

**FUNDAÇÃO INSTITUTO CAPIXABA DE PESQUISAS EM
CONTABILIDADE, ECONOMIA E FINANÇAS – FUCAPE**

SÉRGIO LUIZ LUGON GRECCO

Manutenção corretiva em sistemas complexos com mão-de-obra terceirizada: Análise de Sistemas de Informação Gerenciais de uma Distribuidora de Energia Elétrica no Espírito Santo

VITÓRIA

2007

SÉRGIO LUIZ LUGON GRECCO

Manutenção corretiva em sistemas complexos com mão-de-obra terceirizada: Análise de Sistemas de Informação Gerenciais de uma Distribuidora de Energia Elétrica no Espírito Santo

Dissertação apresentada à Fundação Instituto Capixaba de Pesquisa em Contabilidade e Finanças – FUCAPE – ES, como requisito final para obtenção do título de Mestre em Ciências Contábeis.

Orientador: Prof. Dr. Annor da Silva Junior
Co-orientadora: Prof^a Dr^a Renata Silva Souza Guizzardi

VITÓRIA

2007

Dedico este trabalho a Deus, à minha esposa, aos meus filhos e aos meus pais.

AGRADECIMENTOS

Agradeço a Deus, por ter me dado forças para vencer mais esta etapa de minha vida e ao Pe. Flávio Tadeu Gomes Moreira pela orientação espiritual.

À Gezella, por ser a minha companheira nestes últimos 21 anos e sempre a maior inspiração para vencer os desafios.

A Brenno e Ryann, razões de viver, deixo a mensagem de agradecimento. E que a perseverança de seu pai, que sempre buscou o conhecimento como forma de crescimento, lhes sirva de modelo.

Aos meus pais, tios e avós, agradeço a educação e os valores morais que me foram ensinados.

As minhas irmãs e ao meu irmão, Helder Luciano de Oliveira, pelo incentivo, apoio, amizade e por apostar em mim neste grande desafio da minha vida.

Aos meus amigos, José Ailton Batista e Ricardo Tadeu Mendes, por sempre acreditarem na minha capacidade de vencer. Ao Luiz Fernando Zippinotti, um grande incentivador.

Aos colegas de classe, em especial aos amigos Letícia Barcelos, Ângela Abdo e Wagner Cardozo, que sempre tiveram uma palavra de incentivo.

Aos professores, pelos ensinamentos e pelos momentos vividos.

Ao Professor Dr. Anor da Silva Junior e à Professora Dr^a. Renata Silva Souza Guizzardi, pela simplicidade, obstinação e compromisso com seus orientados, em especial, pela participação neste trabalho.

“Um grão de ouro é capa de dourar grande superfície, mas não tão grande como um grão de sabedoria”.
Thoreau (1860)

RESUMO

Este trabalho foi desenvolvido em uma empresa distribuidora de energia elétrica na área de desenvolvimento de software, no Estado do Espírito Santo, e teve por objetivo pesquisar e analisar se a manutenção corretiva em programas computacionais está associada à complexidade dos programas computacionais e ao uso de mão-de-obra terceirizada. Para sua consecução foram elaboradas duas hipóteses, a primeira para verificar a relação entre a complexidade do programa e a realização de manutenção corretiva; e a segunda para averiguar a relação entre a utilização de mão-de-obra terceirizada no desenvolvimento de um programa e a realização de manutenção corretiva. Os dados foram coletados por meio de análise documental. O trabalho utilizou uma amostra composta de 365 programas que foram analisados sob a ótica de ponto de função. Os dados levantados sofreram tratamento estatístico, no qual as hipóteses foram testadas por meio dos testes de Qui-Quadrado e Binomial para proporção. Complementou-se a coleta de dados com entrevistas, gerando informações qualitativas que, juntamente com a análise dos documentos, ajudaram na compreensão do processo de manutenção. Ao final, a primeira hipótese foi aceita e a segunda rejeitada. Isso significa que a manutenção corretiva está associada à complexidade dos programas e que não se pode afirmar que exista relação entre as manutenções corretivas e o fato de o software ter sido desenvolvido por mão-de-obra terceirizada. A respeito da segunda hipótese, vale um breve comentário, pois, em geral, alta rotatividade (incluindo o emprego de mão-de-obra terceirizada) no desenvolvimento de sistemas pode levar à manutenção, no sentido de que esses funcionários não estão familiarizados com o negócio da empresa. Na empresa estudada, entretanto, isso não se caracterizou, pois boa parte do pessoal terceirizado é composta por ex-funcionários ou por profissionais que mantêm um relacionamento de longo prazo com a empresa.

Palavras-chave: Tecnologia da Informação, Manutenção Corretiva, Complexidade, Terceirização, Ponto de Função.

ABSTRACT

This dissertation has been developed in the software development area of an electricity distribution company in Espírito Santo. The main objective of this work is to investigate and analyze whether or not software corrective maintenance is associated to software complexity and to outsourcing. To accomplish this objective, this work is based on two hypotheses: the first one concerns the relation between software complexity and corrective maintenance; while the second verifies the connection of outsourced software development and corrective maintenance. We collected data through documental analysis and we used a 365-program sample for statistical analysis. First, we analyzed the programs of such sample from the point of view of function point. Then, we submitted the collected data to statistical treatment, in which the hypotheses were tried by means of Chi-square and Binomial tests for proportion. Some interviews provided qualitative information which, together with document analysis, helped making the maintenance process understandable. In the end, the complexity versus corrective maintenance hypothesis was accepted, and the outsourcing versus corrective maintenance hypothesis was rejected. That means that corrective maintenance is associated to program complexity and that one cannot be sure if there is a relation between corrective maintenance and the fact that the software was developed through insourcing or outsourcing. It is, thus, important to make a brief comment regarding the outsourcing versus corrective maintenance hypothesis. In general, volatile personnel (including outsourced personnel) in system development may lead to corrective maintenance, since these workers are not familiar with the core business of the company for which the system is being developed. However, in the case of the company under study, this could not be verified, since a big part of the outsourced personnel was composed of former workers or professionals who maintained a long-term relationship with the company.

Keywords: Information Technology, Corrective Maintenance, Complexity, Outsourcing, Function Point

LISTA DE TABELAS

Tabela 1: Motivos para terceirizar	28
Tabela 2: Matriz de complexidade EE.....	63
Tabela 3: Matriz de complexidade SE e CE.....	66
Tabela 4: Resultado da hipótese sobre complexidade versus manutenção corretiva	66
Tabela 5: Resultado da hipótese sobre mão-de-obra terceirizada versus manutenção corretiva	68

LISTA DE FIGURAS

Figura 1:A TI é implementada como reação à concorrência externa e à influência institucional.....	24
Figura 2:Espiral do Conhecimento.....	26
Figura 3:Ciclo de Vida de um Software	32
Figura 4:Visão da análise de ponto de função – caso Escelsa	44
Figura 5: Organização formal para solicitação de manutenção em software	54
Figura 6: Visão da área de trabalho do Lotus Notes	59
Figura 7: Visão da tela de um programa com funcionalidade EE.....	63
Figura 8: Visão da tela de um programa com funcionalidade CE.....	64
Figura 9:Visão das telas de um programa com funcionalidade SE.....	65
Figura 10: Fluxo do processo: Inovações X Terceirização	70

LISTA DE QUADROS

Quadro 1:Hipóteses.....	22
Quadro 2:Fase do Ciclo de Vida do Software	33
Quadro 3:Tipos de Manutenção.....	34
Quadro 4: Hipótese H0 (Nula) e H1 (Alternativa) – Teste das Hipóteses Complexidade Versus Manutenção Corretiva.....	46
Quadro 5: Hipótese H0 (Nula) e H2 (Alternativa) – Teste das Hipóteses Mão- De-Obra Terceirizada Versus Manutenção Corretiva	47
Quadro 6 : Etapas do Processo de Solicitação de Serviço de Sistemas	56
Quadro 7: Descrição dos Aplicativos Utilizados no Lotus Notes da Escelsa.....	60

LISTA DE SIGLAS

BFPUG	Brazilian Function Point Users Group
IFPUG	International Function Point Users Group
ECT	Economia de Custo de Transação
MOP	Mão-de-Obra Própria
MOT	Mão-de-obra Terceirizada
SSS	Solicitação de Serviço de Sistemas
TI	Tecnologia da Informação
TIT	Superintendência de Informática
UGB	Unidade Gerencial Básica

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa e Identificação do Problema.....	17
1.2	Objetivo do Trabalho e Hipóteses	20
1.3	Metodologia	21
1.4	Organização do Trabalho	22
2	REFERENCIAL TEÓRICO	23
2.1	Gestão do conhecimento	25
2.2	Terceirização	27
2.3	Manutenção de Software.....	31
2.4	Sistema Gerenciador de Base de Dados e Linguagem de Programação	36
2.5	Complexidade de Software.....	37
3	ASPECTOS METODOLÓGICOS	42
3.1	Coleta de Dados.....	42
3.1.1	Complexidade dos programas	43
3.1.2	Ocorrência de Manutenção Corretiva.....	44
3.1.2	Mão-de-Obra	45
3.2	As Hipóteses.....	45
4	O CASO ESCELSA	49
4.1	A Empresa.....	49
4.2	O Processo de manutenção na Escelsa.....	54

4.2.1 Processo de Solicitação de Serviços	55
4.2.2 Sistema Utilizado.....	58
4.3 A Pesquisa Empírica	60
4.2.1 As Hipóteses.....	61
4.2.1.1 Complexidade de Software.....	61
4.2.1.2. Tipo de mão-de-obra	67
4.2.2 Análise dos Resultados	68
5 CONCLUSÕES	72
6 REFERÊNCIAS.....	75
ANEXO A – Relatório do Pac.....	80
ANEXO B - Roteiro para Entrevista Estruturada – Gerente da Área de Desenvolvimento.....	81

1 INTRODUÇÃO

Estamos vivendo a *Era da Informação*, em que grande ênfase é dada à informação e ao conhecimento. Conseqüentemente, grandes investimentos são feitos em tecnologias que viabilizem a melhor utilização da informação e do conhecimento para benefício das organizações. Essas tecnologias são conhecidas como Tecnologias da Informação (TI).

Ao longo dos anos, empresas de vários setores têm alocado expressivo valor de seus orçamentos de investimento e custeio em TI. Conforme pesquisa realizada pela Gartner Group (2003), nos países de primeiro mundo elas chegam a investir cerca de 4,0% de sua receita neste segmento. É importante observar que no caso do Brasil os investimentos têm sido da ordem de 4,9% da receita líquida das empresas de acordo com pesquisa da Escola de Administração de Empresas de São Paulo da Fundação Getúlio Vargas (FGV-EAESP), e com tendência a crescimento (CARVALHO, 2004). Inicialmente paradoxal tal diferença é explicada em função de certa defasagem que existia, em relação aos países mais desenvolvidos, e que precisou ser vencida pelo setor empresarial, gerando esse percentual de investimento.

Castells (2007) justifica a adoção de novas tecnologias ligadas à informação, com base na velocidade das mudanças, na competitividade do mercado, no volume da produção, dentre outros fatores, que fazem da informação não só um instrumento fundamental para o crescimento, como também para a própria sobrevivência das organizações.

Desta forma, o tema tornou-se atrativo tanto para o segmento empresarial

quanto para o setor acadêmico. Pesquisas sobre a interação entre tecnologia da informação e organizações vêm sendo desenvolvidas em diversas áreas: serviços (MORANDIN;VARGAS, 2004), indústrias (MARTENS, 2001), entre outras.

Nesse cenário, inicialmente, a opção por uma nova TI é baseada na redução e controle dos custos, na otimização dos processos e na prestação de um serviço de qualidade aos seus usuários. E, após a implementação do sistema, a preocupação se volta para a manutenção corretiva (PRESSMAN, 2002). Tal tipo de manutenção encontra suas causas na dificuldade de se escolher a TI mais adequada, nas resistências por parte do ser humano e na própria questão da comunicação entre quem demanda a TI e o desenvolvedor. É o caso da empresa estudada, pois, conforme expressa seu gerente, a manutenção está intrinsecamente relacionada a dois aspectos: humanos e técnicos. Em que se pese, quando ainda era empresa estatal, já ter dado os primeiros passos em termos de adoção de sistemas informatizados, a empresa em estudo não obteve solução eficiente para os seus problemas. Assim, no primeiro caso quando se viu premiada pelo Mercado a adotar sistemas mais versáteis para controlar seu faturamento e arrecadação, não encontrou, em seu quadro de pessoal, profissionais suficientes para tal intento. A solução encontrada pela empresa para suprir esta demanda foi a terceirização de parte dos seus serviços de TI.

De acordo com pesquisa da *International Data Corporation* – IDC (2003), investimentos em terceirização tendem a aumentar em torno de 287,50% em relação aos últimos três anos, passando dos U\$ 16 bilhões para U\$ 46 bilhões em 2007. Dessa forma, os não investimentos em TI apresentam grandes riscos, senão de retrocesso, de obsolescência. Depois do *crash* das ações das grandes empresas

emergentes na área de tecnologia, no final dos anos 1990, os empresários não deixaram de monitorar de perto os gastos e investimentos feitos em TI (AMORIM, 2006). No caso da empresa pesquisada, esse monitoramento ficou explícito em relação aos custos de mão-de-obra que é acentuadamente terceirizada.

Se por um lado, a terceirização representa a adoção de mão-de-obra mais especializada em TI, por outro, ela leva à contratação de pessoal que desconhece o negócio principal da organização. Ao longo dos anos, colaboradores internos à organização acumulam conhecimento tácito (i.e. baseado em valores e experiências) e explícito (i.e. contido em manuais e documentos) sobre o negócio da empresa. Esse conhecimento é de grande valor no desenvolvimento de sistemas de TI e pode representar a diferença entre a criação de sistemas eficazes e o desenvolvimento de sistemas propensos a erros, e que, portanto, requeiram manutenções corretivas freqüentes.

Assim, partindo-se do princípio de que pode haver uma relação entre manutenção corretiva e mão-de-obra terceirizada, abre-se o espaço para se questionar a decisão estratégica da empresa de terceirizar parte de suas atividades de TI, no caso, o desenvolvimento de programas, incluindo os argumentos de redução de custos com mão-de-obra especializada.

Por outro lado, quando se atribui determinada tarefa aos colaboradores internos, fica mais difícil se cobrar os resultados e desempenho caso a estrutura organizacional não seja por projetos. Profissionais terceirizados, conforme SIMÕES (2002), têm apresentado melhor desempenho a menores custos e seu comportamento aproxima-se mais do perfil empreendedor, buscando soluções rápidas e eficientes. No entanto, este posicionamento a respeito de terceirizados não

é consenso. Autores como Saad (2006) relatam casos nos dois sentidos, ou seja, tanto do comportamento pró-ativo quando do comportamento dependente do terceirizado. A Razão da diferença desses comportamentos dos terceirizados pode ser encontrada na lógica do próprio mercado de trabalho.

1.1 Justificativa e Identificação do Problema

O início dos anos 1990 marcou para o Brasil a redução do tamanho do estado e o início dos processos de privatização. Na seqüência, fusões para aproveitar sinergias e modelos de gestão orientados para reengenharia de processos e terceirização de mão-de-obra tornaram-se comuns na realidade empresarial.

Apoiando essas tendências, a indústria da TI vivia o fim do domínio do Mainframe – o grande computador central que continha todas as bases de dados da organização - e assistia o início do uso de ambientes gráficos com a versão 3 do Microsoft Windows, com o qual os microcomputadores deixavam de parecer “pequenos mainframes” e ganhavam contornos revolucionários.

A tecnologia de comunicação de dados, cujo desenvolvimento havia se iniciado nos anos 1980 e que permitia que uma tela (não-gráfica) gerada pelo Mainframe fosse exibida em local remoto, passou a ser usada para permitir que um profissional acessasse o ambiente computacional das empresas a partir de um microcomputador instalado em casa, cujo preço, estava em declínio.

A redução de quadro de pessoal via terceirização de atividades não consideradas *core business* da empresa e a capacidade de trabalho remoto com acesso ao ambiente computacional permitiram que os núcleos de Desenvolvimento

de Sistemas existentes nas organizações fossem também reduzidos e tivessem suas atividades entregues a prestadores de serviços.

