Leandra Leal

SABER SOCIAL E DESENVOLVIMENTO DE SOFTWARE: AVALIAÇÃO CRÍTICA DO MODELO DA FÁBRICA DE SOFTWARE

Belo Horizonte
Departamento de Engenharia de Produção da UFMG
2008

Leandra Leal

SABER SOCIAL E DESENVOLVIMENTO DE SOFTWARE: AVALIAÇÃO CRÍTICA DO MODELO DA FÁBRICA DE SOFTWARE

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Engenharia de Produção.

Área de concentração: Produto e Trabalho

Linha de pesquisa: Ergonomia e Organização do Trabalho

Orientador: Prof. Dr. Francisco de Paula Antunes Lima

Belo Horizonte

Departamento de Engenharia de Produção da UFMG

2008

BANCA EXAMINADORA

_____

Profº Dr. Francisco de Paula Antunes Lima
Departamento de Engenharia de Produção - UFMG

_____

Profª Drª Ana Valéria Carneiro Dias
Departamento de Engenharia de Produção - UFMG

_____

Profª Drª Cristina de Castro Frade
Centro Pedagógico - UFMG

AGRADECIMENTOS

Ao meu orientador, prof. Francisco, pelos ensinamentos, pelo carinho e pelo grande incentivo em me fazer acreditar que era possível alcançar aquilo que eu julgava ser inatingível. Com a sua competência e paciência, soube conduzir-me com segurança na realização desta dissertação, ampliando consideravelmente o meu crescimento profissional.

A todos os professores, funcionários e colegas do Departamento de Engenharia de Produção, especialmente a professora Ana Valéria pela sua disponibilidade e especial atenção.

Aos técnicos em informática da empresa pesquisada, pela disposição incansável em viabilizar o acesso a documentos e informações, além de permitirem as inúmeras observações de suas atividades realizadas em seu ambiente de trabalho.

Aos meus amigos de trabalho Maria das Graças, Vauelice, Luciana, Nara, Gabriela e Raquel, pela compreensão de minhas ausências e pelo apoio dado.

Ao Marcos Geraldo por sua compreensão, permitindo o meu ingresso e a conclusão do curso de mestrado.

À minha mãe e irmãos pelo apoio e vibrações positivas.

Ao meu filho Artur, que apesar da pouca idade, me trouxe luz e força de vontade nos momentos difíceis de incertezas e preocupações, trazendo-me a certeza de que as ausências de hoje serão compreendidas amanhã.

Finalmente, muito obrigada ao meu grande marido e constante revisor, que com muita dedicação leu meus textos e esteve sempre presente me ajudando e incentivando a seguir em frente.

RESUMO

A presente dissertação discorre sobre os problemas decorrentes da utilização de uma estrutura organizacional, adotada pela empresa analisada, denominada "fábrica de software", na qual utilizam-se métodos supostamente universais de engenharia, seguindo os modelos da organização industrial, numa tentativa de solucionar problemas da produtividade e melhorar a qualidade de seus produtos. No entanto, tais métodos tendem mais a resultados ineficientes do que a pretendida melhoria de qualidade.

A utilização de regras e modelos formais da Engenharia de Software “amputa” a dimensão imaginativa e social do profissional em informática, o que denota um profundo desconhecimento acerca de como os analistas adquirem o saber das necessidades reais do usuário, especialmente no que diz respeito ao saber que envolve o processo criativo.

Tendo como base o estudo empírico, realizado sob uma abordagem metodológica da análise ergonômica do trabalho, junto aos técnicos em informática de uma empresa prestadora de serviços públicos, aponta-se aqui a importância da socialização entre os agentes envolvidos, bem como toda a capacidade imaginativa dos técnicos em informática, fundamentais na construção de um software de qualidade.

O resultado do estudo demonstrou que a divisão de papéis adotada pelo modelo “Fábrica de Software”, aliada à terceirização do desenvolvimento de software, onde os desenvolvedores distam geograficamente da empresa analisada, apenas agravou os problemas já encontrados no modelo anteriormente utilizado. A separação física entre os desenvolvedores e o analista de negócio, bem como o isolamento destes do contexto da empresa, não permitiram que os desenvolvedores adquirissem o saber necessário para desenvolver um produto que atendesse as necessidades reais dos usuários.

Palavras chaves: fábrica de software, desenvolvimento de software, saber, socialização.

ABSTRACT

This paper address the problems of the organizational structure named "software factory" adopted by the company under analysis. This model utilizes alleged universal engineering methods, according to the fundamentals of industrial manufacturing, aiming to eliminate productivity issues and improve product quality. However, such methods have shown a trend of ineffective results rather than the expected quality improvement.

The utilization of formal Software Engineering regulations and models "mutilates" the social and creative dimension of the information technology personnel and indicates a profound lack of knowledge regarding the means used by the analyst in the identification of user requirements, mainly those related to the innovative process.

The empirical study based on the ergonomic analysis of the work performed by information technology analysts of a public services company demonstrates the importance of socialization in the work environment and the utilization of creative capacity in the development of higher quality software.

The study has also disclosed that the reassignment of roles dictated by the "Software Factory" structure coupled with the outsourcing of software development has further aggravated the problems of the model previously utilized. The transfer of this activity to a remote location has generated difficulties regarding the relationship with business analysts and the integration with company operations. Under these conditions, the programmers have not been able to acquire the required knowledge for the development of products that meet user requirements.

Key Words: software factory, software development, knowledge, socialization.

Lista de quadros:

Lista de figuras:

Lista de siglas:

AD - Administração de Dados

ABD - Administração de Banco de Dados

CMMI -Capability Maturity Model Integration

DS - Desenvolvimento de Software

ES - Engenharia de Software

FABRISOFT - Nome fictício da fábrica de Software terceirizada

FS - Fábrica de Software

IA - Inteligência Artificial

IEC - International Electrical Commission

ISO - International Organization for Standardization

PF - Ponto de Função

PMO - Project Manager Office

SDS - Setor de Desenvolvimento de Software

RUP -Rational Unified Process

XP - Extreme Programming

SUMÁRIO

Capítulo 1 - Introdução

Os ambientes de desenvolvimento de software vêm ao longo dos anos utilizando-se dos modelos industriais para melhorar a qualidade e rapidez na entrega de seus produtos. Assim, a Engenharia de Software (ES), na busca pela superação da natureza "artesanal" do Desenvolvimento de Software (DS), utiliza-se de modelos[1] supostamente universais da organização industrial. Essa sistematização da atividade de DS, diante dos conceitos genéricos das metodologias existentes, tende mais a resultados ineficientes do que à melhoria da qualidade do software. Um exemplo é o da empresa analisada que, na busca da melhoria de qualidade de seus produtos, adotou o modelo industrial da Fábrica de Software (FS), concomitante com a terceirização de alguns serviços. No entanto, a implantação do novo modelo organizacional, aliada à terceirização, não respondeu às expectativas de melhoria, inicialmente pretendidas. Ao contrário, os produtos não atenderam as necessidades do usuário, como ocorreu com o projeto denominado "reajuste tarifário", que será discutido nesta dissertação. Com o fracasso da implantação desses modelos, a empresa não encontrou alternativa senão a de retornar a sua antiga forma de trabalho, ainda que, para a gerência, essa opção seja em

[1] Boyer e Freyssenet (apud Zilbocivius, 87) apresenta a seguinte definição de modelos: "podemos considerar como modelos... processos periódicos que tornam internamente coerentes ou internamente compatíveis, e externamente apropriados, os elementos que estruturam a vida das empresas e as instituições que governam as relações de mercado e de salários". E completam: "um estado de coerência / apropriação será mais ou menos atingido e o grau em que o será vai depender da extensão em que [o modelo] assegura a viabilidade das empresas e das relações profissionais".

Boyer e Freyssenet (apud Zilbocivius, 87) ressaltam que "nenhum modelo foi ou está sendo reproduzido de maneira idêntica" e que um modelo certamente não é um sistema estável e fechado que se desenvolve ou desaparece apenas como resultado de constrangimentos externos".

caráter provisório até que se encontre uma solução que viabilize nova mudança, desde que guarde correspondência com o modelo industrial inicialmente adotado.

> "(...) na realidade estamos trabalhando da mesma forma que trabalhávamos antes. Mudam daqui e mudam dali e no final começaram tudo de novo". *(analista da SERVPUB)*

A ES, como ocorre nas ciências naturais, desconsidera a base cultural da construção do saber ao transformar o DS em regras universais. É desta forma que a atividade de criação software vem sendo tratada, qual seja, a partir da utilização de métodos universais de organizações industriais. Apoiando-se nas prescrições, a ES acredita que por meio de regras e modelos formais é possível transferir o saber dos usuários para os técnicos em informática. Neste sentido, muitas vezes colocam os desenvolvedores distantes dos membros da equipe e da empresa para a qual vão desenvolver o software, reduzindo ou até impedindo os contatos presenciais. Esta forma de organizar a atividade de DS demonstra desconhecimento profundo sobre como os analistas adquirem o saber das necessidades reais do usuário.

A separação presencial do desenvolvedor, suprimindo-o do convívio com o usuário, dificulta o entendimento das necessidades deste usuário, essencial para transformar o artefato em instrumento de trabalho[2]. Segundo Collins (1992), o saber[3], mesmo das regras formais e heurísticas (regras empíricas explicitáveis e de práticas padronizáveis), só é adquirido pelo convívio com os membros daquela cultura, essencial, no caso do DS, para a tradução das necessidades do usuário e para transformá-las em um software de qualidade, isto é, em um instrumento de trabalho.

---

[2] Segundo Rabardel (1995), o instrumento representa a combinação de um esquema e de um artefato. Este último refere-se a artefatos como, por exemplo, partes de uma interface computacional. O instrumento é definido como sendo uma entidade mista definida a partir da associação entre duas entidades. A primeira delas, o artefato, trata-se o meio através do qual o sujeito age. Os artefatos podem ser materiais ou não. Qualquer que seja sua estrutura, ele constitui-se enquanto elemento da cultura.

"(...) o artefato não é em si um instrumento ou componente de um instrumento (mesmo quando concebido para isso), sendo instituído como instrumento pelo sujeito que lhe dá o estatuto de meio para atingir o objetivo da sua ação" (Clot, 2006).

[3] O termo conhecimento e saber recebem diferentes significados, dependendo do contexto de utilização e do idioma. A língua inglesa utiliza o termo "conhecimento" enquanto a literatura em francês adota "saber". Nesta dissertação utilizaremos a palavra saber para descrever o saber aprendido socialmente seguindo a tradução francesa de Collins (1992). O tradutor francês verteu o inglês *know-ledge* por "saber" no lugar de "conhecimento", *"Esta escolha se justifica pela aproximação do termo saber com a dimensão social, mais do que o termo connaissance. [Essa] tradução é assim conforme o espírito que tem inspirado através de todo o trabalho o uso que o autor faz deste termo"* (Collins, 1992, p.15).

A interação presencial entre os envolvidos no desenvolvimento de software (técnicos em informática e usuários) facilita a troca de idéias, experiências e a construção de uma competência coletiva. No entanto, a tendência das organizações de desenvolvimento de software tem sido a de manter a separação entre os envolvidos (técnicos em informática e usuários).

Assim, a engenharia de software fez opção pela adoção de modelos industriais de produção, onde os métodos são efetivamente utilizados numa tentativa de antecipar e de controlar pessoas e resultados.

"No entanto, o projeto de industrialização contradiz a natureza da atividade dos profissionais de informática, que se assemelha mais a uma "criação artística" do que a um produto industrial, produzido em massa. O equívoco da adoção de modelos fabris para lidar com produtos cuja natureza é artística se assenta sobre a concepção errônea de que o trabalho do analista e a produção de software, em geral, seriam ainda artesanais. Compreende-se o raciocínio: assim como a indústria erguer-se-ia sobre as ruínas de uma era artesanal supostamente decadente e ineficiente, o paradigma fabril de eficiência e de sofisticação tecnológica deve também substituir a prática atrasada e ineficiente dos analistas artesãos. Na verdade, duplo equívoco sustenta as premissas desse silogismo: primeiramente, a atividade do analista não tem nenhuma similaridade com o trabalho artesanal, comparação possível apenas por um profundo desconhecimento da época pré-industrial; em segundo lugar, que a organização industrial tenha eliminado inteiramente a natureza artesanal do trabalho pré-fabril, atribuindo a eficiência da produção industrial inteiramente à organização formal (regras, padrões, divisão do trabalho, hierarquia etc.). O artesanato não foi superado pela indústria, pelos menos quando se considera a qualidade dos produtos, nem a prática artesanal deixou de existir na era industrial. Superado no sentido dialético, o artesanal passa a existir nos interstícios da produção de base tecnológica: nenhum produto ou sistema industrial funcionaria sem uma boa cota de trabalhadores "artesãos", que continuam bricolando suas "gambiarras" e desenvolvendo competências tradicionais, transmitidas pela socialização nos assépticos e formais ambientes industriais, no caso em laboratórios de alta tecnologia (ver Latour e Woolgar, 1997)" (Leal e Lima, 2006).

Este estudo pretende mostrar como os modelos organizacionais do trabalho (modelos fabris, etc) podem facilitar ou dificultar a tradução das necessidades reais dos usuários, imprescindível para transformar os artefatos em verdadeiros instrumentos de trabalho. A pesquisa foi realizada no setor de desenvolvimento de software de uma empresa prestadora de serviços públicos, juntamente com os profissionais de uma equipe que trabalha com um software cujo objetivo é atender, de forma integrada, uma rede de unidades com funções distintas: comerciais, operacionais e serviços de atendimento aos clientes da empresa. Os usuários do software desenvolvido por esses profissionais são os trabalhadores dos diversos setores da empresa. Os técnicos em informática da equipe trabalham aproximadamente há doze anos com o sistema.

Além deste capítulo introdutório, a dissertação contempla outros cinco capítulos. No capítulo 2, serão discutidos os problemas de desenvolvimento de software e a utilização dos modelos industriais para a solução destes problemas, que são reais. Este capítulo conta ainda com uma análise crítica sobre a concepção errônea de que o trabalho do analista e a produção de software, em geral, seriam ainda "artesanais". O capítulo 3 discorre sobre a análise crítica dos modelos industriais e sobre a visão tecnocêntrica da engenharia de software, a influência das ciências cognitivas na atividade de desenvolvimento de software, bem como a crítica de alguns teóricos sobre esta visão. O capítulo 4 discute sobre a construção social do saber dos analistas, imprescindível para traduzir e transformar as necessidades do usuário em um software de qualidade. O capítulo 5 detalha a análise ergonômica realizada no setor de desenvolvimento de software de uma empresa prestadora de serviços públicos, evidenciando as dificuldades decorrentes da opção da empresa pela adoção do modelo "fábrica de software" e pela terceirização de serviços, demonstrado com o caso do "reajuste tarifário", bem como uma discussão teórica acerca da viabilidade e conseqüências da adoção desses modelos (fábrica de software e terceirização). O capítulo 6 discute a face artística da atividade de desenvolvimento de software, mostrando que parte da atividade se constitui em verdadeiro trabalho artístico, consubstanciado na sensibilidade aguçada do analista para conseguir expressar as reais necessidades dos usuários e transformá-las em um software que servirá de instrumento de trabalho.

Os nomes dos técnicos em informática, da empresa pública, do setor e da empresa terceirizada, utilizados na dissertação, são fictícios.

Capítulo 2 - A crise do software

2.1 PROBLEMAS DO DESENVOLVIMENTO DE SOFTWARE

Os modelos industriais de organização, planejamento e gestão vêm sendo utilizados pelos fabricantes de software como tentativa de solucionar os problemas encontrados no processo de desenvolvimento de software, tais como, sistemas ineficientes e não funcionais, erros de programação, prazos ultrapassados no atendimento às demandas e orçamentos estourados, "*gerando frustração, perda de tempo em retrabalho e, inevitavelmente, conflitos e desgastes das relações entre os contratantes. Os estudos de avaliação das promessas da informática colocam em dúvida se os ganhos de produtividade são efetivos ou tão significativos como se pretende* (sobre isso, ver Lojkine, 1996; Dertouzos, 1997 e vários estudos em Kling, 1996)" (Ferreira e Lima, 2006).

No que chamou de "o canto da sereia comercial" Dertouzos (2000) escreve sobre os riscos de fracasso econômico que correm as empresas que invadem áreas alheias à sua especialidade, nas quais não possuem experiência, tentando oferecer de tudo, acreditando apenas nas promessas da informática - que cada vez mais vêm sendo questionadas - de ganhos ilimitados proporcionados pela criação de sistemas capazes de reduzir o tempo de produção e de possibilitar a utilização cada vez menor de mão de obra, aumentando a produtividade e, conseqüentemente, os lucros.

Diante das inúmeras dificuldades associadas ao desenvolvimento de software e do conseqüente descrédito das promessas da informática, muitos autores passaram a definir o conjunto de problemas como "crise". Pressman (1995, p. 22) prefere utilizar a palavra "aflição", por ser definida no Webster's Dictionary como "algo que causa dor e sofrimento" e, portanto, exige cura. Independentemente da denominação, seja ela crise ou aflição, os problemas são reais e existem em todas as empresas de desenvolvimento de software.

Assim, como ocorre em todas as empresas de DS, o setor de desenvolvimento de software (SDS) da empresa prestadora de serviço público analisada encontrava-se diante de uma "crise". Na opinião do gerente, o setor apresentava uma forma organizacional considerada "artesanal", o que, aparentemente, refletia em uma grande diversidade de problemas. De acordo com o diagnóstico apresentado pela gerência do setor, os profissionais da área eram submetidos constantemente à pressão temporal para finalizar os sistemas. Além disso, a empresa não mantinha a documentação do sistema em dia, o que a colocava em situação temerária, pois passava a depender da memória dos analistas e programadores. Quando alguém era remanejado da função ou se ausentava por motivo de férias, licença médica ou outro motivo qualquer, os demais profissionais, por desconhecerem o sistema, tinham dificuldades em dar continuidade ao trabalho.

Estes problemas não são específicos da empresa analisada, nem se explicam, embora pudessem ser agravados pela forma como as demandas chegavam ao setor, às vezes por imposição dos superiores hierárquicos, às vezes pela negligência do próprio setor e dos analistas em não documentarem os sistemas. De modo geral, esta é a realidade do desenvolvimento de softwares, já relatada há vários anos (ver Carvalho, 1988), e que perdura até hoje, sem que se entreveja uma solução.

> "(...) os prazos são estabelecidos de modo aleatório e, por conseguinte são descumpridos de modo escandaloso. Não se documenta e formaliza as coisas desde o começo (...)" (Carvalho, 1998, p.205)

2.2 O MODELO "FÁBRICA DE SOFTWARE": SOLUÇÃO INADEQUADA PARA UM PROBLEMA REAL

Os fabricantes de software acreditam que a industrialização da atividade de desenvolvimento de software, utilizando-se dos métodos da engenharia, resolverá os problemas que atingem o DS:

> "(...) há uma consciência bastante generalizada acerca da necessidade de transformar o processo de desenvolvimento de sistemas de um artesanato para uma engenharia verdadeira" (Carvalho, 1998).

O fundamento principal para a adoção deste modelo produtivo apóia-se na assertiva de que o trabalho de desenvolvimento de software é considerado "artesanal" e, por conseguinte, suscetível à industrialização.

Vários fabricantes de software (Ver Greenfield, 2007; Quidgest, 2007; Squadra, 2007 e Meta, 2007) divulgam e promovem propagandas de suas fábricas, utilizando o seu modelo organizacional como forma de superação do modelo "artesanal" e como garantia de qualidade de seus produtos. A fábrica de software utiliza processos formais da ES que "*são baseados em conceitos industriais de planejamento, controle e produção. Por isso difere dos antigos bureais (sic) de programação, que constroem software de maneira artesanal*" (Squadra, 2007).

Segundo Fernandes e Teixeira (2004), até meados da década de 80, o processo de desenvolvimento de software era totalmente artesanal no Brasil e foi a partir daí que tiveram início os processos disciplinados.

A expressão Fábrica de Software é utilizada desde os anos 60. "Cusumano foi um dos primeiros autores a divulgar o termo, a partir de suas pesquisas comparativas entre Estados Unidos e Japão, no final da década de 80, acerca de práticas de desenvolvimento de software" (Fernandes e Teixeira, 2004). Segundo Cusumano, (apud Nomura et al., 2006):

> "O sucesso da FS do Japão e dos Estados Unidos se deve à inclusão de um alto grau de reusabilidade, modularização, uso de ferramentas e controle e

> gerenciamento dos sistemas aumentando a qualidade e a flexibilidade. Os projetos de FS desenvolvidos mostraram que o ganho de produtividade da indústria japonesa pôde ser pela adoção de seus métodos de trabalho, com a simplificação, integridade conceitual, aderência aos padrões e automação seletiva no processo de desenvolvimento".

A SERVPUB está implantando o modelo "fábrica de software" em seu setor de desenvolvimento de software, seguindo a tendência atual do mercado, que busca novos padrões de racionalização da produção, inspirados em modelos fabris de gestão de projetos, certificação de garantia de qualidade do processo e na organização do processo de desenvolvimento segundo modelos industriais. *"O conceito de fábrica de software está baseado na idéia de prover uma linha de produção que atenda às necessidades específicas de cada cliente através da formalização de todas as atividades e seus produtos, trabalhando em linha de produção, com etapas e tarefas bem definidas para cada tipo de profissional, indo da produtividade da linha de produção a rotinas de controle de qualidade*" (Brito, 2004). Foi com base nesse conceito que o gerente do setor construiu o projeto de implantação da FS na empresa.

Fernandes e Teixeira (2004) definem fábricas de software como um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial e orientado para o atendimento a múltiplas demandas de natureza e escopo distintas, visando à geração de produtos de software, conforme os requerimentos documentados dos usuários e/ou clientes, da forma mais produtiva e econômica possível.

Assim, para esses autores *"Há várias conotações para a expressão Fábrica de Software. A maioria das empresas de serviços tem como Fábrica de Software a "Fábrica de Programas", dedicada exclusivamente para a codificação de programas*" (ibid p.115).

Dessa forma, a conotação para a expressão *"Fábrica de Software"* que será utilizada nesta dissertação é a mesma que lhe emprestam os autores Fernandes e Teixeira, também adotada pela empresa analisada. Para estes autores, a Fábrica de Software pode ter vários escopos de atuação, desde um projeto de software completo à codificação de programas de computador.

As fábricas de software são sempre inspiradas em modelos fabris de gestão de projetos, em certificação de garantia de qualidade do processo, tais como CMMI (Capability Maturity Model Integration) e NBR ISO 9000-3 e na organização do processo de desenvolvimento segundo modelos industriais, como o RUP (Rational Unified Process) ou XP (Extreme Programming). A adoção dos processos, que serão detalhados adiante, tem sido o objetivo padrão dos desenvolvedores de software para atender um mercado cada vez mais exigente de certificações, utilizadas como garantia de qualidade dos produtos.

Como na indústria, o planejamento, organização, controle e processos são utilizados pela Engenharia de Software, sendo considerados como o principal mecanismo para se obter qualidade e cumprir corretamente prazos e contratos. Em suma, destinam-se à resolução da "crise do software".

Seguindo os modelos industriais, as fábricas de software trabalham partindo do princípio de que os projetos devem ser realizados seguindo-se rigorosamente os processos formalizados. Processo, segundo Paula Filho (2001, p.11), "*é um conjunto de passos parcialmente ordenados, constituídos por atividades, métodos, práticas e transformações, usado para atingir uma meta. Esta meta geralmente está associada a um ou mais resultados concretos finais, que são os produtos de execução do processo*".

As pessoas, numa fábrica, estão agrupadas em estruturas funcionais, sendo que cada uma delas é responsável por um ou vários papéis ao longo do ciclo de desenvolvimento do serviço, conforme exemplificado no quadro 1.

Quadro 1: Divisão de funções do modelo do RUP

Funções	Papéis
Analistas de sistema: Os analistas de sistemas estão envolvidos principalmente na investigação de requisitos.	• Analistas de sistemas • Designer de negócios • Analista de processos de negócios • Especificador de requisitos

Desenvolvedores: Os desenvolvedores estão agrupados em papéis, envolvidos principalmente no designer e implementação de software.	• Designer de cápsula • Designer de banco de dados • Implementador • Integrador • Arquiteto de software • Designer de interface de usuário
Testadores: Os testadores são os que lidam com habilidades específicas exclusivas para teste. Observe-se que existem papéis adicionais envolvidos na disciplina de Teste e que juntos ampliam as habilidades básicas de outros papéis. Esses papéis adicionais podem ser encontrados nos demais papéis organizados pelas habilidades básicas que ele amplia (por exemplo, Gerente, Designer, Analista).	• Testador • Analista de teste • Designer de teste
Produção e suporte: São aqueles que não estão diretamente relacionados com a definição, gerência, desenvolvimento e teste de software, mas são necessários para o apoio do processo de desenvolvimento de software ou para produzir materiais adicionais necessários para o produto final.	• Escritor técnico • Administrador em sistema • Especialista em ferramenta • Desenvolvedor de cursos • Artista gráfico
Gerentes: Os gerentes estão organizados em papéis principalmente envolvidos na gerência e configuração do processo da engenharia de software.	• Engenheiro de processo • Gerente de projeto • Gerente de controle de mudança • Gerente de configuração • Gerente de montagem • Revisor do projeto • Gerente de testes

Papéis adicionais: São separados dos outros papéis porque eles não se encaixam em nenhum outro grupo de papéis	• Revisor • Coordenador de revisão • Revisor técnico • Parte interessada (Stakeholder) • Convidado

Fonte: RUP (Rational Unified Process) – versão 2003.06.12.01

"*Os papéis são unidades de responsabilidade que podem ser assumidas por um ou mais indivíduos. Esses papéis são usados para descrever funções lógicas" (Paula Filho, 2001). "Os papéis não representam pessoas, apenas descrevem as habilidades necessárias e as responsabilidades que elas têm, ao assumir determinado papel funcional. Um membro da equipe do projeto geralmente desempenha muitos papéis distintos*" (França e Silva, 2007).

