



UNIVERSIDADE FEDERAL DA BAHIA
FACULDADE DE CIÊNCIAS ECONÔMICAS
CURSO DE GRADUAÇÃO EM CIÊNCIAS ECONÔMICAS

LEONARDO RODRIGO BEZERRA

**SOFTWARE DE CÓDIGO ABERTO: OPORTUNIDADES PARA O BRASIL A
PARTIR DA INOVAÇÃO ABERTA**

SALVADOR
2007

LEONARDO RODRIGO BEZERRA

**SOFTWARE DE CÓDIGO ABERTO: OPORTUNIDADES PARA O BRASIL A
PARTIR DA INOVAÇÃO ABERTA**

Trabalho de conclusão de curso apresentado no curso de graduação em Ciências Econômicas da Universidade Federal da Bahia como requisito parcial à obtenção do grau de Bacharel em Ciências Econômicas.

**Orientador: Prof. Dr. Ihering Guedes Alcoforado de
Carvalho**

**SALVADOR
2007**

LEONARDO RODRIGO BEZERRA

SOFTWARE DE CÓDIGO ABERTO: OPORTUNIDADES PARA O BRASIL A PARTIR DA INOVAÇÃO ABERTA

Aprovada em junho de 2007.

Orientador: _____
Ihering Guedes Alcoforado de Carvalho
Faculdade de Economia da UFBA

Bouزيد Izerrougene
Faculdade de Economia da UFBA

Lívio Andrade Wanderley
Faculdade de Economia da UFBA

AGRADECIMENTOS

Deixo aqui o registro da minha gratidão ao professor Dr. Ihering Guedes Alcoforado de Carvalho, pelo profissionalismo, acessibilidade e compromisso com o trabalho aqui realizado. Obrigado pela compreensão e ajuda que foram dadas ao longo dessa jornada.

À minha mãe, Maria José, pela dedicação, apoio, amor e incentivo. Sem essa mulher não seria possível a conclusão de mais uma etapa em minha vida. Agradeço pelos inúmeros esforços que me possibilitaram ser finalmente um economista.

A todos os meus familiares que direta ou indiretamente me auxiliaram em todos esses anos de vida universitária.

Ao grande amigo, Bruno Brandão, pela ajuda incondicional dada ao longo da faculdade. Pelas inúmeras revisões, idéias e incentivos ao longo do trabalho monográfico. À Fabrício Leite, amigo de longa data que sempre demonstrou seu companheirismo, apoio e simplicidade comigo.

À Faculdade de Ciências Econômicas em toda a sua extensão: funcionários, colegas e professores. Um especial agradecimento aos professores Plínio, Oswaldo Guerra e Lielson Coelho, que contribuíram e muito na minha formação acadêmica. Aos colegas de Economia da UFBA, em especial a Everton e Graça que sempre foram muito próximos e prestativos nas mais diversas situações.

Aos colegas da Engenharia do Banco do Brasil, especialmente ao Eng. Cleiton Costa que sempre foi atencioso e flexível às minhas solicitações durante o período em que estagie no banco. Ao amigo Eduardo Oliveira pelos conselhos, motivação e ajuda incondicional.

Enfim, o meu muito obrigado à todos que participaram e participam da minha vida pessoal e profissional, sem os quais não seria possível essa caminhada.

“Sim, não tenhamos pressa. Mas não percamos tempo”.

José Saramago

RESUMO

O subdesenvolvimento em grande medida está assentado no distanciamento tecnológico existente entre os países centrais e os países periféricos. Desta maneira, o presente estudo delinea uma oportunidade de redução do *gap* tecnológico, a partir da antecipação da adoção tecnológica de uma nova realidade que se apresenta: o *software* de código aberto. O desenvolvimento tecnológico cria “janelas de oportunidades” para os países, contribuindo para a elevação dos níveis de desenvolvimento. Para tanto, é imperativo discutir os diversos aspectos que envolvem o *software*, pontuando suas particularidades e indicando características desta realidade, em especial, as associadas ao *software* de código aberto. O trabalho trata da postura adotada pelo governo brasileiro diante do *software* livre e estabelece melhores estratégias de atuação nesse sentido. Este estudo proporá medidas que possibilitem que as firmas brasileiras consigam uma nova inserção internacional, apostando em um novo padrão de apropriação intelectual diferente do modelo vigente, ao mesmo tempo que incita aprofundar as investigações acerca das possibilidades em latência no âmbito das inovações abertas – como o ponto de ancoragem da pesquisa e desenvolvimento no séc. XXI.

Palavras-chave: Inovação. *Software*. *Gap*. *OSS*. Desenvolvimento Tecnológico e Econômico. Marco Institucional. Transversalidade. Inovação Aberta.

SUMÁRIO

1	INTRODUÇÃO	9
2	CARACTERIZAÇÃO DO SOFTWARE	11
2.1	UMA BREVE DISCUSSÃO SOBRE SOFTWARE.....	11
2.1.1	O processo de pesquisa e suas implicações ao setor de software	15
2.2	CARACTERIZANDO O OSS – OPENSOURCE SOFTWARE (SOFTWARE DE CÓDIGO ABERTO).....	17
2.2.1	Aspectos Institucionais	17
2.2.2	Aspectos Organizacionais	22
2.2.3	Aspectos Culturais	23
2.2.4	Aspectos Econômicos	26
2.2.5	Modelo de Negócios	30
2.2.5.1	A produção (Oferta).....	30
2.2.5.2	O mercado (Demanda).....	33
2.2.5.3	Os novos entrantes.....	35
3	POLÍTICA BRASILEIRA DE INCENTIVO E ADOÇÃO DE SOFTWARE LIVRE	37
3.1	MOVIMENTO EM PROL DO SOFTWARE LIVRE NAS INSTITUIÇÕES PÚBLICAS.....	38
3.2	OS FUNDAMENTOS ECONÔMICOS DAS INOVAÇÕES FOCADAS NO USUÁRIO.....	40
3.3	MARCO INSTITUCIONAL.....	45
3.3.1	Direitos Autorais e Legislação Brasileira	45
3.3.2	Licenças	48

3.4	UMA AVALIAÇÃO CRÍTICA DA POLÍTICA ESTRATÉGICA DO SETOR.....	50
4	CONCLUSÕES.....	56
	REFERÊNCIAS.....	58

1 INTRODUÇÃO

O presente estudo visa estabelecer uma “janela de oportunidades” baseada na adoção do *software*¹ de código aberto, possibilitando a países como o Brasil uma redução do *gap* tecnológico existente em relação aos países centrais. Diante da postura do Governo Brasileiro explicitada nas *Diretrizes de Política Industrial, Tecnológica e de Comércio Exterior*, pressupomos que o fomento ao uso e a produção de *software* se apresenta como oportunidade de redução do distanciamento tecnológico existente entre o Brasil e os países ditos desenvolvidos.

É a partir desse pressuposto que estendemos a análise e apontamos que não basta fomentar a atividade de *software*, mas que as políticas públicas devem contemplar novos vetores nessa área. Diante desse entendimento, o trabalho aqui apresentado concorda com a relevância do *software*, mas vai além, apontando que o *software* de código aberto mostra-se uma melhor oportunidade em relação ao modelo convencional.

Políticas públicas têm sido discutidas com o intuito de difundir a adoção de *software* livre, no entanto, sem estabelecer as relações com o código aberto. Existe uma diferença técnica entre *software* livre e de código aberto, entretanto, o mais relevante para nosso estudo diz respeito a inserção não só na esfera de consumo de *software*, mas principalmente na esfera produtiva. Entendemos que modelos como o *software* de código aberto poderão potencializar a difusão do uso do *software* e possibilitarão uma melhor inserção na perspectiva produtiva, constituindo-se, portanto, numa aposta acertada na aproximação tecnológica.

Como forma de abordagem, optamos por fazer um tratamento qualitativo do objeto de estudo – o *software* de código aberto – delineando algumas considerações sobre *software* em geral, passando a seguir para uma análise mais centrada no OSS e, em seguida, finalizando com uma discussão sobre as políticas públicas praticadas pelo Governo Federal. Nosso recorte assume o viés da Escola Institucional, focando nossos esforços em questões e mecanismo institucionais como forma de compreensão do tema abordado.

¹ O conceito de *software* será bastante discutido ao longo do trabalho. No momento, o conceito pode ser o mais difundido pelo senso comum.

Apontamos o surgimento do *software* de código aberto (OSS) como vetor tecnológico por compreendermos que este é um exemplo emblemático da inovação aberta – grande aposta dos setores de alta tecnologia e de grandes aportes intelectuais.

Na parte de referência teórica, abordamos os aspectos institucionais, organizacionais, econômicos, culturais e o modelo de negócio. Este último a partir das duas esferas: a oferta e a demanda.

No ponto seguinte, trataremos dos esforços do Governo Brasileiro em incentivar a adoção do *software* livre, discutindo os interesses que estão por trás dessa postura, além de analisarmos as ferramentas e os marcos legais que institucionalizam a produção de *software*.

Por fim, traçaremos uma análise crítica acerca das iniciativas tomadas, delineando oportunidades a serem exploradas e discutindo mecanismos de aceleração desse movimento.

A análise aqui apresentada não tem pretensões em estabelecer pontos conclusivos em relação a adoção do modelo de código aberto em relação ao modelo tradicional, nem objetiva demonstrar através de dados estatísticos os ganhos diretos que tal aposta possa gerar. Em última análise, estabelecemos um contraponto ao modelo convencional de *software* e alertamos para a importância de novos movimentos nessa área, em especial, na forma da apropriação do valor.

2 CARACTERIZAÇÃO DO SOFTWARE

2.1 UMA BREVE DISCUSSÃO SOBRE SOFTWARE

Segundo Roselino (2006, p.7), “o *software* é, antes de tudo, um não-objeto, uma não-coisa, que pelas suas propriedades satisfaz necessidades humanas de qualquer espécie”. Sendo assim o *software* é “uma mercadoria particular caracterizada pela sua natureza não-material. A sua função de produção não envolve o emprego de matérias-primas consumíveis ao longo de seu ciclo produtivo”. Esta atividade é caracterizada pelo intenso uso de força de trabalho.

Essa definição de *software* pode ser entendida como uma caracterização a partir da ótica constitutiva. É interessante observar que o *software* consegue reunir características de diversos outros bens produzidos no capitalismo, mas ao mesmo tempo essa multiplicidade demanda que façamos um recorte mais adequado a fim de facilitar o seu estudo.

Para nossas pretensões, que não dizem respeito a um estudo técnico do *software* e sim uma análise econômica dos efeitos da adoção do *software* de código aberto em detrimento total ou parcial do modelo proprietário, nos prenderemos a perspectiva de categorizar o *software* de acordo com a forma de aquisição, o pagamento de licenças de uso e a possibilidade de acesso ao código-fonte.

O código-fonte são instruções em linguagem de programação que respondem pelo *software* em si. São essas instruções dadas pelo programador que definirão as funcionalidades do *software*. Uma vez que o usuário tenha acesso a essas instruções e, concomitantemente, possua conhecimentos técnicos para tanto, poderá realizar modificações, melhorias e mudanças no *software*. Dessa maneira, de antemão informamos que o *software* proprietário não permite que o usuário realize qualquer tipo de acesso ao código-fonte, impossibilitando assim quaisquer tipos de modificações.

Para ilustrar nossa discussão, faremos uso do diagrama apresentada na Figura 1.

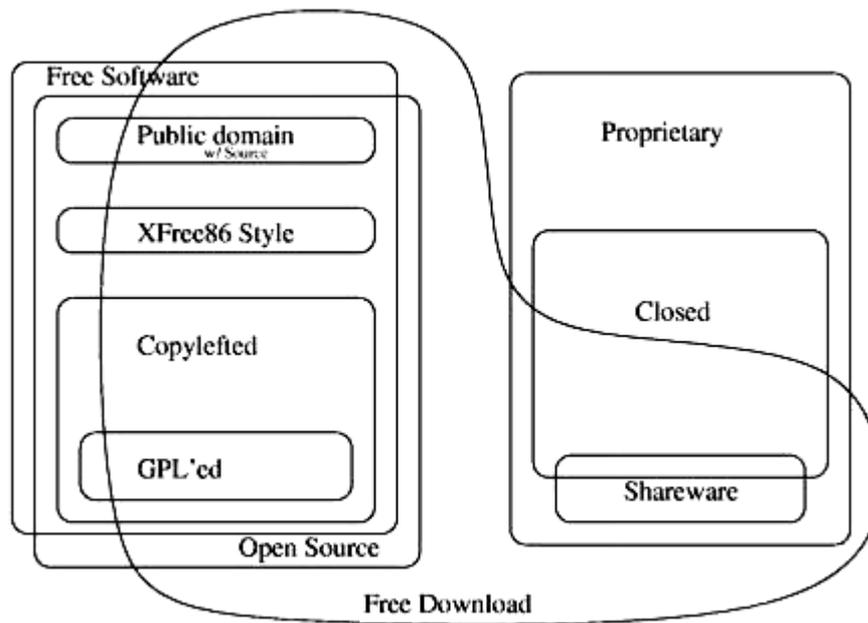


Figura 1 – Diagrama de Chao-Kuei

Disponível em: <http://www.gnu.org/philosophy/categories.html>.

Como podemos observar, em um dos lados temos o *software* proprietário e do outro o *software* livre. No caso do *software* proprietário, duas modalidades são visíveis, sendo que elas podem migrar em determinado sentido.

O *software shareware* está contido no *software* proprietário e configura uma flexibilização estratégica do direito de propriedade. A sua aquisição não depende de qualquer pagamento (normalmente pode ser encontrado em páginas da Internet ou em mídias promocionais), mas tem prazo de utilização e/ou limitações de módulos do sistema e, em nenhuma fase, permite acesso ao código-fonte. Nessa distribuição de *software*, o usuário tem uma espécie de “amostra grátis” do *software*, podendo fazer uso por um determinado tempo a fim de optar pela aquisição, ou não, do *software* com as funcionalidades completas. A migração para o *software* completo poderá ser realizada através de uma chave de ativação do produto ou mesmo através de alguma forma de intervenção por parte do fabricante do *software*.

No *software* proprietário sua “[...] cópia, redistribuição ou modificação são em alguma medida proibidos pelo seu proprietário. Para usar, copiar ou redistribuir, deve-se solicitar permissão ao proprietário, ou pagar para poder fazê-lo” (HEXSEL, 2002, p.6). Nesse modelo de negócio o foco é a comercialização do *software* como um produto, podendo ou não, ter serviços associados – como atualizações, suporte técnico, etc. A cópia e a redistribuição deste

tipo de *software* incide na violação das leis de direito de propriedade intelectual, ou como popularmente é conhecida: a pirataria². Atualmente, ainda observamos uma grande presença dessa modalidade de *software* nos ambientes pessoais e empresariais. Os sistemas operacionais e as *suites* de programas de edição de texto, planilhas de cálculo, criação de apresentações e afins, ainda são dominados por empresas como a Microsoft Corporation³, apesar de já verificarmos alternativas reais a esse monopólio.

Em contrapartida ao modelo proprietário, observamos uma gama de configurações *free*, entretanto iremos nos ater a diferenciar o *software* livre do *software* de código aberto⁴. Como o diagrama nos mostra, eles estão intimamente relacionados, mas não são a mesma coisa.

O *software* livre é “[...] o *software* disponível com a permissão para qualquer um usá-lo, copiá-lo, e distribuí-lo, seja na sua forma original ou com modificações, seja gratuitamente ou com custo” (Ibid., p.4). Nesta configuração, o usuário não paga pela utilização do *software* e a sua cópia ou redistribuição pode ser feita sem qualquer tipo de infração às leis que asseguram os direitos de propriedade intelectual. Estes *softwares*, via de regra, se valem de ferramentas como a Internet para facilitar a sua distribuição aos usuários.

