

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE
DEPARTAMENTO DE PÓS GRADUAÇÃO EM ADMINISTRAÇÃO

ADOÇÃO DE TECNOLOGIA:
UM ESTUDO SOBRE O USO DE SOFTWARE LIVRE NAS EMPRESAS

AMIR MOSTAFA SALEH

São Paulo
Janeiro de 2004

AMIR MOSTAFA SALEH

ADOÇÃO DE TECNOLOGIA:
UM ESTUDO SOBRE O USO DE SOFTWARE LIVRE NAS EMPRESAS

Dissertação de mestrado apresentada ao Departamento de Administração da Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo para obtenção do título de Mestre em Administração, sob a orientação do Prof. Dr. Ronaldo Zwicker.

São Paulo
2004

Direitos Autorais Reservados © 2004, Amir Mostafa Saleh

É permitido a qualquer pessoa copiar e redistribuir este documento, no todo ou em parte, para qualquer finalidade, por quaisquer meios existentes ou que venham a ser criados, desde que citada a fonte. É vedada, entretanto, qualquer modificação em seu conteúdo.

Citações a este trabalho devem ser feitas da seguinte forma:

SALEH, Amir Mostafa. Adoção de tecnologia: Um estudo sobre o uso de software livre nas empresas. Dissertação (mestrado) – Faculdade de Economia, Administração e Contabilidade. São Paulo: Universidade de São Paulo, 2004.

Para maiores informações e contato, acesse na Internet: <http://www.ventosolar.com.br/amir>

AGRADECIMENTOS

A meu orientador, Prof. Dr. Ronaldo Zwicker, por sua ajuda, suas críticas e sugestões, que foram fundamentais no desenvolvimento deste trabalho.

A Simone de Castro Paier, por sua ajuda na transcrição das entrevistas, pela revisão do texto, por seu incentivo e por seu apoio, sempre presentes.

À Profa. Dra. Rosa Maria Fischer, com quem tive o privilégio de me iniciar na filosofia da ciência.

Ao Prof. Dr. Roberto Sbragia, por sua ajuda na formulação do conceito desta pesquisa.

Às pessoas entrevistadas, que forneceram as informações necessárias e sem as quais este trabalho não seria possível.

Àqueles que desenvolvem, utilizam e acreditam no software livre.

A todos, obrigado.

RESUMO

O software livre tem sido, nos últimos anos, objeto de atenção por parte dos profissionais e da imprensa especializada em informática. Na bibliografia sobre o assunto, é possível verificar que existem, e estão disponíveis, softwares livres com sofisticação suficiente para atender a grande parte das demandas empresariais. Verifica-se também que podem existir vantagens concretas em sua utilização, principalmente no que tange a redução de custos. No entanto, a participação de mercado desses programas é muito pequena, e está restrita somente a nichos específicos.

Esta dissertação procura identificar, através de metodologia qualitativa e do uso do Modelo de Aceitação de Tecnologia (MAT), os fatores pelos quais o uso de softwares livres é expressivamente menor que o uso de softwares proprietários nas empresas brasileiras.

Para a condução da pesquisa, o universo de empresas foi limitado àquelas que possuíssem de 30 a 160 teclados, e nelas buscou-se conhecer as percepções acerca do software livre, de modo a trazer à tona os fatores que levam à situação de mercado descrita.

Como resultado, foi possível perceber que a baixa participação atual dos softwares livres pode ser explicada como a conjunção de dois fatores principais: necessidade de usufruto de externalidades proporcionadas pela rede dominante; e pelos altos custos de troca de gerados por questões de aprisionamento à tecnologia existente.

ABSTRACT

Free software has been, in the previous years, subject of attention from technology professionals and the specialized press. In bibliography, it's possible to find that there are free softwares good enough to take care of most of enterprise needs, and that there are real advantages in it's use, mainly because of costs reduction. However, the market share of these programs is very small, and is restricted just to specific niches.

This work searches for, using a qualitative methodology and the Technology Acceptance Model (TAM), the factors that explains why the free software use is expressively lower than the proprietary ones in Brazilian companies.

For the conduction of research, the universe of companies was limited to those which owns more than 30 and less than 160 keyboards. In these companies, we searched for the perceptions concerning free software, in order to find the factors that lead to the described situation.

As a result, we saw that the free software low market share situation can be explained as a conjunction of two main factors: the network externalities and the high migration costs.

LISTA DE SIGLAS

ANUI – Associação Nacional dos Usuários de Internet

ASSESPRO - Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet

BSA – Business Software Alliance

BSD – Berkley Software Distribution

CC – Creative Commons

CompTIA – Computing Technology Industry Association

DARPA – Defense Advanced Research Projects Agency

DBA – Database Administrator

FSF – Free Software Foundation

GE – General Eletric

GPL – General Public License

GNU – GNU is Not Unix

HTML – Hypertext Markup Language

MAT – Modelo de Adoção de Tecnologia

MIT – Massachusetts Institute of Technology

OEM – Original Equipment Manufacturer

PHP – PHP Hypertext Preprocessor

SI – Sistema Integrado

SGI – Silicon Graphics Incompany

TI – Tecnologia da Informação

UML – Unified Modeling Language

USL – Unix System Laboratories

SUMÁRIO

1. INTRODUÇÃO.....	10
2. O SOFTWARE LIVRE.....	13
2.1. INTRODUÇÃO.....	13
2.2. HISTÓRICO.....	14
2.2.1. O UNIX.....	15
2.2.2. O GNU.....	18
2.2.3. O CÓDIGO ABERTO.....	22
2.3. LICENÇAS.....	25
2.3.1. A LICENÇA GPL.....	27
2.3.2. LICENÇAS BSD.....	30
2.3.3. ADEQUAÇÃO À LEGISLAÇÃO BRASILEIRA.....	31
2.4. MODELOS DE NEGÓCIOS.....	33
2.4.1. PRODUÇÃO.....	33
2.4.2. COMERCIALIZAÇÃO.....	38
2.4.3. O USO	39
3. A PESQUISA.....	42
3.1. O PROBLEMA DE PESQUISA.....	42
3.2. PREMISSAS.....	44
3.3. PROPOSIÇÕES.....	45
4. O MODELO DE ACEITAÇÃO DE TECNOLOGIA.....	51
4.1. OS CONSTRUTOS.....	55
5. METODOLOGIA.....	59
5.1. ROTEIRO PARA AS ENTREVISTAS.....	62
6. RESULTADOS.....	65
6.1. EMPRESAS USUÁRIAS DE SOFTWARE.....	65
6.1.1. DESCRIÇÃO DOS CONSTRUTOS.....	67
6.2. EMPRESA DE GRANDE ESTRUTURA USUÁRIA DE SOFTWARE LIVRE.....	80
6.3. EMPRESA DE SOFTWARE LIVRE.....	84
6.4. EMPRESAS DE SOFTWARE PROPRIETÁRIO.....	87
7. DISCUSSÃO.....	93
7.1. ANÁLISE DO MODELO.....	93
7.2. INFLUÊNCIAS EXTERNAS.....	98
7.3. PROCESSO DE ADOÇÃO DE TECNOLOGIA.....	106
8. CONSIDERAÇÕES FINAIS.....	111
8.1. ESTUDOS FUTUROS.....	114
8.1.1. MARKETING.....	114
8.1.2. ECONOMIA.....	114

8.1.3. RECURSOS HUMANOS.....	115
REFERÊNCIAS BIBLIOGRÁFICAS.....	116
GLOSSÁRIO E SIGLAS.....	122
ANEXOS.....	124
ANEXO A - SOFTWARES LIVRES DE ACORDO COM AS CATEGORIAS DA PESQUISA ANUAL DA FGV.....	124
ANEXO B – LICENÇAS BSD.....	127
ANEXO C – GNU GENERAL PUBLIC LICENSE.....	130
ANEXO D – LICENÇA PÚBLICA GERAL.....	135
ANEXO F – LICENÇA CC GPL.....	143

1. INTRODUÇÃO

O software¹ livre tem sido, nos últimos anos, objeto de atenção por parte dos profissionais e da imprensa especializada em informática.

Enquanto os softwares proprietários têm um modelo de desenvolvimento fechado, onde apenas uma empresa ou indivíduo têm o controle sobre as funcionalidades, correções e melhoramentos, o software livre utiliza um modelo aberto, onde qualquer pessoa pode ter acesso ao código fonte e exercer o direito de livremente utilizar, redistribuir ou alterar o programa. Durante muitos anos, o modelo proprietário, que trata o programa de computador como uma obra fechada e secreta, foi aceito como a única forma possível de produção de software de qualidade, uma vez que o esforço do programador seria compensado pela venda de licenças de uso. No entanto, o modelo livre subverte essa ordem, fazendo que os produtos sejam compartilhados, de modo que os custos de seu desenvolvimento sejam divididos entre todos os interessados que os utilizam e desenvolvem. Como esses programas são livremente distribuídos, não é utilizado o conceito de venda de licenças, que é substituído por modelos de negócios que focam principalmente a prestação de serviços.

Na bibliografia sobre o assunto, verifica-se que existem softwares livres com sofisticação suficiente para atender a grande parte das demandas empresariais. E verifica-se também que podem existir vantagens concretas na utilização de software livre, principalmente devido à redução de custos e a características técnicas diferenciadas. No entanto, é possível verificar que a presença geral no mercado desses programas é muito pequena, e está restrita a nichos específicos.

Este trabalho tem por base a Pesquisa Anual de Administração de Recursos de Informática da Fundação Getúlio Vargas (Meirelles, 2003), que mostra em termos quantitativos que a participação de mercado de softwares livres é expressivamente inferior às demais alternativas proprietárias. Somente um software livre – o Linux – em uma das categorias (sistemas operacionais de servidores), e ainda assim num percentual muito inferior aos dos líderes. Para entender essa situação, foi formulada a seguinte pergunta de pesquisa:

¹Em todo o trabalho é utilizado o termo “software”, ao invés de “programa de computador”. Foi feita essa opção pelo fato desse termo ter seu uso generalizado em periódicos, em pesquisas científicas e outros tipos de publicações, inclusive de cunho jurídico. Um exemplo é a sentença proferida pelo juiz Álvaro H. T. de Almeida (Almeida, 2002), que utiliza largamente esse termo.

- Que fatores levam os produtos em software livre a ter participação de mercado expressivamente menor que os produtos em software proprietário?

E as seguintes questões assessórias:

- Acreditam os gestores de TI que as soluções de software livre têm condições de suportar em larga escala a infra-estrutura tecnológica da empresa?
- Qual a percepção dos gestores de TI a respeito das características técnicas e da qualidade do software livre?
- Qual a percepção dos gestores de TI a respeito da adoção futura do software livre em larga escala nas empresas?

O objetivo principal do trabalho é compreender a dinâmica de adoção de software livre nas empresas, e explicitar os fatores responsáveis por sua atual participação de mercado, de modo a servir como guia para condução de ações que tenham como objetivo a alteração das condições existentes.

Roteiro

Inicialmente, de forma a situar o leitor quanto às questões pertinentes ao tema, foi descrita a história do software livre utilizado nos dias de hoje, que tem sua raiz no sistema operacional Unix, e deu origem ao projeto GNU. O projeto GNU, com o objetivo inicial de construir um sistema operacional totalmente livre, criou a base para todo um sistema filosófico e legal cuja meta é garantir a liberdade na sociedade da informação. Em seguida é descrito o movimento do Código Aberto, que procurou eliminar as características filosóficas do movimento do software livre, e focá-lo em suas características utilitárias, como a eficiência do modelo de desenvolvimento e a sofisticação dos produtos, principalmente com o objetivo de uso corporativo. Como consequência de sua adoção no meio empresarial, são discutidas também formas de financiar o desenvolvimento de software livre, e manter modelos de negócio nele baseados.

Na seção seguinte, é explicitado o objetivo da pesquisa, suas premissas e as proposições levantadas, que serviram de guia para a manutenção do foco e para auxílio na resposta às perguntas que se pretenderam responder.

A seguir, é detalhado o Modelo de Aceitação de Tecnologia (MAT), desenvolvido por Davis (1980) e estendido por Mathieson (2001). Esse modelo foi desenvolvido especificamente para criar modelos de aceitação de tecnologia da informação, levando em consideração que estímulos externos influenciam as atitudes pessoais, que são determinantes para adoção ou não de uma determinada tecnologia, ou para adoção de alguma em detrimento de outra. Uma premissa fundamental é que ao entender o pensamento do gestor de TI, é possível compreender os motivos da adoção, e conseqüentemente as razões da situação atual de participação de mercado dos softwares livres. Com o MAT, associado às proposições desenvolvidas anteriormente, foram operacionalizados construtos que procuram explicar as questões pertinentes ao software livre nas empresas.

Foi detalhada, então, a metodologia utilizada, explicadas as razões da utilização da metodologia qualitativa, a amostra e os estratos pesquisados. Nesse capítulo é descrito também o roteiro para as entrevistas, obtido através da associação da proposições apresentadas aos construtos do MAT.

Em seguida são apresentados os resultados das entrevistas, com a descrição dos pontos levantados pelas empresas entrevistadas, utilizando o MAT como guia.

Esses resultados são então discutidos, de forma a extrair deles as razões que explicam a baixa participação de mercado dos softwares livres. Nesse ponto, devido à pertinência dos indicadores encontrados, são introduzidos os conceitos de redes dominantes e de aprisionamento. São discutidos também os fatores externos de influência na decisão de adoção de tecnologia previstos no MAT, e é apresentado um modelo de ciclo de vida de adoção de tecnologia. Nesse caso optou-se por utilizar bibliografias adicionais junto à discussão devido à proximidade dos assuntos abordados, para que a discussão teórica não ficasse dissociada de sua aplicação.

Por fim, as considerações finais buscam sumarizar as descobertas encontradas, e sugerir oportunidades de estudos futuros.

2. O SOFTWARE LIVRE

2.1. INTRODUÇÃO

Neste trabalho, é considerado software livre, independentemente sob que termos esteja legalmente licenciado, todo o software sobre o qual sejam garantidas, sem quaisquer restrições, as seguintes liberdades:

- De executar o programa;
- De a qualquer momento modificar o programa para atender às necessidades próprias ou de terceiros;
- De distribuir livremente cópias do programa original;
- De distribuir livremente cópias das versões modificadas.

Para que essas quatro liberdades sejam efetivas, é necessário que os usuários tenham acesso ao código fonte do programa. Código fonte é um texto, numa sintaxe próxima à da linguagem humana, que contém as instruções sobre o que o programa deve executar. Esse código traz consigo o conjunto de tarefas que devem ser desempenhadas, e pode ser facilmente compreendido pelas pessoas. Isso implica que para que seja possível a alteração e a melhoria de um determinado software, o acesso a seu código fonte é um imperativo. No software livre, o código fonte é livremente distribuído. No software proprietário, ele é escondido, e tratado como segredo industrial.

No software proprietário, em geral a única liberdade garantida ao usuário é a de usar o programa, mesmo assim apenas após seu licenciamento, e normalmente com o compromisso de não redistribuí-lo e nem de modificá-lo. Muitos softwares proprietários são grátis, como por exemplo, alguns navegadores de Internet, anti-vírus e editores de textos. No entanto, como não permitem a visualização de seu código fonte, alteração ou a redistribuição, não são livres. Software livre, portanto, não é sinônimo de software grátis. O software só é livre se atender às quatro liberdades acima.

O termo “software livre” será utilizado neste trabalho como referência a qualquer tipo de software que contemple as liberdades descritas. Existem algumas diferenças filosóficas entre

os termos “software livre” e “código aberto”, como será descrito nas próximas seções. Quando estiverem sendo descritas as diferenças entre os dois modelos, isso estará explícito no texto. Caso contrário, o termo “software livre” é utilizado como referência a todas as categorias, englobando inclusive o “código aberto”.

2.2.HISTÓRICO

O software livre existe desde o início do desenvolvimento da informática (Stallman, 1999). Nos anos de 1960 e na primeira metade dos anos 1970, praticamente todo o software era livre, uma vez que ainda não havia um mercado estruturado para a comercialização de licenças. O foco dos fornecedores de tecnologia era o hardware, sendo os sistemas operacionais muitas vezes fornecidos como uma parte integrante do equipamento. Os aplicativos, então, eram quase que totalmente desenvolvidos sob medida para os usuários, muitas vezes através de uma equipe própria. Os softwares dessa época eram específicos para uma determinada arquitetura, e quando havia uma evolução que envolvesse mudança de arquitetura, freqüentemente os sistemas tinham que ser reescritos. Como os programas não tinham valor comercial em si, havia uma grande troca de código entre os participantes da mesma comunidade, envolvendo universidades, instituições de pesquisa e empresas, sem grandes preocupações com direitos autorais ou questões de propriedade. Esse quadro começou a mudar a partir da segunda metade da década de 70, quando a comercialização de licenças começou a tomar corpo e gerou as primeiras motivações para um movimento organizado pelo software livre.

A história do software livre que é utilizado atualmente começa com o sistema operacional Unix. Esse sistema foi um dos primeiros a desenvolver o conceito de portabilidade, onde o código desenvolvido para uma máquina poderia facilmente ser utilizado em outra, com poucas modificações. Isso eliminava a necessidade de reescrita do sistema operacional e dos aplicativos em caso de necessidade de uso dos programas em outras arquiteturas de hardware. Além disso, tinha um projeto considerado elegante, simples e eficiente, que agradava e agrada até hoje aos técnicos. O Unix foi um grande sucesso, e foi baseado em seus conceitos que os modernos softwares livres começaram a ser desenvolvidos, no início da década de 1980. Os sistemas operacionais livres mais populares hoje, como o GNU/Linux e os sistemas da linha BSD, foram baseados no Unix, e praticamente todos os programas livres são desenvolvidos nessas e para essas arquiteturas. A seguir temos o histórico desses sistemas, e forma como influenciaram os programas livres que temos disponíveis.

2.2.1.O UNIX

O sistema operacional Unix² foi criado em 1969 nos laboratórios Bell, então pertencentes à estadunidense American Telephone and Telegraph (AT&T), que à época dominava, através de monopólio, o mercado de telecomunicações nos Estados Unidos. O objetivo do desenvolvimento desse sistema foi a criação de uma estrutura para controle de centrais telefônicas, que estavam se tornando cada vez maiores e mais complexas. O Unix acabou sendo desenvolvido como uma derivação de um sistema operacional anterior chamado Multics, criado através de um consórcio entre os laboratórios Bell, a empresa General Eletric (GE) e o Massachusetts Institute of Technology (MIT). O Unix foi concebido para ser um sistema mais eficiente, e principalmente mais fácil de utilizar que o sistema Multics. A primeira versão do Unix, então chamada Unics4, foi escrito numa linguagem chamada “B”. Nessa mesma época estava sendo desenvolvida uma linguagem sucessora da linguagem B, que tinha por objetivo ser portátil para vários tipos de processadores, alterando-se apenas o compilador. Em 1973, o Unix foi reescrito nessa nova linguagem, chamada de linguagem C, e foi possível o início de sua adaptação para outras arquiteturas de computadores. O Unix até hoje é desenvolvido na linguagem C, que é uma das linguagens de programação mais populares do mundo.

O Unix, por sua facilidade de uso, controle e elegância de projeto, impressionou positivamente os demais pesquisadores dos laboratórios Bell e várias universidades, que obtiveram cópias do sistema para uso próprio. Como à época a AT&T podia atuar somente no mercado de telecomunicações, esta cedeu o sistema às universidades, estimulando a distribuição de seu código fonte, de modo a fomentar inovações. E isso de fato aconteceu, pois diversos pesquisadores, das mais diversas instituições, corrigiram erros e apresentaram melhorias. O Unix, em pouco tempo, tornou-se um padrão para computação dentro das universidades.

Uma das instituições que mais contribuíram para o desenvolvimento do Unix foi a Universidade de Berkeley. Essa universidade começou a utilizar o Unix desde sua criação em 1973, e sempre foi muito ativa no desenvolvimento de novas funcionalidades, melhoramentos e correções do sistema. Em 1977, foi criada a Berkley Software Distribution (BSD), que distribuía o Unix em conjunto com editores de texto e compiladores.

²O histórico do Unix aqui apresentado baseia-se nos relatos de McKusick (1999), no trabalho de Zorzoli (2003) e nos registros disponíveis no sítio do The Open Group, entidade hoje responsável pela marca e pela padronização do Unix, em <<http://www.unix.org>>, com acesso em novembro de 2003.

A última versão do Unix feita pelos laboratórios Bell foi liberada em 1975. Após esse ano o Unix foi controlado pelo Unix System Laboratories (USL), uma subsidiária dos laboratórios Bell, agora com objetivos comerciais. Com isso, o desenvolvimento que vinha sendo feito em conjunto com as universidades foi prejudicado, devido a questões de licenciamento de software. No entanto, as pesquisas que utilizavam o sistema Unix precisavam continuar. A Universidade de Berkeley então, devido a sua vasta experiência no desenvolvimento do Unix, assumiu o papel antes desempenhado pelos laboratórios Bell. Em 1979, fechou um contrato com o departamento de pesquisas das forças armadas estadunidenses³ (DARPA), que queria unificar e padronizar os sistemas operacionais utilizados em seus centros de pesquisa. Com um projeto dessa magnitude, o BSD teve grande projeção, e aumentou consideravelmente seu número de adeptos. McKusick (1999) expõe que a maior força do BSD era seu caráter aberto. Como todas distribuições de programas binários eram sempre acompanhadas do código fonte, o usuário do sistema tornava-se não somente um utilizador passivo, mas sim um colaborador, uma vez que tinha a possibilidade de criar novas funcionalidades e melhorar as que já existiam. O Unix, por outro lado, que acabou ficando com o código fechado, dependia exclusivamente de sua equipe interna e de seus recursos próprios para o desenvolvimento. O resultado foi que no ano de 1983 o BSD 4.3 tinha uma popularidade e uma reputação muito maiores que o Unix System V da USL, em primeiro lugar devido a suas características técnicas avançadas, como a disponibilidade de ferramentas de rede e de um novo sistema de arquivos de alto desempenho, e também por ser amplamente utilizado nas forças armadas.

No entanto, nessa época, para utilizar o BSD era necessário obter, junto à AT&T/USL, a licença de uso para as partes de código por ela desenvolvidas. Como os custos dessa licença a partir de determinado momento tornaram-se proibitivos, iniciou-se um movimento para separar o código desenvolvido pela AT&T, que era proprietário, do código desenvolvido por Berkeley, que era livre. Dessa iniciativa nasceu, em 1989, a distribuição do software chamado Networking Release 1, que embora não fosse um sistema operacional completo, era totalmente livre. Seu objetivo principal era prover aos desenvolvedores de software os componentes de rede desenvolvidos pela BSD de forma independente, sem influências da AT&T. E daí nasceu também a primeira licença de software livre, que de modo geral dizia que qualquer pessoa poderia utilizar aquele código para qualquer finalidade, com apenas duas restrições: que não fossem removidos os avisos de copyright, e que se o programa tivesse documentação, esta deveria obrigatoriamente trazer uma nota dizendo que naquele produto havia código de Berkeley.

³Defense Advanced Research Projects Agency (DARPA)

O lançamento do Networking Release 1 teve aceitação muito boa, e incentivou os desenvolvedores do BSD a promoverem a eliminação de todo o código proprietário, de modo que fosse possível lançar um sistema operacional completo totalmente livre. Foi então criado um esforço de desenvolvimento que envolveu grande parte dos usuários do BSD no sentido de reescrever os códigos de todas as aplicações do sistema sem utilizar código pertencente à AT&T/USL, partindo somente de suas especificações funcionais. As contribuições aos programas começaram a chegar timidamente no início, e depois em volume cada vez maior, chegando ao ponto em que em 1991, somente seis arquivos no sistema todo ainda continham código proprietário. No entanto como o desenvolvimento desses arquivos demandavam um alto grau de complexidade, resolveu-se lançar mesmo assim a nova distribuição, chamada de Networking Release 2. Esse lançamento causou furor na comunidade usuária, e apenas seis meses depois, em 1992, os substitutos livres para os arquivos que faltavam estavam prontos, e foram integrados à distribuição principal. Na mesma ocasião foi criada uma versão do BSD para ser utilizado na popular arquitetura x86, da Intel.

Com o lançamento do Networking Release 2, foi criada uma empresa, chamada BSD Inc. para desenvolver e distribuir uma versão do código mantida e distribuída comercialmente. Isso desencadeou uma reação imediata por parte da USL contra a universidade de Berkeley, alegando que os produtos da BSD Inc. ainda continham código de sua propriedade. Com isso foi iniciado um processo judicial, e durante sua duração, de 1992 a 1994, qualquer sistema operacional que tivesse como base o Networking Release 2 foi proibido de ser distribuído. Ao final, Berkeley foi condenada a remover 3 arquivos, de um total de 1800 que compunham o sistema, que continham código que no entendimento da justiça eram propriedade da USL. Em pouco tempo esses arquivos foram reescritos, e o trabalho com o BSD recomeçado, com o desenvolvimento da BSD Inc. e a criação de outros grupos para distribuir o software sem uma finalidade diretamente comercial.

O BSD hoje se divide basicamente em três projetos: o NetBSD⁴, que tem foco na portabilidade para diferentes plataformas; o FreeBSD⁵, que procura fazer um sistema operacional geral, capaz de ser utilizado por pessoas com menos conhecimentos técnicos; e o OpenBSD⁶, que tem foco em segurança. Esses projetos, apesar dos enfoques diferentes, compartilham muitas idéias e muito do código. São utilizados principalmente em servidores para aplicações de Internet que exigem alta demanda e desempenho.

⁴Sítio na Internet: <<http://www.netbsd.org>>

⁵<<http://www.freebsd.org>>

⁶<<http://www.openbsd.org>>

2.2.2.O GNU

A Free Software Foundation⁷ (FSF) é hoje a principal organização dedicada a produção e à divulgação do software livre. Ela foi iniciada em 1983, pelo estadunidense Richard Stallman, com o objetivo de promover na comunidade de informática o espírito cooperativo que prevalecia em seus primórdios, onde as informações, códigos e métodos de trabalhos eram livremente compartilhados (Stallman, 2001). Para a FSF, os obstáculos impostos pela indústria de software proprietário são prejudiciais ao desenvolvimento do conhecimento, e por consequência, à sociedade como um todo.

Em 1971, a equipe da qual Stallman (1999) fazia parte, no Massachusetts Institute of Technology (MIT), trabalhava exclusivamente com software livre. Como a indústria do licenciamento de software ainda não era desenvolvida, as universidades e empresas não viam problema na distribuição e no compartilhamento de código. Como para as empresas de informática nessa época o grande negócio era a venda de hardware, o software era considerado sua parte integrante. Com a evolução da indústria de informática, e a dissociação entre o software e hardware, durante os anos 70 houve um movimento progressivo de fechamento dos códigos, sendo que no início dos anos 80, salvo algumas exceções, praticamente todo o software era fechado e proprietário.

Segundo Stallman (1999), a idéia de software proprietário implica numa organização social onde não é permitido às pessoas compartilhar conhecimento, não é permitido ajudar ao próximo e incita à competição ao invés da colaboração: acredita que todo software deveria ser livre. Um de seus principais argumentos é que o custo marginal de distribuição e reprodução de software é próximo de zero. Considera que uma vez pagos os custos de desenvolvimento, restringir o acesso ao software traz muito mais malefícios que benefícios. Uma vez que o potencial usuário resolve pagar pela licença, está acontecendo uma transferência de riquezas que pode ser benéfica para ambas as partes. No entanto, quando esse usuário resolve não utilizar o software devido à necessidade de pagamento, está havendo prejuízo a uma das partes sem que haja benefício de ninguém. Considera que o prejuízo à sociedade pela restrição ao uso do programa é muito maior que os potenciais ganhos individuais do produtor, uma vez que se está impondo restrições a um bem cujo custo marginal de produção é zero, e que os desenvolvedores de software proprietário impedem que a cooperação entre as pessoas ocorra, através da criminalização das cópias. Como os programas proprietários são fechados,

⁷A descrição do trabalho da Free Software Foundation e do projeto GNU baseiam-se nos escritos de Stallman (1999; 2001), nas informações de seu sítio na Internet: <<http://www.fsf.org>>.

é negado às pessoas melhorar suas funcionalidades de forma a atender a necessidades específicas. Sustenta ainda que é prejudicado o desenvolvimento de software de maneira geral, uma vez que a cada novo projeto proprietário, é necessário redesenvolver várias tecnologias que eventualmente já tenham sido criadas, e estão inacessíveis por serem fechadas. Por fim, a idéia que permeia toda os seus argumentos é que fechar o software é negar o acesso à informação e ao conhecimento, elementos sem os quais é impossível construir uma sociedade justa e democrática.

O motivador inicial para o projeto de construção da FSF aconteceu em 1981, quando o laboratório de inteligência artificial do MIT teve seu principal computador substituído. O computador anterior, cujo sistema operacional e aplicativos foram desenvolvidos pela própria equipe da universidade, era operado somente com software livre. No início dos anos 80, houve um esvaziamento desse laboratório, fazendo com que não houvessem pessoas nem recursos suficientes para criar e manter um novo sistema operacional para o novo computador. Com isso, a direção do laboratório decidiu pela utilização do sistema produzido pelo próprio fabricante, que era fechado e proprietário. Stallman, então, para não ferir seus princípios, resolveu que não deveria assinar os acordos de não divulgação impostos pelo fornecedor do novo hardware, e que não deveria se submeter a trabalhar com software proprietário. Por isso, deixou de trabalhar no MIT, e passou a coordenar a FSF, com o objetivo de “criar software suficiente para poder utilizar um computador sem a necessidade de qualquer software proprietário” (Stallman, 1999). Seus objetivos, portanto, foram muito mais filosóficos e ideológicos do que técnicos. Stallman luta contra o modelo de software proprietário, e desenvolve software livre não porque este seja necessariamente melhor ou mais eficiente, mas sim porque considera que essa opção é a única socialmente justa.

No Manifesto GNU (Stallman, 2002 p. 32), escrito à época do lançamento do projeto, Stallman colocou o seguinte:

“Eu acredito que a regra de ouro exige que, se eu gosto de um programa, eu devo compartilhá-lo com outras pessoas que gostam dele. Vendedores de software querem dividir os usuários e conquistá-los, fazendo com que cada usuário concorde em não compartilhar com os outros. Eu me recuso a quebrar a solidariedade com os outros usuários deste modo. Eu não posso, com a consciência limpa, assinar um termo de compromisso de não-divulgação de informações ou um contrato de licença de software. [...] Muitos programadores estão descontentes quanto à comercialização de software. Ela pode trazer-lhes

dinheiro, mas ela requer que eles se considerem em conflito com outros programadores de maneira geral em vez de considerá-los como camaradas. O ato fundamental da amizade entre programadores é o compartilhamento de programas; acordos comerciais usados hoje em dia tipicamente proíbem programadores de se tratarem uns aos outros como amigos. [...] Portanto, de modo que eu possa continuar a usar computadores sem desonra, eu decidi produzir uma quantidade de software suficiente para que eu possa continuar sem nenhum software que não seja livre.”

O esforço inicial de Stallman, mesmo antes de deixar definitivamente o MIT, era o de criar um sistema operacional⁸ completo, cujo nome seria GNU⁹. Esse sistema tinha a intenção de ser um clone do Unix, mas dessa vez totalmente livre de código proprietário. O sistema seguia todas as especificações e parâmetros do Unix, mas os programas foram totalmente reescritos, sem utilizar qualquer código previamente disponível que pudesse ser objeto de disputas relativas a direitos autorais. A compatibilidade do GNU com esse sistema seria conveniente, e devido a sua ampla aceitação, as aplicações desenvolvidas para Unix seriam automaticamente compatíveis com o GNU, e os usuários de Unix poderiam facilmente trocar seu sistema pelo GNU assim que este ficasse pronto. No início, Stallman procurou colaboradores entre os profissionais da área, pedindo que utilizassem as ferramentas que ele próprio já tinha desenvolvido, e que se juntassem ao projeto doando equipamentos, dinheiro e reescrevendo os programas do Unix sob forma de software livre. Para os usuários finais, Stallman passou a divulgar a idéia de que como os códigos estariam disponíveis para todos, seriam livres para buscar suporte e personalizações de acordo com sua necessidade, sem precisar ficar à mercê de um determinado fornecedor que detivesse os direitos exclusivos do programa. E também conclamava as empresas e profissionais para que ganhassem dinheiro com software livre, através de venda de mídias com os programas, suporte técnico, e desenvolvimento de programas para atender a necessidades específicas de seus clientes.

O primeiro grande sucesso de Stallman, e por conseguinte da FSF, foi o editor de textos Emacs, muito admirado pelas facilidades que colocava à disposição dos usuários, principalmente para tarefas de desenvolvimento de software. Esse editor era distribuído de duas formas: através do sítio do MIT na Internet (na época de uso ainda muito restrito) ou através da venda de fitas que continham o programa gravado. A venda das fitas iniciou um dos

⁸ Sistema operacional é o programa básico que controla o hardware do computador e permite a execução dos demais aplicativos. Alguns exemplos: Windows, GNU/Linux, MacOS, DOS.

⁹ GNU - Gnu's Not Unix. Esse nome foi utilizado para deixar explícito que o novo sistema operacional era como o Unix (Unix-like), mas não continha código original Unix. A sigla é recursiva.

primeiros negócios com software livre. O sucesso do Emacs deu visibilidade ao projeto GNU, e estimulou mais pessoas a ajudarem a desenvolvê-lo.

Um das grandes contribuições do projeto GNU, além do software em si, foi o desenvolvimento da General Public License (GPL)¹⁰, a licença de software mais utilizada em projetos de software livre, que dá amparo legal e formaliza a ideologia que permeia o movimento.

Em 1991, o sistema operacional GNU estava praticamente pronto. Ainda faltava, no entanto, um componente de suma importância: o núcleo. O núcleo é a parte central do sistema operacional, responsável principalmente pela execução dos programas, escalonamento de tarefas e acesso aos recursos de hardware. É responsável pela integração entre o hardware e os aplicativos do sistema, que efetivamente executam o trabalho desejado pelo usuário. O núcleo original do projeto GNU, chamava-se (e até hoje chama-se) Hurd. Seu projeto utilizava uma abordagem muito moderna e ousada conhecida como microkernel, que revelou-se de uma complexidade inesperada. Tanto que até o presente momento, no ano de 2003, ainda não tem qualidade suficiente para ser utilizado em ambientes que não sejam de testes e desenvolvimento. No entanto, nessa ocasião, um finlandês chamado Linus Torvalds desenvolveu uma versão inicial de um núcleo de sistema operacional compatível com o Unix, específico para computadores com o processador Intel 386. Esse núcleo, no entanto, utilizava uma abordagem que à época era considerada ultrapassada, chamada de núcleo monolítico. O núcleo monolítico, apesar de ser considerado inferior, permitia o desenvolvimento de forma muito mais simples e eficiente. Tanto que uma vez lançado publicamente, rapidamente foram adicionadas, com a ajuda da comunidade, as funcionalidades necessárias a um núcleo de uso real. Torvalds chamou seu novo núcleo de Linux, e tornou-o disponível na Internet sob a própria licença GNU GPL, da FSF.

Como dito, o sistema GNU estava, nesse período, quase que totalmente disponível, faltando somente o núcleo. E como Torvalds tinha um projeto de núcleo já iniciado, em pouco tempo foi promovida a integração entre o GNU e o Linux. Isso permitiria a criação do primeiro sistema operacional totalmente livre para a arquitetura Intel x86, uma vez que o BSD para esse processador surgiria somente em 1992, e seria imediatamente obscurecido até o ano de 1994, devido à retirada de mercado forçada pelo processo judicial movido pelos detentores dos direitos autorais do Unix.

¹⁰Uma descrição completa da GPL pode ser encontrada à página 27.

Como o GNU/Linux surgiu como um sistema completo e imediatamente disponível, as atenções de desenvolvedores do mundo inteiro voltaram-se para ele, e houve um explosão de interessados em seu uso e em seu desenvolvimento. Em poucos anos atingiu funcionalidades e estabilidades comparáveis a sistemas operacionais proprietários já consolidados (Bretthauer, 2002). Quando o BSD para a arquitetura Intel x86 voltou a estar disponível após o processo judicial, em 1994, o GNU/Linux já tinha altíssima popularidade, que conservou ao longo do tempo até os dias de hoje.

Um dos principais efeitos colaterais, para a FSF, da dificuldade no desenvolvimento do Hurd, foi a popularização do sistema GNU com o núcleo Linux. Isso fez com que o sistema operacional completo passasse a ser muitas vezes referenciado somente como “Linux”, eliminando a referência ao projeto GNU. Isso resultou em confusões por parte dos usuários e principalmente por parte da imprensa, que muitas vezes pensam que Linus Torvalds criou o sistema operacional completo, e mais, que criou o conceito de software livre. Richard Stallman, então, insiste em seus escritos e em suas palestras para que o sistema completo seja denominado GNU/Linux, numa forma de dar créditos à FSF. Na verdade, os sistemas operacionais que utilizam o núcleo Linux¹¹ atualmente contém muito mais que apenas o sistema GNU e o núcleo Linux. Dentro deles há software para as mais diversas finalidades não pertencentes a esses dois projetos, produzidos pelos mais diversos grupos e entidades, com a participação de pessoas de praticamente todos os países do mundo.

A FSF, com o passar dos anos, sofreu outros revezes, como a criação de projetos paralelos de produtos antes desenvolvidos somente por ela, e principalmente pelo lançamento do movimento pelo Código Aberto, que tem o objetivo de eliminar a motivação ideológica, e encara o software livre como eficiente somente por suas razões utilitárias. Hoje a FSF ainda fomenta o desenvolvimento de muito software, mas tem um papel de destaque principalmente na difusão internacional do conceito e da filosofia do software livre.