Esses fenômenos sociais alcançaram a Espírito Santo Centrais Elétricas S/A - Escelsa a partir de julho de 1995, quando de sua privatização. Com sistemas desenvolvidos internamente para todos os seus processos de negócio (Gestão da Rede Elétrica, Gestão Comercial, Suprimentos, Recursos Humanos e Infra-Estrutura de Comunicações e Logística), a empresa – que era estatal, com cerca de 80% de suas ações nas mãos do Governo Federal – viveu uma drástica reorientação para o mercado, e assistiu a um aumento considerável da demanda por funcionalidades em seus sistemas corporativos.

Essas transformações podem ser representadas pelo sistema de Gestão Comercial que tinha, no início da privatização, cerca de 500 mil linhas de código, e chegou ao final de 2002 com mais de 2 milhões de linhas de código. O desenvolvimento de novas funcionalidades nos sistemas existentes na Escelsa significa um aumento considerável em sua complexidade. Os novos sistemas com interface gráfica também tendem a ter complexidade maior que aqueles desenvolvidos para mainframe. Além disso, a equipe que o atendia em julho de 1995 era composta por 09 profissionais; passando a 40 profissionais entre mão-de-obra própria (9 profissionais) e mão-de-obra terceirizada (31 profissionais) entre 1999 e 2002.

Com o crescimento da demanda por novas funcionalidades e o aumento significativo da equipe de trabalho, muitos padrões e métodos de trabalho deixaram de ser usados – importava atender o cliente no menor prazo possível. O conhecimento integral do código-fonte de cada um dos módulos que compunham o

sistema de Gestão Comercial deixou de ser de domínio apenas da mão-de-obra própria, e a maioria dos Prestadores de Serviço tornou-se tão indispensável quanto havia sido, antes, a agora diminuta equipe da mão-de-obra própria. A passagem do pequeno para o grande havia desorganizado o processo de conhecer o sistema ao qual tiveram acesso os terceiros.

A partir dessa estrutura de funcionamento pode-se perceber que:

- Houve um aumento no número de manutenções corretivas, para as quais não foram encontradas razões aparentes.
- A equipe não sofreu alterações no número de profissionais que a integram, a despeito de pressão dos clientes por resultados ter dado mostra de diminuir. Algum elemento não evidente no processo parece manter esses colaboradores na equipe, mesmo que aos olhos de seus pares mostrem-se menos habilidosos.
- O código-fonte escrito já não é tão legível quanto antes e não guarda mais a preocupação de ser legível por todos. No entanto, compreender o código escrito por outro é muito difícil e, às vezes, até impossível. Assim, a pressão por resultados e o pouco entendimento do processo originaram um texto computacional pesado e de baixo desempenho. Descobre-se, então, que muitos terceiros não conhecem com clareza a linguagem computacional com a qual trabalham.

Dessa forma, propõe-se aqui uma reflexão sobre as causas da ocorrência de manutenção corretiva, buscando, assim, encontrar possíveis maneiras de solucionar o problema. Com isso, espera-se contribuir, em particular, para apoiar as decisões

da empresa aqui estudada, bem como, generalizar os resultados aqui encontrados para outros contextos. Logo, a importância deste estudo é incorporar ao arsenal de melhores práticas dos gestores de TI conclusões sobre as relações procuradas, com possibilidades de atuação nos respectivos processos em busca de organização e redução de custos operacionais.

1.2 Questões de Pesquisa e Objetivos

Partindo do problema e do contexto explicitado, o objetivo geral da pesquisa foi analisar ***que aspectos influenciam a manutenção corretiva em TI***. Para isso foram elaboradas duas questões de pesquisa:

- a) qual é a relação entre a complexidade do programa e a manutenção corretiva?
- b) qual é a relação entre a utilização de mão-de-obra terceirizada no desenvolvimento de programas e a realização de manutenção corretiva?

A primeira questão de pesquisa buscou responder se existe relação entre a complexidade no desenvolvimento de um programa e a realização de manutenção corretiva. Essa questão está fundamentada na literatura disponível da área de desenvolvimento de software. Existem alguns fatores que podem gerar a manutenção de um software destacando-se problemas relacionados à modelagem e de projeto. Segundo Pfleeger (2004), por exemplo, quanto maior for o número de funções em um sistema com possibilidade de sofrer alterações, maior a probabilidade de este sistema necessitar de manutenções corretivas.

A empresa em estudo optou, nos últimos anos, por desenvolver os seus

programas e as manutenções por meio de empresas terceirizadas de onde surgiu a segunda questão de pesquisa que buscou verificar a associação ou não da manutenção corretiva ao uso dessa mão-de-obra terceirizada.

1.3 Metodologia

Para o desenvolvimento deste trabalho, foi realizada uma pesquisa quantitativa com a elaboração de duas hipóteses que responderam às duas questões de pesquisa. A primeira hipótese denominada *complexidade versus manutenção corretiva* busca responder a primeira questão de pesquisa. A segunda hipótese denominada *mão-de-obra terceirizada versus manutenção corretiva* irá responder a segunda questão de pesquisa. As hipóteses estão descritas no Quadro 1.

Para essa investigação, levantou-se junto à área de desenvolvimento, os programas elaborados pela equipe de desenvolvimento da área comercial da Escelsa, referentes ao período de 1999 a 2007. O trabalho utilizou uma amostra, composta de 365 programas que foram analisados sob a ótica de ponto de função. Os dados levantados sofreram tratamento estatístico, no qual, as hipóteses foram testadas por meio dos testes de Qui-Quadrado e Binomial para proporção. Além disso, realizou-se entrevistas com o pessoal de TI da empresa, para obter informações qualitativas que, juntamente com uma análise documental, ajudaram na compreensão do processo de manutenção adotado.

Complexidade versus Manutenção corretiva.	Quanto maior a complexidade no desenvolvimento de um programa, maior ocorrência de manutenção corretiva.
Mão-de-obra terceirizada versus manutenção corretiva	Programas desenvolvidos por mão-de-obra terceirizada têm maior incidência de manutenção corretiva do que programas desenvolvidos por mão-de-obra própria

Fonte: Elaborado pelo autor

Quadro 1: Hipóteses

1.4 Organização do Trabalho

Esta dissertação está estruturada em quatro capítulos, além deste primeiro que introduz o tema problema e o objeto de estudo. No capítulo 2, é apresentado um Referencial Teórico que aborda os conceitos estruturais que norteiam a condução da pesquisa. Nos capítulos 3 e 4, estão, respectivamente, os Aspectos Metodológicos e o Caso. Por último, no capítulo 5, são tecidas as considerações finais, resumindo os resultados da pesquisa, em formato de conclusões.

2 REFERENCIAL TEÓRICO

O mercado empresarial tem passado por grandes transformações que têm pressionado as organizações a alterarem suas estruturas e processos em nome da competitividade (CHANDLER, 1962). De acordo com Hall (2006), existem sete dimensões ambientais às quais as empresas devem estar atentas, quais sejam: tecnológicas, legais, políticas, econômicas, demográficas, ecológicas e culturais.

No que tange à tecnologia, existe uma implicação vital em termos das interações entre organização e o ambiente. Isso se dá porque:

Novas idéias entram em circulação e tornam-se partes do ambiente, à medida que deixam de ser propriedade privada de um indivíduo ou organização específica. [...] Em função do ritmo de mudanças tecnológicas e de outras ambientais não ser constante para todas as organizações, varia o grau em que elas precisam desenvolver mecanismos de resposta. Para todas, no entanto a tecnologia permanece uma consideração importante (HALL, 2006, p.195).

De modo geral, as tecnologias de informação estão associadas às inovações. Para Kimberly (1981), elas se caracterizam por se diferenciarem das práticas ou tecnologias existentes, com forte conteúdo desconhecido até então. Complementando, Pennings (1987), argumenta que as inovações incluem desde pequenas variações nas práticas atuais até as modificações radicais que exigem importantes reorientações. A Figura 1 mostra a dinâmica desse processo, ou seja, como a TI é implementada em reação à concorrência externa e à influência organizacional.

De acordo com Zaltman, Duncan e Holbek (1973) existem algumas características que tornam a inovação mais ou menos atrativa para que seja adotada, quais sejam: custo; retorno do investimento; eficiência; risco e incerteza; comunicabilidade; competitividade; complexidade; status científico; vantagem

relativa percebida; ponto de origem; tempo; “*status quo ex-ante*”; compromisso; relações interpessoais; aspecto público versus privado; avaliadores; sustentabilidade a modificações sucessivas; capacidade do processo de avaliação e inovação precursoras.



Fonte: HALL (2006, p 185)

Figura 1: A TI é implementada como reação à concorrência externa e à influência institucional

Em termos de custo, é preciso considerar dois elementos: o econômico e o social, pois a adoção de um novo programa pode implicar perdas para as pessoas, podendo, inclusive, ser causa de demissão.

O retorno do investimento tem sido um dos pontos mais analisados porque os gestores têm sido intensamente cobrados pelos *stakeholders*, em particular, no caso da Escelsa, após a privatização, conforme relato do gerente entrevistado.

Um terceiro ponto a destacar diz respeito à suscetibilidade a modificações constantes. Contudo, ainda não é possível adotar esse critério para o caso de TI que envolve programas como o da Escelsa porque existem poucos com os quais tecer comparações e, a heterogeneidade ainda se faz presente. Basta mencionar que, de 1999 a 2007, conta a empresa com 6000 programas para executar os

procedimentos de faturamento e arrecadação, sem mencionar os das outras gerências.

De qualquer modo, a TI vem se tornando mais e mais necessária, à medida que o conhecimento e a informação vão ganhando mais importância para a empresa. Atualmente, obter a informação já não é o problema principal, já que os meios de comunicação e, em especial, a Internet a disponibilizam em grande quantidade. O mais importante agora é saber gerenciar e aplicar a informação disponível da melhor forma possível e isso não pode ser feito sem o uso de TI. Mas, como gerenciar o conhecimento? É isso que a próxima seção vem esclarecer.

2.1 Gestão do conhecimento

Para Nonaka e Takeuchi (1995, p.5,6)

Quinn (1992) compartilha com Drucker e Toffler a visão semelhante de que o poder econômico e de produção de uma empresa moderna está mais em suas capacidades intelectuais e de serviços do que em seus ativos imobilizados, como terra, instalações e equipamentos. [...] Em uma sociedade baseada no conhecimento [...] o “trabalhador do conhecimento” é o maior ativo.

Nesse sentido Guizzardi (2006) afirma que “as empresas percebem que a qualidade de seus produtos e serviços depende do uso eficiente do conhecimento criado e compartilhado por seus funcionários”.

Para Polanyi (1998) o conhecimento pode ser encontrado em representações consubstanciadas em teorias. Porém, parte desse conhecimento não é passível de ser colocado em formato de texto, ficando o conhecimento pessoal em sua maior parte, sem ser divulgado. Nessa mesma linha Nonaka e Takeuchi (1995) classificam o conhecimento em *tácito* e *explícito*. O primeiro é constituído por tudo aquilo que as

peças sabem de forma implícita, fruto de suas experiências e valores pessoais, enquanto, o segundo é formado por tudo o que pode ser transmitido de forma sistemática em linguagem formal.

Mintzberg, Ahlstrand e Lampel (2000) afirmam que o conhecimento tácito implica em muito mais do que aparenta, sendo relevante sua transformação em conhecimento explícito. Nesse sentido, Nonaka e Takeuchi (1995) sugerem quatro modos de conversão de conhecimento, conforme podem ser vistos na Figura 2, são eles: a socialização, a exteriorização, a combinação e a interiorização.



Fonte: NONAKA E TAKEUCHI (1995)

Figura 2: Espiral do Conhecimento

O ambiente organizacional é permeado por diversos elementos que nem sempre estimulam a socialização e a exteriorização do conhecimento tácito. Nonaka e Takeuchi (1995) relacionam e descrevem cinco condições das quais a empresa deve se valer no processo de gestão do conhecimento: *Intenção*, *autonomia*, *caos criativo*, *redundância* e *variedade*. A *intenção* diz respeito à estratégia e foi definida como sendo a aspiração de uma organização às suas metas. A *autonomia* está relacionada à motivação e à possibilidade de iniciativas inesperadas. O *caos criativo*

se relaciona às questões de mudanças organizacionais, que podem ser provocadas por agentes externos (concorrência) ou internos (metas desafiadoras). A *redundância* diz respeito à disponibilidade intencional de informações que superam as exigências operacionais mais imediatas. E, por fim, a *variedade* faz referência à extensão da capacidade dos funcionários de responder eficientemente aos desafios impostos pelo ambiente de negócio.

Segundo Nonaka e Takeuchi (1995), o conhecimento é criado em três níveis: individual, de grupo e organizacional. A teoria de Gestão do Conhecimento ajuda a compreender os ambientes de desenvolvimento de TI, uma vez que, nesses ambientes, há necessidade da interação entre o conhecimento tácito e o explícito, tanto no nível do indivíduo quanto nos níveis de grupo e organizacional.

2.2 Terceirização

Ao longo da década de 1990, as empresas passaram por um período em que os investimentos em TI oneravam muito o seu orçamento, devido ao alto custo dos equipamentos e às dificuldades de sua importação. No início dos anos 2000, a globalização provocou mudanças profundas no planejamento estratégico das organizações (PERES e ZWICKER, 2002). A busca por maior competitividade e a criação de vantagens competitivas através da informação tornou-se uma necessidade premente (PORTER, 1999), o que fez com que o conceito de investimento em TI fosse revisto e até priorizado.

Em termos de gestão, as decisões relativas à *make or buy* devem ser consideradas e analisadas de forma a possibilitar que a estratégia de TI gere valor para as empresas. Deste modo, surgiram dois cenários para as organizações: a)

fazer altos investimentos em TI ou b) terceirizar o desenvolvimento da TI para quem domina a tecnologia e detém o conhecimento.

Vários são os motivos que levam as empresas a terceirizar os serviços de TI. A Tabela 1 apresenta as razões mais comuns, citadas por diversos autores, para a terceirização dos serviços de TI.

Tabela 1: Motivos para terceirizar

Autor	Motivos para Terceirizar	Ordem de Importância
LEITE (1995)	Acesso ao conhecimento e à tecnologia	1
	Expectativa de redução de custo	2
	Concentração nas atividades-fim da organização	3
	Melhorar eficácia e obter novas facilidades	4
	Outros	5
PRADO (2000)	Expectativa de redução de custo	1
	Acesso ao conhecimento e à tecnologia	2
	Flutuação na carga de Trabalho	3
	Melhoria na prestação de serviço	4
	Atividades Rotineiras	5
	Gestão de Recursos Humanos	6
	Atividades com alto grau de particularidade	7
EARL (1998)	Expectativa de redução de custo	1
	Qualidade do serviço	2
	Baixo desempenho das unidades de TI	3
WANG (1995)	Expectativa de redução de custo	1
	Concentração nas atividades-fim da organização	2
	Gestão de Recursos Humanos	3
	Incertezas no ciclo de vida dos sistemas	4
PINHEIRO (1997)	Racionalidade Limitada	1
	Expectativa de redução de custo	2
	Concentração nas atividades-fim da organização	3
	Sinergia garantindo a inovação empresarial	4
VIDAL (1993)	Expectativa de redução de custo	1
	Concentração nas atividades-fim da organização	2
	Empresa de informática dentro da empresa	3
	Acesso ao conhecimento e à tecnologia	4
HENDRY (1997)	Melhorar eficiência e obter novas facilidades	1
	Acesso ao conhecimento e à tecnologia	2
	Concentração nas atividades-fim da organização	3
	Autonomia - agilidade e flexibilização	4
	Coordenação	5

Fonte: Elaborado pelo autor

Para dar embasamento à decisão de terceirização, existem teorias econômicas, em especial a Teoria de Custo de Transação e a Teoria da Agência.

Conforme se observa na Tabela 1, a maior motivação para a terceirização de atividades de TI é a **busca por redução dos custos**. Neste caso, a utilização da Economia de Custo de Transação (ECT) como *framework* tem ajudado as organizações no processo decisório no que diz respeito à terceirização.