Já o OSS – *Open Source Software Software*, de agora em diante chamado *software* de código aberto – tem quase as mesmas características do *software* livre, no entanto, a principal e mais importante diferença diz respeito ao acesso ao código-fonte. No caso do *software* livre, ele propicia o uso sem ônus financeiro ao usuário, mas não permite o acesso ao seu código-fonte, impossibilitando os indivíduos devidamente capacitados a realizar qualquer tipo de modificação ou melhoria na sua programação. O *software* de código aberto, por sua vez, permite o acesso por parte do usuário ao código-fonte, diante de algumas condições particulares ao projeto em questão, possibilitando um melhor ajustamento do *software* às

² A pirataria de software ocorre quando alguém vende, copia ou distribui um software proprietário sem a autorização do indivíduo ou firma que detém sua propriedade intelectual. A pirataria difere da falsificação, pois na primeira o agente objetiva apenas usar o produto sem pagar por isso, enquanto que na segunda o falsificador tenta reproduzir um produto original se valendo inclusive do impacto da marca. Alguém que copia uma mídia que contém um software e instala em seu computador, está incorrendo em pirataria. Já alguém que reproduz um software - reproduzindo inclusive a embalagem, documentação e etc -, visando a venda como um produto original, incide em falsificação.

³ A Microsoft Corporation é a maior e mais conhecida empresa de *software* do mundo. Foi fundada em 1975 por Bill Gates e Paul Allen com o objetivo de desenvolver e comercializar interpretadores da linguagem BASIC. (Wikipédia – disponível em: <http://pt.wikipedia.org/wiki/Microsoft>, acessado em 14/05/2007).

utilizações almejadas por esse usuário. Dentro desta perspectiva, uma regra bastante difundida e vigente para o *software* de código aberto, é que as modificações e/ou melhorias sejam novamente disponibilizadas a comunidade responsável pela criação e gestão deste *software*.

O *software* de código aberto se apresenta como uma alternativa ao *software* proprietário, tanto do ponto de vista do programador, quanto da ótica do usuário, possibilitando aos envolvidos uma mudança significativa na relação com o conhecimento institucionalizado no *software*, em especial nos meios de apropriação dos direitos de propriedade intelectual. Isso significa dizer, que dentro desta configuração, o *software* deixa de ter uma característica essencialmente de produto, passível de comercialização, e torna-se, em última análise, um modelo de relação de trabalho em um segmento cada vez mais amplo da sociedade com o universo do conhecimento aludido.

De fato, nos primórdios dos *softwares*, estes possuíam seus códigos-fonte abertos a todos os usuários, que nesta fase se restringiam às comunidades de programadores. O movimento de “fechamento” do código ocorreu quando as firmas vislumbraram a possibilidade de tornar o *software* amigável a usuários não-programadores, tornando-o um produto comercial passível de venda como qualquer outro produto. Ancorado nesse *insight* que Bill Gates criou a Microsoft. Vale ressaltar, que essa “volta às origens” teve no sucesso da Microsoft e da Internet sua principal motivação. A Internet possibilitou a formação de redes e/ou comunidades desenvolvedoras de *software* que fisicamente estão dispersos ao longo do globo, mas que virtualmente se reúnem com propósitos comuns. Estes programadores passaram a se mobilizar não só através do desenvolvimento de *software* proprietário, tendo como motivação o salário, mas, também, em projetos de *software* de código aberto, através de relações de prestação de serviços na customização de sistemas.

O *software* de código aberto não é uma alternativa marginal ao modelo proprietário, é sim a ponta do *iceberg* de uma mudança no capitalismo na forma de inovar: a inovação aberta. Dentro da *open innovation*⁵ – doravante, inovação aberta –, os próprios indivíduos responsáveis pelo processo de inovação, seja ele em produto, processo ou em caráter

⁴ As outras distribuições de software – Public domain, Xfree86 Style, Copylefted, GPL’ed – possuem pontos de proximidade com a distribuição “free” e também pequenas particularidades. A abordagem dessas distribuições será feita ao longo do trabalho de acordo com a sua relevância para os objetivos estabelecidos para esse estudo.

⁵ A *open innovation* ou inovação aberta será melhor abordada nos “*FUNDAMENTOS ECONÔMICOS DAS INOVAÇÕES FOCADAS NO USUÁRIO*”, constante nesse trabalho.

organizacional, são os agentes utilizadores destas inovações. Sendo assim, verificamos um processo ainda insipiente em determinada análise, mas que questiona fortemente a forma de apropriação dos ganhos relativos aos trabalhos de natureza intelectual.

Uma vez que diferenciamos os *softwares* a partir da ótica da aquisição, da necessidade de pagamentos pelo uso e pela acessibilidade ao código-fonte, na seção seguinte observaremos o *software* sob um leque de outras abordagens.

2.1.1 O processo de pesquisa e suas implicações ao setor de software

A caracterização do *software* não estaria completa sem uma discussão no tocante ao seu processo de inovação. É extremamente relevante situarmos o *software* diante da perspectiva da inovação e, para tanto, iremos nos valer, inicialmente, de uma representação de um conjunto de modelos tal como sugerida nos quadrantes apresentados por Stokes (2005) disponível na Figura 2.

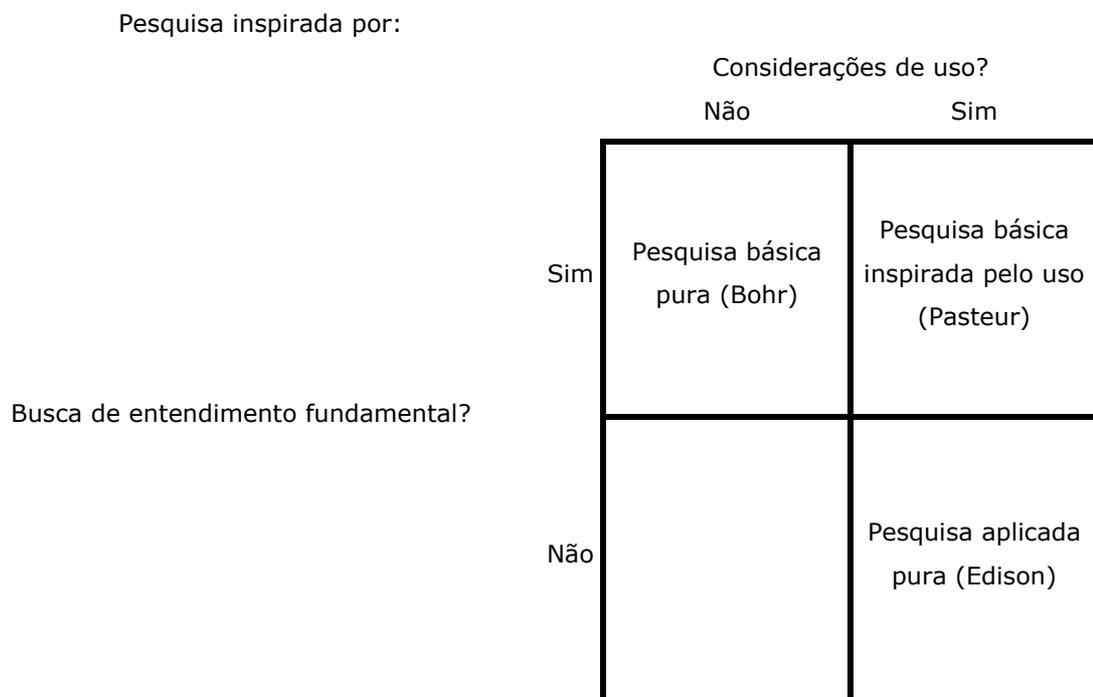


Figura 2 – Modelo de quadrantes da pesquisa científica

Fonte: STOKES, 2005, p. 118.

Nesta representação observamos que duas perguntas são feitas. A primeira é se a pesquisa é inspirada por considerações de uso e a segunda pontua se o objetivo é a busca por

entendimento fundamental. Na verdade, sempre se achou que estas coisas fossem excludentes, ou seja, que uma pesquisa direcionada à busca de entendimento fundamental não poderia ter considerações de uso. A proposta apresentada por Stokes, nesta formulação, mostra que há coerência em pesquisas que contemplem as duas coisas. A depender de como esses dois objetivos se combinem, podemos caracterizar a pesquisa de forma diferenciada.

O Quadrante de Bohr – que está acima e a esquerda (na Figura 2) – responde pela pesquisa básica que tem como proposta a busca do conhecimento sem o comprometimento de levantar utilizações práticas. Essa condição seria o idealizado pelos adoradores da ciência pura e que não buscam conexões com as questões de uso.

Já o Quadrante de Pasteur, localizado na parte superior direita (Figura 2), assinala a pesquisa básica que busca compreender as fronteiras do entendimento, sem abrir mão das considerações de uso. É neste quadrante onde está localizada a “pesquisa estratégica”.

O quadrante à esquerda e abaixo não está vazio. Contempla àqueles que pesquisam fenômenos específicos sem a preocupação de explicações gerais nem qualquer utilização prática à qual se destinem seus resultados.

Partindo desta representação apresentada por Stokes (2005, p.118), o setor de *software* estaria contemplado no Quadrante de Edison – parte inferior direita da Figura 2 – e tem como base a “pesquisa guiada exclusivamente por objetivos aplicados, sem procurar um entendimento mais geral dos fenômenos de um campo da ciência. [...] Grande parte da pesquisa moderna pertencente a essa categoria é extremamente sofisticada, embora dirigida de maneira estreita a objetivos aplicados imediatos”. O fato de não objetivar o entendimento de determinado fenômeno sob a ótica da ciência não diminui sua importância, mas, pelo contrário, indica uma característica importante que norteará posicionamentos dos participantes deste setor. Daí, inferirmos que “somente numa estrutura que trate *ex ante* dos objetivos da pesquisa pode prover as necessidades da política científica e tecnológica” (Ibid., p.125).

Em *software*, as inovações ocorrem fundamentalmente na concepção do produto, uma vez que o processo – basicamente o uso de linguagens de programação – não se modifica com intensidade significativa. Essa forte característica de *design* explica a necessidade de mão-de-obra e os altos custos de concepção. Nesse segmento, o principal fator de produção é a

capacidade intelectual humana responsável pela articulação dos conhecimentos de programação focados, ora na criação de *interfaces* mais amigáveis aos usuários, ora na *performance* técnica do sistema, mas de agrado dos programadores.

Como já mencionado, o *software* envolve altos custos de concepção, *design* e posteriormente *marketing*, mas os seus custos de reprodução – custo marginal de mais uma unidade produzida – tendem a zero. Daí a fragilidade da proteção oferecida pelos direitos de propriedade convencionais que, segundo os defensores do modelo de *software* fechado, sem essa garantia instaura-se uma ameaça aos mecanismos de incentivos do negócio. Mas vale ressaltar que a ameaça maior vem do modelo de código aberto e das exigências do usuário-programador que lentamente vai substituindo o usuário “analfabeto digital”. Com a massificação da Internet, qualquer fabricante de *software*, mesmo os mais modestos, pode disponibilizar não só os seus produtos, mas principalmente os seus serviços aos usuários – desde os mais sofisticados aos menos exigentes, em relação a *performance* e complexidade – sem precisar com isso incorrer em custos de transporte, seguros, impostos de importação, etc.

É da política de incentivo ao OSS que trata esse trabalho, donde passamos a caracterizar o OSS para depois tratarmos da política brasileira pertinente.

2.2 CARACTERIZANDO O OSS – OPENSOURCE SOFTWARE (SOFTWARE DE CÓDIGO ABERTO)

2.2.1 Aspectos Institucionais

O correto delineamento das políticas de incentivo ao OSS não pode ocorrer sem o detalhamento de suas características estruturais e, dessa forma, um esforço de caracterização a partir de diferentes prismas é de extrema importância. O primeiro aspecto que observaremos diz respeito aos aspectos institucionais que balizam o comportamento dos agentes envolvidos neste processo.

Inicialmente, devemos nos valer da teoria econômica para entendermos um conceito bastante importante: a Instituição. Para Douglas C. North (1991), podemos compreender as instituições como:

[...] mecanismos de coação humanos que estruturam as interações políticas, econômicas e sociais. Elas consistem de coação informal (sanções, tabus, tradições e códigos de conduta) e, ao mesmo tempo, de regras formais (constituições, leis, direitos de propriedade. [...] Instituições dão conta da estrutura de incentivos de uma economia, bem como do seu desenvolvimento, direcionando as mudanças econômicas rumo ao crescimento, estagnação ou declínio. (NORTH, 1991, p.97)⁶.

De maneira sintética podemos compreender as instituições como as regras do jogo, ou seja, os mecanismos que balizam a conduta dos indivíduos nos campos econômicos, pessoais, políticos, entre outros. Daí, na sociedade contemporânea, a propriedade privada emerge como uma importante instituição, chave na discussão da produção capitalista. Esta forma de instituição é um dos pilares fundamentais do capitalismo moderno dado ser a condição *sine qua non* do mercado e tem no Estado seu principal guardião. É através da garantia da propriedade sobre os bens físicos/tangíveis e/ou os bens intelectuais/intangíveis que os agentes que detêm as relações de troca e, por conseguinte, de apropriação, realizam suas operações. Diante desta perspectiva, vislumbramos que uma mudança como a proposta pelo OSS só se torna viável a partir do entendimento e da mudança de aspectos institucionais vinculados a propriedade. Isso acontece, pois o OSS questiona a via convencional de apropriação dos ganhos sobre a propriedade intelectual daqueles que desenvolvem *softwares*.

Williamson, importante representante da **Nova Economia Institucional**, nos fornece uma outra apresentação da instituição, formulada em quatro níveis de análise social, representada na Figura 3.

⁶ “Institutions are the humanly devised constraints that structure political, economic and social interaction. They consist of both informal constraints (sanctions, taboos, traditions, and codes of conduct), and formal rules (constitutions, laws, property rights). [...] Institutions provide the incentive structure of an economy; as that structure evolves, it shapes the direction of economic change towards growth, stagnation, or decline.” (NORTH, 1991, p.97)

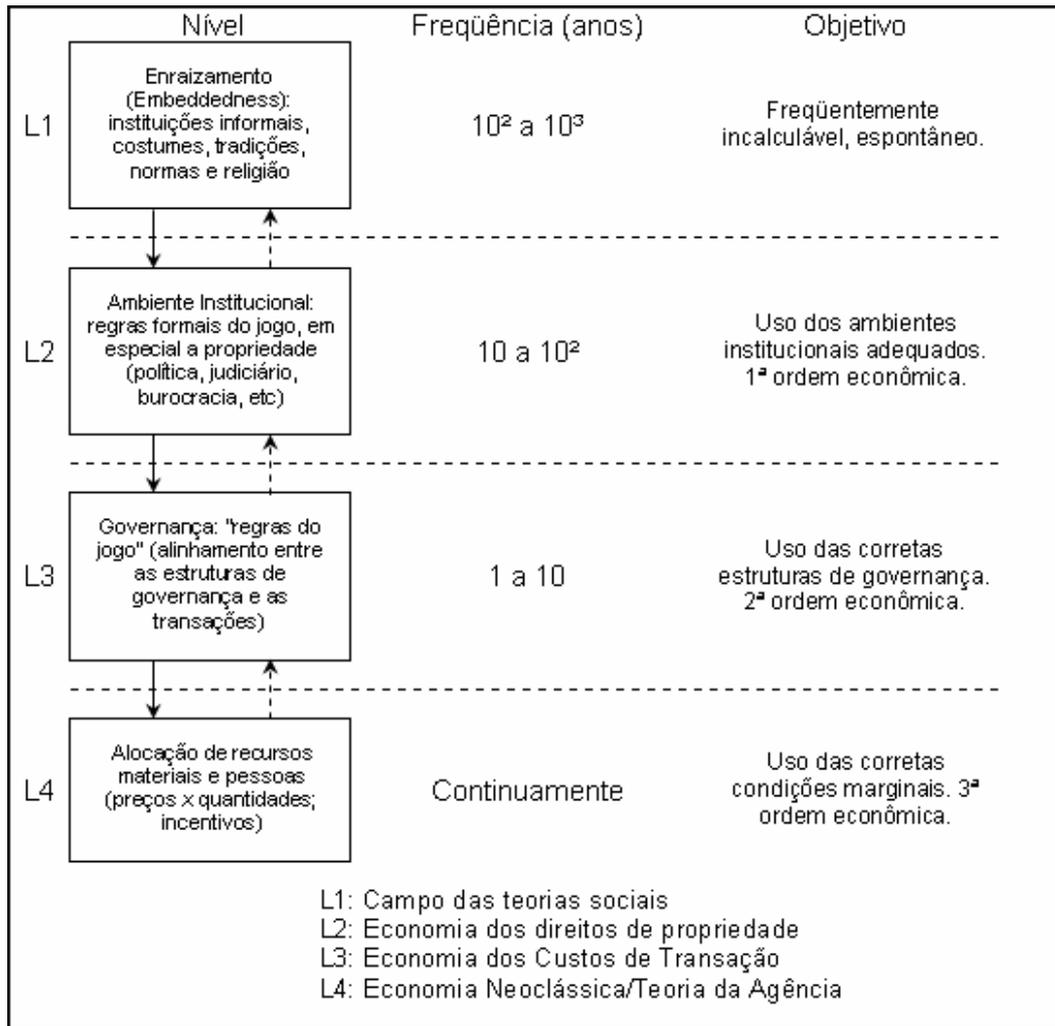


Figura 3 – Níveis Institucionais

Adaptação Própria

Fonte: WILLIAMSON, 2000.