Estabelecendo a crença de ser possível seguir-se rigorosamente os processos formalizados, a engenharia de software entende que um software pode ser desenvolvido por uma ou várias pessoas, admitindo inclusive que partes ou mesmo a integralidade do desenvolvimento de um projeto possa ser terceirizado. Portanto, ao assumir concomitantemente essa divisão de papéis e a formalização dos processos, o modelo da fábrica de software favorece a terceirização.

Demais disso, é possível verificar-se tipos diferentes de fábricas de software com escopos distintos de atuação. Fernandes e Teixeira (2004) denominam "fábrica de programas" aquela que tem por objetivo codificar e testar programas de computadores, e "fábrica de projetos" aquela que possui maior amplitude, indo além das atividades inerentes a fábrica de programas, admitindo, entre outras, fases como a pós-conceitual, de especificação lógica e detalhamento de solução.

Desde a década de 70 já se verificava uma tentativa, frustrada, de implementação da divisão do trabalho entre a concepção (tarefa de especificação) e a execução (tarefa de programação) inspirada no processo manufatureiro. Tavares (1983) já escrevia:

> "a distinção feita pelas empresas entre analistas e programadores prende-se mais aos aspectos formais que a uma diferenciação funcional. As especificações das funções não são claras, mesmo quando algumas subcategorias são criadas ((analistas de sistemas, analista de suporte, analista

> de O&M, programador sênior, júnior e trainee)) as especificações ora se sobrepõem ou se confundem".

No entanto, ainda hoje insistem nessa divisão do trabalho, a qual a Engenharia de software denomina de papéis. É visível que esta divisão só dificulta a atividade dos técnicos de informática. Na SERVPUB, para que estes pudessem levar a cabo suas atividades, deixaram de seguir a prescrição e desenvolveram um sistema de cooperação, interagindo entre si e com os usuários, durante todo o desenvolvimento, para conseguirem construir um software com melhor qualidade, isto é, que atendesse as necessidades do usuário. Portanto, a atual dificuldade encontrada pelos técnicos da SERVIPUB reside na separação geográfica existente entre os desenvolvedores e os analistas da empresa, que são submetidos a uma terceirização à distância. Entretanto, é importante esclarecer que as dificuldades dos analistas não se resumem apenas na distância geográfica, elas surgem também quando os analistas são submetidos a uma divisão do trabalho nos moldes sugeridos pela Engenharia de Software, fator determinante da atitude dos analistas da SERVPUB em deixar de seguir a prescrição e desenvolver por conta própria um sistema de cooperação.

Dessa forma, para se adquirir os saberes dos usuários, necessários para a construção do software, é de fundamental importância a interação entre os envolvidos dentro do contexto social e cultural do qual eles fazem parte. Segundo Collins (1992), este é o único caminho para se adquirir o saber-fazer cultural.

> "A pesquisa sobre o conhecimento científico tem ajudado a mostrar o quanto as atividades são sociais. A história e a sociologia do conhecimento científico mostraram que qualquer transferência científica, mesmo uma competência experimental radicalmente nova, deve ter interação social. Quando nós escrevemos ou falamos por telefone, mesmo todas as vezes que precisamos, nada se compara a uma visita à pessoa com a qual você quer aprender alguma coisa e com a qual é preciso estabelecer um contato social" (Collins, 1992, p. 16, tradução da autora).

Assim, fica visível que a atividade de DS exige uma interação presencial entre os envolvidos para a construção do saber. No entanto, a Engenharia de Software não tem essa percepção da atividade de DS, que é tratada como se fosse puramente lógica. Parece que, diante do fato de seu produto (software) ser lógico, ele, por conta disso, contaminaria a Engenharia de Software ao considerar que o seu desenvolvimento também é lógico.

2.3 A ATIVIDADE "ARTESANAL" DO DESENVOLVEDOR DE SOFTWARE

As fábricas de software podem assumir configurações diferentes, mas um dos pontos comuns ao modelo é que todas as propostas assumem como pressuposto que a atividade do analista é semelhante à dos artesãos, causa fundamental dos problemas mencionados (sistemas ineficientes e não funcionais, erros de programação e prazos ultrapassados no atendimento às demandas, orçamentos estourados, frustração, perda de tempo em retrabalho, conflitos e desgastes das relações com os contratantes), que serão prontamente resolvidos pela industrialização. O trabalho de desenvolvimento de software é considerado "artesanal" pelo fato de os produtos inteiros serem criados desde o início por um indivíduo ou pequenas equipes, *"sem documentar precisamente etapa por etapa todas as idéias e soluções parciais que vão surgindo ao longo do trabalho de elaboração de um sistema de informações"* (Carvalho, 1988, p.149), tornando esse(s) trabalhador(es) o(s) único(s) conhecedor(es) dos processos da construção do software.

No entanto, a atividade do analista não guarda qualquer similaridade com o trabalho artesanal, comparação possível apenas quando se desconhece a produção da época pré-industrial. Para se estabelecer uma diferenciação consistente entre o trabalho artesanal e o trabalho de desenvolvimento de software é importante conhecer melhor as características do trabalho artesanal (Leal e Lima, 2007).

O artesanato foi a forma de produção característica da Idade Média, na qual o artesão detinha os meios de produção (era o proprietário da oficina e das ferramentas) e trabalhava com a família em sua própria casa, "*conduzindo todas as fases de produção de um objeto, desde a concepção até sua execução final*" (Vargas, 1979).

O trabalho artesanal segue rigorosamente os padrões cristalizados nos costumes, que são transmitidos de geração a geração, cujas técnicas e experiências são acumuladas em um ofício.

> "Os trabalhadores numa oficina estavam organizados de forma hierárquica, indo do aprendiz ao mestre, em vários níveis, até atingir o chefe. Todos viviam na mesma casa, participavam da vida do patrão até adquirirem também o grau de mestre, através de um trabalho considerado de alta qualidade" (Canêdo, 1987).

"Ao contrário da divisão hierárquica do trabalho na indústria, no início existe uma divisão do trabalho entre o aprendiz e o mestre artesão, mas ao final do processo de aprendizagem eliminam-se as diferenças e a hierarquia" (Lima, 1998).

O trabalho artesanal segue uma tradição, o artesão não detém uma marca individual ou autoral como acontece no caso do analista de sistemas. Segundo Alexander (1969), o artesão normalmente reproduz o que lhe foi transmitido pela tradição. Não é necessária a introdução de inovações sem forte razão. As pequenas mudanças ocorrem como um processo de adaptação, em virtude de alguns ajustes necessários para sanar os defeitos que impõem correção.

O princípio básico da adaptação depende do fato de que o processo que faz o equilíbrio é irreversível. O desajuste proporciona um incentivo para mudanças, enquanto o bom ajuste não pede mudanças. Na teoria, o processo está obrigado a chegar, com o tempo, ao equilíbrio das formas bem ajustadas (Alexander, 1969).

Entretanto, para que o ajuste se dê na prática, deve-se satisfazer uma condição de importância vital, qual seja, contar com o tempo para que ocorram os ajustes necessários àquele contexto. O processo deve ser capaz de alcançar seu equilíbrio antes que a próxima mudança cultural o transforme novamente. Concretamente, o processo conta com o tempo para alcançar seu equilíbrio cada vez em que é perturbado. Se vemos o processo como algo contínuo e não intermitente, o ajuste deve encerrar-se antes que ocorra nova mudança do contexto cultural. Sem estas condições, o sistema nunca vai produzir formas bem ajustadas, pois o equilíbrio da adaptação não se sustentará. (ibid).

As mudanças tecnológicas do mundo atual ocorrem com mais velocidade que nunca, não existe mais tempo para a adaptação, enquanto que no período da Idade Média os artesãos podiam descansar sobre os ombros dos seus antecessores; o conjunto de tradições facilitava todas as suas decisões. As rápidas mudanças do mundo atual impedem as integrações dos técnicos em informática com os seus antecessores, não é mais possível copiar, as inovações exigem novas formas de produção e assim o indivíduo fica cada dia mais isolado (Ver Alexander, 1969). O ritmo acelerado das mudanças técnicas na informática é fator preponderante para impedir que uma tradição se consolide, o que diferencia ainda mais a atividade do analista da do artesão e o leva a deixar no produto marca pessoal, autoral.

Na produção de software, diferentemente da artesanal, o analista não sabe aonde vai chegar, está constantemente diante de novas situações. O profissional adquire o domínio do seu produto com o término de sua criação, não dispondo de padrões preestabelecidos, uma vez que cada software é desenvolvido em um processo produtivo diferente, único e específico que surge para atender as diversas demandas apresentadas no decorrer de sua produção. Por isso, o software se assemelha mais a uma obra de arte (Lévy, 1992) que a um produto industrial, produzido em massa.

Lévy mostrou que a programação não é tão simplista como coloca a psicologia cognitiva. O curso da programação é sinuoso e imprevisível. Os programadores encontram vários problemas e imprevistos que os guiam sobre novos caminhos. Negociam os constrangimentos que apareceram e saltam de uma singularidade a outra.

> "Não se pode confundir o processo com o produto. É o computador que deve seguir uma série de instruções numa ordem inflexível, não o programador. Este último deve, sem dúvida, respeitar a sintaxe da linguagem de programação e produzir um texto coerente. Estes constrangimentos são mais rigorosos que aqueles aos quais todas as pessoas que escrevem estão submetidas" (Lévy, 1992, p.241, tradução da autora).

Não se pode criar um padrão seguindo padrões. A atividade que produz regularidades e resultados uniformes não é regida por regras pré-definidas (Lima, 2005).

A programação não segue método justamente porque ela é um processo de escritura. Segundo Lévy (1992), a programação é efetivamente um gênero ou uma espécie de escritura, que utiliza uma ideografia ou um sistema semiótico simplesmente um pouco mais rígido do que o que se emprega para produzirem romances. Como no processo de escritura de um romance, cada desenvolvimento de um software é uma criação e por isso ele é único.

O próximo capítulo trás uma a análise crítica dos modelos industriais e discorre sobre a visão racionalista da engenharia de software, que possui a crença de que por meio do controle dos métodos predefinidos é possível produzir os resultados esperados. Além disso, discute também sobre a influência das ciências cognitivas na atividade de desenvolvimento de software, bem como aponta a crítica de alguns teóricos sobre esta visão.

Capítulo 3 - Críticas aos modelos industriais

A racionalização da produção vem sendo questionada pela psicologia, etnogragia industrial e sociologia. A ergonomia e a ergologia demonstram a existência de uma distância ineliminável entre as regras formais e a realidade da produção. Mesmo nas atividades "regulares" da indústria, a prática se diferencia da regra. A ergonomia da atividade evidencia que as diferenças entre norma e atividade persistem mesmo em situações de trabalho extremamente taylorizadas (Lima, 2005).

Como ocorre nas indústrias, as empresas de software buscam a melhoria contínua de seus processos, como se fosse possível reduzir a diferença entre o formal e o real. Além do mais, na atividade de desenvolvimento de software a variabilidade decorre do próprio tipo de trabalho, a atividade é criativa, única no seu desenvolvimento, assim torna-se insustentável pensar em padronizar uma atividade cuja natureza é criar padrões. Criar padrão é estabelecer regras para determinada atividade. As atividades diferem entre si, sendo necessária, assim, a criação de padrões distintos que atendam às especificidades de cada uma delas, o que inviabiliza por completo a padronização de uma atividade que tem por objeto a criação de padrões para diversas outras.

Em contraposição aos modelos industriais e a forma de racionalidade mecanicista, a atividade de desenvolvimento de software é criativa e única. O profissional envolvido nessa atividade somente adquire o domínio do seu produto com o término de seu trabalho. Este profissional não sabe aonde vai chegar, não dispõe de padrões preestabelecidos, uma vez que cada software é desenvolvido em um processo

produtivo diferente, único e específico. Cada software é desenvolvido para atender a uma demanda específica, direcionada a determinados usuários, que fazem parte de uma comunidade e que detêm uma cultura própria.

Percebe-se, atualmente, que a própria indústria faz emergir um novo modelo organizacional que busca superar a organização taylorista/fordista, pois a mercadoria padrão da era Ford não gera mais lucros extraordinários para as empresas, que devem inovar continuamente para sobreviver em mercados saturados (Leal e Lima, 2007).

Mesmo diante das críticas ao modelo industrial e da tentativa da própria indústria em encontrar novos modelos organizacionais, os fabricantes de software insistem em transformar o processo "artesanal" em modelos industriais.

3.1 PROCESSOS E CERTIFICAÇÕES: UMA FALSA ILUSÃO DE QUALIDADE

A Engenharia de software é disciplina nova que, como outras, ainda sustenta uma visão racionalista do mundo, dando continuidade à existência da crença de que por meio do controle dos métodos predefinidos é possível produzir os resultados esperados.

Uma primeira definição de engenharia de software, que espelha bem o pensamento ainda hoje dominante, foi proposta por Fritz Bauer (apud Pressman, 1995, p.31) na primeira grande conferência [NAU69] dedicada ao assunto:

> O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.

Seguindo o mesmo entendimento, a engenharia de software, para Pressman (1995), abrange um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos. As ferramentas proporcionam apoio automatizado ou semi-automatizado aos métodos. Os procedimentos constituem o elo de ligação que mantêm juntos os métodos e as ferramentas. Os métodos, para esse autor, proporcionam os detalhes de "como fazer" para construir o software.

> "A engenharia de software é construção muito recente, pois o próprio software passou a ser reconhecido como artefato há aproximadamente meio século. O conceito de software não surgiu imediatamente após o aparecimento dos primeiros computadores eletromecânicos e das válvulas, na década de 1940. Naqueles computadores, como ainda não havia o conceito de instruções armazenadas na memória, a programação era algo muito próxima do hardware, normalmente sendo realizada pelas mesmas pessoas que projetavam, construíam e operavam o computador. A programação consistia em ligar/interromper a conexão de fios ou ligar/desligar chaves em um painel do computador. Era um cenário em que não existia o entendimento da programação de hoje, como algo separado do hardware. Por isso, até mesmo o surgimento dos cartões perfurados para a programação de computadores representou um expressivo avanço na área" (Teixeira, 2007).

A ES, ao considerar a atividade de DS como um processo lógico, busca cada vez mais a melhoria de seus processos para padronização final de sua atividade, conforme os modelos industriais. Assim a ES adota a premissa na qual quanto mais definidos forem os processos de DS maior será a garantia de qualidade do produto.

O mercado atual, com base na perspectiva da engenharia, vem cada vez mais exigindo certificações como garantia de qualidade. Portanto, para que os fabricantes consigam as certificações existentes, os processos de desenvolvimento de software devem ser bem definidos.

> "Um processo é definido quando tem documentação que detalha: o que é feito (produto), quando (passos), por quem (agentes), as coisas que usa (insumos) e as coisas que produz (resultados). Processos podem ser definidos com mais ou menos detalhes, como acontece em qualquer receita. Os passos de um processo podem ter ordenação apenas parcial, o que pode permitir paralelismo entre alguns" (Paula Filho, 2001, p.11).

A existência de processos é imprescindível para as certificações de qualidade de software. A prática de certificação se baseia nos modelos industriais, os quais seguem a denominada Gestão da Qualidade Total, com princípios de melhoria contínua de seus processos. Um exemplo é o CMMI, uma evolução do CMM, que "*foi baseado em algumas das idéias dos movimentos de qualidade industrial das últimas décadas. Destacam-se entre elas os conceitos de W. E. Deming, que também teve grande influência na filosofia japonesa de qualidade industrial. Esses conceitos foram adaptados para a área de software por Watts Humphrey [Humphrey90]*" (Paula Filho, 2001, p.10).

Outro exemplo é a ISO (*International Organization for Standardization*) que tem como objetivo promover o desenvolvimento de padronização, se constituindo em

uma família composta de um conjunto de normas. Os padrões da ISO estabelecem especificações técnicas, regras e critérios. Atualmente a versão 2000 da ISO 9001 combinou três normas (9001, 9002 e 9003) em apenas uma, agora denominada de 9001. A implementação da ISO é destinada a "produtos" que podem se constituir em um objeto físico, um serviço ou um software.

O modelo ISO/IEC 12207 (IEC significa *International Electrical Commission* e atua, juntamente com a ISO, na área da tecnologia da informação) faz parte do conjunto de normas ISO. O modelo, "*conhecido no Brasil como [ABNT987], descreve os principais processos no ciclo de vida do software, e os relacionamentos entre eles*" (Paula Filho, 2001, p. 81).

O ISO/IEC 15504, também conhecido como Spice (Software Process Improvement and Capability Determination) tem raízes do governo britânico ao definir um processo de avaliação de *software*, visando ajustar expectativas de contratação junto a provedores de *software* e serviços. Esta norma tem as seguintes finalidades: (1) determinar a capacidade de um fornecedor de *software;* (2) auxiliar na melhoria do processo de *software*; e (3) servir como instrumento de auto-avaliação (Fernandes e Teixeira, 2004).

As empresas de DS, para conseguirem as certificações de qualidade, utilizam-se de modelos de processos existentes no mercado atual. Apesar de os processos apresentarem algumas diferenças. Uns são considerados mais rígidos, como o RUP, outros mais maleáveis e ágeis, como o XP. Entretanto, ambos explicitam papéis, estabelecem regras e normas rígidas. O RUP tem uma abordagem voltada para o planejamento detalhado, com fases seqüenciais de processo e artefatos evoluindo de uma fase para a seguinte. Este processo tem recebido críticas por ser notadamente burocrático e por apresentar vasta documentação.

Os métodos tradicionais, como o RUP, trazem problemas de comunicação ao adotar o processo de desenvolvimento seqüencial e linear, separando a concepção da operação e impedindo a mudança do escopo inicial. Desta forma, o modelo exige grande volume de documentação e por isto é taxado como burocrático. A falta de contato direto entre os técnicos de informática e usuários, necessário para um melhor

entendimento das necessidades reais do usuário, se constitui em um dos principais fatores que levam à construção de softwares ineficientes.

Ao contrário dos métodos tradicionais, os métodos ágeis, como o XP, procuram desenvolver o software com o mínimo de documentação. Assim, prefere a comunicação face-a-face, durante o desenvolvimento, que a documentação. Possui uma fase de especificação mais curta, cuja regra é a utilização de códigos simples e genéricos para serem utilizados como documentação. No entanto, o XP também é alvo de críticas, por retroceder ao método de tentativa e erro.

Apesar de os métodos ágeis aumentarem a participação do usuário durante toda a atividade de desenvolvimento de software, sua fase de especificação é curta e o tempo de levantamento de dados não é suficiente para entender as necessidades do usuário. Dessa forma o retrabalho é inevitável. Na realidade, o método de tentativa e erro, alvo de crítica do XP, é um retrabalho.

Mesmo com suas diferenças, os métodos consideram a atividade lógica, como se fosse possível ao usuário informar todas as suas necessidades e ao analista de negócio possuir o completo entendimento de todas elas. É fato que o usuário nunca se lembra de todas as informações específicas de sua atividade. Além disso, para se ter o entendimento dessas informações é preciso uma vivência com a prática diária, "*não é possível alguém que não conhece nada da empresa* (SERVPUB) *chegar aqui* e *conseguir fazer um software de qualidade"* (Analista da SERVPUB).

3.2 A OBSESSÃO PELOS PROJETOS

A sociedade pós-industrial acredita que as dificuldades de controle, verificadas com a ocorrência de imprevistos, encontrados desde o período taylorista e fordista, poderão ser sanadas com a antecipação das situações pelas experiências passadas. As condutas antecipadoras conduzem a uma "obsessão projetiva".

> "Encontramos essa obsessão especialmente nos novos profissionais que desempenham a função de conselheiros de projetos, mormente quando, ao se dirigirem a atores em posição frágil e precária, exigem de seus orientados

> uma lucidez e uma transparência que eles próprios muitas vezes são incapazes de aplicar ao seu próprio devir!" (Boutinet, 2002, p.14).

Ao sabor dessa ilusão, a Engenharia de Software se propaga acreditando na possibilidade de se ter uma visão das situações futuras e assim poder formalizar e controlar as atividades de desenvolvimento de software. O imperativo moderno da representação e formalização está explícito nas inúmeras linguagens, métodos e ferramentas.

> "A perspectiva do controle, algo central no pensamento modernista, jaz na disciplina dos processos de software, herdeiros das idéias de burocracia, que considera possível prever, predeterminar e explicitar papéis, estabelecer regras que garantam o comportamento esperado e a coordenação central necessária. Normalmente parte-se do pressuposto, nos projetos de software, de que é possível saber antecipadamente o que deve ser feito, existindo pouca incerteza acerca das tarefas" (Dahlobom e Mathiassen, 1993, p.16 apud Teixeira, 2007).

A pressuposição subjacente é de que, *a priori*, existe um mundo ordenado bastando ao engenheiro de software descobrir/capturar os requisitos preexistentes, formalizar uma especificação e desenvolver o sistema desejado a partir dela. A maioria das abordagens tende a considerar que é possível, de antemão, definir os requisitos e que eles se manterão estáveis ao longo do desenvolvimento (ibid).

Verifica-se que o projeto se resume a mais uma esperança de se controlar o que é incontrolável. Entretanto, no dizer de Boutinet (2002, p.273), *"(...) o projeto constitui uma maneira de dar novamente esperança aos atores, mas também o risco que essa esperança seja ilusória*".

Diferentemente do que a cultura do projeto define como antecipação, Schwartz (1998) demonstra que é possível antecipar algumas situações que ainda estão por vir por meio de um trabalhador competente, na acepção popular do termo, que tenha se imbuído de uma cultura histórica, se imiscuído na sua rede e nos seus sinais, e, portanto, que ela tenha adquirido, para ele, um *valor patrimonial* [4]como campo pertinente de *sua* atividade e de *sua* vida.

3.3 A VISÃO RACIONALISTA DA ENGENHARIA DE SOFTWARE

A idéia de utilizar processos, métodos e estabelecer regras é considerar que a atividade de desenvolvimento de software pode ser prevista ou predeterminada. *"Pensa-se que as descrições passadas nos colocam em posição de predizer o futuro por exploração*" (Collins, 1992, p.66).

A Engenharia de software discute, quase ou exclusivamente, sobre assuntos técnicos, tais como: processos, padrões e métodos, considerando-os como os responsáveis pela qualidade do software, deixando os assuntos não técnicos (questões sociais, culturais, políticas, organizacionais e econômicas) para outras disciplinas (ver Teixeira, 2007). São campos distintos, que se misturam apenas provisoriamente, seja porque a ES ainda não se desenvolveu suficientemente, seja porque os aspectos irracionais (poder, medo, resistência e outros) do comportamento humano perturbam a racionalidade lógica do DS.

A Engenharia de software, como toda produção científica e tecnológica ocidental, tem uma visão mecanicista e tecnocêntrica que ainda persiste na ciência moderna.

"A visão mecanicista do mundo foi desenvolvida por grandes filósofos do século XVII, dentre os quais despontam René Descartes, Newton e Leibniz. Descartes foi o pai do racionalismo na filosofia. Seu método foi o instrumento matemático da dedução pura. Dos seus vários ensinamentos, o novo racionalismo e o mecanicismo foram, indubitavelmente, os que tiveram maior influência. E seus princípios cartesianos foram adotados, de uma forma ou de outra, pela maioria dos filósofos do século XVII. A inspiração do Iluminismo proveio, em parte, do racionalismo de Descartes, Spinoza e Hobbes, mas os verdadeiros fundadores do movimento foram Newton e Locke" (Lévy, 1998).

A ciência cognitiva clássica, que ainda tem grande influência na tecnologia da informação, surgiu através da cibernética seguindo o mesmo pressuposto lógico-

---

[4] valor patrimonial é a expressão que designa o conjunto de ingredientes sociais e históricos adquiridos por um indivíduo ao longo do tempo e da sua história.

matemático. A cibernética nasceu em 1943 e foi a partir dela, que alguns anos depois, surgiu o cognitivismo. No período da cibernética vários avanços ocorreram para o surgimento dos primeiros computadores. Os cibernéticos "*construíram os fundamentos da inteligência artificial. Introduziram os conceitos e o formalismo lógico-matemático nas ciências neurológicas. No plano mais geral, a cibernética tem deixado profundas marcas em disciplinas científicas, desde a biologia molecular até a sociologia, passando pela psicologia. O movimento cibernético exerceu um papel considerável no estabelecimento do paradigma computacional que pretende tornar-se hoje a linguagem dominante da ciência*". (Lévy, 1998, p.88/89).

Todos os precursores da cibernética enxergavam o mundo do ponto de vista lógico. Talvez isto decorra do fato de todos eles serem matemáticos, à exceção de Mac Culloch, que apesar de não possuir a referida formação, interessou-se pela matemática desde o início de sua carreira (Lévy, 1998).

Toda linha de pensamento cibernético, em sua origem, contém a marca da lógica matemática. Wiener iniciou sua carreia publicando alguns trabalhos de lógica. Em sua obra prima denominada *cybernets*, publicada em 1948, destaca no tópico introdutório que a "influência ciclicamente repetida da lógica matemática levou ao mesmo tempo à mecanização ideal ou real dos processos de pensamento". A costumeira remissão a Leibniz, idealizador concomitante do *calculus ratiocinator* e da "comunicação das substâncias", apenas confirma com maior precisão esse ponto de vista (Lévy, 1998).

> "O empreendimento da cibernética e da inteligência artificial deve muito à filosofia do primeiro Wittgenstein, pois ele foi o primeiro a construir o mundo no qual se estabelece. Os cibernéticos, porém, concebem o homem como um autômato lógico que processa informação. Detiveram-se, pois, no dizível e, esquecendo quem eram, desprezam o inexprimível mostrado por Wittgenstein. Por isso, mesmo que a cibernética e seus rebentos dependam de Wittgenstein, nem sempre são dignos dele. Instalaram-se definitivamente na lógica, no cálculo e na informação, quando Wittgenstein pedia que se os utilizasse para ir adiante. O limite entre o dizível e o indizível continua sendo, ainda hoje, um desafio. A cibernética triunfante dos anos quarenta e cinqüenta fingia nem sequer percebê-lo e, para sê-lo, entrou no jogo das operações, a ronda das traduções e o caleidoscópio lógico dos eventos" (ibid, p.100).