2.2.3.O CÓDIGO ABERTO

O movimento do Código Aberto foi iniciado em 1997, por um grupo de pessoas que trabalhavam com software livre, mas sentiam-se incomodadas com o caráter ideológico impingido pela FSF e sua licença GNU GPL. O movimento do Código Aberto quis, então, retirar do desenvolvimento de software livre qualquer componente social ou ideologizante.

¹¹Uma lista de sistemas operacionais que utilizam a base GNU e o núcleo Linux pode ser encontrada à página 124.

Para esse movimento, o software deve ser aberto não por questões de liberdade, mas sim porque consideram o modelo aberto de desenvolvimento mais eficiente tanto técnica quanto economicamente. A idéia central é que quando os programadores podem ler, modificar e distribuir o código, naturalmente surgem melhorias, adaptações e correções, que têm como consequência a evolução do programa.

Ainda no ano de 1997, o estadunidense Bruce Perens publicou a Definição do Código Aberto (Perens, 1999). Segundo essa definição, qualquer licença de software pode ser considerada aberta se adequar-se aos seguintes termos:

1. O programa deve poder ser redistribuído sem a exigência de pagamentos de qualquer espécie.
2. Todos os programas devem incluir o código fonte. Caso não seja distribuído com o código fonte, deve haver indicações claras de como obtê-lo.
3. A licença deve permitir modificações e trabalhos derivados, que devem poder ser distribuídos nos mesmos termos da licença original do software.
4. A licença pode restringir a distribuição de trabalhos cujo original seja diretamente modificado, desde que garanta a distribuição de arquivos de correção que possam ser aplicados no momento da compilação. Isso visa permitir que o desenvolvedor do programa original tenha a opção de exigir que seu código original não seja modificado. Nesse caso esse arquivos de correção podem ser distribuídos, e alterar o programa original somente no momento da tradução para o código de máquina entendido pelo computador. E o autor pode também exigir que o trabalho derivado seja distribuído sob uma nova nomenclatura ou sob um diferente número de versão.
5. A licença não pode discriminar nenhum tipo de pessoa ou grupo.
6. A licença não pode restringir qualquer tipo de uso do programa.
7. Os direitos sobre o programa devem ser definidos somente através da licença, não sendo permitido nenhum mecanismo adicional. Isso visa evitar que o programa tenha uma licença livre, mas que o usuário seja obrigado, por exemplo, a concordar com um termo de não divulgação.

8. A licença não pode ser condicional à presença do programa numa determinada distribuição ou conjunto de outros programas. Se o programa licenciado for retirado dessa distribuição, a licença deve permanecer válida mesmo que ele esteja atuando separadamente.

9. A licença não deve exigir que outro programa distribuído em conjunto siga a mesma licença. Um programa de código aberto pode, portanto, ser distribuído em conjunto com um proprietário sem que este tenha que se tornar livre.

10. A licença deve ser tecnologicamente neutra, não devendo exigir o uso de uma tecnologia específica ou tipo de interface.

Não é feita, então, na definição de Código Aberto, qualquer referência a aspectos políticos, sociais e de liberdade, ao contrário das licenças de software livre da FSF, onde o aspecto de liberdade é fundamental. Além da questão política, existe uma outra grande diferença: num software licenciado pela GNU GPL da FSF, todo o trabalho derivado necessariamente também deve ser software livre. No entanto, a definição do Código Aberto dá abertura para que sejam criadas licenças que permitam que um software aberto seja fechado e passe a ser distribuído como software proprietário, deixando portanto de ser livre. Devido a essa diferença de posições, chegaram a haver embates públicos entre as duas correntes, notadamente tendo a FSF e a Open Source Initiative¹² como protagonistas (Stallman, 2002 p. 55). Stallman já reiterou em várias oportunidades que o objetivo do Código Aberto é produzir software poderoso e de alta qualidade, e que isso é um objetivo louvável, mas não o mais importante: o mais importante é a liberdade.

E a questão ficou mais complicada a partir do momento em que grande parte da comunidade usuária passou a não mais utilizar o termo “software livre”, mas sim “código aberto”. Isso tinha o objetivo de tornar os termos mais palatáveis a executivos e usuários comerciais que não se importam com questões sociais e de liberdade, e também para evitar a confusão que sempre existiu entre “software livre” e “software grátis”, uma vez que no inglês os dois termos são designados pela mesma palavra: “free”. Linus Torvalds, por exemplo, em livro onde conta a história do desenvolvimento do Linux, utiliza sempre o termo “código aberto” e nunca “software livre” (Torvalds, 2001). Mas a realidade é que devido a sua adoção nas empresas o Código Aberto tem tido cada vez mais adeptos, enfraquecendo a posição da FSF como interlocutora única do movimento mundial do software livre.

¹²Sítio na Internet: <<http://www.opensource.org>>

No anexo, iniciado à página 127, está disponível uma lista de licenças compatíveis com a definição de Código Aberto. Entre elas estão, inclusive a GPL, e a licença do BSD.

2.3.LICENÇAS

A questão da propriedade intelectual¹³ é marcada pela divisão de interesses entre o produtor da obra, que a pode considera-la como um bem qualquer passível de exploração e lucro, e a sociedade, interessada em seu usufruto. Isso fica claro nas posições filosóficas do movimento do software livre da FSF, para quem o benefício da sociedade deve ser sempre prioritário ao ganho individual. As leis de direitos autorais foram criadas com o objetivo de permitir aos autores uma forma de explorar comercialmente suas criações, recompensando-os por seu esforço e estimulando outras inovações.

Existem basicamente dois tipos de propriedade intelectual: o direito autoral e o direito de patentes. O direito de patentes corresponde à concessão, por parte do Estado, de um monopólio de exploração, por um período de tempo determinado, de uma invenção ou de um novo processo de produção. A contrapartida pela obtenção do monopólio é abertura, por parte do inventor, das especificações e do funcionamento do objeto da patente. Todos aqueles que desejem utilizar esse invento após a concessão da patente devem obrigatoriamente obter a permissão do inventor, e submeter-se a suas restrições de licenciamento, que podem incluir a necessidade de pagamentos ou outros impedimentos de atuação. Após o período de concessão da patente, suas especificações se tornam públicas, e é possível para qualquer pessoa utilizá-las sem a necessidade de qualquer tipo de autorização ou desembolso.

No Brasil, não é possível o registro de patentes de software. Seguindo o exemplo de vários países do mundo, os programas de computador são entendidos como passíveis de proteção pelas leis de direitos autorais, da mesma forma que livros, músicas ou trabalhos de artes plásticas. Os direitos que o autor de um programa de computador tem sobre sua obra se fazem valer a partir do momento de sua publicação, e são válidos pelo período de 50 anos, após os quais passam a ser de domínio público. Durante esse período, o autor, seus sucessores legais, ou um terceiro a quem forem cedidos os direitos, têm poder de decidir se será permitido a outros o acesso à obra original e a suas cópias, e sob que condições.

¹³A questão dos direitos autorais é aqui descrito com base nos trabalhos de Blum e Abrusio (2002), Cerdeira (2003) e nas leis nº 9.610, que trata dos direitos autorais, e nº 9.609, que trata dos direitos autorais de programas de computador.

Os direitos de autor podem ser divididos em patrimoniais (copyright) e morais. Os direitos patrimoniais tratam de sua negociação da obra, e podem envolver questões de licenciamento a terceiros, comercialização, aluguel, transferência de propriedade, acesso ao código fonte e estabelecimento de condições para seu uso e alteração. Esses direitos são, portanto, negociáveis. Os direitos morais tratam do direito do autor ter sempre seu nome ligado à obra, e ao mesmo tempo impedir que seu nome seja denegrido por outras pessoas que façam mal uso dela. Os direitos morais do autor permanecem junto à obra independentemente a quem ela pertença e por quem seja utilizada. Por isso, os direitos morais são, segundo a lei nº 9606 de 1998, “inalienáveis e irrenunciáveis”. Uma pessoa jurídica pode ser detentora dos direitos patrimoniais de um software, mas nunca de seus direitos morais, porque “uma pessoa jurídica não é capaz de criar nada; não tem talento, não tem espírito, não tem imaginação” (Almeida, 2003).

E além dos direitos autorais e das patentes, existe ainda o “domínio público”. Uma obra sob domínio público é aquela onde o autor renunciou a seus direitos patrimoniais. Não pode tomar qualquer decisão, ou impor qualquer restrição quanto a seu uso, produção de cópias e distribuição.

Nos Estados Unidos, onde iniciou-se o movimento do software livre, os conceitos são basicamente os mesmos, com algumas diferenças: é permitido patentear software, algoritmos e idéias; e os direitos morais do autor podem ser exercidos separadamente dos direitos patrimoniais. No caso específico do software, é possível fazer a opção, de acordo com o que for mais conveniente: os direitos sobre o software podem ser regidos por patentes, onde o produtor deve abrir suas especificações, e em troca recebe direitos sob forma de monopólio por um tempo especificado; ou então o software é registrado e regido pelo direito do autor.

A intenção da FSF, quando começou a produzir seus programas, era proteger legalmente a comunidade de software livre, de forma que os códigos desenvolvidos não fossem apropriados por desenvolvedores de software proprietário (Stallman, 2002 p. 89). Para esse propósito, simplesmente colocar o software sob domínio público não era aceitável, porque isso implicaria que qualquer pessoa pudesse alterá-lo, redistribuí-lo e eventualmente fechá-lo como sendo um trabalho de própria autoria. A solução para esse problema foi utilizar a própria estrutura legal de proteção aos direitos do autor para garantir a liberdade dos usuários de computadores. A idéia era que, através das leis de copyright, o autor tomasse posse da obra, garantindo assim sua autoria e seus direitos patrimoniais. Em seguida esse autor imporia condições para seu licenciamento, de forma que ao invés de restringir o acesso do público e

controlar as cópias e distribuições, daria ao licenciado a liberdade de utilizar, distribuir e modificar o programa original da forma que bem entendesse. E uma restrição importante foi adicionada: toda obra derivada da obra original teria necessariamente que fornecer ao licenciado exatamente os mesmos direitos. Ou seja, não seria permitido, a partir de um software produzido sob a licença livre da FSF, a produção de um software proprietário. Uma vez livre, o software deveria para sempre permanecer livre. Esse tipo de licenciamento garante a liberdade das obras derivadas, e vai frontalmente contra à forma usual de publicação restritiva utilizada no mercado. Por isso, foi chamado pela FSF de copyleft, num trocadilho com a palavra copyright. É muito comum nos escritos sobre licenciamento da FSF, a menção de uma determinada licença de software ser ou não copyleft. Se for copyleft, obriga os trabalhos derivados a também serem softwares livres. Se não for, permite que o derivado se torne proprietário.

A seguir está a descrição das principais licenças utilizadas para software livre.

2.3.1.A LICENÇA GPL

Em 1989, com o conceito de “copyleft” já formado, foi criada, pela FSF, a primeira licença de software livre, chamada GNU General Public License¹⁴ (GPL). A GPL foi criada com o objetivo de servir como um conjunto de restrições exigidas pelo autor do software a seus potenciais usuários. Suas características básicas são garantir a liberdade de uso, de alteração e distribuição, além de garantir que qualquer trabalho derivado também seja licenciado sob as mesmas condições.

A GPL exige, para qualquer pessoa ou instituição, sem qualquer tipo de discriminação quanto a finalidade, grupo social ou nacionalidade, os seguintes pontos:

1. A liberdade de executar o programa.
2. A liberdade de estudar seu funcionamento, e a qualquer momento modificá-lo para atender às necessidades próprias ou de terceiros.
3. A liberdade de distribuir livremente cópias do programa original.
4. A liberdade de distribuir cópias das versões modificadas, de forma que toda a comunidade possa se beneficiar das melhorias introduzidas.

¹⁴A íntegra da licença GNU GPL pode ser encontrada no Anexo, à página 130.

A GPL enfatiza, em sua introdução (anexo C, à pág. 130), seu caráter ideológico. Expõe que as licenças de muitos programas são feitas com o intuito de restringir a liberdade das pessoas, e que a GPL foi criada justamente para garanti-la, como um instrumento legal para proteger direitos, e fazer com que outros não tenham a opção de removê-los. Para tanto, a GPL tem duas restrições fundamentais:

1. Todos os trabalhos derivados devem necessariamente estar licenciados pela GPL. É permitida, entretanto, a não distribuição de alterações realizadas no programa, desde que mantidas em uso estritamente privado. O trabalho derivado, caso seja distribuído, necessariamente deverá seguir a GPL.

O direito de não distribuição visa garantir a privacidade do usuário, e seu direito de não ter que revelar alterações que digam respeito somente a si a seus métodos de trabalho.

2. Não é permitida a alteração das condições da licença.

E as seguintes restrições adicionais:

3. Todos os programas devem vir acompanhados do código fonte, ou de instruções claras de como obtê-lo.

4. Aqueles que recebem o software devem necessariamente ser informados dos direitos que possuem, e que são garantidos pela GPL.

5. Uma vez que não há cobrança pelo licenciamento do programa, não há qualquer garantia de funcionamento ou de que os programas licenciados funcionem ou sirvam para uma finalidade específica.

6. Ao modificar um programa, não é permitido retirar os créditos referentes ao trabalho de outros autores.

7. Dentro do código fonte deve estar claro que mudanças foram realizadas por qual autor, para que problemas introduzidos por modificações não interfiram na reputação dos autores originais.

8. Caso alguma situação legal impeça o programa de ser distribuído livremente, este deverá imediatamente deixar de ser distribuído. Essa possibilidade é bastante concreta nos EUA, onde são permitidas patentes de software, e os processos para concessão dessas patentes

correm de forma secreta. Se por acaso for concedida uma patente que obrigue qualquer tipo de restrição ou o pagamento de licenças a seu dono, a distribuição deverá cessar até que o software seja modificado para não utilizar mais os recursos descritos na patente. Caso essa restrição ou essa patente não seja válida a todos os países ou regiões, pode ser incluída na licença uma limitação geográfica para que o programa seja distribuído somente nessas localidades.

A GPL também incentiva o uso do software livre como meio para prestação de serviços, seja pela venda de mídia contendo software, pelo oferecimento de garantia, de suporte técnico ou desenvolvimento de novas funcionalidades. Qualquer forma de uso empresarial dos recursos em software livre é fortemente encorajada, desde que não retire qualquer uma das liberdades anteriormente citadas. O foco da GPL é, então, a liberdade, e não a gratuidade.

Além da GPL, a FSF mantém mais duas licenças: a Lesser General Public Licence (LGPL) e a Free Documentation Licence (FDL) (Stallman, 2002 p. 23).

A LGPL foi criada especificamente para uso com componentes chamados bibliotecas. As bibliotecas são coleções de funções usadas para produzir outros programas. Elas são utilizadas para que não seja necessário reescrever código para executar tarefas comuns. Quando quer utilizar essas funções, o programador insere as bibliotecas dentro de seu programa, fazendo com que o produto final seja a combinação de seu próprio código com o código da biblioteca utilizada. Ou seja, o trabalho derivado é necessariamente uma derivação da biblioteca original. Portanto, se essa biblioteca estiver licenciada pela GPL, obrigatoriamente o programa resultante deve também ser GPL. Por motivos de popularização de uso de bibliotecas livres, e para não tornar inviável o surgimento de programas proprietários para o sistema operacional GNU, a FSF resolveu criar a LGPL. Essa licença tem basicamente as mesmas condições da GPL, no entanto, não obriga que o trabalho derivado continue livre. Ou seja, não é uma licença do tipo “copyleft”. A FSF desencoraja fortemente a utilização dessa licença, recomendando-a somente para casos muito específicos, onde não haja outra alternativa. No entanto, ela vem sendo utilizada por vários projetos onde há forte sinergia entre empresas de software proprietário e grupos de desenvolvimento livre. Um exemplo é o Openoffice.org, o mais avançado conjunto de aplicativos de escritório existente para os sistemas operacionais GNU/Linux. Ele é desenvolvido por um grupo independente, mas com grande participação da empresa Sun Microsystems. O trabalho do OpenOffice.org serve de base para um outro produto da Sun, o StarOffice, cujo licenciamento é proprietário.

A última licença da FSF, a Free Documentation License (FDL), refere-se somente à documentação. Conforme seu prefácio:

“O propósito desta licença é deixar um manual, livro-texto ou outro documento escrito livre: assegurar a qualquer um a efetiva liberdade de copiar ou redistribuí-lo, com ou sem modificações, comercialmente ou não. Secundariamente, esta licença mantém para o autor e editor uma forma de ter crédito por seu trabalho, sem ser considerado responsável pelas modificações feitas por terceiros. Esta licença é um tipo de 'copyleft', o que significa que derivações do documento precisam ser livres no mesmo sentido”.

A FDL é, portanto, uma versão da GPL específica para textos escritos. Ela garante que um documento seja livre, e continue livre após modificações. Tem controles também sobre partes que autor considera fundamentais e devem ser invariantes, e elabora uma série de restrições para organizar as modificações de acordo com as orientações do autor original. A FDL foi escrita originalmente para os manuais que acompanhassem o sistema operacional GNU. No entanto, atualmente vários outros tipos de publicações, principalmente de cunho técnico sobre software livre utilizam essa licença. Mas assim como outros aspectos relacionados à FSF, a FDL não é livre de polêmicas. Uma das mais famosas diz respeito ao grupo de desenvolvimento do sistema operacional livre Debian GNU/Linux¹⁵, que tem reservas quanto à FDL devido à possibilidade de existência de seções, dentro do documento, que não podem ser modificadas.

2.3.2.LICENÇAS BSD

Ao contrário da GPL, que tem um forte componente ideológico e impõe uma série de restrições ao licenciado, a licença BSD¹⁶ tem uma estrutura enxuta e simples. Ela diz que a redistribuição e o uso do programa na forma binária e na forma de código fonte, com ou sem modificações, é permitida. As únicas restrições são as seguintes: na redistribuição do código fonte deve ser mantido o aviso de *copyright* original para identificar o autor; as distribuições binárias, sem o código fonte, devem colocar o *copyright* na documentação; e que o nome do autor não pode ser utilizado para promover o endossar versões modificadas do programa.

¹⁵Veja em <<http://www.debian.org/doc/manuals/ddp-policy/ch-common.en.html>>. Acessado em novembro de 2003.

¹⁶A íntegra da licença BSD pode ser encontrada no Anexo, à página 127.

Adicionalmente, ressalta também que não há garantias de que o programa funcionará para qualquer finalidade e que seus autores não têm responsabilidades sobre seu uso.

A licença BSD não tem qualquer componente político ou social, enquadrando-se muito mais na filosofia do movimento do Código Aberto do que no movimento de software livre promovido pela FSF. Como não há restrições, o software pode ser utilizado para qualquer finalidade, inclusive para incorporação em produtos proprietários. Alguns exemplos conhecidos são o programa Xfree86, que é utilizado como base da interface gráfica de vários sistemas operacionais proprietários derivados do Unix; o servidor de bancos de dados PostgreSQL, que tem uma versão proprietária com recursos exclusivos; e o sistema de rede TCP/IP do sistema operacional BSD, que foi incorporado pela Microsoft em seu sistema operacional Windows NT.

Os adeptos dessa licença sustentam que essa é licença “realmente” livre¹⁷, pois não impõe qualquer tipo de restrição ao licenciado. A FSF por outro lado, a considera prejudicial, por não ser preocupar com a questão ideológica e por permitir trabalhos derivados proprietários, o que não incentivaria a produção de mais software livre (Stallman, 2002). No entanto, considera a licença BSD compatível com a GNU GPL, e utiliza programas como o próprio Xfree86 e o servidor de páginas Apache, entre outros. Na ótica da FSF, é preferível resignar-se ao uso dessa licença que iniciar projetos idênticos, com a mesma funcionalidade, somente para licenciá-los sob a GNU GPL.

Atualmente existem várias licenças que têm como base a licença BSD e os conceitos do movimento do código aberto¹⁸. Elas são muito utilizadas por empresas que têm modelos de desenvolvimento híbridos, onde produtos de Código Aberto colaboram para o desenvolvimento de produtos proprietários.

2.3.3. ADEQUAÇÃO À LEGISLAÇÃO BRASILEIRA

O ordenamento jurídico a respeito dos programas de computador no Brasil é muito parecido com o estadunidense, sendo que a única diferença fundamental é a impossibilidade do patenteamento de software. As leis que regulam os programas são a nº 9610/98, que trata dos direitos autorais de forma genérica, e a nº 9609/98, que trata especificamente de software. Os conceitos que se aplicam à GNU GPL e às demais licenças de software livre que utilizam-se das restrições das leis de direitos autorais para prover direitos e liberdades são válidos. E

¹⁷Veja em <<http://www.openbsd.org/policy.html>>. Acessado em novembro de 2003.

¹⁸Uma lista dessas licenças pode ser encontrada no Anexo, à página 127.

segundo tratados internacionais dos quais o Brasil é signatário, a GNU GPL e a licença BSD podem ser normalmente utilizadas.

Entretanto, há pelo menos duas iniciativas de construção de licenças específicas para o Brasil, que contemplem as especificidades do país, e permitam uma força maior em sua aplicação.

A primeira iniciativa nesse sentido partiu do advogado Pablo de Camargo Cerdeira, que criou uma licença chamada Licença Pública Geral (LPG)¹⁹, que pretende ser uma adaptação da licença GNU GPL às especificidades da lei brasileira. Foram feitas adaptações para dirimir eventuais problemas de licenciamento frente à lei de direitos autorais, à lei de programas de computador e ao código de defesa do consumidor. No sítio oficial do projeto²⁰ há a informação de que pelo menos três projetos famosos de software livre brasileiros estão utilizando essa licença: o Direto, sistema de correio eletrônico e agenda utilizado como meio oficial de comunicação no Governo do Rio Grande do Sul; o Labeo, sistema de notas e faltas utilizado pela Faculdade de Direito da Universidade de São Paulo; e o Kurumin, sistema operacional enxuto baseado no GNU/Linux que é executado diretamente a partir de um Mini CD, sem a necessidade de instalação.

A segunda iniciativa partiu do governo federal, que em dezembro de 2003 adotou as licenças Creative Commons GPL²¹ e LGPL como oficiais para o todo software livre produzido pelo governo. O Creative Commons (CC)²² é uma organização nascida nos Estados Unidos, que tem como objetivo levar os conceitos utilizados no licenciamento do software livre para outras áreas do conhecimento, como literatura, música e artes plásticas, e agir pela universalização dos bens culturais. O Comitê para Implementação de Software Livre²³ no governo criou, então, uma licença que reúne conceitos da CC e FSF, com características adaptadas às necessidades brasileiras. Segundo o sítio do Comitê, o primeiro programa licenciado sob a CC GPL foi o Terracrime, que será utilizado pelas polícias para apuração de dados estatísticos de ocorrências.

¹⁹A íntegra da Licença Pública Geral pode ser encontrada no Anexo, à página 135.

²⁰Sítio da Licença Pública Geral: <<http://www.lpg.adv.br>>, acessado em novembro de 2003.

²¹A íntegra da CC GPL pode ser encontrada no Anexo, à página 143.

²²Sítio Creative Commons: <<http://www.creativecommons.org>>, acessado em novembro de 2003.

²³Sítio do software livre no governo federal: <<http://www.softwarelivre.gov.br>>, acessado em dezembro de 2003.

2.4. MODELOS DE NEGÓCIOS

No meio empresarial sempre existiu a dúvida se o software livre, por não exigir guarda de segredos e nem pagamentos de licenças, poderia viabilizar um modelo de negócios sustentável para as empresas fornecedoras. Isso é uma preocupação importante por parte das empresas usuárias, um vez que grande parte delas procuram deixar diversas funções de informática e gerência de sistemas a cargo de fornecedores de serviços, que efetivamente precisam existir, serem saudáveis financeiramente e tecnicamente capazes. Outra questão é sobre a sustentabilidade do próprio modelo de desenvolvimento livre, e se as empresas poderiam basear seus negócios nesse tipo de software sem enfrentar problemas de continuidade, evolução e suporte. Os itens a seguir dão evidência a essas questões, procurando mostrar como o software livre está atualmente inserido no mundo empresarial.

2.4.1. PRODUÇÃO

Um dos maiores questionamentos a respeito da viabilidade ou da própria existência do software livre é a respeito do financiamento de seu desenvolvimento, uma vez que não há remuneração através do pagamento de licenças. Esse tipo de questionamento vem de um raciocínio que enquadra o software como um bem industrial, equiparável a um carro ou alimento. A questão é que o software não é um bem industrial: é um bem de informação.

A produção de qualquer tipo de software necessariamente implica em custos. No entanto, a maior parte desses custos são fixos, e gastos principalmente durante a fase de desenvolvimento. São chamados de custos amortizados (Shapiro; Varian, 1999). Além disso, bens de informação como música, vídeos, textos e software, podem ser reproduzidos infinitamente (principalmente com o advento das mídias digitais), sem qualquer perda de qualidade, e principalmente, sem degradação do objeto original. E com o uso da Internet, a distribuição desses bens tornou-se muito fácil e rápida. A produção de software, portanto, é cara, mas sua reprodução e distribuição são muito baratas. O custo marginal de distribuição de software, é próximo de zero. Isso implica na primeira característica fundamental para que seja possível a existência do software livre: uma vez pagos os custos de desenvolvimento, não é necessário qualquer investimento adicional para produção e distribuição de cópias, pois diferentemente de bens industriais, seu custo marginal é próximo de zero (Raymond, 2003).

O segundo questionamento é sobre a cobertura dos custos de desenvolvimento. O software pode ser desenvolvido para duas finalidades: para uso ou para venda. No software para venda, o valor está na remuneração paga pelo usuário final: o software tem valor em si. No software para uso, o valor está nos benefícios trazidos pelo software. O valor é intermediário, uma vez que o software é utilizado para viabilizar a produção e comercialização dos bens e serviços produzidos pela organização. Ao contrário do que se possa pensar, a maior parte do software desenvolvido no mundo está na categoria “para uso”. Não existem pesquisas específicas sobre esse assunto, mas alguns autores fazem estimativas. Raymond (2003), por exemplo, acredita que devido ao uso intensivo de software personalizado por organizações de serviços, principalmente financeiras, 95% de todo o software produzido é “para uso”, e apenas 5% “para venda”. No entanto, mesmo que esse número não esteja correto, a questão é que parte considerável dos custos globais de desenvolvimento de software é despendida na produção de um bem de valor intermediário, e não de um bem para comercialização. Isso quer dizer que o software, na maioria das vezes, não tem um valor comercial em si. O que tem valor é o benefício que ele traz.

Como visto nas seções anteriores, um software sob uma licença livre pode ser útil para várias empresas ou indivíduos, que podem melhorar, adicionar funcionalidades e resolver problemas. Visto que isso não causaria prejuízos para empresa, uma vez que esse software não tem valor de venda, e sua utilidade seria multiplicada por uma fração de que seria gasto no desenvolvimento individual, por que não tornar o software aberto? Raymond (2003) sustenta que em qualquer tipo de organização, especialmente nas empresariais, na grande maioria dos casos não há motivos para manter os programas fechados. E afirma que fazendo uma análise racional, é muito mais vantajoso, em termos de custos e obtenção de funcionalidades, manter projetos de código aberto, do que tentar desenvolver tudo utilizando o modelo fechado e proprietário. Raymond (2003) descreve, então, três razões pelas quais as empresas poderiam preferir manter seus programas fechados, e as contesta, procurando mostrar que o modelo aberto é mais vantajoso.

A primeira razão é que o software em questão, embora não seja fonte direta de receitas, talvez seja um fator de diferenciação competitiva da empresa. Tomemos o exemplo de uma empresa que fornece serviços de manutenção de máquinas, e utiliza um software desenvolvido internamente para controle de chamados e de intervenções preventivas. Tornar esse software livre pode ser interessante para empresa? A resposta é não, se esse programa for considerado um objeto de diferenciação em relação à concorrência. No entanto, se a empresa

considerar que pode obter melhorias significativas em seu software tornando-o livre, uma vez que outras empresas que prestem serviços com características parecidas vão utilizá-lo, ajudar a mantê-lo e fazer melhorias em seu funcionamento, a resposta é positiva. Ou seja, a empresa pode ter um software melhor, a uma fração do custo normal de desenvolvimento. E pode também analisar que esse software é uma ferramenta importante, mas que seu real diferencial competitivo vem de outros fatores diretamente relacionados a seu negócio, como por exemplo seu conhecimento do mercado, o histórico de bons trabalhos e a capacitação técnica de seu pessoal. Isso leva a que uma outra empresa concorrente, apenas pelo fato de ter disponível seu software não ameaçaria sua posição de mercado. Nesse caso, poderia ser interessante, abrir o código do software e torná-lo livre, uma vez que os benefícios seriam muito maiores que os riscos ao negócio. Um segundo exemplo: o software necessário ao controle de dispositivos (drivers), como interfaces de rede, placas de som, placas de vídeo. Manter esse tipo de software fechado não faz sentido, porque em primeiro lugar, ele não tem um valor de venda em si. O que tem valor de venda é o dispositivo, sendo que o software que o controla deve ser necessariamente fornecido pelo fabricante. Por isso, é melhor que o driver seja livre, e as especificações abertas, de modo que qualquer interessado possa melhorá-lo e escrever outras versões direcionadas para sistemas operacionais e aplicações inicialmente não planejados pelo fabricante. Isso gera economia, uma vez que se o dispositivo for popular, não é necessário manter uma grande equipe de desenvolvimento para produzir drivers para várias plataformas. Em resumo, deve ser feita uma análise de riscos e benefícios ao tomar a decisão de abrir um programa desenvolvido que é utilizado internamente à organização. No entanto, Raymond (2003) sustenta que a maioria das empresas que toma essa decisão subestima seus benefícios: a maior parte dos resultados supera muito as expectativas iniciais.

O segundo motivo para não abrir o software seria a necessidade de manutenção de segredos de negócio. O programa pode conter procedimentos e regras de negócio secretas, cuja abertura corresponderia à perda de diferencial competitivo. No entanto, esse problema pode ser resolvido com otimizações na metodologia de desenvolvimento, de modo que o software base seja separado do módulo que contém as regras de negócio. Com isso, a empresa poderia se beneficiar da ajuda a comunidade software livre abrindo o software base, sem ter que divulgar seus segredos.

O terceiro motivo para manter o código fechado seria a segurança. No entanto, esse tipo de segurança, chamada de segurança por obscurantismo, é fortemente criticada por especialistas (Brunazo, 2001; Rezende, 2002), porque impede a revisão e verificação dos programas por

peças de fora da equipe original de desenvolvimento. Isso faz com que muitas das falhas sejam descobertas somente no momento de sua ocorrência, e impede a prevenção de adição de códigos maliciosos ou estruturas camufladas de acesso.

Podemos citar o exemplo de algumas grandes empresas com tradição no desenvolvimento de programas proprietários, mas que decidiram colaborar com projetos em software livre:

- O sistema de gestão empresarial e de gestão de relacionamento com clientes Compiere²⁴, inicialmente desenvolvido pela empresa Compiere Inc. sob encomenda da Goodyear da Alemanha, é licenciado como software livre.
- A Silicon Graphics (SGI) liberou seu sistema de arquivos XFS²⁵, que tem características ideais para utilização em ambientes com necessidade de grande capacidade e desempenho, como software livre. Desde o anúncio esse software foi muito aprimorado, e fez com que sua funcionalidade dentro do sistema operacional proprietário da SGI, o Irix, também usufrua desses benefícios.
- A IBM e a Oracle²⁶ participam ativamente do desenvolvimento do núcleo Linux, pois têm grande interesse na utilização de seus produtos sobre essa plataforma. As melhorias introduzidas por essas empresas no Linux geram benefícios para a execução de seus produtos, e ao mesmo tempo colaboram para o desenvolvimento global do sistema operacional.
- A Sun Microsystems liberou o programa StarOffice, de sua propriedade, sob uma licença livre. Isso originou o projeto OpenOffice.org²⁷, cujos produtos são amplamente utilizados, e são desenvolvidos por uma vasta gama de pessoas e instituições. As melhorias introduzidas pelo OpenOffice.org servem de base para o StarOffice, da Sun.

E além desses exemplos de grandes empresas, há inúmeros, envolvendo empresas menores, instituições públicas e universidades, que seguem a mesma lógica. No portal Código Livre (www.codigolivre.org.br) podem ser encontrados diversos projetos já surgidos ou tornados livres, executados por desenvolvedores brasileiros.

O financiamento da produção de software livre por parte de empresas e outros tipos de organizações é importante, mas não explica completamente a cobertura de seus custos de

²⁴Disponível em: <<http://www.compiere.org>>. Acesso em novembro de 2003.

²⁵ <<http://www.sgi.com/xfs>>

²⁶<<http://www.ibm.com/linux>> e <<http://www.oracle.com/linux>>

²⁷<<http://www.sun.com/staroffice>> e <<http://www.openoffice.org>>

desenvolvimento. Uma parte significativa desses custos são assumidos por pessoas físicas, que doam seu tempo para melhorar e desenvolver software livre sem o patrocínio formal de instituições ou empresas. Os motivos do desenvolvimento desse trabalho são os mais diversos, e vão desde pura diversão até o uso profissional, passando por necessidades de reconhecimento, de aprendizagem técnica e por motivos ideológicos, como vimos nas seções sobre a FSF e o projeto GNU.

Além disso, o desenvolvimento de software livre tende a ser mais barato que o de software proprietário (Stallman, 2002; Raymond, 2003). Como o código fonte, conceitos e métodos utilizados estão livremente disponíveis, é muito comum o reaproveitamento do conhecimento pré existente. Além disso, como a utilização desses programas é livre de pagamento de royalties, mesmo que não seja utilizado o código fonte diretamente, produtos díspares podem ser integrados. Um corretor ortográfico, por exemplo. Ao fazer um editor de textos, o desenvolvedor não tem que programar novamente um corretor ortográfico caso este já exista. Ele simplesmente utiliza um software que já esteja pronto e disponível. Quando um outro desenvolvedor trabalha num cliente de correio, por exemplo, não precisa necessariamente desenvolver um novo editor de textos. Pode utilizar o que já está pronto, e que por sua vez já utiliza o corretor ortográfico. Os projetos em software livre são desenvolvidos analogamente à construção de um quebra-cabeças, onde peças são encaixadas para formar um conjunto orgânico. Os desenvolvedores de software livre sustentam que o modelo proprietário é menos eficiente, pois é necessário “reinventar a roda” a todo momento, uma vez que não é possível reutilizar tecnologias que já estejam prontas e disponíveis.

No entanto, é importante frisar que o modelo aberto embora de modo geral seja bem sucedido e tenha conseguido bons resultados em inúmeras situações, não é a panacéia para os problemas do desenvolvimento de software. Se o projeto aberto não tiver um bom planejamento, boa documentação e se não for especialmente útil, poucas são suas chances de sucesso. Bezroukov (1999), por exemplo, contesta a visão de Raymond (2003), criticando-a como sendo genérica e idealizada, mostrando que existem vários cuidados a serem tomados, principalmente para evitar problemas de coordenação e direcionamento que levem o projeto à morte.

2.4.2.COMERCIALIZAÇÃO

Para sustentar o uso empresarial do software livre, é necessário que existam modelos de negócios viáveis, que tragam receitas para as empresas de tecnologia da informação interessadas em atuar nesse mercado.

Obviamente, a venda do software em si não faz sentido, uma vez que não podem ser cobradas taxas de licenciamento. Raymond (2003) sugere os seguintes modelos para sustentar negócios baseados em software livre:

- **Base para produto proprietário** – Um software livre é utilizado como base para o desenvolvimento de um produto proprietário com características exclusivas. Esse modelo é utilizado no caso já citado da Sun Microsystem e seu conjunto de escritório StarOffice, e pela Apple com o sistema operacional Mac OS X²⁸, baseado no sistema operacional Darwin, que por sua vez é baseado no BSD.
- **Serviços** – A prestação de serviços é o ramo mais utilizado para obtenção de receitas com software livre. Há várias opções disponíveis, como alterar programas para atender a necessidades específicas, fazer implementações de programas já existentes, prestar consultoria e oferecer treinamento. Exceto quando há alteração de código fonte, esse modelo é essencialmente o mesmo que já é utilizado pela maioria das empresas de serviços de informática. Seu ganho real está na venda de serviços, e não na venda de licenças de software. Com isso, o uso de software livre pode representar um grande benefício, uma vez que o preço global dos projetos apresentados pode ser substancialmente reduzido, devido à inexistência (ou diminuição) de itens envolvendo aquisição de licenças. As margens de lucro com a venda de licenças em geral são pequenas, mas mesmo assim eventuais perdas de receitas com o fim de sua comercialização podem facilmente ser transferidas para o preço dos serviços, e ainda assim manter a oferta de redução de custos para o cliente. Ou seja, por ser mais barato, o projeto com software livre é mais fácil de ser vendido, e ainda assim tem a possibilidade de trazer maiores receitas com a majoração dos preços dos serviços, sem prejuízo para o cliente. Outro exemplo de modelo de serviços é o das “distribuições GNU/Linux”, que vendem pacotes de software livre. O que é oferecido, na verdade, é o serviço da conveniência de ter um sistema operacional completo, integrado e livre de problemas de compatibilidades. As empresas mais famosas que atuam nesse mercado

²⁸Ver em <<http://www.apple.com>> e <<http://www.darwin.org>>. Acesso em novembro de 2003.

são a brasileira Conectiva, a estadunidense Red Hat, a alemã Suse (adquirida pela Novell em novembro de 2003) e a francesa Mandrake.

- **Venda de acessórios** – Há empresas que atuam na venda de produtos relacionados ao software livre, como conjuntos de software, livros, apostilas e revistas. Um exemplo é a editora O'Reilly, especializada e envolvida em projetos de software livre, a brasileira Revista do Linux e as estadunidenses Linux Mall e Linux Magazine.
- **Uso embarcado** – Sistemas operacionais livres podem ser utilizados por fabricantes de dispositivos de hardware, como controladores de máquinas industriais ou computadores de mão. Com isso, é possível conseguir uma grande redução de custos, e alto nível de compatibilidade com o mercado, uma vez que a empresa produtora do hardware não precisa desenvolver um novo sistema proprietário, não precisa licenciar algum já existente e tem a garantia de compatibilidade com os padrões abertos. Um exemplo é o computador de mão Zaurus, da Sharp, que tem como sistema operacional uma versão do GNU/Linux.

