos associados a identificação de informações dos clientes e seus domínios de negócio;
- Trabalhar a definição e implementação de produtos e serviços de informação para as equipes de requisitos, que facilitem os processos de registro, busca, recuperação e uso das informações necessárias ao desenvolvimento;
- Estabelecer fluxos que estimulem o compartilhamento de informações pelas equipes de projetos;

- Definir práticas de reuso das informações, que considerem os aspectos técnicos, culturais e organizacionais identificados na pesquisa.

O Apêndice B descreve uma proposta de modelo que utiliza os elementos identificados na pesquisa, e as recomendações acima descritas, e que poderá ser utilizado pelo Serpro para uma experiência piloto em uma equipe de desenvolvimento e posterior avaliação dos resultados.

8.1. Sugestões para novas pesquisas

Os resultados e conclusões obtidos a partir desta pesquisa podem levar a novos trabalhos, sugeridos a seguir:

- Ampliar a avaliação para a construção de modelo de gestão da informação contemplando todas as macroatividades do PSDS;
- Avaliar aspectos relacionados ao comportamento da informação em empresas de TI e a atuação de profissionais de desenvolvimento de sistemas como profissionais da informação;
- Avaliar modelos de compartilhamento de informações que possibilitem a transformação de empresas de desenvolvimento de software em organizações de aprendizagem;
- Expandir a análise e o modelo para outras áreas do Serpro, com estudos sobre políticas de informação para empresas prestadoras de serviços de TI;
- Ampliar o estudo para outras organizações de desenvolvimento de software. Como muitas organizações utilizam modelos de desenvolvimento similares ao do Serpro, poderia ser feito um estudo comparativo entre estas organizações, avaliando aspectos relacionados a empresas públicas e privadas; e de portes diversos;
- Analisar estratégias de colaboração e compartilhamento de informações organizacionais em empresas de serviços de TI, e de que forma elas são influenciadas por ferramentas típicas da Web 2.0 – como YouTube, Gengibre, Facebook e Twitter. A pesquisa deve trabalhar nos aspectos dos fluxos e modelos de uso da informação.

REFERÊNCIAS

- ALVARENGA NETO, R. C. D., BARBOSA, R. R., PEREIRA, H.J. Gestão do conhecimento ou gestão das organizações na era do conhecimento? Um ensaio teórico-prático a partir de intervenções na realidade brasileira. *Perspectivas em Ciência da Informação*, Belo Horizonte, v. 12, n. 1, p. 5-24, jan/abr 2007. Disponível em < <http://www.eci.ufmg.br/pcionline/index.php/pci/article/viewFile/246/34>>. Acesso em 11.mar.2009.
- ARAÚJO JÚNIOR, R. H. *Precisão no Processo e Busca e Recuperação da Informação*. Brasília: Thesaurus Editora, 2007.
- ARAÚJO JÚNIOR, R. H. *Estudo de Necessidades de Informação dos Gerentes do Setor Editorial e Gráfico do Distrito Federal*. Dissertação de Mestrado apresentada no Departamento de Ciência da Informação e Documentação da Universidade de Brasília. Brasília, 1998.
- BARRETO, A. A. A eficiência técnica e econômica e a viabilidade de produtos e serviços de informação. *Ciência da Informação*, Brasília, v.25, n. 3, p. 405-414, 1996.
- BASILI, V. R., MCGARRY, F. E., PAJERSKI, R., ZELKOWITZ, M. V. Lessons learned from 25 years of process improvement: The Rise and Fall of the NASA Software Engineering Laboratory. Proceedings of the Twenty-Fourth International Conference on Software Engineering (ICSE), Orlando, FL, May 2002. Disponível em: <<http://www.cs.umd.edu/~mvz/pub/basili-et-al-sel-2002.pdf>> . Acesso em: 12 jan.2008.
- BASILI, V., CALDIERA, G., ROMBACH, H. D. The Experience Factory, *Encyclopedia of Software Engineering*, pp. 469-476, John Wiley & Sons, Inc., 1994. Disponível em: <<http://www.cs.umd.edu/~basili/publications/technical/T86.pdf>>. Acesso em: 18 nov. 2007.
- BETTIOL, E. M. Necessidades de Informação: uma revisão. *Revista de Biblioteconomia de Brasília*, v.18, n. 1, p. 59-69, jan/jun 1990.
- BIRK, A., DINGSOYR, T. STALHANE, T. Postmortem: Never Leave a Project Without It. *IEEE Software*, p. 43-45, May/June 2002.
- BLASCHEG, J. R., Gerencia de Requisitos: o principal problema nos projetos de software. Rio de Janeiro, 2002. Disponível em: <<http://www.bfpug.com.br/islig-rio/Downloads/Gerência%20de%20Requisitos-o%20Principal%20Problema%20dos%20Projetos%20de%20SW.pdf>>. Acesso em: 26 jan.2009.
- BOHEM, B., BASILI, V. Software Defect Reduction Top 10 List. *IEEE Computer*. Vol. 34, no. 1, pp. 135-137, 2001. Disponível em: <<http://www.cs.umd.edu/~basili/publications/journals/J81.pdf>>. Acesso em 20 fev.2008.
- BJORNSON, F.O., DINGSOYR, T. Knowledge Management in Software Engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*. Vol. 50, pp. 1055-1068, 2008.

- BUSH, V. As We May Think. *The Atlantic*, jul.1945. Disponível em: <<http://www.theatlantic.com/doc/194507/bush> >. Acesso em: 23 abr.2009.
- CARVALHO, I. M., ISIDRO-FILHO, A. Comportamentos que Contribuem para o Compartilhamento de Conhecimento em uma empresa pública. KMBrasil 2008 – 7º Congresso Anual de Gestão do Conhecimento. Anais do KMBrasil 2008. São Paulo: SBGC, 2008.
- CASTELLS, M. *A Sociedade em Rede*. Tradução: Roneide Venancio Majer – 5. Ed. São Paulo: Ed. Paz e Terra, 2001.
- CHOO, C. W. Towards an Information Model of Organizations. *The Canadian Journal of Information Science*, 1991, vol. 16, n. 3, p. 32-62. Disponível em: <<http://choo.fis.utoronto.ca/FIS/respub/CJIS.html>>. Acesso em: 8 mar.2009.
- CHOO, C. W. *Information Management for the Intelligent Organization*. Medford/NJ. Information Today Inc, 2002.
- CHOO, C. W. *A Organização do Conhecimento*. Tradução: Eliana Rocha – 2. Ed. São Paulo: Editora SENAC, 2006.
- CHUNG, L., NIXON, B., YU, E. Using Non-Functional Requirements to Systematically Select Among Alternatives in Architectural Design. Proceedings, 1st International Workshop on Architectures for Software Systems, Seattle, April 24-28, 1995, pp. 31-43. Disponível em: <<http://www.utdallas.edu/~chung/ftp/IWASS95.ps>>. Acesso em 12 mai.2009
- DAVENPORT, T. H. *Reengenharia de Processos: como inovar na empresa através da tecnologia da informação*. Tradução Waltensir Dutra – 4. Ed. Rio de Janeiro: Editora Campus, 1994.
- DAVENPORT, T. H. *Information Ecology: Mastering the Information and Knowledge Environment*. New York. Oxford University Press, 1997.
- DAVENPORT, T. H., MARCHAND, D. A. Is KM just good information management? 2001. Disponível em: < http://www.providersedge.com/docs/km_articles/Is_KM_Just_Good_Information_Management.pdf>. Acesso em: 1 mar.2009.
- DAVENPORT, T. (org.) MARCHAND, D. A. (org.) *Dominando a Gestão da Informação*. Trad. Carlo Gabriel Porto Bellini e Carlos Alberto Silveira Netto. Porto Alegre: ed. Bookman, 2004.
- DAVIS, A. M. Achieving Quality in Software Requirements. *Software Quality Professional*. Vol. 1, issue 3, pp. 37-44, jun.1999. Disponível em: < <http://www.p2080.co.il/go/p2080h/files/67683.pdf> >. Acesso em: 27.maio.2009.
- DE GEUS, A. *A empresa viva: como as organizações podem aprender a prosperar e se perpetuar*. Tradução Lenke Peres. – Rio de Janeiro: Ed. Campus, 1998.
- DESOUZA, Kevin C., “Barriers to Effective Use of Knowledge Management Systems in Software Engineering”. *Communications of the ACM*, Vol. 46, No. 1, pp. 99-101, 2003. Disponível em:<<http://ssrn.com/abstract=522303>>. Acesso em: 04 nov.2007.
- DINGSOYR, T. An Evaluation of Research on Experience Factory. Workshop on Learning Software Organisations at the Internacional Conference on Product-Focused Software Process Environment, Oulo, Finland, p. 55-66, 2000.

DINGSOYR, T., CONRADI, R. A Survey of Case Studies of the Use of Knowledge Management in *Software Engineering*. *International Journal of Software Engineering and Knowledge Engineering*, v. 12, n. 4 , p. 391-414, 2002. Disponível em: <http://alarcos.inf-cr.uclm.es/doc/masi/doc/lec/parte4/dingsoyr-ijseke2002.pdf>. Acesso em 26 jan.2009.

DRUCKER, P. Viewpoint: What Executives Need to Learn. *Prism Arthur D. Little*, Cambridge, Mass, 4 trimestre, 1990, p. 73-84. Disponível em: <http://www.adl.com/prism.html>. Acesso em: 8 mar.2009.

ESPINDOLA, R. S., MAJDENBAUM, A., AUDY, J. L. Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de *Software*. Workshop de Engenharia de Requisitos – Tandil/Argentina, 2004. Disponível em: http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER04/Rodrigo_Espindola.pdf. Acesso em: 26 jan.2009.

FALBO, R. A. Engenharia de *Software* – Notas de Aula. Universidade Federal do Espírito Santo, 2005. Disponível em <http://www.inf.ufes.br/%7Efalbo/download/aulas/es-g/2006-2/NotasDeAula.pdf>. Acesso em 16.fev.2009.

GLASS, R. L. Reuse: What's Wrong With This Picture? *IEEE Software*, p.57-59, march/april, 1998.

GLASS, R. L. The Realities of *Software* Technology. *Communications of ACM*, vol. 42, issue 2, p. 74-79, feb. 1999.