Essa nova teoria econômica surgiu com Ronald Coase (1937), em seu artigo *The Nature of The Firm*, tentando explicar a competitividade das organizações. A Teoria Econômica Neoclássica, até então, se preocupava somente com os custos de produção. O autor, em seu artigo, chama a atenção para a existência de outros custos que, embora conhecidos, eram negligenciados. Esses custos, chamados custos de transação, são definidos por Williamson (1979) como *ex-ante* e *ex-post*. Os primeiros são os custos de esboçar, negociar e salvaguardar um acordo, enquanto, os últimos são aqueles decorrentes de problemas de adaptação que surgem quando a execução de um contrato é imprecisa como, por exemplo, resultados de atrasos, erros ou omissões. A unidade mínima de análise passa a ser a transação que, segundo Williamson (1979) surge quando um bem é transferível através de uma interface tecnologicamente separável.

Na área de TI, os custos de transação relacionados com a terceirização seriam os custos *ex-ante* (preparação da proposta, da análise e da contratação do serviço) e os *ex-post* (monitoramento e as aplicações de sanções quando do não cumprimento de cláusulas contratuais). Esses custos devem ser considerados além dos custos de produção para que a decisão de terceirizar, ou não, seja tomada de forma mais consciente.

Também a Teoria da Agência esclarece a relação da terceirização com custos extras. No processo de terceirização de serviço de TI, existe uma relação de

agência, que é caracterizada pelo contrato entre a empresa contratante (principal) e a empresa contratada (agente) que irá realizar o serviço. Nessa relação entre principal e agente, Jensen e Meckling (1976) afirmam que problemas de agência irão ocorrer oriundos do conflito de interesses existentes em atividades de cooperação entre pessoas. Esses problemas geram custos definidos pelo autor como custos de agência. Dentre eles, estão os custos de preparar o contrato entre o principal e o agente, gastos com controle das atividades do agente e despesas com a elaboração de relatório visando promover a transparência de informação entre agente e principal.

Problemas de gestão e estrutura na área de TI podem elevar o conflito de agência entre a área de TI e as outras áreas da empresa. Por isso, muitas vezes, as empresas buscam, na terceirização, uma solução para resolver esse conflito. Mas de acordo com Klepper e Jones (1998), a terceirização pode somente transferir esses problemas para um terceiro.

Muitas vezes, o que leva as organizações a buscarem a terceirização é o fato de que a área de TI não consegue atingir a sua máxima eficiência (LACITY, HIRSCHHEIM, 1999). Isso se relaciona a duas questões apresentadas na Tabela 1: o **acesso ao conhecimento** e a **concentração nas atividades fins das organizações**.

A tendência em concentrar esforços nas atividades fins (*core business*) da organização veio como uma das alternativas encontradas por elas para gerar vantagem competitiva. Essa necessidade surgiu a partir da revolução provocada pela informação que, segundo Porter (1999), afeta a competição, oferecendo às empresas novas formas de superar o desempenho dos concorrentes.

De acordo com Peres e Zwicker (2002):

Nas empresas cuja atividade fim (*Core Business*) não é informática, os investimentos em Tecnologia da Informação sempre foram questionados pelas demais áreas da empresa e considerados como sendo preciosos custos adicionais. Com o advento da reengenharia e dos *downsizings* ocorridos entre os anos 80 e 90, a terceirização veio a se tornar excelente opção para que as empresas passassem a contratar outras empresas especialistas e pudessem então se dedicar a suas atividades fins.

Em outras palavras, as empresas, freqüentemente, buscam a terceirização para obter mão-de-obra especializada e atualizada no que diz respeito a TI, bem como, para concentrar sua energia e esforço na realização daquilo que é o negócio principal (*core business*) da organização.

Essa decisão, porém, tem suas conseqüências, já que, se de um lado, a mão-de-obra terceirizada é mais especializada em TI, ela é menos especializada no que diz respeito ao negócio principal da organização contratante. Dessa forma, perde-se a chance de utilizar o conhecimento tácito adquirido pelos colaboradores internos no desenvolvimento de TI, o que pode resultar em produtos mais propensos a erros e, conseqüentemente, mais predispostos à manutenção corretiva.

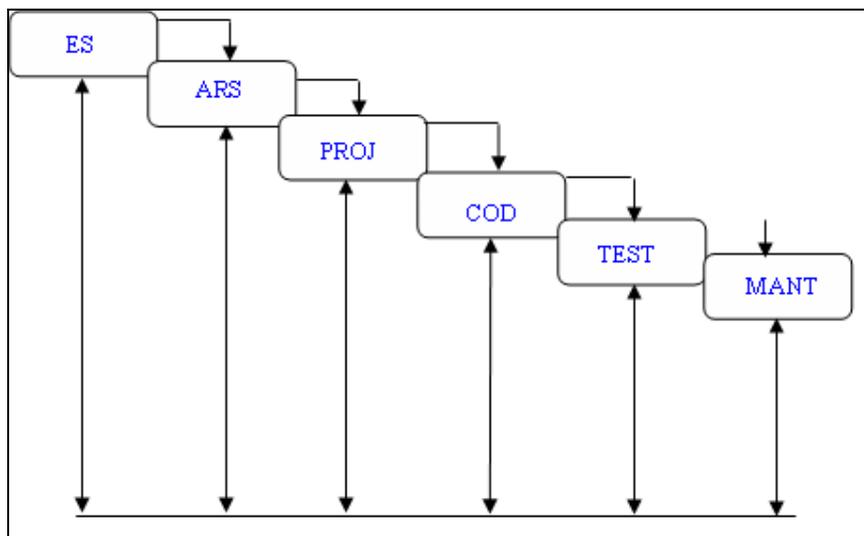
2.3 Manutenção de Software

De acordo com Leffingwell e Widrig (2003), Kotonya e Sommerville (1998), Sommerville e Sawyer (1997) projetos de software apresentam problemas em sua implementação, podendo até ser cancelados por não atenderem às necessidades dos clientes ou gerarem gastos além dos orçados. A manutenção de software demanda o maior volume de esforço dentre todas as atividades de engenharia de software (HANNA, 1993; PRESSMAN, 1995; BRODIE, STONEBRAKER, 1995).

A atividade de manutenção trata de softwares que já existem e que, no seu

desenvolvimento, cumpriram as fases de um ciclo de vida. Assim, a discussão exige a caracterização das fases do ciclo de vida do software, com detalhamento sobre o tema e, definição dos principais conceitos da atividade relacionados à sua manutenção.

O ciclo de vida de um software tem seu início na sua concepção e seu fim quando ele não está mais em produção. Conforme Pressman (1995), um ciclo de vida clássico é constituído de seis atividades ou fases assim definidas: engenharia de sistemas - ES, análise de requisitos de software - ARS, projeto - PROJ, codificação - COD, testes - TEST e manutenção – MANT. Tais fases são ilustradas por meio da Figura 3 e estão caracterizadas no Quadro 2.



Fonte: PRESSMAN, 1995, p 33

Figura 3: Ciclo de Vida de um Software

A Figura 3 apresenta o modelo denominado “Cascata”, onde as fases encontram-se em seqüência, visando mostrar que o desenvolvimento de uma fase deve terminar antes da outra começar. Esse modelo pretende mostrar para o desenvolvedor o que ocorre durante o desenvolvimento e a seqüência de acontecimentos que eles devem esperar encontrar.

Outros modelos de ciclo de vida de software estão descritos na literatura sobre engenharia de software como, por exemplo, a prototipagem, que é um processo que capacita o desenvolvedor a criar um modelo do software que será implementado; e o modelo espiral, que foi desenvolvido para abranger as melhores características tanto do ciclo de vida clássico quanto o da prototipação (PRESSMAN, 1995). Por uma questão didática, esta dissertação descreve o modelo em cascata, pois ele facilita o entendimento das fases do ciclo de vida de um software.

FASES	DESCRIÇÃO
ES	É a fase em que se inicia o estabelecimento dos requisitos uma vez que o software faz parte de um sistema mais amplo.
ARS	É a fase em que o domínio da informação e a função do software são mais bem compreendidos pelo desenvolvedor do software.
PROJ	É a fase da descrição da arquitetura do software e estruturas de dados. Além disso, nesta fase, detalhes procedimentais e caracterização de interface são traduzidos para um conjunto de representações.
COD	É a fase em que a linguagem escrita é traduzida em linguagem de máquina, resultando em instruções executáveis pelo computador.
TEST	É a fase em que se verifica se a construção lógica do software produz os resultados de acordo com os resultados exigidos.
MANT	É a fase em que o software poderá necessitar de alterações após a sua liberação para uso operacional.

Fonte: Adaptada de Pressmam (1995)

Quadro 2:Fase do Ciclo de Vida do Software

Parikh e Zvegintzoy (1983, p. 94) contribuem para um melhor esclarecimento e entendimento sobre manutenção de software, ao afirmarem que:

Manutenção de software é o trabalho feito em um sistema de software depois que se torna operacional. Isso inclui: entendimento e documentação dos sistemas existentes; extensão das funções existentes; adição de novas funções; descoberta e correção de erros; resposta a perguntas de usuários e pessoal de operações; treinamento de pessoal novo nos sistemas; reescrita, reestruturação, conversão e eliminação de software; gerenciamento do software de um sistema operacional; e muitas outras atividades que são incluídas na execução bem-sucedida de um sistema de software.

Segundo Canning (1972), a fase de manutenção de software é considerada uma das mais problemáticas dentre as fases do ciclo de vida do software, podendo

ser caracterizada como um *iceberg* em razão de que os maiores custos estão na parte não-visível dos problemas, tais como: a idade do software, a condição utilizada à época de seu desenvolvimento, documentação ruim ou sua inexistência, dentre outros.

Em outra dimensão a ISO/IEC 14764 (1999), apresenta quatro modalidades de manutenção conforme descritas no Quadro 3.

Tipos de Manutenção	Conceito
CORRETIVA	Usada para corrigir problemas depois que aconteceram
ADAPTATIVA	Usada para adequar o software a novos ambientes e mantê-los úteis nestes mesmos ambientes
PERFECTIVA	Usada para melhorias no desempenho e manutenibilidade
PREVENTIVA	Usada para corrigir as possíveis falhas antes que possam acontecer

Fonte: Elaborado a partir de ISSO/IEC (1999)

Quadro 3:Tipos de Manutenção

Para Pressman (1995, p. 878), a manutenção perfectiva é a que consome mais recursos e é resultante do software estar em uso e ser bem-sucedido. Este tipo de manutenção é responsável pela agregação de novas capacidades, alterações em funções existentes e ampliações gerais solicitadas pelos usuários.

Segundo Pressman (1995), a maior deficiência está na forma como o software foi planejado e desenvolvido. Entre outros problemas destacam-se: a falta de documentação, a dificuldade de entender a lógica usada pelo programador que desenvolveu o software, a alta rotatividade das pessoas nessa área, a inexistência de um projeto de mudanças para o software e o fato de que a manutenção não é vista pelos desenvolvedores como sendo um trabalho relevante (SCHNEIDEWIND, 1987).

Conforme pesquisa realizada com o gerente da área de desenvolvimento da

empresa pesquisada, existem cinco causas principais que levam à manutenção corretiva no desenvolvimento de software: a especificação não pode ser verificada por outra pessoa (documentação informal ou entendimento pessoal do Analista/Programador); há pressão de prazos e custos; há pouco tempo para reflexão sobre a integração do que será construído com o resto já existente; há pouco conhecimento do Analista/Programador sobre o contexto de negócio ao qual aquele programa se refere; há pouca experiência técnica do Analista/Programador no sentido de pouco tempo de experiência naquele ambiente ou linguagem no qual o programa será construído.

Para Allievi (2007), sistemas de informação "legados" devem ter equipes permanentes para manutenções corretivas, adaptativas e preventivas. Ou seja, alguns analistas, dependendo do tamanho do sistema, devem conhecê-lo profundamente, e precisam estar preparados e em permanente trabalho de sua alteração (manutenção).

O processo de manutenção de software nas organizações ainda ocupa um espaço não muito relevante, conforme argumenta Allievi (2007, p. 1)

Algumas organizações estão pouco preparadas para entender a importância da manutenção. Isso se constata pela falta de processos específicos para tratar do assunto ou, colocam analistas e programadores inexperientes e sem supervisão para fazerem as manutenções corretivas e acabam transformando o sistema em "uma colcha de retalhos". Em alguns casos tratam o processo de manutenção perfectiva como projetos de sistemas novos sem distinguirem a enorme diferença entre ambos. Outras vezes, fazem esforços monumentais, alocando grande soma de recursos para a condução de novos projetos que duram de um a dois anos e esquecem que, ao final, existirá um produto que precisará de manutenção, às vezes, por mais de 20 anos.

Várias são as razões que levam os softwares a sofrerem manutenções. Entre elas estão: adaptações às novas exigências do usuário, melhoria de *performance*, manutenções legais, ou seja, aquelas feitas por determinação de lei, mudanças na

regra do negócio, mudanças de ambiente, falha em atender às especificações, correções de falhas, entre outras. Muitas dessas atividades de manutenção estão relacionadas a uma estratégia ou a um planejamento falho ao se desenvolver um novo software (FLAHERTY *apud* PARIKH, 1996).

2.4 Sistema Gerenciador de Base de Dados e Linguagem de Programação

Os sistemas de banco de dados surgiram na década de 1960 e tinham como principal objetivo fornecer recursos capazes de armazenar, organizar, manipular e recuperar dados de forma rápida segura e eficiente. Constituem-se, portanto, em sistemas cujo objetivo global é registrar e manter informações (DATE, 1990). Melo (1997, p.3), define um sistema gerenciador de base de dados por meio de três elementos: “um ambiente de hardware e software composto por dados armazenados em banco de dados, o software que gerencia o banco de dados e os programas de aplicação”.

A história da computação registra várias mudanças em relação ao seu objeto. Grande parte da importância atribuída inicialmente ao hardware passou para o software devido à redução dos custos dos equipamentos, que estão evoluindo e tornando cada vez mais complexos exigindo elevados níveis de sofisticação nas linguagens de programação. (RANGEL, 2007).

Por sua vez, a linguagem de programação é composta por um conjunto de regras gramaticais usadas para escrever programas que instruem o computador a realizar determinadas tarefas. Cada linguagem de programação tem seu conjunto de palavras que ela reconhece, assim como regras específicas para organizar as instruções (ICEA/EAD, 2005).

No caso da empresa na qual foi realizada a pesquisa, a ferramenta para o desenvolvimento dos programas, que foram objeto de estudo, é a linguagem de programação *NATURAL* e, o Sistema Gerenciador de Base de Dados utilizado é o ADABAS.

O *NATURAL* é uma ferramenta de desenvolvimento de quarta geração desenvolvida pelo Software AG, empresa alemã. Algumas características dessa linguagem são: acesso a diversos bancos de dados; a existência de diversos editores para criar programas; funções, telas e área de dados; programação modularizada; o uso de funções e variáveis do sistema; execução *on-line* e *batch*; presença de um utilitário de criação ou manutenção de mensagens de erros de aplicação e portabilidade para outras plataformas (BRAGA, 2003).

2.5 Complexidade de Software

No mundo da engenharia de software a medição está longe de ser uma coisa comum. Existem discordâncias sobre o que medir e dificuldade para mensurar as dimensões consideradas relevantes.

De acordo com Bauer e Woessner (1982), um sistema de software pode ser caracterizado como complexo quando apresenta um conjunto de componentes abstratos (estrutura de dados e algoritmo) encapsulados na forma de procedimentos, entre si. São esses componentes e essas conexões que se configuram na arquitetura de software e que são executados em sistemas computacionais.

Os modelos abstratos e precisos permitem ao engenheiro especificar,

projetar, implementar e manter sistemas de software, avaliando e garantindo mecanismos para se planejar e gerenciar o processo de desenvolvimento.

Os profissionais do segmento da tecnologia da informação incluem a complexidade, tamanho do programa e a modularidade efetiva, como a garantia estatística da qualidade, pois são os elementos que permitem avaliar o software em relação à manutenibilidade, integridade e usabilidade do produto. Eles representam medidas indiretas, pois não medem propriamente a qualidade, mais sim, a sua manifestação. O fator complicador é a relação exata entre a variável que é medida e a qualidade do software. As razões que levam à complexidade do software são: tamanho; número de variáveis e funções; restrições de tempo; gerenciamento de memória; concorrência e interface orientada a eventos (BOOCH, 1990).

Dentre as várias razões para se medir um software estão:

1) indicar a qualidade do produto; 2) avaliar a produtividade das pessoas que produzem o produto; 3) avaliar os benefícios (em termos de produtividade e qualidade) derivados de métodos e ferramentas de software; 4) formar uma linha básica para estimativas; 5) ajudar a justificar os pedidos de novas ferramentas ou treinamento adicional (PRESSMAN, 1995, p.60).