2.4.3.O uso

Sem dúvidas um dos grandes motivadores para a utilização de software livre nas empresas é a não necessidade de pagamento de licenças. Embora existam pesquisas que indiquem que esses custos têm pequena participação no custo total de TI (BOZMAN et al, 2002), ele efetivamente é um componente importante, e eventualmente uma barreira para implementação de soluções, principalmente em empresas com pequena estrutura de informática.

Além de redução de custos com licenciamento, Raymond (2003) elenca outros motivos para o uso de software livre:

- A possibilidade de ter maior qualidade, estabilidade, desempenho e segurança, devido às “revisões pelo pares”, empreendidas pelos demais participantes do desenvolvimento do software. Raymond sustenta que quando o projeto é popular, o desenvolvimento é muito rápido e produz resultados muito satisfatórios, como no caso do desenvolvimento do Linux, do BSD e do GNU.

- A possibilidade de personalizar o software para atender às necessidades específicas do usuário.
- A possibilidade de evitar o monopólio exercido por fornecedores de software proprietário. Possibilita também garantir que o software utilize padrões abertos de comunicação e armazenamento de arquivos, livres de extensões que possam fazer com que as informações fiquem presas a produtos de um fornecedor específico.
- A possibilidade de ter garantia contra a descontinuidade, pois no caso de abandono do produto por seus desenvolvedores, a empresa usuária pode contratar os serviços de desenvolvimento e evolução, baseados em suas necessidades. Um exemplo desse caso é o próprio Linux²⁹, cujas versões anteriores (a versão estável atual é a 2.4.23) ainda são desenvolvidas e atualizadas por aqueles que delas necessitam. Existiam características na versão 1.0.x, por exemplo, que foram suprimidas nas versões posteriores. As pessoas que tinham programas que necessariamente precisavam delas, puderam continuar utilizando a versão antiga, que continua sendo mantida. Outro exemplo de continuidade proporcionado pelo software livre é o do software de modelagem em três dimensões Blender³⁰. Esse programa era proprietário, e chegou a ser um dos mais populares em sua categoria. Em 2001, depois de vários revezes, o produto foi descontinuado devido a dificuldades financeiras de sua proprietária, a holandesa NaN. Em 2002, Ton Roosendaal, o desenvolvedor original do Blender, criou uma fundação, cujo objetivo era comprar os direitos do programa, transformá-lo em software livre e continuar seu desenvolvimento. Os detentores dos direitos à época concordaram em vendê-los pelo valor de 100.000 euros, que foram levantados em 7 semanas, numa campanha mundial de arrecadação de fundos. Hoje o Blender é o mais popular software livre para manipulação em 3D, e está licenciado pela GNU GPL.

A consequência disso é que o software livre então, tem se tornado cada vez mais popular:

- Segundo a 14ª pesquisa anual "Administração de Recursos de Informática" da Fundação Getúlio Vargas (Meirelles, 2003), o uso de sistemas operacionais baseados no GNU/Linux como servidor teve um crescimento de 50% de 2001 a 2002.
- Segundo o IDC e Gartner Group (apud Wheeler, 2002), os sistemas operacionais que mais crescem no mundo, no segmento de servidores, são baseados no GNU/Linux.

²⁹Veja em <<http://www.kernel.org>>. Acessado em novembro de 2003.

³⁰<<http://www.blender3d.org>>

- Segundo o sítio de pesquisas Netcraft (www.netcraft.com), o software servidor de páginas Web líder de mercado, com cerca de 65% de participação em outubro de 2003 é o Apache, um software livre.
- Entre a comunidade de tecnologia da informação, os softwares livres em geral são reconhecidos como estáveis, confiáveis e seguros (Hippel, 2001; Wheeler, 2002; Kenwood, 2001).
- Inúmeros projetos de software livre estão sendo desenvolvidos em toda parte do mundo, envolvendo milhares de pessoas e centenas de empresas, instituições de ensino e órgãos governamentais. Além do já citado portal Código Livre, que hospeda projetos brasileiros, o sítio Sourceforge³¹ hospeda e gerencia cerca de 40.000 projetos de software livre.

O software livre é, portanto, um exemplo notável de organização em grupo. Milhares de pessoas ao redor do mundo estão utilizando e desenvolvendo o software livre, e trabalhando efetivamente para seu suporte e crescimento. As perspectivas para o seu desenvolvimento futuro são interessantes, e percebe-se uma tendência de aumento em seu uso (Meirelles, 2003). Isso pode acarretar que conceitos arraigados durante muitos anos a respeito de como devem se organizar pessoas em torno de projetos e de como deve-se lidar com o uso e a propriedade do software sejam reavaliados.

No entanto, todas essas questões de licenciamento, propriedade, uso e desenvolvimento são desconhecidas pelo público empresarial, até mesmo por grande parte dos gestores de TI. O crescimento o uso de software livre vem sendo impulsionado mesmo pela não necessidade de pagamentos de licenças. Como veremos nas próximas seções.

³¹<<http://www.sourceforge.net>>

3. A PESQUISA

3.1.O PROBLEMA DE PESQUISA

Embora o modelo de desenvolvimento e os resultados obtidos pelos softwares livres sejam, entre os profissionais de informática, reconhecidos como eficientes; e existam produtos disponíveis para as mais diversas finalidades (Hippel, 2001), ainda é pequena sua participação no mercado brasileiro. Segundo a 14ª pesquisa anual "Tecnologia da Informação" da fundação Getúlio Vargas (Meirelles, 2003):

- Embora tenha tido um crescimento expressivo no mercado de servidores de 2001 para 2002, apenas 12% dos servidores são equipados com software livre, especificamente o GNU/Linux.
- Em nenhuma das outras categorias de software da pesquisa (planilhas, processadores de texto, apresentações, bancos de dados, programas integrados de escritório, sistemas operacionais para estações, navegadores, correio eletrônico) qualquer software livre foi citado, embora existam várias alternativas³².

Existe a tendência de que a implementação de software livre seja mais barata que o software proprietário. Algumas características do software livre apontam para essa possibilidade: não são necessários gastos com a aquisição de licenças de uso para implementação nem para atualização dos produtos, e não há necessidade de gerenciamento dessas licenças. Em grandes empresas, dependendo da quantidade de licenças possuídas, somente para gerenciar as licenças possuídas é necessária uma estrutura que demanda espaço físico, tempo de trabalho e programas de gerenciamento. O fato das soluções em software livre serem reconhecidas como mais estáveis também pode contribuir com a redução de custos, uma vez que é necessária menor manutenção do sistema. Ainda não existem estudos conclusivos, mas alguns autores já fizeram análises sobre o tema. A empresa australiana Cybersource³³ preparou um estudo comparativo entre implementações de um ambiente de rede utilizando somente softwares livres, e outra rede utilizando somente softwares proprietários. Chegou à conclusão que a solução em software livre pode ser até 35% mais econômica. Scott (2002), baseado na teoria econômica do monopolista, sustenta que a longo

³² No anexo, à página 124, há uma relação de softwares livres categorizados.

³³ http://www.cyber.com.au/cyber/about/linux_vs_windows_tco_comparison.pdf em maio de 2002

prazo as soluções em software livre devem ser necessariamente mais baratas. E Kenwood (2001) cita estudos divulgados pelo Gartner Group e Computerworld para sustentar que o vencedor em termos de custos varia de acordo com a aplicação e com circunstâncias específicas, mas que as soluções em software livre se destacam na maioria delas.

Além disso, segundo Faldetta (2002), Raymond (2001) e O'Reilly (1999), é difundida no mercado de informática a percepção de que o software livre é estável, eficiente e requer menos manutenção que o software proprietário.

Esses fatores, adicionados às características apresentadas no capítulo anterior, indicam que podem existir vantagens objetivas na adoção de soluções em software livre nas empresas. Isso posto, temos então o problema de pesquisa:

- Que fatores levam os produtos em software livre a ter participação de mercado expressivamente menor que os produtos em software proprietário?

Podemos assumir que a decisão de adotar ou não uma tecnologia depende fundamentalmente da política de informática na empresa, que é avalizada, implementada e fortemente influenciada pelo responsável pela tecnologia de informação (Seddon et al, 2002). Então, responder a essa pergunta implica necessariamente em entender qual é a posição dos gestores de TI nas empresas a respeito do software livre. Isso nos leva aos seguintes problemas acessórios:

- Acreditam os gestores de TI que as soluções de software livre têm condições de suportar em larga escala a infra-estrutura tecnológica da empresa?
- Qual a percepção dos gestores de TI a respeito das características técnicas e da qualidade do software livre?
- Qual a percepção dos gestores de TI a respeito da adoção futura do software livre em larga escala nas empresas?

O modelo conceitual da pesquisa é o seguinte:

1. As variáveis antecedentes são as razões pelas quais uma empresa adota uma determinada tecnologia. Isso envolve principalmente questões relacionadas a funcionalidades, custos e riscos. Inserido nesse aspecto estão os fatores pelos quais o software livre tem uma participação pequena no mercado brasileiro.

2. O critério será a participação de mercado das soluções em software livre e em software proprietário, medidos pela pesquisa da FGV (Meirelles, 2003).

3. Os fatores intervenientes serão as características da empresa e de seu negócio.

O gráfico abaixo representa esse modelo:

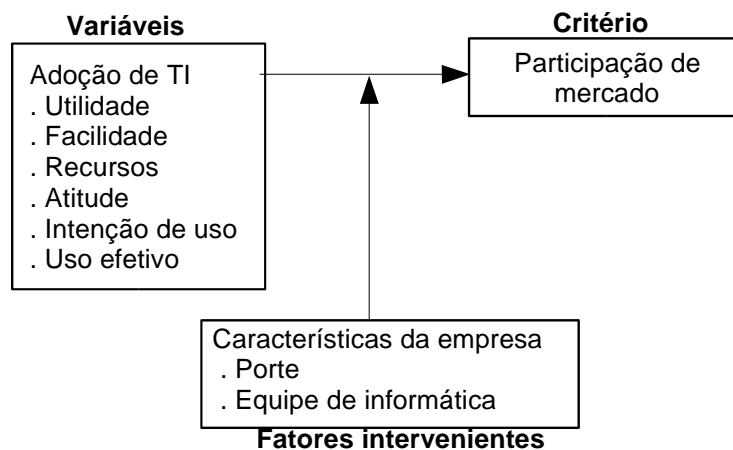


Figura 1 - Modelo conceitual da pesquisa

O objetivo do estudo é, então, fornecer aos envolvidos com o desenvolvimento, comercialização, prestação de serviços e suporte ao software livre uma base de conhecimentos que oriente suas ações no sentido de melhorar, e conseguir o aumento da base instalada.

3.2.PREMISSAS

Para a formulação da pergunta de pesquisa, foram adotadas as seguintes premissas:

1.Os gestores de TI têm influência determinante na escolha e na adoção de softwares nas empresas (Seddon et al, 2003), portanto suas percepções, opiniões e conceitos são fundamentais para entender as escolhas efetuadas.

2.Existem alternativas viáveis em software livre para a maior parte das necessidades empresariais, e esses softwares têm funcionalidades similares, e em alguns casos superiores às alternativas proprietárias (Castellucio, 2000; Behlendorf, 1999).

3.A implementação e a manutenção de produtos em software livre podem ser mais baratas que a implementação de software proprietário (pág. 42).

A pergunta de pesquisa busca saber porque, uma vez que há na bibliografia indicações de vantagens concretas na adoção de software livre, sua participação de mercado ainda é tão pequena.

3.3. PROPOSIÇÕES

Segundo a pesquisa da FGV (Meirelles, 2003), o sistema operacional Microsoft Windows está presente em 97% das estações de trabalho de usuários. Como o sistema operacional é a base dos sistemas de informática, os demais aplicativos tendem a seguir o padrão liderado por ele. Com isso, a Microsoft conseguiu participações expressivas em outros mercados, como o de navegadores para Internet (Internet Explorer - 90%) e aplicativos integrados para escritórios (Office - 93%). Tamanha concentração de mercado acabou criando um padrão "de fato", onde formatos de arquivos e métodos de trabalho são diretamente relacionados ao Windows e seus aplicativos. Como o Windows é um sistema operacional proprietário, as opções de software livre para essa categoria são muito restritas. Isso nos leva à primeira proposição:

- P1. A adoção de sistemas em software livre tem uma forte barreira de entrada, pois o uso do software proprietário perpetua-se de forma inercial devido à ampla base instalada.

Segundo Mingji (2002), a perspectiva da inovação torna a inércia explícita. A inércia está inserida no coração das organizações, e quando uma nova idéia desafia as normas existentes, surgem os focos de tensão. Conscientes dessa possibilidade, os próprios responsáveis pela administração da empresa, e conseqüentemente os gestores de TI, se tornam elementos de resistência à mudança. Além disso, pressões externas podem limitar o espaço de manobra das empresas. Na medida em que a ampla maioria do mercado se mostra favorável a um determinado conceito, é mais difícil, colocar-se contra ele.

Como o Windows e seus aplicativos são um padrão "de fato", o custo de integração e migração para um sistema de software livre pode, pelos gestores de TI, ser considerado alto. Haveria a necessidade de testar, homologar, instalar e oferecer treinamento aos usuários e à equipe de suporte. E isso pode não ser uma tarefa fácil, especialmente em ambientes complexos. Dissel (2000), numa série de entrevistas realizada com usuários de sistemas de

gestão empresarial, concluiu que uma parcela significativa percebeu o processo de migração como "difícil, custoso e trabalhoso". O tempo gasto e os custos incorridos foram vistos como as questões mais importantes, seguidos por problemas com bugs, estimativas não confiáveis de tempos e custos feitos pelos fornecedores, falhas nas ferramentas de migração e o stress gerado em toda a empresa pela convivência com o processo. As empresas que não iniciaram o processo de migração tomaram essa decisão devido principalmente à falta de dinheiro, ou então pelo fato do período de retorno do primeiro investimento ainda não ter se completado. Taudes (2000), também tratando de migração de sistemas de gestão, considera o fator econômico como um dos mais importantes, e propõe um modelo para verificar a viabilidade de projetos de implementação de infra-estrutura de informática. Embora esses autores não se refiram especificamente a soluções de software livre, as variáveis por eles levantadas dão mostras da relevância dos problemas da migração de sistemas. Colocamos, então, os seguintes desdobramentos de P1.

A utilização do software livre é considerada, pelos gestores de TI, inconveniente devido:

- P1a. Aos altos custos de migração.
- P1b. À necessidade de treinamento da equipe de informática.
- P1c. À dificuldade em mensurar a viabilidade do investimento.
- P1d. À grande quantidade de tempo necessária.
- P1e. À possibilidade de problemas técnicos.
- P1f. À possibilidade de incômodo ao usuário dos sistemas, o que pode contribuir para uma visão negativa do setor TI.

Como a adoção de um novo sistema nas estações de trabalho envolve um grande trabalho político, de convencimento e de superação de resistências internas (Mingji, 2002; Dissel, 2000), muitos gestores de TI não se dispõem a sofrer o desgaste necessário à troca da plataforma.

- P1g. A indisposição por parte dos gestores de TI a fazer o trabalho político necessário à substituição da plataforma é uma barreira à adoção do software livre nas empresas.

Podem ser levantadas também questões relativas a externalidades de rede (Ozsomer, 2000). Uma rede relevante é aquela que compartilha tecnologias compatíveis. No caso de hardware de computadores, por exemplo, os proprietários de equipamentos que utilizam processadores Intel e compatíveis usufruem muito mais de externalidades, como preços mais baratos, facilidade de encontrar suporte e a existência de uma ampla rede de distribuição. Ao contrário dos usuários de hardware Apple, que contam com uma rede bem menor. Externalidades de rede existem quando o tamanho da base usuária afeta a decisão, por parte dos usuários, de utilizar uma determinada tecnologia. Ainda segundo Ozsomer (2000), essa rede de externalidades é fundamental na determinação de padrões tecnológicos. Padrão tecnológico é um modelo ou especificação que tem uma posição dominante num determinado mercado. Um padrão pode surgir de uma agência reguladora, de uma comunidade de usuários, ou pode ser o resultado de uma disputa de mercado. Da última alternativa surgem os padrões "de fato". Na maioria das vezes os padrões "de fato", surgem de uma realidade de competição e são propriedade de empresas, que mantêm suas especificações fechadas ou exigem taxas de licenciamento para seu uso. E devido à questão das externalidades, esses padrões são amplamente utilizados no intercâmbio de arquivos entre pessoas e empresas.

Por isso, além da dificuldade de migração, existe a questão da compatibilidade dos arquivos. Os arquivos produzidos em aplicativos para Windows são em sua grande maioria de formato proprietário e secreto, e em muitos casos se tornaram um padrão "de fato" para troca de documentos. Isso acontece no caso do Word, para troca de arquivos de textos, para o Excel, na troca de planilhas e no AutoCad para desenhos de engenharia.

Isso quer dizer que um arquivo produzido no aplicativo Word, da Microsoft, tem seu formato secreto, e em tese não pode ser aberto por outros aplicativos desenvolvidos por outras empresas ou organizações. Como esses formatos não são publicamente documentados, as equipes produtoras de software livre têm que fazer engenharia reversa para descobrir como o formato proprietário é estruturado, e então escrever o aplicativo e os filtros que serão utilizados para abrí-los. Esse é um processo que, apesar de ser penoso e consumir muito tempo, produz resultados até certo ponto satisfatórios. Softwares livres são, então, capazes de utilizar formatos proprietários, mas com certas limitações. Para empresas que têm uma grande base legada ou que trocam muitos arquivos com fornecedores e clientes, isso pode ser um

problema. Daí vem o fato que empresas que utilizam intensivamente arquivos em formatos que se tornaram um padrão "de fato", têm importante barreira para a adoção de software livre.

- P1h. A imperfeição na abertura de arquivos de formato proprietário, que se tornaram padrões "de fato", por softwares livres é uma barreira à adoção.

As equipes que desenvolvem software livre dispõem de poucos recursos, e os existentes (materiais e humanos) são utilizados basicamente para melhorar a funcionalidade dos produtos desenvolvidos (Faldetta, 2002; Raymond, 2001). Por isso há pouco esforço direto de comunicação, e a propaganda dos produtos, recursos e funcionalidades é feita quase que unicamente através da rede de usuários, de sítios na Internet e algumas publicações especializadas. Por falta de informação, muitos gestores de TI não conhecem as possibilidades oferecidas pelo software livre. Além disso, Kenwood (2000) mostra que os sistemas operacionais derivados do GNU/Linux, e as demais ferramentas em software livre são vistas pelas empresas usuárias como recursos desenvolvidos por programadores para programadores. Por isso, existe a visão que o software livre é mais difícil de ser manuseado, aprendido e utilizado por um usuário leigo em informática.

- P1i. O desconhecimento dos produtos e das funcionalidades oferecidas por ferramentas de software livre é uma barreira para sua adoção.
- P1j. As soluções em software livre são consideradas, pelos gestores de TI, como mais difíceis de serem utilizadas por usuários leigos em informática, o que implicaria em grandes necessidades e altos custos de treinamento.

Segundo a pesquisa da FGV (Meirelles, 2003), 88% dos servidores utilizam softwares não livres. No entanto, registrou-se um crescimento de 50% no uso do sistema operacional Linux entre 2001 e 2002. Esse crescimento pode sugerir um processo mais consistente de adoção do software livre nos servidores. Mas, como sua participação geral no mercado de servidores ainda é relativamente pequena, podemos inferir que esse processo está em andamento. Como as alternativas para uso de software livre começaram a vir ao grande público há menos tempo que as alternativas proprietárias, nesse momento provavelmente está sendo superada a marca dos adotantes iniciais, e essas iniciativas estão começando a se espalhar para um âmbito mais geral.

Segundo pesquisa divulgada pelo instituto IDC (2000) em abril de 2000, nos Estados Unidos o uso de software livre em servidores concentra-se em aplicações para Internet, como

gateways, servidores de nomes, servidores Web e de correio. É bem menos utilizado em servidores de bancos de dados e servidores de aplicação, que armazenam os dados de missão crítica da empresa. Ao mesmo tempo, segundo a Fapesp³⁴, houve nesse período um crescimento exponencial de registros de domínios por parte de empresas na Internet brasileira. Conforme o visto nas proposições anteriores, existem fortes indícios que os gestores de TI ainda têm reservas quanto à adoção generalizada do software livre. Baseado nas informações apresentadas, podemos propor que os gestores de TI acreditam que o software livre seja viável principalmente às aplicações tradicionais de Internet. Por isso esse crescimento expressivo do software livre mostrado pela pesquisa da FGV (Meirelles, 2003), pode ter um forte componente referente à adoção da Internet e de tecnologias a ela relacionadas, embora os aplicativos de negócio ainda continuem sendo executados em servidores com software proprietário. Podemos, então, formular as proposições:

- P2. Os gestores de TI vêem o software livre como alternativa viável para determinados serviços em **servidores** corporativos, e têm intenção de utilizá-lo apenas em parte de seu ambiente.
- P2a. O rápido crescimento do uso do software livre nos **servidores** nos últimos anos (2000, 2001 e 2002) tem um forte componente da adoção das tecnologias da Internet, e não de migração de sistemas existentes.

Além disso, ainda é forte nos gestores de TI a percepção da necessidade de que uma grande empresa forneça sua infra-estrutura de software de negócios. Como o software livre é desenvolvido principalmente por grupos independentes não empresariais, há uma desconfiança quanto sua capacidade de suportar os processos corporativos. Segundo Kenwood (2001) e Castellucio (2000), os gestores de TI têm grandes preocupações relativas ao fato das soluções em software livre não serem propriedade de uma empresa específica. Isso leva a questões como disponibilidade de novas versões e responsabilidades pelo funcionamento dos produtos. Ainda segundo os autores, a falta de propriedade também levanta questões relativas à disponibilidade de suporte técnico, tanto para implementação quanto para operação do sistema. Embora a bibliografia disponível sobre o tema (inclusive os autores citados) e a discussão do capítulo anterior se esforcem em mostrar que efetivamente há alternativas de suporte técnico, e que a não existência de "donos" dos sistemas não altera o *status quo* vigente, esses fatores não podem ser ignorados, e são importantes na tomada de decisão sobre a adoção da tecnologia. Isso nos leva, então, às seguintes proposições:

³⁴Sítio de registro de domínio: <<http://registro.br>>. Acesso em novembro de 2003.

- P2b. Os gestores de TI não acreditam que o software livre seja capaz de suportar as aplicações de negócio da empresa.
- P2c. Não há intenção de migrar as aplicações de negócio existentes para uma plataforma de software livre.
- P2d. Existem dúvidas, por parte dos gestores de TI, sobre a disponibilidade de suporte técnico.

Apesar disso, a imprensa especializada em informática, que é normalmente utilizada como fonte de informação pelos gestores de TI, trata na maioria das vezes as soluções de software livre com muita condescendência. Wheeler (2002) traz uma extensa lista de artigos e pesquisas publicadas sobre software livre, e mostra que quase sempre são feitas referências elogiosas, principalmente a aspectos técnicos como confiabilidade, desempenho, capacidade de expansão e segurança. O autor aponta também questões levantadas contra o software livre, mas mostra que ocorrem em menor quantidade e possuem críticas não tão ácidas. Essa carga de informações positivas possibilitaria a formação de um bom conceito geral a respeito do software livre, e criaria uma situação quase que paradoxal: os gestores de TI teriam uma percepção muito positiva a respeito do software livre, mas ao mesmo tempo não se sentiriam seguros para sua adoção como suporte aos aplicativos de negócios.

- P3. Os gestores de TI reconhecem que os produtos de software livre de modo geral têm qualidades técnicas comparáveis ou superiores aos produtos de software proprietário.

O estudo se baseia, então, nas proposições lançadas acima. Essas proposições não são, no entanto, "hipóteses para verificação", inferidas empiricamente por meio de pesquisa específica anterior. São proposições de trabalho baseadas no conhecimento prévio da situação existente e na bibliografia sobre o tema, e foram formuladas apenas com o propósito de orientar a pesquisa, e servir como base para as discussões que se seguem.

4. O MODELO DE ACEITAÇÃO DE TECNOLOGIA

Tendo como premissa que as soluções em software livre são tecnicamente implementáveis e que podem trazer vantagens concretas às empresas, o entendimento do processo de decisão de adoção de tecnologias é fundamental para explicar sua atual posição no mercado.

Davis (1980) introduziu o modelo de aceitação de tecnologia (MAT). Esse modelo tem por objetivo auxiliar os responsáveis pela implementação de sistemas de informação a avaliar sua aceitação atual e futura. Existem estudos que utilizam o MAT, por exemplo, para avaliar o uso e a aceitação de processadores de texto (Chau, 2001), para verificar percepções de compatibilidade entre sistemas (Chau; Hu, 2001), para avaliar reações emocionais ao uso de computadores (Compeau; Higgins, 1995), para avaliar o uso de pacotes de produtividade (Doll, 1998) e para dar suporte a produtores de software que querem verificar a demanda por novas idéias e produtos. (Davis, 1989).

O MAT é uma adaptação da teoria da ação racional, vinda da psicologia, e modificada especificamente para criar modelos de aceitação de tecnologia da informação (Gahtani, 2001). Segundo Davis (1985), o objetivo do MAT é possibilitar uma explicação dos determinantes da utilização de computadores, capaz de contemplar comportamentos de usuários através de uma grande gama de tecnologias e populações. O modelo leva em consideração que estímulos externos influenciam as atitudes pessoais, indiretamente influenciando suas crenças sobre as conseqüências de ter aquele comportamento. Um ponto chave do MAT é, então, verificar o impacto de fatores externos nas crenças e atitudes das pessoas. Foi formulado para atingir esses objetivos através da identificação de um pequeno número de variáveis fundamentais sugeridas por pesquisas prévias, relacionadas com determinantes cognitivos e afetivos da aceitação de tecnologia da informação (Gahtani, 2001).

O MAT se baseia na idéia de que dois conceitos principais, a utilidade percebida e a facilidade percebida (Davis, 1989), podem prever aceitação de sistemas computacionais. Segundo Davis (1989), as pessoas tendem a usar ou não uma aplicação ou tecnologia de acordo com a possibilidade de melhorar seu desempenho no trabalho. Isso é chamado de utilidade percebida. No entanto, mesmo que o usuário entenda que um determinado aplicativo é útil, sua efetiva utilização pode ser prejudicada se o uso for considerado muito complicado, de modo que os benefícios da nova tecnologia não compensem o esforço do uso. Esse conceito é chamado de facilidade percebida (Davis, 1989; Doll, 1998; Gahtani, 2001).

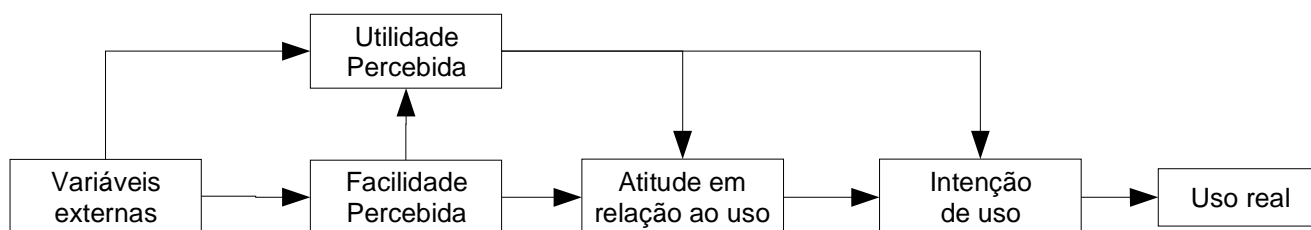


Figura 2 - Modelo de adoção de tecnologia

Definimos, então, os conceitos:

Utilidade percebida é o grau o qual uma pessoa acredita que o uso de um determinado sistema pode melhorar ou trazer vantagens a seu desempenho no trabalho. Mantidas todas as outras condições iguais, um sistema com maior utilidade percebida tende a ser melhor aceito pelos usuários.

Facilidade percebida é o grau o qual uma pessoa acredita que o uso de um determinado sistema pode ser livre de esforços. Esforço é mobilização de recursos físicos, materiais e morais que uma pessoa pode alocar ao realizar determinadas tarefas. Mantidas as outras condições iguais, um sistema com maior facilidade percebida tende a ser melhor aceito pelos usuários.

Segundo testes conduzidos por Davis (1989), Mathieson (2001) e Chau (2001), entre outros, a utilidade é fortemente influenciada pela facilidade de uso, conforme mostra a Figura 2. E tanto a facilidade quanto a utilidade influenciam a atitude que o usuário terá em relação ao sistema, que é um fator determinante da forma como o sistema será aceito.

Atitude é definida como o modo de se comportar, proceder ou agir. É a reação ou maneira de ser em relação a um determinado objeto ou situação. Dentro do contexto do MAT, a atitude representa o desejo do usuário de utilizar o sistema.

E influenciada pela utilidade e pela atitude em relação ao sistema, está a intenção de uso. Entendemos **intenção** como propósito, plano. É a intenção que determinará o uso real do sistema. Segundo Mathieson (2001), diversos testes empíricos sugerem que o MAT efetivamente prevê e possibilita melhor compreensão das intenções do uso real de sistemas de informação. Para isso, cita o estudo de Davis (1989), que mostra que a facilidade e a

utilidade são significativamente correlacionados para o uso de um pacote de automação de escritórios, e alguns outros que previram com sucesso o uso de pacotes de software.

Todos os construtos do MAT se referem a percepções e crenças de um indivíduo tomando decisões de adoção de tecnologia num determinado período de tempo. Nosso foco, nesse caso não é a realidade em si, mas sim as interpretações que a pessoa faz da realidade. No entanto as pessoas, devido a percepções incorretas, podem tomar decisões que as levem a optar por determinados caminhos que influenciem sua visão de utilidade, facilidade ou de recursos. A percepção atual pode ser decisiva para a percepção futura. Embora essas questões sejam importantes e relevantes, o MAT não leva em consideração variações temporais. As variáveis do modelo são medidas somente uma vez, e refletem as percepções naquele ponto do tempo.

Uma das maiores limitações do MAT é que ele assume que não há barreiras que possam impedir as pessoas de utilizar um sistema de informação. No entanto, o acesso dos usuários a recursos necessários pode afetar o uso (Taylor; Todd, 1995). Existem várias situações onde o indivíduo quer utilizar um sistema, mas é impedido por uma série de restrições, como por exemplo orçamento, tempo, conhecimento ou suporte (Mathieson, 2001). Como o MAT não leva esses fatores em consideração, uma importante fonte de mudanças no comportamento dos usuário é ignorada. Por isso, Mathieson (2001) propôs uma extensão ao modelo original, acrescentando o construto "recursos percebidos". A representação do novo modelo é apresentada na Figura 3.

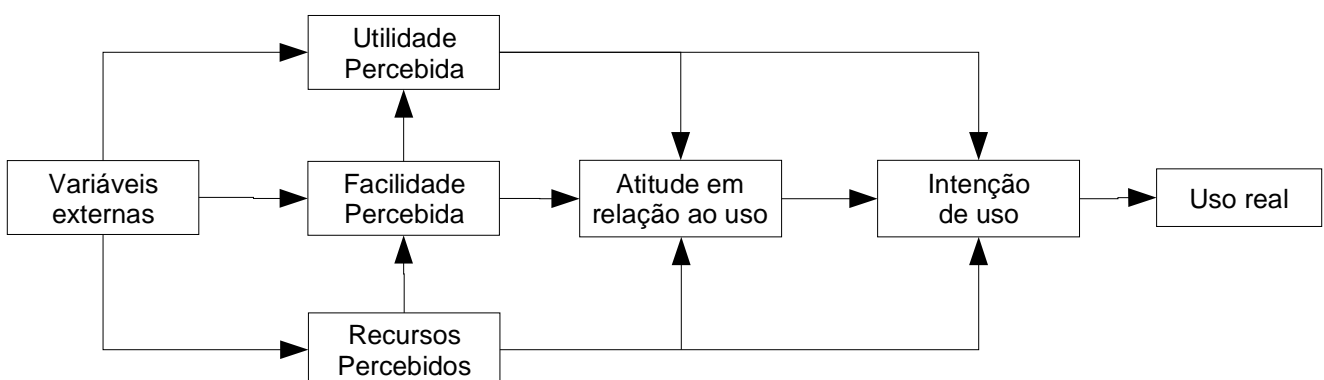


Figura 3 - MAT estendido com "recursos percebidos"

Recurso é aquilo que os usuários acreditam ser necessário estar disponível para que o sistema possa ser implementado e utilizado. Em termos de habilidades, hardware, software,

dinheiro, documentação, dados, suporte e tempo. Os recursos são, então, não somente atributos do sistema, mas de todo o ambiente de trabalho do usuário.

Mathieson (2001) acrescenta que apesar de existirem outros modelos não diretamente desenvolvidos para tecnologia da informação (como o theory of planned behavior - TPB), e que levam em consideração a questão dos recursos, decidiu estender o MAT para "preservar sua simplicidade e manter o foco na Tecnologia da Informação".

Colocando o foco nos recursos, os pesquisadores podem delinear melhor os fatores sobre os quais os gestores de TI podem ter algum tipo de controle. As variáveis relacionadas com recursos foram estudadas por vários autores, sendo que Mathieson(2001) as agrupa em quatro categorias:

1. **Atributos de usuários.** São as características do indivíduo que toma a decisão de uso. A maioria das variáveis reflete algum aspecto de experiência no uso do sistema. As outras são relacionadas com situações individuais, como o tempo disponível para fazer ou aprender uma tarefa.
2. **Suporte.** O nível de suporte disponível, principalmente por parte do pessoal interno à empresa. Refere-se principalmente à disponibilidade de algum tipo de *help-desk*. Nesse sentido, a ajuda de outra pessoa é entendida como um recurso disponível.
3. **Atributos do sistema.** As características do sistema em si, como disponibilidade, documentação, custo de implementação e manutenção.
4. **Controle.** Referem-se às crenças dos usuários a respeito do controle que têm sobre o uso do sistema.

Através desses parâmetros é possível entender, portanto, a noção de risco que o gestor de TI tem a respeito do sistema a ser adotado. Se não acredita que o novo sistema disponha de características fundamentais, como nível aceitável de suporte, adequação técnica ou possibilidade de controle, mais deve acreditar na possibilidade de ocorrência de problemas, e mais seriamente deve considerar que os efeitos desses problemas podem ser potencialmente desastrosos.

Como o modelo é comportamental, pode referir-se somente a questões diretamente relacionadas com o usuário e suas percepções sobre o uso do sistema. Por isso, os construtos

devem ser desenvolvidos de modo a captar opiniões pessoais, e tratar suposições a respeito de terceiros, como outras pessoas ou instituições também como percepções pessoais.

O MAT, apesar de ser aplicado prioritariamente em pesquisas de fundo quantitativo, pode perfeitamente ser utilizado como base para uma pesquisa de metodologia qualitativa. Isso é corroborado pela análise da metodologia utilizada por Davis (1985) e por Mathiesson (2001) no desenvolvimento dos parâmetros utilizados em seus trabalhos, e por estudos como o de Benamati (2002), que faz uso do MAT como base para analisar a aceitação de processos de terceirização de tecnologia, utilizando metodologia qualitativa aliada a técnicas de análise de conteúdo.

4.1. OS CONSTRUTOS

Baseado nos conceitos apresentados, são operacionalizados aqui os construtos necessários à pesquisa. Os construtos "utilidade" e "facilidade" serão implementados tomando por base as escalas criadas por Davis (1985 e 1989) e validadas por Doll (1998). O construto "recursos" será implementado utilizando-se a escala desenvolvida por Mathieson (2001).

Nas próximas páginas estão os construtos, com as variáveis, as proposições a eles relacionadas e a bibliografia de referência.