HANSEN, M. T., NOHRIA, N., TIERNEY, T. What's your Strategy for Managing Knowledge? *Harvard Business Review*, vol. 77, n. 2, p. 106-116, mar. 1999. Disponível em: <http://www.itu.dk/~kristianskriver/b9/Whats%20your%20strategy%20for%20managing%20knowledge.pdf> >. Acesso em: 14 mar.2009.

HOUDEK, F., SCHNEIDER, K., WIESER, E. Establishing Experience Factories at Daimler-Benz. An Experience Report. *Proceedings of the 20th international conference on Software engineering*. April 1998, Kyoto, Japan.

IBGE (Instituto Brasileiro de Geografia e Estatística) – *Relatório “O Setor de Tecnologia da Informação e Comunicação no Brasil 2003-2006”*. Disponível em: <http://www.ibge.gov.br/home/estatistica/economia/stic/publicacao.pdf>. Acesso em: 19 abr.2009.

IEEE SWEBOK – Guide to the *Software Engineerig* Body of Knowledge. IEEE Computer Society, Los Alamitos/CA, 2004. Disponível em: <http://www.swebok.org>. Acesso em: 02.fev.2009.

JOHANSSON, C., HALL, P. A. V., COQUARD, M. Talk to Paula and Peter – They are Experienced” – The Experience Engine in a Nutshell. *Proceedings of the 11th International Conference on Software Engineerign ank Knowledge Applications*. Pp. 171-185, 1999. Disponível em: <http://www.ipd.bth.se/connyj/pt/articles/papers/paper7-1.pdf>. Acesso em: 10.dez.2008.

KIM, Y., STOHR, E. A. Software Reuse: Survey and Research Directions. *Journal of Management Information Systems*. Vol. 14, is. 4, pp. 113-147, March 1998.

KRUCHTEN, P. *The Rational Unified Process – An Introduction*. 2nd ed. Massachusetts: Addison-Wesley, 2000.

- KRUCHTEN, P. The Nature of Software: What's so Special About Software Engineering?. Sítio IBM, 29.apr.2004. Disponível em: <<http://www.ibm.com/developerworks/rational/library/4700.html>>. Acesso em: 26.jan.2009.
- LE COADIC, Y. F. *A ciência da informação*. Tradução de Maria Yêda F. S. de Filgueiras – Brasília: Ed. Briquet de Lemos/Livros, 1996.
- LEFFINGWELL, D., WIDRIG, D. *Managing Software Requirements*. 5.ed: Ed. Addison-Wesley, 2000.
- LYYTINEN, K., ROBEY, D. Learning Failure in information systems development. *Information Systems Journal*, vol. 9, pp. 85-101, 1999.
- MACEDO, F. L. O. Arquitetura da Informação: aspectos epistemológicos, científicos e práticos. Dissertação apresentada ao Departamento de Ciência da informação da Universidade de Brasília, 2005.
- MARCHIORI, P. Z. A ciência e a gestão da informação: compatibilidade no espaço profissional. *Ciência da Informação*, Brasília, v.31, n. 2, p. 72-79, mai/ago 2002. Disponível em:< <http://www.scielo.br/pdf/ci/v31n2/12910.pdf>>. Acesso em: 23 abr.2009.
- MARTIN, L. A. User Studies and Library Planning. *Library Trends*, vol. 24, i. 3, p. 483-496, 1976.
- MICHAUD, C. *Inteligência, Informação e Conhecimento*. Kira Tarapanoff, organizadora. Brasília: IBICT/UNESCO, 2006.
- MORESI, E. A D. Delineando o Valor do Sistema de Informação de uma Organização. *Ciência da Informação*, Brasília: v.29, n. 1, p. 14-24, jan/abr. 2000. Disponível em: < <http://www.scielo.br/pdf/ci/v29n1/v29n1a2.pdf>> . Acesso em: 15 abr.2009.
- NIST (National Institute of Standards and Technology). The Economic Impacts of Inadequate Infrastructure for *Software Testing* – maio/2002. Disponível em: <<https://www.nist.gov/director/prog-ofc/report02-3.pdf>>. Acesso em: 23 abr.2009.
- NUSEIBECH, B., EASTERBROOK, S. Requirements Engineering: A Roadmap. 22nd International Conference on *Software Engineering*, ICSE'00 – IEEE Computer Society. Disponível em: <<http://www.cs.toronto.edu/~sme/papers/2000/ICSE2000.pdf>> . Acesso em: 26 jan.2009.
- OBERG, R., PROBASCIO, L., ERICSSON, M. Applying Requirements Management with Use Cases. Rational *Software White Paper Series* (TP05), 2000. Disponível em: <http://www.uml.org.cn/requirementproject/pdf/wp-applying_requirements_management_withusecases.pdf>. Acesso em: 03 fev.2009.
- OLIVEIRA, M. M. Proposta de Modelo de Representação do Capital Intelectual de Organizações que Desenvolvem Software: Um Estudo do Distrito Federal. Tese de Doutorado. Curso de Pós-Graduação do Departamento de Ciência da Informação e Documentação da Universidade de Brasília, 2008.
- PARREIRAS, F. S., BAX, M. P. A gestão de conteúdos no apoio a engenharia de *software*. In: KMBrazil, 2003, São Paulo. Anais... São Paulo: SBGC – Sociedade Brasileira de Gestão do Conhecimento. 2003. CD-ROM. Disponível em <<http://www.fernando.parreiras.nom.br/publicacoes/pgct142.pdf>>. Acesso em: 26 out.2007.

PARREIRAS, F. S., OLIVEIRA, G. S. Análise comparativa de processos de desenvolvimento de *software* sob a luz da gestão do conhecimento: um estudo de caso de empresas mineiras. Simpósio Brasileiro da Qualidade de *Software*, 3, 2004, Brasília. Disponível em <http://www.fernando.parreiras.nom.br/publicações/WGC_Parreiras04.pdf>. Acesso em: 20 jan.2009.

PRESSMAN, R. S. *Engenharia de Software*. 6. Ed. São Paulo: McGraw-Hill, 2006.

PRIKLADNICKI, R., AUDY, J., EVARISTO, R. Requirements Management in Global *Software* Development: Preliminary Findings from a Case Study in I-CMM Context. Workshop on Global *Software* Development at ICSE 2003, Portland, Oregon, 2003. Disponível em: <<http://gsd2003.cs.uvic.ca/upload/Prikladnicki.pdf>>. Acesso em: 20.dez.2008.

PSDS – Processo Serpro de Desenvolvimento de Soluções. Site da Intranet do Serpro. Disponível em: <<https://psds.portalcorporativo.serpro>> . Acesso em: 14.12.2007. (Site disponível apenas na Intranet do Serpro – conteúdo não disponível ao público externo).

ROBERTS, J. The Drive to Codify: Implications for the Knowledge-Based Economy. 8th International Joseph A. Schumpeter Society Conference, University of Manchester, 2000. Disponível em: <<http://www.cric.ac.uk/cric/events/schumpeter/papers/40.pdf>>. Acesso em: 24 fev. 2009.

ROBREDO, J. *Da Ciência da Informação Revisitada aos Sistemas Humanos de Informação*. Brasília: Ed. Thesaurus; SSRR Informações, 2003.

RUS, I., LINDVALL, M. SINHA, S. S. Knowledge Management in *Software* Engineering. State of Art Report – Department of Defense Information Analysis Center (DACs) – 16.nov.2001. Disponível em: <<https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/6339b090-0201-0010-3099-d539710c0468>> . Acesso em: 4 fev.2008.

SARACEVIC, T. Ciência da Informação: origem, evolução e relações. *Perspectiva em Ciência da Informação*, Belo Horizonte, v.1, n.1, p. 41-62, jan/jun. 1996.

SAYÃO, L. F. Modelos Teóricos em Ciência da Informação – abstração e método científico. *Ciência da Informação*, Brasília, v.30, n. 1, p. 82-91, jan/abr, 2001.

SCHNEIDER, K, HUNNLUS, J. P., BASILI, V.R. Experience in Implementing a Learning *Software* Organization. *IEEE Software*, vol. 19(3): 46-49, May/June 2002. Disponível em:< <http://www.cs.umd.edu/~basili/publications/journals/J85.pdf>>. Acesso em 10.dez.2008.

SEI (*Software* Engineering Institute), Carnegie Mellon University. What is CMMI? Disponível em: < <http://www.sei.cmu.edu/cmml/general/index.html>>. Acesso em: 12 fev.2009.

SHERA, J. H., CLEVELAND, D. B. History and Foundations of Information Science. *Annual Review of Information Science and Technology*, 1977, v.12, p. 249-275.

SOMMERVILLE, I. *Engenharia de Software*. 6. Ed. São Paulo: Ed. Addison Wesley, 2003.

SVEIBY, K. E. What is Information? Oct 1994, atualizado em dec 1998. Disponível em: < <http://www.sveiby.com/articles/Information.html>>. Acesso em: 10 mar.2009.

SCHNEIDER, K., VON HUNNIUS, J. P, BASILI, V. Experience in Implementing a Learning *Software Organization*. *IEEE Software*, may/june 2002, p. 46-49.

TARAPANOFF, K. *Inteligência, Informação e Conhecimento*. Kira Tarapanoff, organizadora. Brasília: IBICT, Unesco, 2006.

TAYLOR, R. S. *Value-added Processes in Information Systems*. Norwood, NJ: Ablex Publishing Corporation, 1986.

THE STANDISH GROUP. Chaos Report – 1995. Disponível em: <<http://net.educause.edu/ir/library/pdf/NCP08083B.pdf>>. Acesso em : 1.dez.2008.

WILSON, T. D. Information Management. In: *International Encyclopedia of Information and Library Science*, 2. Ed. Organizado por John Feather and Paul Sturges. London: Routledge, 2002. Disponível em: <http://informationr.net/tdw/publ/papers/encyclopedia_entry.html>. Acesso em: 25 abr.2009.

WILSON, T. D. The nonsense of “knowledge management”. *Information Research*, vol. 8 No. 1, Oct 2002. Disponível em: < [http:// informationr.net/ir/8-1/paper144](http://informationr.net/ir/8-1/paper144)>. Acesso em: 1 mar.2009.

WIIG, K. M. Knowledge Management: An Emerging Discipline Rooted in a Long History. 1999. Disponível em: <http://www.krii.com/downloads/km_emerg_discipl.pdf>. Acesso em: 10 mai.2009.

