

LUIZ ALBERTO KOROLL



**UMA AVALIAÇÃO ECONÔMICA DO FENÔMENO DO SOFTWARE
LIVRE, NA INDÚSTRIA DE SOFTWARE DO BRASIL**

Monografia apresentada como requisito parcial para conclusão do Curso de Ciências Econômicas, do Departamento de Economia, Setor de Ciências Sociais Aplicadas, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Huáscar Pessali

CURITIBA

2006

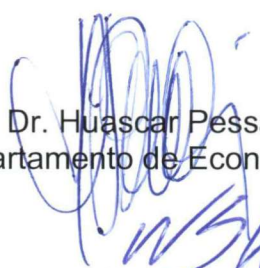
TERMO DE APROVAÇÃO

LUIZ ALBERTO KOROLL

UMA AVALIAÇÃO ECONÔMICA DO FENÔMENO DO SOFTWARE LIVRE, NA INDÚSTRIA DE SOFTWARE DO BRASIL

Monografia aprovada como requisito parcial para conclusão do Curso de Ciências Econômicas, do Departamento de Economia, Setor de Ciências Sociais Aplicadas, da Universidade Federal do Paraná, pela comissão formada pelos seguintes professores:

Orientador:


Prof. Dr. Huascar Pessali
Departamento de Economia, UFPR


Prof. Dr. Walter Tadahiro Shima
Departamento de Economia, UFPR


Prof. Dr. Luciano Nakabashi
Departamento de Economia, UFPR

Curitiba, 10/11/2006.

DEDICATÓRIA

À fé, à esperança e ao sonho de um sistema melhor, dedico este trabalho.

AGRADECIMENTOS

Agradeço primeiramente a Deus, a minha esposa e as minhas filhas, pela compreensão, incentivo e sacrifício compartilhados. Ao orientador Prof . Huáscar Pessali, pelas críticas, paciência, bom senso e sabedoria.

SUMÁRIO

LISTA DE ILUSTRAÇÕES, QUADROS E TABELAS	VI
LISTA DE ABREVIATURAS E SIGLAS	VII
RESUMO	VIII
1 INTRODUÇÃO	9
2 APRISIONAMENTO TECNOLÓGICO.....	10
2.1 APRISIONAMENTO TECNOLÓGICO	10
2.1.1 Comprometimento da cadeia com determinado padrão.....	12
2.1.2 O Ciclo do Aprisionamento	12
2.1.3 <i>Feedback</i> Positivo	14
2.1.4 Estratégia da Evolução	15
2.1.5 Estratégia da Revolução	16
2.1.6 Considerações sobre aprisionamento voltado ao software.....	16
3 DESENVOLVIMENTO DO SOFTWARE LIVRE	21
3.1 HISTÓRIA DO SOFTWARE LIVRE.....	21
3.2 PERSPECTIVAS COMERCIAIS DO SOFTWARE LIVRE	26
3.3 MODELOS DE LICENCIAMENTO DE SOFTWARE	30
3.4 O PROCESSO DE CRIAÇÃO DE SOFTWARE LIVRE	32
3.4.1 Fases de Desenvolvimento de Software Livre	33
3.4.2 Modos de Financiamento.....	34
4 O SOFTWARE LIVRE NO BRASIL	37
4.1 PERFIL DO MERCADO	37
4.1.1 As Comunidades Brasileiras	41
4.2 EXEMPLOS DE UTILIZAÇÃO DE SL NO BRASIL	43
5 CONCLUSÃO	46
REFERÊNCIAS	50

LISTA DE ILUSTRAÇÕES, QUADROS E TABELAS

Figura 1 – Ciclo do aprisionamento	13
QUADRO 1 – TIPOS DE APRISIONAMENTO E CUSTOS DE TROCA RELACIONADOS..	11
QUADRO 2 – O MERCADO DE SOFTWARE.....	39

LISTA DE ABREVIATURAS E SIGLAS

API	Application programming Interface
ARPANET	Advanced Research Projects Agency Network
BASIC	Beginners All-Purpose Symbolic Instruction Code
BSD	Berkeley Software Distribution
DARPA	Defense Advanced Research Projects Agency
DFSG	Debian Free Software Guidelines
CERN	Centro Europeu de Pesquisas Nucleares
FLOSS	Free/Libre and Open Source Software
FSF	Free Software Foundation
GNU	GNU Gnu's Not Unix (projeto do FSF)
GPL	General Public License (FSF)
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines
IDC	International Data Corporation
IP	Internet Protocol
LAMP	Linux, Apache, MySql e PHP
MS	Microsoft
OEM	Original Equipment Manufacturer
OS	Operating System
OSDL	Open Software Development Laboratory
OSI	Open Source Initiative
OSS	Open source software
PC	Personal Computer
PDV	Ponto de Venda
PSL	Projeto Software Livre
RISC	Reduced Instruction Set Computer
TI	Tecnologia da Informação
TCO	Total Cost of Ownership
UFPR	Universidade Federal do Paraná
USENET	Unix User Network
UUCP	Unix yo Unix Copy Protocol

RESUMO

A discussão a respeito do software livre, basicamente prende-se às questões de custo, visto que o software proprietário exige aquisição de licenças de uso, enquanto o software livre não tem esta exigência. A questão do custo de fato é relevante, porém não é o único determinante na decisão das empresas em utilizar software livre, existem outros pontos a serem considerados, tais como as vantagens estratégicas, custos de mudança e outros fatores condicionantes da expansão do software livre. No estudo estaremos tentando identificar questões estratégicas e fatores restritivos à adoção de software livre, baseando-se na teoria do aprisionamento tecnológico e *lock in path dependence*, assim como obter um *overview* da indústria de software no Brasil.

PALAVRAS-CHAVE: software livre, estratégia e aprisionamento

1 INTRODUÇÃO

A utilização de recursos de tecnologia da informação (TI) é essencial para qualquer organização, daí surge a necessidade de utilizar software para a operação dos processos administrativos, organizacionais e operacionais, gerando uma grande demanda por programas para computadores. A aquisição destes programas é onerosa em função dos custos de licenciamento. O software livre vem se apresentando como alternativa de baixo custo em relação ao software proprietário, tanto do ponto de vista estratégico para as corporações como para os governos dos países em desenvolvimento. No estudo estaremos buscando identificar questões estratégicas e fatores restritivos à adoção de software livre, baseando-se na teoria do aprisionamento tecnológico e *lock in path dependence*.

No primeiro capítulo serão expostos pontos relevantes da teoria do aprisionamento tecnológico, no segundo capítulo será apresentada a origem, o desenvolvimento do fenômeno do software livre e sua dinâmica. No terceiro capítulo serão citados alguns dados sobre o software livre e a indústria de software do Brasil, finalizando serão apresentadas as conclusões do estudo.

2 APRISIONAMENTO TECNOLÓGICO

Visando ajudar no entendimento do trabalho, neste capítulo estaremos abordando a teoria do aprisionamento tecnológico.

2.1 APRISIONAMENTO TECNOLÓGICO

Quando uma empresa adota uma determinada tecnologia, pode ocorrer dificuldade de troca dessa tecnologia por outra, criando uma dependência e caracterizando o aprisionamento tecnológico. Como consequência, devido a incompatibilidades entre as tecnologias, ocorrem os custos de mudanças. No caso da tecnologia de informação ao optar por um determinado *software*, poderá haver problemas para migrar para outro *software*, devido à incompatibilidade entre os tipos de arquivos, impossibilidade de migrar diretamente os dados, necessidade de treinamento para usar o novo software, riscos operacionais do negócio, etc.

A substituição de um sistema ou tecnologia, geralmente implica em custos de mudança significativos, razão pela qual, muitas vezes os gestores dos sistemas de TI , preferem continuar utilizando as tecnologias já em uso, mantendo deliberadamente o aprisionamento. Existem determinados momentos nos quais a tecnologia torna-se obsoleta e precisa ser substituída, não obstante, mesmo nestas ocasiões, os custos de mudança interferem nas estratégias e opções das empresas.