Quadro 1 - O construto "recursos"

Recursos		
Variável	Proposição	Referências
Atributos de usuários		
Necessidade de treinamento do usuário	P1j	Kenwood (2000)
Necessidade de treinamento da equipe de informática	P1b	Dissel (2000)
Suporte		
Disponibilidade de recursos humanos para implementação	P2d	Kenwood (2001) e Castellucio (2000)
Disponibilidade de suporte técnico na operação	P2d	Kenwood (2001) e Castellucio (2000)
Atributos do sistema		
Participação de mercado	P1	Ozsomer (2000) e Meirelles (2003)
Incompatibilidades técnicas com arquivos e padrões utilizados por líderes de mercado	P1h	Ozsomer (2000)
Custo de migração	P1a	Dissel (2000) e Taudes (2000)
Tempo de migração	P1d	Dissel (2000)
Problemas técnicos de migração	P1e	Dissel (2000)
Mensuração da viabilidade do investimento	P1c	Taudes (2000)
Funcionalidades implementáveis em estações	P1i	Faldetta(2002) Raymond (2001)
Funcionalidades implementáveis em servidores	P2, P2a	Meirelles (2003), IDC
Disponibilidade de software	P1i	Faldetta(2002) Raymond (2001) Mathieson (2001)
Controle		
Falta de propriedade (produtos "sem dono")	P2b	Kenwood (2001) e Castellucio (2000)
Conhecimentos do gestor e da equipe de TI para controle do ambiente	P1b	Dissel (2000)

Quadro 2 - O construto "utilidade"

Utilidade		
Variável	Proposição	Referências
Utilidade para o trabalho de administração de TI	P3	Wheeler (2002)
Velocidade de execução de tarefas		Davis (1989)
Produtividade nas atividade necessárias		Davis (1989)
Eficiência ao alcançar os objetivos		Davis (1989)
Possibilidade de facilitar a execução de tarefas		Davis (1989)
Utilidade para o trabalho dos usuários	P3	Wheeler (2002)
Velocidade de execução de tarefas		Davis (1989)
Produtividade nas atividades necessárias		Davis (1989)
Eficiência ao alcançar os objetivos		Davis (1989)
Possibilidade de facilitar a execução de tarefas		Davis (1989)

Quadro 3- O construto "facilidade"

Facilidade		
Variável	Proposição	Referências
Facilidade de uso para a administração de TI	P1b	Dissel (2000)
Facilidade de aprendizado		Davis (1989)
Recurso facilmente controlável		Davis (1989)
Claro e de fácil entendimento		Davis (1989)
Flexível, podendo atingir necessidades específicas		Davis (1989)
Fácil de usar		Davis (1989)
Facilidade de uso para os usuários	P1j	Kenwood (2000)
Facilidade de aprendizado		Davis (1989)
Recurso facilmente controlável		Davis (1989)
Claro e de fácil entendimento		Davis (1989)
Flexível, podendo atingir necessidades específicas		Davis (1989)
Fácil de usar		Davis (1989)

Quadro 4 - Atitude, como resultado de "facilidade", "utilidade" e "recursos"

Atitude	
Variável	Proposições
Resistência à mudança (Mingji, 2002)	
Pressões externas, no mercado	P1, P1h
Dificuldade na migração	P1a,b,c,d,e
Necessidade de trabalho político para migração	P1e,f
Dúvidas a respeito da capacidade das soluções em software livre suportarem aplicações de negócio	P2b
Descrédito quanto à possibilidade de uso para aplicações de negócios	P2c
Descrédito quanto ao uso em larga escala pelo mercado em curto e médio prazo	P2b

Quadro 5 - Intenção de uso, como resultado da atitude do gestor de TI

Intenção	
Variável	
Possibilidade de uso de software livre	
Serviços onde há viabilidade	
Serviços onde não há viabilidade	
Intenções de uso futuro	

Quadro 6 - Uso real, como resultado da intenção de uso

Uso real	
Levantamento do uso atual de software livre na empresa	
Avaliação dos resultados obtidos	

5. METODOLOGIA

Os objetivos especificados nas seções anteriores e a verificação das proposições levantadas foram realizados através de pesquisa de campo, onde foram coletadas as informações para análise. O objetivo principal do trabalho é conhecer o panorama do software livre nas empresas, e entender as razões de sua baixa utilização. Uma vez que baseamo-nos em um fato explicitado por um estudo quantitativo (Meirelles, 2003), procuramos compreender o fenômeno apresentado baseando-nos na percepção das pessoas envolvidas nesse processo, através de suas opiniões, visões e conceitos. Por isso a utilização de metodologia qualitativa. Ela permite captar essas nuances, propicia uma visão global e traz à tona as questões pertinentes ao tema. São estudadas então, as opiniões dos atores do processo: usuários de software livre, seus vendedores e concorrentes.

A captação das opiniões das pessoas sobre o software livre foi feita através de entrevistas individuais em profundidade. Nessas entrevistas foi possível conhecer o pensamento dos envolvidos e os detalhes sobre suas percepções acerca dos assuntos tratados. Elas tiveram um caráter não diretivo, para que o entrevistado se sentisse à vontade para falar e revelasse aspectos não determinados previamente pelo pesquisador. Embora não diretivas, essas entrevistas foram semi-estruturadas, com um roteiro baseado nos construtos desenvolvidos. A função desse roteiro não foi restringir os assuntos a serem abordados, mas sim servir como guia, evitando que assuntos importantes não fossem abordados e facilitando a construção de um panorama geral da situação.

Com o objetivo de limitar o universo de investigação a uma base de mensuração possível, nas entrevistas foram referenciadas apenas as categorias de software listadas abaixo, correspondentes às categorias de produtos utilizadas na Pesquisa anual da FGV (Meirelles, 2003)³⁵:

- Pacote integrado de escritório (processador de textos, planilha eletrônica, apresentações);
- Banco de dados utilizado na estação;
- Banco de dados corporativo;

³⁵No anexo, à página 124, há uma relação de softwares livres categorizados

- Sistema operacional no servidor;
- Sistema operacional na estação;
- Linguagem básica de programação;
- Editoração eletrônica;
- Correio eletrônico;
- Navegador de Internet;
- Gráfico técnico – CAD;
- Groupware;
- Antivírus.

A amostra se constituiu de empresas usuárias de software (com atividade fim não relacionada à informática), de empresa de informática que trabalha com software livre, de uma empresa grande usuária de software livre, e de uma associação de empresas de informática que trabalham com software proprietário. As pessoas que fizeram parte da amostra foram selecionadas com base em sua representatividade dentro desses estratos, definida por alguns critérios arbitrários. Segundo o descrito por Malhotra (2001), a amostra foi não probabilística, numa combinação da amostragem por julgamento com a por cotas. Abaixo estão os critérios utilizados para determinar que a pessoa entrevistada fizesse parte dos grupos descritos:

- **Usuários de software.** Trabalhar numa empresa que não seja da área de informática, na área de tecnologia de informação e num cargo que pressuponha a decisão ou influência decisiva na recomendação de compra de produtos e serviços computacionais. Os entrevistados deveriam conhecer a diferença entre software livre e software proprietário, e ter conhecimento da existência de algum software livre. Foram realizadas cinco entrevistas nesse grupo. Para chegar a essa quantidade, foi utilizado o mesmo critério utilizado por Reis (1994): foram feitas entrevistas sucessivas até que novas entrevistas não acrescentassem mais informações adicionais. Segundo Meirelles (2003), existe uma significativa diferença na forma de gastos com informática em função da quantidade de teclados existentes na empresa. Sua pesquisa agrupa as empresas em três estratos (de 30 a 160, de 160 a 550 e acima de 550 teclados), e aponta uma forte

relação entre o porte e o índice de gastos. Isso nos leva a crer que podem existir diferenças significativas na forma e nos critérios de decisão de adoção de software, e conseqüentemente nas opiniões expressas pelos gestores de informática. Por isso, a amostra de usuários de software foi limitada a apenas um desse estratos: empresas que possuíssem de 30 a 160 teclados.

- **Empresa de software livre.** Foi entrevistado o diretor geral de uma empresa que comercializa e fornece suporte técnico prioritariamente a software livre.
- **Associação de empresas de software proprietário.** Inicialmente a idéia foi entrevistar uma pessoa que trabalhasse numa empresa que produzisse, representasse ou comercializasse prioritariamente software proprietário. No entanto, considerou-se mais produtivo procurar uma associação que representasse essas empresas, de forma que fosse possível ter uma visão consolidada da posição geral dos atores desse mercado. Foi, então, entrevistado o presidente de uma associação que teve destaque no ano de 2003 na defesa dos interesses dos produtores de software proprietário.
- **Grande empresa usuária.** Foi entrevistada uma empresa grande usuária de software livre, cujo porte excede o limite de 160 teclados da amostra inicial. O objetivo foi conhecer os motivos que os levaram a optar por essa alternativa, e conhecer os resultados do processo.

As entrevistas foram gravadas, transcritas, e posteriormente analisadas utilizando-se as escalas do MAT como guia. As escalas utilizadas foram comparadas aos resultados, e foram propostas medidas específicas para a realidade do software livre, que poderão ser validadas em eventual estudo futuro.

Como essa amostra é não probabilística, e tomada por julgamento, ela não é representativa do universo de empresas usuárias de software livre. Não será possível, portanto, generalizar as conclusões aqui obtidas. No entanto, a qualidade proporcionada pela metodologia utilizada suplanta essa deficiência. Não é possível generalizar, mas os resultados da pesquisa trazem à tona bons indicadores.

5.1.ROTEIRO PARA AS ENTREVISTAS

Abaixo está o roteiro seguido nas entrevistas. Estão enumerados os tópicos abordados, sem a formulação de perguntas. Como as entrevistas foram não diretivas, os tópicos descritos não foram tratados nessa ordem ou exatamente nesses termos, nem foram feitas tantas perguntas quanto o número de itens abaixo. A intenção foi abordar os assuntos descritos; na ordem e da forma que o entrevistado se sentisse mais à vontade para expressar suas impressões.

O roteiro foi o mesmo para todos, sendo que a diferença foi o enfoque. Enquanto nas empresas usuárias o foco foi o uso próprio, nas outras duas o foco foi o uso por parte de seus clientes.

1. Apresentação

1.1. Apresentação do estudo, de seu objeto, de seu propósito e objetivos.

2. Detalhamento dos construtos

2.1. Recursos

2.1.1. O que sabe sobre o software livre.

2.1.2. Quais os produtos que conhece, e sabe estarem disponíveis para uso.

2.1.3. Que tipo de funcionalidades crê serem possíveis implementar em estações de trabalho.

2.1.4. Que tipo de funcionalidades crê serem possíveis implementar em servidores.

2.1.5. Se acredita que o uso de softwares livres pode diminuir o custo total da informática na empresa.

2.1.6. Como acredita serem compatíveis os arquivos e formatos produzidos em soluções de software livre em relação aos padrões “de fato” do mercado.

2.1.7. Opinião a respeito do efeito da participação de mercado dos programas livres em relação à possibilidade de uso.

2.1.8. Se acredita estarem disponíveis recursos humanos (funcionários ou fornecedores de serviços) para implementação das soluções.

2.1.9. Se acredita estar disponível suporte técnico para os produtos implementados.

2.1.10. Avaliação da necessidade de treinamento da equipe de informática e do usuário final, e quão custoso isso seria.

- 2.1.11. Qual sua opinião sobre o fato dos produtos de software livre não serem propriedade de uma empresa constituída, responsável por seu desenvolvimento e suporte.
- 2.1.12. Se acredita que o uso do software livre pode influenciar (positivamente ou negativamente) o nível de controle da equipe de TI no ambiente existente.
- 2.1.13. Visão a respeito da eventual migração de servidores e estações de trabalho para soluções de software livre, no que diz respeito a custos, tempo, possíveis problemas técnicos e impacto no trabalho do usuário.
- 2.1.14. Como vê a possibilidade de retorno do investimento na implantação do software livre e sobre como mensurar esse retorno.

2.2. Utilidade

2.2.1. Para o trabalho diário da equipe de TI, a possibilidade de que a adoção de soluções em software livre afetem a execução das tarefas em termos de:

- velocidade;
- produtividade;
- eficiência;
- e facilidade.

2.2.2. Para o trabalho diário dos usuários, a possibilidade de que a adoção de soluções em software livre afete a execução das tarefas em termos de:

- velocidade;
- produtividade (incluindo-se expectativas de downtime e mau funcionamento de software);
- eficiência;
- e facilidade.

2.3. Facilidade

2.3.1. Possibilidade que a adoção de produtos de software livre afetem a facilidade de uso dos sistemas para os usuários e depois para a equipe de TI, nos seguintes termos:

- facilidade de aprendizado;
- facilidade de controle do recurso;
- clareza e entendimento;
- flexibilidade para atingir necessidades específicas;
- facilidade de uso.

2.4. Atitude

- 2.4.1. Opinião a respeito da adoção em larga escala de soluções de software livre pelo mercado, e como isso afetaria as decisões internas.
- 2.4.2. Capacidade de soluções em software livre para suportar aplicações de negócio a curto e a médio prazo.
- 2.4.3. Possibilidade de apoio político na organização para uso do software livre.
- 2.4.4. Atitude geral em relação à adoção de software livre (aceitação ou resistência total ou parcial)
- 2.5. Intenção
 - 2.5.1. Intenção de uso presente e futuro, com detalhamento dos serviços onde o entrevistado enxerga e onde não enxerga viabilidade .
- 2.6. Uso real
 - 2.6.1. Levantamento do uso atual de software livre na empresa.

6. RESULTADOS

6.1. EMPRESAS USUÁRIAS DE SOFTWARE

Como exposto na seção anterior, o principal estrato pesquisado foi o das empresas usuárias de software. Abaixo estão os critérios utilizados para que a empresa fizesse parte da amostra:

1. A quantidade de teclados deveria ser necessariamente maior que 30, e menor que 160. Ao contrário do que se possa pensar, essas empresas não são necessariamente pequenas. Segundo o IBGE, empresas pequenas são aquelas cuja receita operacional bruta é inferior a um milhão e duzentos mil reais anuais. Uma das empresas entrevistadas, por exemplo, tem 600 funcionários, e faturamento bem maior que esse valor. Procuramos chamar essas empresas, então, de empresas com pequena estrutura de TI.
2. A empresa deveria ter, necessariamente, uma pessoa responsável pela gestão da tecnologia da informação com poder de decisão ou influência decisiva nos processos de adoção de software.
3. O processo de decisão de adoção de tecnologia da informação deveria, necessariamente, ser realizado internamente, sem a imposição direta de fornecedores, clientes ou da matriz.

O critério para determinar a quantidade de empresas foi a realização de entrevistas sucessivas até que outras adicionais não acrescentassem mais informações significativas. Foram visitadas, então, seis empresas. Uma delas foi descartada por ter seu ambiente de informática estruturado com produtos e tecnologias impostos pela matriz. Foram consideradas válidas, então, cinco entrevistas, que se enquadram totalmente nos três critérios estabelecidos para sua inclusão na amostra. As empresas entrevistadas estão listadas a seguir, identificadas por seu ramo de atuação:

- Comunicação sem fio. Trabalha com o desenvolvimento e implementação de soluções de comunicação sem fio, em projetos desenvolvidos principalmente nas áreas de manufatura, automação de força de vendas, varejo e logística. Considera que para atender a um cliente, é necessário que este entenda a solução de comunicação sem fio como estratégica à seu negócio, e procura selecioná-los de acordo com esse perfil. Hoje

tem como clientes diversas empresas de grande porte, e é uma das mais importantes em seu segmento. Tem 30 funcionários, e a estrutura de informática conta a mesma quantidade de teclados. Foi entrevistado um dos sócios, que é o responsável pelo uso da tecnologia da informação na empresa.

- Indústria química. Produz materiais que têm diversos tipos de aplicações, que vão desde embalagens para alimentos até uso na construção civil. Essa empresa atualmente é, em volume de vendas dentro de seu segmento, a segunda colocada no mercado brasileiro, contando com planos para tornar-se a primeira no ano de 2004. É dividida em duas unidades produtivas, que somam 240 funcionários. A estrutura de informática tem 45 teclados, e foi entrevistado o responsável pela tecnologia da informação.
- Indústria metalúrgica. Atua basicamente nos ramos de móveis de escritório e sistemas de armazenagem, principalmente para clientes que atuam no varejo e no ramo de logística. A empresa tem posição de destaque nos dois segmentos em que atua, e é reconhecida como uma das mais importantes do setor. Está presente em dois endereços: a fábrica, onde está toda a produção e a área administrativa; e um escritório de vendas. Somando-se as duas unidades, tem hoje cerca de 200 funcionários, e conta com 50 teclados. Foi entrevistado o responsável pela tecnologia da informação.
- Indústria têxtil. Empresa do ramo têxtil que produz fios, tecidos e malhas. A atividade principal da empresa é a produção de um tipo específico de fio de algodão, segmento no qual é líder absoluto de mercado, atendendo a clientes no Brasil inteiro e no exterior. Tem duas fábricas, nas quais trabalham cerca de 600 funcionários. A estrutura de informática conta com 65 teclados, distribuídos entre as duas unidades fabris. Foi entrevistado o responsável pela tecnologia da informação.
- Distribuição de bens de consumo. Trabalha com distribuição de bens de consumo em geral, com destaque para fraldas e uma linha especial de bebidas alcoólicas importadas. Está presente em todo o Brasil, através de uma rede de 300 representantes, com estrutura logística pronta para atender a seus pedidos. A empresa em si, além dos representantes, tem cerca de 100 funcionários, e conta com uma estrutura que tem 83 teclados, entre o escritório da matriz e a filial. Foi entrevistado o responsável pela tecnologia da informação.

6.1.1.DESCRICÃO DOS CONSTRUTOS

Nessa seção estão descritas as entrevistas, divididas de acordo com os construtos do MAT, explicados nos capítulos anteriores.

6.1.1.1.RECURSOS

Atributos de usuários

- Necessidades de treinamento de usuários

Proposição P1j (pág. 48): as soluções em software livre seriam consideradas, pelos gestores de TI, como mais difíceis de serem utilizadas por usuários leigos em informática. Isso implicaria em necessidades e custos de treinamento.

Verificou-se que todos os entrevistados realmente acreditam na necessidade de treinamento de seus usuários, e enfatizam sua importância. No entanto, excetuando-se um dos entrevistados, que duvida da capacidade dos colaboradores de sua empresa em assimilar qualquer tipo de treinamento, os demais entrevistados afirmaram que apesar de necessário, o treinamento seria facilmente ministrado, e que os usuários não teriam dificuldades de assimilar as informações. Consideram que o treinamento seria curto, uma vez que as aplicações em software livre consideradas no escopo da pesquisa têm interface parecida com as alternativas atualmente líderes de mercado. O custo de treinamento do usuário final, não seria, portanto um empecilho para a adoção.

- Necessidade de treinamento do pessoal de informática.

Proposição P1b (pág. 46): a utilização do software livre nas estações de trabalho seria considerada, pelos gestores de TI, inconveniente devido à necessidade de treinamento da equipe de informática.

Todos os entrevistados consideram que o treinamento da equipe de informática é importante e necessário. No entanto, enfatizaram que embora o uso do software livre (especificamente os sistemas operacionais baseados no GNU/Linux) requeira uma nova capacitação e a aquisição de novos conhecimentos, os profissionais de informática – eles próprios e sua equipe – são “motivados”, “interessados”, “abertos” e “capazes”, e não teriam dificuldades em assimilar a administração de um ambiente que utilizasse majoritariamente software livre.

Suporte

- Disponibilidade de recursos humanos para implementação

Proposição P2d (pág. 50): Existiriam dúvidas, por parte dos gestores de TI, sobre a disponibilidade de suporte para **implementação** de sistemas utilizando software livre.

Excetuando-se um dos entrevistados, que afirmou que não há pessoas habilitadas a trabalhar com software livre, os demais afirmaram que a quantidade de pessoas habilitadas a fazer a implementação dos sistemas ainda é menor que as habilitadas a trabalhar com produtos proprietário. No entanto, afirmaram que encontrar essas pessoas é fácil, principalmente junto a fornecedores de serviços de informática. Dois deles afirmaram também que esses profissionais seriam mais caros que o usualmente praticado no mercado, mas que em compensação, teriam maior capacidade técnica. Exceto o entrevistado que afirmou que não existe suporte, os demais afirmaram que apesar dessas características, mais e mais empresas estão se capacitando para prestar suporte a software livre, o que pode ser verificado na quantidade crescente de anúncios e de novas empresas comunicando essa modalidade de trabalho. Enfatizam, portanto, que embora haja menos suporte, este pode ser encontrado, e que não é um empecilho para a utilização de software livre em maior escala.

- Disponibilidade de suporte técnico na operação

Proposição P2d (pág. 50): Existiriam dúvidas, por parte dos gestores de TI, sobre a disponibilidade de suporte para a **operação** de sistemas em software livre após sua implementação.

Excetuando-se um dos entrevistados, que acredita que não existe qualquer tipo de suporte técnico para software livre, os demais concordaram que o suporte técnico para a operação do sistema após a implementação existe, mas não é tão fácil de ser encontrado e de ser acionado quando comparado à disponível para produtos proprietários. É possível perceber, nesse ítem, divergências entre as percepções dos entrevistados, e em relação às respostas do ítem anterior. Além daquele entrevistado que afirmou que não existe suporte; um colocou que o suporte existe, mas é insuficiente para atender às necessidades do mercado; dois disseram que existe e está disponível, mas têm dúvidas quanto a sua qualidade, pois ainda não precisaram utilizá-lo; e um afirmou que há uma quantidade pequena de empresas que prestam esse suporte, mas que está melhorando a cada dia. Três dos entrevistados criticaram duramente o modelo de suporte que têm hoje por parte de empresas de software proprietário,

especificamente a Microsoft. Sentem-se abandonados por essa empresa, e dependem exclusivamente do suporte técnico oferecido por seus prestadores de serviços em informática. Dois deles enfatizaram também que mesmo esses prestadores de serviço, quando necessitam de suporte direto da empresa fornecedora, não o conseguem. Além disso, criticaram duramente a política de atualizações e a quantidade de problemas de software que têm que corrigir, falando dos produtos da Microsoft. Um entrevistado relatou problemas encontrados com a compra de produtos OEM³⁶. A empresa fornecedora de hardware vende o software pré instalado, e pelo contrato OEM, seria a responsável por seu suporte. Na prática isso não ocorre, porque os fornecedores de hardware “não são capazes de prestar esse suporte”, e o acesso aos recursos oficiais da Microsoft não é possível porque trata-se de um produto OEM. É possível notar que o suporte técnico é um ponto nevrálgico nas empresas entrevistadas e que todas têm vivido agruras, visto que espontaneamente passaram a criticar a Microsoft, e a mostrar descrédito com as alternativas por ela oferecidas. De uma forma geral, os gestores de TI demonstram querer confiar num fornecedor externo de suporte – não necessariamente o desenvolvedor do software – que esteja próximo do dia a dia e da realidade da empresa, que resolva os problemas e seja o responsável geral pelo sistema. Contam que já têm fornecedores com esse perfil para os sistemas que atualmente utilizam, e demonstram estar dispostos, se necessário, a procurar outros fornecedores e a testar alternativas.

Atributos do sistema

- Participação de mercado

Proposição P1 (pág. 45): A adoção de software livre nas estações de trabalho tem uma forte barreira de entrada, pois o uso do software proprietário perpetua-se de forma inercial devido à ampla base instalada.

A grande participação de mercado dos produtos de software proprietário foi apontada por todos os entrevistados como um razão fundamental para a não adoção de software livre em escala maior, principalmente nas estações de trabalho. Um dos entrevistados afirmou que iniciaria estudos para utilizar software livre nas estações somente se o mercado o adotasse em larga escala, e seus clientes demandassem integrações nesse sentido. Outro colocou que a empresa faz questão de adotar somente o padrão dominante. Todos afirmaram, de uma forma ou de outra, que a atividade fim da empresa não era a informática em si, e que em

³⁶Produtos OEM são aqueles fornecidos em conjunto com um novo hardware. Isso é muito comum no caso de computadores novos, quando o sistema operacional e outros aplicativos já vêm pré instalados.

qualquer movimento de adoção de tecnologia a opção inevitavelmente deve recair sobre os produtos mais comumente utilizados.

- Incompatibilidades técnicas com arquivos e padrões utilizados por líderes de mercado

Proposição P1h (pág. 48): A imperfeição na abertura de arquivos de formato proprietário, que se tornaram padrões "de fato", por softwares livres é uma barreira à adoção.

Três dos entrevistados enfatizaram que a incompatibilidade com arquivos e padrões proprietários existentes no mercado são barreiras intransponíveis para a adoção em larga escala de software livre. Referem-se especificamente aos arquivos de textos, planilhas, figuras e apresentações produzidos em aplicativos proprietários. Um deles cita também a impossibilidade de uso de sistemas operacionais GNU/Linux como clientes de um gerenciador de banco de dados que a empresa possui, devido ao fato do banco ser proprietário e de não existirem drivers para acesso. Duas das empresas já realizaram, inclusive, testes internos de uso do conjunto de escritório StarOffice³⁷, que não foram bem sucedidos devido à incompatibilidade de formatos. Outro entrevistado colocou que sim, existem problemas de abertura de arquivos de formatos proprietários, mas que num ambiente com software livre seria possível conviver com esses problemas, e utilizá-los normalmente. Apenas um dos entrevistados destoa dos demais, afirmando que não há problemas de compatibilidade de formatos.

- Custo de migração

Proposição P1a (pág. 46): A utilização do software livre em todo o ambiente é considerada, pelos gestores de TI, inconveniente devido aos altos custos de migração.

Exceto um dos entrevistados, que não tem intenção de avaliar a possibilidade de migração e afirma não ter nem noção dos aspectos envolvidos, os demais entrevistados consideraram o custo de migração em si como um fator secundário. Disseram que seria necessário investir em treinamento e em consultoria para implementação, mas que os recursos seriam facilmente conseguidos se fosse possível comprovar, nem que fosse simplesmente com uma perspectiva de economia em licenças, a redução de custos ou outros benefícios gerados pelo investimento. A grande questão relacionada à migração não é o custo de sua realização, e sim

³⁷O StarOffice é recorrentemente citado como software livre, embora não o seja. Essa confusão acontece pelo fato desse sistema ter disponível uma versão para uso nos sistemas operacionais baseados no GNU/Linux, e por ter sido distribuído de forma gratuita entre os anos de 1999 e 2002. O sucessor livre do StarOffice é o OpenOffice.org. Veja em <<http://www.openoffice.org>>.

o fato de que o que foi gasto para legalizar o ambiente com o software proprietário ainda não se depreciou, portanto a manutenção do que já existe seria uma forma de preservação do investimento.

- Tempo de migração

Proposição P1d (pág. 46): A utilização do software livre de forma mais intensa é considerada, pelos gestores de TI, inconveniente devido à grande quantidade de tempo necessário à migração.

Exceto um dos entrevistados, que não pretende avaliar o processo de migração, os demais afirmaram que não acreditam em migrações longas e penosas. Enfatizam que os produtos em software livre possuem boas qualidades técnicas, e que o processo de migração deve ser simples e rápido. Um dos entrevistados, ao contrário dos demais, acredita que o processo deve ser propositalmente longo, para que os usuários tenham menos impacto na mudança. Mas mesmo assim, não acredita num processo de migração difícil.

- Problemas técnicos na migração

Proposição P1e (pág. 46): A utilização do software livre nas estações de trabalho é considerada, pelos gestores de TI, inconveniente devido à possibilidade de problemas técnicos.

Exceto um dos entrevistados, que não admite avaliar o processo de migração, os demais entrevistados não acreditam em dificuldades ou problemas técnicos de migração.

- Mensuração da viabilidade de investimento

Proposição P1c (pág. 46): A utilização do software livre nas estações de trabalho é considerada, pelos gestores de TI, inconveniente devido à dificuldade em mensurar a viabilidade do investimento.

Nenhum dos entrevistados considera ser difícil mensurar a viabilidade de investimento, e tratam a questão sob duas perspectivas:

1. Custo de licença. Para todos os entrevistados, a principal questão quanto ao software livre é a gratuidade. Todos ressaltam, em primeiro lugar, que para utilizar software livre não é necessário pagar licenças. As questões de disponibilidade de código fonte, de livre

distribuição e de livre modificação são totalmente secundárias. Um dos entrevistados definiu software livre somente como software grátis. Todos afirmaram, então, que a adoção em maior escala do software livre diminuiria os custos com licenciamento. No entanto, existe uma posição diferenciada entre as empresas que têm seu parque totalmente legalizado daquelas que não têm. Três empresas, que têm todas licenças necessárias, se mostram cautelosas, e declaram que uma migração não seria vantajosa, pois precisam preservar o investimento já realizado e não poderiam usufruir da maior vantagem da adoção de software livre, que é a economia com licenciamento. Uma outra empresa, que não tem suas licenças totalmente regularizadas, afirma que o investimento em treinamento e em serviços de implementação seriam plenamente compensados pela economia com licenças. E a empresa restante, embora acredite que realmente haveria economia em licenciamento, considera que não valeria a pena ter um ambiente fora do padrão aceito pelo mercado somente por esse motivo. Nenhum dos entrevistados expressou preocupações com os custos de manutenção do ambiente após a implementação, que acreditam ser equivalentes aos custos de manutenção já existentes.

2. Custo de atualização de hardware. Outro fator citado como fonte de economia é a não necessidade de troca constante de hardware. Todos os entrevistados afirmaram que os produtos de software livre podem ser executados de forma eficiente com hardware inferior ao atualmente requerido pelos produtos líderes de mercado. Um dos entrevistados colocou que a cada dois anos é obrigado a iniciar um processo de atualização geral nas estações, e que isso não seria necessário se utilizasse software livre.

De modo geral os entrevistados afirmaram que, baseados na economia com licenças de software e com substituições de hardware, seria possível sem grandes dificuldades justificar o investimento na migração. Mas, ao mesmo tempo, não há interesse em realizá-la, porque aqueles que já investiram em licenças precisam preservar seu investimento, e todos, de modo geral, temem utilizar um padrão minoritário no mercado e ter problemas com externalidades de rede. Somente um dos entrevistados afirmou que pretende iniciar um projeto de adoção em larga escala de software livre, com justificativas baseadas nos valores necessários à legalização do parque existente.

- Funcionalidades implementáveis nas estações

Proposição P1i (pág. 48): O desconhecimento dos produtos e das funcionalidades oferecidas por ferramentas de software livre é uma barreira para sua adoção.

Todos os entrevistados expuseram que, apesar de não conhecerem exatamente os nomes dos produtos, sabem que com software livre é possível ter uma estação de trabalho totalmente funcional. Foram citados como disponíveis para uso: sistema operacional, bancos de dados, planilhas, editores de texto, navegadores de Internet, clientes de acesso a sistemas ERP, programas de edição gráfica, de reprodução e edição de som e programas para acesso à rede corporativa.

- Funcionalidades implementáveis em servidores

Proposições P2 e P2a (pág. 49): Os gestores de TI vêm o software livre como alternativa viável para determinados serviços em servidores corporativos, e têm intenção de utilizá-lo apenas em parte de seu ambiente. O rápido crescimento do uso do software livre nos servidores nos últimos anos (2000, 2001 e 2002) tem um forte componente da adoção das tecnologias da Internet, e não de migração de sistemas existentes.

Todos os entrevistados afirmaram que existem soluções em software livre para atender praticamente a qualquer necessidade de aplicações para servidores, principalmente às necessárias aos negócios e de missão crítica. Todos citaram como importantes as funcionalidades de bancos de dados e serviços para Internet. Um dos entrevistados ressaltou a disponibilidade para uso em mainframe, e colocou que grandes usuários de TI, inclusive bancos, estão migrando para sistemas operacionais GNU/Linux.

Ou seja, nenhum dos entrevistados corroborou as proposições. No entanto, embora os gestores de TI acreditem que o software livre possa ser utilizado para quaisquer situações, o uso constatado nas empresas entrevistadas concentra-se realmente em aplicações de Internet.

- Disponibilidade de software

Proposição P1i (pág. 48): O desconhecimento dos produtos e das funcionalidades oferecidas por ferramentas de software livre é uma barreira para sua adoção.

Todos os entrevistados demonstraram conhecer várias funcionalidades possíveis em software livre, tanto para servidores quanto para estações de trabalho. No entanto existe uma certa confusão entre softwares livres e proprietários. Os entrevistados, de um modo geral, consideram que qualquer produto que tenha sua execução possível nos sistemas operacionais GNU/Linux, sejam software livre. O StarOffice é sempre apontado como software livre, embora

não o seja. O mesmo aconteceu com os compiladores Clipper e Cobol, que não têm versões livres prontas para produção, com o BRMultiaccess, que é um software proprietário para acesso à Internet e com o gerenciador de bancos de dados Oracle. Outra confusão é em relação aos sistemas operacionais, onde todos se referem ao Linux como um sistema operacional completo, e desconhecem que este seja somente o nome de seu núcleo.

Foram citados os seguintes nomes de produtos³⁸: Linux, Conectiva Linux (sistema operacional), OpenOffice.org (conjunto para escritório), Apache (servidor de páginas para Internet), Qmail (servidor de correio), Sendmail (servidor de correio), MySQL (gerenciador de bancos de dados) e Mozilla (navegador de Internet).

Controle

- Falta de propriedade (produtos "sem dono")

Proposição P2b (pág. 50): Os gestores de TI não acreditam que os softwares livres sejam capazes de suportar as aplicações de negócio especificamente por não serem propriedade de uma empresa.

Exceto um dos entrevistados, que não tem opinião sobre o assunto, os demais afirmaram que o fato dos produtos em software livre não terem um dono específico não representa um problema. Todos contam com a parceria de um fornecedor de serviços de informática e, cobram deles todo o suporte e a responsabilidade pelas implementações. Um dos entrevistados afirmou que uma vantagem de produtos "com dono" é que certamente num software proprietário seria mais fácil responsabilizar o desenvolvedor por eventuais prejuízos. No entanto, não classifica esse benefício como importante, uma vez que um processo judicial é custoso e demorado, e pode resultar numa indenização de até o valor pago pela licença. Outro entrevistado, citando especificamente a Microsoft, colocou que de qualquer forma se sente abandonado, que seu suporte depende exclusivamente do prestador de serviços, e que a falta de propriedade para ele não faria qualquer diferença prática.

- Conhecimentos do gestor e da equipe de TI para controle do ambiente

Proposição P1b (pág 46): a utilização do software livre nas estações de trabalho seria considerada, pelos gestores de TI, inconveniente devido à necessidade de treinamento da equipe de informática.

³⁸No anexo, à página 124, há uma lista categorizada de softwares livres.

Como o exposto anteriormente, os entrevistados acreditam que o treinamento e acultramento dos gestores de TI e da equipe de informática seriam necessários, mas que por serem profissionais de informática, capazes e interessados no assunto, isso não chegaria a ser um problema.

Falando especificamente da questão do controle, dois dos entrevistados acreditam que com uma estrutura prioritariamente em software livre, seria mais fácil lidar com o ambiente, pois as estações de trabalho poderiam ser melhor gerenciadas, estariam menos sujeitas a alterações feitas por usuários e apresentariam melhor estabilidade no trabalho diário. Os mesmos dois entrevistados criticaram enfaticamente o sistema operacional Windows, por ser instável e por não permitir o controle do ambiente da forma que gostariam.

6.1.1.2.UUTILIDADE

- Utilidade para o trabalho de administração de TI

Proposição P3 (pág. 50): Os gestores de TI reconhecem que os produtos de software livre, de modo geral, têm qualidades técnicas comparáveis ou superiores aos produtos de software proprietário.

De um modo geral, todos os entrevistados consideram que os produtos em software livre, dentro do escopo abordado, têm qualidades técnicas superiores às encontradas em similares proprietários. Consideram que os produtos listados são úteis e totalmente necessários ao trabalho diário de uma empresa. No entanto, um dos entrevistados, apesar de acreditar que os programas livres sejam bons, não tem a intenção de utilizá-los, na certeza de que eles não se sustentarão ao longo do tempo por não cobrarem por licenças.

Os pontos específicos abordados foram os seguintes:

1. Velocidade na execução de tarefas.

Os entrevistados consideram que com o software livre é necessário menos hardware para executar as mesmas tarefas. Portanto, o uso de software livre possibilitaria economia de hardware e maior velocidade nas tarefas diárias de TI.

2. Produtividade nas atividades necessárias

Os entrevistados consideram que os softwares livres são mais estáveis e demandam menos suporte técnico que as alternativas proprietárias. Isso aumentaria a produtividade do departamento de informática, uma vez que haveria menor necessidade de atender a chamados de usuários e liberaria a equipe de informática para outras tarefas. Dois deles ressaltaram que os servidores de acesso à Internet baseados em software livre que utilizam são muito estáveis, e que desde a implantação foram parados somente para manutenções programadas.

3. Execução de tarefas e consecução dos objetivos

Excetuando-se um dos entrevistados, que não soube responder à questão, os demais afirmaram que o uso de software livre é mais eficiente na possibilidade de alcançar os objetivos do departamento de informática e na facilitação de execução das tarefas.

- Utilidade para o trabalho dos usuários

Proposição P3 (pág. 50): Os gestores de TI reconhecem que os softwares livres de modo geral têm qualidades técnicas comparáveis ou superiores aos produtos de software proprietário.

Assim como o abordado no tópico anterior, os entrevistados consideram que também para o usuário final, os softwares livres têm qualidades técnicas superiores. Principalmente em relação à velocidade de execução das tarefas, e à produtividade, pois consideram que os programas livres têm um desempenho melhor que os equivalentes proprietários quando executados no mesmo hardware, e os consideram mais estáveis, pois ocupariam menos o tempo do usuário com reinícios de computador e chamados de manutenção.

Quanto à eficiência para alcançar os objetivos do usuário final e a possibilidade de facilitar a execução de tarefas, os respondentes consideraram que como os produtos em software livre são muito parecidos e têm os mesmos recursos que os produtos utilizados atualmente, os usuários conseguiriam executar seu trabalho exatamente da mesma forma.

6.1.1.3.FACILIDADE

Proposição P1b (pág. 46): A utilização do software livre nas estações de trabalho é considerada, pelos gestores de TI, inconveniente devido à necessidade de treinamento da equipe de informática.

Proposição P1j (pág. 48): As soluções em software livre são consideradas, pelos gestores de TI, como mais difíceis de serem utilizadas por usuários leigos em informática, o que implicaria em grandes necessidades e altos custos de treinamento.

Todos os entrevistados enfatizaram a necessidade de treinamento, tanto para equipe de informática quanto para os usuários finais. No entanto, como o discutido nos ítems anteriores, não acreditam em grandes dificuldades de assimilação por nenhuma das partes, e enfatizam que a operação diária dos sistemas seria tranqüila e produtiva.