> "Assim como 1943 foi sem dúvida o ano no qual nasceu a cibernética, 1956 foi claramente o ano que deu origem ao cognitivismo. Durante este ano, em dois encontros realizados em Cambridge e Damouth, novas vozes como as de Hebert Simon, Nam Chomsky, Marvin Minsky e John M. C. Carthy expuseram idéias que viriam a se tornar as linhas mestras das ciências cognitivas modernas" (Varela, Thompson e Rosch, 2003, p.55).

O cognitivismo

A intuição central por detrás do cognitivismo é que a inteligência – incluindo a inteligência humana – assemelha-se a um computador em suas características essenciais de que a cognição pode ser efetivamente definida como computações de representações simbólicas. Os cognitivistas afirmam que a única forma pela qual podemos explicar a inteligência e a intencionalidade é por meio da hipótese de que a cognição consiste na ação baseada em representações fisicamente realizadas sob a forma de um código simbólico no cérebro ou em uma máquina (Varela, Thompson e Rosch, 2003).

Resumindo as questões fundamentais do cognitivismo, pode-se dizer que a cognição é o processamento de informações sob a forma de computação simbólica – manipulação de símbolos baseada em regras. A cognição funciona por meio de qualquer aparato que possa abrigar e manipular elementos funcionais discretos – os símbolos. O sistema interage somente com a forma dos símbolos (seus atributos físicos) e não com o seu significado. Não há, na hipótese cognitivista sobre o conhecimento, quaisquer tratamentos ao nível semântico. O sistema funciona quando os símbolos representam de forma adequada algum aspecto do mundo real e o processamento de informações leva a uma solução bem-sucedida do problema proposto ao sistema (ibid).

O cognitivismo produziu manifestações em várias disciplinas, mas em nenhum outro lugar tais manifestações foram mais visíveis do que na Inteligência Artificial (IA), que é a implementação literal da hipótese cognitivista.

Grandes nomes da inteligência artificial, como Simon, Papert e Minsky, entendem que as "*atividades cerebrais complexas, não rotineiras, não formalizadas por regras explícitas, são passíveis de redução a atividades simples e programadas*" (Lojkine, 1995, p. 128). Assim, "*a complexidade dos processos de resolução de problemas resulta de interações relativamente simples de um grande número de elementos de base extremamente simples. Cada problema engendra sub-problemas, até que encontremos um sub-problema que saibamos resolver, para o qual temos já um programa na memória*" (H. A. Simon, 1980 apud Lojkine, 1995, p.128).

O conexionismo

Após 25 anos de dominação da ortodoxia cognitivista, emerge, na década de 70, o movimento conexionista que, enxergando a conexão numa contrapartida mais biológica, influenciada por descobertas da neurociência a respeito do funcionamento do cérebro, passa a explicar a cognição como uma propriedade emergente do funcionamento global de uma rede neural.

Sinteticamente, para o conexionismo, um sistema cognitivo é um conjunto de elementos simples e não inteligentes que, como os neurônios, exprimem propriedades globais interessantes quando são interligados. A cognição é a emergência de estados globais em uma rede de componentes simples. As regras locais gerenciam as relações entre os indivíduos e as regras de mudanças gerenciam as relações entre os elementos. O sistema funciona quando as propriedades emergentes e a estrutura daí resultante são identificáveis com uma solução adequada para uma tarefa dada.

3.4 CRÍTICAS ÀS CIÊNCIAS COGNITIVAS TRADICIONAIS

Alguns teóricos acreditam que o "modelo conexionista" é uma transformação decisiva na estrutura conceitual da ciência cognitiva e "*que muitas das dificuldades teóricas e filosóficas que afligem o cognitivismo "clássico" podem ser evitadas, em parte ou no todo, por meio de um exame atento da nova alternativa Conexionista"* (Button, G. et. al, 1998, p.147).

No entanto, os parâmetros principais do "modelo conexionista" não se diferem muito do modelo cognitivista. Se os cognitivistas dizem que cada item físico corresponde a um item externo, um objeto, um símbolo, os conexionistas sustentam que o significado não está localizado em símbolos particulares, mas sim em componentes mais elementares que os próprios símbolos – sub-símbolos. O conexionismo é também idealizado e referido como "paradigma sub-simbólico". No entanto, o paradigma conexionista não rompe com o modelo de processamento descritivo simbólico, uma vez

que os símbolos continuam presentes como representações do mundo objetivo, porém distribuídos nas sinapses neurais.

Nesse sentido, tanto o cognitivismo quanto o conexionismo compartilham da visão na qual o mundo é predeterminado, que é possível prever as situações que ainda estão por vir. Consideram que a partir da abstração da realidade seja possível construir representações, tal como pressupõe a visão do mundo objetivista que permeia muitas áreas da ciência, inclusive a engenharia de software.

A visão do mundo estritamente calculável, considerando o homem como ser puramente lógico, é irreal. Esse mundo "*exclui por baixo nosso meio de vida: o espaço, as cores, os sons, as línguas que formam um meio para mensagens eventuais e, por fim, toda a espessura sensível do mundo. Exclui, por cima, o porquê vivemos e todos os significados em geral*" (Lévy, 98, p.100).

A suposição por detrás desses diferentes tipos de realismo cognitivo (cognitivismo, propriedades emergentes e conexionismo) "*é que o mundo pode ser dividido em regiões discretas de elementos e tarefas. A cognição consiste na resolução de problemas que deve, para ser bem-sucedida, respeitar os elementos, as propriedades e as relações dessas regiões predeterminadas*" (Ibid, p.155).

O capítulo seguinte descreve as contradições entre as regras lógicas da engenharia de software e como os analistas adquirem o saber imprescindível para traduzir e transformar as necessidades do usuário em um software de qualidade.

Capítulo 4 - Saber Social, cognição situada e regras lógicas

A engenharia de software trata a atividade de desenvolvimento de software como uma resolução de problemas, da mesma forma que a ciência cognitiva. "*A resolução de problemas, sob a ótica da psicologia cognitiva, é um processo de exploração ativa do "espaço do problema" constituído de etapas (situação inicial, sub-objetivos e objetivo final). Assim, para se resolver efetivamente um problema considera-se necessário determinar uma seqüência de operações que transformam a etapa inicial em objetivo final*" (Lévy, 1992, p.44, tradução da autora).

A atividade de DS não produz resultados uniformes e, por isso, torna-se impossível obter representações de uma situação que ainda não aconteceu. As próprias atividades "regulares" apresentam variabilidades, o que as diferenciam das regras formais. Pensar na concepção de um software apenas como uma "resolução de problemas" denota uma visão errônea da atividade real dos analistas e programadores. Várias são as situações imprevisíveis e os ajustamentos que ocorrem durante e após o desenvolvimento de um software, o que torna impossível a previsão, em sua concepção, de todas as situações, mesmo porque elas são únicas em cada desenvolvimento.

Outro não é o entendimento de Lévy (1992, p.44-5), senão vejamos:

> "Estudando em detalhe a maneira pela qual um profissional de informática resolve o «problema» que é a concepção de um software, nós constatamos que as coisas não se passam exatamente como pretende a teoria. Primeiramente, não somente o objetivo a atingir não é perfeitamente definido no início, mas ele sofre ao longo do desenvolvimento vários remanejamentos essenciais. Em seguida, os objetivos intermediários que levam ao objetivo final podem adquirir, no decorrer dos processos de resolução de problemas,

> um peso tal que todos os termos do problema são transformados" (tradução Leal e Lima).

O cognitivismo ao abordar a cognição como resolução de problemas *"funciona, até certo ponto, para domínios de tarefas dos quais é relativamente fácil especificar todos os estados possíveis. (...) Entretanto, para domínios de tarefas menos circunscritos ou mais indefinidos, esta abordagem tem-se mostrado significativamente menos produtiva"* (Varela; Thompson e Rosch, 2003, p.155).

As ciências cognitivas, como várias outras ciências, se tornam insuficientes por não considerarem as situações reais e, por corolário, não compreendem a maneira pela qual o trabalhador constrói o problema. E não são poucas as variabilidades encontradas pelo trabalhador, no curso de sua atividade, que podem interferir na construção do problema. Essas variações podem ser de natureza externa ou interna em relação ao indivíduo. Wisner (1995, p.135,136) denomina estas variabilidades de condições secundárias (variabilidade da tarefa, instabilidade das equipes, complexidade dos sistemas) e de condições próprias o funcionamento humano (sexo, idade, dimensões antropométricas, força física, suas deficiências e seu estado de saúde).

4.1. A COGNIÇÃO SITUADA

Na esteira destas reflexões teóricas, autores como Varela, Thompson e Rosch (2003), questionam explicitamente a pressuposição, prevalente nas ciências cognitivas como um todo, de que a cognição consiste na representação de um mundo, que é independente de nossas capacidades perceptivas e cognitivas, por um sistema cognitivo que existe independente desse mundo. Ao contrário, demonstram que a cognição é um fenômeno situado e explicam a cognição através da teoria da "enação".

> "Na enação, o ambiente em que se vive de forma alguma poderia ser pré-determinado, tal como pressupõe a visão do mundo objetivista. Ao contrário, esse "background[5]" tem fronteiras ilimitadas, exigindo e confirmando que a cognição é um processo "criativo" desenvolvido no indivíduo dependente do nicho onde ele vive. De fato, se desejamos recuperar o senso comum, então devemos inverter a atitude representacionista e tratar o conhecimento

[5] Este termo é utilizado por Varela, Thompson e Roche (2003) para conotar o conhecimento do senso comum envolvido nas ações do cotidiano de um ser vivo.

> dependente do contexto, não como um artefato residual que pode ser progressivamente eliminado pela descoberta de regras mais sofisticadas, mas como, na verdade, a própria essência da cognição criativa" (ibid, p.156).

Na cognição como fenômeno socialmente situado não se confundem memória e atividade do sujeito. *"Não se pode confundir cultura e saber, porque a cultura não é um "celeiro" nem um "entreposto", mas antes de tudo, sede de relações produtivas entre as pessoas em ação e o mundo social (p.90-92). A sede dos gêneros, diríamos. Os objetos analisados são elementos da conjuntura histórica e cultural, devendo ser considerado como tais, nem separados do mundo social como o faz uma psicologia cognitivista e funcionalista nem confundidos com ele, como propõe a fenomenologia (p. 149). A idéia central, acentua Lave, é que a "mesma" ação realizada em situações diferentes é estruturada pelas outras atividades e as estruturas. Essa maneira de ver as coisas se opõe claramente, segundo ela, à idéia de que as atividades e as situações são isoladas e não estão vinculadas entre si, ou ainda à idéia de que certas formas de saber são utilizáveis em qualquer situação* (p.122)" (Clot, 2006, p.110).

"As áreas como antropologia, psicologia, sociologia e ciências da computação, estão participando de pesquisas a explorar o desenvolvimento humano da ação de pensamentos humanos em situações práticas (Rogoff, 1984). A pesquisa da prática do dia-a-dia focaliza pessoas em ação, pois há consenso, por parte desses autores, acerca do princípio de que é impossível analisar a prática fora do mundo socialmente material da atividade humana. Portanto, na teoria da atividade situada, a aprendizagem descontextualizada, como propõem os modelos convencionais de aprendizagem, seria uma contradição. Tradicionalmente, a aprendizagem é considerada um processo estritamente mental, o contexto em que se desenvolve a atividade é desconsiderado e, assim, reforça-se a dicotomia entre corpo e mente (já que os problemas são resolvidos "na cabeça") (Lave, 1996)" (Fonseca, 2005).

A cognição situada adota o princípio no qual o ser humano possui representações que são adquiridas com as suas experiências passadas. Mas é diante da situação real que a ação do ser humano vai ser determinada. Portanto, a determinação da ação humana decorre da utilização, pelo indivíduo, de sua experiência passada na construção de um problema.

> <<Um indivíduo representa o que ele compreendeu de suas experiências anteriores, como o saber cultural (profissional), sob diversas formas e ele

> reemprega este saber segundo a sua percepção como apropriado em função do contexto. A representação e a reutilização reforçam simultaneamente todas as duas combinações de experiências e abastecem os elementos de reorganização cognitiva e de criatividade no comportamento e na compreensão>> (Dougherty, 1985 apud Wisner, p.147).

Na realidade o comportamento humano depende de cada situação no singular, e está relacionado com o contexto. Assim, pode-se dizer que a cognição é situada e o saber, que está relacionado com o contexto, não se constitui em um acúmulo de representações, mas na reconstrução da representação dentro de uma situação singular.

4.2. A CONSTRUÇÃO DO SABER: UMA QUESTÃO SOCIAL

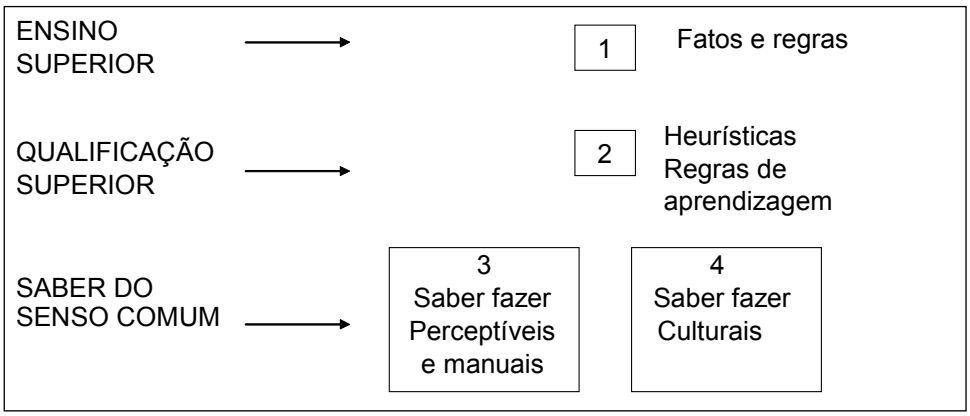
O húngaro Michael Polanyi, em sua obra denominada *Tacit Dimension*, descreve o conhecimento tácito como uma ação interna que o indivíduo não poderia controlar nem sequer perceber. Parte desse conhecimento tácito pode ser explicitada, enquanto a outra parte não pode ser exposta ou declarada, porém isso não significa que o conhecimento não possa ser comunicado ou compartilhado, mas, para o seu desenvolvimento, deve ser considerada sua natureza complexa que envolveria, além dos processos mentais conscientes, os processos subconscientes ou inconscientes e os processos corporais que participariam da nossa percepção, ou seja, as raízes corporais dos pensamentos (Frade, 2003; Fonseca, 2005).

A partir dos estudos fundamentais de Polanyi, considerados referências clássicas, há uma tendência a se identificar o saber tácito com o saber social, o tácito à socialização, ou o tácito ao não explicitável e formalizável, portanto, aquilo que escapa ao conhecimento formalizável, exprimível pela linguagem. Segundo Collins (1992), Polanyi distingue o saber tácito e o saber cultural. Um exemplo bastante conhecido de Polanyi é o saber-fazer necessário para andar de bicicleta, nesse caso ele demonstra apenas o saber tácito (como equilibrar na bicicleta), mas não descreve o saber cultural (ex: como saber andar de bicicleta nas ruas da cidade), isto é, não basta o equilíbrio, o indivíduo tem que conhecer as normas e regras da cidade por onde circula que fazem parte daquela cultura, para saber, por exemplo, que para atravessar uma rua é preciso

antes verificar se há algum veículo em trânsito no local, ou se deparar com um semáforo qual a atitude a seguir para prosseguir na travessia. Polanyi, portanto, diferencia o tácito do formal, enquanto Collins (1992) não faz esta distinção. Collins demonstra que a relação entre a parte explicita do conhecimento (1 e 2 da figura 1) e as habilidades dificilmente explicitadas do conhecimento (3 e 4 da figura 1) são dinâmicas. No entanto, os deslocamentos (que podem ocorrer de baixo para cima ou de cima para baixo) só podem ser compreendidos dentro do contexto em que o conhecimento é utilizado.

As heurísticas e regras de aprendizagem são formalizadas por convenção de uma comunidade e parte dessas regras é interiorizada pelos indivíduos. Portanto, participar de uma mesma cultura é condição indispensável para se ter um entendimento e saber usar as regras e heurísticas.

Figura 1: A hierarquia convencional do saber



Fonte: Collins (1992, p.153)

São quatro os tipos de saberes descritos por Collins (1992, p.151-152), na hierarquia convencional do saber: fatos e regras; heurísticas; saber-fazer perceptível e saber-fazer cultural. Os fatos e regras formais incluem fatos que são fáceis de explicar, tais como "*a água ferve a 100 graus, ou nos jogos de xadrez, quando o rei tornou-se a única peça que se possa mexer, se encontra na impossibilidade de ser deslocada sem se colocar em risco, se diz que a partida termina por xeque-mate*" (Collins, 1992, p.151). As heurísticas são regras empíricas explicitáveis e práticas padronizadas. São encontradas, por exemplo, nos manuais de treinamento esportivo, nas práticas científicas e tecnológicas, como a produção de cristais: *"começar sempre o*

resfriamento da mistura fundida bem acima do ponto de fusão indicado" (Collins, 1992, p.151). Estas regras, ao serem interiorizadas, tornam-se subconscientes para o sujeito. Quanto ao saber-fazer, este seria a dimensão tácita do conhecimento, isto é, o indivíduo sabe fazer, mas não sabe explicar como ele faz.

A hierarquia convencional do saber, representado na figura 1, retrata a noção do ensino no ocidente. Pode-se ver também nessa figura como as categorias mais formalizadas e mais explicitáveis estão hierarquicamente acima do saber-fazer.

Um dos movimentos das ciências é a evolução de baixo para cima do diagrama. Isto seria buscar explicitar ao máximo o saber-fazer e transformá-lo em regras. As ciências desconsideram a base cultural do saber-fazer e tentam transformá-lo em regras universais.

Assim, conforme mencionado anteriormente, mesmo o conhecimento formal tem uma parte explícita e outra não. No entanto, para se desenvolver um software, cuja construção é de um sistema formal, o desenvolvedor deverá deter o entendimento sobre as regras dos usuários, que acontece através da aculturação entre os agentes envolvidos, possibilitando, posteriormente, a criação de uma regra específica para o software.

O resultado da atividade do desenvolvedor são escrituras formais ou lógicas, mas estas escrituras não são meramente lógicas ou formais, pois o desenvolvedor também cria significação a partir de uma construção coletiva. Quando as pessoas compartilham o mesmo universo de signos, torna-se possível o entendimento dos seus significados e a negociação para a construção de novos signos. Assim, a codificação do sistema não é apenas uma escritura lógica, os códigos também têm seus significados, que lhe são atribuídos pelo indivíduo.

O depoimento de um analista de sistemas da SERVPUB demonstra a necessidade de um contato social com os usuários para a compreensão das regras, uma palavra na própria empresa possuía significados diferentes que não foram explicitados na demanda:

> "a palavra "diversos" também tinha significados diferentes para os setores da empresa, eu apanhei até entender isso... o setor de contabilidade chamava de "diversos" todos que não são "produtos"... enquanto o pessoal do faturamento separa alguns itens ((multa, atualização monetária)) que para

> eles não são produtos e nem "diversos"... apareceram alguns erros até eu descobrir que existiam essas diferenças... eles me falavam que o item "diversos" estava com problema... eu não entendia... até que um dia, quando sentei com eles ((representantes de cada setor)), eu percebi que diversos... para a contabilidade... era diferentemente do faturamento... Juntei os dois setores para a gente entrar num acordo" (Analista da SERVPUB).

O analista, especialista no setor de faturamento, não mantinha contato com os usuários de outros setores. Portanto, ele não possuía o conhecimento nem o entendimento de que a palavra "diversos" tinha significados diferentes para os trabalhadores do outro setor. Segundo Collins (1992), para a compreensão e utilização dos fatos, das regras e das heurísticas é necessário compartilhar uma mesma cultura. Disto decorre a capacidade de ler e compreender as informações. Um exemplo, citado pelo mencionado autor, é uma receita de cozinha que contém as seguintes instruções: "Bater os ovos em neve". Para seguir a receita é necessário entender cada palavra, é preciso saber que o branco não é branco e sim transparente antes de bater os ovos. Além disso, temos que saber quebrar o ovo, saber separar a clara da gema, saber bater os ovos e saber a hora que está pronto. Resumindo, para bater os ovos em neve é necessário que se tenha vivido uma experiência anterior e, assim, interiorizado esses saberes.

Uma vez adquirida a compreensão pelos analistas e trabalhadores sobre os significados que a palavra "diversos" tinha para o outro setor, foi realizada uma nova conversão entre os trabalhadores dos setores e analistas, na qual a palavra "diversos" passou a ter apenas um significado, qual seja, "os que não são produtos". Essa regra se constituiu em uma construção social entre os membros envolvidos que, com o tempo, será interiorizada pelos indivíduos e dificilmente será lembrada ou exteriorizada.

"O saber e o fazer não são separáveis. Eu sei como falar falando com outras pessoas (...). Todas as mudanças das regras da fala é um acordo social; é um acordo das práticas comuns. Se as regras da fala mudam, eu faço como os outros, não porque as pessoas me dizem o que eu devo fazer, mas vivendo com os outros - partilhando suas <<formas de vida>> (Wittgenstein, 1953) -, eu mudo com eles. Eu mudarei o que eu sei sobre a maneira pela qual é preciso falar, não por causa de uma escolha, não em função da aceitação de uma regra avaliada conscientemente, não no nível de um processo consciente, mas porque, fazendo com os outros, eu descobrirei que eu sei o que eles sabem. Sabendo o que eles sabem, eu farei o que eles fazem. Isso acontece no falar, no

escrever, no fazer um trabalho industrial ou uma escultura, no praticar a medicina e no descobrir partículas elementares" (Collins, 1992, p.19, tradução da autora).

> "Esses savoir-faire culturais comportam a capacidade de induzir as mesmas conclusões que os outros, no âmbito da ação combinada. São nossos savoir-faire culturais que nos permitem fabricar um mundo de comportamentos combinados. É assim que concordamos, por exemplo, com o fato de certo objeto ser um Rembrandt ou tal símbolo ser a letra s. É deste modo que digitalizamos o mundo. É nossa cultura comum que torna tais consensos possíveis e é nossa atitude de fabricar esses consensos que funda nossa cultura" (Collins, 1992, p.152, tradução Haddad).

4.3. OS MODELOS FORMAIS E O SABER CULTURAL

O modelo formal, a separação funcional e a distância física entre os técnicos em informática e usuários não permitem a criação de uma cultura entre os envolvidos no DS. A falta de interação entre os profissionais é a queixa dos técnicos em informática e usuários de uma empresa prestadora de serviço público. Este modelo organizacional tem se constituído em um obstáculo para os técnicos em informática compartilharem os seus saberes. Tanto os empregados da SERVPUB quanto os empregados da FABRISOFT estão apresentando dificuldades em realizar as atividades dentro do novo modelo organizacional de "Fábrica de Software". A separação dos membros da equipe em salas distintas e a distância geográfica da FABRISOFT, situada fora do estabelecimento físico da SERVPUB, são as principais queixas dos trabalhadores. As dificuldades se revelam na insistência da analista de sistema da FABRISOFT em permanecer trabalhando no setor de desenvolvimento de software da SERVPUB, mesmo após o término do tempo de aprendizado previsto pelas empresas (SERVPUB e a FABRISOFT), que já se estende por mais de um ano após sucessivos adiamentos.

Como visto, o trabalho do analista envolve várias pessoas (usuários leigos e profissionais de informática), implicando interações de diversas naturezas e em diversos momentos. O saber desse profissional faz parte de uma cultura e, para o aprendizado, é preciso que as pessoas compartilhem o mesmo ambiente, tornando a interação presencial entre os envolvidos imprescindível no momento de desenvolver soluções ou resolver problemas.

Quadro 2: Depoimento dos analistas de sistemas

"j*á imaginou quando eu sair daqui? Eu preciso das pessoas o tempo todo para discutir... eu não conheço nada daqui. Eu sou direcionada por eles ((empregados do setor de desenvolvimento))... são eles que sabem a regra do negócio... eu não consigo estar fora daqui... o conhecimento está aqui... o ambiente está aqui... o apoio está aqui... só o desenvolvimento está lá*" (Analista de sistema da FABRISOFT).
"o pessoal da equipe não deixa eu ir para lá... dizem que eu produzo mais aqui... aqui já está difícil eu achar as pessoas... já imaginou lá na fábrica... eu estou precisando de falar com fulano mas não consigo" (Analista de sistema da FABRISOFT).
"estão passando para ela ((analista terceirizada)) uma demanda de faturamento... mas ela não sabe nada... se ela ficar aqui a gente pode discutir e eu vou passando aos poucos as informações... imagina ela longe anotando em um papel... como foi falado na palestra... e tendo que vir aqui trazer o papelzinho... quando ela voltar já vai estar cheia de dúvidas de novo... olha... eu já conheço há muito tempo o faturamento e ainda tenho dúvidas" (Analista da SERVPUB).
"d*izem que não precisa da gente passar as informações porque está tudo registrado... mas o sistema é muito grande... isto não é possível... mesmo com toda a documentação... a pessoa tem dúvida e vem aqui" (*Analista da SERVPUB).

No depoimento acima "(*...) eu não consigo estar fora daqui... o conhecimento está aqui... o ambiente está aqui... o apoio está aqui..."* ressalta-se a importância da interação direta entre os técnicos de informática e usuários, dentro do ambiente de trabalho destes últimos, para a construção do conhecimento. Observou-se que os analistas da FABRISOFT, quando estavam distantes da SERVPUB, apresentaram diversas dificuldades na obtenção do entendimento das necessidades do usuário. Mesmo com toda a documentação sobre a demanda do usuário os analistas tinham dúvidas de interpretação: "(*...) mesmo com toda a documentação... a pessoa tem dúvida e vem aqui* (SERVPUB)*"*. Muitas são as informações necessárias para se desenvolver um software,

verifica-se que o tempo consumido pelo desenvolvedor, na obtenção de um entendimento dessas informações, é maior do que com o gasto com a comunicação presencial: " (...) *aqui já está difícil eu achar as pessoas...já imaginou lá na fábrica...eu estou precisando de falar com fulano mas não consigo"*. O risco da falta de entendimento das necessidades do usuário aumenta o retrabalho e, conseqüentemente, o tempo de desenvolvimento e o custo da atividade.