Nessa tipologia apresentada por Williamson, observamos uma separação em quatro níveis. Temos o primeiro nível na parte superior, decrescendo até o nível mais baixo – L4. Estes níveis estão ligados por setas sólidas que sinalizam que os níveis superiores têm poder de influenciar os níveis mais baixos. Por conseguinte, as setas pontilhadas, que ligam os níveis mais baixos aos mais altos, nos informam que os níveis inferiores dão *feedback*⁷ aos níveis superiores. Observamos a existência de quatro níveis e esses níveis possuem frequência e propostas diferenciadas.

É no primeiro nível, chamado enraizamento (*embeddedness*), onde estão localizadas as instituições informais, os costumes, as tradições, as normas e a religião. Nesse patamar, os

⁷ Tradução: retorno ou retroalimentação

eventos ocorrem em espaços de tempo bastante longos, algo em torno de séculos ou milênios. Este nível apresenta uma dificuldade de acompanhamento e as coisas acontecem de maneira espontânea. É neste nível que as teorias da sociedade serão construídas.

No segundo nível, temos um ambiente institucional onde figuram as “regras formais do jogo”, em especial onde a propriedade privada está inserida. Neste nível estão as formas de particionamento do poder com base no federalismo: executivo, legislativo e judiciário. É também nesta esfera que se encontra a burocracia. O mais relevante para nosso propósito é que neste nível está a propriedade privada, uma importante instituição do mundo capitalista moderno, mas que sofre o impacto de propostas como o OSS.

Segundo Williamson (2000, p.598), nesta esfera de instituições “[...] a maioria das mudanças nas regras do jogo ocorrem em questão de décadas ou séculos”⁸. Como o próprio autor assinala a União Européia seria um bom exemplo do resultado de cinquenta anos de mudanças e negociações que ainda estão em andamento neste momento.

No que diz respeito a propriedade privada, “um sistema baseado em empreendimentos privados não pode funcionar perfeitamente sem que os direitos de propriedade sejam criados em relação aos recursos e, quando isto ocorre, alguém que queira fazer uso desse recurso deverá pagar ao proprietário”⁹. Nestas condições, cabe ao sistema de poder garantir aos agentes que este direito seja respeitado e que existam as perfeitas condições para que essas operações entre os agentes sejam realizadas com base nas “regras do jogo”. Normalmente essas regras são formalizadas em contratos, onde os agentes tentam se antecipar às possibilidades e, conseqüentemente, tentam minimizar as condutas oportunistas. Esta premissa tem sido a base do sistema capitalista até então, mas começa a ser atacada por condições diferenciadas como o OSS que, em última instância, é apenas a ponta do *iceberg* de algo muito mais amplo, mas que este trabalho não tem a pretensão de abordar em sua plenitude: a inovação aberta.

⁸ “[...] major changes in the rules of the game occur on the order of decades or centuries” (WILLIAMSON, 2000, p.598).

⁹ “[...] a private-enterprise system cannot function propoerly unless property rights are created in resources, and, when this is done, someone wishing to use a resource hás to pay owner to obtain it” (COASE *apud* WILLIAMSON, 2000, p. 598).

Num capitalismo essencialmente baseado na informação, os direitos de propriedade intelectual têm sido o centro da discussão. As nações têm tentado criar mecanismo de defesa e aprisionamento desta produção intelectual. No caso do *software*, observamos uma acirrada briga contra a pirataria ou outras formas de burlar os mecanismos estabelecidos, por se entender que os usuários que almejam obter e/ou fazer uso desses *softwares*, necessariamente devem pagar alguma espécie de dividendo aos seus autores intelectuais. Esses pagamentos normalmente são realizados na forma de licenças de uso ou patentes. Dentro desta perspectiva, observamos um duplo processo de polarização onde: de um lado, as nações produtoras de *software* criam um fluxo unilateral de comércio em que os países em desenvolvimento atuam apenas como mercados consumidores e, de outro, o *software* proprietário é ameaçado pelo código aberto e o *software* livre. O OSS rejeita essa forma de apropriação intelectual que transformou o produto *software* em algo passível de ser comercializado e aposta em outra frente de atuação onde o *software* se torna livre e o que surge é um mercado basicamente centrado no serviço.

Diante do exposto, é inquestionável a necessidade de mudança nesta esfera institucional de maneira a viabilizar a atividade de *software* de código aberto. As esferas de poder, tanto as situadas no Estado, como nas redes privadas, devem estar em sintonia com as necessidades de mudanças institucionais e devem possibilitar aos agentes novos instrumentos mais adequados aos seus fins. Por exemplo, legislações específicas que tratem da questão do OSS deverão ser criadas para que a atividade possa transcorrer com a observância do Estado, tendo a devida regulamentação. Em setores de alta tecnologia observamos uma real possibilidade de políticas regulatórias por parte do Estado em complemento à mecanismos de auto-regulação, característicos destes setores. Um interessante movimento que pode surgir desta esfera é a criação de mecanismo de aproximação entre as ICTs (Instituições de Ciência e Tecnologia) e os agentes privados.

O nível seguinte (L3) é caracterizado pela governança. Está assentado em mecanismos de maior frequência – de um a dez anos – e opera conectado ao nível superior L2, por responder pela gestão das “regras do jogo” existentes neste nível. A governança é “uma tentativa de estabelecer regras que tendem a amenizar os conflitos e a proporcionar ganhos mútuos aos agentes”¹⁰ em uma transação. As transações têm na sua constituição o conflito, a necessidade

¹⁰ “[...] is an effort to craft order, thereby to mitigate conflict and realize mutual gains” (WILLIAMSON, 2000. p.599).

de reciprocidade e o estabelecimento de regras. Um importante mecanismo de governança são os contratos que têm a pretensão de alinhar os interesses da estrutura de governança com as transações. É nesta direção que surge uma intensa demanda no âmbito do *software* de código aberto.

O quarto nível institucional – que não atende aos propósitos deste trabalho – é onde a análise neoclássica se impõe devido a proposta verificada de “aparatos de otimização, onde freqüentemente a análise marginal é empregada e a firma, dentro desta proposta, é comumente descrita como uma função de produção. Ajustamentos em preços e saídas ocorrem com mais ou menos continuidade”¹¹. É neste nível que os agentes se preocupam com questões alocativas e a firma aparece como uma função de produção que deverá ser maximizada.

2.2.2 Aspectos Organizacionais

Inicialmente, precisamos sinalizar a possível confusão que ocorre no entendimento dos conceitos de instituição e organização. O conceito de instituição já foi abordado no tópico anterior, mas ainda assim para contrapor a idéia de organização, iremos nos valer do conceito cunhado por Douglas North (*apud* CASTRO, 2004, p. 459) que afirma que “as instituições são o conjunto das regras do jogo, mas as regras devem ser distinguidas dos jogadores”. Por conseguinte, “as organizações são jogadores que, por meio de sua capacidade, suas estratégias e sua coordenação, também exercem influência na criação ou na evolução das próprias regras do jogo, ou, em outras palavras, nas instituições já existentes e/ou novas”. Por este raciocínio, as organizações são os jogadores que atuam de acordo com as regras – a instituição. No entanto, as organizações também são instituições quando entendemos que as primeiras também possuem jogadores e regras internas, conseqüentemente, se configurando como um caso especial de instituição.

Quando ouvimos a palavra organização provavelmente o primeiro pensamento que nos ocorre é o de hierarquia. Isso ocorre, porque imaginamos que a divisão do trabalho, por si só, demanda uma hierarquização dos níveis de atribuição e poder entre os indivíduos. De maneira quase que natural a hierarquia está associada à organização, no entanto, ela não ocorre de

¹¹ “Optimality apparatus, often marginal analysis, is employed, and the firm, for these purposes, is typically described as a production function. Adjustments to prices and output occur more or less continuously” (WILLIAMSON, 2000. p.600).

maneira uniforme nas diversas organizações. Existem organizações que pela própria natureza das suas atividades se estruturam em diversas camadas hierárquicas. Por outro lado, há organizações em que os níveis hierárquicos são bastante sutis e que as diferenças entre o topo e a base da pirâmide são difíceis de serem observadas.

No caso do OSS, a organização assume carácter horizontal onde vislumbramos “uma rede dinâmica e planeada estrategicamente de unidades auto-programadas e auto-dirigidas baseadas na descentralização, participação e coordenação” (BARAÑANO, 2004, p. 69). Em estruturas de rede, os indivíduos não estão sujeitos a níveis hierárquicos fortemente definidos, mesmo que estejam inseridos numa organização vertical. As relações que se estabelecem entre os participantes pressupõem descentralização, participação dos membros e uma coordenação apurada que são ingredientes fundamentais ao sucesso destes projetos.

Todas as atividades produtivas em que o intelecto é fortemente utilizado não se encaixam em estruturas rigidamente definidas. A mobilidade e a flexibilidade de participação dos envolvidos são peças importantes nessa configuração produtiva.

2.2.3 Aspectos Culturais

Os aspectos culturais podem ser considerados como parte das instituições, em especial, como assinalado por Williamson, no primeiro nível institucional. De forma que, a despeito do seu surgimento recente a compreensão do *software* de código aberto exige um entendimento de componentes culturais que influenciarão o sucesso e a difusão deste tipo de *software* entre os agentes econômicos.

No começo da programação dos *softwares* – nos anos 60 e 70 do séc XX –, estes eram desenvolvidos por cientistas e engenheiros pertencentes ao ambiente acadêmico ou em laboratórios corporativos, não tendo na sua origem o domínio do modelo comercial vigente. Fazia parte da cultura de desenvolvimento destes programas que estes fossem distribuídos gratuitamente para que outros profissionais pudessem modificá-los e aperfeiçoá-los. Este princípio comunal era a base da prática *hacker* – termo que se refere a talentosos e dedicados programadores que, mais tarde, assumiu um carácter pejorativo.

A cultura *hacker* incita os participantes de projetos OSS a contribuírem sem a perspectiva convencional de ganhos monetários, no curto prazo, através de pagamentos pelo uso de patentes e/ou da concessão de licenças de uso. No entanto, não devemos equivocadamente entender que estes indivíduos participem destas comunidades impulsionados por um espírito altruísta, em que façam suas contribuições sem esperar ganhos, apenas os ganhos não são mais os tradicionais.

Por extensão do pensamento, uma questão central surge do paradigma do *OSS*: quais as razões/incentivos que um indivíduo tem para contribuir, numa primeira análise de forma gratuita, para o desenvolvimento de *softwares* de código aberto? Ou seja, quais os ganhos que alguém auferir ao dedicar tempo e disponibilizar sua capacidade intelectual num projeto *OSS*?

Alguns estudiosos já se fizeram essas perguntas e as respostas são as mais diversas. Alguns autores acreditam que as motivações podem ser inúmeras: vontade de participar de um movimento social, identificação com o projeto (*Linux*, por exemplo), etc. Outros entendem que “pessoas inovam para melhor satisfazer seus interesses” (FRANKE; HIPPEL *apud* KROGH; HIPPEL, 2003a, p.1154). Dessa forma, os indivíduos se envolvem em projetos *OSS* para conseguir um *software* mais adequado ao que se quer. Partindo de um determinado projeto *OSS*, o indivíduo pode obter um resultado mais ajustado aos seus interesses, diferentemente dos *softwares* proprietários onde, via de regra, o máximo que se consegue personalizar é partindo da ativação ou desativação de módulos de um *software* base.

Na Figura 4, observamos um esquema que sinaliza as motivações dos participantes do *software* de código aberto.

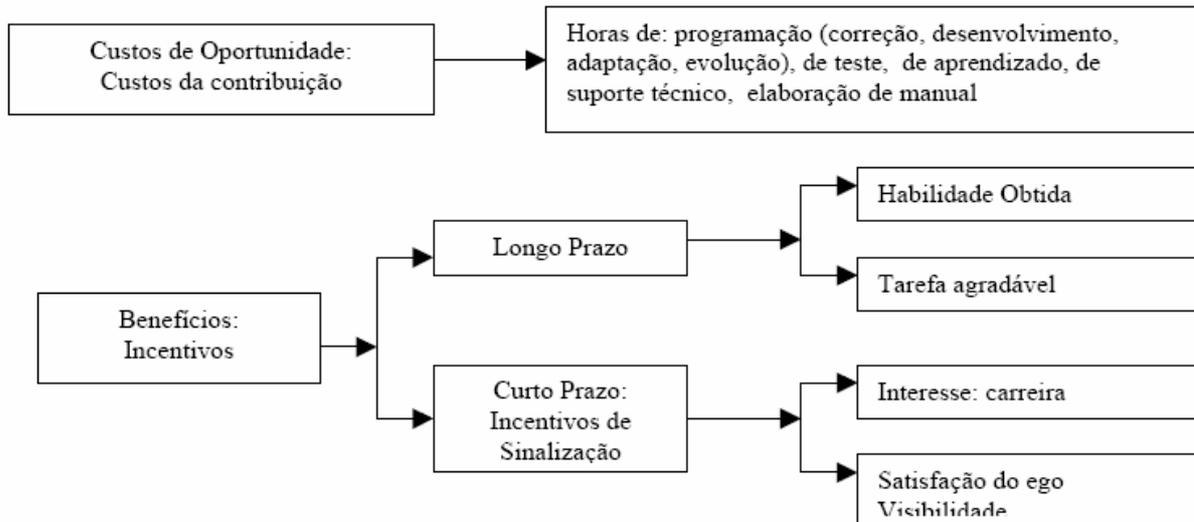


Figura 4 – Motivações dos participantes do *software* de código aberto.

Fonte: ARROYO; MERLO; SIMÕES, 2005, p. 14.