Os softwares podem ser medidos de forma direta e indireta. As medidas diretas do processo de engenharia de software estão relacionadas ao custo e ao esforço como, por exemplo, linhas de código produzidas e velocidade de execução. Já as medidas indiretas do produto estão relacionadas à funcionalidade, qualidade, complexidade, manutenibilidade, etc. (PRESSMAN, 1995). Dentre essas medidas destaca-se uma conhecida como Ponto de Função, relacionada à funcionalidade.

Até os meados da década de 1970, a métrica baseada em linhas de código era a técnica mais utilizada para medir a produtividade de projetos de software, só que o grande número de linguagens existentes tornava inviável uma análise

conjunta de produtividade. Surgiu então, na *International Business Machines* - IBM, uma técnica baseada na análise de Ponto de Função que buscou criar uma medida que pudesse ser utilizada independente da linguagem de programação (VASQUES, 2003).

Essa métrica está focada na funcionalidade ou utilidade do programa e, visa medir o tamanho e a complexidade de uma aplicação, na visão do usuário (PRESSMAN, 1995).

Os objetivos dessa técnica são medir a funcionalidade que o usuário solicita e recebe do software e, medir o desenvolvimento e manutenção do software de forma independente da tecnologia utilizada para sua implementação. Além desses objetivos, a contagem de pontos de função deve ser simples o bastante para minimizar o trabalho adicional envolvido no processo de medição e uma medida consistente entre vários projetos e organizações (VASQUEZ, 2003).

O primeiro passo no processo de contagem é determinar o tipo de contagem de pontos de função, que abrange as funções do tipo “dados” e a do tipo “transação”.

A função do tipo dados representa a funcionalidade fornecida ao usuário para encontrar as informações internas e externas desejadas. Os tipos podem ser arquivos internos lógicos e arquivo de interface externo. O primeiro trata-se de um grupo identificável de dados logicamente relacionados ou informação de controle mantida dentro da fronteira da aplicação. O segundo consiste de um grupo identificável de dados logicamente relacionados ou informações de controle referenciadas pela aplicação que está sendo contada, mas mantida dentro da fronteira de uma outra aplicação.

A função do tipo transação representa a funcionalidade ao usuário para processar dados em uma aplicação. Neste caso os tipos de funcionalidade são definidos como: Entrada Externa, Saída Externa, Consulta Externa.

Entrada Externa - EE: é o processamento de dados ou informações de controle que entram pela fronteira da aplicação. Através de um processo lógico único, esses dados atualizam arquivos lógicos internos e/ou alteram o comportamento do sistema como, por exemplo, em uma aplicação de Cadastro de Clientes, as funcionalidades de: incluir cliente, alterar cliente e excluir cliente. Esta funcionalidade é identificada no código de programação *NATURAL* por meio dos comandos *store*, *update* e *delete*.

Saída Externa - SE: é o envio de dados ou informações de controle para fora das fronteiras da aplicação. Tem como principal intento apresentar informação ao usuário através de lógica de processamento que não seja simples recuperação de dados ou informações de controle. O seu processamento deve conter cálculos. Um exemplo é um relatório de totais de controle de faturamento por cliente. Esta funcionalidade é identificada no código de programação *NATURAL* por meio do comando *write workfile* e comandos de cálculo como, por exemplo, *compute*.

Consulta Externa – CE: é o processo onde uma entrada de dados ocasiona uma recuperação imediata de dados. Os dados exibidos não envolvem fórmulas, cálculos ou nenhum tipo de dado derivado. Um comando na linguagem de programação *NATURAL* que identifica esta funcionalidade é o *display*. Consulta a cadastro de cliente é um exemplo.

Os benefícios da análise de Ponto de Função para as organizações são: funcionar como uma ferramenta para determinar o tamanho de um programa

adquirido ao se contar todas as funções incluídas no programa; ajudar o usuário a determinar o benefício de um programa para suas organizações ao contar funções que especificamente atendam às suas necessidades; ser um veículo para estimar custos e recursos requeridos para desenvolvimento e manutenção de software e como fator de normalização para a comparação de software (VASQUEZ, 2003).

Entre os instrumentos utilizados por profissionais da área de informática para medição de software, a análise de Ponto de Função é uma das mais utilizadas e tem sido aplicada como unidade de medição de contratos de desenvolvimento e como ferramenta de gerência de projetos (VASQUEZ, 2003).

Em 1986, vários foram os trabalhos apresentados à comunidade, destacando a importância desse instrumento de medição (VASQUEZ, 2003). Com o crescimento acelerado do número de usuários, criou-se uma fundação denominada de *International Function Point Users Group* – IFPUG que tinha como objetivo principal a padronização da técnica de análise de Ponto de Função.

Em 1990, a IFPUG lançou no Mercado a primeira versão do Manual de Práticas de Contagem e deste então tem sido o padrão mais difundido. No Brasil, o uso da técnica teve seu início nos anos 1990 com um forte apoio da Unisys, empresa do ramo de TI. Sua consolidação se deu com a criação do *Brazilian Function Point Users Group* – BFPUG.

3 ASPECTOS METODOLÓGICOS

Para a realização da parte empírica da pesquisa foi necessário definir o tipo de pesquisa que deveria ser conduzido, a partir dos objetivos do trabalho e das variáveis envolvidas. Em consonância com esses aspectos, foram definidos o tratamento e análise dos dados.

3.1 Coleta de Dados

Em virtude das hipóteses levantadas de acordo com os objetivos do trabalho, fez-se necessária a condução de uma pesquisa que privilegiasse dados quantitativos com tratamento estatístico.

A população foi composta dos 6000 programas existentes na empresa estudada e, foi calculado o tamanho da amostra por meio de uma amostragem aleatória simples com base na exatidão esperada dos resultados, que é de 0,05% de nível de confiança e 95,00% de probabilidade de confiança, que resultou em 365 programas. Selecionou-se, então, 365 programas por processo randômico para servirem de base para o estudo aqui conduzido.

As variáveis utilizadas para se testar as hipóteses foram:

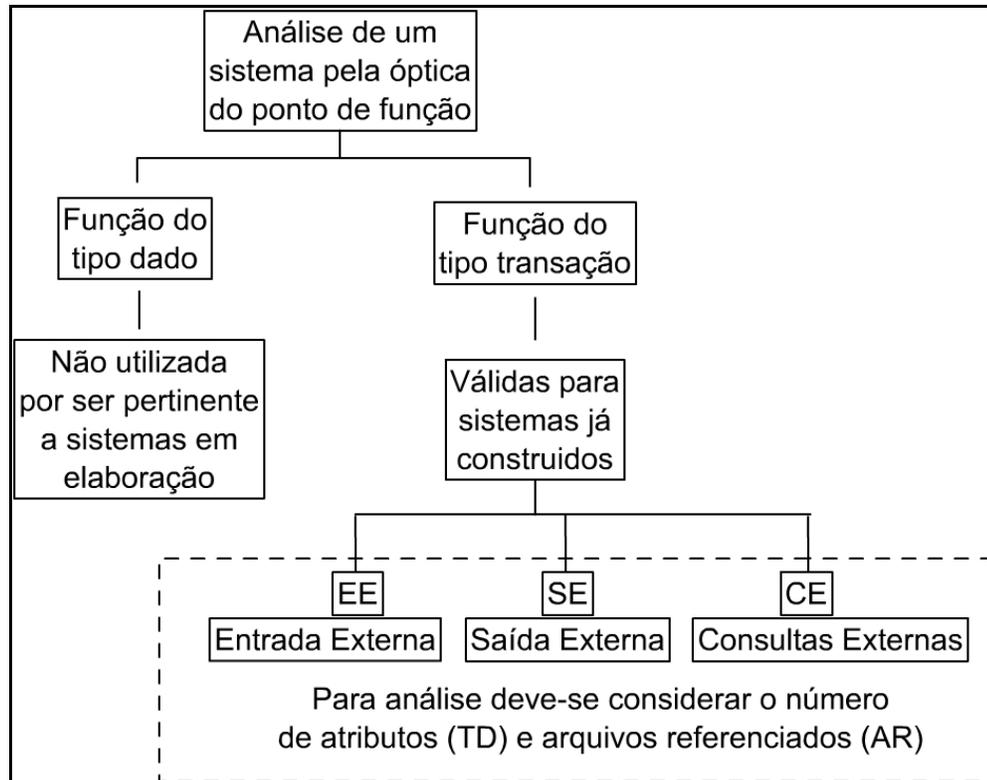
1. complexidade do programa;
2. se ocorreu ou não manutenção corretiva;
3. tipo de mão-de-obra utilizada para o desenvolvimento do programa, mão-de-obra interna ou terceirizada.

Um levantamento de dados foi necessário na identificação da organização: consulta a documentos, coleta de dados para as considerações iniciais junto a profissionais que ali trabalham e mapeamento do problema. Utilizou-se como apoio uma ferramenta qualitativa que foi a entrevista.

Com o intuito de esclarecer a história do próprio sistema, bem como, suas vantagens e desvantagens, foram entrevistados, informalmente, dez usuários do sistema de Solicitação de Serviço de Sistemas. Formalmente, foi entrevistado o gerente da Gerência de Sistemas Comerciais e da Distribuição.

3.1.1 Complexidade dos programas

A análise da complexidade de um sistema (variável 1) feita pela ótica do ponto de função é feita por meio da análise da função do tipo “dados” e do tipo “transação”. Por se tratar de análise de programas já prontos, só é pertinente a análise da função do tipo transação, conforme Figura 4. A partir das funcionalidades foram considerados os atributos (TD) e os arquivos referenciados (AR). De acordo com IFPUG (2001) a relação entre AR e TD define a classificação da complexidade, que pode ser alta, média ou baixa.



Fonte: Elaborado pelo autor

Figura 4: Visão da análise de ponto de função – caso Escelsa

A empresa estudada adota a seguinte abordagem para a determinação da complexidade do programa: se o programa contiver pelo menos uma funcionalidade classificada como alta, então, esse programa é classificado como de alta complexidade.

Para proceder a classificação foi necessário acessar cada um dos 365 programas que compuseram a amostra, identificar a funcionalidade, fazer a contagem dos TDs e dos ARs e plotar na matriz de complexidade estabelecida por IFPUG (2001).

3.1.2 Ocorrência de Manutenção Corretiva

Para verificar a ocorrência de manutenção corretiva (variável 2), a obtenção dos dados foi feita por meio da análise de um relatório extraído do PAC – *Predict*

Application Control (ver anexo B), software que gerencia o número de versões dos programas em produção.

O relatório apresenta o número de manutenções realizadas no software representadas pelo número de sua versão, ou seja, a versão 1 indica que o software ainda não sofreu manutenções, quando está na segunda versão, já sofreu uma mudança e assim sucessivamente.

Por definição da própria organização, considerou-se todos os casos de mudança no sistema como corretiva, pois esse é o tratamento dado internamente. As outras modalidades de manutenção também ocorrem, mas são consideradas como manutenção corretiva na empresa estudada. Assim, foram identificados os programas que sofreram manutenção.

3.1.2 Tipo de Mão-de-Obra

No caso da empresa pesquisada, existe no programa um campo que identifica sua autoria, dado pelo nome da empresa que o desenvolveu. Assim, para a coleta de dados referente ao tipo de mão-de-obra (variável 3) foi necessário acessar cada programa que compôs a amostra, selecionando-se o campo em questão com a respectiva tabulação, feita diretamente em uma planilha do Excel, onde também foram executados os testes Qui-Quadrado e Binominal para proporção usando o software PHStat.

3.2 As Hipóteses e suas Operacionalizações

Para o desenvolvimento da pesquisa foram elaboradas duas hipóteses

conforme os quadros 4 e 5 que estão relacionadas aos problemas de pesquisa da seção 1.2.

A primeira hipótese tratou da possibilidade da relação entre a complexidade no desenvolvimento do programa e a ocorrência de manutenção corretiva.

Assim, o Quadro 4 mostra H0 e H1 que serão testadas para validar a hipóteses complexidade versus manutenção corretiva.

H0	A necessidade de manutenção corretiva independe da complexidade do programa ($p_A=p_M=p_B$)
H1	Quanto maior a complexidade do programa, maior a chance do programa vir a necessitar de manutenção corretiva. ($p_A>p_M>p_B$)

Fonte: Elaborado pelo autor

Quadro 4: Hipótese H0 (Nula) e H1 (Alternativa) – Teste das Hipóteses Complexidade Versus Manutenção Corretiva

Onde:

p_A =proporção de programas de complexidade A que necessitaram de manutenção corretiva

p_M =proporção de programas de complexidade M que necessitaram de manutenção corretiva

p_B =proporção de programas de complexidade B que necessitaram de manutenção corretiva)

Devido aos dados não serem provenientes de uma população Normal, para responder às hipóteses propostas, deve-se utilizar um teste não-paramétrico, pois um teste não-paramétrico testa associações, dependência/independência e modelos, ao invés de parâmetros.

Como os dados são ordenados e o objetivo era testar se existe certa tendência crescente de tal forma que, quanto mais complexo o programa maior a chance do mesmo ser submetido à manutenção, optou-se por utilizar o teste do Qui-Quadrado (associação) que é aplicado a tabelas de contingência 2(linhas) x K(colunas), cujos dados podem revelar associação positiva ou negativa nas proporções das categorias das variáveis observadas.

O valor da estatística obtido dos dados da amostra deve ser comparado ao valor contido na tabela da distribuição Qui-Quadrado, correspondente a 0,05 de nível de significâncias e, 364 grau de liberdade. Se o valor da estatística de teste for maior que o valor tabelado ou se o p-valor for menor que 0,05, a hipótese nula deve ser rejeitada.

A segunda hipótese trabalhou a possibilidade de haver relação entre o uso de mão-de-obra terceirizada e a manutenção corretiva no desenvolvimento do programa.

O Quadro 5 mostra H0 e H2 que serão testadas para validar as hipóteses mão-de-obra terceirizada versus manutenção corretiva.

H0	As proporções de manutenções corretivas nos programas independe da mão-de-obra utilizada. (Isso se traduz em: dentre os programas que necessitaram de manutenção corretiva, a proporção de programas desenvolvidos por mão-de-obra terceirizada é: igual à proporção de programas desenvolvidos por mão-de-obra interna ($p_1=p_2$))
H2	As proporções de manutenções corretivas nos programas não independe da mão-de-obra (Isso se traduz em: dentre os programas que necessitaram de manutenção corretiva, a proporção de programas desenvolvidos por mão-de-obra terceirizada é maior que a proporção de programas desenvolvidos por mão-de-obra interna ($p_1>p_2$)).

Fonte: Elaborado pelo autor

Quadro 5: Hipótese H0 (Nula) e H2 (Alternativa) – Teste das Hipóteses Mão-De-Obra Terceirizada Versus Manutenção Corretiva

Onde :

(p_1 = proporção de programas desenvolvidos por mão-de-obra terceirizada

dentre os que necessitaram de manutenção corretiva)

(p_2 = proporção de programas desenvolvidos por mão-de-obra interna dentre os que necessitaram de mão-de-obra corretiva.)

O teste adequado é o Teste Binomial, pois este é particularmente útil em experimentos que apenas admitem duas alternativas como resposta, tais como mão-de-obra interna ou terceirizada.

Para avaliar a probabilidade de essas proporções serem consideradas estatisticamente não-diferentes, ainda que desiguais, em termos puramente numéricos, o teste utiliza o desenvolvimento matemático binomial de duas frequências relativas complementares p_1 e p_2 .

Assim, os dados experimentais utilizados pelo teste são as frequências relativas p_1 e p_2 , referentes às duas alternativas possíveis. A frequência esperada para p_1 e p_2 , em caso de igualdade perfeita, seria 0,5 para ambos.

Como, num experimento, dificilmente p_1 é igual a p_2 , o teste avalia, em última análise, até que ponto os valores de p_1 e p_2 podem diferir, sem deixarem de ser estatisticamente iguais.

Desta forma, o Teste Binomial foi aplicado para averiguar a hipótese de que, se a mão-de-obra for terceirizada, haverá maior probabilidade de ocorrência de manutenção corretiva. Neste teste, a decisão de rejeitar ou não a hipótese nula é tomada observando-se o p-valor. Se este valor for menor que 0,05, a hipótese nula é rejeitada.