A interação presencial, como descrita acima, permite que as pessoas, diante das tomadas de decisão, criem novas regras. A atividade de DS demanda constantes tomadas de decisão, pois a atividade é processo de construção de uma teoria que não está pronta na mente do seu usuário, é um processo que ainda será elaborado. Portanto, a sua construção exige um processo coletivo, no qual todas as pessoas envolvidas devem participar.

Estar dentro de uma empresa, para a qual vai se desenvolver um software, é estar dentro de um contexto histórico e social, no qual é imprescindível, para um entendimento comum dos significados e para o favorecimento do fluxo rápido de informações, o conhecimento sobre o que os envolvidos sabem e a compreensão dos jargões utilizados na empresa.

Em torno de um software existem duas comunidades (a comunidade interna entre os técnicos e a comunidade que utiliza o software) com atores bem individualizados, com suas regras, hierarquias, conflitos e jogos de interesses. O entendimento dessas (regras, hierarquia...), segundo Wenger (apud Agydelo e Souza, 2003), "*situa-se, a partir do ponto de vista dialógico, numa perspectiva relacional, dinâmica e social ocorrendo, com mais ou menos ênfase, em qualquer comunidade que se configure como uma comunidade de prática*".

"O conceito de comunidades de prática desenvolvido por Wenger surgiu da observação das relações naturalmente feitas pelas pessoas no convívio social diário (por exemplo: na família, na participação em um time de futebol ou nos ambientes de trabalho). A *comunidade de prática*, diferente de uma comunidade comum, *reúne pessoas em torno de uma determinada prática, que a realizam num mútuo comprometimento.* E segundo Wenger, *as pessoas tendem a engajar-se com maior facilidade em ações, cujos significados negociaram entre si*. Isso quer dizer que não

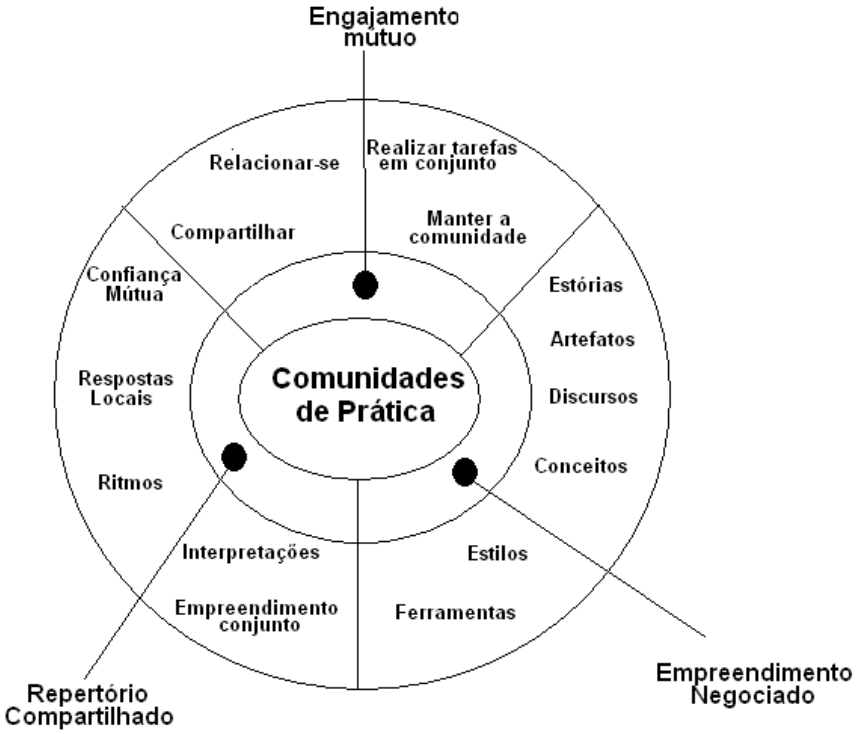
existe um lugar ou lugares específicos do saber, mas que ele acontece ininterruptamente nas *comunidades de prática* e é essencialmente interacionista" (Frade, 2003).

A associação entre prática e comunidade é dada através de três relações (ver na figura 2), são elas:

1. Engajamento/envolvimento mútuo (organizado em torno dos objetivos que a comunidade busca alcançar);
2. Um empreendimento conjunto (e, consequentemente, negociação mútua e responsabilidade, da qual se devem prestar contas);
3. Um repertório compartilhado (rotinas, linguagens, símbolos, modos de fazer as coisas, dentre outros) (Wenger, apud Frade, 2003, p.67).

Wenger reconhece que existem fatores externos que escapam ao controle dos membros de uma comunidade e que podem influenciar e limitar o funcionamento da prática. Porém, diz ele, a realidade do dia-a-dia da prática é, em última instância, produzida por seus membros (ibid).

Figura 2: As três relações que caracterizam uma comunidade de prática



Fonte: Adaptado de Agydelo e Souza (2003)

O próximo capítulo descreve a análise ergonômica realizada com os analistas de um setor de desenvolvimento de software de uma empresa prestadora de serviços públicos, evidenciando as dificuldades decorrentes da opção da empresa pela adoção do modelo "fábrica de software" e pela terceirização de serviços, demonstrado com o caso do "reajuste tarifário", bem como uma discussão teórica acerca da viabilidade e conseqüências da adoção desse modelo.

Capítulo 5 - O estudo no setor de desenvolvimento de software de uma empresa que presta serviços públicos

Este estudo teve origem quando do recebimento de uma demanda destinada ao grupo de trabalho em ergonomia da empresa, do qual a autora fazia parte. O gerente do setor de desenvolvimento de software solicitava um "parecer ergonômico para verificar e qualificar as condições a que estavam sujeitos os funcionários do setor em suas atividades durante o processo produtivo de software". O objetivo da demanda era adequar o mobiliário, otimizar o desenvolvimento operacional, melhorar o nível de conforto do ambiente de trabalho, reduzir a ocorrência de doenças oriundas dos postos de trabalho, reduzir o estresse, reduzir o absenteísmo e atender a legislação trabalhista (NR17). A avaliação ergonômica teve início em dezembro de 2004.

5.1 METODOLOGIA

A metodologia empregada nesta pesquisa é a da Análise da Atividade, que está fundamentada no ponto de vista da atividade real de trabalho, isto é, consiste em evidenciar, por meio de observações, como o trabalhador executa a sua atividade, qual o seu comportamento, as suas condutas, processos cognitivos e interação (Daniellou, 2004).

A compreensão da atividade real dos desenvolvedores de software ocorreu durante a pesquisa de campo, na análise dos dados de observação e nas entrevistas em autoconfrontação que duraram cerca de três anos. Dessa forma foi possível penetrar

realmente na atividade de trabalho dos desenvolvedores, conhecer seu fazer e assim compreender seu trabalho, suas dificuldades, suas variabilidades e dominar seu vocabulário.

Foram observadas situações cotidianas no próprio local de trabalho e realizadas entrevistas em autoconfrontação, o que permitiu avaliar as reais condições em determinada situação de trabalho. *"Esta atenção voltada para o trabalho real, no seu nível mais apurado, encontra sua origem na busca dos fatos, uma atitude comportamentalista*" (Daniellou, 2001, p.29). Os ergonomistas partem da premissa fundamental de que uma parte do comportamento é subconsciente para aquele que o realiza e, por isso, ele é incapaz de relatá-lo espontaneamente, pois ele próprio desconhece o saber ali construído (Vermersch, 1994; Lima, 2001). Esta seria a dimensão tácita do conhecimento, isto é, o indivíduo sabe fazer, mas não sabe explicar como ele faz. As próprias regras, ao serem interiorizadas, tornam-se subconscientes para o sujeito.

A pesquisa teve início em dezembro de 2004 com uma análise ergonômica do trabalho (Leal e Pereira, 2004), com o objetivo de atender a solicitação da gerência de um "parecer ergonômico para verificar e qualificar as condições a que estavam sujeitos os funcionários daquele setor em suas atividades durante o processo produtivo de software".

Segundo Guérin et al. (2001, p.1), compreender o trabalho para transformá-lo é a finalidade primeira da ação ergonômica. Conhecer a atividade do desenvolvedor exige conhecer seu fazer, vê-lo trabalhar, observá-lo em um ciclo completo de um projeto, desde as fases iniciais de levantamento de campo e construção da especificação funcional, até a fase de desenvolvimento propriamente dita, as fases de testes e a implantação do sistema.

Para realização do estudo foi firmada uma negociação formal com o gerente do setor da empresa, viabilizando os trabalhos da pesquisa, que se ateve inicialmente em responder à demanda da empresa, que se constituía em avaliar o leiaute, o mobiliário, a incidência de doenças osteo-musculares e o nível de estresse dos técnicos em informática da SERVPUB.

A avaliação ergonômica do trabalho realizada para a empresa ocorreu no ano de 2005. Em janeiro e fevereiro daquele ano, foram realizadas observações gerais dos trabalhadores (analistas e programadores). Diante dos resultados destas observações surgiu a hipótese de que as atividades dos analistas e programadores se confundem. Um questionário com perguntas variadas sobre posto de trabalho, saúde do trabalhador, leiaute e organizacional foi realizado.

Concomitante à avaliação do questionário, iniciamos observações específicas, entrevistas individuais, em grupo, semi-dirigidas e livres. Após as observações, foram realizadas autoconfrontações dos fatos observados, com a finalidade de compreender as situações de trabalho, as representações que têm do trabalho, os sentimentos a respeito do trabalho. Os resultados qualitativos e quantitativos possibilitaram identificar os impactos da atividade sobre a saúde dos técnicos em informática, o melhor leiaute para o setor e confirmar a primeira hipótese de que as atividades dos analistas e programadores se confundem. Os resultados desta avaliação serão descritos adiante.

Em 2006, após a entrega dos resultados da avaliação ergonômica para a empresa, o estudo foi reiniciado com o objetivo de aprofundar a análise da segunda hipótese, qual seja, que os problemas existentes no DS são, em certa medida, inerentes à natureza dessa atividade, que mais se assemelha a uma "criação artística" que ao produto massificado da era industrial e do papel do social, isto é, da importância da interação entre os agentes envolvidos (analistas e usuários) na construção do saber dos desenvolvedores de software, primordial para a criação do software.

Diante do grande número de analistas e programadores e equipes distintas no setor, foi escolhida uma equipe do sistema comercial para a realização deste estudo.

Foi realizada uma análise documental na proposta do novo modelo organizacional do setor, na qual se descrevia as novas funções e papéis dos técnicos em informática e análise.

Nas observações sistemáticas da atividade do desenvolvedor de software, foram realizadas anotações manuais e gravações em fitas K-7 dos diálogos entre as diversas pessoas envolvidas (analistas de negócio, analistas de sistemas, desenvolvedores, clientes e usuários). Os diálogos aconteceram durante o desenvolvimento do software,

na empresa SERVPUB. Após a coleta das observações, foram realizadas entrevistas em autoconfrontação com as pessoas envolvidas nos diálogos.

A partir das observações diretas do programador, analista e usuário, foram identificados e analisados a natureza da atividade e o caráter social do desenvolvimento de software. Este último se manifesta, sobretudo, pela via da comunicação entre os participantes. Para o entendimento da demanda do cliente é preciso que as pessoas compartilhem com o mesmo grupo social, tornando a interação presencial entre os envolvidos imprescindível no momento de traduzir as necessidades do usuário, desenvolver soluções ou resolver problemas.

Com o entendimento da natureza da atividade dos técnicos em informática e do caráter social do DS surge uma outra hipótese com o trabalho dos técnicos da FABRISOFT: só é possível adquirir o saber, necessário para a construção de um software eficiente, por meio da aculturação entre os membros envolvidos. Assim, novas observações diretas foram realizadas através de entrevistas de autoconfrontação junto aos técnicos da FABRISOFT e usuários, na tentativa de entender como se dá a interação entre os envolvidos e se essa é suficiente para a construção dos saberes.

O resultado obtido, dessa última hipótese, é que os desenvolvedores, distantes da SERVPUB, comunicando com os usuários por meio de reuniões, por documentos e telefones, não conseguiram construir o saber necessário para se desenvolver um software com qualidade, confirmando a hipótese de que só é possível adquirir o saber, necessário para a construção de um software eficiente, pela aculturação entre os membros envolvidos.

As entrevistas de autoconfrontação consistem basicamente em obter comentários do sujeito sobre seu próprio comportamento em diversos níveis, obedecendo a sequência: Como? (se faz, se sabe, escolhe, etc); Para que? (finalidade, objetivos) e Por que? (motivos, razão). Nesta seqüência estão implícitos dois níveis de autoconfrontação, o primeiro se concentrando na explicitação dos procedimentos concretos, modos operatórios, atos observáveis, informações utilizadas na execução do trabalho, elementos que influenciam as decisões, etc; e o segundo explicitando os significados latentes de comportamento observável (Telles, 1995).

O confronto do sujeito com seu próprio comportamento dar-se-á a partir de dados de observação cuidadosamente coletados. Os dados observados não são aleatórios, devem consistir em um conjunto de comportamentos e elementos do contexto, pertinentes ao que se pretende explicitar.

Paralelamente ao trabalho de campo, realizou-se pesquisa bibliográfica em busca de trabalhos sobre o tema, tais como: desenvolvimento de software, engenharia de software, fábrica de software, atividade artesanal, atividade industrial, organização do trabalho, saber ou conhecimento, gestão do conhecimento, cognição humana, competência, processo de tradução, artefatos e instrumentos de trabalho.

A análise dos resultados apresentados aqui, obtidos por meio de observações e entrevistas de autoconfrontação, será descrita no decorrer desta dissertação.

5.2 REORGANIZAÇÃO DO SDS

5.2.1. A empresa e o SDS

A empresa comercializa quatro tipos de serviços, sendo eles, “produto A bruto”, “produto A tratado”, “produto E” e “produto E industrial”. Tais serviços são fornecidos diretamente às comunidades que fazem parte dos municípios que têm contrato firmado com a SERVPUB. A empresa presta serviços para a maioria dos municípios do Estado.

A busca atual da SERVPUB é pela excelência na sua área de atuação. Portanto, a consolidação da imagem de excelência da empresa se dará por meio da implantação da grande meta descrita em seu planejamento estratégico, qual seja, a promoção de um “choque de gestão”, seguindo o objetivo prioritário traçado pelo Governo do Estado. Um dos pontos chave dessa meta é a redução da relação despesa/receita do Governo Estadual.

Traçada a meta, a empresa optou pela adoção de políticas de redução de pessoal, afetando alguns setores da empresa como o SDS, que teve a produção comprometida

por não conseguir atender a sua demanda. Atualmente, conforme sugestão da gerência da SDS, a empresa pública contratou uma fábrica de software terceirizada para atender, em caráter parcial ou total, algumas de suas demandas.

Existem evidências de que a intenção da empresa, no futuro, é a completa terceirização do referido setor de desenvolvimento. Seguem alguns depoimentos:

Quadro 3: Depoimentos dos analistas da SERVPUB

"as pessoas que saíram não foram substituídas... estamos sobrecarregados... estão passando tudo para a fábrica ((terceirizada))" (Analista da SERVPUB).
"d*izem que nós vamos ficar com análise de projeto... mas eu não sei não*" (Analista da SERVPUB).
"a *gente acha que a tendência é acabar com a DSD... estão criando uma divisão a DVIE ((Divisão de Informações Estratégicas)) para fazer atividades para a diretoria... que eram feitas na DSD... o SAP pegou uma parte e a DVIE a outra*" (Analista da SERVPUB).
"*aqui a gente não consegue mais promoção... na DVIE as pessoas que passarem no concurso interno vão conseguir promoção. As pessoas que estão saindo do setor não serão repostas. A divisão abriu vaga para dois analistas e três programadores. Pelo que eu sei aproximadamente 11 programadores e 7 analistas estão concorrendo para a vaga. É um número muito grande de pessoas que estão querendo sair, eu não sei o motivo, mas a impressão é a sensação que a área está acabando*" (Analista da SERVPUB).

5.2.2. Condições organizacionais do SDS antes da fábrica de software

Em 2004, início da análise ergonômica realizada para a empresa, o setor contava com a colaboração de 70 trabalhadores entre analistas de sistemas e programadores. Os trabalhadores se dispunham em equipes heterogêneas lideradas por analistas de sistemas ou por programadores.

Os técnicos em informática, analistas e programadores, desenvolviam sistemas para a própria empresa, realizavam manutenção corretiva, evolutiva e adaptativa,

inclusive a adaptação de pacotes (programas prontos adquiridos de terceiros). Na época a empresa contava com 160 programas ativos, tais como: sistema comercial – Sistema de Informações Comerciais; Sistema de Informações de Apoio à Gestão Operacional; Sistema de Recursos Humanos; Sistema de Administração de Materiais; Sistema Integrado de Controle e Análise de Custos. Todos sob a responsabilidade dos técnicos, encarregados das manutenções.

As demandas que chegavam ao setor eram repassadas aos líderes das equipes, que por sua vez as repassavam a um analista e/ou programador de sua equipe, ou ainda a uma outra equipe. As escolhas dos profissionais para atender as demandas eram feitas de acordo com a disponibilidade, experiência e conhecimento dos analistas e programadores. Algumas demandas eram atendidas apenas pelos especialistas do sistema.

Apesar da existência de classificação para as funções do analista (júnior, pleno, sênior...) e de programador (júnior, profissional, pleno...), na prática a classificação não era observada. Tratava-se apenas de mera formalidade, pois o que a realidade evidenciava era uma mistura de papéis, em que os analistas realizavam as atividades dos programadores e vice-versa.

Formalmente, para se desenvolver um sistema é necessário que o programador obtenha dos analistas uma definição precisa. No entanto, o que acontecia na prática não era exatamente o que se esperava, as definições eram realizadas por meio de pedaços de papéis ("jaquetes", no jargão do ofício) ou verbalmente, sem que as informações fossem completas.

> “o analista me passou uma definição... mas na realidade não é uma definição correta... ele me passou apenas campos... vou ter que procurá-lo para me passar o restante das informações” (programador).

Não se tratava de negligência ou má vontade dos analistas. Durante as observações, os analistas verbalizaram que, devido ao tempo consumido para formalizar o entendimento do processo, a confecção de uma definição completa não compensaria. Outros verbalizaram que preferiam fazer a programação a repassar todas as informações para o programador. Além disso, houve queixas acerca do risco de incompreensão na transmissão de informações, o que poderia levar a um erro de programação.

"o tempo que o analista gasta para fazer a definição e o risco que a gente corre de não ser bem entendido... é melhor fazer a programação de uma vez" (analista).

A pressão temporal e as informações "incompletas" do usuário contribuíam para a definição ser deficiente, conforme se aduz do depoimento abaixo:

> "os usuários nunca informam tudo... quando estou desenvolvendo o sistema tenho que correr atrás deles para fazer as alterações... a pressão também atrapalha, pois não é possível passar tudo para o programador" (analista).

Fator importante na produção de informações incompletas ou insuficientes está relacionado à diferença de linguagem dos interlocutores. A diferença de linguagem é responsável por uma parte considerável de informações inadequadas obtidas pelos programadores (Ver Ferreira, 2004).

Pesquisa realizada com um questionário aplicado a 73% do pessoal no setor (50 técnicos em informática) confirmou a existência da interdependência das fases de concepção e execução de desenvolvimento de um sistema, pois 80% dos trabalhadores pesquisados informaram que as funções se superpõem ou se confundem. O resultado apresentado para a empresa demonstrou que existia interdependência e superposições entre as atividades que não poderiam ser eliminadas por uma rigorosa divisão do trabalho. Ao contrário da crença dos defensores do modelo da FS, não é a "confusão" entre tarefas que gera problemas de qualidade ou ineficiência, mas os problemas é que podem ser agravados se a divisão do trabalho for efetivamente imposta ao setor.

> "Nas diversas etapas do processo de geração do software a diferenciação de funções é dificultada pela interdependência das tarefas, o que exige uma participação simultânea, muitas vezes indiferenciada do ponto de vista de contribuição profissional, do analista e do programador. A interdependência das fases de concepção e execução ((desenvolvimento)) de um sistema requer um ambiente pouco hierarquizado. Percebe-se que existem contradições entre a hierarquização e a sobreposição das atividades dos técnicos de desenvolvimento de sistemas ((analistas e programadores)) que vêm contribuindo para os conflitos entre estes profissionais" (Tavares, 1983).

Quando os profissionais se propõem a realizar apenas a tarefa prescrita (operação padrão), os conflitos aparecem e repercutem em todo o trabalho dos informáticos, da qualidade do serviço, passando pelo descumprimento dos prazos e pela insatisfação do cliente, aos efeitos sobre a saúde dos trabalhadores envolvidos. Eis um depoimento significativo:

> "(...) comuniquei à minha gerência que, a partir daquele dia, eu só ia fazer o serviço de programador. Quando eu tomei esta decisão o analista ficou olhando torto para mim, expliquei que o problema não era com ele, o problema era meu. Mas eu não consegui fazer isso por muito tempo, eu resisti apenas dois meses, então eu não ia trabalhar. Se eu ficar aqui só esperando a definição não ia trabalhar, se eu não corro atrás e não busco as informações, não trabalho ou trabalho pior, trabalho mal. Quem faz isso aqui é discriminado. O programador que só faz o serviço de programação é discriminado" (programador).

5.2.3. Condição organizacional atual do SDS

Atualmente o setor conta com 40 técnicos em informática. Em aproximadamente quatro anos houve redução de 43% do pessoal. Isso se deu em conseqüência da política de remanejamento de pessoal da empresa, que permitiu a saída dos empregados do setor (ex: aposentadoria, mudanças de área por concurso interno ou transferências), mas não os substituiu. Conforme mencionado, a empresa preferiu a contratação de uma fábrica de software terceirizada à contratação de pessoal.

Há aproximadamente dois anos, com a redução dos técnicos do sistema comercial, a empresa já havia iniciado a contratação de profissionais terceirizados para reposição do quadro de pessoal. Transcorrido esse período, deu-se início às mudanças organizacionais com a SERVPUB firmando contrato com uma fábrica de software aqui denominada FABRISOFT.

Os analistas da SERVPUB sugeriram que a FABRISOFT contratasse os profissionais terceirizados que prestavam serviços a SERVPUB, objetivando o aproveitamento da experiência desses profissionais na empresa. Tal solicitação foi aceita e os profissionais foram contratados pela terceirizada.

Durante o desenvolvimento do sistema havia aproximadamente 18 técnicos (analistas e programadores). Após a implantação do sistema, o número de técnicos só vem reduzindo. No ato da contratação da fábrica apenas 9 técnicos estavam trabalhando na equipe do sistema comercial. Com a implantação da fábrica o quadro de pessoal chegou a ser reduzido para apenas 1 técnico.

Assim, os técnicos do sistema comercial ocupavam, inicialmente, duas salas subdivididas. Com a redução do quadro de pessoal, passaram a ocupar apenas uma sala e, após a implantação da fábrica, foram separados em salas distintas, incluindo a analista terceirizada. Nesse momento pode-se perceber a tentativa de divisão dos papéis e funções, bem como o controle hierárquico dos processos formalizados.

Logo após o contrato, várias demandas foram enviadas a FABRISOFT, mas logo o primeiro projeto, demandado e realizado pela FABRISOFT, apresentou diversos problemas, que serão abordados adiante. Com isso, os técnicos de informática reuniram-se com a gerência e solicitaram o remanejamento dos técnicos da FABRISOFT para a empresa. A partir de então se estabeleceu que os técnicos da FABRISOFT trabalhariam nas dependências da SERVPUB, próximo aos analistas da equipe comercial, que haviam sido remanejados para exercerem outros papéis e retornaram conforme determinação da gerência do SDS.

Os analistas e desenvolvedores de ambas as empresas (SERVPUB e FABRISOFT) foram alocados em uma mesma sala. A orientação dos dirigentes era que os analistas e desenvolvedores da FABRISOFT deveriam ser mantidos na empresa durante seis meses, tempo considerado por eles suficiente para se adquirir todo o conhecimento com os analistas de negócio.

5.2.4. A implantação do modelo “fábrica de software”

O contrato da SERVPUB com esta fábrica é amplo no que se refere ao seu objeto, podendo servir para atender total ou parcialmente a projetos, ou podendo ainda ser utilizado para prestar consultoria ou apenas realizar a documentação. O contrato realizado com a referida fábrica de software exige que a metodologia seja da SERVPUB. A metodologia prescrita e implantada pela SERVPUB, como já mencionado, foi baseada no RUP.

Os técnicos em informática da SERVPUB foram agrupados em estruturas funcionais, sendo que cada um deles desempenhava um ou mais papéis e era

responsável por uma parcela de trabalho ao longo do ciclo de desenvolvimento do serviço, conforme a demanda do projeto. A estrutura do setor está representada no quadro 4:

Quadro 4: Estrutura funcional do SDS

Setor	Unidades	Funções/Papéis
Gerência		Gerente
		Secretária
PMO – Project Manager Office	PMO	Gerente PMO
		Assessor PMO
Setor de desenvimento.	Gerente Setor de Desenvolvimento	Gerente Projeto – Desenvolvimento
	Unidade de Requisitos	Analista Negócio/Analista Sistemas
	Unidade de Protótipo	Prototipador
	Unidade de Projeto (desenho)	Projetista
	Unidade de Implementação	Desenvolvedores
Setor de Manutenção	Gerente Setor de Manutenção	Gerente Projeto – Manutenção
	Unidade de Requisitos	Analista Negócio / Analista Sistemas
	Unidade de Projeto (desenho)	Projetistas
	Unidade de Implementação	Desenvolvedores
Setor de Apoio a Sistemas	Unidade de Componentes	Administração de Componentes
	Unidade de Dados	Administração e Dados
	Unidade de Suporte ao Desenvolvimento	Consultor Desenvolvimento
Setor de Controle da qualidade	Setor de Controle da Qualidade	Coordenador da Qualidade
	Unidade de Administração Qualidade	Analista Processos
	Unidade de Administração. Qualidade Produto	Analista Controle Qualidade
		Coordenação Célula de Teste
		Testador
	Unid. Adm. Métrica/Indicador	

Fonte: adaptada dos documentos internos da SERVPUB

O papel do gerente do escritório de projetos é acompanhar os indicadores de performance dos projetos. É também de sua responsabilidade: apoiar o processo de planejamento e a preparação do orçamento dos projetos; auditar os projetos; manter e armazenar as lições aprendidas, bem como as melhores práticas desenvolvidas; além de apoiar e avaliar os gerentes de projeto.