No quadro 1, mostramos de forma resumida os tipos de aprisionamento e os respectivos custos de troca:

QUADRO 1 – TIPOS DE APRISIONAMENTO E CUSTOS DE TROCA RELACIONADOS

Tipo de Aprisionamento	Custos de mudança
Compromissos contratuais	Indenizações compensatórias ou liquidadas.
Compra de bens duráveis	O custo de substituição de equipamento tende a cair à medida que o bem durável envelhece.
Treinamento em marca específica	Aprender um novo sistema demanda tempo e incorre em custos, que tendem a aumentar com o tempo.
Informação e banco de dados	Conservação de dados para o novo formato.O custo tende a aumentar com o tempo, pois a quantidade de dados aumenta.
Fornecedores especializados	Financiamento de novo fornecedor tende a ser maior quanto mais difícil for encontrar um novo fornecedor.
Custos de busca	Custos combinados do comprador e fornecedor incluem o aprendizado sobre a qualidade das alternativas.
Programas de lealdade	Quaisquer benefícios perdidos do fornecedor,mais possível necessidade de reconstruir o uso cumulativo.

Fonte: SHAPIRO e VARIAN (2003, p. 140)

É importante observar que os padrões levantados por Shapiro e Varian, são válidos para um amplo espectro de situações, podendo ser analisados tanto do ponto de vista do consumo como da demanda. Na indústria da tecnologia da informação existe um modelo baseado no oligopólio, que induz ao aprisionamento pela própria natureza do negócio.

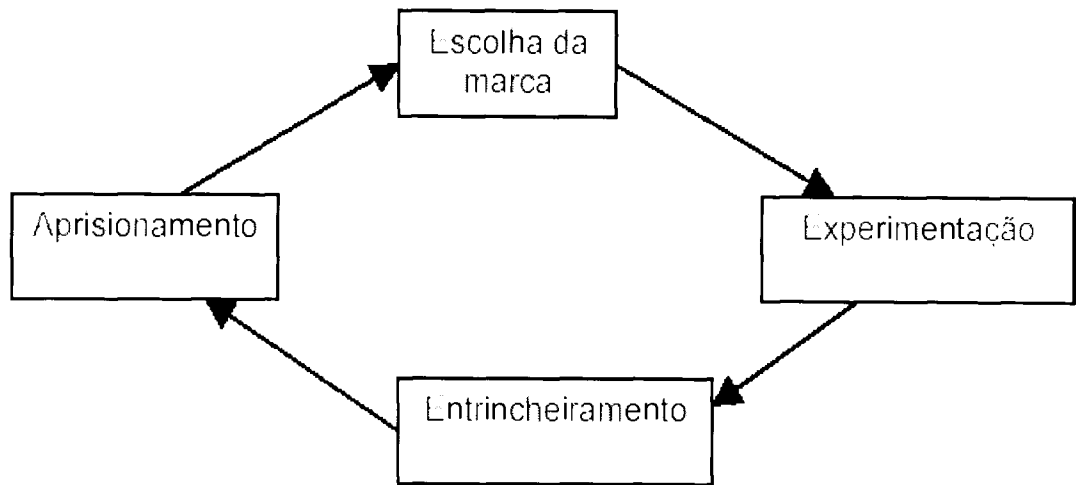
2.1.1 Comprometimento da cadeia com determinado padrão

A produção da indústria de software tende a ser compatível com o padrão dominante do mercado, ou seja, no caso de software para computadores pessoais (PC), o Windows é o padrão de mercado. As empresas tenderão a adquirir programas compatíveis com esse ambiente e produzirão programas que vão reproduzir tal necessidade de padronização. Assim, a oferta de produtos, passa a ser condicionada pelo padrão dominante, qualquer mudança exige alterações tecnológicas tanto no produtor como no cliente.

2.1.2 O Ciclo do Aprisionamento

O ciclo do aprisionamento inicia-se no momento da seleção de uma marca. Após adotar a primeira marca ou tecnologia incorre-se no aprisionamento, iniciando o ciclo do aprisionamento, diminuindo a liberdade para escolher a próxima marca. Após a escolha da marca, entra-se na fase de experimentação, na qual o usuário estará testando a marca e usufruindo as vantagens e incentivos que receber da mesma.

Em seguida entra-se na fase do entrincheiramento, quando ocorre o aprofundamento e o usuário acostuma-se a nova marca, dando preferência a ela em relação às outras marcas. Quanto mais tempo durar esta fase, maiores serão os custos de troca, levando ao aprisionamento, onde os custos de mudança tornam-se altos demais. A figura 1 mostra este processo cíclico do aprisionamento.

FIGURA 1 – CICLO DO APRISIONAMENTO

Fonte : Shapiro e Varian (2003)

Após o primeiro ciclo, a escolha da marca já não mais ocorrerá livremente, uma vez que podem existir contratos com os fornecedores, os usuários estão acostumados com a marca utilizada, os arquivos de dados estão em determinado formato, etc. Assim, os custos de troca tornam-se mais altos, dificultando a migração para outras novas tecnologias que possam ter surgido. Portanto para uma empresa diminuir seu aprisionamento, torna-se necessário que ela consiga antecipar vários ciclos no futuro ao traçar estratégia, procurando minimizar a dependência de determinada tecnologia.

O software livre, embora também mantenha custos de troca, possibilita redução no aprisionamento, uma vez que tem por característica utilizar padrões abertos.

2.1.3 *Feedback* Positivo

O *feedback* positivo ocorre quando um sistema adquire status de padrão de mercado, ou seja, se beneficia do fato de ter maior número de usuários, induzindo que os novos usuários também utilizem o mesmo sistema. Quanto maior for a base instalada de usuários de um determinado sistema, maior será o número de novos usuários a utilizá-lo.

Por exemplo, o sistema operacional Microsoft Windows, junto com a arquitetura de hardware Intel, induz as pessoas a fazerem treinamento para o uso deste sistema operacional, assim como uma nova empresa tenderá a utilizar este sistema devido a maior facilidade de encontrar pessoas treinadas em operá-lo. Além disso, este *feedback* positivo ao sistema operacional Windows, gerou um círculo virtuoso estimulando o mercado a produzir uma enorme gama de programas e dispositivos de hardware compatíveis com este sistema, tornando-o ainda mais atraente para novos clientes.

No entanto, o *feedback* não age somente de forma favorável, podendo acelerar o fracasso de determinado produto, caso o mercado não entenda o produto como promissor. Assim, o produto entra no chamado círculo vicioso ou *feedback* negativo.

Na economia da informação o *feedback* positivo em seu ciclo virtuoso reforça ainda mais a posição de líder, através das economias de escala do lado da demanda, onde um produto como o MS Windows torna-se uma razão para ser usado devido a grande quantidade de pessoas que utilizam esse sistema. Isso faz

com que qualquer sistema operacional concorrente que surja tenha dificuldade para concorrer com o sistema da Microsoft.

A seguir serão discutidas duas estratégias colocadas por Shapiro e Varian, para lidar com os problemas inerciais causados pelo *feedback* positivo e pelo aprisionamento tecnológicos:

2.1.4 Estratégia da Evolução

Como o próprio nome diz, esta estratégia tem o objetivo de fazer a migração para a nova tecnologia em um processo evolutivo, mantendo a compatibilidade com a tecnologia anterior sem ruptura, reduzindo os custos de troca e o aprisionamento.

Podem haver dificuldades técnicas e legais para a adoção desta estratégia, tais como, a dificuldade de criar uma nova tecnologia que se mantenha compatível com a tecnologia anterior e ao mesmo tempo possua recursos inovadores para atrair novos consumidores. Por outro lado, a questão legal surge da possibilidade de não obter o direito de produzir produtos similares aos já existentes, devido a patentes ou propriedade intelectual de outros fabricantes.

As empresas produtoras de software proprietário vêm buscando criar legislação para proteger a propriedade intelectual e patentes de software, impondo barreiras para entrada de novas tecnologias e novos concorrentes no mercado.

2.1.5 Estratégia da Revolução

Esta estratégia parte do princípio da ruptura com a tecnologia anterior, oferecendo uma tecnologia com uma grande quantidade de inovações e melhorias técnicas. Primeiramente atrai os clientes mais preocupados com o desempenho, que são os usuários pioneiros e normalmente são especialistas da área, os quais possuem a capacidade de influenciar os clientes de massa. Ao conquistar esses usuários pioneiros, os fabricantes buscam utilizar a influência desses clientes para começar a impulsionar o *feedback* positivo, fazendo com que os consumidores acreditem que o sucesso do produto é inevitável.

Esta estratégia tende a proporcionar melhores resultados em um mercado em rápido crescimento e com baixo aprisionamento. No entanto, a estratégia da revolução tem um ingrediente de risco, uma vez que é quase impossível saber se a nova tecnologia obterá sucesso ou não.