6.1.1.4. ATITUDE

O resultado das opiniões expressadas nos construtos anteriores influenciam diretamente, a atitude dos gestores de TI quanto à utilização de software livre nas empresas. Baseado na participação de mercado observada em pesquisa conduzida pela FGV (Meirelles, 2003), propôs-se que a atitude seria de resistência a uma adoção generalizada de software livre. Foram considerados os seguintes parâmetros:

1. Resistência à mudança devido a pressões externas, de mercado (externalidades de rede). Esse fator foi preponderante na atitude de todos os entrevistados, que afirmaram ser problemática a adoção de tecnologias que não são largamente utilizadas pelo mercado. Um dos entrevistados enfatizou que em sua empresa são utilizados somente produtos de participação majoritária. Apenas um dos entrevistados pretende utilizar software livre em todo o ambiente, mas mesmo assim colocou que teme o problema com as externalidades de rede, e que eles seriam compensados pela economia com licenças.
2. Resistência à mudança devido a dificuldades na migração. Esse fator, de acordo com as respostas obtidas, não teve um grande peso na atitude geral de rejeição. Excetuando-se um dos entrevistados, que não admite qualquer possibilidade de mudança, os demais entrevistados consideram que o processo de migração, se realizado, não teria grandes custos ou grandes dificuldades.
3. Resistência à mudança devido à necessidade de trabalho político na empresa. Essa atitude foi descrita pelos entrevistados como tendo duas faces: a aceitação dos sistemas pelos usuários finais e a justificativa do projeto perante a direção da empresa. A importância da aceitação pelos usuários finais foi minimizada, porque consideram em primeiro lugar que como as interfaces e ferramentas de trabalho são muito parecidas com as existentes, isso

seria facilmente assimilado; em segundo lugar, que a utilização dessas ferramentas seria uma imposição superior, vinda da diretoria, portanto não passível de contestação. A segunda face seria a justificativa de execução do projeto de migração. Os entrevistados de um forma geral consideram que como já têm o software de seu parque de computadores licenciado, a maior justificativa para adoção, que seria a economia em gastos com licenciamento, perderia sentido. Somente um dos entrevistados, que não tem seu parque licenciado, colocou a utilização de software livre generalizada seria plenamente justificável através do argumento de economia de licenças.

4. Resistência à mudança devido a dúvidas quanto à possibilidade de utilização atual e futura de aplicações de negócio e missão crítica. Todos os entrevistados consideram que os produtos em software livre já suportam, e cada vez mais continuarão a suportar, aplicações de negócios. Foram citados os casos de empresas como IBM e Oracle, que recomendam a execução de seus aplicativos sobre plataforma GNU/Linux, e a credibilidade que essas empresas passam ao mercado. Foi citado também o caso de um banco que migrou parte sua estrutura de missão crítica também para um sistema GNU/Linux. Um dos entrevistados destacou a preocupação mercadológica que a Microsoft tem demonstrado nos últimos meses quanto ao “crescente aumento de utilização de soluções livres”.

5. Resistência à mudança devido ao descrédito quanto ao uso em larga escala pelo mercado a médio e longo prazo. Todos os entrevistados acreditam que uso do software livre terá uma curva ascendente nos próximos anos, principalmente pela opção de grandes corporações como a IBM e Novell em utilizar o GNU/Linux. Dois dos entrevistados citaram como importante a opção feita pelo governo brasileiro em priorizar o software livre em suas políticas públicas, e que essa atitude pode afetar seriamente as decisões de adoção de tecnologia por parte das empresas privadas.

6.1.1.5.INTENÇÃO

- Serviços onde há viabilidade de uso

Excetuando-se um dos entrevistados, que não vê viabilidade de uso real de softwares livres em qualquer tipo de serviço, os demais usuários consideram o uso de software livre tecnicamente possível, em qualquer categoria, para servidores, principalmente para o uso de tecnologias relacionadas à Internet. Considerariam tecnicamente viável para estações se fosse resolvido o problema de abertura de formatos proprietários. Um dos entrevistados acredita na viabilidade de adoção nas estações de uma das filiais, onde é utilizado somente o sistema de gestão empresarial, através de emulação de um servidor Windows localizado no escritório central. Nesse caso, como não são produzidos ou recebidos arquivos, não haveria problemas de formatos. Um dos entrevistados vê viabilidade em migrar todas as estações para software livre, devido à economia que teria com licenças.

- Serviços onde não há viabilidade de uso

De um modo geral, todos os entrevistados acreditam que é tecnicamente possível utilizar software livre em todas as categorias analisadas, mas sua viabilidade real esbarra nas questões das externalidades de rede, principalmente no suporte técnico do ambiente de produção, na questão do intercâmbio de arquivos com formatos proprietários e na justificativa de investimentos para estruturas com licenças já legalizadas.

- Intenções de uso futuro

Excetuando-se um dos usuários, que pretende adotar software livre nas estações de trabalho devido aos custos de licenciamento de software, os demais entrevistados não pretendem utilizar mais produtos de software livre do que os adotados atualmente.

6.1.1.6.USO REAL

Das empresas entrevistadas, apenas uma não utiliza nenhum software livre. Todas as outras têm servidores de Internet sendo executados em algum sistema operacional GNU/Linux, e nele executam aplicativos para compartilhamento de conexão, correio eletrônico, distribuição de acesso à navegação, correio eletrônico e anti-vírus. Uma das empresas, além das funções comuns de Internet, utiliza um sistema GNU/Linux como plataforma para servidor de banco de

dados e servidor de arquivos. Todos os usuários de software livre declararam-se totalmente satisfeitos com os recursos utilizados.

6.2. EMPRESA DE GRANDE ESTRUTURA USUÁRIA DE SOFTWARE LIVRE

Como forma de conseguir uma melhor compreensão do fenômeno de adoção do software livre nas organizações, foi entrevistado também o gestor de TI de uma empresa de grande estrutura de informática³⁹, que é também uma grande usuária de software livre.

Essa empresa existe já há 40 anos, atuando sempre na área de prestação de serviços. Hoje é organizada como um grupo de cinco empresas, que atua em todo o Brasil e na América Latina. Têm atualmente cerca de 1000 teclados, sendo que 350 estão na matriz e o restante espalhado por todas as filiais: 84 no Brasil e 11 em outros países. A empresa tem uma grande infra-estrutura de informática, sendo que o departamento de TI conta hoje com cerca de 80 profissionais.

A diretoria de tecnologia para o grupo todo é única, e a decisão de adoção de novas tecnologias é lá mesmo tomada, sem influências externas diretas de qualquer agente, como parceiros de negócios, clientes ou fornecedores. Para orientar essas decisões, são feitas consultas a profissionais externos, principalmente através de grupos de estudos de tecnologia, e também através de eventos, revistas e órgãos de imprensa especializados. Em todas as decisões são feitas avaliações de custo e benefício, principalmente no que tange a questões de custo agregado, que vão além do gasto inicial com software ou hardware: alocação de mão de obra especializada, tempo de desenvolvimento, tempo de maturação, manutenção e treinamento.

Nessa empresa, todos os sistemas de negócio são desenvolvidos internamente. Atualmente estão passando por uma fase de transição, desenvolvendo e implementando um sistema chamado SI – Sistema Integrado – cujo objetivo é integrar todos os negócios da empresa num único sistema. O gerente de tecnologia considera que o desenvolvimento desse sistema foi uma “aposta pesada”, mas que está dando resultados positivos. A opção pelo desenvolvimento interno foi feita porque, segundo o gerente, “não há um pacote que atenda às necessidades da empresa”, devido à sua presença em todo o território nacional e também fora do Brasil, e à atuação em diversos ramos de negócios dentro da área de prestação de

³⁹A pesquisa de uso de recursos de informática da FGV (Meirelles, 2003) divide o porte da estrutura de informática de acordo com a quantidade de teclados: pequenos (30 a 160), médios (160 a 500) e grandes (mais que 500).

serviços. Consideraram que se fossem adotar um sistema de mercado, a enorme quantidade de adaptações necessárias, somada aos gastos com licenças e consultoria para implementação, tornariam o projeto inviável, além de produzir um resultado que não atenderia totalmente às necessidades da organização. Os primeiros módulos do SI foram colocados em produção em janeiro de 2003, e os demais estão entrando no ar gradativamente até que, segundo o planejamento feito pela equipe de informática, todo o desenvolvimento esteja concluído até o final do ano de 2005.

Após conseguir uma integração maior dos sistemas, têm em seu planejamento a implementação de um sistema de Business Intelligence (BI), para tomada de decisões gerenciais. Segundo o gerente de TI, esse projeto ainda não foi iniciado porque o legado de sistemas desenvolvidos, que utilizam as mais diversas linguagens e operam isoladamente em diversos bancos de dados, inviabiliza um projeto nesse sentido. No entanto, assim que os principais negócios da empresa tiverem migrado para o SI, esse processo será iniciado.

A empresa decidiu, com a implementação do SI, desenvolver os sistemas utilizando dois tipos de tecnologia: linguagem de programação Java com banco de dados Oracle para as aplicações grandes, e linguagem PHP com banco de dados PostgreSQL para aplicações pequenas.

Para o desenvolvimento do SI em Java, a empresa adotou uma estrutura parecida com a que era utilizada na época de grande utilização de mainframes. Há três papéis bem claros na equipe de desenvolvimento: o analista de negócios, o analista de sistemas e o programador. A empresa considerou que o desenvolvimento em Java, principalmente devido à orientação a objetos, é mais complexo, o que gera diferenças de arquitetura tanto na linguagem quanto na metodologia para desenvolvimento dos sistemas. Por isso, na equipe tem também a figura do arquiteto Java, que cuida da parte de orientação a objetos, e um administrador de bancos de dados. No início do projeto, em maio de 2002, havia uma certa escassez de profissionais Java no mercado, o que implicou em dificuldades de contratação e custos salariais maiores.

Java foi escolhida, em primeiro lugar, por ser uma linguagem cuja especificação é livre, e em segundo lugar, por ser considerada um software estruturado e eficiente, com “garantia de continuidade a longo prazo”, segundo o gestor de TI. A empresa chegou a avaliar outras alternativas. O Cobol seria uma linguagem boa e rápida, mas teria limitações para uso com bancos de dados, uma vez que o modo padrão de armazenamento de dados do Cobol é de arquivos compartilhados. Para sanar essa limitação, seriam exigidos investimentos em

softwares e hardwares caros e proprietários, o que aprisionaria a empresa junto a um fornecedor específico dessas soluções. O Visual Basic foi descartado porque a empresa seria obrigada a trabalhar necessariamente com o ambiente Microsoft nos servidores e nas estações. O gerente acredita que os produtos Microsoft para servidores consomem muitos recursos, o que exigiria grandes investimentos e atualizações constantes de hardware. A linguagem C foi descartada porque seu uso seria muito complicado, e seria ainda mais difícil encontrar profissionais disponíveis no mercado. E foi eliminada a linguagem Delphi porque em testes internos os profissionais da empresa teriam encontrado limitações não existentes na linguagem Java. Esta foi, por fim, escolhida como linguagem padrão.

Nessa empresa, o software livre é amplamente utilizado, e é fortemente defendido por profissionais de informática. Em primeiro lugar por uma questão de custos, e em segundo lugar pela liberdade. Segundo o gerente de TI, a economia com licenças é significativa, mas há outros fatores envolvidos. Considera que o uso do software livre proporciona melhor controle e melhor possibilidade de personalização das soluções utilizadas. E o que é investido em pesquisa, suporte e capacitação do pessoal interno para implementação dessa solução é preservado como conhecimento da empresa, cujo investimento para sua construção é diluído por um benefício obtido para mais de mil usuários na corporação.

Atualmente, a empresa tem todos os servidores da matriz – exceto um, que precisa executar um sistema legado – operando com sistemas operacionais GNU/Linux. E está em processo de substituição do sistema das filiais, que em sua maioria ainda executam alguma versão do Windows, da Microsoft. A empresa espera ter, com a mudança, ganhos principalmente de segurança, desempenho e disponibilidade. Está sendo concluído também, um estudo para utilização de software livre em todas as estações de trabalho. A empresa avalia que como o sistema corporativo é executado totalmente no servidor, com aplicações Java ou PHP, e os aplicativos legados podem ser totalmente executados através de emulação de terminal ou de estações remotas, não há necessidade de utilizar Windows nas estações. Algumas já estão operando no novo sistema, que deve ser estendido para as demais durante o ano de 2004.

Mesmo operando com o sistema operacional Windows nas estações, três departamentos da empresa já foram totalmente migrados para o pacote de aplicativos de escritório OpenOffice.org, software livre funcionalmente equivalente ao Microsoft Office. Ao contrário do que, de modo geral, julgam os demais entrevistados desta pesquisa, a empresa não está encontrando dificuldades significativas na troca de documentos com clientes e fornecedores. Além disso, com o SI, pretendem tornar disponível para os clientes, via Internet, diversas

informações que hoje são enviadas através de planilhas produzidas no Microsoft Excel, eliminando definitivamente qualquer potencial problema nesse sentido. O gerente de TI considera que o processo de treinamento e adaptação do usuário para utilização do sistema operacional GNU/Linux e do OpenOffice.org não será fácil, devido ao conhecimento existente e a costumes arraigados. No entanto, aposta num trabalho diário de convencimento e em treinamentos formais organizados internamente para vencer as resistências.

Um dos grandes objetivos com a migração das estações é resolver os problemas de manutenção na ponta do usuário, hoje muito freqüentes. Há uma grande expectativa de que as estações com software livre possibilitem maior controle, maior estabilidade e maior desempenho, uma vez que existe forte preocupação quanto ao custo de uma máquina parada, principalmente em termos de desgaste com usuário, de custos econômicos, de trabalhos atrasados ou não realizados, e de credibilidade para a empresa e para o departamento de informática.

Todo o processo de avaliação dos aplicativos, homologação e treinamento é feito internamente, através de profissionais especificamente designados para isso. O objetivo da empresa é procurar investir em pessoas, e não em licenças de software. Segundo o gerente de TI, a motivação da equipe técnica tem sido expressiva desde quando foi feita a opção pelo uso de software livre. Afirmou que ao ser anunciada a opção por Java e PHP, viu uma motivação incomum na equipe, devido à novidade e ao desafio. E todos os colaboradores são constantemente treinados para utilizar e dar suporte às tecnologias adotadas, como Java, UML, arquitetura de sistemas, OpenOffice.org, PHP, entre outros. Na empresa há uma estrutura própria para treinamento, que é utilizada para essa finalidade.

A empresa avalia que a opção pelo uso de software livre é uma opção viável e consistente ao longo do tempo. O gerente de TI acredita que o movimento de software livre é uma consequência natural da sociabilização da informação promovida pela popularização da Internet. Ressalta que as pessoas que trabalham com software livre não o fazem de graça, e sim para atender a diversos objetivos, que vão desde interesses comerciais de empresas nas quais trabalhem, até a interesses de desenvolvimento pessoal. Cita como exemplo sua própria equipe de trabalho, que participa da comunidade Java e colabora com várias outras pessoas dentro e fora do Brasil. Vê o processo de desenvolvimento e de aprimoramento do software livre como um caminho sem volta, que vai conviver por muitos anos como alternativa viável aos programas proprietários. E diz também que a postura de um usuário de software livre é diferente. Quando alguma coisa não funciona a contento, não é suficiente simplesmente fazer

reclamações de que existe um problema ou que uma funcionalidade não foi atendida. A pessoa deve se expor, e ajudar a resolver o problema. Não existe mais a relação cliente e fornecedor, mas sim uma relação colaborativa, de membro e participante da comunidade.

6.3.EMPRESA DE SOFTWARE LIVRE

Para ampliar a visão a respeito da situação do software livre no mercado foi entrevistada uma empresa especializada em vender e implementar soluções com software livre.

Essa empresa existe desde 1999, sempre trabalhando com treinamento, consultoria e implementação de soluções em software livre, com foco nos sistemas operacionais baseados no Linux⁴⁰. Seu surgimento vem de um relacionamento entre seu atual diretor e um grupo de técnicos que trabalhavam com consultoria em implementação de soluções em informática. A opção pelo Linux aconteceu porque, segundo o entrevistado, este “despontava como uma coisa estratégica, contra o qual não haveria muita concorrência”.

Segundo o diretor, no segmento de empresas com pequena estrutura de informática haverá uma explosão na utilização de soluções em software livre a partir do ano de 2004. Em seu ponto de vista, haverá até mesmo escassez de profissionais no mercado para suprir as necessidades dos clientes: “os profissionais e empresas especializadas que trabalharem bem terão ótimas perspectivas de crescimento, e os profissionais serão disputados no mercado”, considera. Acredita também que um dos maiores motivadores para adoção de software livre é a recente opção governamental. Expõe que o governo é o maior comprador de software do Brasil, e na esteira da adequação a ser realizada pelas empresas que o atende, muitas outras também farão opção por esse tipo de tecnologia. O diretor posiciona-se também favorável ao projeto em discussão no Congresso Nacional pela criação de uma lei que dê preferência ao software livre nas compras governamentais⁴¹.

O gestor de TI de empresas de pequena estrutura (30 a 160 teclados) é considerado, pelo entrevistado, mais agressivo na adoção de novas tecnologias que aquele das empresas maiores. Segundo ele, essas empresas menores têm um orçamento menor, e precisam implementar mudanças e novas funcionalidades num período menor de tempo menor, com um orçamento mais enxuto. Como os produtos livres não têm a barreira de entrada do custo de

⁴⁰Nesta seção será utilizado o termo Linux (ao invés de GNU/Linux) como referência ao sistema operacional completo, porque é assim que o diretor dessa empresa a ele se refere.

⁴¹Esse assunto será melhor discutido na próxima seção, que trata da posição das empresas de software proprietário e sua relação com o governo.

licenciamento, é mais fácil para a pequena empresa colocar rapidamente em operação esse tipo de solução. Apesar disso, o diretor considera que os gestores de TI dessas empresas não estão bem informados quanto às questões relativas ao software livre, e que é papel do fornecedor de serviços levar a esse público mais conhecimento e mais informação.

Acredita que há grande carência de uma comunicação melhor sobre software livre, direcionada especificamente para o público usuário. O software livre é muito falado e discutido por quem já está nesse meio, mas não há um trabalho setorial focado em nichos específicos. Considera isso uma fraqueza do Linux, que tem sua promoção dividida entre várias empresas, sem uma unidade de comunicação. A participação das gigantes HP, Oracle, Intel e IBM foram muito importantes para oferecer credibilidade ao sistema, mas seria necessário um acordo entre os grandes fornecedores para unificar a comunicação, que seria depois aberta e envolveria o resto da comunidade. Acredita também que o Linux tem todo o potencial para se tornar a plataforma de software hegemônica no mercado, mas que isso depende da forma como essa comunicação unificada será conduzida, caso chegue a existir.

Quanto à grande participação de mercado dos produtos proprietários, cujas categorias estão dentro do escopo desta pesquisa, o diretor enxerga nessa situação uma oportunidade singular de crescimento para as empresas que trabalhem com software livre. Vê a venda de soluções com software livre como estratégica, uma vez que não é vendido somente um programa, mas sim toda uma solução, enfatizando ganhos com licenciamento, redução de ativos, suporte e segurança. Com isso, é colocada para o cliente uma solução completa, com ganhos que vão muito além da simples redução de custos. E se o fornecedor conseguir fazer com que o cliente compartilhe com sua visão, o valor percebido é muito maior que aquele fornecido por empresas que somente vendem licenças, possibilitando parcerias duradouras e a melhor remuneração pelos serviços prestados.

Considera que a percepção de incompatibilidade com formatos proprietários é temida por seus potenciais usuários, mas que isso é contornado de diversas formas, como por exemplo a adoção de uma estação com MS Office para conversão de arquivos e macros. Quando há um temor por parte do cliente quanto à troca de arquivos, a empresa faz demonstrações para mostrar que os problemas são mínimos, e a convivência é possível. No entanto, teme que se a Microsoft começar a mudar os formatos com frequência, pode efetivamente ser prejudicada a adoção do Linux nas estações de trabalho.

O diretor acredita que a notícia de uma participação de mercado maior não influenciaria diretamente o processo de adoção do software livre nas empresas. Os clientes pensariam em primeiro lugar, não em software livre, e sim em Linux. A utilização dos outros softwares livres seria apenas uma decorrência da mudança da plataforma, pois as empresas conhecem e ouvem falar somente do Linux, e não dos outros aplicativos em software livre. Começariam a questionar primeiro a utilidade do Windows, e em seguida aceitariam o GNU/Linux como alternativa, mesmo que continuando a utilizar softwares proprietários para necessidades específicas.

Quanto à dúvida em relação à disponibilidade de suporte técnico e de mão de obra, o diretor as classifica como mitos, e acredita que é mais forte nas empresas maiores: “as empresas grandes colocam a pergunta de quem vai dar suporte, mas já sabem a resposta, pois é uma questão já resolvida”. Diz que sua empresa, assim como várias outras, prestam suporte técnico de alta qualidade e disponibilidade. Segundo ele, nas empresas pequenas essa questão já está mais esclarecida, e a barreira é menor, porque há escassez de recursos e necessidade de velocidade na tomada de decisões. Enquanto a empresa pequena se dispõe a testar compatibilidades e funcionalidades, as grandes bloqueiam esse tipo de iniciativa. Segundo o diretor, atualmente não há muitos questionamentos sobre se o sistema Linux funciona ou não funciona, se tem suporte ou não. O questionamento não é mais sobre o que é, mas sim sobre como e quando implementar. Para o público com o qual a empresa trabalha, não existem mais esses mitos, e como possui um trabalho de esclarecimento junto aos potenciais clientes, a entrada no projeto já se dá com esse tipo de questão resolvida.

Quanto ao treinamento, o diretor enfatiza que há boa receptividade por parte dos clientes, pois os técnicos não precisam, ao contrário dos produtos proprietários, ser retreinados a cada nova versão. O treinamento é encarado como um processo constante, onde o profissional evolui, e não tem seu conhecimento revolucionado a cada momento por novas versões. Existe uma certa dose de resistência, no início, com alguns profissionais de informática acostumados a trabalhar com produtos proprietários. Mas com o contato mais profundo com o sistema acabam gostando, e tornando-se entusiastas. O Linux, segundo ele, é “apaixonante, porque é possível prever o futuro”. “Com o Windows, há sempre a dúvida se determinada atividade será ou não possível, se a instalação dará certo ou não. Com o Linux as coisas são previsíveis, é possível saber exatamente o que se está fazendo”, enfatiza. E acredita que os clientes sentem esse controle, citando casos de ambientes que ficaram mais estáveis e produtivos após a migração para o GNU/Linux, com benefícios como maior estabilidade, maior controle, menor

necessidade de suporte, menos problemas com vírus e o fim da necessidade de controle de licenças.

Quanto à migração das estações, o diretor acredita que o processo é sempre mais difícil que nos servidores. Para isso, cita que foi desenvolvida por sua empresa uma metodologia específica, cujo objetivo principal é detectar incompatibilidades, e solucioná-las antes da migração. O trabalho para migração das estações é sempre mais demorado e custoso, porque envolve uma série de detalhes que não existem na migração de servidores. Acredita que o Linux é perfeito para execução de aplicativos de missão crítica, mas que o maior desafio está nas estações, porque o sistema operacional Windows está muito bem consolidado, e a resistência à mudança, por parte do usuários, ainda é grande.

Segundo ele, a diretoria das empresas clientes tem sempre preocupação com os riscos inerentes ao processo de mudança, porque nela são adotados novos conceitos; são instalados novos produtos; são feitas mudanças na estrutura de funcionamento da tecnologia da informação. E como qualquer empresa hoje depende muito da TI, o negócio do cliente não pode ser colocado em risco. Mas enfatiza que se o fornecedor souber mostrar os benefícios, e fizer bem as implementações, de forma a gerar benefícios usando software livre como meio, “ganha-se o cliente, ganham-se negócios, amplia-se a empresa”.

6.4. EMPRESAS DE SOFTWARE PROPRIETÁRIO

Para confrontar as posições entre os modelos livre e proprietário, buscamos conhecer a posição das empresas de software proprietário. Para isso, foi entrevistado o principal executivo de um entidade brasileira que congrega diversas empresas de tecnologia da informação, que trabalham principalmente com comércio eletrônico, produção e comercialização de software.

A posição oficial da entidade é que na questão da adoção de tecnologia, deve haver liberdade total para que as empresas, e principalmente o governo, façam suas opções. Essa entidade tem a mesma posição, e já fez parte da base de apoio do Movimento pela Livre Escolha⁴², criado pelas estadunidenses Computing Technology Industry Association (CompTIA) e Business Software Alliance (BSA). Na opinião dessas entidades, não deve haver, por parte do governo, qualquer tipo de lei, orientação ou política que direcione a escolha de software. Segundo eles, a escolha deve ser livre, de acordo com as necessidades identificadas pelo gestor público responsável pela adoção.

⁴²Sítio na Internet: <<http://www.softwarechoice.org>>.

No início de 2003, o responsável pela Coalizão pela Livre Escolha, Raphael Mandarino Júnior, então membro do comitê gestor da Internet e presidente da Associação Nacional dos Usuários de Internet (ANUI), declarou na imprensa⁴³ que considera que um dos riscos da obrigatoriedade do uso do software livre na administração pública é o que ele classifica como “nivelamento da mediocridade”, porque para não se expor a punições com base na lei, esses órgãos vão optar por “soluções medíocres, mas permitidas, em vez de procurar o melhor”. Acredita também que empresas de software proprietário podem quebrar, porque não teriam como exportar “produtos que são rejeitados no Brasil”, e que a adoção de software livre pelo governo representa um retrocesso, com a volta da reserva de mercado, e que é uma medida “que não interessa ao país como um todo, mas apenas a um grupo específico”.

Esse assunto veio à tona no ano de 2003, quando o governo federal e o Congresso Nacional sinalizaram o início de um programa de apoio à adoção de software livre no poder público. Essa posição se tornou evidente à sociedade com a realização, em agosto desse mesmo ano, da Semana de Software Livre no Legislativo⁴⁴. Esse evento foi patrocinado pelo Congresso Nacional, em Brasília, onde foram discutidos e apresentados os mais diversos aspectos relacionados ao uso e ao desenvolvimento do software livre no governo, nas empresas e na sociedade. A adoção do software livre no governo foi defendida principalmente pelos seguintes motivos:

1. Economia de recursos – Uma questão insistentemente formulada pelo governo é o déficit de mais de um bilhão de reais da balança comercial somente com o pagamento de licenças de software proprietário. Argumenta-se também que um país pobre como o Brasil não pode se dar ao luxo de transferir recursos para empresas estrangeiras, sendo que há alternativas livres disponíveis. E além disso, os gastos feitos com o software livre seriam feitos com a capacitação de profissionais e investimento em serviços cujos agentes permanecem no Brasil.
2. Independência tecnológica – O uso do software livre tornaria o Brasil independente do desenvolvimento de software produzido no exterior. Como os códigos e métodos de trabalho são abertos, qualquer instituição, empresa ou pessoa pode ter controle sobre o funcionamento dos programas.
3. Segurança nacional – Como o software livre é distribuído acompanhado dos códigos fontes, fica fácil para as organizações governamentais auditá-lo, e garantir que não há

⁴³Revista E-Commerce. Edição 32, março de 2003, pág. 64.

⁴⁴Sítio na Internet: <<http://www.congresso.gov.br/softwarelivre>>.

seções maliciosas, que poderiam eventualmente transmitir informações sem autorização ou quebrar dados confidenciais.

A entidade entrevistada, que é uma das representantes da indústria nacional de comércio eletrônico e de software proprietário, contesta essa visão, e afirma que a questão da dependência nacional de software é uma questão muito mais profunda, que não será resolvida somente com a adoção de software livre no poder público.

O que as empresas de software proprietário cobram é criação, por parte do governo federal, de uma política nacional de tecnologia da informação e comércio eletrônico. Essa política nortearia toda a ação governamental direcionada à tecnologia da informação nas atividades relacionadas à sociedade, à educação e às políticas para o setor privado e para o governo. Todos os demais assuntos seriam subordinados a essa política geral, que conteria o uso de software livre apenas como um de seus tópicos.

As reivindicações das empresas de software proprietário junto ao governo são as seguintes⁴⁵:

1. Criação de um marco regulatório para o software, que defina seu enquadramento como produto, serviço ou mercadoria, e dê a ele um tratamento tributário claro, com uma regulação específica que lhe defina a natureza jurídica intrínseca e de suas operações no mercado.
2. Incentivos fiscais para empresas produtoras de software proprietário, principalmente àquelas que façam exportação, para que tenham uma carga tributária equiparada a outros países com os quais competem. Um dos principais pontos nesse sentido é a redução de encargos trabalhistas, sob forma de desonerar o principal insumo, que é o trabalho.
3. Regulamentação das profissões em tecnologia da informação, associada ao estímulo ao desenvolvimento de mão de obra especializada, através de incentivos a escolas e bolsas de estudo de pós graduação.
4. Incentivo ao desenvolvimento de programas produzidos no Brasil, através da preferência de uso destes em relação aos estrangeiros nas licitações. Reivindicam também linhas de financiamento especiais para empresas que investirem em certificações internacionais de desenvolvimento de software.

⁴⁵Obtidas através da entrevista pessoal, através da revista B2B Magazine (Padrão Editorial Ltda.)– agosto de 2003, págs. 34 e 35, e do documento “Política de Software”, divulgado pela Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet – ASSESPRO, através de seu sítio na Internet <<http://www.assespro.org.br>> em novembro de 2003.

5. O governo deve deixar de utilizar empresas estatais, órgãos públicos e recursos humanos de contratação própria para implementar projetos de TI, e passar a fazer contratações somente junto a empresas privadas que atuem no mercado.

6. Devem ser feitas contratação separadas de software, hardware e serviços, preferencialmente com fornecedores locais, ao invés de fazer contratos completos para todo o Brasil. Isso daria uma maior margem de participação para empresas nacionais, que em sua maioria tem atuação focada em apenas um dos ramos e têm atuação local. Deve permitir também a formação de consórcios de pequenas e médias empresas para viabilizar sua participação em projetos maiores, e evitar nas concorrências critérios direcionadores, como porte, desempenho anterior e funcionalidades que sejam superiores às reais necessidades.

7. Financiamento subsidiado pelo governo, com baixas taxas de juro, para compra de software proprietário pelas empresas usuárias.

8. Não adoção, por parte do governo, de qualquer lei orientando critérios para a escolha de software. A escolha deve ser livre, de acordo com critérios adotados pelos gestores públicos responsáveis pela compra.

9. Criação de uma agência nacional de tecnologia da informação, que seja independente, que defina padrões e cuide da competição entre as empresas.

10. Exigir, por parte das empresas estrangeiras de software, uma contrapartida, que seria utilizada para estimular o desenvolvimento das empresas de software brasileiras.

11. Reforçar a proteção à propriedade intelectual, entendida como fator determinante para a inovação tecnológica.

Com todas essas pretensões, sem dúvida os desenvolvedores brasileiros de software proprietário temem por uma atitude pró software livre por parte do governo. A posição da entidade entrevistada é a seguinte: o governo ainda não deu nenhum sinal de vontade para a criação de uma política genérica de tecnologia da informação; e mais, nenhum sinal de uma política específica para incentivo à indústria nacional de software proprietário, por isso uma política geral de adoção de software livre seria inaceitável. Além disso, a entidade entrevistada considera que a opção governamental de preferência pela adoção de software livre é totalmente ilegal, inconstitucional e contrária à lei de licitações.

Essa resistência acontece também por duas questões adicionais importantes. O primeira é que as empresas de software proprietário, de uma forma geral, não acreditam no modelo de negócios utilizado pelas empresas de software livre. Isso fica evidente na visão da entidade entrevistada, cujo gestor colocou em dúvida a capacidade das empresas que trabalham com software livre de se manterem. A idéia é que o desenvolvimento sustentado de software deve necessariamente depender da venda de licenças de uso, pois somente através delas seria possível obter recursos necessários à inovação e à evolução dos produtos. Esse posição é veementemente sustentada por Eduardo Campos de Oliveira – Gerente de Produtos Servidores Windows – da empresa Microsoft, que em entrevista⁴⁶ divulgada em 2002, afirma que o software livre é inviável porque “destrói a cadeia de valor das empresas” de TI, impedindo a venda de licenças e inibindo os lucros com prestação de serviços.

A segunda questão se refere aos custos de adaptação da estrutura existente ao uso de software livre. Segundo a associação entrevistada, cerca de 12% do mercado brasileiro de software é formado pelo governo. Hoje há uma gama de empresas que atendem a esse mercado, produzindo e dando suporte a software proprietário sobre plataformas prioritariamente proprietárias, especificamente o Microsoft Windows. Com a opção do governo em adotar software livre na máxima extensão possível, essas empresas deverão modificar sua estrutura de desenvolvimento e suporte, o que acarretaria custos de treinamento, readequação e adaptação das ferramentas existentes. Com isso, essas empresas temem que o mercado privado seja influenciado em cadeia. Além da influência direta sobre as empresas usuárias, consideram que se os 12% de mercado representados pelo governo se adequarem para oferecer soluções livres, sua influência sobre o mercado geral será cada vez mais decisiva, fazendo com que os produtos de software livre tenham muito mais visibilidade e sejam muito mais utilizados, diminuindo a participação de empresas que têm soluções proprietárias já estabelecidas.

Além das empresas brasileiras, empresas estrangeiras também têm se manifestado a respeito, sendo que uma das mais ativas nesse sentido tem sido a própria Microsoft. Um exemplo é o estudo encomendado em julho de 2002, à International Data Corporation (Bozman, 2002), onde argumenta-se que o custo total de propriedade do sistema operacional Windows 2000 é mais barato que o do GNU/Linux quando considerados custos globais num período de cinco anos. Outro exemplo é o documento “Perspectiva da Microsoft sobre o Linux para tomadores de decisão”, produzido pela própria Microsoft (Microsoft, 2002), que apresenta

⁴⁶Entrevista concedida à revista eletrônica Baguete Jornalismo Empresarial Digital em dezembro de 2002. Disponível em <<http://www.baguete.com.br>>.

motivos pelos quais a adoção dos sistemas operacionais da série Windows nas empresas é mais vantajosa em termos de custos, desempenho e funcionalidades.

Em resumo, a posição das empresas produtoras de software proprietário é contrária à promoção do software livre pelos seguintes motivos:

1. Como o modelo de negócios com o qual trabalham é fortemente baseado na venda de licenças, consideram que a migração para um modelo onde o software livre seja prioritário totalmente inviável.

2. Acreditam que modelo de negócio de software livre é inviável mesmo para outros tipos de empresas – como as voltadas para o suporte técnico – porque o único benefício que o software livre traria seria a economia de licenças, o que seria insuficiente para sustentar sua permanência no mercado e justificar sua adoção pelas empresas usuárias.

3. Seriam necessários grandes investimentos relacionados a treinamento, adequação e redensolvimento para atender a requisições de trabalhos em plataformas livres, caso estas fossem adotadas pelo governo e conseqüentemente se espalhassem pelo mercado privado.

7. DISCUSSÃO

7.1. ANÁLISE DO MODELO

Inicialmente, foram propostos uma série de fatores que se dispunham a explicar a razão da baixa participação de mercado dos produtos em software livre. De acordo com as respostas obtidas, em confronto com as proposições iniciais, foi possível sumarizar o seguinte quadro:

Quadro 7 - Relevância das proposições

Proposições	Relevância
P1 – Baixo uso devido a problemas com externalidades	Alta
P1a – Altos custos de migração	Baixa
P1b – Treinamento da equipe de informática	Baixa
P1c – Dificuldades de mensuração de viabilidade de investimento	Alta
P1d – Tempo necessário à migração	Baixa
P1e – Possibilidade de problemas técnicos	Baixa
P1f – Incômodos para o usuário	Baixa
P1g – Necessidade de trabalho político	Baixa
P1h – Imperfeições na abertura de formatos proprietários	Alta
P1i – Desconhecimento de produtos disponíveis	Baixa
P1j – Software livre difícil de ser utilizado	Baixa
P2 – Viabilidade somente para serviços específicos	Alta
P2a – Uso de software livre relacionado com a adoção da Internet	Baixa
P2b – Descrença de suporte a aplicações de negócios	Baixa
P2c – Não há intenção de migração das aplicações de negócios	Alta
P2d – Não está disponível suporte técnico	Alta
P3 – Produtos em software livre têm boas qualidades técnicas	Alta

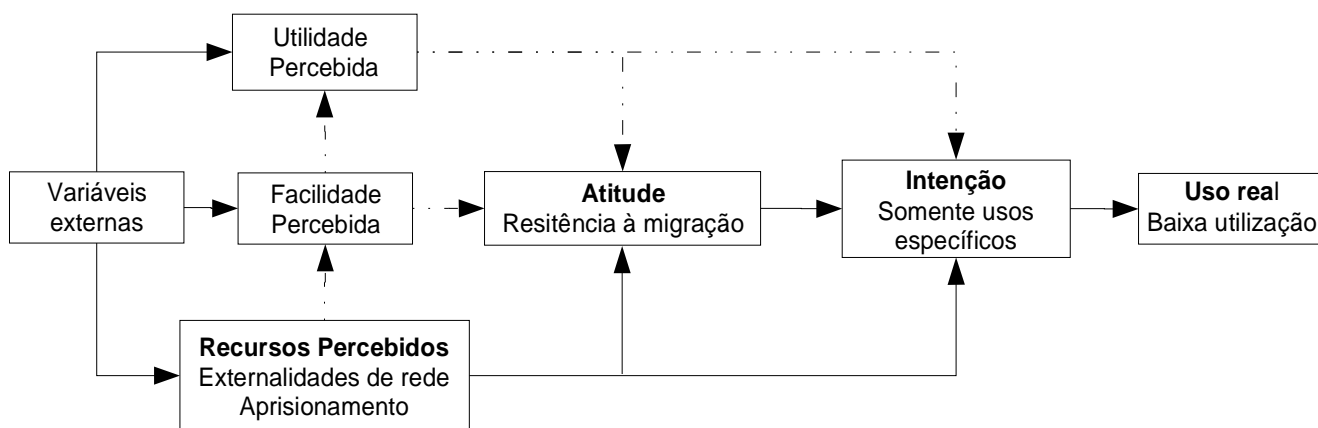


Figura 4 - Diagrama do MAT com destaques para representar o resultado das entrevistas

Esse quadro, então, transposto para a representação gráfica do modelo de adoção de tecnologia (MAT), poderia ser representado da seguinte forma:

A Figura 4 representa o MAT, com os resultados obtidos nas entrevistas. As linhas cheias representam influências significativas, e as linhas tracejadas influências fracas. Abaixo do nome de cada construto estão os fatores mais relevantes encontrados, que se encadeiam de forma a resultar o quadro final, que é a baixa participação de programas livres no mercado brasileiro de software.