O gerente de projeto atua de forma a atingir os objetivos propostos dentro de determinados parâmetros de qualidade, obedecendo a um planejamento prévio de prazos (cronograma) e custos (orçamento). Dadas as metas e as restrições de recursos e tempo, cabe ao gerente de projetos garantir que ele atinja os objetivos propostos.

A gerência de requisitos envolve todas as atividades responsáveis por identificar, registrar e catalogar e controlar uma versão para acompanhar as mudanças e assegurar que elas sejam implementadas no produto final.

A Administração de Dados (AD) é de natureza estratégica e de planejamento, sendo que a Administração de Banco de Dados (ABD) se constitui no nível mais técnico, ligado aos meios físicos e à tecnologia disponível. Estas podem atuar em conjunto.

O Arquiteto de software é o responsável por definir as combinações de técnicas e tecnologias mais indicadas para cada projeto, orientando o desenvolvimento de sistemas e buscando garantir a melhor integração entre estas combinações e a reutilização das estruturas já existentes ou iminentes.

O Analista de Ponto de Função realiza a métrica por Ponto de Função, isto é, a medida para expressar a quantidade de funcionalidade que um sistema de informação provê a um cliente, independentemente da tecnologia a ser utilizada no seu desenvolvimento, na perspectiva do mesmo.

O Controle de qualidade é a garantia da conformidade do produto gerado em relação aos processos da metodologia.

O Analista de negócio é o que detém o conhecimento do “negócio” da área do solicitante, ele é o responsável pela interface entre o cliente e/ou usuário e o analista de sistemas.

Como descrito anteriormente, as atividades de DS entre os analistas e programadores (desenvolvedores) se sobrepõem. No entanto, a nova divisão de papéis, aliada à terceirização de uma fábrica onde os desenvolvedores distam geograficamente da SERVPUB, apenas agravou as dificuldades já encontradas no modelo anterior. A separação física entre os desenvolvedores e o analista de negócio, bem como o isolamento destes do contexto (da empresa) não permitiram que os desenvolvedores adquirissem o saber necessário para desenvolver um produto que atendesse as necessidades reais dos usuários. Um exemplo de dificuldade ocorreu quando uma demanda foi enviada a FABRISOFT para desenvolver um relatório das atividades dos leituristas da SERVPUP por meio do pocket (aparelho que realiza leitura *on line* e *off line* do consumo do cliente). Após o início dos trabalhos, a empresa terceirizada repassou a SERVPUB um outro preço para dar continuidade ao serviço contratado de desenvolvimento do software. Como o preço foi considerado muito alto pela SERVPUB, esta resolveu repassar a tarefa para o seu setor de desenvolvimento.

A FABRISOFT, então, repassou a SERVPUB todos os documentos pertinentes às etapas do trabalho que havia desenvolvido. Durante a avaliação do programa e das informações descritas nos documentos, os analistas da SERVPUB, responsáveis pela demanda, apontaram várias críticas sobre ele, conforme os depoimentos abaixo:

Quadro 5: Depoimentos dos analistas da SERVPUB

"estou vendo que a fábrica terceirizada fez uma coisa a mais... eles criaram um processo só para o registro de ociosidade... eles estão dando uma volta a mais que não precisava... como eu sempre digo, tem vários caminhos para chegar ao centro, mas você pode ter que dar mais volta para chegar" (Analista da SERVPUB).
"estou tendo que fazer o serviço de novo... assim que eles começassem a rodar o relatório iriam aparecer os problemas... tive que mudar o leiaute... eu conheço mais a empresa... sei dos detalhes... já fiz outros relatórios" (Analista da SERVPUB).
"estou retirando este processo de ociosidade a mais que eles redefiniram para não precisar fazer uma comunicação a mais entre o POCKET e o servidor... a nossa empresa paga por cada comunicação... estou aproveitando a comunicação da própria leitura para

mostrar a ociosidade... na realidade eu mudei o nome... não chamo de ociosidade... mudei para intervalo entre leitura... eu achei pesado colocar ociosidade... sei que não ia ficar bom para os leituristas... já sei o que eles pensam... então perguntei para o gerente da leitura e ele achou melhor colocar intervalo entre leitura... o registro de ociosidade ia mostrar apenas o tempo de intervalo sem executar a leitura e não ia ter a hora que iniciou e finalizou... e é claro que é isso que o gerente quer... exemplo... ele fez 15 minutos de intervalo... mas qual foi a hora que ele iniciou e a que finalizou o intervalo" (Analista da SERVPUB).

Neste caso, os analistas e programadores da SERVPUB, com o conhecimento real da atividade dos leituristas, sabiam que estes não ficavam ociosos entre uma leitura e outra. O termo "ocioso" está incorreto. "(...) eu *achei pesado colocar ociosidade... sei que não ia ficar bom para os leituristas... já sei o que eles pensam (...).* Os analistas tinham um julgamento de valor compartilhado com os leituristas sobre o tempo entre as leituras. Ociosidade para eles significava estar sem atividade, é como se eles estivessem, no período, desocupados ou "fazendo hora". O termo que os leituristas utilizam para expressar o tempo gasto entre uma leitura e outra é "intervalo de leitura". Para eles esta expressão denota que estão em atividade e não ociosos. Os analistas sabiam que o tempo entre as leituras podia variar em conseqüência de diversos fatores inerentes à atividade (distância entre as residências, cliente que demora a atender, sistema lento, cliente que solicita informações e outros).

Os analistas e usuários estão integrados de tal maneira que compartilham valores e significados. Pode-se dizer que as pessoas envolvidas no DS (analistas da SERVPUB e usuários) são uma comunidade de prática (seguindo a definição de Wenger). Isto porque, além do repertório compartilhado (linguagem, símbolos), do envolvimento mútuo entre analistas e usuários em torno de um mesmo objetivo (construção de um software eficiente), existe um empreendimento conjunto (e, conseqüentemente, negociação mútua e responsabilidade, das quais se deve prestar conta).

Fazer parte da mesma comunidade foi o que possibilitou ao analista da SERVPUB conhecer a necessidade do usuário (gerente*): "é claro que o gerente não*

quer só o tempo de intervalo... ele também vai querer saber o horário que iniciou e o término de cada leitura, eu já sei o que eles querem... já fiz muitos relatórios... a gente só aprende com a experiência... mas é difícil passar isso à distância... eles tinham que estar aqui".

O analista da SERVPUB já concluiu outros relatórios, nos quais também foram criadas regras para atender as necessidades dos usuários, conforme se vê em sua verbalização: *"conheço mais a empresa, sei dos detalhes... já fiz outros relatórios"*. Como mencionado anteriormente, as regras são formalizadas por convenção da comunidade e parte dessas regras é interiorizada pelos indivíduos (Collins, 1992), assim, "*a gente só aprende com a experiência..."*. Portanto, participar de uma mesma cultura é condição indispensável para se ter um entendimento e saber usar as regras criadas por aquela comunidade. É por isso que fica *"difícil passar isso à distância..."*, "*eles tinham que estar aqui..."*.

A principal queixa dos analistas da SERVPUB é a distância física dos analistas da FABRISOFT. Nas entrevistas, os técnicos da empresa analisada informaram que todas as tentativas de terceirização, com técnicos de informática lotados fora da empresa, não tiveram bons resultados. Segue depoimento do analista de negócio da SERVPUB:

"as terceirizações que funcionaram foram quando os analistas ficaram lotados na SERVPUB".

Durante as observações, verificou-se a importância das interações presenciais no cumprimento da tarefa e na constituição da competência. "*A copresença das pessoas facilita a comunicação, a escuta direta ou difusa, o intervir para ajudar, o compartilhar informações úteis à tarefa, a cooperação"* (Jamil, 2004). Diante da necessidade de copresença permanente, seis meses de treinamento dos analistas da FABRISOFT nas dependências da SERVPUB não foram suficientes para que eles se sentissem seguros e retornassem a FABRISOFT. O prazo de permanência estipulado para o término em setembro de 2007 não pôde ser cumprido. Os analistas ainda continuam no setor e continuarão ainda pelo menos até fevereiro de 2008, prazo negociado entre as empresas para o fim do contrato firmado entre a FABRISOFT e a SERVPUB.

"*eles estão fazendo o que a gente faz... porque a equipe está reduzida, aí o que acontece... estão no processo de aprendizagem... vai ser só com o tempo mesmo... eles estão atendendo demandas... mas estão em um processo de aprendizagem... não sei quanto tempo... daqui um ano ou dois, quem sabe? Nessa nova estrutura a gente fica em reunião o tempo todo... então eles estão aprendendo o tempo todo... na realidade estamos trabalhando da mesma forma que trabalhávamos antes ((todos dentro da SERVPUB e na mesma sala))... mudam daqui e mudam dali e no final começaram tudo de novo*" (Analista da SERVPUB).

A individualização dos papéis e a distância física da fábrica terceirizada dificultaram a construção do saber para poder interpretar as regras e adquirir experiência. Além do mais, dificultaram também as trocas e contribuições de cada indivíduo, primordiais para a criação de uma competência coletiva.

O fato de os analistas da FABRISOFT não fazerem parte da mesma comunidade dos usuários dificultou, ou mesmo impediu, que estes adquirissem o saber necessário para realizar um software eficiente.

Um dos principais motivos de fracassos nas vendas de software é o fato de os fabricantes desconsiderarem a ligação indissociável entre a atividade de desenvolvimento de software, a cultura e o contexto. "*A aquisição dos saberes não pode ser considerada exclusivamente como um fenômeno mental e individual, mas como um fenômeno constituído de relações no interior de contextos precisos*" (Lave, 1996 apud Fonseca).

Exemplo de fracasso na venda de um produto ocorreu na empresa analisada. A SERVPUB realizou parceria com uma fornecedora de software em tecnologia de sistemas com o objetivo de construir um software para ser comercializado junto a outras empresas que visassem o mesmo fim.

De acordo com os dados levantados, a comercialização do sistema não foi bem sucedida. O software foi vendido apenas para uma empresa pública de outro Estado. Segundo informações prestadas pelo gerente do setor comercial e pelos analistas da empresa, o sistema encontrou vários problemas na sua utilização, negando-se os adquirentes a pagar as necessárias manutenções. O gerente da empresa sob estudo,

envolvido na comercialização, acredita que o insucesso da venda tenha ocorrido em virtude do fato da empresa adquirente ser também empresa pública e possuir interesses políticos diversos, além da falta de verba. Quanto aos analistas da empresa sob estudo, segue o seguinte depoimento:

> "a fornecedora vendeu para uma empresa pública... mas não deu certo... falaram que foi porque era uma empresa pública e por isto não deu certo... falaram que eles não tinham dinheiro para pagar a implantação... a fornecedora não tinha o que receber... mas também não tinha o que dar... quando nós fomos implantar o sistema na nossa empresa... a fornecedora de software saiu fora e sobrou para a gente...ela não participou da implantação... quem implantou foi a gente... quando a fornecedora foi realizar a implantação na empresa que vendemos o software ela deu o tratamento como se fosse uma fábrica de software... a implantação precisava de um analista... ela mandou um analista disponível para lá que não participou do desenvolvimento do sistema... colocou junto com o pacote e falou assim é esse cara aí que vai fazer a implantação... aí o que aconteceu? Ele não conseguiu dar o atendimento que o usuário lá precisava... ele não tinha o conhecimento que precisava... isso não se adquire da noite para o dia não... a fornecedora não deu a verdadeira importância que tinha que dar... para ela qualquer analista que fosse lá saberia implantar... não houve um tratamento legal para o primeiro cliente" (Analista da SERVPUB).

Segundo este depoimento, os analistas da FABRISOFT seguiram o modelo racionalista da Engenharia de Software: "(...) *ela deu o tratamento como se fosse uma fábrica de software...*". O software não foi adaptado de forma que atendesse as necessidades dos usuários. Apesar das empresas terem o mesmo fim, as práticas são diferentes, inseridas numa cultura própria. Não é possível extrair o conhecimento de uma empresa e utilizá-la da mesma forma em outra empresa, situada em outro estado, com outra cultura.

Buscar as necessidades do usuário para a construção de um software utilizando-se apenas do trabalho prescrito e do conhecimento de outra empresa, que realiza atividades similares, é contrapor-se à natureza da atividade situada. Do ponto de vista da atividade situada, ação, sentimento, valor, formas culturais e históricas não se separam da atividade que é necessariamente localizada, proposital, significativa e potencialmente conflituosa (Lave 1996, p.7 apud Fonseca, 2005).

5.3 SISTEMA COMERCIAL

História

Transcorridos 14 anos de sua implantação, o antigo sistema comercial se encontrava em fase de declínio do seu ciclo de vida, o sistema já tinha sido bastante alterado, numa busca de atender as necessidades dos usuários. Ademais, o sistema representava o ano com apenas dois dígitos e, com a iminência do chamado "bug do milênio", havia necessidade imediata de alterá-lo. A equipe de manutenção era grande e já não mais conseguia atender à crescente demanda de informatização das rotinas comerciais. Uma das demandas que chegou ao setor de desenvolvimento de software foi a solicitação de integração das informações dos setores comercial, operacional e de atendimento. Em função de tudo isso, decidiu-se por desenvolver um novo sistema computacional mais abrangente, flexível e adaptável que fosse capaz de conferir maior agilidade e confiabilidade aos processos comerciais.

Na busca por uma gestão comercial estruturada, o sistema comercial foi desenvolvido com o objetivo de atender, de forma ágil, segura e flexível, as demandas dos seus clientes, incorporando em sua estrutura as políticas, normas de procedimentos e processos estabelecidos pela Empresa.

5.3.1. Características do sistema comercial

O sistema comercial foi concebido para se constituir na ferramenta ideal de prestação de serviços aos clientes da empresa, conferindo ao processo de atendimento maior agilidade e segurança. É um sistema que opera de maneira integrada às funções comerciais e operacionais de atendimento e serviços aos clientes da empresa, proporcionando maior agilidade à prestação de serviços ao cliente.

Projetado e desenvolvido em módulos funcionais, o sistema comercial permite administrar as funções comerciais e de interface (operacional e atendimento). O sistema é composto por dez módulos, a saber:

Quadro 6: Módulos do sistema comercial

Módulos	Funções
1 - Planejamento e Controle	Gerencia os calendários das atividades do ciclo de faturamento, produz informações gerenciais e de interface contábil e operacional. Permite o bloqueio de faturamento, destinação de mensagens e malas diretas em faturas, entre outras. Administra o remanejamento de localidades, atendendo à necessidade de alteração da estrutura organizacional da empresa, conforme a política da empresa.
2 - Segurança e Auxílio	Gerencia as autorizações para execução das funções do sistema pelo usuário, permitindo o uso destas somente para os previamente cadastrados e autorizados, além de orientá-los no seu uso e no preenchimento dos campos das telas.
3 - Informações Municipais	Gerencia as informações dos municípios onde a empresa opera, tais como bairros, logradouros, além de informações dos contratos com a prefeitura para a execução dos serviços. Por extensão, pode conter outras localidades para efeito de marketing e planejamento.

4 - Infra-estrutura de saneamento	Gerencia as informações das unidades organizacionais por: - Diretorias - Superintendências e Distritos; - Zonas de abastecimento e distritos pitométricos; - Sub-bacias de Drenagem. Administra os Setores/Rotas por classe de serviços e bairros, permitindo a distribuição de serviços para as unidades operacionais.
5 - Cadastro	Gerencia as informações dos clientes, necessárias ao Atendimento e Serviços, Medição, Faturamento, Ações de Cobrança e apoio ao Planejamento e Controle Comercial/Operacional.
6 - Atendimento e Serviços	Gerencia as solicitações dos clientes, gerando solicitações e ordens de serviços. Efetua o encaminhamento automático das ordens de serviços para a área responsável, possibilitando a programação, execução e baixa, disponibilizando, desta forma, informações sobre as etapas da execução. Trabalha com cadeias de serviços, onde a baixa de uma ordem de serviço dispara automaticamente a próxima.
7 - Medição e Apuração de Consumo	Gerencia o processo de micromedição tradicional e/ou coletor eletrônico de dados, de acordo com calendários de atividades, resultando na apuração do consumo. Módulo concebido na administração

	de parâmetros de equilíbrio do consumo, absorvendo a tendência individualizada do comportamento de consumo dos clientes da empresa. Disponibiliza, automaticamente, para o módulo de atendimento, a solicitação para geração de serviços, com base em ocorrências encontradas no imóvel quando da execução da leitura do hidrômetro.
8 - Faturamento	Gerencia todas as atividades de faturamento em massa ou individual, cancelamentos, desdobramento e agrupamento de faturas. Gera os lançamentos de tarifas, subsídios, subvenções, descontos, devoluções, serviços, multas, parcelamentos, financiamentos, atualização monetária, etc., bem como efetua o controle da entrega de faturas.
9 - Arrecadação	Gerencia a baixa de faturas pagas pelos clientes, gera lançamentos de acertos de valores pagos incorretamente, controla o crédito diário informado pelos "agentes arrecadores", permitindo os acertos necessários.
10 - Cobrança	Gerencia e apura os débitos e aciona as políticas de cobrança, identificando os clientes inadimplentes e aplicando as sanções cabíveis, bem como acompanha o

	índice de evasão da receita. Disponibiliza, automaticamente, para o módulo de atendimento, a solicitação para gerar ordens de serviços de acordo com parâmetros para priorização das suspensões, tamponamentos e supressões do abastecimento.

5.3.2. Evolução do Sistema

O sistema comercial foi implantado com programas e relatórios básicos estritamente necessários às atividades comerciais básicas. Na época, estabeleceu-se o consenso pela continuidade evolutiva do sistema, atendendo, dentro do possível, as demandas definidas pelos usuários comerciais e de interface. A dinâmica das funções comerciais irão sempre exigir uma evolução do sistema comercial, seja para o atendimento do cliente ou de natureza legal.

O sistema comercial informatizou várias rotinas da empresa, conforme se vê nos exemplos descritos no quadro abaixo:

Quadro 7: Exemplos de informatizações das rotinas da empresa

Consulta de dados de Localidade: -Disponibilização de serviço (por produto: produto A e produto E) -Informações de contrato (operação e faturamento)
Disponibilização de cadastramento do imóvel, através de infra-estruturas (Zonas de Abastecimento – Sub-bacias de drenagem e Organizacional);
Cobrança automática de valores, através da geração e baixa de serviços;
Estatística de ocorrência de leitura por leiturista;
Consistência automática da leitura no ato da digitação;
Faturas avulsas para cobrança de valores de serviços pagos à vista;
Avisos de débitos com tratamentos específicos (mensagens) à natureza do Cliente ou situação cadastral;

Minimização na utilização de papel com a disponibilidade da visualização de relatório em tela;
Programação automática de atividades através de calendários preestabelecidos: - Calendário de processamento; - Calendário de ações de cobrança;
Segurança de dados por unidade de atuação e empregado;

Este sistema é uma ferramenta que operacionaliza o faturamento, a arrecadação, o cadastro e o atendimento de 2.870.000 clientes e 794 localidades, sendo 559 municípios, dentro dos inúmeros modelos de atuações de mercado, definidos pela empresa (Dados fornecidos pela empresa em 2005).

5.3.3. Política tarifária da empresa

O primeiro projeto demandado e realizado pela FABRISOFT foi a alteração do sistema para o reajuste tarifário dos serviços da empresa do ano 2007, adequando os cálculos deste reajuste, que variam a cada ano, segundo a política da empresa e os parâmetros legais permitidos.

A empresa comercializa seus produtos com valores diferenciados para os seus clientes. Existem descontos por categoria (residência, comércio, residência e comércio, indústria e pública), para residentes por área construída (tarifa social) e descontos pela quantidade do consumo do cliente (desconto tarifário). Outros descontos podem aparecer, ou os existentes podem ser modificados, de acordo com a política do Estado, dos Municípios ou por determinação legal.

Além dos descontos da empresa, alguns Municípios oferecem subsídios às suas comunidades, podendo ser parciais ou totais em relação ao valor da fatura. A prefeitura de cada Município é que determina se concede ou não o subsídio, o seu valor e se ele abrangerá alguns ou todos os moradores do respectivo Município. Alguns Municípios concedem o subsídio apenas para o “produto A”, outros apenas para o “produto E”,

alguns vão de acordo com a renda dos moradores da residência, outros de acordo com o volume faturado (ex: residências que consomem até 20 m³ do produto A) e outros.

A SERVPUB tem realizado os reajuste das tarifas de seus produtos anualmente. Segue abaixo o quadro que demonstra a variabilidade das Resoluções da Secretaria de Estado de Desenvolvimento e Política Urbana que interferiram na política de preço da empresa, nos anos de 2003 a 2007.

Quadro 8: Resoluções da Secretaria de Estado de Desenvolvimento e Política Urbana

Resolução	Data de Publicação	Data de Aplicação	Reajuste		
004/2003 de 28/02/2003	01/03/2003	01/03/2003	Reajuste diferenciado por faixa de consumo, entre 29,80% e 35%. O Reajuste de 29,80% foi aplicado para 87% dos clientes da SERVPUB.		
			- CLIENTES CONTRATADOS - (REAJUSTE MÉDIO)		
			Contratado Residencial :	32,92%	
			Contratado Público :	33,60%	
			Contratado Comercial :	31,46%	
			Contratado Industrial :	32,09%	
			CONTRATADO TOTAL MÉDIO:	32,52%	
003/2004 de 26/02/2004	28/02/2004	01/03/2004	Reajuste de 9,50% , com aumento na progressividade das faixas de consumo.		
002/2005 de 02/02/2005	05/02/2005	01/03/2005	Reajuste de 11,77%, com aumento na progressividade das faixas de consumo.		
005/2006 de 21/02/2006	25/02/2006	01/03/2006	- Reajuste médio de 7,60% para clientes do produto A+E- Redução do consumo mínimo de 10m³ para 6m³ - Redução da tarifa do produto E de 100% para 90% em relação ao valor da tarifa do produto A.		
22/2007 de 15/02/2007	17/02/2007	01/03/2007	- Reajuste médio de 6,72% - Categoria residencial até 10m³ : ...- Consumo até 6m³: sem reajuste ...- Consumo > 6m³ até 10m³: reajuste inferior ao IGP-M (3,77%) - Redução da tarifa do produto E de 90% para 60% ou 40% em relação ao valor da tarifa do produto A.		

No ano de 2007 o reajuste tarifário da SERVIPUB atendeu a referida resolução: reduzir em 40 ou 60% os valores de coleta do "produto E", que até então era de 90% do "produto A". Mas, para reduzir o valor do serviço de coleta do "produto E" em 40 ou 60%, a empresa aumentou nos mesmos patamares o valor do serviço do "produto A". Para as pessoas que não utilizam o serviço de coleta do "produto E", o valor do aumento do serviço de fornecimento do "produto A" foi compensado através da concessão do desconto promocional, no qual a empresa compensa o aumento do "produto A" através do referido desconto. Exemplificando:

Quadro 9: Exemplo de fatura do cliente que utiliza os produtos A e E

Situação anterior da fatura de consumo	Situação atual da fatura de consumo
Produto A: 60,00 Produto E: 54,00 (90% do produto A) Valor total: 114,00	Produto A: 60+11,25 = 71,25 (aumento para compensar a redução do produto E) Produto E: 42,75 (60% do "produto A") Valor total: 114,00 (o valor permanece igual à fatura anterior)

Quadro 10: Exemplo de fatura do cliente que utiliza apenas o produto A

Situação anterior da fatura de consumo	Situação atual da fatura de consumo
Produto A: 60,00 Valor total: 60,00	Produto A: 60+11,25 = 71,25 Desconto promocional: 11,25 Valor total: 60,00 (o valor permanece igual à fatura anterior)

5.4. O CASO DO REAJUSTE TARIFÁRIO

Essa demanda foi enviada do setor comercial da SERVPUB ao seu setor de desenvolvimento de software (SDS). Apesar de tê-lo enviado ao SDS, a SERVPUB terceirizou, integralmente, o projeto de reajuste tarifário deixando-o a cargo da FABRISOFT.

Vários problemas ocorreram após a execução do projeto, originando grande quantidade de erros. Para resolver alguns problemas, a gerência de projetos (da SERVPUB) solicitou ao analista Marco Antônio do sistema comercial, especialista na área de faturamento da empresa, que se reunisse com os técnicos da FABRISOFT numa tentativa de identificar e solucionar os erros. Segundo o analista Marco Antônio, não foi possível, em uma reunião, descobrir onde se encontrava o erro, pois não havia participado da construção do sistema. O analista foi julgado pelo colega (analista de sistemas) como "amarrador de conhecimento".

Transcorridos 30 dias da alteração do sistema, apesar da aparente tranqüilidade sem grandes intercorrências, surgiram novos problemas, que foram considerados a gota d'água para os técnicos do sistema comercial; o reajuste tarifário não era mais confiável. Segue um depoimento do Analista Marco Antônio:

"nós forçamos a barra porque não tinha mais jeito de suportar o que estava acontecendo aqui... nós, analistas do sistema comercial... fizemos uma reunião e falamos que do jeito que estava não tinha condições... do jeito que estava não dava para ficar... porque a gente não ia ter controle do sistema que a gente trabalhava... da forma que estava sendo tratada... foi aí que sugerimos esta estrutura que está aqui ((os técnicos da FABRISOFT dentro da SERVPUB)) com essa estrutura a gente ia preparar esse pessoal que está aqui".

Diante dos vários erros, o sistema estava sendo mais dificultador que um facilitador para os usuários, estes estavam sendo submetidos a um retrabalho: tinham que realizar as atividades manualmente, conferir os erros no sistema e, só assim, proceder à alteração de dados no sistema para obter relatórios corretos:

"O consolidado está todo errado... estou somando tudo na mão e depois mudando" (trabalhador do setor de faturamento).