2.1.6 Considerações sobre aprisionamento voltado ao software

Para as empresas, a forma de aprisionamento mais crítica está relacionada à preservação e conservação dos dados e sistemas já existentes, podendo inviabilizar a adoção de um novo software. A empresa que decidir substituir um software precisa garantir que os sistemas que possui possam trocar dados, mantendo a compatibilidade. Por exemplo, quando se deseja trocar de banco de dados, é necessário garantir que os programas que acessavam os dados do banco de dados antigo consigam acessar a nova base de dados. Outro exemplo se refere à mudança

de sistema operacional, onde programas feitos para o sistema atual podem não funcionar no novo sistema operacional, assim como a compatibilidade de hardware.

O comprometimento da cadeia com determinado padrão poderá agravar o problema da incompatibilidade, pois o sistema operacional líder de mercado, tenderá a possuir um número maior de sistemas e programas feito para esta plataforma dominante, limitando assim, as opções de aplicativos disponíveis para os sistemas operacionais concorrentes.

Shapiro e Varian (2003, p 204), colocam uma diferença essencial entre a velha e a nova economia: a velha economia industrial era movida pelas economias de escala, a nova economia da informação é movida pela economia de redes. Dentro deste conceito, o *feedback* positivo, ou a aprovação do mercado, onde o sucesso gera mais sucesso, induzem a formação de redes virtuais em torno de padrões dominantes. O valor de ligar-se a uma rede varia em função do número de pessoas já conectadas a ela, sendo mais interessante conectarem-se a uma rede maior¹.

A combinação das economias de escala do lado da demanda e do lado da oferta tem ocorrido em muitos setores da tecnologia da informação, sendo que o crescimento da demanda reduz o custo do lado da oferta e torna o produto mais atraente para outros usuários – acelerando ainda mais o crescimento da demanda. A consequência é um *feedback* muito forte, que proporciona a criação ou até a destruição de setores inteiros. Esta proposição fundamental é comumente

¹ A lei de Metacalfe: O valor da rede eleva-se como quadrado do número de usuários. Se houver n pessoas em uma rede e o valor da rede para cada uma dessas pessoas for proporcional ao número de outros usuários, então o valor total da rede (para todos os usuários) é proporcional a $n \times (n-1) = n^2 - n$ (Shapiro Varian, 2003, p217).

denominada como efeitos de rede, exterioridades de rede e economias de escala da demanda.

As vantagens da empresa dominante na determinação do padrão são explicáveis de acordo com uma lógica, na qual uma solução tecnológica torna-se crescentemente atrativa, desenvolvida, difundida e útil quanto maior o seu grau de adoção. Assim, a trajetória mercadológica de muitas soluções desenvolvidas em software seria determinada por fatores externos às suas especificações técnicas ou funcionalidades particulares. Para muitos produtos em software o valor depende não apenas de suas características intrínsecas, mas se eleva com o número de usuários da mesma solução compatível. O sucesso ou insucesso mercadológico de um software seria, em grande parte, determinado pelos elementos relacionados à opção tecnológica adotada no segmento para o qual se destina. A eficiência técnica do software, ou mesmo características inovadoras superiores, podem não se impor sobre os produtos concorrentes tecnicamente inferiores graças a essas externalidades. As vantagens competitivas do *first mover* fundam-se na adoção, por parte de uma ampla base de usuários, de uma determinada solução tecnológica. A partir disso, os efeitos resultantes das externalidades positivas associadas à difusão do padrão tecnológico resultam num efeito de *lock-in* dos mercados (Roselino , 2006, p.13).

Uma consequência das externalidades de rede é desafio para a introdução de novas tecnologias no mercado, onde o tamanho da rede deve superar os custos de troca coletivos, isto é, o custo de troca combinado de todos os usuários. Um exemplo citado por Shapiro e Varian (2003), é o formato de teclado QWERTY das máquinas de escrever, que se manteve em detrimento ao modelo mais eficiente

,Dvorak. Uma explicação para isto é que o layout do teclado Dvorak, embora tecnicamente melhor, não seria suficientemente bom para superar os custos de troca coletivos.

Considerando a indústria de software e o fenômeno do software livre, podemos identificar uma correlação, focando o mercado de software para computadores de mesa, pelo fato de haver um padrão dominante estabelecido tanto pelo lado da demanda como do lado da oferta. Sendo, que do lado da demanda, existe um claro *feedback* positivo para a o sistema operacional Microsoft Windows, que por sua vez influencia a indústria como padrão dominante.

Shapiro e Varian(2003), em sua abordagem sobre os padrões dominantes, expõem as estratégias da evolução e da revolução, bem como, os conflitos dos padrões proprietários e abertos. A evolução é o uso de novas tecnologias mantendo a compatibilidade com a tecnologia anterior, a revolução, por outro lado, é a ruptura em busca de um desempenho superior. A restrição do uso de determinado padrão, a proteção de patentes, caracteriza um padrão proprietário sobre a tecnologia, normalmente utilizado como barreira à entrada. Por outro lado o padrão aberto, cuja filosofia é permitir qualquer um utilizar e desenvolver baseado no padrão proposto, é a estratégia básica do fenômeno do software livre.

Os padrões abertos e a Internet forneceram o cenário ideal para o desenvolvimento do fenômeno do software livre, tendo como base o desenvolvimento compartilhado e utilizando a filosofia da abertura total como estratégia para a melhoria constante. A Microsoft por sua vez tem sido acusada de

manter fechadas certas interfaces de programas de aplicação (APIs)² , limitando a capacidade de fornecedores de software em desenvolver aplicações para a plataforma baseada no sistema operacional Windows.

² Do inglês, Application programming interfaces.

3 DESENVOLVIMENTO DO SOFTWARE LIVRE

Neste capítulo faremos um relato sobre desenvolvimento do movimento do software livre, buscando a origem deste modelo de desenvolvimento de software e sua dinâmica.

3.1 HISTÓRIA DO SOFTWARE LIVRE

A história do software livre se confunde com origem do sistema operacional Unix e da Internet. Em 1969, as forças armadas do Estados Unidos, financiaram a criação da ARPANet³, projetada para trocar informações entre laboratórios de pesquisa, universidades e entidades de defesa. No mesmo ano, os desenvolvedores dos laboratórios Bell e da AT&T, criaram o sistema operacional UNIX (Lerner et al, 2000, p. 4). Nesta época a AT&T era um monopólio na área de telecomunicações e em função de restrições de regulamentação não podia atuar na área de software, então a AT&T passou a fornecer e divulgar nas universidades o Unix e os programas com os códigos fonte completos. A partir de então se criaram encontros científicos em torno do Unix, onde surgiram novas idéias e melhoramentos que foram rapidamente absorvidos pelo sistema. Esta prática, além de rapidamente promover o desenvolvimento deste software, fez também com que surgissem diversos sistemas operacionais baseados em Unix.

³ Arpanet, do inglês *Advanced Research Projects AgencyNetwork*, rede criada com o objetivo de ser uma alternativa de comunicação que continuasse funcionando mesmo que parte da rede fosse destruída, por exemplo, por um ataque nuclear.

Em 1970 foi criada a linguagem de programação C⁴, que passou a ser utilizada para compilar o sistema operacional Unix, permitindo assim a portabilidade do sistema operacional Unix em hardware diferentes. Isto permitiu criar software para executar em um sistema operacional, independente da plataforma física de hardware.

Nesta época os programas para computadores pessoais eram distribuídos e compartilhados entre entusiastas da microinformática, que se denominavam hobbistas, os quais criavam, modificam e trocavam programas entre si. Contudo este método de desenvolvimento não foi suficiente para atender a demanda crescente de programas e sistemas, oriunda do barateamento dos computadores pessoais.

Surgiu então um modelo comercial para suprir esta necessidade do mercado, até então haviam soluções somente para o mercado composto pelas grandes corporações e para os computadores de grande porte. Bill Gates, foi um dos primeiros a observar esta demanda e já em 1975 desenvolveu um interpretador da linguagem de programação BASIC, para o microcomputador Altair, um dos primeiros computadores pessoais. O programa Altair BASIC estava disponível somente para venda e sem o código fonte, contudo entre a comunidade de hobbistas era comum a trocar cópias entre si.

Bill Gates considerou isto um roubo e expressou sua insatisfação em uma carta aberta aos hobbistas, escrita em 1976:

⁴ C é uma linguagem de programação estruturada e padronizada criada na década de 1970 por Dennis Ritchie e Ken Thompson para ser usada no sistema operacional UNIX. Ver http://pt.wikipedia.org/wiki/Linguagem_C.

"...As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid? Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software".⁵

Na visão de Gates, o mercado jamais conseguiria softwares e documentação de qualidade sem um incentivo financeiro.