Este comportamento por parte dos agentes em relação ao OSS não coloca em xeque a teoria econômica convencional que postula um indivíduo caracterizado pela racionalidade e, que *a priori*, visa auferir o máximo ganho com o menor esforço, mas flexibiliza a racionalidade e aponta para um novo modelo de negócios. No entanto, a racionalidade vigente neste tipo de projeto não é mais a racionalidade das organizações hierarquizadas, mas, sim, a racionalidade da lógica das organizações em rede. O modelo de negócio não se baseia mais num pagamento direto pelo trabalho executado, na forma de licenças ou outra forma similar, busca-se, através das inserções em redes, a produção e a apropriação de valor. Esta perspectiva sofre uma mudança oriunda da forma conceitual: o *software* deixa de ser produto, em última instância, e passa a ser fonte de serviço. Isto é, os participantes não estão interessados em ganhar com a comercialização dos *softwares* desenvolvidos pelas comunidades que participam, estão vislumbrando os ganhos na forma de serviços de desenvolvimento, manutenção, garantia, customização, gravação em mídia ou pela prestação de demais serviços paralelos.

A primeira vista talvez seja difícil imaginar que algo que não visa ganho financeiro, de imediato, possa ter sucesso devido ao fato de não conseguirmos visualizar onde esses ganhos ocorrem. Entretanto, além dos incentivos subjetivos já apresentados, os envolvidos em projetos OSS apostam suas fichas num mercado essencialmente voltado aos serviços a serem desenvolvidos junto com o sistema.

Por fim, um outro aspecto muito interessante está presente na configuração que sustenta esses projetos: a articulação em rede. O desenvolvimento dos *softwares* OSS deixa de acontecer em núcleos de P&D, instalados fisicamente numa determinada empresa, e acontece a partir da conexão em comunidades que operam em rede, principalmente através da Internet. È como se as capacidades intelectuais fossem potencialmente aumentadas, já que o número de participantes num determinado projeto pode chegar a números bem elevados. Essa característica é bastante interessante, pois derruba o modelo convencional de fabricação de *software*. Via de regra, a indústria de *software* mantém em seu quadro funcional um determinado número de programadores que estão sujeitos ao comando central que define as diretrizes produtivas. Nesta forma, os programadores se comportam pura e simplesmente como outros operários de outras indústrias. A configuração em rede possibilita uma maior liberdade e flexibilidade aos participantes de projetos OSS. Os indivíduos definem a divisão do trabalho a partir de motivos outros que não os apontados pela direção de uma empresa.

2.2.4 Aspectos Econômicos

O *software* é informação, é produto, é insumo, é ferramenta, entre tantas outras coisas. Como apontado por Sampaio (2006, p.49): “[...] o *software* é uma informação que interage e tem poder de comando sobre o funcionamento de qualquer equipamento informatizado (*hardware*), servindo como insumo tecnológico e ferramenta de *design* nas diversas atividades humanas”.

Da ótica da informação, o *software*, assim como as demais informações, tem como característica marcante o alto custo de produção, além da componente de risco, mas que possui baixo custo marginal. Isto quer dizer, que a produção de *software* é uma atividade *design oriented*¹², que envolve altos custos de elaboração e criação do produto, onde a principal etapa está no *design*, na concepção do produto. Apresenta baixos custos na sua reprodução – o custo marginal tende a zero. Sendo assim, seus ganhos de escala são muito maiores que outros bens de natureza material. A Internet tem se mostrado um importante mecanismo de reprodução e que torna quase zero o custo de acesso do usuário ao *software*.

Apesar de ser uma informação, o *software* está sujeito a ser patenteado (em alguns países, como nos Estados Unidos) e para que os indivíduos possam fazer uso, precisam pagar aos seus criadores algum tipo de dividendo – *royalties*, etc. É desta forma, que o modelo tradicional do *software* proprietário se coloca para o mercado, cobrando dos usuários alguma forma de pagamento para que estes possam fazer uso de seus produtos. Em contrapartida, esta cobrança gera um efeito bastante característico do capitalismo moderno, onde surge paralelamente ao mercado formal, a indústria da pirataria. Esta indústria movimentava grandes quantias anualmente, desviando os ganhos que deveriam ser auferidos pelos proprietários intelectuais de diversos produtos, entre eles, o *software*. Inclusive alguns autores sinalizam, em contraponto, as diversas discussões levantadas por esse tema, que a pirataria pode ser consentida em alguns casos. Mas isso abordaremos mais a frente, ainda nesta seção.

Segundo Messerschmide e Szyperski (*apud* SAMPAIO, 2006, p.51), o *software* possui ainda outras características interessantes:

- É uma tecnologia ubíqua, pois tornou-se parte de praticamente qualquer coisa, seja de forma explícita ou escondida em algum equipamento informatizado;
- Possui interatividade, devido ao fato de que o *software* está criando um ambiente que nós interagimos com objetos inanimados de maneiras cada vez mais sofisticadas;
- O *software* aparece como algo muito importante na era do “capitalismo informacional”, onde os acessos à informação e ao conhecimento são condições indispensáveis a qualquer nação capitalista, comportando-se como principal *commodity*¹³. Além disso, o *software* apresenta uma transversalidade que por si só já merece especial atenção.
- Tem como base as pessoas, por ser expressão e representação dos desejos e ações de um programador humano, que está inserido em alguma organização humana;
- O seu desenvolvimento é social: a atividade de desenvolver *software* se torna a cada dia uma atividade coletiva, multidisciplinar, com forte interação entre os agentes e demandante de altos níveis de coordenação e gerenciamento.

¹² O *design oriented* significa que o vetor que norteia a concepção de produto é o seu desenho, sua arquitetura. A etapa de concepção do produto é fortemente guiada pela necessidade de facilitar a interação do indivíduo com o produto.

¹³ Bem negociado no mercado internacional que tem seu preço determinado pelas condições de demanda e oferta deste mesmo bem no mercado mundial.

- Sempre poderá ser melhorado, pois as restrições estão associadas a questões financeiras e humanas;
- Cada vez mais sofisticado e complexo: isto é possível devido ao melhoramento crescente das condições de infra-estrutura em conjunto com a necessidade de sistemas computacionais;
- A indústria de *software* está em processo radical de mudança – este tópico abordaremos com mais profundidade em ocasião oportuna;
- Pode ser regulado pelo Estado: possibilita um vasto campo para políticas regulatórias.

A indústria de *software* também tem como característica a vantagem competitiva para aqueles que são pioneiros. Isso ocorre devido as externalidades de rede – quando o valor de um bem para um consumidor ou usuário depende do número de outros usuários –, na difusão dos programas e padrões de tecnologia, com forte *lock-in* (aprisionamento) em mercados ou segmentos dominados por poucas ou uma única empresa (ROSELINO apud SAMPAIO, 2006, p.55).

Este aprisionamento ocorre, independentemente da existência de superioridade técnica do produto, em virtude do comportamento e da aceitação desta por parte dos seus primeiros usuários. Uma vez que os usuários optem por uma tecnologia, a migração para outra tecnologia incide em custos e problemas de compatibilidade, o que cria custos de troca e acaba por aprisionar o agente.

Para gerar esse efeito de *lock-in*, alguns autores chegam a levantar que alguns produtos permitem em alguma medida a pirataria dos seus *softwares*, pois através desta difusão dos seus produtos, o aumento do número de usuários causaria o aprisionamento dos indivíduos a esta tecnologia. De forma bem simples, quanto mais pessoas utilizam um *software*, mais ele se torna padrão pra os demais usuários e, por conseguinte, a migração para outras tecnologias dependerá de muitos custos de troca.

Talvez do ponto de vista de políticas públicas, o ponto mais interessante do *software* está na sua transversalidade. Isso significa dizer que o *software* tem a possibilidade de criar externalidades positivas a outros setores da economia. Abordaremos esta característica em outro momento.

O *software*, atualmente, auxilia a gestão de pessoas, de processos, possibilita melhoria em logística, acelera transações, permite interação dos clientes, entre outras coisas. A tecnologia desenvolvida a partir da indústria de *software* possibilitará que outras firmas evoluam na ótica tecnológica, causando uma espécie de amplificação dos resultados. Desta forma, o Estado precisa assumir uma postura de agente de fomento do *software*, não através da criação de linhas de financiamentos, pura e simplesmente, mas através da proposição e gestão de mudanças institucionais que possibilitem a produção de *software*.

Além de tudo o que já foi dito em relação ao *software* de maneira geral, o OSS ainda apresenta uma outra faceta apontada por diversos estudiosos. O *software* de código aberto é um *commons*. Isto significa dizer que ele é de domínio público. Devemos entender *commons* como um:

[...] recurso que qualquer participante de uma comunidade significativa pode fazer uso sem a necessidade de permissão de outros. Esta permissão não é necessária, pois o recurso não é objeto de controle legal [...]. Ou esta permissão é desnecessária devido ao fato desta já ter sido garantida. Nos dois casos, usar ou construir algo a partir desse recurso demanda apenas o acesso a ele¹⁴. (LESSIG, 2006, p.198).

No entanto, o próprio Lawrence Lessig aponta a ameaça existente aos *commons* no *cyberspace* (espaço virtual, Internet). Segundo Lessig (2006, p. 198), “podemos arquitetar o *cyberspace* visando preservar os *commons* ou não”. Mas, via de regra, “devemos escolher uma arquitetura baseada nos *commons*¹⁵”. Esta arquitetura é construída com base em questões institucionais, em especial, a legislação.

Esta arquitetura utilizada pelo OSS esta assentada nas licenças¹⁶ e tem sido adaptada aos outros *commons* que estão no *cyberspace*. No entanto, esta liberdade só está assegurada através da legitimação do Estado. Portanto, o Estado é um agente importante e deve criar legislações que possibilitem tais interesses.

¹⁴ Resource that anyone within a relevant community can use without seeking the permission of anyone else. Such permission may not be required because the resource is not subject to any legal control (it is, in other words, in the public domain). Or it may not be required because permission to use the resource has already been granted. In either case, to user or to build upon this resource requires nothing more than access to the resource itself (LESSIG, 2006, p.198).

¹⁵ We can architect cyberspace to preserve a commons or not. (Jefferson thought that nature had already done the architecting, but Jefferson wrote before these was code). We should choose to architect it with a commons (LESSIG, 2006, p.198).

¹⁶ As licenças serão devidamente tratadas em tópico específico deste trabalho.

2.2.5 Modelo de Negócios

O modelo de negócios original tem como fonte a indústria norte-americana de *software*, por ter sido esta a pioneira e, conseqüentemente, ter assumido a liderança mundial neste segmento. O *software* surge já inserido numa perspectiva mundial, mantendo ainda mais essa característica ao longo do seu desenvolvimento. O surgimento e o domínio da indústria de *hardware* nos EUA possibilitou um desenvolvimento paralelo da indústria de *software*, assegurando a liderança neste setor. Este desenvolvimento foi fortemente influenciado no pós-guerra e amplificado pela interferência governamental, em especial sob a forma de investimentos militares.

Este modelo estadunidense foi o mais difundido e tinha na sua base a proposta proprietária. Essa modelagem se vale da comercialização do *software* na forma de produto, onde os usuários realizam pagamentos aos detentores dos direitos autorais do *software*.

Quando se pensa em *software* de código aberto, a primeira pergunta que surge é: Como viabilizar um negócio em *software* sem a cobrança pela venda do produto? Ou seja, como o modelo tradicional baseia seus ganhos na comercialização do *software*, um modelo como o OSS que rejeita essa perspectiva parece inviável do ponto de vista dos negócios.

A viabilidade de negócios baseados em OSS, passa por duas esferas: a produção e o consumo.

2.2.5.1 A produção (Oferta)

Em relação à produção, o *software* como todos os bens de informação – música, vídeos, textos, etc – possuem altos custos na elaboração e concepção do produto. São atividades orientadas pelo *design*, onde a criação de um novo produto demanda a maior parte dos recursos. Da ótica da reprodução, os bens de informação podem ser “reproduzidos infinitamente, sem qualquer perda de qualidade, e principalmente, sem degradação do objeto original” (SALEH, 2004, p.33). Sendo assim, a reprodução e a distribuição desses bens apresenta custo marginal próximo a zero. Com a Internet, a distribuição de um *software* pode ser realizada com custos extremamente baixos, basicamente os custos de hospedagem em algum servidor para que as pessoas possam fazer o *download*.

Por ser um mercado fortemente marcado pelo *first mover*, as empresas que descobrirem nichos de atuação e tentarem se estabelecer de maneira sólida nessas novas possibilidades terão um grande diferencial de mercado.

Além disso, uma característica marcante do setor de *software* é a alta participação, se não o maior responsável, da força de trabalho dentro deste processo produtivo. A indústria é caracterizada pela grande demanda por profissionais capacitados, tendo na força de trabalho seu principal insumo. É a figura do “programador” o responsável pela criação e *design* do *software*. A força de trabalho é sem dúvida o maior insumo utilizado na produção de *software*. As linguagens de programação, apesar de terem evoluído ao longo do tempo, de nada servem sem a capacidade humana de geração do produto.

Para *software*, a localização não desempenha papel significativo, haja vista que a distribuição dos seus produtos não acarreta altos custos de transporte, nem a aquisição de insumos demanda uma estrutura logística bem definida. Além disso, poderíamos dizer que trata-se de uma indústria “limpa” uma vez que a sua operação não incorre na geração de resíduos ou de outras formas quaisquer de expurgo. É interessante ilustrarmos essa irrelevância da localização, mas não desconsiderando o papel de condições que, em última instância, estão fisicamente dispostas em determinado lugar. Ou seja, facilidade de acesso a instituições de formação profissional, disponibilidade de profissionais qualificados e outras condições de infra-estrutura, desempenham um fator de fomento a esta atividade.

A necessidade de profissionais qualificados leva a obrigatoriedade de instituições de capacitação que, em última instância, passa pelas ICTs. Portanto, mecanismos institucionais que privilegiem a aproximação destas firmas aos centros acadêmicos, parecem ser indispensáveis ao êxito deste processo.

A principal dúvida em relação a projetos OSS diz respeito aos mecanismos de financiamento, ou seja, como se dá o aporte de recursos em projetos que não miram na comercialização de um produto final.

Em projetos OSS, grande parte dos indivíduos contribuem sem uma contrapartida direta financeira, como já discutido em ponto anterior nesse trabalho. Mesmo na ótica das empresas, a participação em projetos OSS através da manutenção de pessoal técnico em seu quadro

funcional ainda é viável em relação aos custos. Pois, de qualquer forma, o desenvolvimento num sistema proprietário é mais ineficiente e/ou mais caro porque “é necessário *reinventar a roda* a todo momento, uma vez que não é possível reutilizar tecnologias que já estejam prontas e disponíveis” (ibid., p.37). Ou seja, em OSS os programadores podem reutilizar soluções já conhecidas e incorporá-las a seus novos *softwares*, aumentando assim a eficiência e reduzindo os trabalhos desnecessários.

Em relação à comercialização, as formas de negócios podem ser diversas. Uma primeira possibilidade diz respeito a utilização de um OSS como base para a criação de um produto proprietário. Essa possibilidade existe quando o licenciamento do *software* original permite essa condição – como em licenças BSD discutidas nesse trabalho em tópico adequado.

Um segunda realidade, e talvez a mais forte, é o foco na prestação de serviços. Indivíduos que participam do desenvolvimento do OSS, têm uma vantagem relativa no mercado e podem vender sua força de trabalho na forma de consultorias, suportes, gestão de segurança, entre outras formas. A prestação de serviço é sem dúvida o mais forte vetor do OSS e talvez a base para um novo modelo de negócio. Empresas como Conectiva (Brasil), Red Hat (EUA), Suse (Alemanha) e Mandrake (França) apostam suas fichas na oferta de serviços a seus parceiros, além de outras que oferecem seus produtos no mercado de produto tradicional, a exemplo da IBM.