5.4.1 A questão da terceirização e a construção de um sistema como um instrumento de trabalho

O software só é eficiente quando deixa de ser apenas um artefato e se torna um instrumento de trabalho (Rabardel, 1995), mas, para que isso ocorra, é necessário que o profissional em informática consiga identificar as necessidades do usuário e construir um software capaz de contribuir, ao invés de complicar, para a tarefa do sujeito (Clot, 2006).

A partir do momento em que os usuários do sistema comercial não conseguiram mais, através deste, obter os resultados reais do consolidado, o sistema deixou de ser um instrumento de trabalho. Tiveram, então, que deixar de usar a função e fazê-la manualmente. Os analistas da FABRISOFT não tiveram o entendimento necessário para a construção deste instrumento.

Para que os computadores sejam instrumentos de trabalho, eles não podem ser tratados como cérebros isolados, como sugerem os cognitivistas. Os computadores devem ser considerados "próteses sociais" – destinados ao *input* dos seres humanos dentro de uma comunidade. Do mesmo jeito que um coração artificial só pode estar contido dentro do contexto do corpo, os dispositivos do computador só podem estar contidos no contexto do grupo social para o qual ele contribui através do seu funcionamento (Collins, 1992).

Rabardel (1995) contrapõe as idéias do cognitivismo, que tem uma concepção do instrumento informático centrado no objeto, e apresenta uma concepção dos instrumentos centrada no sujeito (abordagem antropotécnica dos instrumentos). Essa abordagem reinsere o homem no cerne do processo de concepção, criação, modificação e usabilidade dos instrumentos. Portanto, o instrumento é tido como uma entidade mista, que considera tanto o sujeito quanto o artefato. "*Este último cede estatuto de instrumento durante a ação. Um artefato passa de seu uso efetivo pela mediação de uma criação instrumental, ela mesma depende da atividade do sujeito. Um instrumento resulta, por conseguinte, de uma dupla seleção progressiva: ao mesmo tempo seleção no artefato das operações realmente necessárias à sua utilização num dado tipo de*

situações e, no sujeito, seleção dos esquemas solicitados pelo uso desse artefato nesse mesmo tipo de situações" (Clot, 2006, p.120).

Segundo Rabardel (1995), um instrumento representa a combinação de um esquema e de um artefato. Dessa forma, um artefato se torna um instrumento quando se insere num esquema, quando é um meio para o usuário poder realizar determinado objetivo.

Assim, "*Uma ferramenta formal não é necessariamente um instrumento real mesmo que cristalize usos possíveis ou preconizados. Ela pode tornar-se um instrumento efetivo se servir para realizar a ação do sujeito. O artefato não é em si um instrumento ou componente de um instrumento (mesmo quando foi concebido pra isso), sendo instituído como instrumento pelo sujeito que lhe dá o estatuto de meio para atingir os objetivos de sua ação*" (Clot, 2006, p.120). Com esse ponto de vista percebe-se a importância do analista em compreender não somente a forma operatória desse instrumento, mas o uso variado que ele dá ao sujeito.

Os problemas do sistema

Um dos vários problemas que ocorreram após a implantação do reajuste tarifário, desenvolvido pelos analistas da FABRISOFT, se refere ao cálculo do subsídio das prefeituras do interior realizado pelo sistema, no qual o valor apurado para o subsídio era superior àquele autorizado pela prefeitura.

"o subsídio é um jargão da empresa... acho que eles não entenderam direito o que significa subsídio" (Analista da SERVPUB).

Na realidade, os subsídios oferecidos por algumas prefeituras são considerados, pelos usuários e analistas de sistemas, exceções do programa. E essa foi mais uma das exceções que o usuário não informou ao analista antes do desenvolvimento do software. O cálculo do subsídio é automático no sistema, sendo os valores percentuais pré-definidos pelas prefeituras. Entretanto, o sistema calculou o valor do subsídio, considerando o valor total do produto, sem a incidência do desconto promocional fornecido pela empresa, conferindo um aumento irregular ao subsídio. Detectado o erro, o analista Marco Antônio e o cliente entenderam que a analista da FABRISOFT não compreendeu o que eram os subsídios das prefeituras. Segue um depoimento:

"a gente falava em subsídio mas acho que eles não entenderam" (usuário).

Segundo a analista da FABRISOFT, ela só ficou sabendo da existência do subsídio após o aparecimento do erro na fatura.

Mesmo após a descoberta da existência dos subsídios, a analista da FABRISOFT sentiu necessidade de manter-se próxima do analista Marco Antônio (especialista no setor de faturamento) e dos usuários, visando um melhor entendimento das exceções durante as correções do sistema. O depoimento abaixo demonstra a necessidade de interações:

> "o Marco Antônio pediu para eu levantar lá com os usuários... porque mês passado a gente teve problema com umas fatura de subsídios de duas localidade... Prado e Barroso... que a gente teve que acertar na mão... aí a gente acertou as faturas mas não corrigiu o erro no programa... a gente viu que estava com problema... mas não sabíamos o que fazer... Marco Antônio é que olhou o programa para nos ajudar e pediu pra gente cancelar as faturas" (Analista da SERVPUB).

Segue um diálogo entre a analista da FABRISOFT e o analista da SERVPUB para o cancelamento das faturas geradas incorretamente.

Quadro 11: Diálogo entre analista da SERVPUB e analista da FABRISOFT

Analista da FABRISOFT: "*o vencimento é dia 29... temos três dias para fazer*"
Analista da SERVPUB: "*primeira coisa... você tem que pegar todo mundo do grupo 106 e 107* ((são os grupos de clientes que tiveram erros na fatura))... *esquece o que foi errado e cancela débito automático... não é cancelar... você vai levar ao banco o estorno do valor... tem uns que conseguiram ir para o banco... são os que têm vencimentos alternativos... quem que tem vencimento alternativo? aquele cara que (...) esse cara você vai ter que debitar... pode ter ido cancelamento? pode... pode não ter ido? pode"*
Analista da FABRISOFT: "*porque você acha que pode não ter ido*?"
Analista da SERVPUB: "*porque o vencimento não é dia 29? o vencimento dele alternativo poderia ser dia 30... não tem problema... não vai também... entendeu? poderia ir no dia 5*

do mês seguinte aí esse cara foi cancelado... q*uem tinha a fatura do dia 25/03 para frente... esta turma aqui está OK... foi cancelado... esta não tem que cancelar... só que esta turma aqui estava no banco e foi debitado aí o que fez? faturou aqui olha... 106 e 107 aqui tá? isso aqui... qual que é o vencimento desta turma... dia 29/03 não é? essa turma que venceu maior que isso aqui oh... foi cancelado e foi enviado correto... tá? porque? todo mundo que tiver diferente desse aqui é alternativo... agora é alternativo como? esse número do alternativo é maior que o dia 25... é... esse cara foi cancelado lá atrás... está entendendo? mas pode não ter dado tempo... e se esse cara tiver fatura automática? pode já estar no banco não pode*?"

Analista da FABRISOFT: "*isso* (...)"

Na realidade, existem várias exceções nas regras e no momento em que elas vão sendo corrigidas outras vão aparecendo. O diálogo acima mostra que aparentemente um simples cancelamento da fatura do cliente, para a analista da FABRISOFT, não ocorria daquela forma. No momento da intervenção, o analista mostrou que fazem parte também do cancelamento os clientes que têm fatura alternativa (o cliente pode alterar a data da fatura) e que esta pode ser faturada antes do dia 29, que é o dia normal de vencimento da fatura. O analista da SERVPUB, ao dizer "*... e se esse cara tiver fatura automática? pode já estar no banco... não pode*?", demonstrou que os clientes detentores de faturas com vencimento entre os dias 25 e 29, com débito automático, talvez não tivessem mais tempo de cancelá-las, pois, na data de realização do cancelamento das faturas, elas já poderiam estar no Banco.

Grande parte dos erros cometidos pelos analistas da FABRISOFT decorreu da falta de entendimento e desconhecimento das exceções e jargões utilizados pelos trabalhadores da empresa, bem como da inabilidade no trato das exceções, como por exemplo, o subsídio e o CF03:

> "esses erros são provenientes da fábrica... olha lá... arroz com feijão é fácil... quando sai do normal..." (Analista da SERVPUB).
>
> "tem várias negociações que são complicadas que a gente trata... o subsídio é um deles... a maioria da fatura está dando certo... o subsídio está dando

> errado... outra exceção é o CF03... o consumo máximo faturado está com problema". (Analista da SERVPUB)

CF03 é jargão utilizado pelos analistas e usuários ao se referirem à regra aplicada às faturas dos clientes que estão acima ou abaixo do valor esperado pela empresa. Os analistas da FABRISOFT não sabiam da existência dessa regra no início da concepção do software. Tal regra só foi comunicada aos desenvolvedores no curso do desenvolvimento, mas não foi por eles assimilada e, de acordo com a analista Joana (da FABRISOFT), o tempo predeterminado não foi suficiente para introduzi-la antes da implantação. Assim, o CF03 é bom exemplo de que não basta apenas conhecer o significado dos jargões utilizados na empresa, é preciso que eles sejam apreendidos e que a sua utilização pelos analistas se torne uma prática, o que possibilitaria a sua inserção no software.

"*só no teste que apareceram os erros... foi aí que o Hudson ((cliente)) me falou de algumas exceções... aí já não dava mais tempo de mudar dentro do prazo*" (Analista da FABRISOFT).

"*ele*s ((analista da FABRISOFT)) *não entenderam o que era CF03* (...)"

A aplicação das regras varia de inúmeras maneiras para os diversos clientes, sendo necessário se ter prática para saber lidar com uma infinidade de detalhes e saber interpretá-los corretamente. Os analistas da FABRISOFT não faziam parte da cultura da empresa e não tiveram convívio suficiente com os usuários para a construção do conhecimento e de práticas necessárias ao desenvolvimento do software. Entretanto, o analista da SERVPUB vem compartilhando dessa cultura há vários anos, e foi a partir dessa aculturação que ele adquiriu a competência para a resolução dos problemas, conseguindo antecipar as disfunções, como o fez, no exemplo acima, ao solicitar à analista da FABRISOFT que verificasse as faturas com vencimentos alternativos (vencimentos diferentes do dia 29, dia normal da fatura), que já poderiam estar no Banco para desconto em débito automático.

É importante a relação presencial para poder compartilhar os saberes, as tomadas de decisões e antecipar situações problemáticas. A interação na atividade de desenvolvimento de software é necessária para a construção de um software eficiente.

A falta do conhecimento das regras, ou a falta de compreensão delas, aliada à ausência de prática, favoreceram o aparecimento de diversos erros no sistema. Dos vários problemas encontrados, o de maior repercussão e magnitude surgiu no consolidado, módulo do sistema que contém informações gerenciais sobre a arrecadação. A partir de então, os analistas da SERVPUB foram convocados para participar da análise dos erros do sistema e os analistas da FABRISOFT foram lotados na SERVPUB.

Na busca do entendimento sobre o desenvolvimento realizado pelos analistas terceirizados, os analistas da SERVPUB perceberam que o reajuste tarifário somente foi considerado, pelos analistas da FABRISOFT, na fase de faturamento, não o sendo na fase do consolidado.

"*a fatura do cliente saiu correta, mas o consolidado ficou todo errado*" (Analista da SERVPUB).

O usuário (gerente do setor comercial) não se deu conta da necessidade de informar aos analistas sobre a existência do consolidado. Para ele estava clara a regra de que toda fatura tem que gerar consolidado. Segue os depoimentos dos usuários:

> "isso para mim é óbvio... eu não iria lembrar de falar que toda fatura tem que ser consolidada" (usuário - gerente do setor comercial).
>
> "eles foram lá e alteraram a fatura... mas não pensaram nos outros" (usuário - gerente do setor comercial).

Na verdade os analistas da FABRISOFT não sabiam da existência do consolidado e isso provocou um erro no sistema. Para os analistas da FABRISOFT, a regra era criar um desconto promocional para o produto "A" e demonstrá-lo na fatura do cliente, mas eles nem foram informados sobre o significado do desconto promocional dentro dos diversos contextos da empresa e do sistema.

> "o sistema comercial tem que mandar informações para vários outros sistemas que recebem dados... ERP... CR04 e outros... só pensaram na fatura... não pensaram nas informações que estão interligadas no sistema comercial... do jeito que foi pensado não consegue mandar correto para outro sistema... o pessoal da fábrica não sabia disso" (analista da SERVPUB).

O consolidado faz parte da cultura da empresa e já tinha sido interiorizada pelos usuários: *"isso para mim é óbvio... eu não iria lembrar de falar* (...). No entanto, os

analistas terceirizados não faziam parte da cultura da empresa e, por isso, não conheciam as regras construídas pelos trabalhadores (usuários) e analistas da SERVPUB, situação evidenciada na fala: "(...) o *pessoal da fábrica não sabia disso*".

Os Analistas da FABRISOFT, com a orientação do cliente (gerente do setor comercial da SERVPUB), criaram um novo código no sistema comercial, o código (9), para o desconto promocional do produto "A". O código (1), referente ao produto "A", já existente no sistema, não sofreu alterações de valor. Dessa forma, a fatura do cliente continha as informações sobre o valor do produto "A" tratado, o valor do desconto promocional e o valor total da fatura, conforme solicitação dos próprios usuários. O quadro 12 exemplifica a fatura do cliente.

Quadro 12: Exemplo de fatura do cliente que utiliza apenas o produto "A"

Descrição de lançamentos	
Produto A: Valor bruto (código / 1)	+ 40,00
Produto A: Desconto promocional (código / 9)	- 10,00
Valor total da fatura:	30,00

No entanto, a criação do código (9), separando o desconto promocional do código (1) referente ao produto "A" tratado, foi o que ocasionou o problema no consolidado. Isto porque o sistema envia automaticamente os dados do código (1), referente ao produto "A" tratado, para o consolidado. Mas, o valor enviado, nesse caso, diz respeito somente ao valor bruto, desconsiderando o desconto promocional. O valor do código (9), correspondente ao desconto promocional, foi enviado automaticamente para os códigos denominados "diversos" do consolidado.

Quadro 13: Fluxo dos valores da fatura do cliente no sistema comercial

Faturamento		Consolidado	
Código	Serviços / Produto	Código	Serviços / Produto
1	Prod A tratado	1	Prod A tratado
2	Produto A Bruto	2	Produto A Bruto
3	Produto E domiciliar	3	Produto E domiciliar
4	Produto E industrial	4	Produto E industrial
9	Desconto promocional		
29	Vistoria		Diversos
30	Multa		
31	Ligação provisória		

Assim, o desconto promocional deveria aparecer no consolidado junto com o produto (1). No entanto, foi para os códigos "diversos" no consolidado. Dessa forma, o sistema entendeu que a empresa arrecadara 40,00 (valor bruto, sem o desconto promocional), quando, na verdade, ela somente arrecadara 30,00, em virtude da incidência do desconto promocional, no valor de 10,00, conforme exemplo representado no quadro 12.

Quadro 14: Exemplo de fatura do cliente que utiliza apenas o produto "A" no consolidado

Descrição de lançamentos	
Produto A: Valor bruto (código / 1)	+ 40,00

O equívoco cometido na apuração do valor do produto "A" tratado determinou a alteração nos valores dos "diversos", que por sua vez também provocou o surgimento de outros erros no consolidado.

Na realidade, os trabalhadores do faturamento entendem que, para o sistema, os códigos acima de 21 são os "diversos" e os abaixo são os produtos. Portanto, o número 9 para eles se refere ao produto. O sistema, porém, não foi definido dessa forma. Não são os códigos menores de 21 que são produtos. Os códigos foram definidos para cada produto (produto "A" tratado, produto "E", produto "A" bruto e produto "E" industrial) e também para os que não são produtos (multa, atualização monetária, ligação

provisória do produto “A” e outros), de forma que cada um possua seu código no sistema. Por exemplo, o código do produto “A” tratado é (1).

Quando se muda de contexto, no caso do faturamento para o consolidado, as regras são utilizadas de diferentes maneiras. No caso do módulo do faturamento, ele é utilizado apenas para a fatura do cliente. Quando os dados vão para o consolidado, são usados por usuários de diversos setores, com objetivos diversos. O reajuste tarifário deveria ter sido construído de forma a atender os dois contextos: faturamento e consolidado. No entanto, os analistas da FABRISOFT utilizaram as regras apenas para atender o faturamento. Pela falta de conhecimento do consolidado, seguiram a sugestão do usuário de criar o código (9) para o desconto promocional.

O usuário não entende a lógica do programa, apenas o utiliza como instrumento de trabalho. Uma das funções do instrumento é demonstrar a fatura do cliente, outra é consolidar os valores das faturas e demonstrá-las em forma de relatório, conforme o modelo de relatório do setor comercial no anexo 1. Com a presença dos erros no sistema, este não estava servindo de instrumento de trabalho para os usuários. Segue um depoimento do usuário:

“*estou tendo que conferir no manual... não dá para confiar nesses dados*”.

Os erros do sistema poderiam ter sido evitados, se os analistas da FABRISOFT tivessem compreendido as atividades reais dos usuários. Dessa forma saberiam da existência dos dois contextos (fatura e consolidado). O usuário não sabe quais são as informações necessárias para repassar aos analistas ou não se lembra de informar.

> “o usuário só entende que o lançamento de produto está aqui e o lançamento de diversos está aqui... na verdade os analistas tinham que ter interpretado... o que era esse desconto? e o que era produto... aí eles iriam perceber que não poderiam ter tratado o desconto da forma que foi tratado” (Analista da SERVPUB).

Para que os analistas terceirizados conseguissem perceber a existência do consolidado, seria necessário que conhecessem a lógica criada pelos usuários e analistas da SERVPUB sobre o que é produto, e o significado dele para os diversos usuários do sistema.

O analista da SERVPUB com o conhecimento interiorizado, já tinha experiência para conseguir alterar o sistema para o novo reajuste tarifário, sem necessitar recorrer às regras. Segundo Dreyfus e Dreyfus (apud Collins, 1992) o sujeito experiente não tem mais nenhuma consciência sobre as tomadas de decisões, ele age sobre as regras empíricas caso a caso sem que seja necessário explicitar as regras subjacentes, apenas faz. Não existe planejamento e reflexão em seus atos, apenas certos elementos de escolha e de análise conscientes para ajudar nas suas decisões. Abaixo, depoimento do analista da SERVPUB.

> "a gente sabe que tem coisa que a comercial pede... mas a gente tem que absorver aquilo ali... mas não pode ser por isso... por isso e por isso... mesmo o cliente pedindo um lançamento (9)... o analista tinha que ter falado assim... eu vou te dar o resultado que você quer... mas não vai ser um lançamento (9)... para o cliente isso é transparente... (9) ou (1) tanto faz... mas para o sistema o (9) e o (1) faz diferença" (Analista da SERVPUB).

No caso em questão, ocorreu inversão de papéis. O usuário definiu as regras do sistema, certo de que tal função é de responsabilidade do analista, cuja atividade é abstrair e generalizar situações reais e específicas, regras abstratas e gerais adequadas à informatização (sobre este assunto, ver Ferreira, 2004).

5.4.2 Competência: a combinação de experiências individuais na solução de problemas

Para as correções dos erros do sistema foi preciso contar com a experiência do técnico Marco Antônio da SERVPUB na busca de soluções para os problemas. Os analistas da FABRISOFT com a ajuda do analista da SERVPUB criaram um conversor capaz de enviar os dados do código (9) para o código (1). Dessa forma o sistema começou a reconhecer o desconto promocional como parte do produto "A".

Mesmo após a conversão, os analistas perceberam que ainda existiam problemas. O analista Marco Antônio, analisando a questão, descobriu que o produto "A" não fora tratado como produto "A" no consolidado.

"o consolidado da SERVPUB é por categoria... volume medido... volume faturado... eu enxergo dessa forma... quando eu falo produto "A" eu enxergo dessa forma" (analista da SERVPUB)".

Para os trabalhadores do faturamento, quando se fala em tarifa do produto "A" pensa-se em categorias, tais como, residência, comércio, residência e comércio, público e industrial. As categorias, por sua vez, são detalhadas em volume medido e volume faturado e outros parâmetros que não foram considerados no código (9). Além disso, se pensa, ainda, nos descontos promocionais do produto "A", preestabelecidos pela empresa, em subsídios das prefeituras e outros. Assim, o consolidado do produto "A" é separado por categoria e por volume medido de consumo do cliente, dentro de cada categoria, para atender os diversos usuários (trabalhadores, gerentes e diretores da empresa).

Apesar de o analista Marco Antônio ser especialista do módulo do faturamento, ele tem o conhecimento de que os módulos são interligados e, assim, qualquer alteração no módulo do faturamento pode interferir nos outros módulos do sistema. Desta forma, quando o analista Marco Antônio se depara com algumas alterações sobre as quais ele ainda não adquiriu o conhecimento dos impactos que elas podem ocasionar nos outros módulos, ele recorre aos colegas.

*"toda vez que a gente vai alterar algum módulo do sistema... a gente pergunta para os nossos colegas se vai interferir no outro módulo... o pessoal da FABRISOFT não tem essa visão"(*Analista da SERVPUB) .

As regras do reajuste tarifário dentro do sistema são diferentes das regras isoladas de cada usuário, tomado isoladamente. No próprio sistema comercial existem contextos sociais diversos, pelo fato de o sistema atender a várias áreas. Assim, só é possível alterar o sistema a partir do compartilhamento dos conhecimentos de cada analista e dos usuários que se constituem em parte integrante dos diversos contextos.

Uma mesma ferramenta (o sistema) *"é uma interseção de vários instrumentos que atendem e se definem por inserirem em atividades diferentes" (Lima, 2008). Esse mesmo sistema pode ser uma ferramenta diferente "de sujeito para sujeito e para um mesmo sujeito de acordo com as situações e os momentos"* (Rabardel, 1995, p.119).

"Existem situações onde se extrapolam as funções convencionais, caracterizando uma atividade propriamente criativa da parte do usuário. São comuns as catacreses instrumentais (Rabardel, 1995; Clot, 1997), quando os usuários inventam novas funcionalidades e utilizações para os instrumentos e produtos que manipulam, tanto no trabalho quanto na vida cotidiana: "*Os objetos técnicos não são jamais ajustados e tampouco acabados*" (Clot, 1999, p.6). De modo geral, nenhum produto ou objeto fala por si mesmo ou funciona de modo realmente automático, exigindo sempre ações compensatórias do usuário para poder funcionar adequadamente" (Lima, Soares e Leal, 2002).

Os diferentes técnicos adquiriram, ao longo do tempo e da história de cada um, ingredientes sociais e históricos, armazenados em forma de patrimônio (Schwartz, 1998), que possibilitam identificar, mais rapidamente, as melhores soluções para o sistema. Assim, os analistas experientes, competentes no entendimento popular do termo, são capazes de prever disfunções em seu ambiente de trabalho e de apresentar soluções.

Para Schwartz (1998), a competência "comporta pelo menos três polaridades diferentes: o grau de apropriação de saberes conceitualizáveis, o grau de apreensão das dimensões propriamente históricas da situação e o debate de valores a que se vê convocado todo indivíduo no meio de trabalho particular".

No entanto, as modalidades de armazenamento na forma de patrimônio não criam indivíduos supostos todos iguais (Schwartz, 1998). Ninguém pode ser de modo igual em todos os registros. Os técnicos de informática da SERVPUB, cada qual com o seu patrimônio, se integram em um coletivo de trabalho como um elemento de suporte para a intervenção no processo. Segue um depoimento do analista da SERVPUB:

> "eu conheço mais da empresa... sei dos detalhes... a gente tem conhecimento do sistema na verdade a gente já mexeu em vários pontos tá... como a gente sabe que a fatura gerada aqui vai ser consolidada lá nos outros... sistemas ou então vai ser enviada para outros sistemas a gente conhece esse caminho todo aqui... ou se eu não conheço procuro o João, Marcelo ou alguém que saiba esse caminho até chegar no consolidado. Porque quando eu mexo com faturamento no consolidado eu preciso de alguém para me ajudar. Até porque para eu gerar os dados corretos aqui e serem consolidados aqui, correto". (analista da SERVPUB)

Assim, cada analista é especialista de um módulo do sistema e detém um conhecimento específico, mas todos eles trabalham integrados, dando suporte uns aos outros sempre que necessário: *"... a gente conhece esse caminho todo aqui... ou se eu não conheço procuro o João, Marcelo ou alguém que saiba esse caminho até chegar no consolidado"*.

"*O problema das eficiências coletivas será o de constituir equilíbrios variados e complementares de ingredientes (da competência), conforme o tipo e o nível da tarefa ou da missão a ser realizada. (...) a eficiência de conjunto não é a soma de competências individuais avaliadas separadamente, mas a combinação dos espectros pessoais que conseguem cooperar localmente, cada qual completando as lacunas dos outros*" (Shwartz, 1998). No lugar de "competência coletiva" Schwartz prefere utilizar a expressão qualidade sinérgica, "construção sinérgica".

"As qualidades sinérgicas não são, portanto, um simples dado, um traço de caráter. Não são evidentemente independentes de predisposições individuais. Pode-se falar, e há quem o faça, em "sociabilidade", em capacidade para trabalhar em equipe. As políticas de trabalho podem mostrar a sua genialidade ao tecer a sinergia ou, pelo contrário, podem mostrar sua fraqueza, ao buscar "implodir as equipes" em concordância com a tradição taylorista" (ibid).

Além da criação das competências coletivas, a sociabilidade, como já mencionado, também é importante para entender os próprios jargões e os diversos significados das palavras utilizadas pelos trabalhadores da empresa. Além do mais, esses significados dentro da cultura da empresa e de seus setores são diferentes das noções universais aplicadas independentemente dos contextos. Tais significados foram construídos pelos trabalhadores (usuários) e analistas da FABRISOFT que desenvolveram o sistema.

Como ocorre na língua, não basta ter conhecimento das regras gramaticais para saber utilizá-las nos diversos contextos. Segundo Collins (1992), as palavras são utilizadas em função de uma história, de relações e convenções sociais atadas ao contexto, não se referem exclusivamente à estrutura intrínseca da língua como questões sintáticas e gramaticais.