Em 1980 foi criada a USENET⁶, usando um protocolo de comunicações UUCP⁷, cujo meio físico eram linhas telefônicas. Nesta época também foi criado o protocolo TCP/IP⁸, criado pela DARPA (Defense Advanced Research Projects Agency), foi utilizado inicialmente para comunicação a distância entre sistemas operacionais Unixes, modelo que viria a se tornar padrão na Internet.

Os sistemas Unix já eram amplamente utilizados. Em 1984 o governo norte-americano dividiu a AT&T em várias companhias independentes, permitindo a ela

⁵ <http://www.blinkenlights.com/classiccmp/gateswhine.html>. Acesso em: 19 Out 2006.

⁶ Usenet, do inglês *Unix User Network*, é um meio de comunicação onde usuários postam mensagens de texto em fóruns que são agrupados por assunto, chamados de newsgroups. Ao contrário das mensagens de e-mail, que são transmitidas quase que diretamente do remetente para o destinatário, os artigos postados nos newsgroups são retransmitidos através de uma extensa rede de servidores interligados.

⁷ UUCP, do inglês *Unix to Unix Copy Protocol*.

⁸ Do inglês, Transmission Control Protocol/Internet Protocol – Protocolo de Controle de Transmissão/Protocolo Internet

criar uma subsidiária que operasse no ramo de informática, levando o Unix a ter código fonte fechado passando a ser cobrado pela licença de uso.

Ainda em 1984, Richard Stallmann, que trabalhava no MIT, decidiu criar um sistema operacional que qualquer pessoa poderia copiar, usar, modificar e distribuir. Segundo Stallman, o desenvolvimento de um software deveria ocorrer de forma evolucionária, no qual um programador pudesse, a partir de um programa já existente, adicionar melhorias, novas funcionalidades, novos recursos e até mesmo criar novos programas sem a necessidade de começar do zero.

Stallman criou o projeto GNU (Gnu Is Not Unix – Gnu não é Unix), cujo objetivo específico era criar um sistema operacional completo e distribuí-lo para quem quisesse, gratuitamente. Naquele mesmo ano Stallman publicou o GNU Manifesto, onde declarava seus princípios:

"I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free".⁹

Foi criada a licença GNU *General Public License* (GPL), segundo a qual qualquer pessoa poderia copiar, executar, alterar, modificar e distribuir o programa, desde que o mesmo permaneça livre e que as modificações do projeto ou extensões do mesmo, sejam livres, passando adiante a liberdade de copiar e modificar

⁹ <http://www.gnu.org/gnu/manifesto.html>. Acesso em 19 Out 2006.

novamente o projeto. Posteriormente esta licença foi chamada de "Copyleft", um trocadilho substituindo o termo "right" por "left", uma alusão à ideologia política envolvida nos movimentos de democratização da informação.

No ano de 1985, Stallman criou a FSF (*Free Software Foundation* – Fundação Software Livre) e o conceito de "*Free Software*", ressaltando o significado de liberdade e não de gratuidade, sendo comum encontrar o seguinte comentário em textos que falam sobre software livre em inglês:

"Free software is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer'".¹⁰

A FSF¹¹ define quatro liberdades fundamentais para o software livre:

- A liberdade para usar o programa, para qualquer propósito.
- A liberdade para estudar como o programa funciona, e adaptá-lo para suas necessidades.
- A liberdade para distribuir cópias, para que você possa ajudar o próximo.
- A liberdade para melhorar o programa e disponibilizá-lo com suas melhorias ao público, para que toda a comunidade se beneficie.

¹⁰ <http://www.gnu.org/philosophy/free-sw.html>. Acesso em: 19 Ago. 2006.

¹¹ <http://www.fsf.org/licensing/essays/free-sw.html> Acesso em: 19 Ago. 2006.

Em 1990 o sistema GNU estava praticamente completo, no entanto faltava desenvolver o núcleo, programa que faz a interação diretamente com o hardware, componente essencial para qualquer sistema operacional.

Em 1991, Linus Torvalds, do Departamento de Ciência da Computação da Universidade de Helsinque na Finlândia, criou um núcleo compatível com o Unix, que recebeu o nome de Linux. Torvalds, postou o código fonte do Linux na USENET, para que a comunidade de programadores o ajudassem a melhorar o programa. Em 1992, o kernel Linux foi incorporado aos programas GNU, gerando um sistema operacional livre e completo, que recebeu o nome de GNU/Linux, atualmente conhecido somente como Linux, sendo distribuído por diversas empresas com diferentes denominações.

3.2 PERSPECTIVAS COMERCIAIS DO SOFTWARE LIVRE

De acordo com Eric S. Raymond ¹², o público em geral teve acesso a Internet a partir da década de 1990, atraindo o interesse comercial e realimentado o *feedback* positivo para este novo ambiente. Da mesma forma a cultura do Unix deslocava sua atenção ao fenômeno Linux e para a maneira diferente de fazer o software, agora com código fonte aberto desenvolvido de forma compartilhada.

O fenômeno do software livre teve sua origem basicamente em universidades e centros de pesquisa, permeada da ideologia do livre acesso ao conhecimento e a pesquisa. Embora esta filosofia fosse endossada por muitos desenvolvedores,

¹² <http://www.catb.org/~esr/writings/cathedral-bazaar> Acesso em 19 Out 2006.

sendo parte essencial do fenômeno do software livre, esta posição política não oferecia o suporte para a indústria de software comercial. Em 1998, logo após a Netscape ter anunciado que abriria o código fonte dos seus browsers¹³, foi proposto por alguns desenvolvedores, mudar o nome do “software livre” para “software de código fonte aberto” ou em inglês “Open Source Software” (OSS). Esta mudança de nome baseava-se puramente em razões práticas: a indústria de software tinha interesse no fenômeno, mas o rótulo “software livre” não dava credibilidade ao negócio. A intenção era mudar o rótulo e a imagem do software livre, dando um nome que o mundo do negócio pudesse adotar e vender. A intenção era criar confiabilidade para a utilização do “Open Source Software” por empresas comerciais, priorizando as vantagens técnicas em relação aos princípios ideológicos:

“The winning substance has not changed, the losing attitude and symbolism have”¹⁴

O termo Open Source foi criado para estimular o uso do software livre através de argumentos pragmáticos:

“We realized it was time to dump the confrontational attitude that has been associated with ‘free software’ in the past and sell the idea strictly on the same pragmatic, business-case grounds that motivated Netscape. We brainstormed about tactics and a new label. ‘Open source’, contributed by Chris Peterson, was the best thing we came up with.”¹⁵

¹³ Browser – programa usado para fazer a conexão com sites Web.

¹⁴ <http://www.fsf-europe.org/documents/freesoftware.en.html> Acesso em 19 Out 2006.

¹⁵ <http://www.opensource.org/does/history.php> Acesso em 19 Out 2006.

O software para ser classificado como *Open Source*, deve atender alguns pré-requisitos:

- A licença não deve restringir a redistribuição do programa e de trabalhos derivados;
- O código-fonte deve estar disponível junto com o programa, ou deve estar disponível para download gratuitamente;
- A licença não deve discriminar pessoas ou grupos, nem restringir a forma de uso do programa (por exemplo, não deve restringir o uso comercial).¹⁶

Embora mantendo regras para garantir o acesso ao código fonte, Stallman, não aprovou o uso do termo *Open Source*, uma vez que desviava a idéia de liberdade como um valor em si mesmo.

A discussão sobre o nome e as políticas comerciais relacionadas ao software livre ainda existem. Recentemente foi criado o termo FLOSS (Free/Libre and Open Source Software) para descrever o fenômeno inteiro ou a comunidade, incluindo assim dentro desta denominação os movimentos do software livre e do software aberto. Neste estudo as referências a software livre e software aberto referem-se ao fenômeno inteiro. A questão filosófica e política do confronto da comunidade OSS com os interesses comerciais, podem ser vista como o confronto de uma sub-cultura, composta pelos membros da comunidade desenvolvedora, que vêem na comercialização uma ameaça a essa sub-cultura. Em outro plano, a competição de

¹⁶ <http://www.opensource.org/docs/definition.html> Acesso em 19 Out 2006.

ícones do software, tais como o sistema operacional Linux e produtos derivados, de encontro às companhias comerciais do software tais como Microsoft , Oracle e IBM, oferece um inimigo comum e parece criar a coesão na comunidade.