Uma outra possibilidade é a comercialização de mídias contendo os *softwares*, bem como manuais, revistas e apostilas que orientem o uso desses *softwares*. Diversas revistas brasileiras, por exemplo, apostam nesse sentido.

Também existe a possibilidade da utilização de sistemas OSS por parte dos fabricantes de *hardware*, como é comumente chamada, na forma embarcada. Dispositivos de conexão ao computador, computadores de mão, telefones celulares, telefonia via satélite e muitas outras áreas precisam de *softwares* para gerir sua parte física.

Enfim, as possibilidades aqui listadas são apenas algumas formas que as firmas podem se posicionar para auferir ganhos a partir do código aberto. Além disso, diversas outras formas podem surgir ao longo do processo evolutivo desse modelo.

Em relação ao uso, a adoção de *software* de código aberto apresenta diversas vantagens. Além da redução dos custos com pagamento de licenças, se apresenta a possibilidade de quebra do monopólio por parte de uma empresa, tornando o usuário menos vulnerável e garantindo uma maior independência.

A adoção de OSS permite uma maior adaptabilidade do *software* ao uso almejado pelo usuário. A possibilidade de alterar o código, corrigindo problemas e melhorando alguns pontos, permite ao usuário ter um *software* mais ajustado a seus interesses.

Tecnicamente, os OSS não são inferiores em relação aos *softwares* proprietários. Em verdade, alguns especialistas afirmam que os *softwares* de código aberto são mais estáveis e apresentam boa qualidade em virtude das revisões realizadas constantemente por suas comunidades. Um único ponto ainda discutível em relação ao OSS é a pouca prioridade dada as documentações dos *softwares* - manuais, suporte, etc. – em função de ainda estarem alicerçados na cultura *hacker*.

Diante do exposto, é viável o empreendimento em *software* de código aberto. No entanto, é importante remodelar as práticas vigentes de negócios em *software*, pois a comercialização do *software* como produto deixa de ser o foco e surge o serviço como principal possibilidade de renda.

2.2.5.2 O mercado (Demanda)

A demanda por *software* cresce mundialmente ao longo dos anos, apresentando tendência de continuar crescendo, e se constitui numa interessante opção de inserção para países como o Brasil. O próprio mercado interno brasileiro, no período de uma década, passou de 0,2 % do PIB em 1991 para 0,7% em 2001¹⁷, um montante de cerca de R\$ 8 bilhões, tamanho compatível com países como Índia, Irlanda e Israel. É interessante pontuar, que médias e grandes empresas contam em seus quadros funcionais com profissionais de TI – Tecnologia de Informação – que respondem pela produção interna de *software* e que não são

¹⁷ Dado disponível no documento *Inserção Competitiva do Brasil no Mercado Internacional de Software*, de autoria de Eratóstenes Edson Ramalho de Araújo e Silvio Romero de Lemos Moreira, disponível em: <[http://golden.softex.br/portal/softexweb/uploadDocuments/Insercao-competitiva-do-Brasil-no-mercado-internacional-de-software\(1\).pdf](http://golden.softex.br/portal/softexweb/uploadDocuments/Insercao-competitiva-do-Brasil-no-mercado-internacional-de-software(1).pdf)>. Acesso em: 23 mai. 2007>

contabilizados nesses números por não ser a atividade fim da empresa. Tal pensamento nos leva a ponderar que a participação do *software* no Brasil é ainda maior.

Em mercado de *software* básico, como os aplicativos fornecidos por empresas como a Microsoft, a quebra destas barreiras a entrada se torna uma tarefa quase impossível. Apesar de iniciativas como o Linux que aparece com uma opção ao sistema operacional predominante, a derrubada deste monopólio é extremamente complexo.

Mas em outras áreas do *software*, como os *softwares* aplicativos – aqueles que normalmente são desenvolvidos para aplicações específicas, como controle de pessoas, automação bancária, etc – as firmas conseguem uma maior competitividade.

As firmas brasileiras se caracterizam por atuarem no mercado interno, com raras exceções de atuação no mercado mundial. Estão concentradas na produção de *softwares* aplicativos e têm como característica preponderante seu pequeno tamanho. Essa atomização dificulta a inserção destas firmas no mercado mundial. Políticas públicas que incentivem a cooperação entre essas firmas de forma que ganhem tamanho e, por conseguinte, auferam economias de escala e de escopo, poderão sinalizar uma saída para este problema.

Mais uma vez, verificamos a importante interação entre instituições e organizações e como o Estado, no papel de uma grande instituição capaz de modificar outras, pode ser decisivo na conquista de mercado por parte das empresas brasileiras.

Vale ressaltar, que as características do mercado mudam quando nos referimos às estruturas convencionais – *software* proprietário – e nova estrutura – como o *software* de código aberto.

No caso do *software* proprietário, as empresas objetivam vender seus produtos e lucrar com os dividendos provenientes de patentes e ou licenças de uso.

No caso do *software* de código aberto, os inovadores são os mesmos agentes que irão fazer uso dos ganhos resultantes das inovações. Esses indivíduos não são “comerciantes” de *software*, mas agentes que estabelecem uma parceria a partir da força de trabalho. Percebemos, portanto, uma mudança no fluxo e posturas deverão ser diferentes quando pensarmos cada uma destas possibilidades. Mas o fato dos usuários fazerem uso das inovações não significa

dizer que não há circulação monetário. Pelo contrário, os ganhos monetários serão gerados a partir de outras formas, como sinalizado no modelo de negócios discutidos no ponto anterior.

2.2.5.3 Os novos entrantes

Um caso bastante relevante de como é possível “fazer dinheiro” em código aberto, está na IBM, gigante da área de tecnologia que tem investido fortemente em OSS. A IBM é a empresa americana que detém o maior número de patentes, algo na ordem de 29.000 registros¹⁸. De acordo com a Figura 5, podemos observar a evolução de serviços relacionados a Linux – sistema operacional em código aberto disponível em várias distribuições.

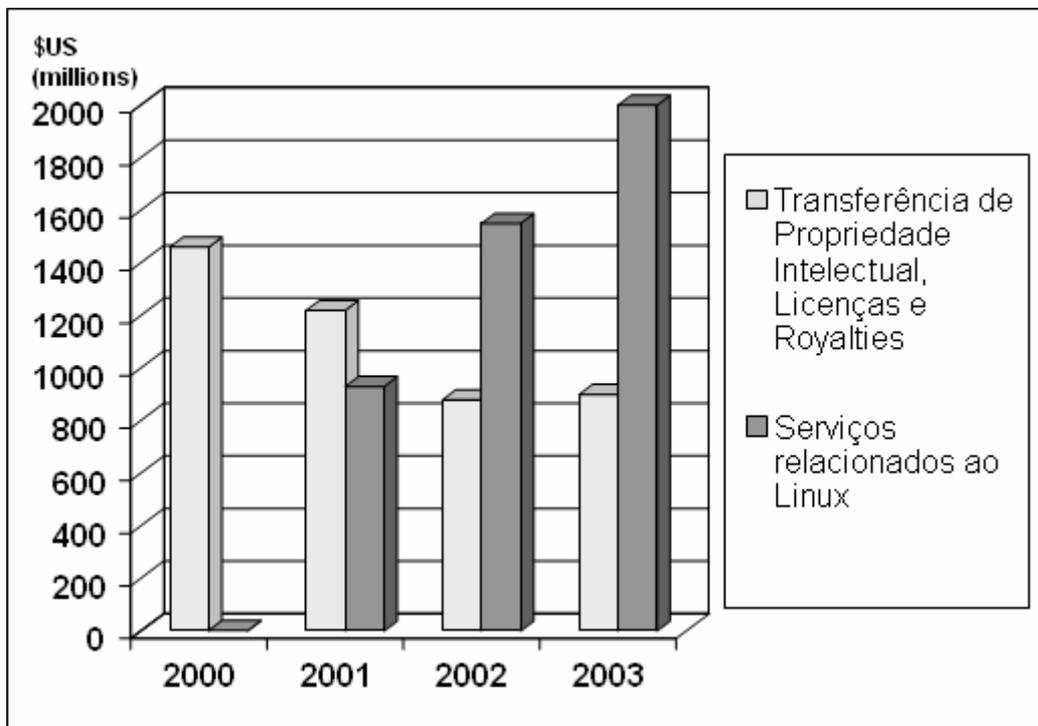


Figura 5 – Receitas da IBM, 2000-2003.

Fonte: BENKLER, 2006, p.47

Como observamos na figura 5, em um período de três anos a IBM simplesmente mudou o *mix* das suas receitas referentes a transferências de propriedade intelectual e serviços ligados ao Linux, em relação a sua receita total. As transferências representavam um montante da ordem de US\$ 1,4 bilhão em 2000, em vista de uma participação inexpressiva dos serviços em Linux neste mesmo ano. Em 2003, o quadro se reverte e os serviços relacionados ao Linux já geram

¹⁸ Benkler, 2006, p. 46.

para IBM uma receita da ordem de US\$ 2 bilhões, em detrimento de uma queda das receitas com transferência de propriedade intelectual.

A IBM é apenas uma das empresas que tem feito grandes investimentos nesta área. Empresas como HP, Red Hat, Amazon.com e CNN.com já investem, operam, prestam serviços ou utilizam *softwares* de código aberto. Fica evidente que estes gigantes não colocariam seus negócios em perigo apenas para “economizar” no pagamento de licenças. Eles utilizam essa via por acreditarem na capacidade técnica de competir e até superar os produtos oriundos de tecnologias proprietárias.

No Brasil, empresa como o Banco do Brasil, já envidam esforços para utilizar *softwares* livres na suas estações de trabalho, bem como nos seus terminais de auto-atendimento. Um dos braços do Banco, a Cobra Tecnologia, já atua fortemente na adequação e desenvolvimento de sistema em código aberto para o sistema bancário.

Portanto, as empresas que forem capazes de atuar em *software* de código aberto não entrarão num mercado “vazio” do ponto de vista da possibilidade de ganhos monetários, apenas precisarão adequar seus modelos de negócio a essa nova realidade.

3 POLÍTICA BRASILEIRA DE INCENTIVO E ADOÇÃO DE SOFTWARE LIVRE

O Governo Federal ao longo das suas últimas duas gestões – Fernando Henrique e Lula – tem demonstrado forte interesse nas questões relacionadas ao *software* no Brasil e é sobre como esse interesse se manifesta que trataremos nessa seção. Essa postura do Estado em relação a indústria de *software* surge em virtude de dois pontos que consideramos centrais: a sua relevância estratégica e econômica no capitalismo contemporâneo e uma oportunidade de melhor gestão da máquina pública.

Ainda no governo FHC, um documento intitulado *Diretrizes de Política Industrial, Tecnológica e de Comércio Exterior* assinala as intenções do governo federal em fomentar a atividade de *software* no Brasil. Essa conduta está pautada na relevância de alguns setores na corrida de redução do distanciamento tecnológico existente entre os países centrais e os países em desenvolvimento. Este documento especificamente versa sobre a necessidade de fomento de quatro atividades tidas como estratégicas pelo Estado Brasileiro: bens de capital, fármacos/medicamentos, semicondutores e *softwares*. O fomento a estas atividades levariam o país a reduzir ou eliminar os gargalos existentes em nossa esfera produtiva, possibilitando um efetivo crescimento da nossa capacidade tecnológica e, por conseguinte, causaria uma aproximação aos padrões de tecnologia vigentes nos países centrais.

A escolha destes segmentos não foi realizada de maneira aleatória. Esta opção foi norteada por um princípio bastante relevante que é a transversalidade. Entenda-se por transversalidade a capacidade que um setor possui de alavancar outros setores, diretamente relacionados ou não. No tocante ao *software*, o desenvolvimento de uma indústria de *software* irá impulsionar o desenvolvimento a partir de duas premissas, como apontadas por Freire e Brisolla (2005):

Basicamente, o desenvolvimento de uma indústria de *software* local, visando tanto o mercado interno quanto o externo, pode impulsionar o desenvolvimento regional por dois motivos: [...] primeiro, o *software* é um produto facilitador de interações e segundo, [...] o desenvolvimento de uma indústria de *software* requer a construção de competências que podem gerar sinergias para o desenvolvimento de outras atividades de alta tecnologia. (FREIRE; BRISOLLA, 2005, p.103).

O fomento a atividade de *software* tem uma contribuição para o Brasil extremamente relevante e o aproveitamento da capacidade técnica empresarial já disponível neste setor é um

fator facilitador para o crescimento sustentável desta atividade. A existência de uma indústria brasileira de *software*, apesar de caracterizada pela atomização das firmas, incita o pensamento de que o primeiro passo já foi dado e que, com algum esforço e ações bem direcionadas, poderemos tornar o setor competitivo tanto no âmbito interno quanto no comércio internacional. De maneira estratégica, para o Brasil, “[...] a internacionalização de sua indústria de *software*, [...] além de uma necessidade, é uma exigência para se conseguir inclusive competitividade interna, aproveitando as capacidades locais em muitos segmentos e nichos específicos”. (Ibid., p.124).

No entanto, o incentivo ao *software* não pode se restringir ao modelo vigente, assentado no *software* proprietário, nem muito menos fazer uso de antigos instrumentos estatais de intervenção. Precisa, acima de tudo, estar voltado para os novos vetores sinalizados pelo capitalismo que está em mutação a todo momento.

Associado a condição estratégica característica do *software*, também observamos a intenção do governo em casar o desenvolvimento tecnológico à melhoria da gestão pública. Em especial, seguindo o receituário neoliberal de redução dos gastos públicos e, sendo assim, as corretas opções em *software* podem resultar em significativas reduções dos gastos provenientes do pagamento de licenças de uso.

É neste sentido, que observamos um esforço generalizado nas esferas federais de fomentar a adoção e difusão do *software* livre em substituição aos convencionais *softwares* proprietários. Esta será a discussão abordada no tópico seguinte.

3.1 MOVIMENTO EM PROL DO SOFTWARE LIVRE NAS INSTITUIÇÕES PÚBLICAS

No Brasil, o primeiro estado da federação a adotar o *software* livre foi o Rio Grande do Sul, ainda no governo de Olívio Dutra – que governou o Estado no período de 1999 a 2003 – e que motivou as discussões na esfera federal. A adoção do *software* livre pelo governo federal tem como justificativa o alcance de mais estabilidade e segurança no uso dos programas, além de independência em relação a fornecedores. Viabilizando o acesso ao código-fonte dos *softwares* adotados pelo Estado, é possível uma maior independência no fornecimento do *software* – se libertando do monopólio de fornecedores como a Microsoft -, além de permitir

um ambiente mais seguro, por possibilitar a identificação e a correção de eventuais falhas de segurança – *backdoors*.

No entanto, é também sabido que a adoção de *softwares* livres trarão uma economia considerável de recursos do Tesouro Nacional. Ainda hoje, a União gasta um montante significativo no pagamento de licenças de uso dos *softwares* que utiliza. Segundo matéria publicada na Folha de São Paulo¹⁹, é possível reduzir estes gastos em até 80% em três anos:

O governo economizou aproximadamente R\$ 28,5 milhões em 2004 com a adoção de *software* livre. O valor, apurado pelo Comitê Técnico de Implantação do *Software* Livre (CISL), se refere ao que deixou de ser gasto com licenças de *softwares* proprietários [...] A economia representa de 7% a 9,5% do gasto anual do governo federal com pagamento de licenças, que é estimado entre R\$ 300 milhões e R\$ 400 milhões. [...] Nos próximos três anos, a expectativa do ITI (Instituto Nacional de Tecnologia da Informação, ligado à Casa Civil) é que se possa chegar a economizar até 80% do que o governo gasta com licenças, caso haja verba suficiente para treinamento. (MEDINA, 2005).