4 O CASO ESCELSA

Para realizar o caso em questão, foi necessário conhecer o processo de manutenção da Escelsa, o que foi possível graças a uma extensa análise documental, complementada por entrevistas com pessoal interno à empresa. Após essa primeira etapa, passou-se à verificação das hipóteses propostas, seguindo a metodologia descrita no capítulo 3.

Este capítulo apresenta os resultados obtidos, iniciando-se com uma descrição do contexto de realização do caso e passando, depois, para a discussão dos resultados do experimento quantitativo.

4.1 A Empresa

A Escelsa é uma companhia aberta de capital autorizado, regida pela Lei das Sociedades por Ações, concessionária de serviço público de energia elétrica, com sede na cidade de Vitória, Estado do Espírito Santo. Foi criada em 1º de julho de 1968, sob o controle acionário das Centrais Elétricas Brasileiras S.A. – ELETROBRÁS. No leilão de privatização realizado em 11 de julho de 1995, 50% do capital, mais uma ação foram adquiridas pela IVEN S.A e GTD Participações S.A. O seu capital é aberto desde 19 de janeiro de 1996 (ESCELSA, 2005).

A administração da empresa foi regulada por um acordo de acionistas, da data da privatização até junho de 2002, quando expirou, passando o grupo Energias de Portugal S.A - EDP, como o maior acionista da IVEN S.A, a exercer o controle da empresa. Em 2005 foi criado a *holding* do grupo EDP no Brasil denominada Energias do Brasil onde a Escelsa está inserida.

A Energias do Brasil é uma *holding* que detém investimentos no setor de energia elétrica, consolidando atividades de geração, comercialização e distribuição de energia em quatro importantes estados do Brasil – São Paulo, Espírito Santo, Mato Grosso do Sul e Tocantins.

Em 2005, a Energias do Brasil distribuiu 23,1 mil GWh (gigawatt-hora) de energia elétrica, atendendo cerca de 3,0 milhões de clientes, o que representa quase 10 milhões de habitantes (ESCELSA, 2005). A capacidade instalada atual de geração do grupo, de 531 MW, e passou a 1.434 MW (megawatt) em 2006, com a conclusão de três projetos, destacando-se a Usina Hidrelétrica Peixe Angelical, no rio Tocantins.

Em termos de distribuição de energia, controla integralmente as empresas Bandeirante Energia S.A., Espírito Santo Centrais Elétrica S.A. e Empresa Energética de Mato Grosso do Sul S.A. Na comercialização, por meio de sua controlada Enertrade Comercializadora de Energia S.A, é um dos agentes mais ativos do mercado livre de energia. Em geração, participa dos seguintes empreendimentos: EDP Lajeado Energia S.A, Enerpeixe S.A e Energest S.A.

Controlada pela Energias de Portugal S.A - EDP, uma das maiores operadoras europeias do setor elétrico, a Energias do Brasil abriu seu capital no novo mercado da Bolsa de Valores de São Paulo, em julho de 2005, aderindo aos mais elevados padrões de governança corporativa. Mantém apenas ações ordinárias com direito a voto e concede *tag along* integral a seus acionistas.

A área de informática na Escelsa foi criada nos meados de 1970 e tem sofrido transformações ao longo dos anos. Com a privatização em 1995, surgiram novos investimentos em outras áreas da empresa que levaram ao aumento da demanda

dos serviços com fortes componentes de TI. Neste caso, foram feitas aquisições em hardware, software e pessoal para suprir essa demanda. Em 1998, adquiriu o controle acionário da Enersul – Empresa Energética do Mato Grosso do Sul e, visando diminuir os custos através da sinergia entre as empresas integrantes do grupo, optou-se por centralizar toda a área de TI na Escelsa. Essa sinergia só foi possível devido à política de uniformização do hardware e software.

Ao longo dos anos, visando a alavancar a competitividade, a área de Tecnologia de Informação continuou recebendo os investimentos necessários para viabilizar maior produtividade e qualidade nos serviços da empresa.

Os anos de 2000 a 2005 foram marcados pela consolidação do uso de ferramentas importantes, disponibilizadas em anos anteriores, como o Geoprocessamento, Data Warehouse, Sistema de ERP, Rede de Computadores, entre outras. Ocorreram também importantes avanços com o incremento de novas soluções para processos da empresa, boa parte delas voltadas para a Internet.

Em termos do processo de gestão, a Escelsa é composta por um conselho de administração, um presidente e quatro diretorias. A Superintendência de Informática – TIT está localizada na diretoria Administrativa Financeira e conta atualmente com quatro gerências: Gerência de Sistemas Técnicos e Administrativos, Gerência de Sistemas Comerciais e da Distribuição, Gerência de Produtos e Serviços e Gerência de Infra-estrutura de Tecnologia da Informação.

A Superintendência de Informática tem como principais atribuições:

- Prover a empresa com recursos de tecnologia da informação, conforme necessidades dos atuais e novos negócios, no prazo adequado e ao menor custo para os padrões de qualidade preestabelecidos;
- Elaborar o planejamento plurianual de Tecnologia da Informação, contemplando as prioridades do negócio dentro dos orçamentos autorizados pelo Conselho de Administração;
- Planejar, especificar, implantar e manter a infra-estrutura de TI para os ambientes *open* e *mainframe*, incluídos *hardware*, *software* básico e banco de dados;
- Manter os atuais sistemas de informação em produção, frente às necessidades de manutenção em seus aspectos legais, corretivos ou evolutivos;
- Desenvolver novos sistemas e/ou avaliar soluções oferecidas pelo Mercado;
- Implantar sistemas desenvolvidos *in house* ou adquiridos no Mercado;
- Operar o *Data Center* e controlar os resultados dos processamentos;
- Operar o sistema de impressão a laser;
- Planejar, especificar, testar, implantar e manter o parque de microinformática da empresa, tanto em nível de hardware como de software;
- Dar suporte aos usuários de Tecnologia da Informação, através do *help-center* ;

- Desenvolver e implantar projetos especiais com o uso de Tecnologia da Informação, visando à atualização e à racionalização de procedimentos.

A Gerência de Sistemas Técnicos e Administrativos tem como atribuições: a avaliação e a aquisição de sistemas de informação para áreas técnicas e administrativas; a implantação e a integração desses sistemas ao ambiente vigente em termos de novas soluções, a disponibilização de dados e informações gerenciais da empresa no ambiente *Data Warehouse*.

As principais atribuições da Gerência de Sistemas Comerciais e da Distribuição são: a avaliação de sistemas de informação; o desenvolvimento e manutenção; a implantação de sistemas de informação; a integração ao ambiente vigente de novas soluções para as áreas comerciais e da distribuição e a disponibilização de dados geo-referenciados para pesquisas temáticas diversas.

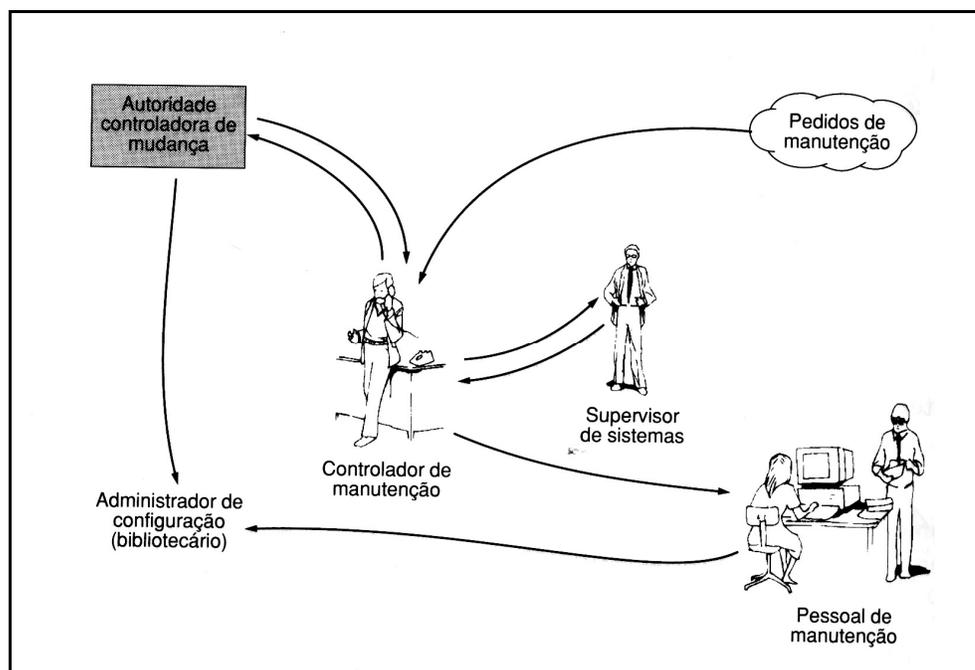
A Gerência de Produtos e Serviços tem como atribuições: a operação de computadores das plataformas *mainframe* e *risc*, a programação, o controle e a expedição de todos os serviços *batch* processados nesses ambientes. Além disso, é responsável também pela operação dos sistemas de impressão a laser central e distribuídos, o gerenciamento do parque de informática e a execução de ampliações e manutenções da rede de dados.

Com relação à Gerência de Infra-estrutura de TI, ela responde pela pesquisa de novas tecnologias; pelo planejamento, especificação, implantação e manutenção dos sistemas operacionais das redes de dados, dos bancos de dados dos ambientes *risc* e *mainframe*, bem como, pela análise e controle do desempenho dos ambientes-rede, *risc* e *mainframe* visando garantir níveis adequados de atendimento aos clientes.

A presente pesquisa foi desenvolvida na Gerência de Sistemas Comerciais e da Distribuição devido à sua importância para a empresa e pelo número de programas desenvolvidos para dar suporte ao *core business*.

4.2 O Processo de manutenção na Escelsa

Segundo Pressman (1995, p. 888), raras são as organizações que têm um modelo formal para solicitação de manutenção em software. Ao invés disso, a manutenção é usualmente feita à base de “luta livre”. Ele afirma não ser necessária uma organização formal, mas que tenha pelo menos uma estrutura informal onde existam delegações de responsabilidades, o que pode ser visto por meio da Figura 5. que mostra, também, o fluxo formal para solicitação de manutenção em software.



Fonte: Pressman (1995, p. 888)

Figura 5: Organização formal para solicitação de manutenção em software

Várias são as razões para a existência dessa organização, dentre elas: a redução da confusão sobre a quem deve ser feito o pedido, a melhoria no fluxo das atividades de manutenção e uma melhor distribuição de atividades, pois,

anteriormente, havia o risco do deslocamento de um analista que estava alocando seu recurso em desenvolvimento de um novo sistema para dar manutenção em sistemas já existentes.

Uma das atividades da Superintendência de Informática é o desenvolvimento de novos sistemas, assim como a alteração, a melhoria e a implementação de novos recursos nos sistemas existentes. Essa atividade é desenvolvida a partir de um pedido emitido pelas diversas áreas da empresa, conhecido como “Solicitação de Serviço de Sistemas”, neste documento, chamado genericamente de SSS.

A necessidade de registrar e documentar os pedidos que dão origem a esses trabalhos levou ao desenvolvimento de um sistema no ambiente *Notes* usado por todas as áreas da empresa. Com a criação desse sistema, a empresa adotou um modelo formal de solicitação de serviços de desenvolvimento de novos softwares e de manutenção em softwares já existentes.

4.2.1 Processo de Solicitação de Serviços

O processo de solicitação de serviço de sistemas é composto de 12 etapas descritas no Quadro 06. Na Escelsa existem alguns casos considerados particulares, dentro das solicitações de serviços. São eles: serviços de urgência e serviços de correção de anomalias em sistemas.

Os serviços de urgência são aqueles que não são planejados pela área solicitante, mas que, por exigência legal ou por solicitação de órgão externo ou superior, devem ser atendidos imediatamente. Este tipo de SSS deve ser aprovado pela gerência solicitante com, no mínimo, dois dias úteis de antecedência da data

necessária de conclusão, de modo que a TIT possa realizar uma análise do serviço e a alocação dos recursos necessários. Nesse caso a SSS deve conter, em seu detalhamento, justificativa clara que comprove seu caráter de urgência.

Etapas	Descrição
1	O interessado acessa o Sistema SSS - Solicitação de Serviço de Sistemas e faz a solicitação.
2	Automaticamente, a SSS é enviada à sua Gerência para aprovação.
3	Se a gerência da área solicitante, ou órgão superior, aprovar a SSS, esta é automaticamente enviada para a TIT.
4	Na TIT, a SSS é recebida pelas gerências de sistemas e passa por um processo de avaliação que visa a determinar: o escopo do trabalho; a viabilidade técnica de atendimento; as atividades necessárias para atendimento da Solicitação; uma estimativa de tempo para atendimento; a data mais provável de conclusão do serviço; o custo total estimado e a parcela que cabe a cada empresa se o serviço se destinar a atender a mais de uma; a identificação de disponibilidade de verba orçamentária para execução do serviço.
5	Sendo tecnicamente viável o atendimento e, havendo disponibilidade orçamentária, a gerência de sistemas da TIT e a gerência solicitante negociam as condições e prazos de atendimento.
6	Caso a gerência de sistemas da TIT considere que o conteúdo do campo <i>Detalhes do Serviço Solicitado</i> seja insuficiente para documentação da SSS e/ou para seu entendimento preliminar, poderá devolvê-la à Gerência solicitante, para complementação da definição do serviço.
7	Com as negociações levadas a bom termo, a Gerência de Sistemas da TIT aprova a SSS, que passa a compor a carga de trabalho da gerência.
8	Aprovada pelo gerente a SSS cairá automaticamente no correio do analista controlador que fará a planejamento da SSS.
9	O analista controlador, após análise da SSS e da descrição do serviço, faz a programação encaminhando para o analista/programador, levando em consideração a área de negócio em que o desenvolvedor atua e a sua disponibilidade.
10	O desenvolvimento de cada etapa de trabalho para atendimento da SSS é realizado conforme cronograma estabelecido e gerenciado pela Gerência de Sistemas da TIT.
11	Finalizado o atendimento da SSS, a gerência solicitante é informada e deve executar os testes necessários e registrar o aceite do serviço, no próprio sistema SSS.
12	Após o aceite do solicitante, a TIT disponibiliza a solução em ambiente de produção.

Fonte: Elaborado pelo autor

Quadro 6 : Etapas do Processo de Solicitação de Serviço de Sistemas

Serviços de correção de anomalia são aqueles que visam corrigir erro em um sistema, rotina, tela ou relatório já em uso e que esteja gerando resultado diferente do especificado pelo solicitante.

Várias foram as vantagens e as desvantagens citadas pelo gerente da área de desenvolvimento e pelas áreas usuárias, em termos da implementação do Sistema de Solicitação de Serviços.

Na ótica dos usuários entrevistados as principais vantagens foram: melhor

gerenciamento; melhor planejamento da mão-de-obra; a possibilidade de execução de tarefas paralelas, já que o solicitante não necessitaria estar junto ao desenvolvedor e, principalmente, o estabelecimento formal do prazo de entrega dos serviços contratados.

Em relação às desvantagens, foi percebido pelo solicitante: aumento da burocracia; a demora no atendimento, visto que antes esse atendimento era imediato, fruto de uma cultura de “quebra galho”, e, principalmente, dificuldade em definir a real necessidade da solicitação, devido à limitação dos campos da SSS.

Ao implantar o sistema de gestão de ordens de serviços, a área de TI vislumbrava muitas expectativas positivas, algumas das quais se confirmaram, segundo o gerente, entrevistado por meio dos seguintes resultados:

- conhecer quem estava pedindo o quê;
- conhecer quantos pedidos havia para uma certa Unidade de Atendimento (que, internamente chamada de Unidade Gerencial Básica - UGB);
- quantificar suas necessidades de recursos;
- quantificar as fontes de demanda, tanto em quantidade de ordens como em quantidade de homem hora para atendimento;
- custear o atendimento e permitir às áreas usuárias confrontar nossos custos contra o Mercado.