O software de OSS está disponível na Internet, esta, por sua vez, é o meio para a troca do código de fonte e outras informações entre as comunidades, sendo utilizada para a comunicação e produção. As companhias de software proprietário tentaram adotar alguns dos aspectos bons do software aberto, tentando criar comunidades de usuário, fornecendo o código de fonte de alguns softwares para finalidades educacionais, avaliando possibilidades de negócios oferecidas com a abertura de produtos de software e revendo modelos do negócio. Um bom exemplo disso é a iniciativa shared source, da Microsoft:

*" Through the Shared Source Initiative, Microsoft is sharing source code with customers, partners, and governments worldwide. The Shared Source Initiative encompasses the spectrum of programs and licenses offered by Microsoft to various communities of customers, partners, developers, and other interested individuals. "*¹⁷

A licença *shared source*, não se caracteriza como um software livre, pois retêm a propriedade intelectual, de maneira que as instituições selecionadas para receber o código fonte não têm liberdade para usá-lo, modificá-lo ou distribuí-lo. Por isto é mais adequado utilizar o termo software livre, para referir-se a softwares de código aberto, ou *open source*.

¹⁷ <http://www.microsoft.com/resources/sharedsource/Initiative/Initiative.msp> . Acesso em: 15 out. 2006.

Em parte o sucesso da OSS é devido ao fato de que os mercados do software são globais e com a Internet um produto pode alcançar rapidamente todos os países no mundo. Outro ponto facilitador é o fato de a legislação da maioria das nações ser diferente e haver dificuldades para atualizar-se com o ritmo do desenvolvimento tecnológico e social.

3.3 MODELOS DE LICENCIAMENTO DE SOFTWARE

As licenças de software foram criadas para vários propósitos de negócios, tais como definir um escopo nos relacionamentos entre desenvolvedores, consumidores e parceiros. Os negócios de software são baseados no licenciamento de uso dos códigos binários e na propriedade intelectual do código fonte.

Os modelos de licenciamento de software estão baseados nos modelos de desenvolvimento, subdividindo o ecossistema do software basicamente em duas categorias: software proprietário e software livre.

No modelo de licenciamento de software proprietário ou comercial, o acesso ao código fonte é limitado e restritivo devido ao fato de que o valor e a viabilidade comercial, está no conhecimento único do código fonte do produto pelo desenvolvedor.

O modelo de licenciamento “Open Source Software” (OSS), se divide em duas categorias: licenças não restritivas e licenças restritivas. As licenças não restritivas disponibilizam o código fonte e impõe condições não significativas sobre o uso subsequente do software e seu código fonte.

Assim o desenvolvedor pode utilizar o código licenciado em novos programas, incluindo aqueles que serão vendidos como produtos comerciais e licenciados sob novos termos definidos pelo novo desenvolvedor. Por esta razão, estas licenças são consideradas verdadeiramente livres. Um exemplo deste tipo de licença é Berkeley Software Distribution (BSD) .

Licença restritiva de software livre , em contraste, disponibiliza livremente o código fonte, mas para evitar seu uso em desenvolvimento de software proprietário, requer que qualquer trabalho derivado também seja licenciado pelos termos do código fonte original. Os exemplos os mais comuns deste tipo de licença são a licença GNU (GPL), esta licença é referenciada como viral, fazendo alusão ao fato de que software derivado deve passar a usar a licença GPL.

O tema licenciamento de software é um tópico complexo, desenvolvedores de software utilizam diversas licenças, que incluem um grande número de termos divergentes e específicos. Embora seja pertinente ao assunto software livre, neste trabalho não estudaremos detalhadamente os métodos de licenciamento de software. No site da “Free Software foundation”¹⁸, existem mais informação sobre categorias de licenças de software.

¹⁸ <http://www.fsf.org/licensing/essays/categories.html> Acesso em 19 Out 2006.

3.4 O PROCESSO DE CRIAÇÃO DE SOFTWARE LIVRE

O processo de criação de um software livre se inicia com o surgimento da primeira versão do código, com sua posterior publicação pelo autor ou coordenador, na Internet. Alguns usuários e também desenvolvedores de software melhoram o código e o retornam ao coordenador. Este absorve as alterações e reinicia o ciclo.

Com o aumento do tamanho dos projetos surgiram sites centralizadores dos softwares livres entre eles The Apache Group, Mozilla.org, Openoffice.org, Linux Online. Nestes sites os programadores podem transferir os códigos fontes, comunicar erros e falhas, apresentar correções e novos recursos, trocar experiências, realizar documentação e realimentar o ciclo de desenvolvimento.

No seu artigo, *The Cathedral and the Bazaar*¹⁹, Eric S. Raymond descreve dois modos de organização mais empregados no desenvolvimento de software, que o autor chama de catedral e bazar.

O modelo tradicionalmente empregado na indústria é similar ao projeto de uma catedral medieval, no qual um restrito grupo de projetistas exerce controle sobre o trabalho de um pequeno exército de operários. O modelo da catedral é empregado na maioria dos projetos de desenvolvimento de software proprietário, onde o modelo descreve o relacionamento entre a gerência de projeto, que estabelece metodologias, tarefas e prazos, que devem ser cumpridos pelos programadores engajados no projeto.

¹⁹ <http://www.catb.org/~esr/writings/cathedral-bazaar/> Acesso em: 19 Ago. 2006

O outro modo de organização, freqüentemente empregado pela comunidade de software livre se assemelha a um anárquico bazar, onde não há hierarquia entre os participantes e todos cooperam para que o bazar seja atrativo aos compradores, ao mesmo tempo em que competem pela atenção destes mesmos compradores. Na produção de software no bazar os projetos são informalmente organizados ao redor da proposta de desenvolvimento de algum aplicativo, do qual os interessados participam voluntariamente, e o líder do projeto emerge por seus méritos como programador ou projetista.

3.4.1 Fases de Desenvolvimento de Software Livre

O ciclo de vida de um projeto de software livre passa pelas seguintes fases (Hexsel, 2002): o processo se inicia quando o autor escreve uma versão inicial do programa Pv0 e publica o código fonte desta versão. Se o programa cria interesse da comunidade, outros programadores o instalam e experimentam. Erros são descobertos e corrigidos, e melhorias são propostas ou introduzidas no programa. Estas correções e melhorias são submetidas ao autor, que as incorpora e publica a nova versão de seu programa, Pv1. A versão melhorada atrai mais usuários, que fazem *feedback* e introduzem novas melhorias, o que leva a uma nova versão. Para programas considerados úteis, este ciclo se repete a cada poucos meses. Após algumas iterações, o programa P atinge estabilidade e passa a contar com um grupo razoavelmente grande de usuários. Nestas condições, a comunidade de suporte ao programa atinge massa crítica, e isso garante a continuidade de seu desenvolvimento e suporte.

Em grande parte, a evolução da funcionalidade de um programa desenvolvido “no bazar” é orientada pelos usuários do programa, especialmente aqueles que são desenvolvedores ou programadores. Por outro lado, em uma “catedral” ou na indústria, a orientação é definida pelos projetistas e pressões de mercado.

Dependendo da popularidade ou importância de um programa, a comunidade de usuários e desenvolvedores rapidamente atinge massa crítica, com centenas de programadores e milhares de usuários participando da tarefa de testá-lo e eventualmente corrigi-lo. Por outro lado se o projeto não atrai a comunidade, é provável que o projeto venha a estagnar. Isto não é necessariamente ruim, porque pode existir algum outro programa com funcionalidade similar mas com melhor qualidade.

3.4.2 Modos de Financiamento

Os sistemas de software livre são financiados indiretamente pelos vários níveis de governo e academia e segmentos da indústria nos modos listados abaixo:

- Voluntários individuais

Há um grande número de desenvolvedores de software livre que trabalham em universidades e institutos de pesquisa. O desenvolvimento é parcialmente financiado pela instituição, como por exemplo o desenvolvimento do sistema BSD-Unix na Universidade de Berkeley, o protocolo HTTP no Centro Europeu de Pesquisas Nucleares (CERN).

- Suporte indireto

Em médias e grandes empresas ou organizações governamentais, funcionários podem participar do desenvolvimento de projetos fora do escopo de interesse da empresa, como a primeira versão do Unix e da linguagem C no Bell Labs, embora estes sistemas não tenham sido publicados como software livre.

- Suporte parcial

Existem empresas que financiam o desenvolvimento de software livre como forma de promover a venda de seus equipamentos ou como estratégia de suporte a outros negócios. Como exemplos pode-se citar a IBM que suporta o desenvolvimento do Apache, que é instalado nos computadores da série AS400, e a Silicon Graphics que suporta desenvolvimento do Samba²⁰. Outra forma de suporte parcial é representada pelo consórcio que está financiando a instalação e operação do Open Software Development Laboratory (OSDL)²¹, para suportar o desenvolvimento de aplicativos para sistemas de grande porte, especialmente sistemas empresariais com características de alta confiabilidade e disponibilidade. Destacam-se entre os participantes Hewlett-Packard e IBM.