De acordo com o que vimos, a decisão do Governo em apoiar iniciativas de *software* livre estaria assentada nesses dois pilares: transversalidade e independência de fornecedores/redução de gastos públicos.

Além destas perspectivas, alguns estudiosos, entre eles Lawrence Lessig, apontam a relevância do *software* como agente garantidor de transparência dos processos democráticos. Esta transparência só seria assegurada mediante a utilização de *software* de código aberto. Segundo Lessig²⁰ (2006, p.143) “em certos contextos, a transparência do código aberto é necessária” e só através da “abertura do código e [...] da certificação desses códigos por inspetores independentes” é possível assegurar a transparência necessária. Um exemplo interessante diz respeito a votação eletrônica. O Brasil já realiza suas eleições de maneira eletrônica desde 1996 e todo o processo eleitoral é realizado através de práticas computacionais: o voto, a transmissão das informações a central de processamento e a apuração dos eleitos. Todo o processo é gerido através de um *software* e daí vem a pergunta: Como poderemos ter a certeza que a votação e apuração é realizada de maneira correta, sem nenhum tipo de manipulação das informações ou favorecimento de indivíduos? A resposta é

¹⁹ MEDINA, Humberto. Governo Federal poupou R\$ 28 mi com software livre. **Folha de São Paulo**, São Paulo, 20 mar. 2005, Dinheiro. Disponível em < <http://www1.folha.uol.com.br/folha/dinheiro/ult91u94554.shtml>>. Acesso em: 02 mai. 2007

²⁰ Os trechos e idéias associados a Lessig estão disponíveis no seu livro *Code – version 2.0*, no capítulo 8, página 143, conforme referência bibliográfica.

bastante simples: Não temos como ter essa certeza. Isso se dá por não termos acesso ao código-fonte utilizado no programa que gerencia a eleição. Uma maneira de garantir a transparência do processo eleitoral seria a abertura do código-fonte para que as pessoas tecnicamente capacitadas pudessem entender o funcionamento do processo e julgar a validade dos métodos utilizados. Dessa maneira, um princípio básico da democracia seria ainda mais consolidado através da transparência dos processos de eleição.

Mas o fomento a atividade de *software* deve partir sem dúvida de uma discussão sobre os mecanismos instituições que irão viabilizar essa prática produtiva. É neste sentido, que o próximo tópico do trabalho irá discutir as questões legais.

3.2 OS FUNDAMENTOS ECONÔMICOS DAS INOVAÇÕES FOCADAS NO USUÁRIO

Toda discussão traçada até aqui, não estaria completa sem algumas observações sobre o *background* que valida o *software* de código aberto. Como já exposto ao longo do trabalho, o OSS não é apenas uma questão marginal ao modelo vigente, em verdade, se apresenta como uma faceta de um movimento ainda maior: a inovação aberta, que abdica da racionalidade econômica ortodoxa e relativiza os modelos de negócios vigentes. Apesar de não ser o principal foco do trabalho, no entanto impossível de ser desprezada, devemos fazer algumas ponderações sobre a inovação aberta para que a questão do *software* de código aberto fique ainda mais consolidada.

Da perspectiva econômica duas características são marcantes nos bens de caráter informacional. A primeira delas é que os bens de informação – inclusive *software* – são bens *nonrival*, isto é, são bens que quando consumidos por uma pessoa ainda podem ser consumidos por outra sem nenhum tipo de perda.

A segunda condição é que a informação é retroalimentada. Ou seja, a informação é insumo de entrada e produto de saída dentro do seu processo produtivo. O que se produz hoje em relação a informação tem como base a produção de informação feita no passado.

Essas duas características legam ao *software* duas importantes condições. Do ponto de vista da rivalidade, o consumo de *software* por um indivíduo não impossibilita que outro possa

fazer uso. Isso se afirma no baixo custo marginal característico do *software* – que tende a zero –, isto é, a sua reprodução e/ou distribuição não acarreta grandes aportes de investimento, pelo contrário, o foco dos recursos está na sua concepção.

Já a retroalimentação impõe que as informações já produzidas deveriam estar disponíveis para serem utilizadas na produção de nova informação. Essa disponibilidade não é característica do sistema proprietário. Apenas o produtor desse tipo de *software* pode contar com o que “já foi produzido”. Todos aqueles que queiram produzir novos *softwares*, não poderão contar com essas informações em virtude do código estar fechado, inacessível.

Dessa maneira, o *software* de código aberto por natureza se presta às duas condições. Os agentes podem copiá-lo, distribuí-lo ou modificá-lo livremente e ainda podem fazer uso do que já foi produzido para novos produtos.

Continuando o raciocínio, devemos diferenciar os modelos de inovação em *software*. De maneira espectral, poderíamos colocar o modelo de investimento privado de inovação em uma extremidade e na outra ponta estaria o modelo coletivo de inovação.

O primeiro se baseia no investimento privado, realizado pelas empresas, como mecanismo de viabilidade da inovação. Nesse modelo, as empresas investem em P&D e os retornos alcançados com os resultados da pesquisa são apropriados pelo investidor que financiou o projeto. É o modelo vigente na maioria das firmas que mantêm em seu quadro um grupo ou vários grupos inseridos na estrutura organizacional da firma e que respondem pelas inovações. Em geral, as firmas se valem de mecanismos institucionais – leis, patentes, direitos autorais, etc – para aprisionar esses ganhos, evitando que terceiros façam uso dessas inovações sem a devida autorização e, por conseguinte, sem o pagamento referente a este uso.

O modelo coletivo de inovação “requer que os participantes cedam o controle do conhecimento que eles desenvolveram para um projeto e torne-o um bem público, de maneira incondicional, transformando-o em um *common pool*”²¹ (KROG; VON HIPPEL, 2003b, p.213). Ou seja, este modelo preza pela reparação da perda social do aprisionamento do

²¹ It requires that contributors relinquish control of knowledge they have developed for a project and make it a public good by unconditionally supplying it to a “common pool” (KROG E VON HIPPEL, 2003, p.213).

conhecimento por parte do modelo privado de inovação. No entanto, este modelo incide num problema de motivação para os pesquisadores desenvolverem seus projetos. Geralmente, esse conflito é transposto através de financiamento por bolsas de pesquisas pagas pelo governo e/ou premiações.

O *software* de código aberto, de fato, se vale dos dois modelos. Krog e von Rippele chamam esse modelo de *private-collective*, em português, privado-coletivo. Segundo esses autores, neste modelo:

“[...] os participantes de um projeto de *software* de código aberto fazem uso dos seus próprios recursos para investir em um novo código-fonte. Em princípio, estes inovadores podem ser tidos como os proprietários dos direitos sobre o código criado, mas, em vez disso, eles optam por tornar esses produtos bens públicos. Desta forma, o resultado dessa conduta traz para a sociedade o melhor dos dois mundos – novo conhecimento é criado a partir de fundos privados e depois ofertado gratuitamente para todos²²” (KROG; VON RIPPELE, 2003b, p. 213).

Dentro desta perspectiva o código aberto possibilita que a inovação ocorra de maneira aberta, onde os indivíduos que inovam também são os que fazem uso dessas inovações. O *software* de código aberto é em essência um perfeito exemplo de inovações desenvolvidas e consumidas em comunidades onde tudo é feito por e pelos usuários. Neste sentido, as lógicas de orientação impostas pelo mercado perdem sentido. A inovação deixa de ter um caráter essencialmente fechado, onde apenas um pequeno grupo detém sua propriedade e passar a ser aberta, onde os indivíduos podem acessá-la livremente.

Sendo assim, as relações com a propriedade não são mais as tradicionais. Conectando o pensamento de Paolo Grossi (2006) – que discute a partir da relação homem-terra – com a nossa abordagem, verificamos um surgimento da “propriedade coletiva”. Para Grossi, a propriedade coletiva:

“[...] em toda a sua forma, tem – em meio a mil variações, segundo os lugares, os tempos e as causas mais diferentes – uma plataforma comum; e é a de ser garantia de sobrevivência para os membros de uma comunidade [...], de ter um valor e uma função [...], em que o conteúdo fundamental é um

²² [...] participants in open source software projects use their own resources to privately invest in creating novel software code. In principle, these innovators could then claim proprietary rights over their code, but instead they choose to freely reveal it as a public good. Clearly, the net result of this behavior appears to offer society the best of both worlds – new knowledge is created by private funding and then offered freely to all. (KROG, VON RIPPELE, 2003b, p.213)

gozo condicionado do bem, com um indiscutível primado do objetivo sobre o subjetivo” (GROSSI, 2006, p.8).

As relações tradicionais indivíduo-propriedade abrem espaço a uma nova realidade onde a comunidade se vale do uso dos bens que lhes pertencem. Tal apropriação ocorre por e através da construção de redes, onde a hierarquia, característica do mercado, não mais existe. O entendimento de rede vai:

[...] substituindo uma realidade piramidal feita de comandantes e comandados porquê calcada sobre comando (e, portanto, inervada de inelimináveis relações hierárquicas) por uma realidade estruturalmente diferente dominada por interconexões entre vários sujeitos e posições – precisamente a rede – onde não emerge uma graduação hierárquica, mas um jogo (e entrelaçamento) de recíprocos condicionamentos e integrações, que se desenvolve não em projeções geográficas restritas e fracionadas como eram os territórios das velhas entidades estatais mas em áreas sempre mais largas e abertas, tendencialmente mundiais. (*Ibid*, p. 109).

Tal movimentação em rede, propicia que na inovação aberta exista uma maior participação de elementos externos à firma. Chesbrough (2005) sinaliza que este é o caminho para o desenvolvimento de um modelo de negócio baseado na inovação aberta. Para ele:

“Inovação Aberta é um paradigma que pressupõe que as firmas podem e devem usar idéias externas e internar à ela [...] como possibilidade de alavancar sua tecnologia. Os processos em Inovação Aberta combinam idéias internas e externas à firma, a partir de arquiteturas e sistemas. O modelo de negócios utiliza ambos – idéias externas e internas – de maneira a criar valor, enquanto desenvolve mecanismos internos de apropriação de parte desse valor²³” (CHESBROUGH, 2005, p.4).

Em modelos desta natureza, em especial em código aberto, von Hippel (2004, p.84) sinaliza que três elementos são fundamentais:

1. pelo menos alguns usuários devem ter suficiente incentivo para inovar;
2. ao menos alguns usuários devem ter incentivos para voluntariamente abrir mão de suas inovações;
3. e a difusão dessas inovações pelo usuários devem serem capazes de competir com a produção e a distribuição comercial.

²³ Open Innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology. Open Innovations processes utilize business models to define the requirements for these architectures and systems. The business

Como já discutidos nos aspectos culturais que envolvem o *software* de código aberto, as motivações são diversas, mas todas possibilitam e viabilizam a produção destes projetos. A própria cultura *hacker*²⁴ de compartilhamento do conhecimento já assegura o segundo ponto. E, por fim, a Internet veio possibilitar o terceiro ponto, onde os custos de distribuição são extremamente baixos, possibilitando uma efetiva concorrência com os produtos comerciais.

A queda dos preços dos equipamentos computacionais, aliada a difusão do acesso a grande rede – barateamento das opções de acesso e dos preços de hospedagem de *websites* – tornaram a produção *peer-to-peer*²⁵ uma realidade. Essa mudança na organização da produção aproximou a capacidade técnica dos indivíduos ou de pequenos grupos à capacidade produtiva de grandes empresas como a Microsoft. Isso significa que do ponto de vista dos bens de produção – acesso a computadores, acesso a Internet, conhecimento das linguagens de programação, entre outras coisas – as comunidades que produzem OSS “não estão aquém” tecnicamente falando das grandes corporações.

É interessante observar que o surgimento deste novo paradigma só foi possível a partir do desenvolvimento do paradigma vigente. Se observarmos da ótica cronológica, o movimento de *software* livre passou por três etapas e/ou gerações. A primeira geração foi baseada na cultura *hacker*, onde a principal preocupação residia no fato de alargar as bases de conexão entre os indivíduos: a construção de *networks*. A seguir veio o movimento encabeçado e difundido pelo comércio que “empacotou” o conhecimento, desenvolvendo *interfaces* amigáveis aos usuários não-técnicos, procurando desqualificar a cultura hacker como “fora-da-lei”, mas da lei fossilizada que não mais reflete os interesses da sociedade e da economia dos segmentos inovadores intensivos em conhecimento e *design*. Foi a utilização comercial da Internet que guiou os anos 90 e que definiu pilares fundamentais da rede que conhecemos hoje. Só mais recentemente, observamos um fortalecimento do *software* livre associado a retomada da cultura *hacker* e, conseqüentemente, o seu reconhecimento como algo relevante por parte dos formuladores de políticas públicas.

model utilizes both external and internal ideas to create value, while defining internal mechanisms to claim some portions of that value (CHESBROUGH, 2005, p.4).

²⁴ Sinalizamos a importância do entendimento do termo *hacker* a partir do exposto neste trabalho, renegando qualquer caráter criminoso ou marginal.

²⁵ A expressão *peer to peer* – ou P2P – se refere a uma arquitetura de rede onde os participantes possuem capacidades e responsabilidades equivalentes. Essa configuração é o oposto a chamada “cliente/servidor”, onde existe uma relação de hierarquia entre os participantes. Essa expressão também é utilizada para denominar programas de computador que compartilham informação entre os usuários. Como exemplos, temos o Kazza, o Emule, o Napster, etc.

Notemos que foi a lógica de mercado, sempre voltada para o acesso de novos consumidores, que possibilitou uma propagação dos mecanismos que mais tarde colocariam essa mesma lógica em cheque. Em outras palavras, foi o desenvolvimento do *e-commerce* que possibilitou que as pessoas tivessem cada vez mais acesso aos computadores e a Internet, insumos fundamentais a construção do modelo OSS. A busca por mercados consumidores gerou ganhos de escala para a indústria de *hardware* fazendo com que mais e mais pessoas tivessem acesso à tecnologia necessária a construção de projetos em código aberto.

3.3 MARCO INSTITUCIONAL

A partir da visão apresentada sobre os fundamentos econômicos, precisamos retornar a importância do marco institucional para a viabilidade do OSS. Diante da contribuição de Williamson (2000), a discussão tratada neste trabalho está centrada nos dois níveis intermediários da análise institucional. Ou seja, estamos lidando com as “regras formais do jogo”, em especial, a propriedade privada e com a estrutura de governança. É a partir dessa visão que trataremos dois importantes aspectos institucionais: os direitos autorais, associados as “regras do jogo”, e as licenças, vinculadas as estruturas de governança.

3.3.1 Direitos Autorais e Legislação Brasileira

Antes de entrarmos na discussão sobre os mecanismos legais utilizados pelo Brasil relacionados ao *software*, entendemos a necessidade de tratar brevemente sobre as ferramentas legais que se propõem a defender os produtores de *software*.

Podemos separar três ferramentas legais que objetivam proteger os produtores de *software*: a marca registrada, os direitos autorais e a patente.

A marca registrada garante ao possuidor da marca que qualquer indivíduo que venha a citar esta marca precise fazer referência a seu proprietário – por exemplo, *softwares* compatíveis com o Windows devem referenciar a propriedade da Microsoft sobre esse produto – ou caso venha utilizar essa marca com fins comerciais deverá pagar ao possuidor do direito alguma espécie de remuneração pela utilização – um fabricante de camisetas que queira usar uma marca em seus produtos deverá pagar ao detentor desta marca.

Os direitos autorais²⁶ protegem um produto já existente de ser copiado ou modificado sem a autorização do seu detentor. No caso dos *softwares*, “os direitos autorais protegem um programa específico de ser copiado ou reescrito, [...] mas não pode evitar que outros escrevam um programa com funcionalidades similares²⁷” (DI COSMO, 2003, p.163 e 164). Nesse sentido, um *browser*²⁸ como o Internet Explorer (IE) – que possui código-fonte fechado – é protegido pelos direitos autorais, sendo dado a seu detentor o direito de definir a sua utilização, mas não pode evitar que programas similares como o Mozilla Firefox – *browser* de código-aberto e principal concorrente do IE atualmente – possam coexistir.