O sistema acabou tornando-se um canal de comunicação entre o desenvolvimento de sistemas e o restante da organização (Usuários dos processos internos da Escelsa), o que gerou uma pressão para que aumentasse a capacidade

de atendimento da Superintendência. Ficou estabelecida, então, a “crise do crescimento”. Nessa crise, surgiram alguns questionamentos importantes, tais como:

- como transferir conhecimento de modo rápido e eficaz, visando a aumentar a quantidade de recursos (mão-de-obra) em condição de ajudar no atendimento?
- como manter o nível de performance do código-fonte?
- como evitar que cada novo colaborador usasse seus próprios padrões, tornando o código-fonte ilegível?
- como, afinal, crescer “mantendo a ordem”?

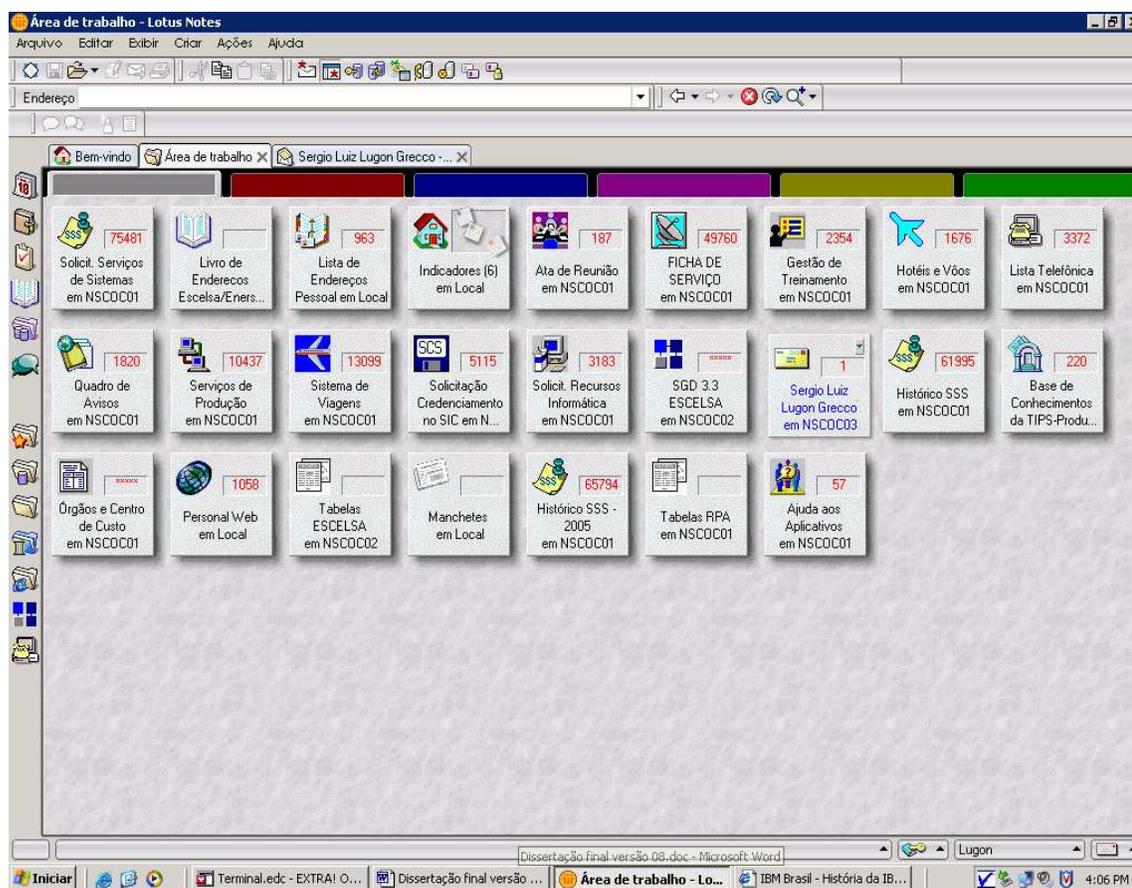
Até o momento não foram obtidas respostas robustas, o bastante, para todas essas questões a ponto de minimizar a crise gerada pelo crescimento desordenado da equipe de desenvolvimento. Apesar disso, foram encontradas, na implementação desse sistema, algumas vantagens como, por exemplo, a criação de um *backlog* (ordem) para o processo, que tornou visível a demanda, sua espera na fila, seu atendimento, etc. Enfim, o sistema gerou uma transparência à área de desenvolvimento de sistemas, suficiente para evitar que ocorram conflitos entre as áreas usuárias, o que foi muito bem-vindo, já que essa área realiza prestação de serviço a outras áreas da empresa.

4.2.2 Sistema Utilizado

A Escelsa adotou como ferramenta de trabalho o Lotus Notes que é um sistema cliente-servidor colaborativo e e-mail. Ele foi inspirado no Plato Notes, criado

por David Woolley na Universidade de Illinois, em 1973. Seu desenvolvimento atual é de responsabilidade da Lotus Software, do grupo IBM.

O Lotus Notes é um dos softwares de colaboração mais utilizados no mundo. Entre as suas funcionalidades, destaca-se o fato de servir como: cliente de comunicador instantâneo; navegador, caderno eletrônico e calendários. Além disso, possui outros recursos, bem como, uma plataforma para interagir com as aplicações corporativas.



Fonte: Escelsa (2007)

Figura 6: Visão da área de trabalho do Lotus Notes

Na empresa em estudo, a ferramenta Lotus Notes serve a vários propósitos: correio eletrônico, solicitação de serviço, lista telefônica, fichas de solicitação de serviços para a telecomunicação, solicitação de serviços de informática, entres outros. Uma visão da área de trabalho do Notes e as aplicações desenvolvidas,

estão representadas na figura 6.

O Quadro 7 apresenta a descrição dos usos do Lotus Notes, em termos de seus aplicativos e finalidades.

Aplicativo	Finalidade
Solicitação de Serviços de Sistemas	Permite solicitar serviços de sistemas à área de informática. Pode ser uma solicitação para criar um novo sistema ou para dar manutenção em sistema existente.
Solicitação de Recursos de Informática	Solicitação de aquisição de softwares, hardwares e outros equipamentos de informática.
Ficha de Serviço	Formulário para solicitar quaisquer serviços na área de telecomunicação como remanejamento ou defeito em ramal, rádio, fibra ótica, etc.
Gerência de Processos Administrativos	Sistema que substitui as folhas de encaminhamento de expediente tanto de origem externa quanto de origem interna.
Hotéis e Vôos	Consulta de hotéis por cidade e vôos saindo de Vitória e chegando a Vitória para facilitar o agendamento das viagens.
Contas Telefônicas	Consulta das contas telefônicas da ESCELSA. Mostra a conta de todos os telefones, sejam estes ramais, agregados (telefones que não fazem parte dos troncos de ramais) ou celulares.
Gestão de Treinamento	Sistema que permite criar, controlar e gerenciar solicitações de treinamento e acompanhar o atendimento dessas solicitações pela área de recursos humanos.
Serviços de Produção (SSP)	Sistema que permite fazer solicitações de impressão, de suporte e criação/alteração de formulários da área de produção na informática.
Pedido de Serviços	Sistema que serve para automatizar o fluxo de Pedidos de Serviço da Escelsa.
Solicitações da ANEEL	Sistema que veio substituir o antigo sistema de Reclamações e Pedidos da ANEEL
Passagem e Hospedagem	Sistema que permite solicitar viagens e hospedagem em Hotéis.

Fonte: Elaborado pelo autor

Quadro 7: Descrição dos Aplicativos Utilizados no Lotus Notes da Escelsa

4.3 A Pesquisa Empírica

Conforme já mencionando no capítulo 3, a pesquisa empírica foi desenvolvida com foco central na dimensão quantitativa representada, neste caso, pelos dados coletados na análise dos programas, que contemplaram as hipóteses. Para a

complementação das informações e consecução do objetivo fez-se necessária a utilização de entrevistas (instrumentos utilizados em pesquisa qualitativa).

4.3.1 As Hipóteses

Ao se iniciar o estudo do objeto de pesquisa desta dissertação partiu-se do princípio de que a manutenção corretiva está associada à complexidade do programa. Conforme já visto na seção 2, existem quatro tipos de manutenção que são: corretiva, adaptativa, perfectiva e preventiva. No caso da empresa na qual se fez a pesquisa, todas as manutenções são consideradas manutenções corretivas, uma vez que, para o gerente, ambas geram custos, quais sejam: o custo da própria manutenção e outros relacionados à indisponibilidade do programa que, em alguns casos, pode parar o sistema comercial da empresa, gerando perdas financeiras e outras ligadas à insatisfação do cliente.

No caso da empresa estudada e, de acordo com o Gerente o custo da manutenção de seus programas é de 38,30% do total do investimento realizado pela Gerência de Sistemas Comerciais e da Distribuição, sendo considerado muito alto. É importante ressaltar que, comparativamente ao encontrado no texto técnico, (PRESSMAN, 1995) até a década de 1980 esses custos chegavam a atingir 70,0% dos investimentos realizados.

4.3.1.1 Complexidade de Software

A Hipótese complexidade versus manutenção corretiva afirma que: Quanto maior a complexidade no desenvolvimento de um programa maior a ocorrência de

manutenção corretiva.

Para sua verificação foi utilizado o *Predict Application Control* - PAC, que por fazer o controle do número de versões dos programas em uso, permitiu identificar que em uma amostra de 365 casos (95,0% de confiança e 5,0% de erro) 67 foram objetos de manutenção. O levantamento (censo) realizado mostrou que dos 6.000 programas existentes, 1.104 sofreram modificações. No caso, quando foi solicitado o relatório via PAC (anexo B) os resultados foram apresentados por meio de números indicadores da versão que o programa se encontrava no momento. O número 1 indica que o programa ainda não sofreu nenhuma modificação. O número 2 indica que o programa sofreu uma modificação. O número 3 que sofreu duas modificações e, assim sucessivamente.

Para atribuir complexidade aos programas, a Escelsa utiliza a métrica de Ponto de Função. A contagem desses pontos abrange dois tipos específicos que são função do tipo dado e função do tipo transação. Dada a natureza do objeto de estudo dentro da temática, que são os programas, o foco recaiu sobre a “função transação” que representa a funcionalidade apropriada ao usuário para processar dados em uma determinada aplicação.

Para cada tipo de funcionalidade existe uma matriz de complexidade que, para ser entendida, necessita de algumas definições sobre arquivo e elemento de dado.

Arquivo Referenciado - AR: é um arquivo interno lógico ou um arquivo de interface externa lido pela função.

Tipo de Elemento de Dado - TD: é um campo reconhecido pelo usuário,

mantido em um arquivo interno lógico pela entrada externa.

Assim, a matriz de complexidade para a entrada externa está representada na Tabela 2. A Figura 7 apresenta um exemplo desta funcionalidade e o cálculo de sua complexidade.

```

> + Program CTB439P Lib PESQUISA
.....1.....2.....3.....4.....5.....6.....7..
1550 IF *NUMBER (F1.) GT 0
1560 FIND OFICINA WITH CODG-OFICINA-CREDENCIADA = #CODG-OFICINA-CREDENCIADA
1570 WHERE CODG-MUNIC-OFICINA-CREDENCIADA = #MUNIC
1580 MOVE CODG-TIPO-CONCERTO-OFIC-CREDEN TO #TCO
1590 MOVE CODG-ESCRIT-OFICINA-CREDENCIADA TO #ESCRIT
1600 MOVE CODG-MUNIC-OFICINA-CREDENCIADA TO #MUNIC
1610 PERFORM CONVALIDAR-DADOS
1620 MOVE #ANO-OFIC TO #ANO-CREDENCIAMENTO-OFICINA
1630 MOVE #MES-OFIC TO #MES-CREDENCIAMENTO-OFICINA
1640 MOVE #DIA-OFIC TO #DIA-CREDENCIAMENTO-OFICINA
1650 PERFORM BUSCAR-NOME-MATRICULA
1660 INPUT USING MAP 'CTB439M2'
1670 IF *PF-KEY = 'PF7'
1680 FETCH 'CTB439P' #PROGRAMA #MODULO #ROTINA
1690 END-IF
1700 IF *PF-KEY = 'PF12'
1710 FETCH 'CTB000P' #NN
1720 END-IF
1730 IF (#TCO NE 0)
1740 FIND TIPO WITH CODG-TIPO-CONCERTO-ELETRODOM =#TCO
.....1.....2.....3.....4.....5..... S 333 L 155

```

Fonte: Escelsa (2007)

Figura 7: Visão da tela de um programa com funcionalidade EE

No caso da empresa estudada, optou-se pela metodologia utilizada pela IFPUG (2001) que classifica a complexidade de programas em alta, média e baixa. Esses três níveis são determinados pela relação entre a quantidade de arquivos referenciados e o tipo de elemento de dado. Se o programa apresentar pelo menos uma funcionalidade alta ele será classificado como de alta complexidade.

Tabela 2: Matriz de complexidade EE

	1-4 TD	5-15 TD	>15 TD
0-1 AR	Baixa	Baixa	Média
2 AR	Baixa	Média	Alta
>2 AR	Média	Alta	Alta

Fonte: IFPUG 4.1 (2001)

Esta funcionalidade tem 6 TD e 1 AR. Consultado a matriz de complexidade na tabela 2, a funcionalidade é classificada com complexidade BAIXA.

A matriz de complexidade para a consulta externa e saída externa é representada na Tabela 3. As Figuras 8 e 9 apresentam um exemplo destas funcionalidades e os cálculos de sua complexidade.

Exibe informação da pesquisa de consumidor no arquivo referenciado

```

> + Program CFT996B1 Lib PESQUISA
...+.1...+.2...+.3...+.4...+.5...+.Mode Structured.
2781 READ (#QTDE-REG) PARCELA-FATURA-HIST-ENER
2782 BY SUPER-CDC-FAT-SIT-PARCELA-FATH
2783 EQ WSP-CDC-FAT-SIT-PARCELA-FATURA
2790 IF #DISPLAY EQ 'A'
2800 DISPLAY 'CDC' CODG-CONSUMIDOR-PARCELA-FATH
2810 'DATA' DATA-FAT-COBRANCA-PARCELA-FATH
2820 'CODG' CODG-PARCELA-FATH
2830 'VALOR' VALR-PARCELA-FATH
2840 'QTDE' QTDE-TOTAL-PARCELA-FATH
2850 'NUM PARC' NUME-DA-PARCELA-FATH
2860 'DT REF' NUME-REF-ORIGEM-PARCELA-FATH
2870 END-IF
2880 IF *ISN (LER-HENER.) LT #MENOR-ISN
2890 MOVE *ISN (LER-HENER.) TO #MENOR-ISN
2900 END-IF
2910 *
2920 IF *ISN (LER-HENER.) GT #MAIOR-ISN
2930 MOVE *ISN (LER-HENER.) TO #MAIOR-ISN
2940 END-IF
2950 *
...+.Current Source Size: 13820 Char. Free: 117501 .+... S 364 L 279
  