Como visto, o construto que tem maior influência na atitude e na intenção de uso, e conseqüentemente no uso real, é o “recursos percebidos”. De acordo com o descrito na seção anterior, e resumido no Quadro 7, várias proposições apresentadas puderam ser descartadas. Isso quer dizer que muito do que se supunha ser uma barreira para a adoção do software livre nas empresas, na verdade não é descrito como tal na visão dos gestores de TI. Foi possível perceber, então, que as proposições que não puderam ser descartadas estão intimamente relacionadas com externalidades positivas de rede.

Dois pontos fortemente citados foram o temor de incompatibilidade com arquivos ou padrões normalmente utilizados, e a falta de suporte técnico para a operação diária. Mas independentemente dessas questões específicas, foi possível perceber uma grande vontade, por parte dos gestores, de sempre utilizar tecnologias aceitas pela maior parte do mercado. Curiosamente, o próprio uso de soluções com software livre para a estrutura de Internet também se enquadra nesse aspecto, uma vez que os dois estão intimamente ligados, e seu uso não afeta diretamente os aplicativos que estão de frente com o usuário, e nos quais a

empresa confia para condução de seus negócios. Isso nos remete então à questão da participação das empresas usuárias na rede dominante, e do usufruto de suas externalidades.

Segundo Shapiro e Varian (1999), a economia da informação não pode ser entendida através dos conceitos vindos da economia industrial, mas sim por uma outra perspectiva: a economia de redes. As redes da economia da informação são formadas ao redor de tecnologias que se tornam dominantes através do sistema de realimentação positiva: quanto mais usuários, maior é seu valor, e quanto maior o valor, mais usuários querem participar da rede. Por isso, em mercados onde é possível observar o movimento de realimentação, existe a tendência de que uma determinada tecnologia ou empresa se torne dominante, e atue no mercado com a força de um monopolista. Na tecnologia da informação, o caso mais evidente é o da Microsoft, com seu sistema operacional Windows e suas aplicações para estações de trabalho.

O próprio mercado de estações é um caso onde a força da rede se mostra patente. Como praticamente todos os usuários de computador têm que trabalhar com esse tipo de ambiente, realimentações positivas fortes podem afetar significativamente uma vasta gama de pessoas, fazendo com que suas influências tenham um impacto determinante no produto líder, e fulminante para os demais. Um caso diferente ocorre nos servidores, onde o efeito da rede muitas vezes é mitigado, pois uma quantidade menor de pessoas têm contato direto com eles. Por isso, podemos considerar que o mercado de estações de trabalho é especialmente oscilante, uma vez que iniciado o processo de realimentação positiva durante a construção da rede, muito provavelmente a participação de mercado dos demais sistemas será realimentado negativamente, fazendo com que apenas uma tecnologia se torne amplamente dominante. Isso aconteceu com o Microsoft Windows, que tem hoje cerca de 96% do mercado de estações de trabalho (Meirelles, 2003). Na metade dos anos 90, existiam ainda dois concorrentes mais próximos, o MacOS, da Apple, e o OS/2, da IBM. O processo de realimentação decretou a morte do OS/2, e fez com que a Apple se contentasse a ter o MacOS prioritariamente como um sistema operacional de nicho para aplicações gráficas. Caso semelhante aconteceu com navegadores de Internet, no caso Internet Explorer contra Netscape no final dos anos 1990; nas planilhas eletrônicas, em Visicalc contra Lotus 123 nos anos 1980, e depois em Lotus 123 contra Excel no início dos anos 1990.

O padrão de realimentação positiva é descrito por Shapiro e Varian (1999, p. 210) como uma curva no formato de “S”:

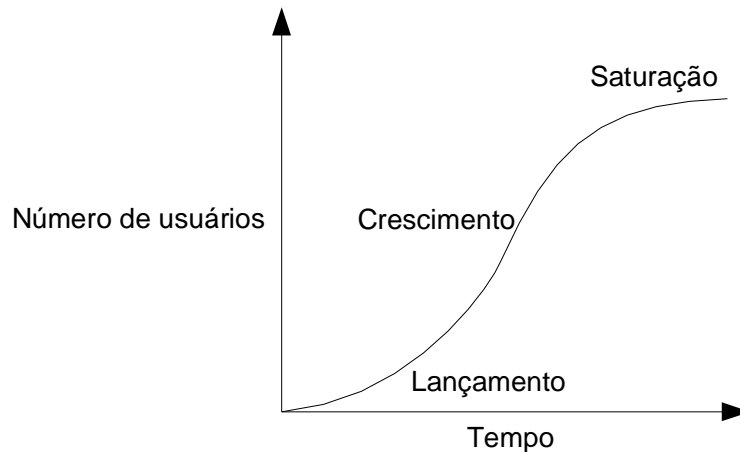


Figura 5 - Dinâmica de realimentação positiva

No processo de formação da rede, inicialmente há poucos adotantes, o que é indicado no gráfico como o período de lançamento. A adoção é lenta até que ocorra um ponto de inflexão, onde o crescimento tem um padrão quase que exponencial, uma vez que é acelerado pelo processo de realimentação positiva. Esse processo tem continuidade até que seja atingido um ponto de saturação, onde a maior parte do mercado já foi conquistada, e não há mais espaço para crescimento.

Como o software intrinsecamente tem seu custo marginal de reprodução muito próximo de zero, a empresa dominante tem seu custo médio cada vez mais diminuído. Nesse caso, há economias de escala com a quantidade, mas que não vêm pelo lado da oferta, como na economia industrial. As economias de escala industriais classicamente se comportam como uma curva em forma de “U”. Conforme a escala aumenta, os custos se tornam menores, até que seja atingido um ponto de inflexão, onde tornam a crescer, mesmo que a demanda continue aumentando. Com o software isso não ocorre, pois praticamente não há custos variáveis de produção, e o custo médio continua a cair qualquer que seja a quantidade vendida. A economia de escala vem pelo lado da demanda: quanto mais o software é utilizado, maior a razão para que outras pessoas o utilizem.

A tarefa da empresa dominante, uma vez atingido o ponto de saturação, é em primeiro lugar tentar se reinventar. Novos produtos, sob forma de inovações e atualizações, devem ser oferecidos, uma vez que quase todo o mercado já foi conquistado e é necessário estimular os clientes a fazerem novas compras. Caso contrário, a receita da empresa seria estancada pelo simples fato de já não haver mercado para seus produtos. E em segundo lugar, deve evitar

que outros produtos iniciem o processo de realimentação positiva seguindo crescimento acelerado iniciado por ela mesma no passado.

Por tudo isso, embora a suposta incompatibilidade de formatos de arquivos e a suposta falta de suporte sejam as questões objetivamente apontadas como restritivas do ponto vista das externalidades de rede, são somente a ponta de um *iceberg*. Os gestores de TI resistem à adoção do software livre de forma generalizada porque sentem que se tomarem essa atitude, estarão deixando de fazer parte de uma rede muito forte com a qual estão acostumados a se relacionar, e têm confiança, para passarem a fazer parte de uma rede aparentemente fraca e desconhecida.

Mas a questão da permanência na rede não explica totalmente a questão. Além das vantagens de ficar numa rede estabelecida, existem os custos de sair dela. E todos os entrevistados apontaram uma questão que os incomoda a respeito da adoção de software: sentem-se aprisionados. O aprisionamento é configurado quando os custos de troca de um determinado produto, conjunto de produtos ou sistema é tão grande que torna-se economicamente inviável sua substituição por alternativas de concorrentes. O aprisionamento na maioria das vezes é consciente. O cliente, quando opta por uma determinada tecnologia, nem que seja para participar de uma rede já existente, sabe que a posterior mudança para outro padrão pode ser difícil. O que não impede certas surpresas, como o de um entrevistado que queixou-se por ter tido que assinar um contrato chamado *Software Assurance*, da Microsoft, para ter direito a atualizações futuras de produtos que ele já possuía. Quatro dos entrevistados criticaram a posição comercial da Microsoft, acusando-a de ser oportunista e de maltratar seus clientes. Apesar disso, o uso de software livre não é visto por nenhum dos entrevistados como forma de diminuir o nível de aprisionamento, pois consideram que mesmo utilizando-o serão vítimas dos mesmos problemas, mas desta vez em relação ao prestador de serviços contratado para fazer sua implementação e manutenção.

Das formas de aprisionamento sugeridas por Shapiro e Varian (1999, p. 140), nas entrevistas foram citadas as seguintes:

- Compras de bens duráveis – O software é considerado um bem durável, que tem um tempo normalmente em torno de três anos para depreciação. Como a maioria dos entrevistados tem a boa parte de seu parque de software já licenciado, seria necessário primeiro que esse investimento tivesse apenas um valor residual pós depreciação, para que a troca fosse financeiramente compensatória e justificável.

- Treinamento em marca específica – A questão da necessidade de treinamento foi citada, mas não foi considerada um fator forte de aprisionamento. Os entrevistados consideraram que este seria fácil e rápido para usuários finais devido à semelhança de interfaces, e fácil para os profissionais de informática por suas capacidades intrínsecas de aprendizagem de novas tecnologias. Além do mais, não haveria problemas de aceitação, uma vez que esta seria imposta, com apoio da diretoria da empresa. Embora os gestores de TI entrevistados tenham essa percepção, é notório que a aprendizagem de novas tecnologias é um custo muitas vezes subestimado. Como sugere o próprio estudo realizado para o desenvolvimento do MAT (Davis, 1989), esses são fatores que não podem ser desconsiderados, e que certamente trarão dificuldades durante o processo.
- Dados armazenados – Como dito anteriormente, a falta de compatibilidade com formatos proprietários é citada como fonte de aprisionamento, uma vez que há o temor de dificuldades na troca de arquivos entre parceiros de negócios que utilizem outros programas. Um fator importante de aprisionamento, que não foi citado mas ficou implícito nas falas dos entrevistados, foi também a possível dificuldade para a conversão de arquivos existentes para os formatos utilizados nos novos programas.

Outras fontes de aprisionamento citadas por Shapiro e Varian (1999), como o custo de encontrar novos fornecedores, e os custos da execução de testes e homologações, foram citados, mas considerados como pouco importantes para a decisão de não adoção. A participação de mercado atual do software livre pode ser explicada, então, como uma conjunção de dois fatores principais, na seguinte ordem de importância: externalidades de rede e custos de troca de tecnologia gerados por questões de aprisionamento.

7.2. INFLUÊNCIAS EXTERNAS

De acordo com as entrevistas realizadas nas empresas usuárias, foi possível perceber que estas confiam boa parte do seu suporte e de seu apoio em tecnologia da informação aos fornecedores de soluções. Nas empresas entrevistadas, foi verificado que muitas das soluções de software adotadas, tanto em infra-estrutura (sistemas operacionais, correio, bancos de dados), quanto em aplicativos de negócios, foram sugeridas e implementadas pelos prestadores de serviço em informática.

Essa capacidade de influência é exacerbada pelo diretor da empresa que comercializa soluções em software livre, que considera ser possível influenciar decisivamente seus clientes, com argumentações de modo a direcionar sua estrutura à utilização de software livre. Para justificar suas posições, ele cita a forma como tem trabalhado e como considera uma tarefa de sua empresa os esclarecimento das questões pertinentes ao software livre para seus clientes.

Suas percepções e argumentos, enquadrados nos construtos do MAT, são os seguintes:

Quadro 8 - Argumentação da empresa fornecedora de software livre enquadrada no MAT

Recursos

Atributos de usuários

Necessidade de treinamento do usuário	Considera necessário, mas simples devido à semelhança dos aplicativos.
Necessidade de treinamento da equipe de informática	Considera necessário e mais difícil e custoso, no entanto procura enfatizar que os resultados são muito melhores e não há necessidade de retreinamentos.

Suporte

Disponibilidade de recursos humanos para implementação e depois para a operação	Sua própria empresa, e os próprios funcionários do cliente por ela treinados são habilitados a fornecer o suporte.
---	--

Atributos do sistema

Participação de mercado	Vê a baixa participação do software livre como oportunidade de crescimento
Incompatibilidades técnicas com arquivos e padrões utilizados por líderes de mercado	Destaca que são poucas as incompatibilidades, e propõem-se a fazer demonstrações e adotar soluções alternativas para provar
Custo de migração	Os custos de migração seriam compensados por ganhos de economia e de eficiência
Tempo de migração	As migrações são cuidadosamente planejadas, e duram pouco tempo
Problemas técnicos de migração	Com o planejamento da migração esses problemas são mitigados
Mensuração da viabilidade do investimento	Facilmente mensurável, com economia de licenças e ganhos em estabilidade e desempenho
Funcionalidades implementáveis em estações	Há funcionalidades disponíveis para uma estação completa
Funcionalidades implementáveis em servidores	Estão disponíveis quaisquer tipos de funcionalidades para servidores
Disponibilidade de software	Existe software disponível para as mais diversas utilizações

Controle

Falta de propriedade (produtos "sem dono")	Não vê como um problema e não sente que os clientes vejam
Conhecimentos do gestor e da equipe de TI para controle do ambiente	O gestor de TI tem mais controle devido à previsibilidade e maior estabilidade do ambiente

Utilidade

Utilidade para o trabalho de administração de TI	Há ganhos em velocidade, produtividade e eficiência devido a diferenças nas características técnicas dos sistemas utilizados
Utilidade para o trabalho dos usuários	O usuários teriam as mesmas características de velocidade, produtividades e eficiência que têm com os sistemas proprietários

Facilidade

Facilidade de uso para a administração de TI	Considera que há dificuldades iniciais de aprendizagem por parte dos profissionais de informática, que são compensadas por ganhos com controle, flexibilidade e facilidade de utilização na operação diária
Facilidade de uso para os usuários	O usuários seriam facilmente treinados para utilizar softwares livres, uma vez que suas interfaces são muito parecidas com as alternativas hoje líderes de mercado

Essa empresa considera que as questões referentes às externalidades de rede, nas empresas de pequena estrutura de informática, podem ser vencidas principalmente sugerindo aos gestores de TI que ao adotarem softwares livres, e deixarem a rede predominante, estes fariam parte de outra rede também muito forte, com uma estrutura crescente e cada vez melhor. E a questão do aprisionamento seria vencida com argumentos de redução de custos associados a aumento de funcionalidades em curto prazo. Isso traria como consequência a mudança de atitude dos gestores de TI, que aceitariam melhor as soluções em software livre, e fariam com que sua intenção de uso fosse a máxima possível. Como podemos verificar no diagrama do MAT abaixo, modificado para refletir essa situação:

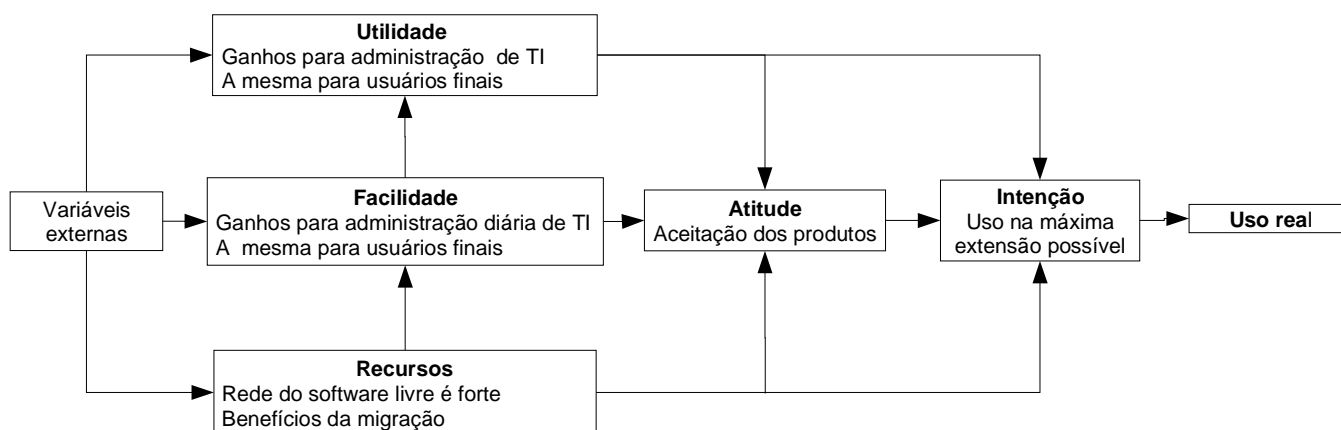


Figura 6 - MAT com destaques para representar a visão da empresa de software livre

Embora nenhuma das empresas entrevistadas tenha fornecedores que defendam o uso de software livre nessa extensão, pode-se dizer que seu uso atual, constatado nas entrevistas, tem grande influência de fornecedores de serviços que viram nesse tipo de programa uma alternativa utilitária, principalmente para a estrutura de Internet. O diretor da empresa fornecedora de software livre enfatizou que os preços dos serviços envolvendo esse tipo de tecnologia, por utilizarem componentes estratégicos e mais sofisticados, são mais caros e mais rentáveis. Em sua opinião, essa característica aliada ao crescimento da adoção governamental, fará com que surjam mais fornecedores de tecnologia defensores do software livre num futuro próximo, e que a influência que têm sobre seus clientes será decisiva para o aumento de sua utilização.

Entretanto, embora as empresas entrevistadas possam começar a sofrer influências de fornecedores comprometidos majoritariamente com software livre, não é isso o que atualmente acontece. Foi possível perceber que os fornecedores de serviços das empresas entrevistadas procuram oferecer soluções a respeito das quais têm bons conhecimentos, o que pela própria situação de participação de mercado acaba levando a produtos proprietários. Pode-se dizer que, na amostra pesquisada, os fornecedores que mais influenciam esse tipo de escolha são os que trabalham com sistemas de gestão empresarial. Das cinco empresas entrevistadas, quatro utilizam sistemas de gestão que foram desenvolvidos para ser executados, tanto nos servidores quanto nas estações, com sistemas operacionais da linha Microsoft Windows. Embora o diretor da empresa de software livre sustente que é possível adotá-lo na maior parte do ambiente, mesmo com esse tipo de restrição, o caminho natural para as empresas – reforçando a questão da participação nas redes estabelecidas – é utilizarem a plataforma exigida pelo sistema de gestão adotado.

De acordo com o levantamento realizado junto às empresas que trabalham prioritariamente com software proprietário, descrito na seção 6.4, seu principal discurso é voltado às macro políticas governamentais, e à defesa ferrenha da liberdade de escolha de tecnologia tanto dentro do poder público quanto nas empresas privadas. No entanto, é possível perceber uma grande preocupação quanto ao risco da adoção de software livre de forma generalizada, devido à descrença da sustentabilidade econômica do modelo adotado pelas empresas que trabalham majoritariamente com esse tipo de tecnologia. Dentro do MAT, de acordo com as informações obtidas, a visão geral da indústria de software proprietário pode ser enquadrada da seguinte forma:

Quadro 9 - Visão das empresas de software proprietário, enquadrada ao MAT

Recursos

Atributos de usuários

Necessidade de treinamento da equipe de informática	Treinamento difícil e custoso, devido à necessidade de maior especialização
---	---

Suporte

Disponibilidade suporte	Descrença na sustentabilidade do modelo das empresas de software livre, o que pode gerar crise no fornecimento de suporte e parada nos projetos em desenvolvimento
-------------------------	--

Atributos do sistema

Custo e tempo de migração	Migração difícil e custosa, devido às soluções serem pouco amigáveis
Mensuração da viabilidade do investimento	Não há ganhos adicionais além da economia com licenças
Funcionalidades implementáveis	Menos funcionalidades que os produtos similares proprietários
Disponibilidade de software	Não existe software livre para funções chave nas empresas, principalmente sistemas de gestão e para condução de negócios

Controle

Falta de propriedade (produtos "sem dono")	Críticas ao suporte fragmentado e dependente da ajuda de pessoas físicas
--	--

Utilidade

Utilidade para o trabalho de administração de TI	Não há ganhos significativos em termos de velocidade, produtividade e eficiência
Utilidade para o trabalho dos usuários	O usuários teriam as mesmas características de velocidade, produtividades e eficiência que têm com os sistemas proprietários

Facilidade

Facilidade de uso para a administração de TI	Grandes dificuldades de treinamento e aprendizagem para utilização de soluções livres
Facilidade de uso para os usuários	O usuários seriam facilmente treinados para utilizar softwares livres, uma vez que suas interfaces são muito parecidas com as alternativas hoje líderes de mercado

Com isso, podemos verificar que essas empresas apostam muito na continuidade e no aprofundamento da rede proprietária existente, e avaliam que o aprisionamento atual tornaria os custos de troca insuportáveis, tanto para as empresas usuárias quanto para elas próprias, e que exigiria investimentos de adequação, substituição de plataformas e de ferramentas de desenvolvimento. Entretanto, algumas empresas desenvolvedoras de software proprietário estão preparando-se para uma eventual mudança no mercado, através da modificação de

seus produtos para que possam ser utilizados em plataformas livres. Um exemplo é a empresa Microsiga⁴⁷, que modificou seus sistemas de gestão empresarial para que possam também ser executados em bancos de dados livres.

O diagrama do MAT, destacado em acordo com as variáveis mais expressivas descritas acima, teria a seguinte forma:

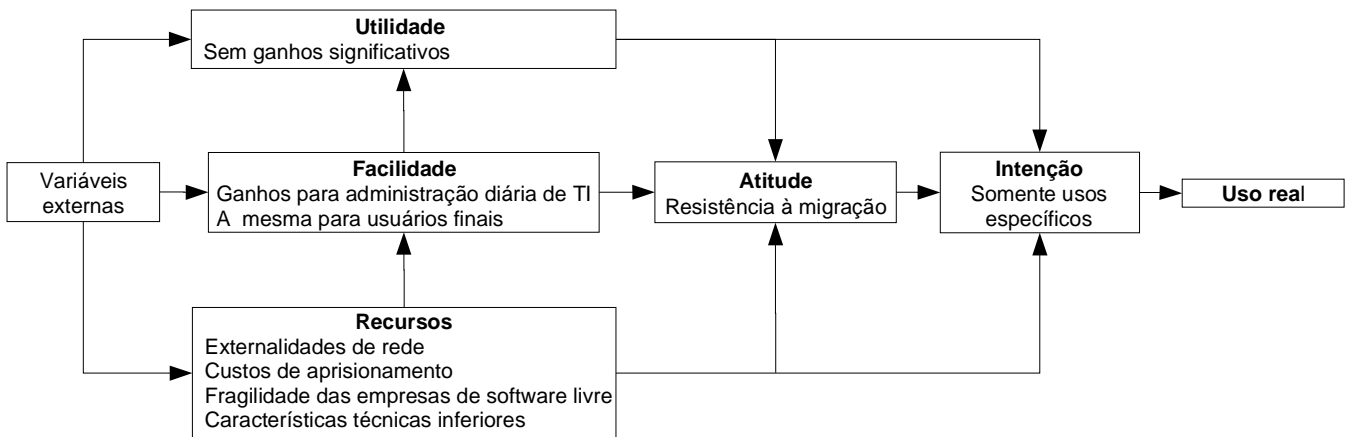


Figura 7 - MAT com destaques para representar a visão das empresas de software proprietário

Um último foco de influência externa para as empresas entrevistadas é a comparação de suas decisões com aquelas tomadas por outras empresas. Foi possível perceber, em boa parte das entrevistas, que os gestores de TI das empresas sentem-se ansiosos por saber os rumos que o mercado está tomando, e que tipos de soluções outras empresas estão adotando. Embora isso não sirva diretamente para explicar a situação atual de uso de software livre nessas empresas, sem dúvida configura-se como influência decisiva. Por isso, foi descrito na seção 6.2 o processo de adoção de software livre de uma grande empresa do setor de serviços.

No quadro abaixo estão os fatores que levaram essa empresa a tomar essa decisão, enquadrados nos construtos do MAT.

⁴⁷Veja em <<http://www.microsiga.com.br>>

Quadro 10 - Fatores para adoção de software livre pela grande empresa de serviços, enquadrados no MAT

Recursos

Atributos de usuários

Necessidade de treinamento do usuário	Considera necessário, mas simples devido à semelhança dos aplicativos
Necessidade de treinamento da equipe de informática	Considera necessário e mais difícil e custoso, mas afirma estar obtendo bons resultados

Suporte

Disponibilidade de recursos humanos para implementação e depois para a operação	Alguns serviços são contratados junto a fornecedores externos, mas a maior parte das implementações e do suporte são realizados pela própria equipe interna
---	---

Atributos do sistema

Participação de mercado	Não considera a baixa participação de mercado um problema, pois vê a rede em torno do software livre como suficientemente forte
Incompatibilidades técnicas com arquivos e padrões utilizados por líderes de mercado	Cita testes internos e uso diário para sustentar que são poucas as incompatibilidades, e que não comprometem o trabalho diário
Custo de migração	Os custos de migração estão sendo compensados por ganhos de economia e de eficiência
Tempo de migração	As migrações são cuidadosamente planejadas, e têm sido feitas dentro dos prazos planejados
Problemas técnicos de migração	Com o planejamento da migração esses problemas são mitigados
Mensuração da viabilidade do investimento	Facilmente mensurável, com economia de licenças e ganhos em estabilidade e desempenho
Funcionalidades implementáveis em estações	Há funcionalidades disponíveis para uma estação completa, que já foi testada e homologada
Funcionalidades implementáveis em servidores	Estão disponíveis quaisquer tipos de funcionalidades para servidores
Disponibilidade de software	Existe software disponível para as mais diversas utilizações. Os que não estão disponíveis, e referem-se diretamente ao negócio, são desenvolvidos internamente.

Controle

Falta de propriedade (produtos "sem dono")	Não vê como um problema
Conhecimentos do gestor e da equipe de TI para controle do ambiente	A gestão de TI tem mais controle do ambiente devido à maior estabilidade e possibilidade de personalizações

Utilidade

Utilidade para o trabalho de administração de TI	Há ganhos em velocidade, produtividade e eficiência devido a diferenças nas características técnicas dos sistemas utilizados
Utilidade para o trabalho dos usuários	O usuários têm as mesmas características de velocidade, produtividades e eficiência que com os sistemas proprietários

Facilidade

Facilidade de uso para a administração de TI	Considera que há dificuldades iniciais de aprendizagem por parte dos profissionais de informática, que são compensadas por ganhos com controle, flexibilidade e facilidade de utilização na operação diária
Facilidade de uso para os usuários	O usuários estão sendo treinados para utilizarem softwares livres, sendo que há resistências que são vencidas com um trabalho de convencimento

Uma característica interessante dessa empresa é que ao contrário dos entrevistados da amostra principal, o gestor de TI considera que a rede em torno do software livre é forte o suficiente para justificar sua utilização na empresa, e que os problemas gerados pelo abandono da rede dominante não são significativos. Considera até mesmo que seus funcionários hoje fazem parte de uma comunidade maior, que atende a questões que vão muito além da realidade da própria empresa. Adicionalmente, a empresa considera que os custos de troca por aprisionamento são muito menores que os benefícios gerados pela adoção de software livre, que o gestor prova citando o quão criteriosos são na análise de investimentos, e mostrando a substituição de praticamente todos os servidores, e o processo de substituição das estações, que está em andamento.

O diagrama do MAT, com destaque para os principais pontos nessa empresa, é o seguinte:

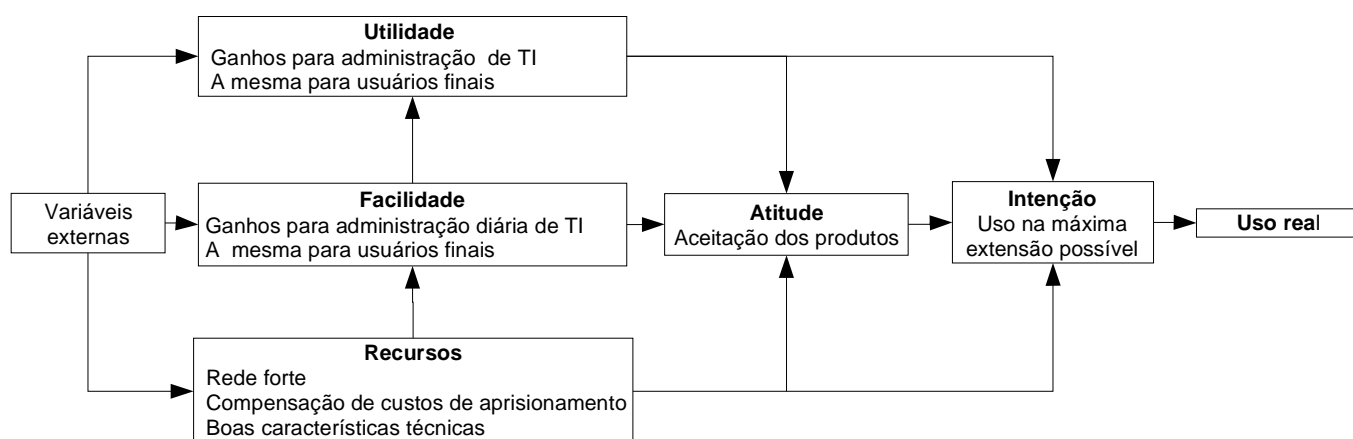


Figura 8 - MAT com destaques para representar a visão da grande empresa de serviços

7.3. PROCESSO DE ADOÇÃO DE TECNOLOGIA

O processo de adoção de uma nova tecnologia, produto ou padrão é descrito por Webster (2001), como uma curva normal. Esse autor sustenta que a taxa de difusão da nova tecnologia é determinada basicamente por duas variáveis: o retorno que a inovação oferece ao potencial adotante em relação ao investimento necessário; e o investimento necessário em relação aos ativos totais da empresa. Webster (2001) categorizou as empresas adotantes em cinco estratos, cada um deles proveniente de um desvio padrão da curva. A um desvio padrão do centro, está a maior parte dos usuários da nova tecnologia. São chamados de “maioria inicial” e “maioria tardia”, de acordo com o período de tempo onde fazem sua adoção. A dois desvios padrão, estão os “adotantes iniciais” e os retardatários”. E a dois desvios em direção ao início dos períodos de tempo estão os inovadores. Como as compras empresariais são totalmente dependentes da composição e da dinâmica do centro de compras (Shapiro, 1995; Brierty, 1998), e como o gestor de tecnologia nas empresas de pequena estrutura de TI tem papel fundamental dentro dele, podemos inferir que essa curva mostra muito de seu comportamento em relação à tecnologia.

Essa idéia pode ser representada através da figura a seguir:

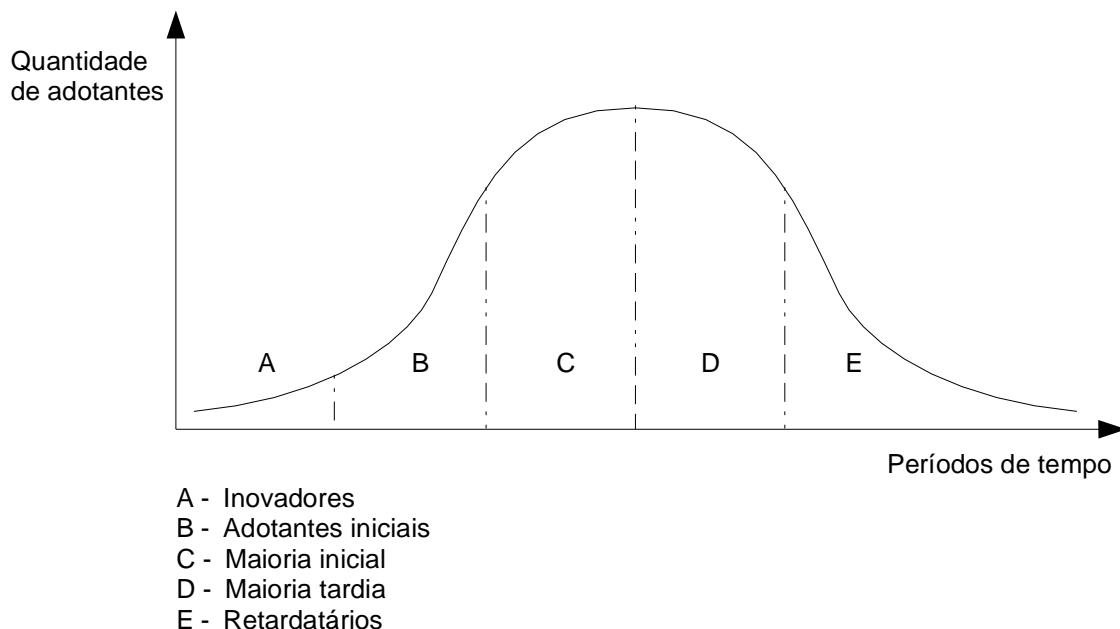


Figura 9 - Ciclo de vida de adoção de tecnologia

Moore (1996) prossegue com a descrição desse modelo, definindo da seguinte forma cada um dos estratos:

1. Entusiastas – Fortemente comprometidos com o uso e a experimentação de novas tecnologias. São principalmente experimentadores, que buscam as novidades em si, e normalmente são os primeiros a utilizarem os novos produtos. No caso do software livre, são principalmente técnicos e desenvolvedores, que buscaram a novidade por curiosidade técnica ou para resolver algum problema específico. Essas pessoas, embora sejam capazes tecnicamente, respeitados em suas posições, capazes de influenciar os demais, muitas vezes não têm poder de compra, e não são capazes de diretamente gerar novos negócios.

2. Adotantes iniciais – Compõem o primeiro grupo forte de adoção do novo paradigma, e enxergam formas de beneficiar os negócios com sua utilização. Representam o primeiro mercado consumidor forte para a tecnologia emergente, freqüentemente são responsáveis por sustentar as empresas que as fornecem e, através de seu entusiasmo, ajudam a divulgar os novos produtos. Essas pessoas patrocinam a utilização das novas tecnologias porque acreditam que o custo de mudança pode trazer-lhes benefícios superiores àqueles obtidos por fazer parte da rede e do paradigma dominantes.

Esses dois primeiros estratos representam o mercado inicial para a nova tecnologia: os entusiastas tentando explorar, e os adotantes tentando aproveitar suas novas vantagens técnicas e econômicas. No entanto, devido às questões de externalidades de rede, discutidas nas seções anteriores, esses são os únicos grupos que têm o interesse de serem os primeiros. Os demais desejam ver a novidade funcionando, e muito bem aceita, antes de iniciar o processo.

3. Maioria inicial – Essas pessoas, que formam o estrato mais importante e mais lucrativo do ciclo de adoção, não estão interessados na tecnologia em si. Elas estão interessadas apenas em fazer com que os sistemas de suas empresas funcionem de forma a atingirem seus objetivos de negócios. Qualquer coisa fora disso é automaticamente descartada. E embora Moore(1996) – que os chama de pragmáticos – não cite explicitamente a questão da inserção na rede, mostra sua problemática ao descrevê-los:

“Os pragmáticos são as pessoas mais passíveis de serem responsáveis pelos sistemas cruciais à missão de uma empresa. [...] Quando finalmente fazem a mudança, os pragmáticos preferem comprar do líder do mercado por duas razões. Primeira, todo o restante do mercado torna seus produtos compatíveis com o do líder. [...] Segunda, o líder atrai muitas empresas de terceiros a seu pós mercado,

de modo que, ainda que o líder não responda prontamente às solicitações do cliente, o mercado, como um todo, o fará.

4. **Maioria tardia** – Não esperam obter benefícios com a tecnologia, e só a adotam quando não têm mais opção, e estão arriscados a ficarem de fora de uma rede forte que está se consolidando. Geralmente são sensíveis a preços, e enxergam a tecnologia da informação simplesmente como um custo, e não como uma função estratégica para a empresa. No entanto, podem ser abordados com produtos focados para suas necessidades específicas, com apelos diretos a suas realidades de negócio, sem enfoque tecnológico e com preços convidativos. Conseguir formas lucrativas de atender a esse mercado pode representar uma extensão importante ao mercado inicial, e uma forma de extrair renda de produtos aparentemente exauridos.

5. **Retardatários** – São empresas cujos decisores não vêem benefício nenhum no uso da tecnologia, e trabalham contra sua utilização.

O ideal de sucesso para uma nova tecnologia introduzida no mercado é ganhar a aceitação dos entusiastas, que servirão para introduzi-la aos adotantes iniciais, que por sua vez servirão de boas referências à maioria inicial, de onde virá a maior parte dos lucros e serão fixados os padrões “de fato”. Ou seja, esse é o caminho para formação de uma nova rede. Quando o mercado começa a atingir a maioria inicial, naturalmente é iniciado o processo de realimentação positiva, responsável pelo aumento exponencial do valor de ser participante dessa rede. E a partir daí o mercado pode ser ampliado, com o ataque à maioria tardia através de personalizações, simplificações e barateamento de produtos.

Moore (1996), no entanto, mostra que essa dinâmica nem sempre acontece, porque existem grandes dificuldades na transposição do mercado inicial, representado por inovadores e adotantes iniciais, para o mercado de maioria, indicado na Figura 9 pela letra “C”. Segundo o autor, o problema é que “esses dois grupos, embora adjacentes no ciclo de vida da adoção, diferem tanto em termos de valores subjacentes, que a comunicação entre eles é quase impossível”. Compara-os, então, da seguinte forma:

Quadro 11 - Comparação entre adotantes iniciais e a maioria inicial – adaptado de Moore (1996 p. 30)

Adotantes iniciais	Maioria inicial
Intuitivos	Analíticos
Apóiam a revolução	Apóiam a evolução
Contrários ao status quo	Conformistas
Admitem sair da rede dominante	Desejam pertencer à rede dominante
Tomam decisões independentes	Fazem consultas para tomar decisões
Assumem riscos	Administram riscos
Motivados por oportunidades futuras	Motivados por problemas presentes
Buscam o possível	Buscam o provável

Há um abismo entre o mercado inicial, formado pelos entusiastas e pelos primeiros adotantes, que não se importam com a questão na participação na rede; e o restante do mercado, para quem a participação na rede é muito importante. A transposição desse abismo, apesar de ser difícil, é fundamental para que a nova tecnologia se consolide e forme uma rede forte, que tão importante é para seu desenvolvimento. Ainda segundo Moore (1996), a única forma de fazê-lo é adotar uma estratégia focada na busca de nichos, “com o conjunto mínimo de produtos e serviços necessários para assegurar que o cliente-alvo atingirá sua razão irresistível para comprar”. E se esses nichos específicos forem conquistados com produtos que os satisfaçam, nichos adjacentes podem ser atendidos, e progressivamente generalizados para chegar cada vez mais perto do mercado de massa.