Os técnicos em informática estão aproximadamente há doze anos realizando manutenções e novos desenvolvimentos para o software integrado. Nesse meio tempo, os trabalhadores foram adquirindo conhecimento acerca das atividades dos usuários, das regras e da cultura da empresa e compartilhando-o entre os membros. A competência desse coletivo de trabalho, que é essencial para a eficiência do processo, não se adquire de um dia para o outro, mas ao longo do tempo, realizando atividades específicas para a SERVPUB, daí a dificuldade em explicitá-las.

> "a gente tem uma maior dificuldade de passar como o sistema comercial funciona... como ele trabalha... a gente não consegue convencer ninguém aqui... todo mundo vem cobrar sem considerar essa dificuldade... o que eu espero é que continue com esse jeito que a gente está trabalhando agora ((todos na mesma sala))... e que esse grupo ((analistas da FABRISOFT)) dê a opinião deles" (analista da SERVPUB).

Cada analista vem, ao longo dos anos, desenvolvendo atividades para setores diferenciados da SERVPUB que fazem parte dos módulos do sistema comercial. Com isso, cada um foi se tornando parte de um meio cultural específico e, a partir dessa aculturação, os analistas construíram os seus saberes específicos, pertinentes às atividades de determinadas áreas e de partes (módulos) do sistema do qual eles fazem parte. Apesar das especialidades dos analistas, todos compartilham da cultura da empresa como um todo, o que facilitou a elaboração de uma representação das atividades que seriam formalizadas.

> "quem vive no sistema comercial sabe da complexidade dele, eu não tenho condição de pegar um módulo que o analista João tem um determinado conhecimento, falar de um dia para o outro que é meu... tenho que ter um treinamento... preciso de um apoio de alguém que conhece" (Analista Marco Antônio).

5.5 AS DIFICULDADES DE DOCUMENTAR

Sabe-se que existem problemas de falta de documentação do sistema. Ao longo dos anos ocorreram várias modificações e alterações que não foram documentadas. Foram vários os motivos que levaram os analistas a não documentar o sistema, podendo ter ocorrido, em alguns casos, até mesmo negligência. Mas isso não se constituiu no principal motivo para a não documentação. A pressão temporal para entregar a demanda

e a carência de recursos humanos foram os fatores que mais contribuíram para a falta de documentação do sistema.

Quadro 15: Depoimentos dos Analistas da SERVIPUB

"o problema são as atualizações... é muito difícil atualizar... a gente conserta o erro por que é urgente e depois outro e outro... e aí a gente esquece de atualizar" (Analista da SERVPUB).
"são mais de 5000 programas ou objetos... não tem jeito de documentar... chega uma hora que a gente tem que ir atrás do erro... aí a gente faz o que tem que resolver... e se a gente lembra... documenta" (Analista da SERVPUB).
"nós temos documentação aqui que se alguém pegar para entender... vai ficar lendo de 4 a 5 meses... elas não foram atualizadas... nós viemos perdendo recurso e a documentação não foi dada a prioridade... nós sempre trabalhamos com recurso escasso... agora estão dando a importância... mas só que com a equipe que está aqui é inviável" (Analista da SERVPUB).

O tempo que o analista gasta para obter as informações por meio dos documentos é muito maior do que pelo contato presencial. Mas é importante salientar que mesmo feita a leitura de todas as informações documentadas, isso, por si só, não seria suficiente para o entendimento de todas as regras. Muitos acreditam que a falta de entendimento das regras decorre do fato de estarem elas incompletas. Na verdade, as principais causas já foram mencionadas no caso do reajuste tarifário (o desconhecimento ou a falta do entendimento dos significados das regras e a ausência de prática). Por mais detalhado que seja o documento, sempre lhe faltarão as informações das regras que foram interiorizadas pelos indivíduos. No mais, mesmo as regras explícitas precisam ser compreendidas, pois fazem parte de uma cultura específica. Entretanto, para que elas sejam entendidas é preciso que as pessoas estejam sempre interagindo com o mesmo grupo social. Ademais, elas podem ter os significados alterados, de acordo com o contexto, serem modificadas ou outras regras podem surgir.

5.6 ROTATIVIDADE, FLEXIBILIDADE E EXPERIÊNCIA

Além das dificuldades que os analistas enfrentam com o modelo racionalista da "fábrica de software", existem outros dificultadores, a rotatividade do pessoal na fábrica e a imposição legal de licitação para a contratação de uma fábrica de software, devido ao fato de ser a SERVPUB uma empresa pública. O contrato da atual fábrica está no fim e, se esta não vencer a nova licitação, outra ou outras fábricas assumirão o seu lugar e terão que reiniciar todo o processo de construção do saber que os analistas da fábrica atual construíram. O contrato atual termina no mês de outubro/2007 e o pessoal será mantido até fevereiro/2008, enquanto a empresa realiza a nova licitação.

"*se outra for entrar nós vamos voltar no zero... isso tudo que nós fizemos aqui vamos voltar do zero... e outro problema nós estamos sem pessoa... dependemos deles*" (analista da SERVPUB).

As dificuldades encontradas pelos trabalhadores da SERVPUB, contratos em curto prazo e grande rotatividade de pessoal na empresa terceirizada, impedem a existência de um contato social duradouro, ambiente essencial para a construção de um software eficiente.

As novas mudanças nas modernas estruturas institucionais, como a flexibilidade, focalização e terceirização da atividade-meio (ou atividades de apoio), propiciam uma nova relação do trabalhador com a empresa. Segundo Sennett (2003, p.24):

> "a expressão "Não há longo prazo" se constitui em um verdadeiro princípio utilizado pelas instituições modernas que corrói a confiança, a lealdade e o compromisso mútuo. A confiança pode, claro, ser uma questão puramente formal, como quando as pessoas concordam numa transação comercial ou dependem de que as outras observem as regras do jogo. Mas em geral as experiências mais profundas de confiança são informais, como quando as pessoas aprendem em quem podem ou com quem podem contar ao receberem uma tarefa difícil ou impossível. Esses laços sociais levam tempo para surgir, enraizando-se devagar nas fendas e brechas das instituições".

""*Não há longo prazo" significa mudar, não se comprometer e não se sacrificar; os fortes laços sociais, como a lealdade, deixaram de existir, as pessoas mantêm apenas relações superficiais com a empresa*" (ibid, p.25*). "A própria*

instabilidade das organizações impõem a necessidade de os indivíduos serem aventureiros e correrem riscos com seus trabalhos" (ibid, p.90)

A falta de compromisso dos trabalhadores nas organizações leva a uma grande rotatividade de pessoal, o que impede a formação dos laços sociais. Os laços fracos se concretizam no trabalho em equipe, onde a equipe passa de tarefa em tarefa e muda o pessoal no caminho (ibid). "*Não há eficiência possível sem reconstruir a trama social que permite a aquisição e manutenção de experiências concretas, geradas graças ao contato duradouro com o processo produtivo*" (Lima, 2000).

O capítulo 6 discute a face artística da atividade de desenvolvimento de software, mostrando que parte da atividade se constitui em verdadeiro trabalho artístico, consubstanciado na sensibilidade aguçada do analista, permitindo-o expressar as reais necessidades dos usuários e conseguir criar um software "único" que servirá de instrumento de trabalho.

Capítulo 6 - A arte do desenvolvimento de software

A atividade de DS se assemelha mais a uma obra de arte que a um produto industrial, diferentemente da visão mecanicista e tecnocêntrica da engenharia de software. Os profissionais de informática nem sempre assumem a dimensão artística e social de sua profissão. Desde as escolas de engenharia, tornam-se especialistas em máquinas e linguagens formais. Segundo Lévy (1992), essa formação das escolas de engenharia é um pouco como se somente fosse ensinado aos arquitetos a resistência dos materiais e a economia da construção, sem fazer nenhuma alusão à História da arte à sociologia urbana.

"Numerosos analistas de sistemas [*informaticiens*] se tomam por "técnicos puros". Vários dentre eles, ainda quando estão em condições de conceber e de propor, esperam dos clientes uma definição perfeitamente precisa do problema a resolver, de modo que eles teriam apenas o trabalho de <<codificação>>, sem se colocar outras questões. Desse modo, eles infelizmente amputam toda a dimensão imaginativa do seu ofício. De outro modo, um informático que se percebesse como um verdadeiro operador cultural e intelectual, teria certamente uma escuta mais fina dos futuros utilizadores de seus produtos, uma visão mais ampla das conseqüências de suas decisões e um sentido de responsabilidade mais apurado" (Lévy, 1992, p.8, tradução Ferreira e Lima).

A arte do desenvolvimento de software está presente em vários momentos da atividade de DS: na sensibilidade social dos analistas para conseguirem expressar estas necessidades em roteiros (script); na tradução das necessidades reais do usuário e, por meio de toda sua capacidade imaginativa, na codificação dos dados e introdução do software na atividade real. A arte do DS está no caminho que o analista percorre para criar um software "único" que servirá de instrumento de trabalho. No entanto, para que

o caminho encontrado pelo analista seja convincente, é preciso que os resultados não sejam obras do acaso ou um erro dentro do processo. Ao contrário, os resultados obtidos têm que ter validade perante os seus parceiros.

O executivo-chefe de pesquisa e estratégia da Microsoft, Paul Meller, diz em uma entrevista que o software falha porque o seu desenvolvimento é mais uma arte que uma ciência (Apud. Greenfilta, 2006). Nós diríamos o contrário. Falha quando o software é desenvolvido como uma rotina alheia à compreensão das necessidades do usuário, indiferente às relações intersubjetivas entre conceptor e usuário. Quando há arte, porém, ele funciona bem.

A arte do desenvolvedor está justamente em se criar o software que funcione como um verdadeiro instrumento de trabalho. Essa atividade não é apenas a codificação de um conjunto de dados e regras, trata-se de se transformar linguagem prática em linguagem formal de códigos, que atenderá as necessidades dos usuários. Isto torna o seu desenvolvimento complexo e difícil. Para Lévy (1992), uma das grandes dificuldades do programador, geradora de sofrimento gigantesco, é a de respeitar a sintaxe da linguagem da programação e produzir um "texto" coerente.

Quando o sistema comercial da SERVPUB foi construído, os analistas já faziam parte da empresa e participaram do desenvolvimento do sistema anterior. Os analistas vieram ao longo dos anos, fazendo parte do mesmo grupo social. Dessa forma, o analista Marco Antônio, especialista do módulo de faturamento, tem como grupo social o setor de faturamento. Desde a concepção do sistema comercial, os analistas vêm construindo um saber que é específico dessa cultura.

Os analistas da SERVPUB desenvolveram o sistema com base no entendimento das necessidades dos usuários, que adquiriram durante anos de contato com os trabalhadores do setor comercial. Sua arte aparece ao conseguirem construir um instrumento de trabalho que atende usuários de diversos setores da empresa (setor de atendimento ao cliente, setor operacional e setor comercial).

Encontraram eles diversas dificuldades para adequar as atividades reais dos usuários ao sistema, cuja criação exigiu do analista o desenvolvimento de uma

sensibilidade social aguçada, capaz de traduzir as necessidades reais dos usuários por meio da confrontação de diversas formalidades inerentes às suas atividades.

Um exemplo é o caso da criação da fatura do produto "A". O analista precisou discernir o que era "produto" para os trabalhadores e para as gerências, o que isso significava, as mudanças que ocorriam nas regras, quais as exceções (ex: subsídios, descontos por categoria e outros). Adquiriu o entendimento acerca dos significados das regras e como elas estão organizadas para os trabalhadores. Grande parte do conhecimento do analista se deu pelos contatos com os trabalhadores. Segue um depoimento:

"*são muitos anos de convívio com o pessoal do faturamento... as demandas vão chegando e para a gente entender... temos que voltar muitas vezes no cliente.... a empresa é cheia de exceções a gente sabe que tudo depende da política da empresa... aqui tudo muda o tempo todo e tudo tem prazo"* (Analista de sistemas da SERVPUB).

A construção do sistema e as manutenções vêm seguindo uma lógica que atende os diversos usuários da empresa. Um exemplo são os códigos criados numa seqüência numérica, de forma que os números abaixo de 21 seriam considerados produtos e os acima deste, diversos. Essa regra, como já mencionada, foi um consenso entre os analistas e usuários de setores diversos.

"a palavra "diversos" também tinha significados diferentes para os setores da empresa, eu apanhei até entender isso... o setor de contabilidade chamava de "diversos" todos que não são "produtos"... enquanto o pessoal do faturamento separa alguns itens ((multa, atualização monetária)) que para eles não são produtos e nem "diversos" (...) Juntei os dois setores para a gente entrar num acordo" (Analista da SERVPUB).

Com os códigos exclusivos para cada produto e para cada serviço (que não são produtos para a empresa), é possível que os usuários dos diversos setores utilizem os dados de várias maneiras possíveis.

Para o sistema, cada código criado pelo analista tem o seu significado, como, por exemplo, o código 1 (produto A). Exemplificando:

"quando eu falo... eu enxergo dessa forma... quando eu falo produto "A" eu enxergo dessa forma (desenho demonstrado pelo analista no quadro abaixo)".

Quadro 16: Esquema do analista de sistema

Categoria		Natureza	
1	R	D	´´
1	I	D	´´
1	C	C	´0 - *
1	C e R	D	´´
R- Residência		D- débito	
C- Comércio		C- Crédito	
I- Indústria			
A natureza do lançamento é débito ou crédito. Utiliza-se este mecanismo para saber se o sistema irá somar ou subtrair.			
A regra do sistema é somar todos os lançamentos e onde houver *, subtrai-se.			

O esquema acima foi elaborado pelo analista após adquirir o entendimento da política tarifária da empresa, em diversos contextos (reajustes tarifários, subsídios descontos e outros), bem como das necessidades dos diversos usuários, como por exemplo, trabalhadores (usuários), que necessitam dos valores das faturas do produto "A" de cada cliente, separados por categorias: residência, comércio, residência e comércio, público e industrial, para realizar relatório gerencial. Dessa forma o analista elaborou um esquema no qual foram postas, numa mesma linha, todas as informações do cliente, necessárias para atender os usuários e dados advindos das tarifas praticadas pela empresa. Em cada linha visualiza-se o código (1), que é produto "A", a categoria (R, C, I, D e C) e se a natureza da fatura do cliente é débito ou crédito (D ou C), como demonstrado no quadro acima.

O analista, sabendo dos diversos descontos e subsídios concedidos pela empresa ou prefeituras e da existência de clientes que possuem débito ou crédito com a empresa,

inseriu no esquema a natureza da fatura (D ou C) para facilitar a elaboração do programa que realiza o cálculo. Assim, toda vez que surgir um (D), soma-se o valor do débito na fatura do cliente e quando surgir um (C), subtrai-se o valor do crédito na fatura do cliente. Com a adoção desse procedimento o analista criou uma lógica para o programa somar todos os valores e subtrair apenas quando surgir o símbolo (*) na fatura. O símbolo (*) existe em todos os créditos. Por isso, quando se fala em produto "A" o analista já pensa no esquema: "*... quando eu falo produto"A" eu enxergo dessa forma".*

Os analistas organizam parcialmente o trabalho, as atividades, o universo de signos dos usuários de seus produtos de forma que atenda as necessidades reais do sistema. No caso do sistema comercial, os analistas construíram os dados e as funções para que os usuários realizem atividades diversas em diversos contextos. Os usuários do faturamento apenas consultam os dados cadastrais do cliente, já os usuários do setor de atendimento ao cliente ou os usuários dos setores operacionais são os que lançam os cadastros. As atividades são interligadas, uma depende da outra. Por isso, os analistas responsáveis em cada módulo estão sempre interagindo entre si. Um exemplo é a transferência de dados do módulo do faturamento, pertinentes à fatura do cliente, para o consolidado, onde as regras são utilizadas de diferentes maneiras. Assim, no módulo do consolidado, os dados são utilizados com objetivo diverso do adotado no módulo da fatura do cliente, como por exemplo, na elaboração de um relatório gerencial ou apenas em uma consulta. Portanto, o sistema foi construído de forma a atender os dois contextos: faturamento e consolidado.

A separação entre produtos e não produtos, cada um com seu código e detalhamento da fatura, foi estabelecida para atender os usuários do consolidado e também para adaptar o programa à política tarifária da empresa e às suas diversas alterações, discutidas no capítulo 5, tais como, o reajuste tarifário anual, os descontos por categoria (residência, comercio, residência e comércio, indústria e pública) para residentes por área construída (tarifa social), descontos por volume consumido (desconto tarifário) e demais descontos, tanto os que possam surgir quanto os já existentes, que podem ser modificados de acordo com a política do Estado, dos Municípios ou por determinação legal.

Além dos descontos concedidos pela empresa, existem também os subsídios das prefeituras. Cada município é que determina se concede ou não o subsídio, sendo que alguns municípios concedem o subsídio apenas para o "produto A", outros apenas para o "produto E", alguns vão de acordo com a renda dos moradores da residência, outros de acordo com o volume faturado (ex: residências que consomem até 20 m³ do produto A) e outros.

Os analistas foram construindo um saber das atividades dos usuários e da política da empresa em diversos contextos e, com a sua sensibilidade para obter o entendimento e a competência coletiva, construíram o sistema que vem sendo utilizado como instrumento de trabalho. A tela representada no quadro abaixo mostra como o instrumento de trabalho é utilizado pelos seus usuários:

Quadro 17: Tela do sistema comercial

```
                          Cadastro                              19/11/07
                        Tabelas                                 11:11
                      Tipos de Produto


              Opcao    : P

          Codigo  : A_

              Descricao: produto "A" tratada_____________________

              Classe produto: "A"_


Comando: _____________________________________________________________ 3.T.Y

SERVPUB                        Faturamento
sistema comercial-DESEN                        Tabelas
SC16TB00                        Lancamentos


       Opcao          : P

       Codigo         : 1__

       Tipo servico   : 001

       Nome resumido  :PRODUTO "A" TRATADA___

       Nome           : :PRODUTO "A" TRATADA
TRATADA_____________________________________

                               Gera lancamento

         Atendimento de Servico              Processamento grupo
                   N                                  S
```

Comando: _____ 6.T.B

Esta tela, manuseada por diversos usuários da empresa, foi construída de forma que pudesse ser utilizada para a criação ou consulta de um cadastro, para realizar a fatura do cliente e ainda para consolidar os dados cadastrais. A primeira parte da tela destina-se aos dados cadastrais do cliente e a segunda parte à fatura do cliente. Os dados da fatura aparecem separados para serem utilizados no consolidado. Foi dessa forma que o analista organizou (parcialmente) o trabalho dos usuários e é por meio das opções e seqüências que utilizam o software.

O caso do vale-serviço

Outro caso analisado foi o desenvolvimento de um programa que deveria atender a nova atividade da empresa, denominada vale-serviço: trocar latinhas e garrafas-pet por desconto na fatura do serviço da empresa. O analista já possuía o domínio da fatura, mas o serviço era novo e seria implantado após o desenvolvimento do software. No caso, o analista da empresa, juntamente com a analista terceirizada (com um ano e meio de experiência na empresa), desenvolveu a primeira etapa do sistema. A segunda etapa foi desenvolvida pela analista terceirizada, enquanto o analista da empresa teve o papel de analista de negócio, seguindo as mudanças organizacionais do setor.

Inicialmente imaginaram uma situação e, na primeira etapa, o analista e a analista terceirizada dividiram partes do desenvolvimento. Foi um processo de escritura coletiva em várias etapas, que contou também com a participação dos usuários.

A demanda

O usuário informou ao analista as suas necessidades e as funções que achava necessário conter no software. A tarefa do atendente comercial, usuário do sistema,

seria recolher o material reciclável e descontar na fatura do cliente (consumidor) o valor estipulado por peso do material. O material recolhido seria vendido a um reciclador. De início, havia algumas idéias que foram sendo alteradas no decorrer dos encontros entre analista e usuário. Várias perguntas surgiram, na tentativa de buscar informações sobre a nova atividade, como por exemplo: Como os atendentes iriam receber esse material? Como iriam utilizar o sistema? O analista de sistema e o usuário realizaram vários encontros para estudar a melhor maneira de se formalizar a atividade. O depoimento do analista de negócio no quadro abaixo descreve partes de alguns diálogos com o usuário nessa tentativa:

Quadro 18: Depoimento dos analistas da SERVPUB

"((usuário*)) eu tenho que poder abrir lote e fechar o lote... aí eu perguntei pra ele... o que é lote? ah... porque é o seguinte... o pessoal vai entregando lata no atendimento... aí depois quando eu vejo que tem muita latinha lá eu vejo que tem que fechar o lote... vou pegar tudo e vou enviar tudo pra o reciclador... aí ta bom... mas como é que você vai enviar isso para o reciclador? a gente não pode pegar um material de uma pessoa jurídica... pôr num caminhão sem uma nota fiscal, não é? principalmente se ele estiver vendendo ((Analista))... ah... então tem a geração da nota fiscal... .isso tudo é a gente conversando... então ta bom... a gente vai gerar uma nota fiscal... criamos uma função pra gerar uma nota fiscal pra ele... pra que? pra ele poder enviar o material pra lá... ah... mas essa nota fiscal pode ter tido um erro... por que? a quantidade de lata que foi é diferente da quantidade de lata que chegou lá... ah é? então o que o cara vai fazer? vai fazer um ajuste... vai fazer uma correção na nota fiscal... olha aqui... depois que eu abrir um lote que eu receber o material... que eu gerei uma nota fiscal... já fiz um ajuste... o que eu tenho que fazer... tudo tem uma seqüência... eu tenho que cobrar desse cara... não tenho? a geração da fatura aí... tá vendo?* (Analista da SERVPUB).

Como a atividade ainda não existe na empresa, nota-se no depoimento acima que o analista e o usuário tentaram antecipar as situações que poderiam ocorrer na atividade. Com a experiência adquirida no setor de faturamento da empresa, o analista e o usuário

sabem que vão ter que gerar nota fiscal para o produto que a empresa vender: "*a gente não pode pegar um material de uma pessoa jurídica... pôr num caminhão sem uma nota fiscal..., não é? principalmente se ele estiver vendendo*", isso é rotina da empresa, existe uma norma fiscal. Os analistas possuem a vivência de que os atendentes necessitam de alternativas para a correção de eventuais erros que eles cometem durante as suas atividades e, por isso, anteciparam: "*mas essa nota fiscal pode ter tido um erro*", mas não detinham o conhecimento da atividade para saber quais os tipos de erros que poderiam ocorrer. Foi por intermédio de experiências acumuladas no setor de faturamento e na aguçada percepção dos analistas na busca das possibilidades que poderiam encontrar, tais como *"eu ficava pensando em tudo que poderia acontecer... um dia eu vi uma lata caindo de um caminhão e já fui logo pensando... e se no deslocamento dessas latas alguma cair? as latas podem chegar numa quantidade menor do que estava na nota fiscal" e "a quantidade de lata que foi é diferente da quantidade de lata que chegou lá",* que os analistas desenvolveram o software.

A formalização da demanda foi conseguida por meio das diversas interações entre o analista e usuários. Segue um depoimento:

Quadro 19: Depoimento da analista da SERVPUB

"foram várias reuniões. Fui lá, a gente conversou a primeira parte, aí quando eu comecei a fazer eu vi, opa! tá faltando isso: aglomerado. Eu cheguei num ponto que eu fiz isso aqui, tava pra a empresa toda, só que aí tem um detalhezinho, eu falei pra ele assim: é pra empresa toda? não, é só pra aglomerado. Aí eu perguntei pra ele: o que é aglomerado? e ele já tinha esse conceito lá. Mas a gente não tinha informatizado o que era aglomerado. Como é que funciona? Aí eu fui desenvolvendo. Aí a gente desceu, conversou oh! Participa de uma localidade, sei lá... dentro desses aglomerados aqui eu tenho vários clientes lá dentro. Isso aí que a gente bolou. Como é que você quer enxergar isso aqui na tela? Ah, você me dá um nome e tal aí... nós é que bolamos essa saída aqui oh. Primeiro ele informa a localidade. Vem pesquisando porque? Porque primeiro eu tenho que ter a localidade, depois eu tenho uma agência de atendimento e depois eu tenho os aglomerados. Porque tudo tem uma seqüência. E os imóveis é que fazem parte daquela localidade. O município é uma localidade. Hoje eu falo isso tudo, mas a gente reuniu várias vezes. A gente voltou lá e falou assim: desse jeito não vai dar certo. Vamos pensar pra este lado" (Analista da

SERVPUB).

O software foi desenvolvido com as seguintes funcionalidades:

- Abrir um lote para receber o material; fechar o lote quando o espaço físico para guardar o material estiver cheio. O sistema permitirá somente um lote em aberto por agência.
- Receber o material reciclável dos clientes. Através de um registro do material recebido o sistema calculará o valor do crédito.
- Incluir o lançamento do crédito na fatura do cliente.
- Emitir o comprovante do crédito onde constatará as informações de quantidade, valores e o mês de referência com o valor do crédito em conta.
- Criar relatórios do material recebido.
- Quando fechar o lote e o material for enviado para a recicladora, emitir a fatura avulsa para a empresa recicladora após a conferência do peso pela recicladora.
- Se for preciso, ajustar a alteração do peso e emitir a fatura avulsa com o novo valor.
- A SERVPUB irá definir o preço do material.
- Apenas uma recicladora poderá comprar os materiais.

Este software não atendeu as necessidades do usuário devido a uma situação real que não foi prevista. No momento de fechar um lote e repassá-lo a recicladora, viu-se que existia uma recicladora que só comprava latas e outra que comprava os dois

materiais (latas e garrafas pet), mas esta pagava um preço menor. O usuário (trabalhador do setor comercial) disse que tinha dificuldades em negociar preço com a recicladora que comprava os dois materiais. Antes, utilizando-se a tabela de preço da SERVPUB, realizava-se um leilão e vencia quem melhor negociasse com a empresa. Mas isso ficou inviável para a SERVPUB, pois não havia muita concorrência. Diante disso, o usuário solicitou que o software fosse adaptado para poder atender a mais de uma recicladora. A demanda gerou grande dificuldade para os analistas, pois o software foi todo concebido seguindo a lógica de que a empresa venderia os materiais apenas para uma recicladora. Com a mudança, o software deveria sofrer uma grande reformulação. Após várias discussões entre analistas e usuários, decidiu-se que o software deveria ser refeito para que a empresa pudesse operar com mais de uma recicladora.