Os direitos autorais sofrem uma dicotomia a partir de duas óticas: a patrimonial e a moral. Do ponto de vista patrimonial, os direitos autorais garantem a seu possuidor que este determine as regras de sua negociação, possibilitando definir sob quais condições sua obra pode ser usado e/ou comercializada. Já os direitos morais asseguram que o nome do autor seja atrelado a obra, garantindo também que sua imagem não possa sofrer qualquer tipo de dano por indivíduos que façam mau uso dela.

Já as patentes têm na sua criação, segundo Di Cosmo (Ibid., p.164) o intuito de “motivar os inventores a difundir ao público os detalhes de suas invenções, fazendo-as acessíveis a outras maiores invenções, ao invés de manter em segredo estes detalhes com o intuito de proteger seus negócios²⁹”. Por conseguinte, os inventores receberiam o pagamento extra pela utilização de suas invenções, em contrapartida a disponibilização das invenções e dos conhecimentos. Vale ressaltar, que, ao contrário dos EUA, no Brasil não é possível patentear *software*. Para nossa legislação, os *softwares* estão sujeitos as leis de direitos autorais. Esses direitos são assegurados por um período de cinquenta anos e depois disso passam a figurar como domínio público.

²⁶ Direito autoral é também conhecido como *copyright*, em inglês. O *copyright* tem como símbolo a letra C dentro de uma circunferência - © - e se refere aos direitos patrimoniais da obra.

²⁷ El derecho de autor protege un programa específico ya existente, del mismo modo que protege una novela existente y específica [...] de ser copiada o reescrita sin autorización. [...] pero usted es muy dueño de escribir un programa con funcionalidad similar (DI COSMO, 2003, p.163 e 164).

²⁸ Browser ou navegador é um programa de computador que permite ao usuário acessar páginas da Internet.

²⁹ [...] motivar a los inventores a difundir al público los detalles de sus invenciones, haciéndolas accesibles pronto para fomentar mayores innovaciones, en lugar de mantener em secreto esos detalles para proteger su negocio (DI COSMO, 2003. p. 164).

Diante do exposto, podemos intuir que os direitos autorais possibilitam a concorrência, permitindo que outros indivíduos possam criar elementos de funcionalidade similar, sem a perda da proteção aos direitos do criador. Em outras palavras, quem inventa determinado produto ou processo continuará tendo a possibilidade de receber por sua criação, tendo a proteção da lei contra ações que venham a infringir esses direitos – cópias ilegais, pirataria, etc –, mas não exclui a liberdade dos usuários em optar por outro bem de características e funcionalidade similares. Isso possibilita a livre concorrência e a independência de fornecedores.

Por outro lado, as patentes têm sido sistematicamente utilizadas como agentes criadores de monopólios. Uma vez que a utilização da invenção e/ou inovação depende da autorização de seu detentor, os indivíduos podem se valer dessa condição para barrar os mecanismos concorrenciais. Uma inovação que depende de outra para ser materializada em produto fica dependente do possuidor da patente de autorizar a sua utilização ou mesmo do valor solicitado por este como pagamento pelo uso. Esse tipo de ferramenta legal, em certa medida, favorece as grandes corporações em detrimento das pequenas firmas.

Sendo assim, é de extrema relevância a discussão sobre qual ferramenta atende de maneira mais ajustada as características da produção de *software*.

No Brasil, as leis que regem o *software* são as leis nº 9.610/96 e a nº 9.609/98. A primeira define a proteção aos direitos autorais, delimitando quais obras se encaixam nesse tipo de proteção, além de versar sobre os direitos patrimoniais e morais sobre a obra.

A Lei nº 9609/98, trata das questões relativas aos programas de computador – *software*; que são tidos como:

[...] a expressão de um conjunto organizado de instruções em linguagem natural ou codificada, contida em suporte físico de qualquer natureza, de emprego necessário em máquinas automáticas de tratamento da informação, dispositivos, instrumentos ou equipamentos periféricos, baseados em técnica digital ou análoga, para fazê-los funcionar de modo e para fins determinados (BRASIL, 1998).

Essa lei dispõe sobre a proteção dos direitos autorais e de registro de *software*. Para a legislação brasileira, o *software* – por se tratar de bem de informação – não é passível de registro de patente, ao contrário, da legislação norte-americana.

3.3.2 Licenças

No entanto, além das leis que regulam os direitos autorais devemos precisar as licenças que estão conectadas a esta legislação.

Em relação ao *software* proprietário, o Brasil segue o caminho dos países centrais, a partir de acordos internacionais que estabelecem a proteção aos direitos autorais baseados no *copyright*. A Microsoft, principal representante e defensora dessa forma de distribuição de *software*, faz uso da EULA – *End User License Agreement* – que estabelece uma concessão a um usuário-final ou qualquer entidade legal para a utilização de seus produtos, firmando os limites permitidos de uso. Via de regra, essa licença impede que o usuário copie, distribua ou modifique esse produto sem o consentimento da empresa e concede a seu possuidor o acesso ao suporte e as atualizações de segurança disponibilizadas por essa firma.

No caso do *software* livre, a primeira licença utilizada foi a GNU GPL – *General Public License*. Segundo Saleh (2004, p. 27):

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. (SALEH, 2004, p.27).

Existem quatro pontos defendidos por esta licença:

1. Liberdade de execução do programa;
2. Liberdade de estudar e modificá-lo para um atendimento mais preciso das necessidades próprias ou de terceiros;
3. Direito de livre distribuição de cópias do programa original;
4. Liberdade de distribuição das versões modificadas.

No Brasil, o Governo Federal adotou a CC-GNU-GPL³⁰ que tenta adequar a GPL a realidade brasileira. No caso dessa licença, verificamos um forte viés de mercado, uma vez que o uso de

³⁰ Conforme própria definição do Governo, a CC-GNU-GPL acrescenta os meta-dados gerados pelo Creative Commons e a parte explicativa, chamada "Commons Deed", à Licença Pública Geral (GPL) da Free Software Foundation. Esta licença é a GPL oficial da FSF, traduzida em português. Para mais informações, acessar: < <http://www.softwarelivre.gov.br/Licencas/LicencaCcGplBr/view>>.

software livre pode gerar receitas a partir da prestação de serviços, oferta de garantia, de suporte técnico ou desenvolvimento de novas funcionalidades. As utilizações do *software* livre com foco em práticas empresariais são fortemente motivadas, desde que não prejudiquem nenhuma das liberdades acima citadas. GPL não significa gratuidade em si, mas apenas uma forma de garantia de liberdade.

Também temos a LGPL – Lesser General Public License (antes, Library General Public License) – diz respeito a uma licença que foca as bibliotecas de *software*. Essas bibliotecas são conjuntos de funções usadas na produção de programas. Elas visam evitar que o programador reescreva um código para tarefas comuns. Segundo Saleh (ibid., p.29) “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 licença BSD - Berkeley *Software* Distribution – é uma licença de *software* livre inicialmente criada para sistema de mesmo nome, mas outros sistemas também já são distribuídos sob essa licença. Segue as condições adotadas na GPL, mas acresce a condição de que as redistribuições do código-fonte deverão conter o *copyright* original de identificação do autor, além de que as distribuições binárias – onde o código-fonte não está presente – tenham o *copyright* na documentação. Também define que o nome do autor não deve ser usado como forma de endossar as versões modificadas.

Essa licença difere da GPL por não ter um compromisso com a liberdade, não apresentando caráter político ou social. Segundo Saleh (ibid., p.31), ela enquadra-se “[...] muito mais na filosofia do movimento do Código Aberto do que no movimento de *software* livre promovido pela FSF³¹. Como não há restrições, o *software* pode ser utilizado para qualquer finalidade, inclusive para incorporação em produtos proprietários”. São bastante utilizadas por empresas que têm modelos de concepção híbridos, onde produtos de OSS são usados no desenvolvimento de produtos proprietários.

³¹ Free Software Foundation. Para saber mais, acessar: <www.fsf.org>

3.4 UMA AVALIAÇÃO CRÍTICA DA POLÍTICA ESTRATÉGICA DO SETOR

Verificamos a inserção do *software* livre na agenda de discussões dos fazedores de política no Brasil. É inegável que o governo brasileiro enxerga no *software* livre uma realidade em contraponto ao modelo proprietário. No entanto, as discussões ainda estão apenas começando.

Na página *softwarelivre.gov.br*, está disponível um planejamento estratégico para implementação de *software* livre que elenca dezoito diretrizes do governo federal. São elas:

1. Priorizar soluções, programas e serviços baseados em *software* livre que promovam a otimização de recursos e investimentos em tecnologia da informação;
2. Priorizar a plataforma Web no desenvolvimento de sistemas e interfaces de usuários;
3. Adotar padrões abertos no desenvolvimento de tecnologia da informação e comunicação e o desenvolvimento multiplataforma de serviços e aplicativos;
4. Popularizar o uso do *software* livre;
5. Ampliar a malha de serviços prestados ao cidadão através de *software* livre;
6. Garantir ao cidadão o direito de acesso aos serviços públicos sem obrigá-lo a usar plataformas específicas;
7. Utilizar o *software* livre como base dos programas de inclusão digital;
8. Garantir a auditabilidade plena e a segurança dos sistemas, respeitando-se a legislação de sigilo e segurança;
9. Buscar a interoperabilidade com os sistemas legados;
10. Restringir o crescimento do legado baseado em tecnologia proprietária;
11. Realizar a migração gradativa dos sistemas proprietários;
12. Priorizar a aquisição de hardware compatível às plataformas livres;
13. Garantir a livre distribuição dos sistemas em *software* livre de forma colaborativa e voluntária;
14. Fortalecer e compartilhar as ações existentes de *software* livre dentro e fora do governo;
15. Incentivar e fomentar o mercado nacional a adotar novos modelos de negócios em tecnologia da informação e comunicação baseados em *software* livre;
16. Promover as condições para a mudança da cultura organizacional para adoção do *software* livre;

17. Promover capacitação/formação de servidores públicos para utilização de *software* livre;
18. Formular uma política nacional para o *software* livre.

Dentre os pontos colocados, alguns merecem uma maior atenção. A garantia de auditabilidade é, sem dúvida, algo louvável. Como já discutido em parte adequada desse trabalho, a transparência é algo fundamental como garantia da democracia. É importante esclarecer que o *software* livre em essência é a ferramenta mais adequada para essa premissa.

Outro ponto interessante é a visão de compartilhamento de experiências entre o governo e os demais agentes. A troca de conhecimento entre todos os envolvidos no processo é condição indispensável ao êxito do *software* livre. O fluxo não pode ocorrer de maneira unilateral, é preciso uma interação entre o governo, as ICTs, os empresários e também os usuários e mais uma vez a proposta do *software* livre privilegia essa interação.

O incentivo à adoção de modelos de negócio baseados em *software* livre mostra a preocupação e a relevância do *software* livre como opção estratégica. É fundamental dotar nossas firmas de capacidade competitiva nesse ambiente e, sem sombra de dúvidas, o governo é um agente capaz de definir esse vetor de desenvolvimento.

A proposta de mudança na cultura organizacional em direção a adoção de *software* livre, pontua a preocupação do governo em se valer de mecanismos mais amplos do que os tradicionais – fundos, subsídios, etc. É sem dúvida um desafio mudar as percepções de algo que ficou enraizado como “melhor”. A migração do *software* proprietário para o *software* livre no âmbito organizacional não é apenas uma mudança de caráter técnico é, em mesma importância, uma mudança de cultura.

Por fim, em relação as diretrizes estabelecidas, a formulação de uma política nacional de adoção de *software* livre deve ter em foco não só a esfera estatal, mas precisa assumir uma perspectiva mais ampla. Deverá ser contemplada em tal política a sinergia entre todos os participantes do ambiente econômico, nas mais diversas esferas. Não podemos incorrer nos erros passados assumindo que as coisas acontecem de maneira pontual. Em especial, na atual fase do capitalismo, todo processo de mudança exige a interação de todos os envolvidos, seja direta ou indiretamente.

Como ações de implementação desse plano, o governo lista alguns pontos interessantes. Discute a criação de uma campanha publicitária para dar visibilidade a questão do *software* livre. Sem dúvida, neste sentido a mídia pode ser um aliado na colocação de questões estratégicas, especialmente para o público mais geral.

Também se propõe a colocar o *software* livre como pauta nas reuniões da alta administração pública, confirmando sua visão estratégica desse modelo.

O Estado ainda faz uso de sua capacidade de consumir e financiar, se propondo a elaborar padrões de requerimentos favoráveis ao *software* livre a serem incluídos em processos de licitação e contratação da administração pública, além de favorecer o acesso fundos, linhas de financiamento e programas de fomento àqueles que utilizem *software* livre.

Mas as duas mais importantes propostas no nosso entendimento estão na criação de uma política de apoio à exportação de *software* livre e na articulação de uma rede de parceiros (dentro e fora do governo) para capacitação, pesquisa, desenvolvimento e testes em *software* livre. Essas duas iniciativas são complementadas pelo desenvolvimento de premiações para incentivo na criação de *software* livre.

É inexorável que um projeto de tal grandeza não pode ser feito e conduzido a partir de “canetadas” por parte do governo. A articulação com o todo da sociedade se faz indispensável para um projeto como o *software* livre. Este, como vetor estratégico, só alcançará êxito se os mecanismos institucionais forem arquitetados de maneira a envolver e facilitar a atuação nas mais diversas esferas.

Já foi elaborado um *Plano de Padronização de Ambiente e Migração para Software Livre* pelo Ministério do Planejamento, Orçamento e Gestão criando “procedimentos técnicos, metas, prazos, custos e atividades a serem desenvolvidas, segundo a realidade da instituição³²”, com a finalidade de promover a migração do sistema proprietário para *software* livre. Segundo esse plano:

O Governo Federal Brasileiro define, entre as diretrizes e princípios para gestão de Tecnologia da Informação, a utilização de *software* livre como

³² Citação retirada do documento *de Padronização de Ambiente e Migração para Software Livre*, vide referências.

opção estratégica. Tal iniciativa reflete o interesse na promoção da eficiência da Administração Pública, decorrente da independência tecnológica e de fornecedor, além da racionalização dos recursos aplicados, sem perda da qualidade dos serviços. Para tanto, torna-se fundamental o planejamento cuidadoso da transição para esse novo ambiente, considerando custos e riscos inerentes ao processo, de forma a permitir a realização de migrações gerencialmente estáveis, com garantia de interoperabilidade e salvaguarda das informações (BRASIL, 2005, p.4) .

Mais uma vez sinalizamos que a adoção de *software* livre não pode e não deve ser um mero mecanismo de ajuste fiscal, gerando redução dos gastos com pagamento de licença. Nem muito menos à independência de fornecedores deve se restringir a um aumento no número de distribuições dos *softwares* utilizados pelo governo. A proposta do *software* livre transborda esses aspectos se constituindo em uma aposta estratégica de redução das diferenças tecnológicas provenientes do antigo modo de apropriação intelectual.

A própria nomenclatura utilizada pelo Governo ainda não deixa claro as perspectivas que temos pela frente. Ainda não está claro se o uso do termo *software* livre é apenas uma forma de facilitar a divulgação da idéia ou se o foco é efetivamente o não pagamento de licenças de *software*. Em documentos oficiais, nas páginas mantidas pelo governo, em manchetes de jornais e outras formas de divulgação, observamos um foco no *software* livre. Como já demonstrado nesse trabalho, *software* livre e *software* de código aberto não são a mesma coisa. Todo *software* de código aberto é livre, mas nem todo *software* livre tem código aberto. Essa diferença está longe de ser irrelevante.