WURMAN, R. S. *Ansiedade de Informação*. 5. Ed. São Paulo: Cultura Editores Associados, 1991.

GLOSSÁRIO

- Ambiente operacional – designação genérica para o ambiente em que um sistema de informação irá funcionar. Pode incluir itens como o *hardware* do computador, sistema operacional e outros componentes de *software*, como servidor de aplicação e sistema gerenciador de banco de dados. (fonte: elaborado pelo autor)
- Artefato - são produtos de trabalho gerados durante a execução das atividades no desenvolvimento de *software*. Modelos, diagramas, planos, relatórios ou documentos são tipos de artefatos comumente produzidos. As atividades utilizam artefatos como insumo e produzem artefatos como resultado de sua execução. (fonte: Site PSDS)
- Atividade - é uma unidade de trabalho que uma pessoa executa quando desempenha algum papel no processo de desenvolvimento de *software*. A atividade possui um propósito bastante claro, normalmente expresso em termos de criar ou atualizar algum artefato. Toda atividade é atribuída a um determinado papel. (fonte: Site PSDS)
- CMMI - Modelo Integrado de Maturidade de Capacitação (CMMI) do *Software Engineering Institute* (SEI), da Carnegie Mellon University. Tem como proposta a obtenção de um processo mensurável e controlado para atividades, com ações de melhoria contínua, permitindo que a empresa busque a maturidade gradativa e planejada do processo de desenvolvimento e manutenção de *software*. (fonte: Site PSDS)
- Desenvolvimento de software
- Domínio – em engenharia de software, é um conjunto de sistemas ou módulos que partilham um determinado conjunto de características, como atender a um determinado negócio ou usar tecnologias similares. De forma geral diz-se que um projeto de software usa pelo menos dois domínios de conhecimento: o do negócio a ser atendido, e da tecnologia de desenvolvimento de sistemas. (fonte: Wikipedia, com adaptação do autor)
- Engenharia de *software* - disciplina da engenharia que se ocupa de todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação (fonte: Sommerville, 2001); a criação e a utilização de sólidos princípios de engenharia a fim de obter *software* de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais. (fonte: Pressman, 2002)
- Papel - são responsabilidades atribuídas a uma ou mais pessoas. Uma mesma pessoa pode executar diversos papéis num mesmo projeto. Um papel é uma definição abstrata que especifica um conjunto de atividades realizadas e artefatos produzidos. (fonte: Site PSDS)

- Pólo de Desenvolvimento – unidade organizacional responsável por atividades de desenvolvimento de software no Serpro. Atualmente existem 11 Pólos de Desenvolvimento, em 11 localidades diferentes.
- PSDS (Processo SERPRO de Desenvolvimento de Soluções) – processo criado com o objetivo de padronizar e documentar as atividades realizadas para conceber soluções de TI para os clientes do Serpro. O termo “soluções” é usado para representar a abrangência do PSDS, que trata os aspectos de negócio, contratação, planejamento, desenvolvimento e implantação de um sistema de informação. Para este trabalho, o PSDS será usado apenas nos aspectos relacionados ao desenvolvimento de sistemas. (fonte: elaborado pelo autor)
- Requisito – capacidade de um *software*, necessitada pelo usuário para resolver um problema ou alcançar um objetivo; capacidade de um *software* que deve estar presente em um sistema, ou componente de sistema, para satisfazer um contrato, padrão, especificação ou outra documentação formal. (fonte: Leffingwell e Widrig)
- Requisito funcional – declarações das funções que um sistema deve oferecer, como o sistema se comporta em cada situação em sua operação. Em alguns casos os requisitos funcionais podem também explicitamente definir o que o sistema não deve fazer. (fonte: elaborado pelo autor)
- Requisito não-funcional - são normalmente registrados como restrições, ou condições específicas, para a operação do sistema – de tempo, custo, desempenho ou padrões a serem atendidos. Eles são parte importante na descrição do contexto de desenvolvimento de um sistema. Exemplos de requisitos não-funcionais são: o tempo de resposta do sistema deve ser inferior a 10 segundos; ele deve atender a até 1.000 usuários simultâneos; o sistema deve trabalhar em ambiente Linux.
- Reuso - uso de recursos de *software* previamente desenvolvidos, em todas as fases do ciclo de desenvolvimento, para uma nova aplicação. Um recurso reusável pode ser um módulo de programa, especificações de requisitos, estruturas de dados, casos de testes e decisões de projetos, dentre outros. (fonte: Kim e Stohr)
- Versionar – versão é uma variante de algum artefato, que evolui de versões anteriores. Versionar, em desenvolvimento de *software*, significa manter controle ou gerenciar as diversas versões de um artefato em um período, registrando as diferenças entre as versões e quem fez as alterações. É comum o seu uso para documentos e códigos fonte de sistemas. (fonte: Site PSDS e autor)

APÊNDICE A - INSTRUMENTO DE COLETA DE DADOS

PESQUISA SOBRE A POSSIBILIDADE DE UTILIZAÇÃO DE MODELO DE
GESTÃO DA INFORMAÇÃO NO PROCESSO SERPRO DE
DESENVOLVIMENTO DE SOLUÇÕES – ATIVIDADE DE REQUISITOS

Questionário

Data: / /

O questionário foi apresentado aos empregados selecionados pela Gerente do Pólo de Brasília, com o seguinte texto:

Caros,

Nosso colega Roberto Pontual está trabalhando em pesquisa para mestrado na área de ciência da informação, que trata da aplicação de um modelo de gestão da informação ao PSDS, especificamente na macroatividade de requisitos. Para a conclusão do trabalho ele elaborou questionários que aborda estes tópicos, e para o qual você foi selecionado para responder, em função de sua experiência com requisitos no SERPRO.

O tempo máximo para resposta é de até 15 minutos, e o resultado será divulgado para todos os empregados do Pólo. Para apoiar vocês ao responderem o questionário, peço participarem de uma reunião com o Roberto Pontual, na sala de reunião ao lado da minha sala.

Louise Neves, PMP

Gerente do Polo de Desenvolvimento Brasília – DEBSA/SUPDE

SERPRO - Serviço Federal de Processamento de Dados

SGAN QD. 601 Módulo G - Regional Brasília

O trabalho tem como objetivo propor a integração de um modelo de gestão da informação ao processo de desenvolvimento do Serpro, com foco na atividade de levantamento e gestão de requisitos.

Será avaliada a forma que as informações sobre os clientes, seus negócios e suas necessidades são trabalhadas, considerando os processos que formam o ciclo de gestão da informação.

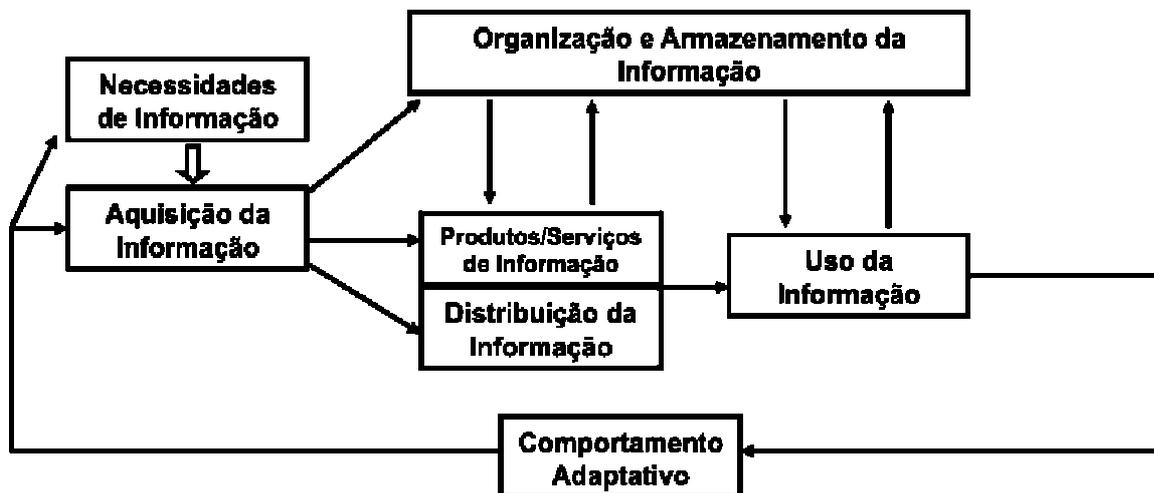


Figura - Ciclo da gestão da informação

Um dos focos da pesquisa é a avaliação da especificação de requisitos não-funcionais e a possibilidade de reuso de informações de projetos anteriores, como forma de reduzir erros e melhorar a qualidade dos sistemas desenvolvidos pelo Serpro. O questionário está dividido em 5 blocos de perguntas, e o tempo esperado para completar as respostas é de até 25 minutos.

BLOCO I – Informações sobre clientes

1) Você acha importante dispor de informações sobre o cliente e seu domínio de negócio antes de iniciar o levantamento de requisitos?

1. Sim

2. Não

2) Antes de iniciar o levantamento a equipe identifica as informações que devem ser levantadas junto ao cliente (além das funcionalidades para o sistema)?

1. Sempre

2. Algumas vezes

3. Nunca

3) Antes de iniciar o levantamento de requisitos você dispõe de alguma informação sobre o cliente?

1. Sim

2. Não

4) Informe abaixo onde as informações sobre o cliente, caso disponíveis, podem ser encontradas (mais de uma opção pode ser assinalada)

1. Base de dados sobre clientes
2. Relatório de projeto anterior
3. Base de Soluções do PSDS
4. Requisitos de projeto anterior
5. Site do cliente na Internet
6. Conversa com colega
7. Outra (especifique a fonte)

5) De que formas são obtidas as informações com os clientes sobre o sistema a ser desenvolvido? (mais de uma opção pode ser assinalada)

1. Entrevistas
2. Análise documentos cliente (relatórios, normas ou legislação)
3. Reuniões
4. Outras (descreva)

6) Onde as informações obtidas com os clientes sobre o sistema são armazenadas?

1. Registradas em artefatos do PSDS
2. Registradas em relatórios do projeto
3. Não são armazenadas

7) As informações obtidas junto ao cliente são compartilhadas com outras pessoas da equipe?