Todos os gestores de TI entrevistados têm justamente o perfil da maioria inicial, e prezam muito a participação numa rede consolidada. No entanto, foi possível perceber que soluções em software livre já conseguiram transpor um dos abismos, para atender ao nicho específico de infra-estrutura para acesso à Internet, uma vez que todos os entrevistados que o utilizam reconhecem sua capacidade, estão satisfeitos e tecem elogios. Mas o abismo ainda não foi transposto para as demais categorias.

Como visto nas seções anteriores, o maior problema relacionado à não adoção de softwares livres é realmente a participação na rede, uma vez que os custos de troca por aprisionamento são colocados como secundários. Por isso, a chave para que a participação geral de mercado dos softwares livres aumente, é o desenvolvimento de produtos completos que ataquem nichos adjacentes aos já atualmente conquistados, de forma a criar massa crítica para a ampliação da rede e incentivar o uso por parte dos clientes pragmáticos.

No caso de um dos entrevistados, que pretende adotar o software livre em todo o ambiente a partir de 2004, a necessidade de redução de custos com licenciamento de software representa, conforme a terminologia de Moore (1996), o “motivo irresistível” para a troca. Para a grande empresa de serviços entrevistada, que já está fazendo a migração de toda a estrutura, os motivos irresistíveis foram, além da redução de custos de licenciamento, a segurança, a possibilidade de personalização, a possibilidade de redução de custos de manutenção e o reaproveitamento do hardware antigo existente.

8. CONSIDERAÇÕES FINAIS

O motivador inicial deste trabalho foi a constatação de que, apesar de estarem disponíveis soluções em software livre para grande parte das necessidades empresariais, e apesar de existirem indicadores que sugerem vantagens concretas em termos de custos e funcionalidades, sua utilização é expressivamente menor que as alternativas proprietárias. Por isso, a objetivo principal da pesquisa foi entender os fatores que condicionam a adoção de tecnologias de informação nas empresas, e tentar descobrir as razões da baixa participação de mercado.

Para isso, foi utilizado o Modelo de Aceitação de Tecnologia (MAT), que possibilita fornecer uma explicação dos determinantes da utilização de computadores, capaz de contemplar comportamentos de usuários através de uma grande gama de tecnologias e populações. O modelo leva em consideração que estímulos externos influenciam as atitudes pessoais, indiretamente influenciando suas crenças sobre as conseqüências de ter aquele comportamento. Foi formulado para atingir esses objetivos através da identificação de um pequeno número de variáveis fundamentais sugeridas por pesquisas prévias, relacionadas com determinantes cognitivos e afetivos da aceitação de tecnologia da informação. Foram desenvolvidas, então, uma série de proposições, que através da literatura existente, buscaram explicar as razões para a baixa utilização do software livre. As proposições levantadas foram associadas ao MAT, formando assim o corpo que nortearia o restante da pesquisa.

Para a coleta dos dados, foi utilizada a metodologia qualitativa, de modo que fosse possível compreender o fenômeno apresentado utilizando a percepção das pessoas envolvidas nesse processo, através de suas opiniões, visões e conceitos. Esse tipo de metodologia permite captar essas nuances, propicia uma visão global do tema e traz à tona as questões pertinentes.

Como a bibliografia sobre o assunto sugere a existência de diferenças significativas no processo de adoção de tecnologias de acordo com o porte da empresa, a amostra foi limitada a organizações com pequena estrutura de informática, de 30 a 160 teclados. Além disso, para que fosse possível ter uma compreensão mais ampla do processo, foi entrevistada uma empresa que trabalha somente com soluções em software livre, uma associação de empresas que trabalham prioritariamente com software proprietário e uma empresa grande usuária de software livre, cujo porte ultrapassa os 160 teclados da amostra inicial.

Essas entrevistas foram analisadas utilizando o MAT como apoio, de modo a verificar as proposições inicialmente levantadas. Curiosamente, a maior parte das proposições foi descartada, por terem demonstrado ser pouco relevantes na explicação do fenômeno estudado.

Alguns pontos objetivos, presentes na proposições iniciais, foram recorrentemente citados pelo entrevistados, como o temor de incompatibilidade com arquivos ou padrões, e a falta de suporte técnico para a operação diária. Mas independentemente dessas questões específicas, foi possível perceber a vontade, por parte dos gestores de TI, de sempre utilizar tecnologias aceitas pela maior parte do mercado. As empresas usuárias querem estar na rede dominante, e usufruir de suas externalidades. As redes da economia da informação são formadas ao redor de tecnologias que crescem através do sistema de realimentação positiva: quanto mais usuários, maior é seu valor, e quanto maior o valor, mais usuários querem participar da rede. Por isso, em mercados onde é possível observar o movimento de realimentação, existe a tendência de que uma determinada tecnologia ou empresa se torne dominante, e atue no mercado com tal força que a entrada de novos concorrentes seja muito dificultada. Os gestores de TI resistem à adoção do software livre de forma generalizada porque sentem que se tomarem essa atitude, estarão deixando de fazer parte de uma rede muito forte, com a qual estão acostumados a se relacionar e têm confiança, para passar a fazer parte de uma rede aparentemente fraca e desconhecida.

Mas a questão da permanência na rede não explica totalmente a questão. Além das vantagens de ficar numa rede estabelecida, existem os custos de sair dela. E na pesquisa ficou clara uma questão que os incomoda na adoção de software: sentem-se aprisionados, pois os custos de troca dos produtos atualmente utilizados é considerado tão grande que torna-se economicamente inviável sua substituição por alternativas concorrentes. Os fatores de aprisionamento mais importantes encontrados foram os seguintes: compras de bens duráveis (licenças de software proprietário) já realizadas e ainda não depreciadas; treinamentos e conhecimentos já adquiridos em marcas específicas; e o temor da impossibilidade de utilização de dados já existentes devido a incompatibilidades. Surpreendentemente, o custo da migração em si não foi considerado como um fator dificultador importante.

A participação de mercado atual do software livre pode ser explicada, então, como uma conjunção de dois fatores principais: necessidade de usufruto de externalidades

proporcionadas pela rede dominante; e pelos altos custos de troca de gerados por questões de aprisionamento à tecnologia existente.

E ao mesmo tempo, nesse cenário, outras forças atuam para influenciar esse mercado.

As empresas de software proprietário opõem-se fortemente à adoção majoritária do software livre pelo governo. Através de argumentos que grandemente passam pela inexistência de uma política governamental clara de desenvolvimento da tecnologia da informação, essas empresas temem que o governo passe a utilizar e a fomentar o software livre, e que essa atitude passe a influenciar o mercado como um todo. Para essas empresas, além de ser não ser interessante que o mercado passe a preferir e exigir software livre, é ruim também que sejam obrigadas a desenvolver software proprietário sobre plataformas livres. Nesse caso, destacam a existência de grandes custos de aprisionamento, e os altos custos que teriam para adequar seus processos de produção a novas plataformas. Além disso, não acreditam que o modelo do software livre, que não demanda a venda de licenças, possa sustentar as empresas prestadoras de serviço, temendo que isso possa trazer certa desestabilização do mercado. A influência dessas empresas é um fator muito importante na amostra pesquisada, pois nelas estão incluídas as fornecedoras de sistemas de gestão, que tem presença marcante e grande poder de influência.

E de outro lado, estão as empresas que trabalham prioritariamente com software livre, e que têm a expectativa de uma forte adoção desse tipo de tecnologia por parte das empresas. A empresa de software livre entrevistada exacerba a capacidade de influência que tem sobre sobre as decisões de adoção de tecnologia de seus cliente, e demonstra um grande otimismo com esse mercado. E em sua opinião, o crescente uso por parte do governo fará com que a rede do software livre seja expandida, e os clientes pragmáticas passem a demandar por sua utilização.

A maior limitação dessa pesquisa é a impossibilidade de generalização dos resultados obtidos. Como a amostra é não probabilística, e tomada por julgamento, ela não é representativa do universo de empresas usuárias de software. No entanto, os resultados da pesquisa explicam vários pontos chave para a compreensão do fenômeno do uso do software livre, que embora tenham validade restrita à amostra, trazem à tona indicadores que podem ser utilizados em estudos futuros.

8.1.ESTUDOS FUTUROS

8.1.1.MARKETING

A utilização futura do software livre, e a evolução de sua participação nos mercados de informática, certamente deve ser acompanhada nos próximos anos. No entanto, alguns tópicos demandam maior atenção.

Existe uma vasta bibliografia sobre marketing empresarial geral e marketing empresarial tecnológico. Dentre eles podemos citar as obras de Webster (1991), Brierty (1998), Anderson e Narus (1999) e principalmente Shapiro e Varian (1999) e Moore (1996), que tratam especificamente sobre tecnologia da informação. Nessa bibliografia são feitas várias análises, das quais uma das mais interessantes é a descrição do ciclo de vida do produto, com suas fases de introdução, crescimento, maturidade, saturação e declínio, com margens de lucro variantes em cada um dos estágios.

Esses autores consideram os produtos vendidos pela empresas necessariamente como uma propriedade e um segredo competitivo do produtor. No entanto, no mundo do software livre, o software em si não pode ser considerado o produto a ser comercializado, porque não é possível cobrar por sua posse e por seu uso. O conceito de produto em software livre deve, então, ser necessariamente estendido para um conjunto entre software, serviços e eventualmente outros tipos de acessórios.

Esse conceito não é totalmente contemplado pela literatura, porque abrange uma gama de produtos estendidos fornecidos por várias empresas, onde todos um núcleo principal comum: um determinado software livre. Por isso, seria interessante estudar como as empresas prestadoras de serviços e produtoras de software livre montam sua carteira de produtos, como são comercializados, como lidam com parceiros e concorrentes, e como é sua rentabilidade.

8.1.2.ECONOMIA

Outro tema é a economia do desenvolvimento do software livre. Seria necessária a análise, sob o aspecto econômico, do processo de desenvolvimento, com ênfase na cobertura de seus custos, nos benefícios obtidos e nas formas utilizadas para sustentar o modelo. Isso pode ser obtido através de um estudo de caso, que analise uma empresa que esteja presente no mercado como a maior patrocinadora de algum produto livre.

8.1.3. RECURSOS HUMANOS

O desenvolvimento de software livre é, inegavelmente, um notável exemplo de organização em grupo. Milhares de pessoas ao redor do mundo desenvolvem e trabalham nesses programas, algumas vezes motivados por necessidades empresariais, outras vezes motivados por interesses próprios. O fato é diversas comunidades, com nível maior ou menor de organização formam-se com participantes de todas as partes do mundo, e coordenam-se para produzir programas notavelmente sofisticados.

Poderia ser realizado um estudo para entender como comunidades tão dispersas conseguem coordenar-se sem contato pessoal, e sem uma relação formal de subordinação, para desenvolver atividades tão complexas.

E assim como existem projetos bem sucedidos, a exemplo do Linux, do KDE ou do Apache, vários outros não conseguiram seguir adiante. Pode-se fazer uma comparação entre a coordenação e atitude dos membros desses projetos, e conseguir uma relação de fatores de sucesso para projetos livres, que eventualmente podem ter seus conceitos aproveitados para uso em projetos empresariais.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALMEIDA, Álvaro Henrique Teixeira de. Sentença do processo 99.001.122036-4 da 12ª vara cível da cidade do Rio de Janeiro, sobre violação de direitos autorais de software. 8 de fevereiro de 2002 *in* LEMOS, Ronaldo; WAISBERG, Ivo. Org. Conflitos sobre nomes de domínios. Rio de Janeiro: Editora Revista dos Tribunais, 2002, p. 327.
- ANDERSON, James C. & NARUS, James A. *Business Marketing Management*. New Jersey: Prentice Hall, 1999
- BARDIN, Laurence. *Análise de conteúdo*. Trad. Luís Antero Reto e Augusto Pinheiro. Lisboa: Edições 70, 1995.
- BEHLENDORF, Brian. *Open Source as a Business Strategy in Open Sources: Voices from the Open Source Revolution* - O'Reilly & Associates Inc., 1999.
- BENAMATI, John & RAJKUMAR, T. M. *The application development outsourcing decision: An application of the technology acceptance model* .The Journal of Computer Information Systems; Stillwater, volume 42, issue 4, pág. 35, 2002.
- BEZROUKOV, Nikolai. *Open Source Software Development as a Special Type of Academic Research: Critique of Vulgar Raymondism*. First Monday Peer Reviewed Journal. Outubro de 1999. Vol. 4, número 10. Disponível em <http://firstmonday.org/issues/issue4_10/bezroukov/index.html>.
- BLUM, Renato M. S. Opice; ABRUSIO, Juliana Canha. Direito Autoral Eletrônico *in* LEMOS, Ronaldo; WAISBERG, Ivo. Org. Conflitos sobre nomes de domínios. Rio de Janeiro: Editora Revista dos Tribunais, 2002, p. 289.
- BOZMAN, Jean et al. *Windows 2000 versus Linux in enterprise computing: an assessment of business value for selected workloads*. Framingham: IDC, 2002 22 p.. (Relatório de pesquisa)
- BRASIL, Lei nº 9.609, de 19 de fevereiro de 1998. Dispõe sobre a proteção da propriedade intelectual de programa de computador, sua comercialização no País, e dá outras providências

- BRASIL, Lei nº 9.610, de 19 de fevereiro de 1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.
- BRETTTHAUER, David. *Open source software: A history*. Information Technology and Libraries. Março de 2002. Vol. 21 pág. 3
- BRETTTHAUER, David. *Open source software: A history*. Information Technology and Libraries. Volume 21 nº 1 pág. 3, 2002.
- BRIERTY, Edward. Business Marketing. 3ª ed. New Jersey: Prentice Hall, 1998.
- BRUNAZO FILHO, Amílcar. *Critérios para Avaliação da Segurança do Voto Eletrônico*- Artigo apresentado em 06 de Março de 2001 no Workshop em Segurança de Sistemas Computacionais - Wseg'2001 UFSC, Florianópolis, SC, Brasil. Disponível em <<http://www.brunazo.eng.br/voto-e/textos/Wseg2001.htm>>. Acesso em dezembro de 2003.
- CASTELLUCIO, Michael. *Can the enterprise run on free software?* Strategic Finance. Volume 81 nº 9 pág. 50, 2000.
- CERDEIRA, Pablo Camargo. Obras livres: Os bens imateriais da propriedade intelectual pós-Internet. Revista Consultor Jurídico, disponível em: <<http://conjur.uol.com.br/textos/21487>>. Acesso em setembro de 2003.
- CHAU, Patrick Y. K. & HU, Paul Jen-Hwa. *Information technology acceptance by individual professionals: A model comparison approach*. Decision Sciences. Volume 32 nº 4 pág. 699, 2001.
- CHAU, Patrick Y. K. *An empirical assessment of a modified technology acceptance model*. Journal of Management Information Systems. Volume 13 Issue 2 Pág. 185-204, 1996.
- CHAU, Patrick Y. K. *Influence of computer attitude and self-efficacy on IT usage behavior*. Journal of End User Computing. Volume:13 Issue 1 Pág. 26-33, 2001.
- COMPEAU, Deborah R & HIGGINS, Christopher. *A Computer self-efficacy: Development of a measure and initial test*. MIS Quarterly. Volume 19 Issue 2 Pág. 189, 1995.
- COYLE, Karen. *Open source, open standards*. Technology and Libraries. Março de 2002. Vol. 21, pág. 33

- DAVIS, Fred D. *A technology acceptance model for empirically testing new end-user information systems: theory and results*. Tese de doutorado. Sloan School of Management, Massachusetts Institute Technology, Cambridge, EUA, 1985.
- DAVIS, Fred D. *Perceived ease of use, and user acceptance of information technology*. MIS Quarterly, 13(3), 319-338, 1989.
- DISSEL, Han van & KREMERS, Mark. *ERP system migrations*. Association for Computing Machinery. Communications of the ACM. Volume 43 nº 4 pág. 52, 2000.
- DOLL, William J. & HENDRICKSON, Anthony & DEANG, Xiaodong. *Using Davis's perceived usefulness and ease-of-use instruments for desision making: A confirmatory and multigroup invariance analysis*. Decision Sciences; Atlanta. Volume 29 Issue 4 Pág. 839-869, 1998.
- FALDETTA, Guglielmo. *The content of freedom in resources: The open source model*. Journal of Business Ethics. Volume 39 Issue: 1/2 Pág. 179, 2002.
- FERNANDES, Aguinaldo Aragon. *Gerência estratégica da tecnologia da informação: obtendo vantagens competitivas*. Rio de Janeiro: Livros técnicos e científicos editora, 1992.
- GAHTANI, Said Al. *The applicability of TAM outside North America: An empirical test in the United Kingdom*. Information Resources Management Journal. Volume14 Issue 3 Pág. 37-46, 2001.
- GIL, Antonio Carlos. *Projetos de pesquisa*. 3ª ed. São Paulo: Atlas, 1991.
- GOODE, William J. & HATT, Paul K. *Métodos em pesquisa social*. 3ª ed. São Paulo: Nacional, 1969.
- HIPPEL, Eric von. *Innovation by user communities: Learning from open-source software*. MIT Sloan Management Review; Cambridge. Volume 42 Issue 4 Pág. 82, 2001.
- HIRANO, Seidi - organizador. *Pesquisa social: projeto e planejamento*. São Paulo: T.A. Queiroz, 1979.
- IDC. *Linux Servers Gain Shares*. Publicado em abril de 2000. Disponível em <<http://www.idc.com/getdoc.jhtml?containerId=itf20000411>>. Acesso em novembro de 2003.

- KENWOOD, Carolyn A., *A Business Case Study of Open Source Software in* http://www.mitre.org/support/papers/tech_papers_01/kenwood_software/index.shtml, julho de 2001.
- KRIPPENDORFF, Klaus. *Content Analysis: an introduction to its methodology*. Beverly Hills: Sage publications, 1986.
- KRIPPENDORFF, Klaus. *Metodología de análisis de contenido: teoría y práctica*. Trad. Leandro Wolfson. 1º ed. Barcelona: Ediciones Paidós, 1990.
- LEITE, José Alfredo Américo. *Metodologia de elaboração de teses*. São Paulo: Mc Graw Hill do Brasil, 1978.
- MALHOTRA, Naresh K. *Pesquisa de marketing: uma orientação aplicada*. Trad. Nivaldo Montingelli Jr. e Alfredo Alves de Farias. 3º ed. Porto Alegre: Bookman, 2001.
- MARTIN, E. Wainright. *Managing Information Technology: what managers need to know*. 2º ed. New Jersey: Prentice-Hall, 1994.
- MATHIESON, Kieran & PEACOCK, Eileen & CHIN, Wynne. *Extending the technology acceptance model: The influence of perceived user resources*. Database for Advances in Information Systems; New York.. Volume 32, Issue 3 , pg. 86-112, 2001
- MCKUSICK, Marshall Kirk. *Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable in Open Sources: Voices from the Open Source Revolution* - O'Reilly & Associates Inc., 1999
- MEIRELLES, Fernando S. – *Pesquisa Anual – Administração de Recursos de Informática – CIA – Centro de Informática Aplicada, São Paulo, 14ª edição, FGV/EAESP, março de 2003.*
- MICROSOFT. *Perspectiva da Microsoft sobre o Linux para tomadores de decisão*. Publicação institucional de agosto de 2002. Disponível em <<http://www.microsoft.com.br>>.
- MINGJI, Diana J. Wong; MILLETE, Wayne R. *Dealing with the dynamic duo of innovation and inertia: The theory of organization change*. in Organization Development Journal. Volume 20 nº 1 pág. 36, 2002.
- MOORE, Geoffrey A. *Dentro do furacão: estratégias de marketing para empresas de ponta*. São Paulo: Futura, 1996.

- O'REILLY, Tim. *Lessons from open-source software development*. Association for Computing Machinery. Volume 42 nº 4 pág. 32, 1999.
- OZSOMER, Aysegul; CAVUSGIL, Tamer. *The effects of technology standards on the structure of the global PC industry*. European Journal of Marketing. Volume 34 nº 9/10 pág. 1199, 2000.
- PERENS, Bruce. *The Open Source Definition in Open Sources: Voices from the Open Source Revolution* - O'Reilly & Associates Inc., 1999.
- RAYMOND, Eric S. *The Cathedral & the Bazaar - Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, 2001.
- RAYMOND, Eric S. The Magic Cauldron. Disponível em <<http://www.tuxedo.org/~esr>>. Acesso em novembro de 2003.
- REIS, Ana Helena Martins de Andrade. *Pesquisa qualitativa em marketing*. Uma visão crítica a respeito da utilização desta metodologia no Brasil, a partir do testemunho de clientes e usuários de São Paulo. Dissertação (mestrado) - Faculdade de Economia, Administração e Contabilidade. São Paulo: Universidade de São Paulo, 1994.
- REZENDE, Pedro Antonio Dourado de. *Informática, Panacéia e Arma*. Publicado na coluna Segurança, Bits & Cia do Jornal do Comercio em abril de 2002. Disponível em <<http://www.cic.unb.br/docentes/pedro/trabs/jcsbc4.htm>>. Acesso em dezembro de 2003.
- ROBBINS, Stephen. *Comportamento organizacional*. Rio de Janeiro: LTC. 1998.
- SALOMON, Décio Vieira. *Como fazer uma monografia*. Belo Horizonte: Interlivros, 1977.
- SCOTT, Brendan. *Why Free Software's Long Run TCO must be lower in* <http://members.optushome.com.au/brendanscott/papers/index.html>, julho de 2002.
- SEDDON, Peter B & GRAESER, Valerie & WILLCOCKS, Leslie P. *Measuring organizational is effectiveness: An overview and update of senior management perspectives*. Database for Advances in Information Systems. Volume 33 nº 2 pág. 11, 2002.
- SELLTIZ, Claire. *Métodos de pesquisa nas relações sociais*. São Paulo: EPU, 1960.
- SHAPIRO, Benson P. *Conquistando clientes*. São Paulo: Makron, 1995.

SHAPIRO, Carl & VARIAN, Hal R. *A Economia da Informação*. 4ª ed. Rio de Janeiro: Campus, 1999

STALLMAN, Richard M. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston: GNU Press, 2002.

STALLMAN, Richard M. *Porque o software deveria ser livre*. Disponível em: <<http://www.gnu.org/philosophy/shouldbefree.pt.html>>. Acesso em janeiro de 2001.

STALLMAN, Richard M. *The GNU Operating System and the Free Software Movement in Open Sources: Voices from the Open Source Revolution* - O'Reilly & Associates Inc., 1999

TAUDES, Alfred & FEURSTEIN, Markus & MILD, Andreas. *Options analysis of software platform decisions: a case study*. MIS Quarterly. Vol. 23 nº 2 pág. 227, 2000.

TAYLOR, Shirley & TODD, Peter. *Assessing IT usage: The role of prior experience*. MIS Quarterly. Volume 19 Issue 4 Pág. 561, 1995

TORVALDS, Linus. *Só por prazer: Linux : os bastidores da sua criação*. Rio de Janeiro: Campus, 2001.

WARD, John. *Strategic planning for information systems*. New York: John Wiley & Sons, 1994.

WEBSTER, Fredrick E. *Industrial marketing strategy*. New York: John Wiley & Sons, Inc, 1991.

WHEELER, David A. *Why Open Source Software / Free Software (OSS/FS) in* http://www.dwheeler.com/oss_fs_why.htm, agosto de 2002.

ZORZOLI, Pablo Luis. *Investigación sobre el Movimiento del Software Libre*. Disponível em <<http://www.megasitios.com/softwarelibre>>. Acesso em novembro de 2003.

GLOSSÁRIO E SIGLAS

ANUI – Associação Nacional dos Usuários de Internet

ASSESPRO - Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet. Sítio na Internet: www.assespro.org.br

Bibliotecas – Coleções de funções que servem para produzir outros programas. Elas são utilizadas para que não seja necessário reescrever código para executar tarefas comuns.

BSA – Business Software Alliance. Organização estadunidense que congrega produtores de software proprietário.

BSD – Berkley Software Distribution

Bug – Problema de codificação que tem por consequência o mau funcionamento do software. Utilizado sempre como sinônimo de erro de programação.

CC – Creative Commons

Código fonte – É um texto, numa sintaxe próxima à da linguagem humana, que contém as instruções sobre as tarefas que o programa deve executar. No software livre, o código fonte é livremente distribuído. No software proprietário, ele é tratado como segredo industrial.

CompTIA – Computing Technology Industry Association. Organização estadunidense que congrega produtores de software proprietário.

DARPA – Defense Advanced Research Projects Agency

DBA – Database Administrator – Profissional administrador de bancos de dados, responsável por sua manutenção e gerenciamento.

FSF – Free Software Foundation

GE – General Electric

GPL – General Public License

GNU – GNU is Not Unix

HTML – Hypertext Markup Language – Linguagem de marcação de hipertexto, utilizada para a criação de páginas para Internet.

Hurd – Núcleo original do sistema operacional GNU

Mainframes – Computadores de grande porte, responsáveis por centralizar todas as operações de negócio da empresa.

MAT – Modelo de Adoção de Tecnologia

MIT – Massachusetts Institute of Technology

OEM – Original Equipment Manufacturer

PHP – PHP Hypertext Preprocessor – é uma linguagem de script livre e de uso geral, muito utilizada e especialmente criada para o desenvolvimento de aplicações Web embutíveis dentro de código HTML.

SGI – Silicon Graphics Incompany

SI – Sistema Integrado – Sigla utilizada para designar o sistema de gestão em desenvolvimento pela empresa prestadora de serviços entrevistada, usuária de software livre.

Software de infra-estrutura – programas responsáveis pelo funcionamento dos computadores e por suas funções básicas, não diretamente relacionadas ao negócio da empresa. Esses programas podem ser utilizadas por organizações de qualquer ramo de atividade, sem qualquer tipo de personalização. Alguns exemplos: sistemas operacionais, sistemas de correio eletrônico, bancos de dados, anti-vírus, *firewalls*.

TI – Tecnologia da Informação

UML – Unified Modeling Language

USL – Unix System Laboratories

ANEXO A - SOFTWARES LIVRES DE ACORDO COM AS CATEGORIAS DA PESQUISA ANUAL DA FGV

1. Pacote integrado de escritório

- OpenOffice.org (www.openoffice.org)
- Koffice (www.koffice.org)
- Gnome Office (www.gnome.org/gnomeoffice)

2. Banco de dados (servidores e clientes)

- PostgreSQL (www.postgresql.org)
- MySQL (www.mysql.org)

3. Sistemas operacionais (servidores e clientes)

Núcleo Linux

Os sistemas operacionais abaixo utilizam o núcleo (*kernel*) Linux. Núcleo é a parte central do sistema operacional, responsável principalmente pela execução dos programas, escalonamento de tarefas e acesso aos recursos de hardware. Embora o núcleo seja uma parte importantíssima do sistema operacional, ele de forma alguma pode ser tomado pelo sistema completo. Um sistema operacional, é composto, além do núcleo, por uma infinidade de programas, utilizados para as mais diversas tarefas. A imprensa (mesmo especializada), e os profissionais da área de informática freqüentemente incorrem no erro de utilizar o nome Linux para referir-se a um sistema completo. O correto é chamá-los de sistemas operacionais de “núcleo Linux”. Uma referência também muito comum é a junção GNU/Linux, que denota a união entre o sistema operacional GNU, desenvolvido pela Free Software Foundation com o núcleo Linux.

- Debian (www.debian.org)
- Slackware (www.slackware.org)
- Red Hat (www.redhat.com)
- Conectiva (www.conectiva.com.br)

Núcleo BSD

- FreeBSD (www.freebsd.org)
- OpenBSD (www.openbsd.org)
- NetBSD (www.netbsd.org)

4. Linguagem de programação

Ambiente integrado de desenvolvimento

- Anjuta – C/C++ (www.anjuta.org)
- Qt Designer – C++ (www.trolltech.com)
- Kdevelop – C++ (www.kdevelop.org)

Linguagens de programação

- C, C++, Object Pascal, Basic, Cobol, entre outras

5. Editoração eletrônica, desenho e CAD

- Scribus – editoração de páginas (web2.altmuehlnet.de/fschmid)
- OpenOffice.org Draw – gráficos vetoriais (www.openoffice.org)
- Sodipodi – gráficos vetoriais (www.sodipodi.com)
- Gimp – gráficos bitmap (www.gimp.org)
- Blender – gráficos 3D (www.blender3d.org)
- Qcad – desenhos CAD (www.qcad.org)
- Dia – desenho de diagramas (www.lysator.liu.se/~alla/dia)

6. Correio Eletrônico

Servidores de transferência de mensagens

- Sendmail (www.sendmail.org)
- Exim (www.exim.org)
- Qmail (www.qmail.org)

Servidores de armazenamento e leitura de mensagens

- WU IMAP (www.washington.edu/imap)
- Courier IMAP (www.inter7.com/courierimap.html)

Webmail

- IMP/Horde (www.horde.org)
- OpenWebmail (openwebmail.org)

Cientes de correo

- Mozilla Mail (www.mozilla.org)
- Evolution (www.ximian.com)
- Kmail (www.kde.org)

7. Navegador de Internet

- Mozilla (www.mozilla.org)
- Konqueror (www.konqueror.org)

8. Groupware

- PHPGroupware (www.phpgroupware.org)
- PHPProjekt (www.phpprojekt.org)
- Tutos (www.tutos.org)

9. Antivírus

- Clam AV (clamav.elektrapro.com)

ANEXO B – LICENÇAS BSD

Abaixo está a licença original BSD, que continha uma cláusula (número 3) dizendo que todo o material que fizesse referência ao software deveria conter uma aviso dando créditos à Universidade de Berkley. O problema é que conforme outras pessoas e instituições foram adicionando funcionalidades, começaram a requerer também que seus nomes fossem listados. O aumento do número de linhas desse aviso gerou vários problemas, a ponto da FSF não considerar a licença BSD compatível com a GPL. A versão reproduzida abaixo já contém uma modificação, da forma a eximir o licenciado de obedecer a cláusula 3.

The 4.4BSD Copyright

All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by The Regents of the University of California.

Copyright 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Institute of Electrical and Electronics Engineers and the American National Standards Committee X3, on Information Processing Systems have given us permission to reprint portions of their documentation.

In the following statement, the phrase ``this text" refers to portions of the system documentation.

Portions of this text are reprinted and reproduced in electronic form in the second BSD Networking Software Release, from IEEE Std 1003.1-1988, IEEE Standard Portable Operating System Interface for Computer Environments (POSIX), copyright C 1988 by the Institute of Electrical and Electronics Engineers, Inc. In the event of any discrepancy between these versions and the original IEEE Standard, the original IEEE Standard is the referee document.

In the following statement, the phrase "This material" refers to portions of the system documentation.

This material is reproduced with permission from American National Standards Committee X3, on Information Processing Systems. Computer and Business Equipment Manufacturers Association (CBEMA), 311 First St., NW, Suite 500, Washington, DC 20001-2178. The developmental work of Programming Language C was completed by the X3J11 Technical Committee.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Regents of the University of California.

NOTE: The copyright of UC Berkeley's Berkeley Software Distribution ("BSD") source has been updated. The copyright addendum may be found at <ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change> and is included below.

July 22, 1999

To All Licensees, Distributors of Any Version of BSD:

As you know, certain of the Berkeley Software Distribution ("BSD") source code files require that further distributions of products containing all or portions of the software, acknowledge within their advertising materials that such products contain software developed by UC Berkeley and its contributors.

Specifically, the provision reads:

"3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the University of California, Berkeley and its contributors."

Effective immediately, licensees and distributors are no longer required to include the acknowledgement within advertising materials. Accordingly, the foregoing paragraph of those BSD Unix files containing it is hereby deleted in its entirety.

William Hoskins

Director, Office of Technology Licensing

University of California, Berkeley

Abaixo está a licença de uma das variantes do BSD original, o FreeBSD. Sua licença é mais simples, e já não inclui a cláusula que obriga a inserção de avisos nos materiais derivados.

The FreeBSD Copyright

Copyright 1994-2004 The FreeBSD Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE FREEBSD PROJECT ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

ANEXO C – GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or

with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on

consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

ANEXO D – LICENÇA PÚBLICA GERAL

LPG–PC - LICENÇA PÚBLICA GERAL - PARA PROGRAMAS DE COMPUTADOR

Versão 2.1.0

12 de outubro de 2003

Adaptação para o Brasil da:

G.N.U. - GENERAL PUBLIC LICENSE

Version 2, June 1991

Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Para ter acesso à versão mais atualizada desta LPG–PC acesse:

Licença: <http://www.lpg.adv.br/licencas/lpgpc.rtf>

Assinatura digital: <http://www.lpg.adv.br/licencas/lpgpc.rtf.sig>

Advogado responsável: Pablo de Camargo Cerdeira

E-mail: pcc@adv.oabsp.org.br

Chave pública do advogado responsável pela licença:

<http://www.lpg.adv.br/licencas/pcc.asc>

Utilize esta chave para a validação da LPG–PC em caso de dúvidas sobre sua integridade.

NOTA PRELIMINAR

Esta LPG–PC (Licença Pública Geral para Programas de Computador) – é organizada por Pablo de Camargo Cerdeira, advogado, OAB/SP 207.570, tomando-se por base a GPL - General Public Licence - da Free Software Foundation, publicada no site <http://www.gnu.org>. Entretanto, em razão das diferenças existentes entre o ordenamento jurídico estadunidense, base da GPL, e o brasileiro, algumas alterações precisaram ser feitas. Objetivamos, contudo, apenas garantir que a GNU/GPL possa ser aplicada no Brasil sem qualquer conflito com nossa legislação.

Para a consecução desta LPG–PC foi de fundamental importância a colaboração dos membros de diversas listas de discussões, dentre elas a lista do sítio deste licença, <http://www.lpg.adv.br>. Para se inscrever basta enviar um e-mail para lista@lpg.adv.br com o assunto SUBSCRIBE. A inscrição também pode ser feita no próprio site.

INTRODUÇÃO

EXPLICAÇÕES BÁSICAS SOBRE OS TERMOS DESTA LICENÇA

NOTA: Esta Introdução não integra o termo de licenciamento da LPG-PC. Seu caráter é meramente ilustrativo e visa explicar de forma resumida e simplificada os termos jurídicos da LICENÇA.

Esta licença:

1. garante que qualquer interessado possa usar, publicar, distribuir, reproduzir ou alterar o programa aqui licenciado;
2. obriga que o programa seja distribuído acompanhado de seu código-fonte;
3. obriga que todos esses direitos sejam transferidos gratuitamente;
4. aceita que possa haver cobrança pelos serviços prestados, como, por exemplo, a gravação de um CD, o serviço de entrega, o serviço de fazer alterações no programa, etc;
5. aceita que se o programa for licenciado envolvendo pessoas de outros países ele será licenciado de acordo com a GNU/GPL;
6. declara que o programa não tem qualquer garantia por não ser comercializado. Estas existem apenas sobre os serviços cobrados, como os que foram listados no parágrafo 4 deste resumo;
7. não-permite que o programa ou trechos dele sejam distribuídos em sistemas apenas compilados e/ou mediante pagamento (sistemas proprietários);

LPG-PC

LICENÇA PÚBLICA GERAL

PARA PROGRAMAS DE COMPUTADOR

Versão 2.1.0

12 de outubro de 2003

I. PARTES:

São as partes deste contrato de licença:

I.1 O(s) titular(es) dos direitos de autor do Programa de Computador especificado abaixo – doravante denominado(s) apenas LICENCIADOR –, e

I.2 Toda pessoa, física ou jurídica – doravante denominado apenas LICENCIADO – que USAR, PUBLICAR, DISTRIBUIR, REPRODUZIR ou ALTERAR O PROGRAMA especificado abaixo.

As partes têm entre si justo e acordado firmar o presente CONTRATO DE LICENCIAMENTO GRATUITO DE USO, PUBLICAÇÃO, DISTRIBUIÇÃO, REPRODUÇÃO E ALTERAÇÃO DO PROGRAMA DE COMPUTADOR, denominado apenas PROGRAMA.

II. TERMOS UTILIZADOS NESTA LICENÇA

II.1 O termo PROGRAMA deste licenciamento deve, necessariamente, englobar suas duas formas distintas, conforme previsto no artigo 1º da Lei 9609/1998:

A. Compilada: termo que representa toda distribuição preparada para ser entendida pelas máquinas a qual se destina a execução do PROGRAMA;

B. Código-fonte: termo que representa a forma adequada para se fazer alterações no PROGRAMA. O código-fonte distribuído deve incluir todos os códigos-fonte de todos os módulos contidos e necessários ao funcionamento normal do PROGRAMA, mais as rotinas utilizadas para controlar a compilação e a instalação do mesmo. Não é necessário incluir no código-fonte nada que já seja normalmente distribuído, não importa se na forma de código-fonte ou compilada, como os componentes do sistema operacional em que o PROGRAMA será executado.