O incentivo e a adoção de *software* livre é por si só um movimento de mudança que merece ser reconhecido. No entanto, não devemos nos ater ao simples fato da liberdade de uso do *software* sem o pagamento de licenças. É indubitável a significativa economia que o Estado fará ao se livrar do pagamento das licenças dos *softwares* proprietários, mas isso é muito pequeno em relação a proposta trazida por esse movimento. O Brasil não deve estar apenas ligado a ótica do consumo de *software*, mas fundamentalmente inserido no contexto de produção. É necessário se discutir além dos mecanismos de adoção do *software* o caminho que se deverá trilhar para a inserção do Brasil na produção de *software*. É dentro desta proposta que o diferencial entre código aberto e *software* livre é de extrema relevância.

Como já explanado, um *software* pode ser livre sem necessariamente ser passível de modificações, ou seja, sua licença pode permitir a sua utilização sem o pagamento por isso,

mas não permitir o acesso a seu código-fonte. É preciso estabelecer propostas de incentivo ao *software* de código aberto, por este permitir uma inserção das firmas e dos agentes brasileiros no processo de construção dos *softwares*, representando uma real redução das diferenças tecnológicas entre o Brasil e os países centrais – *gap* tecnológico.

Apenas com a definição de um plano estratégico centrado no *software* de código aberto, poderemos usufruir as duas facetas desse modelo: liberdade de fornecedores associada ao não pagamento de licenças e redução do *gap* tecnológico a partir da inserção no processo produtivo de *software*.

Além disso, o *software* de código aberto representa em si uma mudança significativa na maneira como os agentes se apropriam pelos seus esforços de inovação. A partir dessa “janela de oportunidades”, o conhecimento deixa de ser aprisionado por pequenos grupos e passa a ser gerido e utilizado por um maior número de agentes. É nesta direção, que o conhecimento assume um caráter de *commons* e, dessa forma, a construção de uma arquitetura que possibilite essa perspectiva assume caráter imperativo. Como sinalizado por Lessig (2006) existe uma ameaça aos *commons* no *cyberspace* e, além disso, se estende aos códigos-fonte. Com sabe nessa análise que se torna indispensável um posicionamento político na construção dessa arquitetura de maneira a fortalecer e a expandir a produção dos *commons*. Neste sentido, entra em vigor o debate sobre as formas de apropriação intelectual e a maneira como a liberdade desse ser assegurada nesse novo ambiente.

Os mecanismo de proteção da propriedade intelectual precisam ser repensados em virtude de um aprisionamento das possibilidades de evolução do conhecimento. Os mecanismos tradicionais têm demonstrado limitação em atender às novas demandas. Exemplo disso, são as patentes, que foram criadas como uma maneira de recompensar aqueles que inovam e possibilitar a criação de novas inovações. No entanto, como apontado por Stiglitz (2007), as patentes têm representado apenas uma forma de geração de monopólio e que em alguns casos não limitam a produção de novos conhecimentos. Dessa forma, é imperativo criar uma alternativa a esse mecanismo. Segundo ele:

Há uma via alternativa para financiar e incentivar a investigação que, ao menos em alguns casos, poderiam ter resultados muito melhores que as patentes, tanto na forma de direcionar as inovações, como na maneira de assegurar que os benefícios desse conhecimento sejam repartidos o mais

amplamente possível: um fundo de premiação [...] que recompense àqueles que façam novas descobertas³³...(STIGLITZ, 2007).

No caso desse artigo, Stiglitz discute a produção de fármacos por parte das grandes corporações, sinalizando que elas tem como foco “onde o dinheiro está”, sem se preocupar com os ganhos sociais de suas inovações. Mas esse pensamento pode ser extrapolado para outras formas de produção.

Portanto, a proposta do *software* livre é interessante, mas limita a atuação dos agentes brasileiros, em primeira análise, ao uso do *software* tendo como foco a libertação do modelo proprietário. É necessário ir mais além, definindo estratégias produtivas que possibilitem uma maior inserção e o fortalecimento das firmas brasileiras que queiram apostar na produção de *software*.

³³ There is na alternative way of financing and incentivizing research that, al least, in some instances, could do a far better job than patents, both in directing innovation and ensuring that the benefits of that knowledge are enjoyed as widely as possible: a medical prize fund that would reward those discover cures and vaccines. (STIGLITZ, 2007)

4 CONCLUSÕES

A principal contribuição deste trabalho foi apontar o *software* de código aberto a partir de um *framework* institucionalista, como um movimento evolutivo do que alguns autores chamam de *capitalismo informacional*.

Defendendo a idéia de que é possível reduzir o *gap* tecnológico do Brasil em relação aos países centrais – em consonância com intenções de política tecnológica apresentadas pelo Governo Brasileiro – a partir da antecipação e aposta num movimento que começa a ganhar corpo, mas que já se apresenta como uma ruptura ao modelo tradicional de apropriação.

O *Software* de Código Aberto simboliza de maneira quase que precisa um movimento de proporções ainda maior: a inovação aberta. Esta proposta coloca em cheque os mecanismos de apropriação intelectual vigentes e propõe novas formas de auferir ganhos dentro do processo produtivo da informação.

Grandes corporações como IBM, Amazon, HP e outras já aportam grandes investimentos nesse sentido e o fazem não por se tratar de um *software* “grátis”, gerando uma redução nos seus custos de licenciamento. Estas empresas acreditam na capacidade técnica dessa proposta como substituta, sem qualquer perda, do modelo de *software* proprietário. É evidente que nenhuma empresa desse porte colocaria em risco sua operação se não tivesse certeza das possibilidades apresentadas pelo OSS.

Como abordado ao longo do trabalho, não se deve confundir o termo *free* ou *open source* como algo não passível de retornos financeiros. Estes retornos se apresentam de formas outras, em especial na substituição do *software produto* pelo *software serviço*. A viabilidade do negócio diante dessa nova forma de apropriação é possível desde que observado novas formas de gestão nesse tipo de produção informacional.

Também não devemos associar o *software* livre ao *software* de código aberto diretamente. O primeiro prima pela liberdade de uso, mas é negligente em questões de produção e viabilidade econômica diante da ótica capitalista. Só o segundo apresenta uma melhor relação com a esfera produtiva por possibilitar a abertura do código-fonte, elemento crucial para um melhor aproveitamento na produção de *software*.

O *Sourceforge*, o maior repositório de projetos em *open source* da Internet, conta com 150.112 projetos e mais de 1.600.000³⁴ participantes em projetos OSS nesse momento. Tais números relevam a dimensão dessa realidade que não pode e não deve ser ignorada pelo Brasil. *Softwares* como o Apache – servidor de *webpages* – já superam seus concorrentes fechados em larga vantagem (BENKLER, 2006, p.64).

Todas essas informações só confirmam a necessidade do Brasil de analisar e criar meios de inserção neste movimento, não por questões puramente ideológicas, mas, acima de tudo, por questões econômicas. Para tanto, é indispensável a elaboração de uma arquitetura de mecanismos institucionais que possibilitem a aproximação dos agentes e que desenvolva um ambiente de sinergia entre os participantes.

Diante de tudo que foi exposto, o *software* de código aberto representa uma grande possibilidade de redução do distanciamento tecnológico brasileiro em relação aos países desenvolvidos e a inserção do Brasil nessa proposta poderá acelerar o processo de desenvolvimento do país. O caminho para essa redução passa pela transversalidade do *software* que possibilita a alavancagem de outros setores da economia. Além disso, o *software* de código aberto possibilita que os usuários brasileiros possam assumir a postura de usuários inovadores. Essa mudança de conduta só é possível devido a inovação aberta, como apontado por Von Hippel, pois possibilita que o usuário seja o produtor e o consumidor da inovação.

Portanto, o presente trabalho não apontou caminhos conclusivos para a política de fomento ao *software*. Centrou os esforços na discussão de um cenário que cada dia mais está presente nos ambientes de alta tecnologia – como *software* e biotecnologia – e que apresenta uma nova possibilidade de inserção do Brasil na corrida pelo desenvolvimento tecnológico. A inovação aberta, tendo no OSS seu principal “exemplo aplicado”, trará novos horizontes de inserção tecnológica para todos os agentes que apostarem nessa proposta.

³⁴ Estatística disponível no site <<http://sourceforge.net/>>, no dia 06 jun. 2007.

REFERÊNCIAS

ARROYO, Cristiane S.; MERLO, Edgard M.; SIMÕES, André X. **A Economia do Software de Fonte Aberta: Razões que levam os desenvolvedores de *software* a participar das comunidades de fonte aberta.** São Paulo: USP, 2005. 30p.

BARAÑANO, Ana Maria. Gestão da Inovação Tecnológica: Estudo de Cinco PMEs Portuguesas. **Revista Brasileira de Inovação.** Rio de Janeiro: FINEP, v.4, n.1, p.57-95, jul./dez, 2004.

BENKLER, Yochai. **The Wealth of Networks: How Social Production Transforms Markets and Freedom.** New Haven: Yale University Press, 2006. 528p.

BRASIL. Decreto-lei n. 9279, de 14 de maio de 1996. **Regula direitos e obrigações relativos à propriedade industrial.** Disponível em: < www.planalto.gov.br/CCIVIL/Leis/L9279.htm>. Acesso em: 22 mai, 2007.

_____. Decreto-lei n. 9609, de 19 de fevereiro de 1998. **Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.** Disponível em: <<http://www.planalto.gov.br/ccivil/Leis/L9609.htm>>. Acesso em: 24 mai, 2007.

_____. Decreto-lei n. 9610, de 14 de maio de 1996. **Regula direitos e obrigações relativos à propriedade industrial.** Disponível em: www.cff.org.br/Legislação/Leis/lei_9279_96.htm. Acesso em: 22 mai, 2007.

_____. Ministério do Planejamento, Orçamento e Gestão. **Plano de Padronização de Ambiente e Migração para Software Livre.** 2005. Disponível em: < <https://www.governoeletronico.gov.br/anexos/plano-de-padronizacao-de-ambiente-e-migracao-para-software-livre>>. Acesso em: 06 jun, 2007.

_____. Projeto-lei nº.: 3.476, de novembro de 2004. **Dispõe sobre incentivos à inovação e à pesquisa científica e tecnológica no ambiente produtivo e dá outras providências: Lei de Inovação,** Brasília, 2004. Disponível em: < <http://www.camara.gov.br/sileg/integras/215493.pdf>>. Acesso em: 29 mai, 2007.

CASA CIVIL DA PRESIDÊNCIA DA REPÚBLICA. **Diretrizes de Política Industrial, Tecnológica e de Comércio Exterior.** Brasília, dez, 2004.

CASTRO, Ana Célia. Construindo Pontes: Inovações, Organizações e Estratégias como Abordagens Complementares. **Revista Brasileira de Inovação.** Rio de Janeiro: FINEP, v.3, n.2, p.449-473, jul./dez, 2004.

CHESBROUGH, Henry. Open Innovation: A new paradigm for understanding industrial innovation. In: DRUID TENTH ANNIVERSARY SUMMER CONFERENCE, 2005. **Anais...** Copenhagen, Denmark, 2005. p. 1-27. Disponível em: < <http://www.openinnovation.net/Book/NewParadigm/Chapters/01.pdf>>. Acesso em: 17 dez, 2006.

DI COSMO, Roberto. Eligiendo la herramienta legal correcta para proteger el software. **Novática**. Barcelona, n.163, p.21-23, mai.-jun, 2003. Disponível em: <<http://www.ati.es/novatica/2003/163/163-21.pdf>>. Acesso em: 16 dez, 2006.

FIGUEIREDO, Paulo N. Aprendizagem Tecnológica e Inovação Industrial em Economias Emergentes: uma Breve Contribuição para o Desenho e Implementação de Estudos Empíricos e Estratégias no Brasil. **Revista Brasileira de Inovação**. Rio de Janeiro: FINEP, v.3, n.2, p.323-361, jul./dez, 2004.

FREIRE, Emerson Freire; BRISOLLA, Sandra de Negraes. A Contribuição do Caráter “Transversal” do *Software* para a Política de Inovação. **Revista Brasileira de Inovação**. Rio de Janeiro: FINEP, v.4, n.1, p.97-128, jan./jun, 2005.

GROSSI, Paolo. **História da Propriedade e Outros Ensaio**s. Rio de Janeiro: Renovar, 2006. 146p.

HEXSEL, Roberto A. **Software Livre: Propostas de Ações de Governo para Incentivar o Uso de Software Livre**. 2002. 48 p. Relatório Técnico - Departamento de Informática. Curitiba: Universidade Federal do Paraná, 2002.

KROGH, Georg von; HIPPEL, Eric von. Special issue on open source *software* development. **Research Policy**, n.32, 1.149-1.157, 2003a.

_____. Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. **Organizations Science**, v.14, n.2, 209-223, mar.-abr, 2003b.

LESSIG, Lawrence. **CODE: version 2.0**. Nova York: Basic Books, 2006, 410p. Disponível em: <<http://pdf.codev2.cc/Lessig-Codev2.pdf>>. Acesso em: 07 mai, 2007.

MEDINA, Humberto. Governo Federal poupou R\$ 28 mi com software livre. **Folha de São Paulo**, São Paulo, 20 mar, 2005. Dinheiro. Disponível em <<http://www1.folha.uol.com.br/folha/dinheiro/ult91u94554.shtml>>. Acesso em: 02 mai, 2007.

NORTH, Douglas C. Institutions. **Journal of Economic Perspectives**. v.5, n.1, p.97-112, winter.1991. Disponível em: <http://www-econ.stanford.edu/academics/Greif_228_2006/North%20JEP%2019911.pdf>. Acesso em: 18 abr, 2007.

ROSELINO, José Eduardo. **A indústria de Software: o “modelo brasileiro” em perspectiva comparada**. 2006. 236 f., Tese (Doutorado em Ciências Econômicas) – Instituto de Economia – UNICAMP, Campinas, 2006.

SALEH, Amir Mostafa. **Adoção de Tecnologia: Um estudo sobre o uso de software livre nas empresas**. 2004, 149 f. Dissertação (Mestrado em Administração) – Faculdade de Economia, Administração e Contabilidade, USP, São Paulo, 2004.

SAMPAIO, Sérgio Eduardo Ketelhute. **O desenvolvimento da aglomeração produtiva de software de Curitiba**. 2006. 172 f. Dissertação (Mestrado em Desenvolvimento Econômico) – Departamento de Economia – UFPR, Curitiba, 2006.

SEPPÄ, Arto. **Open Source in Finnish Software Companies**. Helsinki – Finlândia: The Research Institute of the Finnish Economy, 2006. 36p. (Keskusteluiheita, Discussion Papers, n.1.002).

STIGLITZ, Joseph E.. **Prizes, Not Patents**. Disponível em: <<http://www.project-syndicate.org/commentary/stiglitz81/English>>. Acesso em: 05 jun, 2007.

STOKES, Donald E. **O Quadrante de Pasteur: a ciência básica e a inovação tecnológica**. Campinas: Unicamp, 2005. 246p.

VON HIPPEL, Eric. Innovation by User Communities: Learning from Open Source Software. **MIT Sloan Management Review**, v.42, n.04, p. 82-86, jul, 2004. Disponível em: <<http://sloanreview.mit.edu/wsj/insight/pdfs/4248.pdf>>. Acesso em: 17 dez, 2006.

WILLIAMSON, Oliver E.,. The New Institucional Economics: Taking Stock, Looking Ahead. **Journal of Economic Literature**. v.38, p.595-613, set. 2000. Disponível em: <www.uv.es/~fatas/ipei/williamson_jel00.pdf>. Acesso em: 28 mar, 2007.