1. Sim

2. Não

8) Indique de que forma as informações obtidas junto aos clientes são compartilhadas com outras pessoas da equipe

1. Alerta da base do PSDS quando nova informação é adicionada

2. Email para membros da equipe

3. Reuniões da equipe

3. Não há mecanismo específico para compartilhar informações do projeto

BLOCO II – Informações sobre requisitos não-funcionais

9) Há uma relação de quais requisitos não-funcionais devem ser obtidos junto aos clientes?

1. Sempre

2. Algumas vezes

3. Nunca

10) Caso afirmativo, onde fica armazenada a relação de requisitos não-funcionais para uso nos projetos?

1. Base de dados

2. Artefatos de requisitos

3. Relatório de projeto

4. Orientação do PSDS

11) O cliente consegue fazer uma boa especificação dos requisitos não-funcionais para seus sistemas?

1. Sempre
2. Algumas vezes
3. Nunca

12) É possível afirmar que sistemas diferentes de um mesmo cliente tendem a ter requisitos não-funcionais semelhantes?

1. Sempre
2. Algumas vezes
3. Nunca

13) É possível afirmar que a equipe de requisitos utiliza informações sobre requisitos não-funcionais de projetos anteriores?

1. Sempre
2. Algumas vezes
3. Nunca

14) É possível afirmar que a equipe de requisitos dispõe de produtos e serviços de informação para suporte à execução de suas atividades?

1. Sim
2. Não (passar para pergunta 16)

15) Descreva, em sua opinião, quais são os produtos e serviços de informação que estão disponíveis para a equipe de requisitos

BLOCO III – Informações sobre reuso

16) É possível afirmar que no PSDS há orientação para registro de experiências e práticas durante a execução de um projeto de software no Serpro?

1. Sim

2. Não

17) Você faz registro de informações sobre o projeto de software em que atua, que você entenda que sejam relevantes para outros projetos?

1. Sempre

2. Algumas vezes

3. Nunca

18) Você consulta a base de soluções (BS) antes do início de um projeto de software, em busca de informações ou artefatos que possam auxiliar seu novo projeto?

1. Sempre

2. Algumas vezes (responder a pergunta 19)

3. Nunca (responder a pergunta 19)

19) Qual a razão de você não consultar a base de soluções antes do início de projeto de *software*? (marque todas que julgar adequadas)

1. As informações que ela oferece não são relevantes

2. Meu projeto possui um contexto diferente de todos os anteriores

3. A tecnologia a ser utilizada é nova – não há experiências registradas

4. A ferramenta não oferece facilidades de busca e recuperação

5. Não acho que seja relevante para o projeto

20) Você registra na base de soluções as decisões de projetos e aspectos que considera importantes para outros projetos?

1. Sempre
2. Algumas vezes (responder a pergunta 21)
3. Nunca (responder a pergunta 21)

21) Qual a razão para você não registrar aspectos importantes do projeto na base de soluções? (marque todas que julgar adequadas)

1. Acho que não é relevante para o processo e qualidade do produto
2. Não acho que outras pessoas irão consultar esta base
3. Não tenho tempo, os prazos de entrega são muito curtos
4. A ferramenta não é adequada para registro e pesquisa

22) Nos projetos que você participou foi realizada reunião de avaliação final do projeto?

1. Sim
2. Não

23) Se ela ocorreu, houve registro de quais tipos de informação?

1. Relação de decisões tomadas
 2. Pontos fortes do projeto
 3. Pontos fracos (oportunidades de melhoria)
 4. Erros cometidos, e como foram corrigidos
 5. Outros (especifique abaixo)
-

Bloco IV – Aspectos Gerais

24) Você conhece o conceito de gestão da informação?

1. Sim
2. Não

25) Você acredita que o reuso de informações é importante para a melhoria da qualidade de projetos de *software*?

1. Sim
2. Não

26) Em sua opinião, em quais aspectos o reuso de informações pode auxiliar projetos de *software*? (marque todas que julgar adequado)

1. Redução do tempo de projeto
2. Facilitar o tratamento dos requisitos-não-funcionais
3. Evitar erros já cometidos em projetos anteriores
4. Melhoria na qualidade geral do produto

27) Em sua opinião, o Serpro considera que o reuso de informações é importante para os projetos de *software*? (marque todas que julgar adequado)

1. Sim, mas não oferece ferramentas adequadas
2. Sim, mas o processo não oferece suporte ao reuso
3. Sim, mas não oferece incentivo aos empregados
4. Sim, mas não há uma estratégia geral para implantação do conceito
5. Não, a empresa considera que o assunto não é relevante

BLOCO V – Informações sobre os participantes

28) Formação Superior – informar qual sua graduação

29) Informar se possui Pós-Graduação, Mestrado ou Doutorado

30) Você ocupa cargo gerencial?

1. Sim

2. Não

31) Qual seu tempo de experiência trabalhando com requisitos?

APÊNDICE B - PROPOSTA DE MODELO DE GESTÃO DA INFORMAÇÃO

B.1. Modelos e Informação

O Novo Dicionário Aurélio da Língua Portuguesa define modelo como “[...] objeto destinado a ser reproduzido por imitação; representação, em pequena escala, de algo que se quer executar em grande”.

Michaud (2006, p. 212) define que modelagem é imitação ou reprodução de um modelo, de uma realidade específica que se deseja modificar, ou sobre a qual se deseja intervir.

Um outro conceito é o definido por Sayão (2001, p. 83):

Modelo é uma criação cultural destinada a representar uma realidade, ou alguns de seus aspectos, a fim de torná-los descritíveis qualitativa e quantitativamente e, algumas vezes, observáveis. ... são representações simplificadas e inteligíveis do mundo, que permitem vislumbrar características essenciais de um domínio ou campo de estudo.

A proposição de um modelo de gestão da informação para o processo de desenvolvimento de *software* do Serpro é justificada pela afirmativa de Michaud (2006, p. 236):

Os modelos são também poderosas ferramentas, desempenhando importante papel como recurso metodológico para todas as áreas relacionadas diretamente ou indiretamente com dados, informação e conhecimento. Aplicados às organizações, eles facilitam a transformação de parte do conhecimento tácito em conhecimento (informação) explícito e favorecem a compreensão, comunicação e gestão das empresas.

Tarapanoff (2006), mostrou que a informação é fator determinante para a melhoria de produtos, processos e serviços, tendo valor estratégico em organizações, razão pela qual deve ser gerenciada. A gestão dos recursos informacionais gera resultados positivos em relação à eficiência operacional.

Por outro lado, os estudos em engenharia de *software* mostram que a evolução da qualidade dos sistemas de informação está relacionada ao uso de experiências e informações de projetos anteriores.

Assim, a proposta deste trabalho – a construção de um modelo que integre o conceito de ciclo de gestão da informação ao processo de desenvolvimento de *software* do Serpro – pretende acrescentar uma nova visão para o processo, na macroatividade de requisitos. A proposta de integração tem como objetivo incentivar e tornar mais eficientes os processos associados ao ciclo de gestão da informação, e melhorar a qualidade dos produtos de *software* desenvolvidos pela empresa.

B.2. Modelo para Integração da Gestão da Informação e o PSDS

B.2.1. Visão Geral do Modelo

O modelo de gestão da informação para integração com o PSDS é baseado no projeto de fábrica de experiências, da Universidade de Maryland, e no ciclo da gestão da informação proposto por Choo. Ele foi adaptado considerando os relatos das experiências da NASA, Ericsson e da DaimlerChrysler, além de análises destes modelos feitas por outros autores. Cabe destacar que a proposta adotada na Ericsson é que mais se aproxima do cenário do Serpro, razão pela qual algumas idéias citadas por Joahanson, Hall e Coquard (*op. cit.*) serão aproveitadas.

O modelo é baseado aspectos da realidade do Serpro, como o modelo de desenvolvimento de *software*, a percepção da realidade da atividade de requisitos, levantada pela aplicação de questionários aos componentes da equipe de requisitos de Brasília, e características organizacionais, que de alguma forma afetam o processo de desenvolvimento de sistemas. A Figura 10 resume a origem das idéias usadas para a proposta.

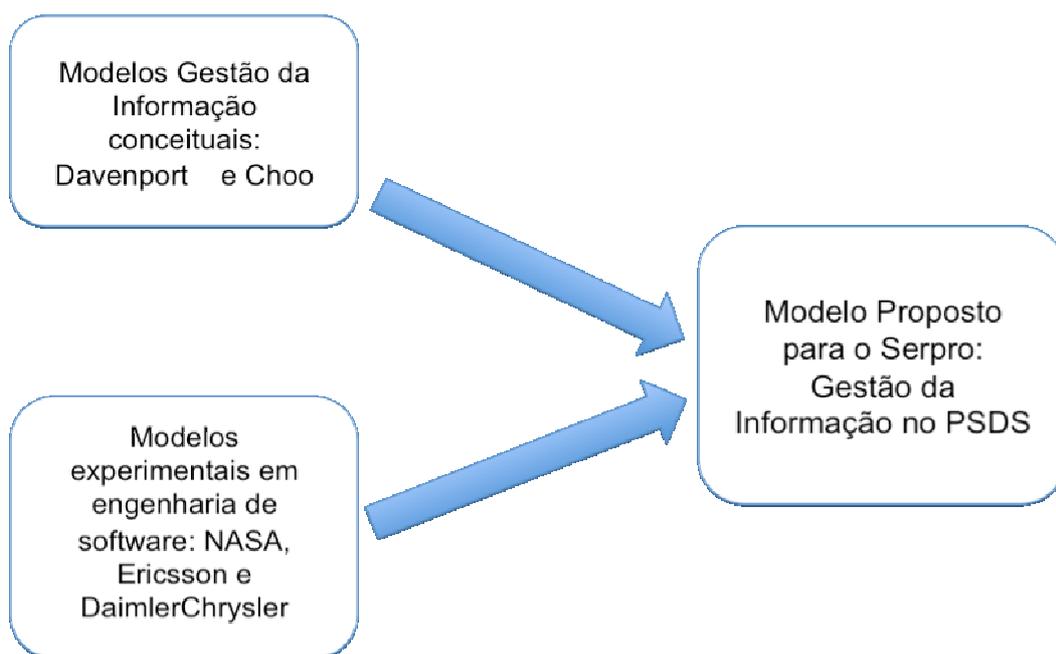


Figura 10 – Esquema – modelo de Informação para o PSDS

Fonte: elaborado pelo autor, 2009.

O modelo, representado na Figura 11, é composto de três pontos principais: objetivos, premissas e pontos chave. Ele é complementado pelos aspectos considerados mais relevantes para sua descrição: atividades, artefatos, papéis e fluxos de informação.