Mesmo tentando antecipar as situações que estão por vir, é incerta a previsão de todas elas. Várias são as situações singulares, um exemplo foi o problema ocorrido com as recicladoras. Além disso, nem sempre se tem uma representação correta da atividade do usuário, o problema surge da interface do sistema com os processos reais de venda. Como era um produto novo, não havia experiência acumulada, daí a necessidade de realizar alguns ajustes para melhor atendê-lo.

Dessa forma, o problema, com as recicladoras, não foi resolvido com um simples reajuste, o software teve que ser refeito integralmente. Esta situação demonstra claramente como uma pequena mudança no mundo real pode determinar grandes mudanças na formalização (informatização).

A organização do sistema

Repassada a demanda, os analistas Maria da Conceição e Marco Antônio começaram a analisar como deveriam proceder para desenvolver o sistema:

"Na hora que o usuário falou que tinha que ter recicladora diferente aí a gente já foi repensando o que estava feito. Ah não, na inclusão do lote já não vai poder ser

mais assim a partir de agora a gente vai ter que tratar tudo diferenciado por material. Aí a gente já foi enxergando o que estava pronto e o que tinha que mudar"

É por meio do entendimento da atividade dos usuários que os analistas vão organizar o trabalho no sistema. Isto seria o mesmo que ditar as regras de como os trabalhadores deverão realizar as suas atividades. Pode-se dizer que, para o sistema ser eficiente, os analistas devem entender a atividade real do usuário e assim poder reorganizar a atividade dentro de um programa de computador. Esta é uma arte que os desenvolvedores realizam: entender a atividade dos trabalhadores, que é informal, e organizar as atividades em um sistema, que é formal. Como já foi dito, no caso do vale-serviço a situação era ainda mais difícil, o analista tinha que desenvolver um software para uma atividade que ainda não existia. Mesmo após a implantação do software continuaram a pensar em situações que poderiam ocorrer na nova atividade. Seguem abaixo partes do raciocínio do analista Marco Antônio durante o desenvolvimento do sistema.

> "por exemplo, se você fosse lá pra entregar lata e pet, só que seu irmão chegou e trouxe mais três latinhas aqui pra entregar para você. Como é que eu ia fazer pra poder somar essa lata que você está trazendo com essa aqui, aí eu tive esta dificuldade, aí eu ia programar mais eu não sabia como eu ia fazer isso, eu tive essa dúvida. Fiquei só pensando: eu ia perguntar pro Marco Antônio. Aí eu pensei: eu não preciso fazer isso em momento diferente não, eu vou tratar o lote de lata recebendo lata e lote de pet do mesmo jeito e vou continuar emitindo recibo do mesmo jeito. Aí levei pra ela (usuária) e essa tela aqui a gente estava com essa questão, a gente não tinha definido com ela como a gente ia emitir esse recibo, se a gente ia criar uma opção que retira o recibo daqui e criar uma opção lá no menu pra poder emitir o recibo onde você ia ter a possibilidade de marcar qual o material que você queria que saísse no recibo. Mas na hora que você tivesse essa opção fora, você concorda que a emissão do recibo ela perderia o ato do recebimento?" (Analista da SERVPUB).

Os analistas não utilizaram nenhum método para a construção do software. As novas situações iam aparecendo no decorrer do desenvolvimento, como *"(...) se você fosse lá pra entregar lata e pet, só que seu irmão chegou e trouxe mais três latinhas aqui pra entregar para você. Como é que eu ia fazer pra poder somar essa lata que você está trazendo com essa aqui, aí eu tive esta dificuldade",* e novos caminhos e negociações tiveram que ser realizados para a construção do sistema.

O profissional de informática é uma espécie de escritor, mas de uma escrita não repertoriada e sim a definir (Lévy, 1992). Este profissional não segue padrões, ele os

cria a partir de cada nova realidade que se lhe apresenta, transformando-a em artefato interativo.

Capítulo 7 - Conclusão

O modelo "fábrica de software" surgiu como tentativa de solucionar os problemas encontrados no processo de desenvolvimento de software, tais como, sistemas ineficientes e não funcionais, erros de programação, prazos ultrapassados no atendimento às demandas e orçamentos estourados. No entanto, o emprego desse modelo, utilizando-se de métodos da engenharia baseados em modelos fabris, não vem correspondendo às expectativas de garantia de melhoria da qualidade do software e aumento de sua produtividade, pretendidas pelos fabricantes de software em geral. Isto porque o modelo adotado traz em si a pretensão de que o saber é passível de transferência, dos usuários para os técnicos em informática, por meio de regras e modelos formais, no interior de uma divisão do trabalho dentro da equipe e pela formalização do processo de desenvolvimento do software, incluindo a própria atividade dos analistas. .

Utilizando-se da metodologia da análise da atividade, foi possível demonstrar aqui que a separação presencial do desenvolvedor, suprimindo-o do convívio com o usuário, bem como a individualização dos papéis, conforme determina o modelo “Fábrica de software”, dificultou a construção social do saber dos analistas, fundamental para desenvolver uma experiência coletiva e poder interpretar as regras, viabilizando a construção de um software de qualidade. Assim, foi possível confirmar a hipótese de que a atividade de desenvolvimento de software tem toda uma dimensão social (inclusive artística) e por isso a aquisição do saber, necessária para a construção de um software eficiente, somente é possível pela aculturação dos membros envolvidos.

Mesmo o conhecimento formal (regras e heurísticas) possui uma parte explícita e outra não. As regras se constituem em uma construção social entre os membros

envolvidos que, com o tempo, serão interiorizadas pelos indivíduos e dificilmente serão lembradas ou exteriorizadas. Todas as mudanças e criações de regras ocorrem mediante um acordo social, por meio de convenções, que se traduz em um acordo das práticas comuns. Daí a importância da aculturação entre os técnicos de informática e usuários, dentro do ambiente de trabalho destes últimos, para a construção do saber. Conforme se demonstrou nas situações analisadas, quando os desenvolvedores estão distantes geograficamente da empresa contratada para o desenvolvimento do software, a comunicação com os usuários se dá por meio de reuniões, remessas de documentos e telefonemas, impedindo a construção do saber necessário para se desenvolver um software com qualidade. O depoimento da analista da FABRISOFT, quando diz: "*(...) eu não consigo estar fora daqui... o conhecimento está aqui... o ambiente está aqui... o apoio está aqui (...)"* mostra a importância da aculturação entre os técnicos de informática e usuários para a construção do saber.

Além do mais, as trocas e contribuições de cada indivíduo são primordiais para a criação de uma competência coletiva. A atividade de DS demanda constantes tomadas de decisão, pois se constitui na construção de um processo formal que ainda não se encontra pronta na mente do seu usuário, e que será elaborada no curso desse processo.

Os diferentes técnicos da empresa analisada adquiriram com a experiência saberes individuais, que dentro de um coletivo de trabalho são combinados completando as lacunas dos outros, formando uma competência coletiva. Dessa maneira, os analistas da SERVPUB vinham desenvolvendo suas atividades, cada qual como especialista em um módulo do sistema e, portanto, detentores de um saber específico, mas todos eles trabalhando de forma integrada, dando suporte uns aos outros sempre que necessário.

O sistema comercial da empresa atende a contextos sociais diversos. Assim, somente é possível a alteração do sistema a partir do compartilhamento dos conhecimentos de cada analista e dos usuários, que se constitui em parte integrante dos diversos contextos.

Os diferentes técnicos adquiriram, ao longo do tempo e da história de cada um, competências com "*ingredientes sociais e históricos, armazenados em forma de patrimônio"* (Schwartz 1998), que possibilitam identificar, mais rapidamente, as melhores soluções para o sistema. Assim, os analistas experientes são capazes de prever

disfunções que ainda estão por vir em seu ambiente de trabalho e de apresentar soluções.

A competência de cada analista é valorizada dentro de uma coletividade na medida em que as combinações das experiências individuais evitam os erros e ajudam na solução de problemas. Pôde-se ver nesta dissertação como a experiência do analista da SERVPUB foi imprescindível para ajudar os analistas da FABRISOFT no momento da busca de soluções para os problemas do sistema no caso do reajuste tarifário.

Dessa forma, a interação ocorrida entre os analistas da SERVPUB e da FABRISOFT solucionando problemas, foi fator determinante para a retomada do modelo organizacional adotado anteriormente à implantação da "Fábrica de Software", conforme denota o depoimento abaixo do analista da SERVPUB:

> "(...) na realidade estamos trabalhando da mesma forma que trabalhávamos antes. Mudam daqui e mudam dali e no final começaram tudo de novo".

Assim, fica visível que a prática dos desenvolvedores de software é social, que a interação entre os membros (analistas e usuários) envolvidos é essencial para a construção do software. Mas, pode-se dizer que apenas a interação não basta para a construção de um software de qualidade. Precisa-se mais do que isso. As pessoas envolvidas têm que se constituírem em uma comunidade de prática, cuja característica "se revela em três dimensões – engajamento mútuo, empreendimento comum e compartilhamento de repertório" (Wenger, apud Frade 2003). Os analistas da FABRISOFT não integram uma comunidade prática, eles mantêm apenas relações superficiais com a empresa, impossibilitando a criação em seu ambiente de trabalho das três dimensões que caracterizam essa comunidade.

Os contratos em curto prazo e a grande rotatividade de pessoal na empresa terceirizada impedem a formação de laços sociais duradouros, essenciais para a construção de um software eficiente. "*Os laços fracos se concretizam no trabalho em equipe, onde a equipe passa de tarefa em tarefa e muda o pessoal no caminho*" (Sennett, 2003), dessa forma, o profissional não tem comprometimento. O fato da SERVPUB ser uma empresa pública impõe que o contrato firmado com a fábrica terceirizada seja por prazo determinado e, se esta não vencer a nova licitação, outra ou outras fábricas assumirão o seu lugar e terão que reiniciar todo o processo de construção

do saber que os analistas da fábrica atual construíram. Isto inviabilizaria por completo o surgimento de uma comunidade de prática.

A atividade de DS é uma construção social que envolve tanto os usuários quanto os analistas. Viabilizar ações que conduzam as pessoas envolvidas a se tornarem uma comunidade de prática é a forma mais coerente para se construir um software que atenda as necessidades reais do usuário. A utilização do modelo "Fábrica de software" como forma de garantia de qualidade não resolve os problemas reais existentes na atividade de desenvolvimento de software. A sistematização da atividade de DS, diante dos conceitos genéricos das metodologias existentes, não leva aos resultados esperados de melhoria e podem estar distantes do que o cliente considera qualidade.

Portanto, minimizar os problemas decorrentes da falta de entendimento das necessidades reais do usuário, que afetam a qualidade do produto, é questão fundamental a ser enfrentada pelos fabricantes de software, que deverão introduzir novas formas de organização que permitam a constituição de uma comunidade de prática.

A autora, ao fazer uma análise crítica da dissertação, acredita que foi possível demonstrar a dimensão social da atividade de desenvolvimento de software, necessária para a construção de um software de qualidade. Mas, quanto a proposta de demonstrar a dimensão artística da atividade de desenvolvimento de software, esta poderia ter sido melhor evidenciada. No período em que foram realizadas as observações para confirmar a hipótese da dimensão artística da atividade, os analistas da empresa estudada eram os responsáveis pela realização das atividades de desenvolvimento de software e não dispunham de tempo para implementar as auto-confrontações necessárias, segundo a metodologia utilizada, para explicitar os procedimentos concretos e os significados latentes do comportamento observável, o que dificultou melhor evidenciar a dimensão artística da atividade desses profissionais.

No decorrer da presente dissertação foram mencionados alguns outros problemas que ocorrem na atividade de desenvolvimento de software, tais como, pressão temporal; falta de documentação; contratos em curto prazo e a rotatividade de pessoal da empresa terceirizada, que ainda não foram resolvidos e que merecem ser melhor analisados.

Capítulo 8 - Referência

AGYDELO, L.P.P.; SOUZA, M.S.L. *Premissas para o design conceitual de Sistema de Informação Ambiental (SAI) a partir da abordagem teórica de Wenger*. Desenvolvimento e Meio Ambiente, n.7, p.95-105, Jan./Jun. 2003. Editora UFPR. Disponível em: http://calvados.c3sl.ufpr.br/ojs2/index.php/ made/article/viewFile /3053/ 2444. Acesso em 7/02/2008.

ALEXANDER, C. *Ensayo sobre la syntesis de la forma".* Buenos Aires: Ediciones Infinito, 1969.

BOUTINET, J. –P. *Antropologia do projeto.* Tradução de Patrícia Chittoni Ramos. 5 ed. Porto Alegre: Artmed, 2002.

BUTTON, G. et al. *Computadores, mentes e conduta.* São Paulo: UNESP, 1998.

BRITO, J. *Metodologia para Gestão do Processo de Qualidade de Software para Incremento da Competitividade da Mobile.* Disponível em: <http://www.Maria da Conceiçãot.gov.br>. Acesso em 25/05/2006.

CANÊDO, L.B. *A Revolução Industrial.* São Paulo: Atual: Campinas: Unicamp, 1987.

CARVALHO,L.C. *Análise de sistemas: o outro lado da informática*. Rio de Janeiro: Livros Técnicos Científicos, 1988.

CLOT, Y. *A Função Psicológica do Trabalho.* Editora Vozes. Petrópolis, 2006.

COLLINS, H.M. *Artificials experts*. Cambridge: MIT Press, 1992.

DANIELLOU, F. et al. *Ficção e realidade do trabalho operário*. Ver. Brás. De Saúde ocupacional, 1989.

DANIELLOU,F. *A Ergonomia em Busca de seus Princípios: Debates Epistemológicos.* Editora Edgard Blucher, 2001.

DERTOUZOS,M. *O Que Será: Como o Novo Mundo da Informação Transformará Nossas Vidas*.Tradução Celso Nogueira. São Paulo: Companhia das Letras, 1997.

FERNANDES, A.A. e TEIXEIRA, D.S. *Fábrica de software: implantação de gestão de operação.* São Paulo: Atlas, 2004.

FERREIRA, R.B. *Diálogos de surdos: a difícil explicitação do saber entre programadores de software e operadores de fábrica*. Dissertação de mestrado pela Engenharia de Produção: UFMG, 2004.

FERREIRA, R.B e LIMA, F.P.A. *Definição de Requisitos na Concepção de Sistemas Informatizados: da elicitação à cooperação*". In Workshop um olhar Sociotécnico sobre a Engenharia de Software, Rio de Janeiro, 2005.

FERREIRA, RB e LIMA, F.P.A. *Metodologias Ágeis: Um Novo Paradigma de Desenvolvimento de Software*. In Workshop um olhar Sociotécnico sobre a Engenharia de Software, 2, Vila Velha - Espírito Santo , 2006.

FONSECA, G.C. *Aprendizagem, socialização e conflito no trabalho: a dimensão tácita do (des)conhecimento nas organizações*. Dissertação de mestrado pela Engenharia de Produção: UFMG, 2005.

FRADE, C.F. *Componentes tácitos e explícitos do conhecimento matemático de áreas e medidas.* Tese de doutorado da Faculdade de Educação: UFMG, 2003.

FRANÇA,C.A. e SILVA, F.Q.B. *Um estudo sobre Relações de Papéis do RUP e o Comportamento Pessoal no Trabalho e equipe Fábricas de Software*. In Workshop um olhar Sociotécnico Sobre a Engenharia de Software, 3, Porto de Galinhas – Pernambuco, 2007.

GREENFIELD.J *O caso das fábricas de software.* Disponível em: http://www.microsoft com/ brasil/sdn/Tecnologias/arquitetura/FabricasdeSoftware.mspx.. Acesso em 21/05/2007.

GUÉRIN, F, et. al. *Compreender o trabalho para transformá-lo*. São Paulo, Edgard Bluchüer, 2001.

LATOUR, B. e WOOLGAR, S. *A vida de laboratório*. Rio de Janeiro: Relume Dumará, 1997.

LÉVY, P. *De la programmation considérée comme une des beaux-arts*. Paris, La Découverte, 1992.

LÉVY, P. *A Máquina Universo – criação, cognição e cultura informática*. Tradução: Bruno Charles Magne – Porto alegre. Artmed, 1998.

LEAL. L; LIMA F.P.A. *O analista de sistemas, o artesão e a fábrica: os equívocos do modelo da «fábrica de software»*. 14º Congresso Brasileiro de Ergonomia, o 4º. Fórum Brasileiro de Ergonomia e o 2º. Congresso Brasileiro de Iniciação em Ergonomia. Curitiba, 2006.

LEAL. L; LIMA F.P.A. *«Fábrica de software» e saber tácito dos profissionais de informática*. In Workshop um olhar Sociotécnico sobre a Engenharia de Software, 3, Porto de Galinhas – Pernambuco, 2007.

LIMA, F.P.A. *Noções de organização do trabalho*. In: OLIVEIRA, C.R. (organização.) *Manual Prático de LER*. Belo Horizonte, editora Health, 1998, 1998, p.167-190.

LIMA, F.P.A. *A transcedência do valor: flexibilidade, focalização, terceirização e a relação capital-trabalho*, 2000 (mimeo).

LIMA,F.P.A. *Norma e atividade humana: modelos dinâmicos da prescrição e historicidade das situações de trabalho,* artigo publicado em trabalho e abordagem pluridisciplinar: estudos Brasil, França e Argentina. DIEESE/CESITE (Orgs.). São Paulo (DIEESE) e Campinas (CESIT), 2005. pp.51-68.

LOJKINE, J. *A revolução informacional*. São Paulo: Cortez, 1995.

META. *Núcleo de Tecnologia de Software*. Disponível em: http://www.metainf.com.br/index. php? option=com_content&task=view&id =23& Itemid=52#fab Acesso em 21/05/2007.

NOMURA,L., et.al. *FSP – MDP: Um Modelo de Definição de Processos de Fábrica de Software*. XXVI ENEGEP – FORTALEZA, CE, 2006. Disponível em: http://www.abepro.org.br/ biblioteca/ENEGEP2006_TR450304_7257.pdf. Acesso em 3/02/2007

PAULA FILHO,W.P. *Engenharia de Software: fundamentos, métodos e padrões*. Rio, LTC, 2001

PRESSMAN, ROGER. S. *Engenharia de Software*. São Paulo: Makron Boks, 1995.

QUIDGEST. *A fábrica de software do futuro***.** Disponível em: www.quidgest. com/q _ genioes.asp ?lt=esg - 15k. Acesso em 21/05/2007.

RABARDEL, P. *Les hommes et les technologies. Approche cognitive des instruments contemporains*. Paris : Armand Colin, 1995.

SHWARTZ, Y. *Os ingredientes da competência. Um exercício necessário para uma questão insolúvel*. Educação & sociedade, n.65, p. 101-139.

SENNETT, R. *A corrosão do caráter;* trad. Marcos Santarrita. Editora Record – Rio de Janeiro – São Paulo, 2003.

SOARES, RG. *Da dor ao riso: a relação de serviço entre saber fazer e saber atender* Dissertação de mestrado pela Engenharia de Produção: UFMG, 2005.

SQUADRA. *Fábrica de Software.* Disponível em: http://www.squadra.com.br/website/servicos/fabricadesw /fabricadesw.html Acesso em 21/05/2007.

TAVARES, S.R.S. *Da crise do software ao projeto estruturado: a submissão não real do trabalho em programação*. In: FLEURY, A.C.C.; VARGAS, N. Organização do trabalho. São Paulo: Atlas, 1983.

TEIXEIRA,C.A.N. *Um olhar sóciotécnico sobre a Engenharia de software: o caso do BNDES.* Dissertação de mestrado pela Engenharia das Ciências e Computação: COPPE/UFRJ, 2007.

TELLES, A.L.C. *A ergonomia na concepção e implantação de sistemas digitais de controle distribuído.* Rio de janeiro, Dissertação de Mestrado, COPPE, 1995. CAP.2.

VARELA, J.F., THOMPSON, E. e ROSH, E. *A Mente Incorporada: Ciências Cognitivas e Experência Humana*; trad. Maria Rita Secco Hofmeister. – Porto Alegre: Artmed, 2003

ZILBOVICIUS, M. *Modelos para a produção, produção de modelos: Contribuição à análise da lógica e difusão do modelo japonês*. Tese de doutorado pela Engenharia de Produção, USP, 1997.

WISNER A. *La constitution de problèmes, sa description par l'analyse ergonomique du travail »*, texte français d'un article dans *Ergonomics*, réédité *in* A. Wisner, *Réflexions sur l'ergonomie (1962-1995)*, Toulouse, Octarès, 1995.

WISNER A. *La cognition et l'action situées : conséquences pour l'analyse ergonomique du travail et l'anthropotechnologie,* Communication au congrès de l'Iea, rééditée in A. Wisner, Réflexions sur l'ergonomie *(1962-1995)*, Toulouse, Octarès, 1995.

Capítulo 9 - Anexos

Quadro resumo de dados comerciais (consolidado)

		V A L O R F A T U R A M E N T O						
		RESIDENCIAL	COMERCIAL	INDUSTRIAL	PUBLICO	SUBTOTAL	OUTROS	TOTAL GERAL
NORMAl	PRODUTO A	96.678.917,67	16.390.316,29	3.656.009,53	12.629.987,10	129.355.230,59		129.355.230,59
	PRODUTO E	30.786.588,24	6.361.761,12	1.098.594,88	3.715.576,55	41.962.520,79		41.962.520,79
	SUBTOTAL	127.465.505,91	22.752.077,41	4.754.604,41	16.345.563,65	171.317.751,38	12.390.596,88	183.708.348,26

		VOLUME FATURADO(m3)					VOLUME MEDIDO(m3)				
		RESIDENC.	COMERCIAL	INDUSTRIAL	PUBLICO	TOTAL	RESIDENC.	COMERCIAL	INDUSTRIAL	PUBLICO	TOTAL
NOR	Prod.A	2656888,08	4363219,84	847726,92	2617937,76	50485772,60	41227338,32	4002557,04	836691,12	2544249,52	48610836,00
	Prod E	22352974,56	2737515,72	416358,47	1223486,79	26730335,54	21659615,51	2392608,83	387640,27	1179159,39	25619024,00
	Subtotal	65009862,64	7100735,56	1264085,39	3841424,55	77216108,14	62886953,83	6395165,87	1224331,39	3723408,91	74229860,00

IMOVEIS POR PERFIL DO CLIENTE (PRODUTO A E PRODUTO E) - SITUACAO: REAIS

	IMOVEIS SO PRODUTO A	IMOVEIS SO PRODUTO E	PRODUTO A+ E	TOTAL IMOVEIS
NORMAL.....	1.726.001	43.911	1.439.846	3.209.758
ESPECIAL.....	18	29	372	419
CONTRATADO.....	102	11	432	545
TOTAL.....	1.726.121	43.951	1.440.650	3.210.722

IMOVEIS POR PERFIL DO CLIENTE - SUBDIVIDIDO EM CATEGORIAS

		RESIDENC.	(%)HID	COMERCIAL	(%)HI	INDUSTRIA	(%)HID	PUBLIC	(%)HID	MISTO	(%)HID	TOTAL	(%)HID
NORMAL	PRODUTO A	2835.472	99,70	170.839	99,76	19.429	99,78	48.940	99,60	91.170	99,57	3165.850	99,70
	PRODUTO E	1316.808	96.79	96.596	95,40	9.742	95,26	16.785	97,62	43.826	96,62	1483.757	96,70
ESPECIAL	PRODUTO A	136	100,00	99	100,0	38	100,00	35	100,00	82	100,00	390	100,00
	PRODUTO E	137	99,27	109	85,32̂	33	81,81	39	89,74	83	97,59	401	92,76
CONTRATADO	PRODUTO A	133	100,00	178	100,0	142	100,00	52	100,00	29	100,00	534	100,00
	PRODUTO E	126	99;20	167	97,60̄	86	95,34	40	97,50	24	100,00	443	97,74
TOTAL	PRODUTO A	2835.741	99,70	171.116	99,76	19.609	99,78	49.027	99,61	91.281	99,57	3166.774	99,70
	PRODUTO E	1317.071	96,79	96.872	95,40	9.861	95,22	16.864	97,60	43.933	96,63	1484.601	96,70

Capítulo 10 - APÊNDICE

Normas para transcrição[6]

OCORRÊNCIAS	SINAIS	EXEMPLIFICAÇÃO
Hipótese do que se ouviu	(hipótese)	(estou) meio preocupado (com o gravador)
Truncamento (havendo homografia, usa-se acento indicativo da tônica e/ou timbre)	/	e comé/ e reinicia
Entonação enfática	Maiúscula	porque as pessoas reTÊM moeda
Prolongamento de vogal e consoante (como s,r)	:: podendo aumentar para::::ou mais	Ao emprestarem os ... Éh:::... o dinheiro
Silabação	-	por motivo tran-sa-ção
Interrogação	?	e o Banco...Central...certo?
Qualquer pausa	...	são três motivos... ou três razões...que fazem com que se retenha moeda... existe uma... retenção
Comentários descritivos do transcritor	((minúscula))	((tossiu))
Comentários que quebram a seqüência temática da exposição;	-- --	... a demanda de moeda – vamos dar essa notação—

[6] Quadro retirado do livro Análise de Textos Orais, Preti, D. (2001:11-12 apud. Soares, 2005).

desvio temático		demanda de moeda por motivo
Superposição, simultaneidade de vozes		
Indicações de que a fala foi tomada ou interrompida em determinado ponto. Não no seu início, por exemplo	(...)	(...) nós vimos que existem...
Citações literais ou leituras de testos, durante a gravação	“ ”	Pedro Lima... ah escreve na ocasião “O cinema falado em língua estrangeira não precisa de nenhuma baRREIra entre nós”...

OBSERVAÇÕES:

1. Iniciais maísuculas: só para nomes próprios ou para siglas
2. Fáticos: ah, éh, ahn, ehn, uhn, tá
3. Nomes de obras ou nomes comuns estrangeiros grifados
4. Não se indica ponto de exclamação
5. Não se anota cadenciamento de frase
6. Não se utilizam sinais de pausa, típicos da língua escrita, como ponto-e-vírgula, ponto final, dois pontos, vírgula. As reticências marcam qualquer tipo de pausa.