```

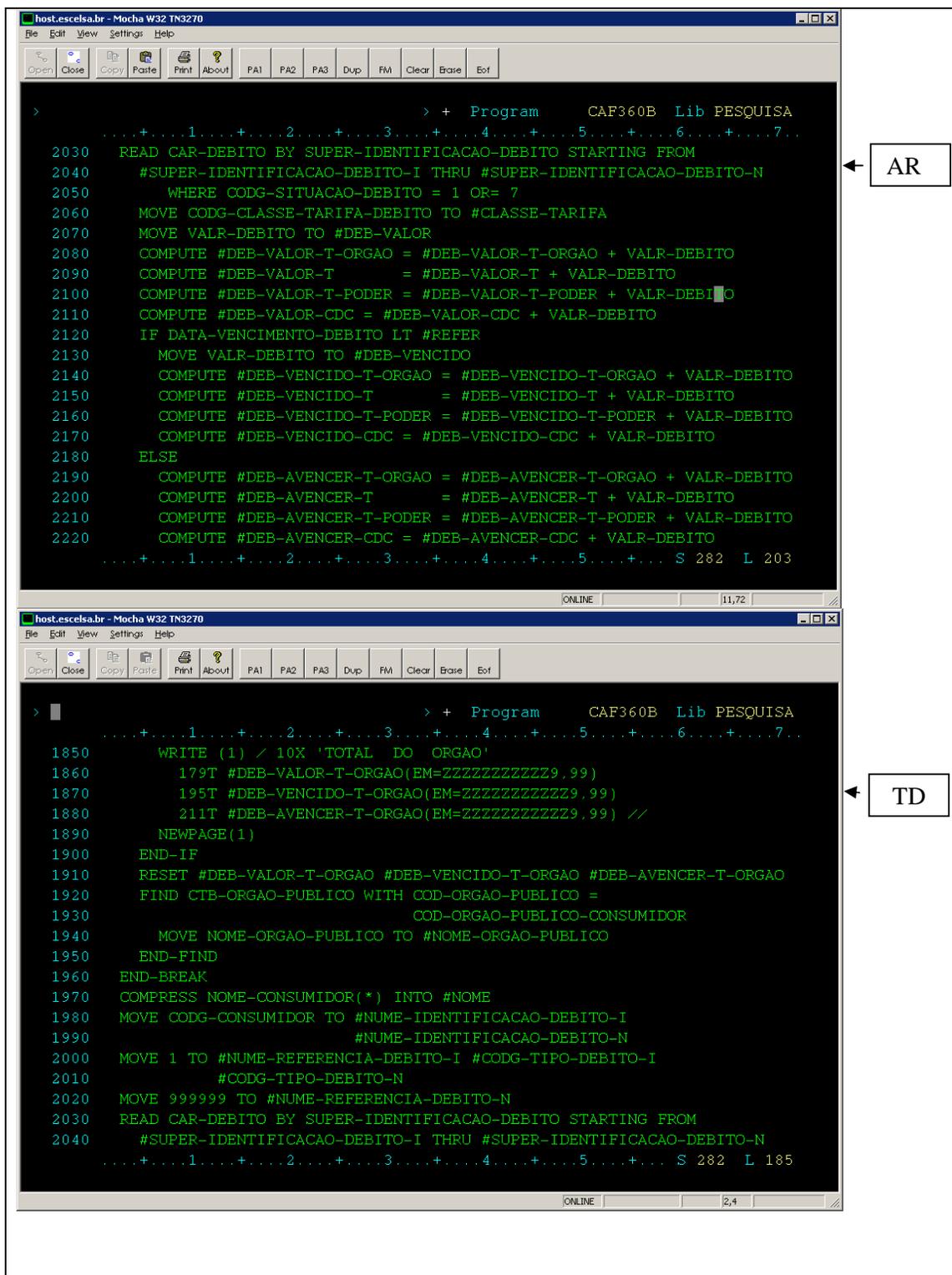
AR

TD

Fonte: Escelsa (2007)

Figura 8: Visão da tela de um programa com funcionalidade CE

Esta funcionalidade tem 7 TD e 1 AR. Consultada a matriz de complexidade na tabela 3, a funcionalidade é classificada com complexidade BAIXA.



Fonte: Escelsa (2007)

Figura 9: Visão das telas de um programa com funcionalidade SE

Esta funcionalidade tem 3 TD e 1 AR. Consultada a matriz de complexidade

na tabela 3, a funcionalidade é classificada com complexidade BAIXA.

Tabela 3: Matriz de complexidade SE e CE

	1-5 TD	6-19 TD	>19 TD
0-1 AR	Baixa	Baixa	Média
2 –3AR	Baixa	Média	Alta
>3 AR	Média	Alta	Alta

Fonte: IFPUG 4.1 (2001)

A Tabela 4 mostra o resultado do teste Qui-Quadrado de hipótese para a complexidade e manutenção corretiva.

Tabela 4:Resultado da hipótese sobre complexidade versus manutenção corretiva

Hipótese	Descrição	alfa	P- valor	Conclusão
H0	A necessidade de manutenção corretiva independe da complexidade do programa (pA=pM=pB)	0,05	0.0022	Rejeita a hipótese
H1	Quanto maior a complexidade do programa maior a chance do programa vir a necessitar de manutenção corretiva (pA>pM>pB)			

Fonte: Elaborado pelo autor

P-valor: é o nome que se dá à probabilidade de se observar um resultado tão ou mais extremo que o da amostra, supondo que a hipótese nula seja verdadeira:

$$p = P(Z \geq z | H_0) \quad (1)$$

Em outras palavras, p é a probabilidade de, supondo que a hipótese nula seja verdadeira, observar-se o valor registrado em um experimento por acaso.

Para o presente o caso, o valor encontrado para esta estatística foi igual a 9,3608 e o p-valor foi igual a 0,0022, indicando que a hipótese nula é rejeitada, ou seja, há evidências de que há realmente uma tendência de aumentar o número de manutenções corretivas, conforme aumenta a complexidade do programa.

Portanto, pode-se concluir que, para a complexidade versus manutenção corretiva existem evidências que mostram que as manutenções corretivas estão associadas à complexidade do programa.

4.3.1.2. Tipo de mão-de-obra

No caso da Hipótese mão-de-obra terceirizada versus manutenção corretiva no processo de desenvolvimento do software partiu-se da afirmativa de Pfleeger (2004, p. 391) de que:

É preciso um tempo considerável para aprender o suficiente sobre um sistema, a fim de se poder entendê-lo e modificá-lo. O esforço de manutenção é prejudicado quando os membros da equipe são rotineiramente substituídos por outros grupos, se alguns membros da equipe deixam a organização para trabalhar em outros projetos ou se existe a expectativa de que os membros da equipe façam a manutenção de diversos produtos diferentes, ao mesmo tempo.

Ao tratar de problemas associados à manutenção de software Schneidewind (1987), dentre outros fatores, destaca:

A “outra pessoa” freqüentemente não está por perto para explicar. A mobilidade entre o pessoal da área de software é elevada. Não podemos contar com uma explicação pessoal do desenvolvedor de software quando a manutenção for necessária.

Para a segunda hipótese, foram consultados e analisados os 365 elementos da amostra configurados nos programas, a fim de identificar se seus autores eram do quadro da empresa ou se foram contratados como prestadores de serviço. Com relação à manutenção corretiva, foram utilizados os mesmos parâmetros da primeira hipótese.

A Tabela 5 mostra o resultado do teste binomial de proporção relacionado à manutenção corretiva e à mão-de-obra terceirizada.

Tabela 5: Resultado da hipótese sobre mão-de-obra terceirizada versus manutenção corretiva

Hipótese	Descrição	alfa	P- valor	Conclusão
H0	As proporções de manutenções corretivas nos programas independe da mão-de-obra utilizada. (Isso se traduz em: dentre os programas que necessitaram de mão-de-obra corretiva, a proporção de programas desenvolvidos por mão-de-obra terceirizada é: igual à proporção de programas desenvolvidos por mão-de-obra interna ($p_1=p_2$))	0,05	0,7963	Não rejeita a hipótese
H2	As proporções de manutenções corretivas nos programas não independe da mão-de-obra (Isso se traduz em: dentre os programas que necessitaram de mão-de-obra corretiva, a proporção de programas desenvolvidos por mão-de-obra terceirizada é maior que a proporção de programas desenvolvidos por mão-de-obra interna ($p_1>p_2$)).			

Fonte: Elaborado pelo autor

Para o presente o caso, o valor encontrado para o p-valor foi igual a 0,7963, indicando que, a hipótese nula não é rejeitada, ou seja, não há evidências de que a proporção de manutenções corretivas realizadas por mão-de-obra interna e terceirizada são estatisticamente diferentes.

Para mão-de-obra terceirizada versus manutenção corretiva, pode-se concluir que, de acordo com os dados da amostra não há evidência de que as manutenções corretivas estejam associadas ao tipo de mão-de-obra utilizada.

4.2.2 Análise dos Resultados

Autores como Pressman (1995) e Pfleeger (2004) afirmam que existe uma correlação entre a manutenção e a rotatividade de pessoal e a manutenção e o conhecimento do negócio. Assim, intuitivamente, poder-se-ia pensar que a

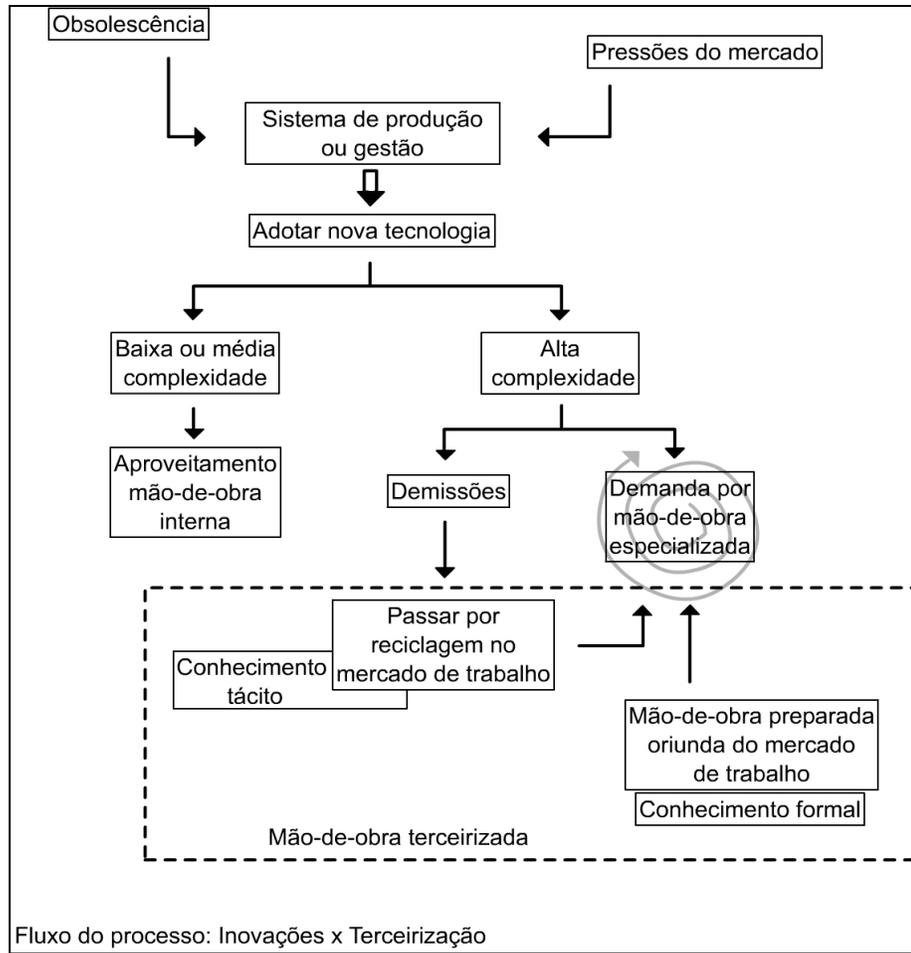
terceirização levaria, necessariamente, ao desenvolvimento de programas de menor qualidade, resultando, conseqüentemente, em um maior número de manutenções. A rejeição da hipótese mão-de-obra terceirizada versus manutenção corretiva suscitou a busca pelas principais causas que levaram a não confirmação dessa intuição. As principais razões encontradas foram: a longevidade da relação entre os terceiros e a empresa, e o número de ex-colaboradores que trabalham nas empresas prestadoras de serviço. Seja pela questão de longevidade ou por serem ex-colaboradores da empresa, tais profissionais têm “intimidade” tanto com a estrutura organizacional e seu funcionamento quanto com o próprio programa que está sendo desenvolvido. Isso justifica que o número de manutenções não venha a subir com o uso de mão-de-obra terceirizada.

Para aprofundar as reflexões sobre as questões acima, é preciso compreender o que ocorre quando a empresa necessita adotar uma nova tecnologia. A figura 10 mostra o fluxo do processo desde a adoção da nova tecnologia até a contratação ou re-contratação da mão-de-obra especializada, de modo geral via terceirização.

No primeiro momento, por pressões do mercado ou por obsolescência as empresa se vêem “obrigadas” a buscar novas soluções tecnológicas para continuar sendo competitivas. Essa nova tecnologia, que pode ser de baixa ou alta complexidade, leva as empresas a tomar uma decisão em relação aos seus profissionais: aproveitamento da mão-de-obra existente ou a demissão e a busca por outros com maior capacitação no mercado.

Note que, no segundo caso, os colaboradores demitidos levam consigo todo o conhecimento tácito acumulado nos anos de trabalho na empresa. Isso é bastante

prejudicial para a organização que perde, com isso, parte do conhecimento útil para o seu bom funcionamento.



Fonte: Elaborado pelo autor

Figura 10: Fluxo do processo: Inovações X Terceirização

Se, por outro lado, um ex-colaborador é re-contratado, agora por via da terceirização, como mostra a Figura 10, então esse conhecimento pode ser re-aplicado dentro da empresa, gerando novamente vantagem competitiva. É o que o experimento aqui realizado acabou de mostrar: já que muitos ex-colaboradores eram re-contratados para o desenvolvimento de programas, a necessidade de manutenção não crescia com a decisão de terceirizar esse desenvolvimento.

Um problema constatado, entretanto, é que, no caso da empresa pesquisada, poucos foram os registros formais do conhecimento gerado com o desenvolvimento de programas. A maior parte ficou consubstanciada em conhecimento tácito. Esse tipo de comportamento traz conseqüências negativas para a organização, pois qualquer tentativa de participação de novos profissionais no processo, o que inclui a manutenção de programas, fica cerceada por aqueles que detêm o conhecimento. Isso se configura em um problema gerencial.

No início, os programas eram desenvolvidos por mão-de-obra interna. Posteriormente, a empresa optou por terceirizar tais serviços, o que poderia ter ocasionado muitos problemas, já que não havia conhecimento registrado para subsidiar o trabalho dos terceirizados. Isso, felizmente, não aconteceu porque parte desses terceirizados são oriundos da própria organização, o que significa que o conhecimento gerado no ambiente de desenvolvimento de software não se perdeu. Isso no entanto, é uma característica específica do caso estudado. A Escelsa, bem como as organizações em geral, devem, portanto, ter o cuidado de registrar o conhecimento adquirido com o desenvolvimento de programas de TI para evitarem problemas durante a manutenção ou, no desenvolvimento de novas aplicações.

5 CONCLUSÕES

Este estudo de natureza quantitativa teve por objetivo realizar uma análise do processo de manutenção de software da Escelsa, por meio da verificação da possível relação entre a manutenção corretiva e a complexidade em programas e ao uso de mão-de-obra interna ou terceirizada. O local escolhido para a pesquisa foi a Gerência de Sistemas Comerciais e da Distribuição de uma Empresa Distribuidora de Energia no Estado do Espírito Santo. Foram feitas análises documentais das solicitações de serviço e entrevistas com o gerente da Gerência de Sistemas Comerciais e da Distribuição e usuários, visando buscar o entendimento do fluxo desse processo. Os programas, objeto de estudo, foram analisados buscando identificar as suas funcionalidades e atribuir valores de complexidade.

Para responder às duas questões de pesquisas foram utilizadas técnicas estatísticas. A população em estudo foi composta de 6000 programas e a composição da amostra foi feita de forma aleatória. Sobre a amostra de 365 programas, 18,36% sofreram manutenções que representaram gastos acima de 38,0% do orçamento de investimento da gerência.

Para a questão da relação entre a complexidade do sistema e a realização de manutenção corretiva, o teste Qui-Quadrado permite afirmar que a complexidade está associada às manutenções corretivas executadas em programas. Em relação à questão da utilização do uso da mão-de-obra terceirizada com a manutenção corretiva, o teste binomial revela que não se pode afirmar a existência de associação entre as variáveis.

No caso da empresa estudada, 30,00% dos profissionais terceirizados na

área de TI são oriundos de seu próprio quadro de pessoal. Isso significa que eles trazem consigo não só o conhecimento explícito como também o tácito e, pode ser esse conhecimento tácito o elemento que viabiliza a terceirização do desenvolvimento de software, por serem conhecedores do domínio dos sistemas, da linguagem de programação, do ambiente, etc. Tal situação mostra-se por si só, como diferente daquelas nas quais o terceirizado não teve contato *ex-ante* com as fases de desenvolvimento do programa.

Outro ponto que deve ser considerado é a duração dos contratos feitos para a prestação dos serviços de desenvolvimento e manutenção do software, que são de longo prazo. Isso pode explicar, também, em parte, porque não se verifica a ocorrência de um maior número de manutenções quando a mão-de-obra é terceirizada.