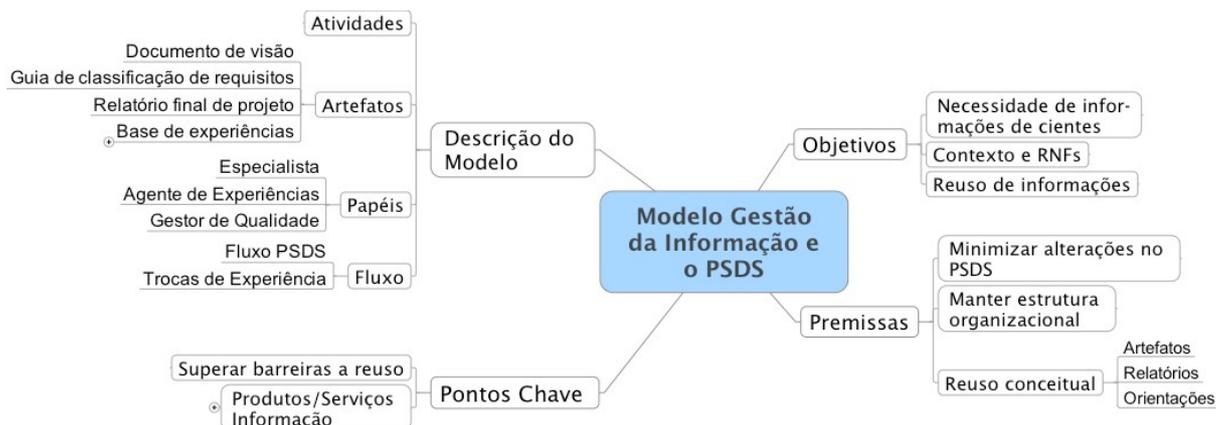


Figura 11 – Representação do Modelo

Fonte: elaborada pelo autor (2009)

Os objetivos básicos a que o modelo deve atender estão associados aos pressupostos, comprovados pelos dados coletados junto a equipe de requisitos:

- Atender necessidade de informações relacionadas aos clientes, sua organização e o domínio de negócio a ser atendido pelo sistema. Estas informações são relevantes para a atividade de elicitação de requisitos;
- Auxiliar a identificação do contexto dos projetos de *software*, para facilitar tarefa de registro e reuso de experiências. Uma das formas para replicação de contextos é a identificação do domínio e dos requisitos não-funcionais;
- Oferecer condições para reuso de informações, com registro de experiências e ambiente que facilite sua recuperação e utilização.

O modelo trabalha ainda com as seguintes premissas:

- Alterar o mínimo possível o PSDS, aproveitando definições já existentes de papéis, artefatos, atividades e subatividades. O objetivo, assim como no caso da Ericsson, é incluir conceitos e práticas da gestão da informação ao processo, mas sem exigir esforços significativos em treinamento e revisão de procedimentos;
- Manter a estrutura organizacional. A experiência da NASA mostrou a viabilidade de criar uma estrutura em paralelo, apenas para coleta de indicadores e empacotamento de experiências, mas com esforço expressivo da equipe e geração de custos adicionais. O esforço necessário para proposição e implantação de nova estrutura organizacional para coleta de indicadores e análise de oportunidades de reuso seria alto, e não é compatível com objetivo do estudo. Cabe ainda considerar que as equipes de desenvolvimento de sistemas do Serpro passaram por três alterações de estrutura nos últimos quatro anos, e uma nova proposição de mudança dificultaria as ações de adoção do modelo proposto.

- Trabalhar com foco em reuso conceitual. A macroatividade de requisitos trata de informações registradas em documentos – artefatos de requisitos, relatórios de reunião e orientações – razão pela qual a abordagem de reuso de informações a ser aplicada neste estudo é pertinente, na categoria de reuso conceitual definida por Glass. Embora alguns conceitos gerais sejam aplicáveis, o trabalho não tratará os aspectos envolvidos em reuso de código ou componentes de *software*;

Devem ainda ser considerados dois pontos chave para a implantação do modelo: barreiras para o reuso de informações e a disponibilidade de produtos e serviços de informação.

B.2.1.1. Barreiras ao reuso de informações

A revisão de literatura – item 3.5.2 e Tabela 1 – apontou um conjunto de barreiras ao reuso de informações em projetos de *software*, relacionadas a aspectos técnicos, culturais e organizacionais que, no ambiente do Serpro, foram confirmadas pela coleta de dados. É importante que estas barreiras sejam tratadas para facilitar a implementação de estratégias de reuso de informações para as equipes de desenvolvimento. As principais barreiras identificadas são descritas abaixo.

Dificuldade na captura de conhecimento – projetos de *software* são dependentes de contexto e de tecnologias, razão pela qual nem sempre é fácil o registro de experiências que possam ser reutilizadas em outros projetos. As Questões 17 e 20 confirmam esta barreira no Serpro, em que cerca de 85% dos analistas consultados afirmaram que nunca registram decisões ou outros aspectos que julgam relevantes para outros projetos;

Meio/ferramenta para troca de informações – este item é citado por Desouza (2003) como um dos principais obstáculos ao reuso de informações, já que a maior parte das ferramentas e ambientes em uso nas organizações não oferece uma boa experiência aos usuários. Ele é comprovado no caso do Serpro, nas Questões 19/4 (item 4), 21/4 e 27/1, em que as ferramentas são mencionadas como um obstáculo ao registro e recuperação de experiências;

Pressão de prazos – a área de desenvolvimento de *software* tem um histórico de prazos elevados e não cumpridos para a entrega de seus produtos, situação que se agrava na medida em que os sistemas têm papel cada vez mais relevante para os negócios. Rus et. Al. (2001) explicam que as equipes não têm tempo para avaliação e registro de suas experiências. As respostas obtidas na Questão 21/3 mostram que 68,8 % dos analistas apontam os prazos curtos como razão para o não registro de experiências;

Aspectos culturais – são apontados por Kim e Stohr (1994) como um problema difícil de ser superado, já que as organizações de TI são orientadas a tecnologia e confundem inovação com mudanças de tecnologia. Também não há treinamento e incentivo ao reuso de informações. Os dados obtidos no Serpro confirmam este cenário, nas Questões 18, 19/1 e 21/2 – as pessoas não registram informações por achar que os demais não irão utilizá-las;

Aspectos organizacionais – Kim e Stohr (*op. cit.*) apontam a falta de comprometimento, suporte organizacional e a dificuldade de medir benefícios como razões para as organizações de desenvolvimento de *software* não obterem sucesso em projetos de reuso. A Questão 21, itens 3 e 4, confirma que isto também ocorre no Serpro.

B.2.1.2. Produtos e Serviços de Informação

A disponibilidade de produtos e serviços de informação, que possibilitem a agregação de valor às atividades dos usuários, é apontada por Choo (2006) e Taylor (1986) como um item fundamental para o sucesso no uso da informação. Na atividade de desenvolvimento de *software* os produtos e serviços estão associados a ferramentas de *software* utilizadas durante os projetos.

No projeto da Ericsson, houve um grande destaque inicial para a base de dados criada para armazenar as experiências, para descobrir-se mais a frente que ela não gerava os resultados esperados, sendo abandonada por outra abordagem, que considerava o fluxo de informações e a interação entre as pessoas.

Os dados coletados no Serpro mostram que o PSDS obteve sucesso no uso dos artefatos como meio principal de registro de experiências e informações sobre as diversas etapas do projeto. As Questões 6/1, 10/2, 14 e 15 – mostram que os analistas utilizam os artefatos, e até os consideram como produtos e serviços de informação disponíveis na execução de seus projetos.

Entretanto, as próprias limitações apontadas por eles – respostas as Questões 19, 21 e 27 – apontam para um cenário em que o acesso aos artefatos possui limitações, que dificultam seu uso em outros projetos. Os artefatos e as ferramentas devem passar por revisão, que possibilite atender as características apontadas por Taylor como relevantes para adicionar valor à informação: facilidade de uso, qualidade, adaptabilidade e redução de custo/tempo.

No Serpro, embora as interações pessoais entre as equipes sejam uma forma importante de troca de informações e experiências, outras características organizacionais devem ser consideradas: distribuição das equipes de desenvolvimento, a rotatividade de pessoas nos clientes, a exigência de manutenção dos sistemas por muitos anos, e a proximidade de ciclo grande aposentadoria.

Neste cenários é recomendável uma estratégia consistentes de codificação e armazenamento de informações sobre o conhecimento dos negócios dos clientes, razão pela qual é fundamental a existência de produtos e serviços de informação que possuam as características mostradas anteriormente, e que atendam às necessidades das equipes de desenvolvimento.

B.2.2. O Modelo – Descrição

O modelo para a atividade de requisitos do processo do Serpro considera alguns pontos relevantes dos projetos de FE avaliados, e do processo de desenvolvimento do Serpro. Em relação aos projetos de FE, quatro aspectos serão aproveitados:

- Experiências passadas – uso de experiências de projetos passados, seja na forma de requisitos não-funcionais, seja na forma de práticas de sucesso nas áreas de engenharia de *software* e gestão de equipes. Estas experiências devem ser empacotadas, para seu armazenamento e organização como produtos ou serviços de informação;
- Base de experiências – será utilizada uma base de experiências, em substituição à BS já prevista hoje pelo PSDS. O impacto não deve ser significativo, já que ela não é usada de forma sistemática pelas equipes;
- Fluxo de informações – estímulo do fluxo de informações e experiências entre as diversas equipes, clientes e outras áreas da organização. Devem ser criados papéis voltados para a distribuição e compartilhamento de informação, assim como no modelo da Ericsson;
- Abordagem para uso do conhecimento – o PSDS trata prioritariamente ações e procedimentos voltados para a codificação de conhecimentos em artefatos de projeto. O modelo proposto na pesquisa, assim como as implementações utilizadas como referência, prevê o uso das abordagens de codificação e personalização, conforme definido por Hansen, Nohria e Tierney (*op. cit.*). Entretanto, deve ser considerado traço cultural importante na empresa, que é o compartilhamento de experiências, informações e conhecimento em interações pessoais, conforme demonstrado por Carvalho e Filho (*op. cit.*). Este modelo de relacionamento pessoal é muito importante, e será utilizado na proposta, como forma de melhorar o fluxo e o compartilhamento de informações.