- Suporte direto

Existem empresas que financiam o desenvolvimento de software livre para ganhar mercado ou competir diretamente no mercado, como a Netscape com o desenvolvimento do browser Mozilla. A Sun Microsystems lidera o desenvolvimento

²⁰ Samba permite o compartilhamento de sistemas de arquivos de Microsoft, instalados em máquinas Unix, através da rede.

²¹ <http://www.osdlab.org>. Acesso em: 20 Ago. 2005.

da linguagem Java, ou empresas como Red Hat e Conectiva que vendem serviços e produtos baseados em software livre.

4 O SOFTWARE LIVRE NO BRASIL

Não existe base de dados ou informações estatísticas abrangentes sobre o assunto software livre no Brasil, neste capítulo estaremos colocando dados sobre a indústria de software no Brasil e em alguns casos, para efeito comparativo dados de outros países.

4.1 PERFIL DO MERCADO

Pesquisa realizada em 2005 pelo Softex²², revela perfil do usuário de software livre e código aberto no Brasil: as empresas que utilizam software livre e código aberto no Brasil, são, em sua maioria, de grande porte, dos setores de tecnologia da informação e comunicações, governo, comércio e educação, 64% faturam acima de R\$ 1 milhão, sendo que destas, 48%, faturam acima de R\$ 50 milhões. 87% são empresas nacionais (Softex, 2005, p.47). O perfil dos desenvolvedores brasileiros de software é semelhante ao perfil Europeu, é bastante profissionalizado e qualificado: 42% tem nível superior e 65% é administrador de sistemas ou técnico de redes. As empresas desenvolvedoras são predominantemente pequenas empresas. Quanto a localização geográfica, os desenvolvedores estão concentrados principalmente na região Sudeste do país (52%) e na região Sul (26%)²³. As duas regiões somam 78% dos respondentes, evidenciando a disparidade existente no avanço do software livre nas diferentes regiões do país (Softex, 2005, p.22).

²² <http://golden.softex.br/portal/softexweb/uploadDocuments/pesquisa-swl.pdf>. Acesso em: 22 Nov. 2006.

²³ O maior número de desenvolvedores está concentrado no Estado de São Paulo (32%), seguido de Rio de Janeiro (10%) e Rio Grande do Sul (10%).

As principais motivações para utilização de software livre são a redução de custo, o desenvolvimento de novas habilidades técnicas, além de questões ideológicas. Existem recomendações e diretrizes para adoção ou migração para soluções de software livre nas diversas esferas da administração pública, tanto em países em desenvolvimento como em países desenvolvidos²⁴. No Brasil o governo federal, criou o Guia Livre – Referência de Migração para Software Livre do Governo Federal²⁵, o principal objetivo do Guia Livre é ajudar os administradores públicos a definir uma estratégia de migração planejada e gerenciada, o Guia pode servir também de referência para empresas que queiram optar pela substituição de programas proprietários por livres.

No mundo corporativo, o software livre já é largamente empregado em servidores, foi inclusive criado o acrônimo “LAMP”, que significa a combinação dos softwares Linux, Apache, MySql e as linguagens de programação Perl, PHP e Python. A migração de servidores para o software livre pode ser feita, de forma relativamente segura e transparente para os usuários, normalmente é por onde se começa.

Já o uso de software livre nas estações de trabalho, embora possa proporcionar uma maior economia de custos, considerando a escala, ainda envolve maiores desafios. Mostrando-se como um potencial nicho de mercado para os desenvolvedores e prestadores de serviços de TI.

²⁴ <http://europa.eu.int/idabc/en/document/2623>. Acesso em : 20 Nov 2005.

²⁵ http://www.governoeletronico.gov.br/governoeletronico/publicacao/down_anexo.wsp?Imp.arquivo=F15_469GuiaLivre_v099.pdf. Acesso em 19 Out 2005.

No Quadro 2, podemos comparar a dimensão do mercado de software do Brasil com diversos países:

QUADRO 2 – O MERCADO DE SOFTWARE

País	Faturamento (US\$ milhões)	Exportações (US\$ milhões)	Empregados (milhares)	Faturamento/PIB
EUA*	200.000	N.D.	1.042	2.0%
Japão**	85.000	73	534	2.0%
Alemanha	39.844	N.D.	300	2.2%
Inglaterra	15.000	N.D.	N.D.	1.0%
Índia	8.200	6.220	350	1.7%
Brasil	7.700	100	158	1.5%
Coréia do Sul	7.694	35	N.D.	1.8%
Irlanda	7.650	6.500/3.000#	25	7.4%
China	7.400	400	186	0.6%
Espanha	4.330	N.D.	20	0.7%
Taiwan	3.801	349	N.D.	1.2%
Israel	3.700	2.600	15	3.4%
Finlândia	1.910	185	20	1.6%
Singapura	1.660	476	N.D.	1.9%
Argentina	1.340	35	15	0.5%
México	<1.000	N.D.	N.D.	<0.2%

Fonte: Veloso, *et al* (2003)

* 2000; ** 2002; # valor excluindo-se as exportações da Microsoft

De acordo com estudo patrocinado pelo MIT (Veloso, *et al*, 2003), comparando o mercado de software do Brasil, China e Índia, a indústria brasileira

tem uma dimensão de US\$ 7,7 bilhões, sendo que o faturamento tem origem basicamente no mercado interno, as exportações são irrelevantes, contudo apontam para um mercado potencial, observando o desempenho da Índia , cujo volume de exportações é 40 vezes maiores que o Brasil.

O mercado de Sistemas operacionais baseados em Software Livre tem uma dimensão de R\$ 77 milhões (Softex, 2005, p.6) , considerando somente a venda de distribuições Linux e serviços correlatos. O mercado de software livre está fortemente associado a prestação de serviços, uma vez, que é possível obter muitos softwares gratuitamente, que é uma vantagem inicial e um estímulo para a adoção de soluções baseadas em SL, contudo isto gera uma demanda por serviços técnicos especializados. Faltam dados estatísticos para a mensuração deste mercado, bem como das distribuições linux OEM.

Do ponto de vista da concorrência, o software livre ameaça fortemente o modelo de pacotes (plataformas e sistemas operacionais); componentes de software; e produtos customizáveis, exatamente porque esses modelos tem na propriedade intelectual um fator essencial de concorrência. Já os modelos de serviços e de embarcados, por terem maior especificidade e menor importância de apropriabilidade por meio de códigos fechados, são, na verdade, modelos que apresentam as maiores oportunidades de investimento. O software livre acelera a transição da indústria de software dos produtos para os serviços (Softex , 2005, p.74). Diante desta ameaça os produtores de software proprietário, vem adotando ações contrárias ao movimento open source, baseando-se na legislação internacional de patentes e da propriedade intelectual, embora , por outro lado

também estão apoiando ou se associando a projetos baseados no modelo de desenvolvimento open source.

4.1.1 As Comunidades Brasileiras

O perfil das comunidades brasileiras de desenvolvedores de software livre é composto de pequenos grupos, com cinco indivíduos em média, em que tanto são desenvolvedores quanto usuários dos softwares em torno do qual se organizam. É muito freqüente que existam participantes com mais de cinco anos de experiência nestas comunidades, o que representa uma base sólida para que um projeto avance. Alguns exemplos de comunidades brasileiras de desenvolvimento de software são:

- OpenOffice: comunidade dedicada à tradução ao português do Brasil e ao desenvolvimento complementar (dedicados ao usuário brasileiro) do conjunto de ferramentas de escritório OpenOffice.org²⁶.
- Mozilla: comunidade de usuários, desenvolvedores e interessados no Mozilla para o Brasil. Além da tradução das aplicações, o foco é a divulgação e suporte para os usuários do país²⁷.
- MonoBASIC: comunidade de desenvolvimento que se propõe criar um compilador livre para a linguagem VisualBasic.Net, integrado ao projeto Mono²⁸.
- Care2xBrasil: sistema integrado de aplicativos para área da saúde²⁹.