No estudo, utilizou-se a métrica de ponto de função para atribuir valores de complexidade às funcionalidades dos programas, apesar de várias outras métricas de medição de qualidade e complexidades estarem disponíveis no Mercado e no meio acadêmico como, por exemplo, as métricas de Halstead, complexidade ciclomática de McCabe e quantidade de linhas de programação. Pesquisar a correlação entre a complexidade e manutenção corretiva utilizando uma outra métrica poderia ser objeto de outros estudos. É possível, ainda, utilizar a mesma métrica deste trabalho, incluindo a variável experiência do desenvolvedor, natureza do domínio, técnicas de engenharia de requisitos, etc. para verificar se existe relação com a manutenção corretiva em programas.

A maior dificuldade encontrada para a realização do trabalho foi a falta de documentação dos processos de desenvolvimento e manutenção do software. Um

procedimento que poderia ser adotado pela empresa para evitar a perda do conhecimento tácito e uma diminuição do tempo das manutenções seria a externalização do conhecimento com a elaboração de documentação que contemplem o processo de desenvolvimento do software, desde o levantamento dos requisitos, pelo desenvolvedor, até a construção do código de programação.

Ao final, considera-se que os objetivos da pesquisa foram atingidos e que no caso da Escelsa, um novo olhar sobre a manutenção e a natureza da mão-de-obra parece se descortinar, devendo-se levar em consideração, no processo de gestão da Gerência de Sistemas Comerciais e da Distribuição, os aspectos relacionados à mão-de-obra, à complexidade e à manutenção dos programas. Torna-se também interessante a replicação de estudos como este em outras empresas a fim de se verificar a possibilidade da generalização dos resultados.

6 REFERÊNCIAS

ALLIEVI, O. P. **Reflexões sobre manutenção de sistemas legados**. Disponível em: <<http://www.imasters.com.br/artigo/5919/desenvolvimento>>. Acesso em: 5 mai. 2007.

AMORIM, A. C. **O empurrão da TI: PNAFE é a realidade**. Brasília, abr. 2006. Disponível em:<<http://www.fazenda.gov.br/ucp/pnafe/docs/Revista%20PnafeRealidade.pdf>>. Acesso em: 5 mai. 2007.

BAUER, F. L; WOESSNER, H. **Algorithmic Language and Program Development: monographs in computer science**. New York: Springer-Verlag, 1982.

BOOCH, G. **Object Oriented Design**. Benjamin-Cummings: Califórnia, 1990.

BRAGA, L. R. P. **Curso Básico de Programação Natural**, 2003. Disponível em: <http://araujodomingues.googlepages.com/apostila_natural_basico.pdf>. Acesso em: 30 de mai. 2007.

BRODIE, M. L.; STONEBRAKER, M. **Migrating legacy systems: gateways, interfaces & the incremental approach**. San Francisco: Morgan Kaufmann Publishers Inc., 1995.

CANNING, R. The Maintenance “Iceberg”. **EDP Analyser**, v.10, n. 10, 1972.

CARVALHO, T. C. **Falta à chamada governança de TI**, 2004. Disponível em: <<http://www.itweb.com.br/noticias/artigo.asp?id=50613>>. Acesso em: 08 mai. 2007.

CASTELLS, M. **Sociedade em rede**. 10. ed. São Paulo: Paz e Terra, 2007.

CHANDLER Jr., A. D. **Strategy and structure**. Cambridge, MA: MIT Press, 1962.

COASE, R. H. The nature of the firm. **Economica**, v. 4, n. 16, p. 386-405, 1937.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 9. ed. Rio de Janeiro: Campus, 1990.

EARL, M. J. Deve-se terceirizar a informática? **HSM Management**, São Paulo, v. 1,

n.6, ano 6, p. 126-132, jan./fev. 1998.

ESCELSA – Espírito Santo Centrais Elétricas S.A. **Relatório Anual**. Vitória, 2005. <http://www.energiasdobrasil.com.br/aempresa/informacoes_financeiras/relatorio_anual/2005/inicio/pdf/Energias_Final.pdf>. Acesso em: 08 mai. 2007.

GUIZZARDI, R. S. S. **Agent-Oriented Constructivist Knowledge Management**. 2006. 351 f. Tese (Doutorado) – Centre For Telematics and Information Technology, University of Twente, The Netherlands, 2006.

HALL, R. H. **Organizações: estruturas, processos e resultados**. 7. ed. São Paulo: Pearson Prentice Hall, 2006.

HANNA, M. Maintenance Burden Begging for Remedy. **Software Magazine**, 53 f., p. 53-63, abr. 1993.

HENDRY, J. O custo oculto da terceirização. **HSM Management**. São Paulo, v. 0, n. 2, p. 82-90, maio/jun. 1997.

IDEA/EAD - Introdução à linguagem de programação. Disponível: <<http://www.icea.gov.br/ead/anexo/23101.htm>>. Acesso em: 21 set. 2007.

IFPUG - Counting Practices Manual, v 4. n. 1. 2001. Disponível: <<http://www.ifpug.org/publications/manual.htm>>. Acesso em: 21 set. 2007.

ISO/IEC 14764. International Organization for Standardization. Information technology - Software Maintenance, 1999. Disponível: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39064>. Acesso em: 21 set. 2007.

JENSEN, M. C.; MECKLING, W. H. Theory of the firm: managerial behavior, agency costs and ownership structure. **Journal of Financial Economics**, v. 3, 1976.

KIMBERLY, J. R. **Managerial innovation**. Handbook of Organizational Design, v.1. Eds. Paul C. Nystrom e William Starbuck. New York: Oxford University Press, 1981.

KLEPPER, R.; JONES, W. O. **Outsourcing information technology, systems & services**. Upper Saddle River: Prentice Hall, 1998.

KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering: process and**

techniques. New York: John Wiley & Sons Ltd, 1998.

LACITY, M. C.; HIRSCHHEIM, R. **Information technology outsourcing: what problems are we trying to solve?** Oxford: Oxford. Univ. Press, 1999.

LEFFINGWELL, D; WIDRIG, D. **Managing Software Requirements: a use case approach.** 2nd ed. New York: Addison-Wesley, 2003.

LEITE, J. C. **Terceirização em informática no Brasil: Investigação sobre a situação da Terceirização em Informática no contexto brasileiro.** NPP – Núcleo de Pesquisas e Publicações e relatórios de Pesquisa. Relatório nº 13/1995,1995.

MARTENS, C. D. P. **A tecnologia de informação em pequenas empresas industriais do Vale do Taquari/RS.** 2001. Dissertação (Mestrado em Sistemas de Informação e de Apoio a Decisão) – Mestrado Interinstitucional PPGA - UNIVATES, Porto Alegre, 2001.

MELO, Rubens N.; SILVA, Sidney D.; TANAKA, Asterio K. **Banco de Dados em Aplicações Cliente/Servidor.** Rio de Janeiro: Infobook, 1997.

MINTZEBERG, H.; AHLSTRAND, B.; LAMPEL J. **Safári de estratégia: um roteiro pela selva do planejamento estratégico.** Tradução: Nivaldo Montingelli Júnior. Porto Alegre: Bookman, 2000.

MORANDI, F.; VARGAS L. M. **Estratégia de gestão do conhecimento e o uso de tecnologia da informação: um estudo de caso em uma empresa de software.** Porto Alegre. Facef Pesquisa, v.7, n. 2, 2004.

NONAKA, I; TAKEUCHI, H. **Criação do conhecimento na empresa.** Tradução de Ana Beatriz Rodrigues, Priscila Martins Celestes. Rio de Janeiro: Campus, 1997.

NOTES. Disponível: <<http://www-128.ibm.com/developerworks/lotus/library/ls-NDHistory>>. Acesso em: 03 de jun. de 2007.

PARIKN, G. **Techniques of program and system maintenance.** 2nd. ed. Wellesley, MA: QED Information Sciences, Inc. 1988.

PARIKN, G.; ZVEGINTZOV, N.; **Software Maintenance,** IEEE Computer Society Press, 1983.

PENNINGS, J. M. **On the nature of new technology as organizational innovation**. In *New Technology in Organizational Innovation: The Development and Diffusion of Microelectronics*. Eds. Johannes M. Pennings e Arend Buitendam, p. 3-12. Cambridge, MA: Ballinger, 1987.

PERES, G.; ZWICKER, R. **Terceirização da tecnologia da informação**: um estudo sobre os motivos que levaram as empresa brasileiras a terceirizar TI na década de 90 e início dos anos 2000. São Paulo: Working Papers nº03/11, FEA-USP 2002.

PFLEEGER, S. L. **Engenharia de Software**: teoria e prática. 2. ed., São Paulo: Prentice Hall, 2004.

PINHEIRO, I. A. **Dos fundamentos à pratica da subcontratação**. In: ENCONTRO NACIONAL DA ANPAD (EnANPAD), 21, Rio das Pedras: Associação Nacional de Pós-Graduação e Pesquisa em Administração, 1997. CD-ROM.

POLANYI, M. **Personal knowledge**: towards a post-critical. Chicago: University Of Chicago Press, 1998.

PORTER, M. **On Competition**. Boston: Harvard Business Review Book, 1999.

PRADO, E. P. V. **Terceirização da Tecnologia da Informação**: Uma avaliação dos fatores que motiva sua adoção em empresas do setor industrial de São Paulo. 2000. 169 f. Dissertação (Mestrado em Administração) – Programa de Pós Graduação da Faculdade de Economia, Contabilidade e Contabilidade, Universidade de São Paulo, São Paulo, 2000.

PRESSMAN, R. P. **Engenharia de software**. São Paulo: Makron Books, 2002.

PRESSMAN, R. P. **Engenharia de software**. São Paulo: Makron Books, 1999.

PRESSMAN, R. P. **Engenharia de software**. São Paulo: Makron Books, 1995.

RANGEL, J. L. **Linguagem de programação**. Disponível em: <<http://www-di.inf.puc-rio.br>>. Acesso em: 21 set. 2007.

SAAD, A. C. **Terceirização de serviço de TI**. São Paulo: Brasport, 2006.

SCHNEIDEWIND, N. F. The State of Maintenance, **IEEE Trans. Software Engineering**, v. SE-13, 1987.

SIMÕES JÚNIOR, C. A. **Terceirização de atividade produtiva: um estudo de caso na indústria do vestuário capixaba.** . 2002. Dissertação (Mestrado em Administração) – Programa de Pós Graduação em Administração, Universidade Federal do Espírito Santo, Vitória, 2000.

SOMMERVILLE, I.; SAWYER, P. **Requirements Engineering – a good practice guide.** New York: John Wiley & Sons, 1997.

The Standish Group International. **Chaos Report**, 1995. Disponível em: <www.standishgroup.com/sample_research/index.php>. Acesso em: 27 jul. 2004.

VAZQUEZ, C. E. **Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software.** São Paulo: Érica, 2003.

VIDAL, A. G. **Terceirização: a arma empresarial.** São Paulo: Érica, 1993.

WANG, C. B. **O novo papel do executivo de informática.** São Paulo: Makron Books, 1995.

WILLIAMSON, O. E. Transaction Cost Economics: The Governance of Contractual Relations. **Journal of Law and Economics**, v. 22, p. 233-261, 1979.

ZALTMAN, G., DUNCAN, R.; HOLBEK, J. **Innovations and organizations.** Nova York: Wiley Interscience, 1973.

ANEXO A – RELATÓRIO DO PAC

105/07/2007 ULTIMA VERSAO DO OBJETO NO REPOSITORIO PAC HR: 15:46:40
 APLICACAO: COMERCIAL PAG. 6

OBJETO	VRS	DATA-VRS	EVENTO
0CAC-OBSERV	1	06/07/2005 14:31:15	COM10675_IN_06A
CAC-PRESTA	1	05/08/2005 14:22:21	COM10530_IN_99
CAC-PROG-C	1	28/06/2005 20:43:20	COM10630_IN_08
CAC-PROJET	2	05/02/2006 18:05:16	COM99999_IN_205
CAC-PROJET	1	12/07/2005 10:38:36	COM10530_IN_19
CAC-RESSAR	1	26/08/2005 15:12:34	COM10530_IN_202
CAC-SOLICI	1	29/10/2005 15:47:19	COM10530_IN_275
CAC-SOLICI	1	05/02/2006 18:11:28	COM99999_IN_206
CAC-SOLICI	4	29/12/2006 16:54:03	COMDELTA-I-002E
CAC-TRANSM	2	18/07/2005 14:24:12	COM10676_IN_24
CAC001N	1	19/06/2005 03:51:29	INCorp-COMERCIAL
CAC003N1	1	19/06/2005 03:51:29	INCorp-COMERCIAL
CAC003N2	1	19/06/2005 03:51:29	INCorp-COMERCIAL
CAC003N3	1	19/06/2005 03:51:29	INCorp-COMERCIAL
CAC007N	1	19/06/2005 03:51:29	INCorp-COMERCIAL
CAC007N1	1	19/06/2005 03:51:29	INCorp-COMERCIAL
CAC008N	1	19/06/2005 03:51:30	INCorp-COMERCIAL
CAC009N	1	19/06/2005 03:51:30	INCorp-COMERCIAL

105/07/2007 ULTIMA VERSAO DO OBJETO NO REPOSITORIO PAC HR: 15:46:41
 APLICACAO: COMERCIAL PAG. 7

OBJETO	VRS	DATA-VRS	EVENTO
0CAC010N	1	19/06/2005 03:51:30	INCorp-COMERCIAL
CAC011N	1	19/06/2005 03:51:30	INCorp-COMERCIAL
CAC011N1	1	19/06/2005 03:51:30	INCorp-COMERCIAL
CAC012N	1	19/06/2005 03:51:30	INCorp-COMERCIAL
CAC015N	1	19/06/2005 03:51:30	INCorp-COMERCIAL
CAC017	1	19/06/2005 03:51:30	INCorp-COMERCIAL
CAC017B2	1	19/06/2005 03:51:31	INCorp-COMERCIAL
CAC017N	1	19/06/2005 03:51:31	INCorp-COMERCIAL
CAC018P0	2	19/12/2006 17:13:43	COMDELTA_IN_00642
CAC019N	1	19/06/2005 03:51:32	INCorp-COMERCIAL
CAC019N1	1	19/06/2005 03:51:32	INCorp-COMERCIAL
CAC040P0	1	19/06/2005 03:51:32	INCorp-COMERCIAL
CAC100B	1	19/06/2005 03:51:32	INCorp-COMERCIAL
CAC101B	1	19/06/2005 03:51:32	INCorp-COMERCIAL
CAC102B	1	19/06/2005 03:51:32	INCorp-COMERCIAL
CAC103B	1	19/06/2005 03:51:33	INCorp-COMERCIAL
CAC150B	1	19/06/2005 03:51:33	INCorp-COMERCIAL
CAC160B	1	19/06/2005 03:51:33	INCorp-COMERCIAL

105/07/2007 ULTIMA VERSAO DO OBJETO NO REPOSITORIO PAC HR: 15:46:41
 APLICACAO: COMERCIAL PAG. 8

OBJETO	VRS	DATA-VRS	EVENTO
0CAC200B	1	19/06/2005 03:51:33	INCorp-COMERCIAL
CAC210B	1	19/06/2005 03:51:33	INCorp-COMERCIAL

ANEXO B - ROTEIRO PARA ENTREVISTA ESTRUTURADA – GERENTE DA ÁREA DE DESENVOLVIMENTO

- a) Qual ou quais são os critérios utilizados na distribuição das solicitações de serviços (SS), entre mão de obra própria e mão de obra contratada?
- b) Quais as principais alterações na gestão da área de desenvolvimento que ocorreram nos últimos seis anos?
- c) Qual o critério usado na seleção e contratação dos profissionais na área de desenvolvimento de sistema?
- d) O que explica a redução do número de SS no ano de 2004 em relação aos outros anos?
- e) Há quantos anos você está trabalhando na área de desenvolvimento de sistemas? Baseado em sua experiência profissional, qual seria a quantidade de horas gastas para que se possa classificar um programa ou sistema como complexo?
- f) Em sua opinião, quais são as causas que levam um programa a ter manutenção corretiva?
- g) Existe uma relação entre experiência, complexidade e manutenção corretiva?
- h) Que práticas (uso de sistemas ou processos específicos) são utilizadas para nivelar o conhecimento dentro da equipe de desenvolvimento da Escelsa?
- i) A seu ver, o conhecimento dos profissionais mais experientes é bem aproveitado pelo grupo? Se não, o que poderia melhorar?
- j) Como é feita a sinergia entre o trabalho dos profissionais da equipe da Escelsa com membros de equipes terceirizadas? A coordenação e o monitoramento de trabalhos externos favorecem essa sinergia?