Embora uma das premissas básicas do modelo fale em alterações mínimas no processo de desenvolvimento, algumas mudanças devem ser feitas para atender as necessidades de gestão da informação. Em relação a estrutura do PSDS, os seguintes pontos serão considerados:

B.2.2.1. Atividades

A macroatividade de requisitos é formada por seis atividades (ver Figura 7), que por sua vez são formadas por subatividades. O modelo não prevê alterações nas atividades atuais do PSDS, mas sugere a criação de uma nova subatividade, chamada “Identificar Informações do Cliente”, a ser incluída na atividade “Elicitar Requisitos”. Seu objetivo é a identificação de informações do cliente e do domínio da aplicação a ser desenvolvida. As informações capturadas devem ser armazenadas no artefato Documento de Visão.

Ela deve considerar a ampliação de fontes, externas e internas, para obtenção de informações. Como principal fonte externa adicional deve ser incluída a Internet, e dentre as fontes internas as citadas nas respostas aos

questionários, como colegas das equipes, analistas de negócio dos clientes e informações de projetos anteriores.

B.2.2.2. Artefatos

O modelo prevê alterações em três artefatos, a substituição de um artefato já existente e a emissão de uma orientação técnica. Os três artefatos a serem alterados são:

Documento de Visão

Inclusão de item para a coleta de informações sobre os clientes – “Informações sobre Cliente” – onde serão descritas informações sobre o cliente e definição do domínio do sistema em desenvolvimento.

Guia de Classificação de Requisitos

O artefato deve descrever os padrões mínimos de RNFs a serem listados para os sistemas, bem como valores de referência que podem ser propostos, com base no domínio do sistema, ou de outros sistemas do mesmo cliente. Este artefato deve servir como base para uma proposta de requisitos não-funcionais para cada sistema.

Relatório Final do Projeto

Este artefato, embora não faça parte da macroatividade de requisitos, será considerado, já que ele pode tratar de aspectos relacionados a todas as macroatividades de um projeto, e por sua importância para a gestão da informação em projetos de *software*, conforme destacado por Birk, Dingsoyr e Stalhane (2002). O artefato deve descrever os pontos positivos e negativos, observados pela equipe na reunião final de avaliação do projeto, e considerados relevantes para registro como informações para novos projetos. A recomendação é que todos os pontos identificados neste relatório sejam automaticamente registrados na Base de Experiências (BE).

Base de Experiências

O modelo prevê a criação de uma base de experiências (BE). É sugerido que ela substitua a base de soluções – pouco usada desde que o PSDS foi criado.

Uma das justificativas de mudança do nome deste artefato é que ele não é relevante para as equipes de desenvolvimento. O nome “Base de Soluções” já é utilizado no processo e na ferramenta de gestão do atendimento a clientes. Ela é usada pela central de atendimento do Serpro, para registro das soluções de atendimento aos chamados dos clientes.

O termo “experiências” tem abordagem mais ampla, e significa o registro de experiências boas e ruins, mas que sejam relevantes para projetos no futuro. Em projetos de *software*, é recomendável que as decisões ou ações que

resultaram em erro também sejam registradas. Os erros são oportunidades ricas de aprendizado, e seu registro é tão importante quanto o dos acertos.

A base de experiências terá o seguinte processo de construção:

- Relato de experiências – qualquer pessoa envolvida no projeto pode relatar uma experiência que considere útil de registro. Elas podem ser registradas a qualquer momento do projeto, e serão incluídas na BE com estado de “Em análise”. O responsável pelo cadastramento pode sugerir em quais categorias ela deve ser enquadrada, mas as sugestões terão que ser aprovadas pelo agente de experiências (um novo papel no processo);
- Aprovação de experiências – as experiências registradas base, em estado inicial “Em análise”, serão avaliadas por um agente de experiências, para verificar se atendem os requisitos mínimos necessários para constar da base, e classificá-las de acordo com a macroatividade do PSDS e domínio do sistema. Caso aprovadas elas migram de estado para “Aprovada”. Caso não atenda aos requisitos estabelecidos ela pode ser devolvida para alterações, ou simplesmente rejeitada. Todas as experiências permanecerão registradas na base e poderão ser consultadas por todas as equipes de desenvolvimento de sistemas da empresa, não importando seu estado.
- Formato das experiências – é razoável estabelecer um formato padrão mínimo para o relato das experiências, para permitir seu aproveitamento no futuro. Este formato não será proposto neste estudo;
- Organização das experiências – as experiências devem ser organizadas de acordo com a macroatividade do PSDS, bem como o domínio da aplicação. Esta classificação tem como principal objetivo facilitar a recuperação por um mecanismo de busca, para atender necessidades das equipes na diversas fases do projeto;
- Categorias de experiências – Dingsoyr (2000) observa que empacotar uma experiência permite a sua generalização, formalização e customização, para que seja aproveitada em outro projeto. Este estudo não proporá um modelo para empacotamento, mas propõe que ele seja iniciado de forma simples, e refinado posteriormente. O objetivo é que a coleta de experiências seja iniciada de forma rápida, para avaliação e evolução do modelo ao longo de sua implantação. O estudo propõe que os pacotes de experiências sejam categorizados como proposto por Johansson, Hall e Coaquard (1999):
 - Experiências de gestão – informações de referência para gerentes de projeto, tratando aspectos diversos de gestão, em atividades administrativas, equipe e relacionamento com outras áreas da empresa.
 - Experiências técnicas – informações sobre os aspectos técnicos relacionados ao projeto. Caso necessário podem ser divididas em subcategorias de requisitos (requisitos das aplicações, incluindo os requisitos não-funcionais); ferramentas, testes e processo de *software*. Podem descrever boas práticas com ferramentas, experiências de

aspectos técnicos do projeto, oportunidades de reuso e lições aprendidas;

- Negócios – informações relacionadas aos clientes, seus domínios de negócio, bem como sobre processos de negociação em assuntos técnicos e comerciais. Estas experiências podem ser relevantes para as equipes comerciais.

As categorias de pacotes de experiências podem ser ampliadas, desde que a equipe de gestão do PSDS entenda que isto contribui para atender as necessidades de registro de tópicos considerados relevantes.

Orientação Técnica – Gestão da Informação

O PSDS utiliza um modelo de artefato denominado “Orientação Técnica” para tratar assuntos de interesse geral, e que devem ser considerados durante os projetos. O modelo propõe a emissão de uma orientação com o título – Gestão da Informação e o PSDS – que descreva o ciclo de gestão da Informação e descreva suas principais interfaces com o PSDS. A orientação deve explorar todos os processos do ciclo, com destaque para a identificação, aquisição e uso da informação. Ele deve descrever ainda os produtos e serviços de informação disponíveis para uso das equipes.

B.2.2.3. Papéis

Serão aproveitados, sempre que possível, papéis já definidos pelo PSDS, ou já implantados no desenvolvimento de sistemas do Serpro. A implantação do PSDS criou papéis para as equipes de desenvolvimento, que já estão incorporados às rotinas de trabalho, como gestor de qualidade do processo e gestão de configuração. O modelo prevê o aproveitamento destes papéis, com a inclusão de novas atividades em suas atribuições atuais. Esta alteração não será significativa, e deve produzir resultados melhores do que a introdução de novos papéis, já que uma das críticas que o PSDS recebe é de alocar muitas pessoas para tarefas de gestão do processo, em detrimento a alocação de mais pessoas para o desenvolvimento de *software*.

O papel selecionado para atuação direta no modelo é o de especialista técnico, já previsto no PSDS hoje. Outros papéis, como líder de projeto, gestor de qualidade e de configuração de *software*, terão participação indireta, em suas atividades já previstas no PSDS.

Cabe aqui destacar a importância do líder de projeto para o funcionamento do modelo, já que sua atitude junto a equipe será determinante para mostrar o comprometimento da organização com o processo, em especial na elaboração de estratégias de compensação de prazos, para permitir o alívio da pressão de prazos mencionada nos questionários, sem gerar problemas com as necessidades dos clientes. Os líderes de projeto, por seu contato diário com a equipe, têm real possibilidade de incentivar a prática de reuso com suas orientações, ou então inibi-la, razão pela qual podem ser considerados como agentes de reuso

É proposta a criação de um novo papel – o agente de experiências. As atividades previstas para estes papéis são descritas a seguir.

Especialista técnico

O papel de especialista técnico já existe hoje, e atua em questões específicas relacionadas ao processo de desenvolvimento (PSDS). Eles atuam na avaliação de pedidos de suporte e análise de pedidos de alteração do processo. São formalmente designados para esta função, com compromisso de dedicar parte de suas horas de trabalho a atividade. No restante do tempo eles atuam em projetos de *software*.

A proposta do modelo é que este papel seja ampliado para outras áreas, além do processo de *software*, como arquitetura de *software*, ferramentas de desenvolvimento e testes. Uma possível forma de organização dos especialistas seria:

- domínio de negócio – especialistas em aspectos de contextos e domínios de negócio dos principais clientes do Serpro, como administração tributária e financeira, comércio exterior, dentre outros. Este grupo incluiria especialistas em requisitos e arquitetura de *software* para estes domínios;
- tecnologias – especialistas em tecnologias e ferramentas utilizadas pelo Serpro, com responsabilidade de atuar em ações de consultoria e suporte ao seu uso para projetos de desenvolvimento. Este grupo poderia ter uma divisão adicional, por grupos de tecnologias, como gerenciadores de bases de dados, servidores de aplicação, linguagens de desenvolvimento e ferramentas;
- processo – especialistas em atividades do PSDS, e responsáveis por atender demandas das equipes de desenvolvimento relacionadas ao uso do processo.

Sua atuação será semelhante a dos *experience communicators*, presente no projeto da Ericsson, e devem ter perfil semelhante ao descrito em 3.5.6, profissionais de reconhecida competência técnica em suas áreas de atuação, e com desejo de compartilhar seu conhecimento ao participar em atividades de mentoria, sempre que solicitado. É importante que eles não sejam vistos como parte da equipe do projeto, e sim como apoio para algumas atividades, e que atuarão apenas por um tempo determinado, até que a própria equipe possa dar sequência nas atividades.

Este aspecto é importante para garantir a circulação dos especialistas entre diversos projetos, permitir a aquisição de novas experiências e permitir que as equipes obtenham novas informações e conhecimento.