²⁶ <http://openoffice.org.br> . Acesso em 19 Out 2006

²⁷ <http://mozilla.org.br> . . Acesso em 19 Out 2006

²⁸ <http://monobasic.sl.org.br> . . Acesso em 19 Out 2006

²⁹ <http://care2xbr.codigolivre.org.br/> .. Acesso em 19 Out 2006

Os projetos mais conhecidos, segundo pesquisa realizada pelo site br-Linux.org em 2003³⁰, são o Kurumin (distribuição Linux de uso fácil), WindowMaker (ambiente gráfico), txt2tags, rau-tu, dsearch, brazip, slackpkg e sarg.

No Brasil também há uma profusão de comunidades temáticas, cujo escopo costuma variar muito, desde pequenos grupos locais, até grandes grupos nacionais. Exemplos claros são as diversas comunidades intituladas Projeto Software Livre (PSL), onde existem os PSL estaduais (PSL-SC, PSL-RJ, PSL-BA, por exemplo) e os temáticos (PSL-Mulheres, PSL-Jurídico, etc), além do PSL-Brasil, que, em tese, reúne toda a comunidade de software livre brasileira interessada em debater questões estratégicas e articular-se nacionalmente em prol do SL/CA.

A UFPR hospeda um site espelho (*mirror*) do site sourceforge.net, que é o maior site do mundo em desenvolvimento de software livre, lá estão cadastrados 99 mil projetos, desenvolvidos por mais de um milhão de pessoas. Apenas 12 instituições oferecem o serviço na internet e a UFPR é a única da América Latina.

A Companhia de Informática do Paraná – CELEPAR, tem diversos projetos baseados em software livre, que vão desde a migração dos sistemas governamentais para software livre ao desenvolvimento de programas de computador customizados para as diversas áreas do governo paranaense.

O Estado do Paraná economizou 127,3 milhões de reais, de 2003 até agora, com o uso e o desenvolvimento de programas de software livre em órgãos da administração pública direta e indireta. Mais da metade desse valor, 78,5 milhões de

³⁰ <http://brl.inux.Linuxsecurity.com.br/noticias/001434.html#001434> . Acesso em 19 Out 2006

reais, é creditada ao uso de software livre no programa Paraná Digital, da Secretaria da Educação, que está sendo implementado em todas as escolas estaduais³¹.

4.2 EXEMPLOS DE UTILIZAÇÃO DE SL NO BRASIL

Com objetivo de dar ao leitor informações práticas sobre a utilização de software livre no Brasil, com base em estudo realizado em 2005 pelo Softex , a seguir citaremos alguns casos de uso de software livre no Brasil:

- Grupo Pão de Açúcar: Irá utilizar o Linux no Pontos de Venda para executar o Sistema Itautec de Automação Comercial nas 500 lojas da rede, Pão de Açúcar, Extra e Compra Bem, somando 8.500 PDVs.
- Grupo Sonae: possui 5 mil PDVs distribuídos nos supermercados Big, Candia, Mercadorama, Nacional e Maxxi Atacado utiliza um piloto em 3 de suas 174 lojas em Linux. Todos os servidores de missão crítica utilizam o Linux como sistema operacional e equivalem a 10% do total de servidores do grupo.
- Petrobrás: utiliza o Linux em aplicações para análise de dados de prospecção. São mais de 1000 CPUs de variados fornecedores em cluster³² com Linux.
- GVT: trocou alguns servidores RISC por outros com a plataforma Intel-Linux. Tiveram, com isso, uma economia superior a 1 milhão de reais. Pretendem trocar todo o parque de servidores de Risc/Unix.
- Infoglobo: empresa responsável pelos jornais O Globo, Diário de São Paulo e Extra. Precisavam de agilidade, pois como seu produto é jornal e este

³¹ fonte: <http://www.softwarelivreparana.org.br/modules/news/article.php?storyid=1326>

tem um prazo exato a ser cumprido, qualquer atraso significa perda de dinheiro. Escolheram o Linux e tiveram um custo de 60% do previsto. A Infoglobo investiu na migração aproximadamente 200 mil dólares, contra uma previsão inicial de gastos da ordem de 650 mil dólares.

- Lojas Colombo: plataforma Linux implantada em toda planta operacional da rede. A migração levou 2 anos e aproveitou equipamentos que teriam que ser trocados, como micros com configuração Pentium 100 MHz, fator que foi decisivo na decisão da empresa. Todos os micros das lojas utilizam o Linux da Conectiva. O Linux roda em 3200 PDVs e 290 servidores. Estima-se economia de 80% em relação a uma solução com produtos da Microsoft.
- Extracta: empresa de biotecnologia situada no Rio de Janeiro. O software livre domina a área de tecnologia da empresa, que conta com todo o processamento de testes químicos e biológicos. O Windows sobrevive em apenas alguns poucos desktops. A economia estimada é de US\$160 mil/ano.
- Metrô SP: na análise da implementação de serviços de e-mail para os funcionários, os primeiros cálculos mostravam que o preço da licença do software de e-mail proprietário girava em torno de 100 dólares por usuário, valor que poderia cair para 30 dólares com a compra em grande quantidade. Mesmo assim, seria um investimento pesado, já que o Metrô tinha cerca de 3 mil usuários de correio eletrônico. Além disso, a implantação do Notes exigiria a instalação de vários servidores e a troca de praticamente todo o parque de micros da empresa. Este custo foi evitado com a utilização de uma solução livre. Além disso, com a utilização de pacotes livres de escritório a economia anual tem sido de R\$700 mil para o parque de 1.600 microcomputadores hoje existentes no Metrô

³² Um cluster de computadores, é formado por um conjunto de computadores, que utiliza-se de um serviço de sistema operacional classificado como sistema distribuído, onde vários computadores são ligados em rede e comunicam-se através do sistema de forma que trabalham como se fosse uma única máquina de porte maior.

Assim como no resto do mundo, podemos observar uma amostragem da utilização de soluções baseadas em software livre no Brasil, em empresas de diversos portes e setores da economia, demonstrando a abrangência da permeabilidade do fenômeno do software livre, como alternativa ao software proprietário.

5 CONCLUSÃO

Mostramos neste trabalho que historicamente o desenvolvimento de software surgiu utilizando o modelo semelhante ao utilizado no desenvolvimento de software livre, inicialmente o software era fornecido embutido no equipamento, onde estava o custo.

O software livre possibilita a inovação e a diferenciação, e a própria filosofia do modelo de desenvolvimento do software livre, permite que toda comunidade se beneficie e obtenha vantagens, uma vez que o ciclo de desenvolvimento contínuo, torna pública as melhorias realizadas. A comunidade de usuários e desenvolvedores, são em sua maioria voluntários e não sofrem pressões comuns no desenvolvimento de software comercial, como prazos para lançamento de novas versões, permitindo melhorar os níveis de teste e qualidade.

Outra análise que pode ser feita, com base na teoria de Porter, é o efeito das forças competitivas na indústria de software, onde a entrada do movimento do software livre afetou as estratégias tanto dos fornecedores de software proprietário como de fabricantes de hardware. As principais estratégias adotadas pelos oligopólios da indústria de TI, é a implantação de projetos usando o modelo de desenvolvimento open source, a aquisição de empresas desenvolvedoras de software open source, o patrocínio de projetos open source e o desenvolvimento de versões de software compatíveis com o sistema operacional linux, se inserindo e passando a disputar também o mercado de produtos complementares deste sistema operacional.

Citamos abaixo alguns exemplos de alterações ocorridas em grandes corporações da indústria de TI:

- A IBM adquiriu a empresa de software open source GlueCode e patrocina diversos projetos Open Source;
- A Oracle adquiriu a empresa Sleepcat e parte essencial do software de indexação do banco de dados MySQL, além de disponibilizar versões compatíveis com Linux;
- A Novell adquiriu a distribuição Linux SUSE.
- A distribuição Linux Red Hat, foi dividida em duas empresas, uma que passou a patrocinar o projeto Linux Fedora e outra com objetivos comerciais, que comercializa a versão corporativo do sistema operacional Red Hat Enterprise Edition e serviços de suporte. A Red Hat adquiriu a JBoss, empresa open source, que desenvolve servidor de aplicações Java.
- A Microsoft adotou um modelo de desenvolvimento semelhante ao modelo open source, para alguns produtos, denominado Shared Source Initiative. Outra estratégia adotada é a diversificação, passando a atuar em outros mercados, como fornecimento de conteúdo, serviços de mensagens instantâneas e jogos para computadores.

A utilização do sistema operacional Linux já está consolidado para servidores, sendo largamente utilizado para aplicação Web, banco de dados, servidor de correio, etc. Já no ambiente de estação de trabalho, onde se obteria maior vantagem, considerando o ganho de escala e custos de licenciamento, ainda existe resistência, embora alguns fabricantes de computadores já estejam vendendo equipamentos com sistema operacional Linux pré instalado.