II.1.1 Como exceção, o termo PROGRAMA pode, eventualmente, abarcar apenas a sua forma código-fonte quando esta for a forma normal de sua distribuição e utilização. Esta exceção se aplica a programas elaborados em linguagens destinadas à compilação em tempo de execução, interpretada ou qualquer outra tecnologia similar; a título de exemplo podemos citar linguagens como Perl, PHP, VBScript, e outras semelhantes. Nestes casos não se aplicam as obrigações desta LICENÇA relativas à distribuição da forma compilada do PROGRAMA.

II.2 A licença General Public Licence, da Free Software Foundation, será tratada apenas por GNU/GPL, e está publicada no endereço <http://www.gnu.org/licenses/gpl.txt>

III.2 Doravante este termo de licenciamento será tratado apenas por LICENÇA ou LPG-PC.

1. OBJETO: LICENCIAMENTO PARA USO, PUBLICAÇÃO, DISTRIBUIÇÃO, REPRODUÇÃO E ALTERAÇÃO DE PROGRAMA DE COMPUTADOR.

Constitui objeto deste contrato o licenciamento gratuito, por parte do LICENCIADOR ao LICENCIADO, dos direitos de USO, PUBLICAÇÃO, DISTRIBUIÇÃO, REPRODUÇÃO e ALTERAÇÃO, entendidos conforme o artigo 5º da Lei 9610/1998, do PROGRAMA pertencente ao LICENCIADOR. O PROGRAMA necessariamente acompanha sua forma compilada e sua forma código-fonte, respeitada a exceção prevista no item II.1.1.

A gratuidade do licenciamento ora apresentado não obsta a cobrança por outros serviços ou custos, como o de desenvolvimento ou adequação do PROGRAMA para determinados fins, ou ainda os custos de sua DISTRIBUIÇÃO. Tais serviços suplementares não se sujeitam a este termo de licenciamento, salvo o de DISTRIBUIÇÃO.

O LICENCIADOR não poderá, após a PUBLICAÇÃO ou a DISTRIBUIÇÃO do PROGRAMA, se opor a terceiros com relação a qualquer dos direitos objetos deste termo, desde que exercidos em acordo com o ora acertado.

Fica desde já claro que o PROGRAMA não foi colocado em domínio público e os direitos de autor continuam pertencendo ao LICENCIADOR, independentemente de registro, de acordo com o art. 2º, § 3º da Lei 9609/1998.

2. APLICAÇÕES DESTA LICENÇA E DE SUA RELAÇÃO COM A GNU/GPL.

O PROGRAMA regula-se conforme aqui especificado quando:

A. o PROGRAMA estiver situado no Brasil, conforme o artigo 8º da Lei de Introdução ao Código Civil. Por tal entende-se as situações em que o PROGRAMA é usado, publicado, distribuído, reproduzido ou alterado em qualquer meio ou suporte que se possa fisicamente localizar em território brasileiro; ou

B. quando as obrigações concernentes ao PROGRAMA forem constituídas no Brasil, conforme o artigo 9º da Lei de Introdução ao Código Civil; ou ainda

C. nas hipóteses em que o LICENCIADOR optar por utilizá-la. Tal vontade se expressa com a simples utilização desta LICENÇA para o licenciamento do PROGRAMA.

Nos casos em que o PROGRAMA for licenciado com a presença de elemento estrangeiro serão utilizados os termos da GNU/GPL. Nos licenciamentos subseqüentes dos programas derivados do PROGRAMA, se o

elemento estrangeiro deixar de se fazer presente, estas derivações serão licenciadas de acordo com o conteúdo desta LICENÇA.

Quando houver dúvidas com relação aos direitos e deveres estabelecidos por esta LICENÇA, estas deverão ser sanadas primeiramente se recorrendo aos termos da GNU/GPL, para então se recorrer às demais regras tradicionais do direito para interpretação dos contratos.

3. GARANTIAS

O PROGRAMA é distribuído gratuitamente, como se um presente à comunidade fosse, não sendo ele nunca objeto de comercialização. Assim, não se aplica a ele a GARANTIA prevista pelo artigo 8º da Lei 9609/1998 e nem QUALQUER DAS GARANTIAS previstas pela Lei 8078/1990. O LICENCIADOR não terá obrigação de prestar suporte, assistência ou esclarecimentos ao LICENCIADO, exceto nos casos de ordem judicial.

Como exceção, os titulares dos direitos de autor dos programas de computador derivados poderão garantir apenas a parte que lhes compete, sem qualquer comprometimento dos autores das versões originárias.

Salvo nas hipóteses de má-fé do LICENCIADOR, todos os prejuízos decorrentes do USO, PUBLICAÇÃO, DISTRIBUIÇÃO, REPRODUÇÃO ou ALTERAÇÃO do PROGRAMA são de inteira responsabilidade do LICENCIADO.

Esta ausência de GARANTIA diz respeito única e exclusivamente ao PROGRAMA e a seu funcionamento. Qualquer outro serviço prestado não englobado por este termo, como suporte técnico ou assistência, por exemplo, estarão sujeitos a termo próprio e à legislação que rege tais relações.

4. CONDIÇÕES GERAIS DE LICENCIAMENTO

Fica vedado o USO, PUBLICAÇÃO, DISTRIBUIÇÃO, REPRODUÇÃO ou ALTERAÇÃO do PROGRAMA em desacordo com as condições estabelecidas nesta LICENÇA.

A utilização desta LICENÇA implica no reconhecimento da chave pública disponível em <http://www.lpg.adv.br/licencas/pcc.asc>, emitida e certificada pela ICP-OAB (Infra-estrutura de Chave Pública da Ordem dos Advogados do Brasil) como sendo válida para garantir a autoria e autenticidade do documento, conforme a M.P. 2200-02/2001, em seu artigo 10, § 2º.

5. DIREITOS GERAIS DO LICENCIADO

O LICENCIADO tem o direito de USAR, PUBLICAR, DISTRIBUIR, REPRODUZIR e ALTERAR o PROGRAMA, seja a forma compilada seja a forma código-fonte, conforme disposto nesta licença.

6. OBRIGAÇÕES GERAIS DO LICENCIADO

O LICENCIADO obriga-se a aceitar este termo. Caso o LICENCIADO não concorde integralmente com o aqui exposto a ele não será permitido USAR, PUBLICAR, DISTRIBUIR, REPRODUZIR ou ALTERAR o PROGRAMA.

O LICENCIADO não poderá jamais transformar o PROGRAMA ou parte dele, ou qualquer programa de computador derivado do PROGRAMA ou de parte dele, em:

A. Programa comercial, licenciado mediante paga;

B. Programa proprietário, distribuído sem o respectivo código-fonte.

O LICENCIADO não poderá retirar os créditos do LICENCIADOR pela parte do PROGRAMA que a este compete. Em conformidade com o item 7 abaixo, poderá ocorrer tal retirada apenas quando solicitada pelo LICENCIADOR.

É também obrigação do LICENCIADO não registrar, ou utilizar proteção equivalente, o PROGRAMA ou qualquer aspecto deste, com o fim de restringir a plena transferência de todos os direitos objetos desta LICENÇA para terceiros. Assim, a título de exemplo, o LICENCIADO fica impedido de buscar patentes para o PROGRAMA ou de registrar o nome ou qualquer outro sinal distintivo deste como marca registrada.

7. DIREITOS GERAIS DO LICENCIADOR

O LICENCIADOR mantém para si o direito de reivindicação da paternidade do PROGRAMA, conforme previsto na Lei 9609/1998, em seu artigo 2º, § 1º.

É também direito do LICENCIADOR ter todos os sinais convencionais indicativos de sua autoria, por ele ou a seu pedido apostos, mantidos tal como originalmente os colocou. Esse direito se estende inclusive aos programas de computador derivados.

Entretanto, o LICENCIADOR poderá, a qualquer tempo, solicitar a remoção de todos os sinais convencionais indicativos de sua autoria, por ele ou a seu pedido apostos, de programa de computador derivado do PROGRAMA quando julgar que as alterações realizadas naquele possam prejudicar sua honra ou sua reputação. Tal requerimento deverá ser realizado por escrito ou por meio digital; neste último caso deverá ser utilizado algum meio de comprovação da autoria e integridade do documento.

8. OBRIGAÇÕES GERAIS DO LICENCIADOR

O LICENCIADOR deverá sempre DISTRIBUIR o PROGRAMA em duas formas, quais sejam, compilada e código-fonte. À esta obrigação aplica-se a exceção do item II.1.1, qual seja, é possível a distribuição apenas da forma código-fonte quando esta for a forma normal de distribuição e utilização do PROGRAMA.

O LICENCIADOR não poderá jamais revogar qualquer direito aqui cedido; também não poderá alterar os direitos de qualquer versão anterior já licenciada do PROGRAMA.

É obrigação do LICENCIADOR, para que o PROGRAMA possa ser distribuído de acordo com esta LICENÇA, desistir de seu direito de opor-se a alterações no PROGRAMA. Poderá, caso não concorde com alterações realizadas em programas de computador derivados do PROGRAMA, solicitar a supressão de seus sinais distintivos daquele derivado, em conformidade com o item 7 acima.

9. USO DO PROGRAMA

Por USO entende-se, com relação à forma compilada ou interpretada do PROGRAMA, o seu uso normal, conforme descrito na documentação técnica do mesmo.

O USO também pode ser relativo à forma código-fonte do PROGRAMA, destinando-se esta a tornar pública a estrutura da forma compilada do mesmo, bem como permitir o desenvolvimento de novos programas de computador derivados ou que utilizem trechos do PROGRAMA.

Não existe qualquer impedimento para o USO do PROGRAMA com finalidade comercial, desde que este não seja o objeto do negócio.

Nas situações em que trechos do PROGRAMA forem utilizados em outros programas de computador, estes deverão ser licenciados obrigatoriamente por esta LICENÇA. Como exceção à esta regra, o programa de computador que utilizar trechos do PROGRAMA poderá ser licenciado de outra forma desde que seu funcionamento normal independa da parte copiada e que esta seja acompanhada da sua forma código-fonte.

10. PUBLICAÇÃO DO PROGRAMA

A PUBLICAÇÃO do PROGRAMA engloba apenas a sua forma código-fonte. O LICENCIADO poderá publicar o código-fonte, em qualquer meio e para qualquer finalidade. O objetivo de tal liberdade de PUBLICAÇÃO é fazer com que a forma código-fonte do PROGRAMA torne-se tão difundida quanto possível.

Toda PUBLICAÇÃO do PROGRAMA deverá ser acompanhada de cópia integral desta LICENÇA ou, pelo menos, das referências indicadas abaixo. A forma código-fonte do PROGRAMA deverá conter, em todos os arquivos de texto componentes do mesmo, no início, referência a esta LICENÇA, que deverá ser feita da seguinte forma:

MARCADOR DE INÍCIO DE COMENTÁRIO

Este programa é licenciado de acordo com a LPG-PC (Licença Pública Geral para Programas de Computador), versão 2.1.0 ou qualquer versão posterior.

A LPG-PC deve acompanhar todas PUBLICAÇÕES, DISTRIBUIÇÕES e REPRODUÇÕES deste Programa.

Caso uma cópia da LPG-PC não esteja disponível junto com este Programa, você pode contatar o LICENCIADOR ou então pegar uma cópia em:

Licença: <http://www.lpg.adv.br/licencas/lpgpc.rtf>

Para poder USAR, PUBLICAR, DISTRIBUIR, REPRODUZIR ou ALTERAR este Programa é preciso estar de acordo com os termos da LPG-PC

MARCADOR DE FINAL DE COMENTÁRIO

Caso o PROGRAMA seja obra coletiva, conforme especificado no item 13, não se aplicará à distribuição o artigo 88 da Lei 9610/1998.

11. DISTRIBUIÇÃO DO PROGRAMA

Os direitos de DISTRIBUIÇÃO do PROGRAMA aplicam-se às formas compiladas e código-fonte. Apenas poderá o LICENCIADO distribuir o programa concomitantemente em suas duas formas já citadas, sendo vedada a DISTRIBUIÇÃO de apenas o código-fonte ou apenas a forma compilada, salvo quando o PROGRAMA for desenvolvido de acordo com o item II.1.1.

Toda DISTRIBUIÇÃO do PROGRAMA deverá ser acompanhada de cópia desta LICENÇA.

O arquivo da LICENÇA que acompanha o PROGRAMA deverá ter seu nome mantido como LPGPC.rtf ou poderá ter seu nome alterado para LICENÇA.RTF. Admite-se também a utilização de outras extensões e formatos para a LICENÇA, desde que sejam formatos abertos e não proprietários, como o TXT (texto puro) ou o HTML (HyperText Markup Language). Entretanto, recomenda-se que o programa sempre seja acompanhado da versão no formato RTF, pois é a única assinada digitalmente e capaz de garantir a completa integridade do texto.

Não existe qualquer óbice à DISTRIBUIÇÃO com fins comerciais, desde que fique claro que não haverá cobrança pelo PROGRAMA, mas sim pelos meios e serviços de distribuição.

12. REPRODUÇÃO DO PROGRAMA

Não há qualquer restrição ao número de cópias do PROGRAMA que o LICENCIADO poderá fazer. Sua REPRODUÇÃO é livre em qualquer meio, mas deverá respeitar as restrições supras, dos itens 10 e 11, de acordo com a destinação de cada REPRODUÇÃO.

Toda REPRODUÇÃO do PROGRAMA deverá ser acompanhada de cópia desta LICENÇA, respeitando o exposto no item anterior.

13. ALTERAÇÃO DO PROGRAMA

É permitido ao LICENCIADO fazer alterações no PROGRAMA. Toda ALTERAÇÃO, entretanto, deverá ser identificada e comentada no código-fonte para fins de identificação da autoria.

Não é permitido fazer alterações nos créditos e marcas distintivas apostas pelo LICENCIADOR e por qualquer autor de versões originárias, salvo quando permitido pelo LICENCIADOR.

Os programas de computador produtos da alteração do PROGRAMA também se sujeitam a esta LICENÇA, seja novo programa ou apenas nova versão do PROGRAMA.

Os autores do PROGRAMA poderão utilizar outras formas de licenciamento para a sua parte de contribuição, se esta puder ser utilizada separadamente, conforme o artigo 15º parágrafo 2º da Lei 9610/1998.

O PROGRAMA sujeito à LPG-PC sempre conterà uma referência à esta LICENÇA em local de fácil acesso aos usuários do PROGRAMA. Caso o PROGRAMA seja executado em terminal de comando, sem interface gráfica, no início de sua execução normal deverá constar o seguinte aviso:

Este Programa é licenciado de acordo com a LPG-PC (Licença Pública Geral para Programas de Computador).

Para poder USAR, PUBLICAR, DISTRIBUIR, REPRODUZIR ou ALTERAR este Programa é preciso estar de acordo com os termos da LPG-PC

Maiores informações podem ser obtidas em: <http://www.lpg.adv.br>

Se o PROGRAMA tiver interface gráfica, deverá ser aposta em local apropriado – normalmente junto aos itens “Ajuda” ou “Sobre o Programa” – cópia da LICENÇA, e referência à localização da versão mais atual da mesma, qual seja, <http://www.lpg.adv.br/licencas/lpgpc.rtf>.

Nas situações em que o PROGRAMA contar com participações diversas na sua elaboração ele será considerado obra coletiva, conforme o artigo 5º, VIII, h da Lei 9610/1998, ficando os direitos patrimoniais sob responsabilidade do organizador, que aceita integralmente os termos desta LICENÇA ao utilizá-la para licenciamento do PROGRAMA.

A participação na obra coletiva aqui licenciada presume-se não-onerosa. Os autores outros que não o organizador da obra coletiva abdicam de seus direitos patrimoniais sobre ela, bem como do direito previsto no artigo 88, II da Lei 9610/1998.

14. ANEXOS À ESTA LICENÇA

Esta LICENÇA não admite qualquer alteração em seu conteúdo, inclusive para validação de sua integridade. Entretanto, como alternativa para pequenas adaptações ou complementações ela aceita anexos em arquivos separados.

Tais anexos não poderão, em hipótese alguma, conflitar com o conteúdo desta licença, sendo consideradas nulas as disposições que estiverem nesta situação. Também não terão validade nos casos em que o PROGRAMA estiver sob licença da GNU/GPL, conforme o item 2 desta LICENÇA, salvo anexos que impõem restrições territoriais para o licenciamento, aceitos pela GNU/GPL.

Quando a LICENÇA estiver acompanhada de anexos, estes serão considerados como integrantes da LICENÇA e deverão acompanhar todos os licenciamentos subsequentes.

Os nomes dos arquivos anexos deverão respeitar a seguinte regra: LPG-PC-ANEXO_<número do anexo com dois dígitos>-<título do anexo>.RTF ou LICENÇA-ANEXO_<número do anexo>-<título do anexo>.RTF. Outras extensões são admitidas em conformidade com o item 11 deste termo. Por exemplo, LICENÇA-ANEXO_01-GNUGPL_INGLÊS.TXT.

15. PRAZO

O presente instrumento vigorará por prazo indeterminado.

16. CESSAÇÃO DO LICENCIAMENTO

O não-cumprimento de qualquer das cláusulas desta licença ensejará a cessação instantânea de todos os direitos do LICENCIADO sobre o PROGRAMA, sem a necessidade de denúncia deste texto.

17. NOVAS VERSÕES DESTA LICENÇA

Esta LICENÇA será automaticamente substituída pela versão mais atual disponível em <http://www.lpg.adv.br/licencas/lpgpc.rtf>. As partes poderão optar livremente por utilizar a versão desta LICENÇA que acompanha o PROGRAMA ou a última versão disponível no endereço retro citado. As novas versões da LICENÇA não poderão, entretanto, em hipótese alguma, retirar direitos já garantidos para qualquer das partes na versão em vigor quando da sua aceitação.

18. FORO

As partes elegem o foro da comarca de residência ou sede do detentor dos direitos de autor do PROGRAMA, com renúncia a qualquer outro, por mais privilegiado que seja, para dirimir as dúvidas e questões oriundas do presente instrumento que não possam ser resolvidas administrativamente.

ANEXO F – LICENÇA CC GPL

Licença Pública Geral do GNU (GPL) [General Public License]

Versão 2, Junho de 1991 Direitos Autorais Reservados © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite [conjunto] 330, Boston, MA [Massachusetts] 02111-1307 USA [Estados Unidos da América]

É permitido a qualquer pessoa copiar e distribuir cópias sem alterações deste documento de licença, sendo vedada, entretanto, qualquer modificação.

Introdução

As licenças da maioria dos software são elaboradas para suprimir sua liberdade de compartilhá-los e modificá-los. A Licença Pública Geral do GNU, ao contrário, visa garantir sua liberdade de compartilhar e modificar softwares livres para assegurar que o software seja livre para todos os seus usuários. Esta Licença Pública Geral é aplicável à maioria dos softwares da Free Software Foundation [Fundação do Software Livre] e a qualquer outro programa cujos autores se comprometerem a usá-la. (Em vez dela, alguns outros softwares da Free Software Foundation são cobertos pela Licença Pública Geral de Biblioteca do GNU). Você também poderá aplicá-la aos seus programas.

Quando falamos de software livre, estamos nos referindo à liberdade, não ao preço. Nossas Licenças Públicas Gerais visam garantir que você tenha a liberdade de distribuir cópias de software livre (e cobrar por isso se desejar), que receba código-fonte ou possa obtê-lo se desejar, que possa modificá-lo ou usar partes dele em novos programas livres; finalmente, que você tenha ciência de que pode fazer tudo isso.

Para proteger seus direitos, necessitamos fazer restrições que proíbem que alguém negue esses direitos a você ou que solicite que você renuncie a eles. Essas restrições se traduzem em determinadas responsabilidades que você deverá assumir, se for distribuir cópias do software ou modificá-lo.

Por exemplo, se você distribuir cópias de algum desses programas, tanto gratuitamente como mediante uma taxa, você terá de conceder aos receptores todos os direitos que você possui. Você terá de garantir que, também eles, recebam ou possam obter o código-fonte. E você terá a obrigação de exibir a eles esses termos, para que eles conheçam seus direitos.

Protegemos seus direitos através de dois passos: (1) estabelecendo direitos autorais sobre o software e (2) concedendo a você esta licença, que dá permissão legal para copiar, distribuir e/ou modificar o software.

Além disso, para a proteção de cada autor e a nossa, queremos ter certeza de que todos entendam que não há nenhuma garantia para este software livre. Se o software for modificado por alguém e passado adiante, queremos que seus receptores saibam que o que receberam não é o original, de forma que quaisquer problemas introduzidos por terceiros não afetem as reputações dos autores originais.

Finalmente, qualquer programa livre é constantemente ameaçado por patentes de software. Queremos evitar o risco de que redistribuidores de um programa livre obtenham individualmente licenças sob uma patente, tornando o programa, com efeito, protegido por direitos de propriedade intelectual. Para impedir isso, deixamos claro que qualquer patente deve ser licenciada para o uso livre por parte de qualquer pessoa ou, então, simplesmente não deve ser licenciada.

Os exatos termos e condições para cópia, distribuição e modificação seguem abaixo.

TERMOS E CONDIÇÕES PARA CÓPIA, DISTRIBUIÇÃO E MODIFICAÇÃO

0. Esta Licença se aplica a qualquer programa ou outra obra que contenha um aviso inserido pelo respectivo titular dos direitos autorais, informando que a referida obra pode ser distribuída em conformidade com os termos desta Licença Pública Geral. O termo "Programa", utilizado abaixo, refere-se a qualquer programa ou obra, e o termo "obras baseadas no Programa" significa tanto o Programa, como qualquer obra derivada nos termos da legislação de direitos autorais: isto é, uma obra contendo o Programa ou uma parte dele, tanto de forma idêntica como com modificações, e/ou traduzida para outra linguagem. (Doravante, o termo "modificação" inclui também, sem reservas, a tradução). Cada licenciado, doravante, será denominado "você".

Outras atividades que não a cópia, distribuição e modificação, não são cobertas por esta Licença; elas estão fora de seu escopo. O ato de rodar o Programa não tem restrições e o resultado gerado a partir do Programa encontra-se coberto somente se seu conteúdo constituir uma obra baseada no Programa (independente de ter sido produzida pela execução do Programa). Na verdade, isto dependerá daquilo que o Programa faz.

1. Você poderá fazer cópias idênticas do código-fonte do Programa ao recebê-lo e distribuí-las, em qualquer mídia ou meio, desde que publique, de forma ostensiva e adequada, em cada cópia, um aviso de direitos autorais (ou copyright) apropriado e uma notificação sobre a exoneração de garantia; mantenha intactas as informações, avisos ou notificações referentes a esta Licença e à ausência de qualquer garantia; e forneça a quaisquer outros receptores do Programa uma cópia desta Licença junto com o Programa.

Você poderá cobrar um valor pelo ato físico de transferir uma cópia, e você pode oferecer, se quiser, a proteção por uma garantia em troca de um valor.

2. Você poderá modificar sua cópia ou cópias do Programa ou qualquer parte dele, formando, dessa forma, uma obra baseada no Programa, bem como copiar e distribuir essas modificações ou obra, de acordo com os termos da Cláusula 1 acima, desde que você também atenda a todas as seguintes condições:

a. Você deve fazer com que os arquivos modificados contenham avisos, em destaque, informando que você modificou os arquivos, bem como a data de qualquer modificação.

b. Você deve fazer com que qualquer obra que você distribuir ou publicar, que no todo ou em parte contenha o Programa ou seja dele derivada, ou derivada de qualquer parte dele, seja licenciada como um todo sem qualquer custo para todos terceiros nos termos desta licença.

c. Se o programa modificado normalmente lê comandos interativamente quando rodado, você deverá fazer com que ele, ao começar a ser rodado para esse uso interativo em sua forma mais simples, imprima ou exiba um aviso incluindo o aviso de direitos autorais (ou copyright) apropriado, além de uma notificação de que não há garantia (ou, então, informando que você oferece uma garantia) e informando que os usuários poderão redistribuir o programa de acordo com essas condições, esclarecendo ao usuário como visualizar uma cópia desta Licença. (Exceção: se o Programa em si for interativo mas não imprimir normalmente avisos como esses, não é obrigatório que a sua obra baseada no Programa imprima um aviso).

Essas exigências se aplicam à obra modificada como um todo. Se partes identificáveis dessa obra não forem derivadas do Programa e puderem ser consideradas razoavelmente como obras independentes e separadas por si próprias, nesse caso, esta Licença e seus termos não se aplicarão a essas partes quando você distribuí-las como obras separadas. Todavia, quando você distribuí-las como parte de um todo que constitui uma obra baseada no Programa, a distribuição deste todo terá de ser realizada em conformidade com esta Licença, cujas permissões para outros licenciados se estenderão à obra por completo e, conseqüentemente, a toda e qualquer parte, independentemente de quem a escreveu.

Portanto, esta cláusula não tem a intenção de reivindicar direitos ou contestar seus direitos sobre uma obra escrita inteiramente por você; a intenção é, antes, de exercer o direito de controlar a distribuição de obras derivadas ou obras coletivas baseadas no Programa.

Além do mais, a simples agregação de outra obra que não seja baseada no Programa a ele (ou a uma obra baseada no Programa) em um volume de mídia ou meio de armazenamento ou distribuição, não inclui esta outra obra no âmbito desta Licença.

3. Você poderá copiar e distribuir o Programa (ou uma obra baseada nele, de acordo com a Cláusula 2) em código-objeto ou formato executável de acordo com os termos das Cláusulas 1 e 2 acima, desde que você também tome uma das providências seguintes:

a. Incluir o código-fonte correspondente completo, passível de leitura pela máquina, o qual terá de ser distribuído de acordo com as Cláusulas 1 e 2 acima, em um meio ou mídia habitualmente usado para intercâmbio de software; ou,

b. Incluir uma oferta por escrito, válida por pelo menos três anos, para fornecer a qualquer terceiro, por um custo que não seja superior ao seu custo de fisicamente realizar a distribuição da fonte, uma cópia completa passível de leitura pela máquina, do código-fonte correspondente, a ser distribuído de acordo com as Cláusulas 1 e 2 acima, em um meio ou mídia habitualmente usado para intercâmbio de software; ou,

c. Incluir as informações recebidas por você, quanto à oferta para distribuir o código-fonte correspondente. (Esta alternativa é permitida somente para distribuição não-comercial e apenas se você tiver recebido o programa em código-objeto ou formato executável com essa oferta, de acordo com a letra b, acima).

O código-fonte de uma obra significa o formato preferencial da obra para que sejam feitas modificações na mesma. Para uma obra executável, o código-fonte completo significa o código-fonte inteiro de todos os módulos que ela contiver, mais quaisquer arquivos de definição de interface associados, além dos scripts usados para controlar a compilação e instalação do executável. Entretanto, como uma exceção especial, o código-fonte distribuído não precisa incluir nada que não seja normalmente distribuído (tanto no formato fonte como no binário) com os componentes principais (compilador, kernel e assim por diante) do sistema operacional no qual o executável é rodado, a menos que este componente em si acompanhe o executável.

Se a distribuição do executável ou código-objeto for feita mediante a permissão de acesso para copiar, a partir de um local designado, então, a permissão de acesso equivalente para copiar o código-fonte a partir do mesmo local será considerada como distribuição do código-fonte, mesmo que os terceiros não sejam levados a copiar a fonte junto com o código-objeto.

4. Você não poderá copiar, modificar, sublicenciar ou distribuir o Programa, exceto conforme expressamente estabelecido nesta Licença. Qualquer tentativa de, de outro modo, copiar, modificar, sublicenciar ou distribuir o Programa será inválida, e automaticamente rescindirá seus direitos sob esta Licença. Entretanto, terceiros que tiverem recebido cópias ou direitos de você de acordo esta Licença não terão suas licenças rescindidas, enquanto estes terceiros mantiverem o seu pleno cumprimento.

5. Você não é obrigado a aceitar esta Licença, uma vez que você não a assinou. Porém, nada mais concede a você permissão para modificar ou distribuir o Programa ou respectivas obras derivativas. Tais atos são proibidos por lei se você não aceitar esta Licença. Conseqüentemente, ao modificar ou distribuir o Programa (ou qualquer obra baseada no Programa), você estará manifestando sua aceitação desta Licença para fazê-lo, bem como de todos os seus termos e condições para copiar, distribuir ou modificar o Programa ou obras nele baseadas.

6. Cada vez que você redistribuir o Programa (ou obra baseada no Programa), o receptor receberá, automaticamente, uma licença do licenciante original, para copiar, distribuir ou modificar o Programa, sujeito a estes termos e condições. Você não poderá impor quaisquer restrições adicionais ao exercício, pelos receptores, dos direitos concedidos por este instrumento. Você não tem responsabilidade de promover o cumprimento por parte de terceiros desta licença.

7. Se, como resultado de uma sentença judicial ou alegação de violação de patente, ou por qualquer outro motivo (não restrito às questões de patentes), forem impostas a você condições (tanto através de mandado judicial, contrato ou qualquer outra forma) que contradigam as condições desta Licença, você não estará desobrigado quanto às condições desta Licença. Se você não puder atuar como distribuidor de modo a satisfazer simultaneamente suas obrigações sob esta licença e quaisquer outras obrigações pertinentes, então, como consequência, você não poderá distribuir o Programa de nenhuma forma. Por exemplo, se uma licença sob uma patente não permite a redistribuição por parte de todos aqueles que tiverem recebido cópias, direta ou indiretamente de você, sem o pagamento de royalties, então, a única forma de cumprir tanto com esta exigência quanto com esta licença será deixar de distribuir, por completo, o Programa.

Se qualquer parte desta Cláusula for considerada inválida ou não executável, sob qualquer circunstância específica, o restante da cláusula deverá continuar a ser aplicado e a cláusula, como um todo, deverá ser aplicada em outras circunstâncias.

Esta cláusula não tem a finalidade de induzir você a infringir quaisquer reivindicações de patentes ou direitos de propriedade, nem a contestar a validade de quaisquer reivindicações deste tipo; a única finalidade desta cláusula é proteger a integridade do sistema de distribuição do software livre, o qual é implementado mediante práticas de licenças públicas. Muitas pessoas têm feito generosas contribuições à ampla gama de software distribuído através desse sistema, confiando na aplicação consistente deste sistema; cabe ao autor/doador decidir se deseja distribuir software através de qualquer outro sistema e um licenciado não pode impor esta escolha.

Esta cláusula visa deixar absolutamente claro o que se acredita ser uma consequência do restante desta Licença.

8. Se a distribuição e/ou uso do Programa for restrito em determinados países, tanto por patentes ou por interfaces protegidas por direito autoral, o titular original dos direitos autorais que colocar o Programa sob esta Licença poderá acrescentar uma limitação geográfica de distribuição explícita excluindo esses países, de modo que a distribuição seja permitida somente nos países ou entre os países que não foram excluídos dessa forma. Nesse caso, esta Licença passa a incorporar a limitação como se esta tivesse sido escrita no corpo desta Licença.

9. A Free Software Foundation poderá de tempos em tempos publicar novas versões e/ou versões revisadas da Licença Pública Geral. Essas novas versões serão semelhantes em espírito à presente versão, mas podem diferenciar-se, porém, em detalhe, para tratar de novos problemas ou preocupações.

Cada versão recebe um número de versão distinto. Se o Programa especificar um número de versão desta Licença que se aplique a ela e a "qualquer versão posterior", você terá a opção de seguir os termos e condições tanto daquela versão como de qualquer versão posterior publicada pela Free Software Foundation. Se o Programa não especificar um número de versão desta Licença, você poderá escolher qualquer versão já publicada pela Free Software Foundation.

10. Se você desejar incorporar partes do Programa em outros programas livres cujas condições de distribuição sejam diferentes, escreva ao autor solicitando a respectiva permissão. Para software cujos direitos autorais sejam da Free Software Foundation, escreva para ela; algumas vezes, abrimos exceções para isso. Nossa decisão será guiada pelos dois objetivos de preservar a condição livre de todos os derivados de nosso software livre e de promover o compartilhamento e reutilização de software, de modo geral.

EXCLUSÃO DE GARANTIA

11. COMO O PROGRAMA É LICENCIADO SEM CUSTO, NÃO HÁ NENHUMA GARANTIA PARA O PROGRAMA, NO LIMITE PERMITIDO PELA LEI APLICÁVEL. EXCETO QUANDO DE OUTRA FORMA

ESTABELECIDO POR ESCRITO, OS TITULARES DOS DIREITOS AUTORAIS E/OU OUTRAS PARTES, FORNECEM O PROGRAMA "NO ESTADO EM QUE SE ENCONTRA", SEM NENHUMA GARANTIA DE QUALQUER TIPO, TANTO EXPRESSA COMO IMPLÍCITA, INCLUINDO, DENTRE OUTRAS, AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA. O RISCO INTEGRAL QUANTO À QUALIDADE E DESEMPENHO DO PROGRAMA É ASSUMIDO POR VOCÊ. CASO O PROGRAMA CONTENHA DEFEITOS, VOCÊ ARCARÁ COM OS CUSTOS DE TODOS OS SERVIÇOS, REPAROS OU CORREÇÕES NECESSÁRIAS.

12. EM NENHUMA CIRCUNSTÂNCIA, A MENOS QUE EXIGIDO PELA LEI APLICÁVEL OU ACORDADO POR ESCRITO, QUALQUER TITULAR DE DIREITOS AUTORAIS OU QUALQUER OUTRA PARTE QUE POSSA MODIFICAR E/OU REDISTRIBUIR O PROGRAMA, CONFORME PERMITIDO ACIMA, SERÁ RESPONSÁVEL PARA COM VOCÊ POR DANOS, INCLUINDO ENTRE OUTROS, QUAISQUER DANOS GERAIS, ESPECIAIS, FORTUITOS OU EMERGENTES, ADVINDOS DO USO OU IMPOSSIBILIDADE DE USO DO PROGRAMA (INCLUINDO, ENTRE OUTROS, PERDAS DE DADOS OU DADOS SENDO GERADOS DE FORMA IMPRECISA, PERDAS SOFRIDAS POR VOCÊ OU TERCEIROS OU A IMPOSSIBILIDADE DO PROGRAMA DE OPERAR COM QUAISQUER OUTROS PROGRAMAS), MESMO QUE ESSE TITULAR, OU OUTRA PARTE, TENHA SIDO ALERTADA SOBRE A POSSIBILIDADE DE OCORRÊNCIA DESSES DANOS.

FINAL DOS TERMOS E CONDIÇÕES

Como Aplicar Estes Termos para Seus Novos Programas

Se você desenvolver um programa novo e quiser que ele seja da maior utilidade possível para o público, o melhor caminho para obter isto é fazer dele um software livre, o qual qualquer pessoa pode redistribuir e modificar sob os presentes termos.

Para fazer isto, anexe as notificações seguintes ao programa. É mais seguro anexá-las ao começo de cada arquivo-fonte, de modo a transmitir do modo mais eficiente a exclusão de garantia; e cada arquivo deve ter ao menos a linha de "direitos autorais reservados" e uma indicação de onde a notificação completa se encontra.

<uma linha para informar o nome do programa e uma breve idéia do que ele faz.>

Direitos Autorais Reservados (c) <nome do autor>

Este programa é software livre; você pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU conforme publicada pela Free Software Foundation; tanto a versão 2 da Licença, como (a seu critério) qualquer versão posterior.

Este programa é distribuído na expectativa de que seja útil, porém, SEM NENHUMA GARANTIA; nem mesmo a garantia implícita de COMERCIALIZAÇÃO OU ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA. Consulte a Licença Pública Geral do GNU para mais detalhes.

Você deve ter recebido uma cópia da Licença Pública Geral do GNU junto com este programa; se não, escreva para a Free Software Foundation, Inc., no endereço 59 Temple Street, Suite 330, Boston, MA 02111-1307 USA.

Inclua também informações sobre como contatar você por correio eletrônico e por meio postal.

Se o programa for interativo, faça com que produza uma pequena notificação como esta, quando for iniciado em um modo interativo:

Versão 69 do Gnomovision, Direitos Autorais Reservados (c) ano nome do autor. O Gnomovision NÃO POSSUI QUALQUER TIPO DE GARANTIA; para detalhes, digite 'show w'. Este é um software livre e você é bem-vindo para redistribuí-lo sob certas condições; digite 'show c' para detalhes.

Os comandos hipotéticos `show w' e `show c' devem mostrar as partes apropriadas da Licença Pública Geral. Naturalmente, os comandos que você utilizar poderão ter outras denominações que não `show w' e `show c'; eles poderão até ser cliques do mouse ou itens de um menu - o que for adequado ao seu programa.

Você também pode solicitar a seu empregador (se você for um programador) ou sua instituição acadêmica, se for o caso, para assinar uma "renúncia de direitos autorais" sobre o programa, se necessário. Segue um exemplo; altere os nomes:

A Yoyodyne Ltda., neste ato, reuncia a todos eventuais direitos autorais sobre o programa `Gnomovision' (que realiza passagens em compiladores), escrito por James Hacker.

<Assinatura de Ty Coon>

1º de abril de 1989, Ty Coon, Presidente

Esta Licença Pública Geral não permite a incorporação do seu programa a programas proprietários. Se seu programa é uma biblioteca de sub-rotinas, você poderá considerar ser mais útil permitir a ligação de aplicações proprietárias à sua biblioteca. Se isso é o que você deseja fazer, utilize a Licença Pública Geral de Biblioteca do GNU, ao invés desta Licença.

Este trabalho foi desenvolvido com software livre:

Editor de textos: OpenOffice.org Writer

Editor de figuras vetoriais: OpenOffice.org Draw

Editores de texto puro: KWrite e Vim

Editor de áudio: Audacity

Reprodutor de áudio: XMMS

Gerador de arquivos PDF: Ghostscript e OpenOffice.org

Visualizador de PDF: KGhostView

Visualizador de imagens: KView

Navegadores de Internet: Mozilla e Konqueror

Cliente de correio: Evolution

Ambiente gráfico: KDE sobre XFree86

Sistema de impressão: Cups

Sistemas operacionais: Debian, Conectiva e Kurumin, todos baseados no sistema GNU

Núcleo de sistema operacional: Linux