Eles não devem ser formalmente designados para a função, mas reconhecidos pela organização, sendo recomendado a criação de um serviço de informação que permita sua identificação pelos agentes de experiência e pelos líderes de projeto. Este serviço de identificação e localização é importante para empresas com características similares a do Serpro, com muitas equipes dispersas no país.

Agente de Experiências (AE)

Este papel é semelhante ao *experience broker*, descrito por Johansson, Hall e Coaquard (*op. cit.*), sendo responsável por facilitar o registro de experiências e sua troca entre as equipes. Seu perfil é de um generalista com experiência em tipos diversos de projetos, de forma que ele conheça um pouco de muitos assuntos, conheça a estrutura da organização e seus processos de desenvolvimento e produção de serviços.

Deve ser uma pessoa com bom trânsito em diversas áreas, que tenha habilidade para ouvir e aprender e identificar pontos críticos em projetos, e como eles podem ser solucionados. As pessoas devem ser formalmente designadas para este papel, mas é fundamental que elas tenham reconhecimento da estrutura da organização, caso contrário não serão procuradas para auxiliar as equipes no compartilhamento de informações e experiências.

O agente de experiências será responsável pelas seguintes tarefas:

- Identificar especialistas e incentivar seu cadastramento na base de especialistas;
- Facilitar o contato das equipes com os especialistas técnicos;
- Incentivar o registro de experiências na base de experiências;
- Aprovar as experiências registradas na BE;
- Participar das reuniões de encerramento de projetos, como forma de conhecer os pontos fortes e fracos de cada equipe e projeto encerrado.

Devem ser estabelecidas regras mínimas para a alocação das pessoas a este papel. Como regra geral, elas não devem ter posição gerencial, e podem ser alocadas em tempo parcial ou tempo integral. Esta posição tem dependência do perfil de cada pessoa, e algumas podem ficar mais confortáveis atuando apenas nesta posição, enquanto outros podem render mais dividindo o tempo como AE e atuando em projetos.

Também é possível ter pessoas que atuam como AE por um ou dois anos, e depois retornam para equipes de projeto, enquanto outros podem permanecer na posição por mais tempo. Uma sugestão para a implantação inicial é usar todas as possibilidades citadas acima, e fazer uma avaliação após 12 meses, para verificar qual a mais adequada. A sugestão inicial é que cada Pólo de Desenvolvimento do Serpro tenha dois agentes de experiências. O ponto fundamental para o sucesso da iniciativa é que a organização tenha confiança e comprometimento com o papel, e com as pessoas para ele designadas.

B.2.2.4. Fluxos de Informações

O fluxo de informações é crítico em projetos de desenvolvimento de sistemas, pela complexidade do processo e envolvimento de pessoas com especialidades diversas, além dos clientes, responsáveis pela definição dos

requisitos e homologação final do serviço antes de sua implantação. O modelo destaca dois fluxos mais relevantes.

Fluxo de Informações no PSDS

O PSDS prevê que informações geradas ao longo do projeto sejam registradas em artefatos, e utilizadas em atividades e subatividades específicas, com a descrição dos artefatos de entrada e saída para cada caso. Este fator é relevante para a implantação de um modelo de gestão da informação, aliado ao fato de que os analistas de requisitos percebem os artefatos como produtos de informação, e o citam como forma preferencial de registro de informações dos projetos de *software*.

Entretanto, o processo não descreve aspectos relacionados ao compartilhamento de informações, em especial as relacionadas a interações pessoais, que são consideradas pelas equipes de desenvolvimento como a principal forma de troca de informações, conforme dados obtidos com os questionários e relatados por Carvalho e Filho (2008).

Sugere-se assim que o processo seja adaptado para prever e incentivar esta modalidade de troca de informações e experiências, sendo avaliado posteriormente a necessidade de implementação de ações de incentivo ao registro na BE. Caso o processo não seja alterado, é recomendável que este tipo de ação seja tratado nos programas de treinamento, possivelmente com palestras que mostrem a importância do compartilhamento de informações entre equipes distintas para o sucesso dos projetos.

Troca de Experiências

A transferência de experiências é uma atividade importante para organizações de *software* como o Serpro, e o modelo propõe a utilização de dois mecanismos para desenvolver valores comuns e facilitar sua percepção pelas equipes de desenvolvimento de sistemas.

O primeiro deles trata de dúvidas específicas sobre algum assunto, e que normalmente pode ser solucionada em contatos rápidos, de poucos minutos. Eles ocorrem em comunicações informais, contatos telefônicos ou em encontros casuais no café ou durante o almoço. Os agentes de experiências devem estimular estas interações, circulando entre as equipes em busca de novas informações, e atuando na identificação de potenciais especialistas nas equipes.

Outra forma tradicional para troca de experiências ocorre em eventos planejados, como reuniões, workshops ou conferências, com duração entre duas ou três horas e alguns dias. Esta modalidade de troca de experiências é denominada Situação de Aprendizagem, sendo comum quando pessoas com treinamento formal, mas falta de experiência profissional, necessitam auxílio para completar tarefas de seus projetos.

Ela deve ser realizada em situações reais de projetos, conhecidas como mentoria ou *on-the-job-training*, sendo considerada uma das formas mais eficazes de transferência de informações e experiências. Na proposta para o Serpro, as solicitações deste tipo de evento devem ser feitas aos agentes de

experiências, que serão responsáveis por sua coordenação e coleta de resultados, possíveis candidatos a inclusão na base de experiências.

B.2.2.5. Estratégia de Implantação

A implantação do modelo proposto deve prever uma estratégia de integração com o PSDS, já que haverá alteração e inclusão de atividades no processo já estabelecido. A integração deve seguir o procedimento já existente para alterações no processo – Pedido de Melhoria no Processo (PMP) – e seguir com os demais pontos em paralelo, já que os prazos estipulados para esta ação podem ser aproveitados para ações relacionadas a divulgação e treinamento das pessoas.

Na implantação devem ser considerados dois pontos importantes, destacados na literatura e nos dados coletados: a definição de uma estratégia para implementação do modelo de gestão e reuso de informações; e evolução da ferramenta. Assim, as seguintes ações devem ser consideradas para a implantação deste modelo:

- Definir uma estratégia geral para o projeto e divulgá-la para a empresa. O objetivo desta ação é mostrar o comprometimento de todos com o projeto. Pode ser avaliada uma experiência piloto, com a escolha de um ou dois Pólos ou de um domínio de negócios. Como o PSDS possui uma agenda de alterações já previstas, com duas versões por ano, um piloto permitiria o início da experiência, mesmo antes da alteração no PSDS;
- Definir indicadores que permitam a avaliação do benefício que as ações podem trazer para o resultado final – considerar os benefícios previstos pelos empregados na coleta de dados (Questões 25 e 26);
- Estabelecer um projeto piloto em dois ou três Pólos de desenvolvimento, para avaliar o impacto das mudanças nos projetos em andamento. Estes Pólos devem designar os agentes de experiências e já implantar as alterações previstas no processo;
- Estabelecer um modelo de incentivo aos empregados para as atividades relacionadas ao compartilhamento de experiências e reuso de informações para desenvolvimento de *software*;
- Definir ferramenta(s) que trate(m) os aspectos relacionados como limitantes na coleta de dados: facilidades de registro, recuperação e acesso às informações registradas nos artefatos e na base de experiências. Em função da dificuldade que algumas pessoas tem na codificação de conhecimentos e experiências em forma textual, e também do tempo para esta atividade, é sugerido que esta ferramenta utiliza facilidades para registro e recuperação de arquivos de vídeo e voz. Isto facilitaria o registro de experiências, evitando a necessidade de transcrição em texto, e reduzindo o tempo para a tarefa. Adicionalmente, tornaria a base de experiências compatível com serviços de Internet já existentes hoje, como portais de vídeos e voz, e usados de forma intensiva pela maior parte dos desenvolvedores.

B.2.3. O Modelo e o Ciclo de Gestão da Informação

A seguir os principais pontos da proposta são descritos, considerando os processos do ciclo de gestão da informação de Choo, conforme Figura 12.

Necessidades de Informação

Este processo deve considerar as etapas previstas por Le Coadic, e complementar itens não atendidos hoje pelo PSDS. O ponto mais relevante é ter informações detalhadas sobre o cliente de cada sistema, considerando os seguintes aspectos: descrição do cliente, pessoas importantes para o trabalho, outros sistemas já existentes, ambientes operacionais em uso e outras experiências do cliente com o Serpro, caso existam. O objetivo é obter o máximo de informações possível sobre o cliente, para que ela seja usada como subsídio durante a aquisição de informações.

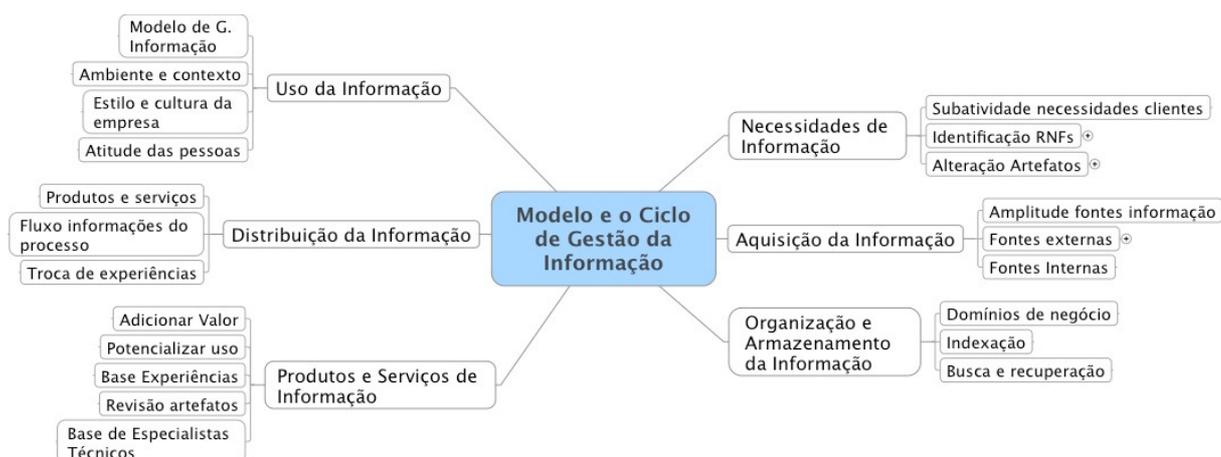


Figura 12 – Modelo e o Ciclo de Gestão da Informação

Fonte: elaborada pelo autor (2009)

A expectativa é que as alterações propostas nos artefatos, e a proposição da subatividade “Identificar Informações do Cliente”, atenda as necessidades de informação registradas na literatura e confirmadas nos dados coletados junto a equipe de requisitos.