O software livre agregado ao crescente e veloz desenvolvimento tecnológico, vem contribuindo para reduzir o aprisionamento tecnológico, viabilizando para as corporações a migração e utilização de soluções baseadas em software livre para os mais diversos fins, podendo vir a alterar o atual paradigma comercial, marcado por fortes componentes monopolistas. Um exemplo, são os fabricantes de computadores de arquitetura RISC, que tradicionalmente utilizam hardware e sistemas operacionais Unix proprietário, vêem no sistema operacional Linux como uma ameaça, uma vez que é um sistema operacional compatível com Unix, portátil em diversas arquiteturas de hardware, reduzindo efetivamente o aprisionamento neste segmento como alternativa de baixo custo, sendo que muitas empresas tem adotado Linux para execução de aplicações de missão crítica, utilizando hardware "genérico". Outro ponto é a redução do aprisionamento a um único fornecedor.

Com relação ao software livre no Brasil, existem poucos dados normalizados e compilados especificamente sobre software livre. Contudo recentemente foram realizados estudos patrocinados pelo Ministério da Ciência e Tecnologia visando incentivar a utilização de software livre pelo governo, bem como, avaliar a indústria de software no Brasil. Estes estudos mostram o volume do mercado de software do Brasil, que fica em torno de 1,5% do PIB, sendo que a indústria está basicamente voltada para o mercado interno, mostrando uma oportunidade de crescimento, considerando tanto o mercado interno, como o mercado externo, onde a participação da indústria brasileira ainda é irrelevante.

Podemos concluir que o software livre pode ajudar a reduzir a exclusão digital, reduzir a dependência de sistemas informatizados fornecido por companhias estrangeiras, além de proporcionar efeito sobre o comércio internacional, na medida

em que reduz o pagamento de licenças e royalties em razão da importação de software.

Abstraindo a conclusão para um contexto mais amplo, considerando a filosofia que contesta as formas de licenciamento de software, o acesso ao código fonte e coloca os direitos de propriedade intelectual em um segundo plano, priorizando o desenvolvimento tecnológico. Este movimento pode ser a origem de um novo paradigma da produção de inovações tecnológicas em todas as áreas do conhecimento, democratizando o acesso à informação e à evolução tecnológica.

REFERÊNCIAS

BACIC, Nicolas Michel . **O software livre como alternativa ao aprisionamento tecnológico imposto pelo software proprietário** . 2003. 135 f. TESE - Tecnologia da Informação , Unicamp SP, São Paulo, 2003.

BLINKENLIGHTS Archaeological Institute. **Pop Quiz: What was the first personal computer?** Disponível em: <<http://www.blinkenlights.com/pc.shtml>>. Acesso em: 20 set. 2005.

EUROPEAN eGOVERNMENT SERVICES (UE). **Documentation on Open Source Software (OSS)**. Disponível em: <<http://europa.eu.int/idabc/en/document/2623>>. Acesso em: 20 nov. 2005.

FERRAZ, Nelson Corrêa de Toledo. **Vantagens Estratégicas do Software Livre para o Ambiente Corporativo** . 2002. 106 f. TESE (Mestrado) - Master Business Information Systems, PUC SP, São Paulo, 2002.

FREE SOFTWARE FOUNDATION (EUA). **The Free Software Definition**. Disponível em: <<http://www.fsf.org/licensing/essays/free-sw.html>>. Acesso em: 20 ago. 2005.

FREE SOFTWARE FOUNDATION (EUA). **CATEGORIES of Free and Non-Free Software**. Disponível em: <<http://www.fsf.org/licensing/essays/categories.html>>. Acesso em: 20 set. 2005.

GATES III, William Henry. **AN OPEN LETTER TO HOBBYISTS**. General Partner, Micro-Soft. Disponível em: <<http://www.blinkenlights.com/classiccmp/gateswhine.html>>. Acesso em: 20 nov. 2005.

GNU PROJECT (EUA). **The Free Software Definition**. Disponível em: <<http://www.gnu.org/philosophy/free-sw.html>>. Acesso em: 20 ago. 2005.

GOLDMAN, Ron; GABRIEL, Richard. **Innovation Happens Elsewhere: Open Source as Business Strategy**. Santa Clara, CA: Morgan Kaufmann Publishers, 2004. p. 111-145.

Governo Eletrônico. **Guia Livre Referência de Migração para Software Livre do Governo Federal**. Disponível em: <http://www.governoeletronico.gov.br/governoeletronico/publicacao/down_anexo.ws?tmp.arquivo=E15_469GuiaLivre_v099.pdf>. Acesso em: 20 nov. 2005.

Hexsel, Roberto A. Curitiba. **Software Livre Propostas de Ações de Governo para incentivar o uso de Software Livre**. 2002. 53 p. Relatório Técnico RT-DINF04/2002. UFPR.

IDC (EUA). **Windows 2000 Versus Linux in Enterprise Computing:**An Assessment of Business value for Selected Workloads. Disponível em: <<http://www.microsoft.com/windows2000/docs/TCO.pdf>>. Acesso em: 20 nov. 2005.

Lerner, J. & Tirole, J. (2000). **The Simple economics of open source**. NBER Working Paper, no 7600. Cambridge, MA.

Lindman, Juho . **Effects of Open Source Software on the Business Patterns of Software Industry**. Masters Thesis, : Helsinki School of Economics., 2004. 90 p.

MICROSOFT CORPORATION (EUA). **Microsoft Shared Source Initiative**. Disponível em: <<http://www.microsoft.com/resources/sharedsource/Initiative/Initiative.msp>>. Acesso em: 20 nov. 2005.

Porter, Michael E. **Estratégia Competitiva: Técnicas para a análise de Indústrias e Concorrência**, Rio de Janeiro. Ed. Campus, 1997.

OPENOFFICE.ORG.BR. **Projeto Brasil para o Brasil**. Disponível em: <www.openoffice.org.br>. Acesso em: 20 out. 2005.

OPEN SOURCE INITIATIVE (EUA). **History of the OSI**. Disponível em: <<http://www.opensource.org/docs/history.php>>. Acesso em: 20 nov. 2005.

OPEN SOURCE INITIATIVE (EUA). **The Open Source Definition**. Disponível em: <<http://www.opensource.org/docs/definition.html>>. Acesso em: 20 ago. 2005.

RAYMOND, Eric S.. **The Cathedral and the Bazaar**. Disponível em: <<http://www.catb.org/~esr/writings/cathedral-bazaar/>>. Acesso em: 20 nov. 2005.

ROSELINO, José Eduardo, **A INDÚSTRIA DE SOFTWARE: o “modelo brasileiro” em perspectiva comparada**, Tese de Doutorado, Instituto de Economia, UNICAMP, Campinas, 2006.

Softex. Campinas. **O impacto do software livre e de código aberto na indústria de software do Brasil**: Softex, 2005. 76 p.

SHAPIRO, Carl, VARIAN, Hal. **A economia da informação**: Como os princípios econômicos se aplicam à era da Internet. Rio de Janeiro: Ed. Campus, 2003.

STALLMAN, Richard. **The GNU Manifesto**. Disponível em: <<http://www.gnu.org/gnu/manifesto.html>>. Acesso em: 20 nov. 2005.

TED SCHADLER (USA). **Your Open Source Strategy**. Cambridge, MA: Forrester Research, Inc., 2003. 24 p.

UFPR. **C3SL - Centro de Computação Científica e Software Livre**. Disponível em: < <http://www.c3sl.ufpr.br/pt-br/index.html/>>. Acesso em: 03 out. 2006.

Unicamp. **RAU-TU OpenOffice - Sistema Colaborativo de Perguntas e Respostas**. Disponível em: <<http://www.rau-tu.unicamp.br/openoffice/>>. Acesso em: 20 out. 2005.

UNIVERSIDADE FEDERAL DO PARANÁ. Sistema de Bibliotecas. **Normas para apresentação de documentos científicos**. Curitiba: Ed. Da UFPR, 2000. (Teses, dissertações, monografias e trabalhos acadêmicos, v. 2)

VELOSO, Francisco et al. **SLICING THE KNOWLEDGE-BASED ECONOMY IN BRAZIL, CHINA AND INDIA: A TALE OF 3 SOFTWARE INDUSTRIES**. Massachusetts - Usa: Massachusetts Institute Of Technology, 2003.

WHEELER, David A.. **Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!**. Disponível em: <http://www.dwheeler.com/oss_fs_why.html>. Acesso em: 20 nov. 2005.