Em termos de necessidades, deve ser identificado qual o domínio de negócio da aplicação, e quais os requisitos não-funcionais devem ser usados como base para este domínio. Os RNFs de domínios específicos já devem dispor de valores de referência a serem propostos para o cliente, com base na capacidade do ambiente operacional do Serpro, e no histórico de contratações do cliente. A utilização deste conjunto de requisitos e dos valores propostos

de dependerá de negociação com cliente, com base nas necessidades específicas de cada aplicação ou ambiente.

Aquisição da Informação

O processo de aquisição da informação para a área de requisitos confunde-se com a própria atividade, já que sua principal função é a aquisição de informações e conhecimento junto ao cliente para o entendimento do domínio da aplicação e identificação dos requisitos funcionais e não-funcionais.

Ele deve atender a dois pontos principais – a amplitude nas fontes de informação e a seleção de mensagens. Em relação às fontes de informação, devem ser consideradas as informações relativas ao cliente, sua organização e o domínio da aplicação, identificadas no processo anterior.

O modelo, com base nas respostas dos questionários, propõe a formalização das fontes externas e internas citadas pelos analistas, como a Internet, opinião de colegas de equipe e informações já existentes em artefatos de outros projetos.

Organização e Armazenamento da Informação

A definição dos domínios de negócio, ou áreas de conhecimento, das aplicações do Serpro é a principal ação a ser desenvolvida para a organização da informação. Os domínios definidos serão a principal forma de organização das informações, permitindo que todos os sistemas existentes hoje, mais de 600 sistemas, ou pelo menos os clientes, sejam agrupados por assuntos ou interesses comuns.

O Anexo B apresenta uma proposta de classificação de domínios de negócios para o Serpro, e que poderia ser utilizada como base para os projetos novos. As informações relacionadas aos projetos seriam organizadas por domínio de aplicação, o que facilitará os aspectos de organização, busca e recuperação, bem como a identificação e geração de experiências pelos analistas.

Outra deficiência do PSDS apontada pelos dados coletados refere-se a ausência de procedimentos para indexação e armazenamento, o que dificulta a recuperação da informação. Este é um ponto complexo, uma vez que envolve a substituição das ferramentas e dos procedimentos adotados hoje.

Entretanto, é sugerido que seja iniciado um trabalho de reavaliação desta infraestrutura, com base em critérios voltados para organização, armazenamento, busca e recuperação de informações. Para esta tarefa, cabe destacar que, além das pessoas alocadas aos Centros de Documentação e Informação (denominação atual das bibliotecas) a empresa dispõe hoje de cerca de 10 empregados com formação (mestrado ou doutorado) em ciência da informação, e que poderiam ser utilizados nesta tarefa.

Produtos e Serviços de Informação

O modelo prevê a criação de produtos e serviços que aumentem o potencial de uso das informações, e adicionem valor à informação processada, permitindo que os usuários tomem decisões melhores. As decisões de um projeto de desenvolvimento de *software* são relacionadas a técnicas de levantamento de requisitos, forma de negociação com os clientes, modelo de organização da equipe e seleção de arquitetura e tecnologias.

As informações devem estar disponíveis para todas as pessoas envolvidas nos projetos, em níveis definidos de qualidade e facilidades de acesso e uso. Os produtos e serviços devem ser organizados e construídos para atender as necessidades dos usuários.

O modelo prevê dois serviços principais de informação para as equipes de desenvolvimento do Serpro: a base de experiências (BE) e o conjunto de artefatos de requisitos do PSDS. Dentre os artefatos relevantes para o levantamento de requisitos podemos destacar o Documento de Visão e o Guia de Classificação de Requisitos, que permitirão a identificação de informações sobre o cliente, seu domínio de negócios e requisitos não-funcionais que podem ser adotados.

Outras informações relevantes para equipes de desenvolvimento, como relação de tecnologias aprovadas para uso, arquiteturas de software recomendadas e mapas de domínios de negócio; também devem ser consideradas como serviços de informação, e organizadas de modo a atender as premissas descritas.

Os produtos e serviços de informação devem ter como premissa o atendimento as principais características definidas por Taylor que adicionam valor a informação, como facilidade de uso, qualidade, adaptabilidade e redução de tempo na obtenção de respostas.

Distribuição da Informação

A disseminação e compartilhamento de informações são tratados pelo modelo com a evolução dos produtos e serviços, e com as ações a serem implementadas pelos especialistas técnicos e agentes de experiências, pois eles terão o papel de circular entre as diversas equipes técnicas e conduzir o processo de transferência de informações, além de fomentar atividades de codificação de conhecimentos.

Os produtos e serviços devem garantir que a informação seja entregue por meios que estejam de acordo com os hábitos e preferências dos usuários, atendendo ao princípio de informação certa para a pessoa certa, no formato, local e hora certos. Por outro lado, o mecanismo de troca de experiências facilitará o fluxo de informações em situações de problema.

Os líderes de projeto tem papel importante no processo, pois são responsáveis por parte do fluxo de informações do PSDS, além da disseminação de informações organizacionais para os demais membros de suas equipes.

Uso da Informação

O uso da informação envolve a sua seleção e processamento, de modo a resolver um problema, entender uma situação ou tomar uma decisão. Taylor (1986) propõe que o contexto de trabalho seja definido como ambiente de uso da informação, já que eles afetam o fluxo e uso das mensagens, e determinam critérios para estabelecimento de valor das mensagens.

A proposta busca influenciar o estilo e cultura da organização, aumentando a percepção da importância da informação, seus fluxos e disponibilidade. A implantação de um modelo de gestão, ainda que restrito ao PSDS, deve afetar a atitude das pessoas em relação a informação. A estrutura projetada procura flexibilizar os elementos do ambiente, adequando-os às necessidades e expectativas expostas pelos analistas de requisitos nas repostas ao questionário.

Este processo deve ser um ponto de atenção para os líderes de equipe e demais papéis envolvidos, já que ele permite a geração de novas informações, conhecimentos e experiências.

APÊNCIDE C – PROPOSTA PARA DOMÍNIOS DE NEGÓCIOS NO SERPRO

O conceito de domínio de aplicações, ou negócio, trata de um conjunto de sistemas de informação, que podem ser agrupados de acordo com características definidas pela organização. Estas características podem ser de funcionalidades, tipo de processo atendido, cliente dos sistemas ou ambientes de operação.

A estruturação por domínios pode ser usada para alocação de equipes e seleção de tecnologias; mas um dos principais usos do conceito é para projetos de reuso de *software*. Segundo Diaz-Herrera¹⁶, o reuso sistemático deve ser planejado, e requer a consolidação do entendimento dos sistemas de um domínio, para explorar pontos comuns e antecipar diversidade, processo conhecido como engenharia de domínio.

Ela prevê esforços para criar ativos de *software* para uso futuro em múltiplos sistemas, o que gera benefícios como redução do tempo de desenvolvimento para novas aplicações e redução de riscos. Entretanto, antes que estes benefícios sejam alcançados, deve ser feito planejamento para que os ativos reusáveis sejam criados com os atributos necessários de qualidade, performance e sem erros.

No caso do Serpro, o atendimento a clientes e o desenvolvimento de sistemas de informação é tradicionalmente realizado associado a um órgão da administração pública federal, que atua como contratante e especificador da solução. Há poucos casos em que o sistema atende a um processo, e onde mais de um órgão é responsável pela especificação do sistema.

Desta forma, uma classificação natural de domínios seria por órgão da administração pública, o que facilita o agrupamento dos sistemas, mas pode gerar duplicações de sistemas ou processos. Um exemplo desta ocorrência é a existência de sistemas similares para diversos órgãos, como protocolo de documentos, solicitação e gestão de viagens e controle de acesso aos prédios.

Apesar disto, será adotada a classificação de domínio por órgão da administração pública, pois ela será facilmente entendida e absorvida pela estrutura da empresa. A situação de problemas similares será tratada com a criação de subdomínios, ou seja, cada domínio definido poderia ter subdomínios separados por assunto ou processo em cada órgão.

Desta forma seriam identificados subdomínios similares, como por exemplo, gestão de treinamento para a Receita Federal do Brasil e para a Secretaria do Tesouro Nacional.

¹⁶ J. L. Diaz-Herrera – Domain Engineering. Disponível em: < <ftp://cs.pitt.edu/chang/handbook/29.pdf> >

Desta forma, é proposta a seguinte definição de domínios para a elaboração deste trabalho:

- SIEF (Sistema de Informações Econômico-Fiscais) - Receita Federal do Brasil 1 (RFB);
- MacroProcesso Tributário - Receita Federal do Brasil (RFB) 2;
- Outros sistemas da Administração Tributária - Receita Federal do Brasil (RFB) 3;
- Comércio Exterior e Transportes (inclui sistemas de gestão de procedimentos aduaneiros e fiscais relacionados a transportes aéreos, marítimos e terrestres);
- Procuradoria Geral da Fazenda Nacional (PGFN);
- Secretaria do Tesouro Nacional (STN ou Administração Financeira);
- Comissão de Valores Mobiliários (CVM);
- Gestão do Ministério da Fazenda;
- Ministério do Planejamento (MP) 1 – Sistemas de Pessoal;
- Ministério do Planejamento (MP) 2 – Sistemas Administrativos;
- Departamento de Polícia Federal (DPF);
- Ministério das Relações Exteriores (MRE);
- Ministério do Trabalho e Emprego (MTbE);
- Infraestrutura de Transportes (inclui sistemas para atendimento ao DNIT e Ministérios dos Transportes);
- Gestão Veículos e Habilitação (sistemas para gestão de veículos automotores e emissão de carteira de habilitação).

A empresa poderá criar quantos domínios julgue adequado para viabilizar sua estratégia de agrupamento de serviços e reuso de informações e experiências. O número de domínios deve ter granularidade adequada, para que auxilie nestes objetivos, lembrando que um número excessivo de domínios pode gerar uma fragmentação excessiva de equipes, artefatos e experiências, o que prejudicaria a estratégia de reuso.