

UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE GEOCIÊNCIAS

PÓS-GRADUAÇÃO EM POLÍTICA CIENTÍFICA E TECNOLÓGICA

ANA MARIA CARNEIRO

Proteção de ativos na indústria de software: estratégias e tendências de Propriedade Intelectual

Tese apresentada ao Instituto de Geociências como parte dos requisitos para obtenção do título de Doutor em Política Científica e Tecnológica.

Orientador: Profa. Dra. Maria Beatriz Machado Bonacelli

CAMPINAS - SÃO PAULO

Agosto - 2007

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA CENTRAL DA UNICAMP**

Bibliotecária: Helena Joana Flipsen – CRB-8ª / 5283

Carneiro, Ana Maria, 1976-.

C215p Proteção de ativos na indústria de software: estratégias e tendências de propriedade intelectual/ Ana Maria Carneiro. - Campinas, SP: [s.n.], 2007.

Orientadora: Maria Beatriz Machado Bonacelli.

Tese (Doutorado) - Universidade Estadual de Campinas, Instituto de Geociências.

1. Propriedade Intelectual. 2. Software - Patentes. 3. Indústria de Software. I. Bonacelli, Maria Beatriz Machado. II. Universidade Estadual de Campinas. Instituto de Geociências. III. Título.

Título e subtítulo em inglês: Asset protection in the software industry: intellectual property strategies and trends.

Palavras-chave em inglês (Keywords): Intellectual property, Software - Patents, Computer software industry.

Titulação: Doutor em Política Científica e Tecnológica.

Banca examinadora: Paulo Bastos Tigre, Sérgio Luiz Monteiro Salles Filho, Wilson Suzigan, Sérgio Medeiros Paulino de Carvalho.

Data da Defesa: 28-08-2007.

Programa de Pós-Graduação em Política Científica e Tecnológica.



UNICAMP

**UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE GEOCIÊNCIAS
PÓS-GRADUAÇÃO EM POLÍTICA CIENTÍFICA E
TECNOLÓGICA**

AUTORA: ANA MARIA CARNEIRO

**PROTEÇÃO DE ATIVOS NA INDÚSTRIA DE SOFTWARE: ESTRATÉGIAS E
TENDÊNCIAS DE PROPRIEDADE INTELECTUAL**

ORIENTADORA: Profa. Dra. Maria Beatriz Machado Bonacelli

Aprovada em: ____/____/____

EXAMINADORES:

Profa. Dra. Maria Beatriz Machado Bonacelli

_____ - **Presidente**

Prof. Dr. Paulo Bastos Tigre - IE/UFRJ

Prof. Dr. Sergio Luiz Monteiro Salles Filho - IG/UNICAMP

Prof. Dr. Wilson Suzigan - IG/UNICAMP

Prof. Dr. Sergio Medeiros Paulino de Carvalho - INPI

Campinas, 28 de agosto de 2007

Dedico este trabalho a meus pais,
Manoel e Maria.

Agradecimentos

Primeiro de tudo, agradeço imensamente a minha orientadora Bia, que foi muito mais que orientadora, foi amiga, companheira, chefe, sempre compreensiva, paciente e animadora. E principalmente por ter aceitado o desafio de orientar alguém que a toda hora vinha com uma idéia diferente de tese (nas minhas contas foram 3 ou 4). Ah essas aquarianas...

E aproveitando o embalo, agradeço imensamente aos amigos do peito do GEOPI. Ao Claudenício, amigo de todas as horas, que fez a ponte para a entrada no início. À Sônia Paulino, primeira chefe no tempo das ONGs e sempre grande amiga. Ao Sergio Paulino, grande inspiração intelectual e conselheiro para todos os tipos de assunto. Ao Mauro, pelas boas discussões e pelas saídas tipo Barão Geraldo Profundo. Ao Sergio Salles, chefinho querido pela insistência para voltar para casa. À Adriana Bin pela preciosa amizade, carinho e cuidado, pelo socorro nas horas em que precisava de energia. À Paula, companheira do tempo das ONGs, que também fez o papel de “bom filho que à casa torna” pelas ajudas bibliográficas e pelos insistentes convites para baladas, o que me animava cada vez mais a terminar logo a tese. Aos amigos e amigas bolsistas do projeto monitoramento tecnológico Ana Serino, Paola, Lilian, Bruna que seguraram as pontas no meu afastamento para o final da redação da tese e, como se isso não bastasse, botaram também a mão na massa no exercício de monitoramento feito para a tese. Também à equipe da Elabora André, Francisco, Juliana, Douglas e Eduardo. À Sonia Tilkian e ao David, sempre queridos e prestativos. Ao par Rui e Solange, junto com a Manu, pela ótima convivência e alegria. Ao Tuca, Zé Maria, Benami, Cristiane Quental e Celso Lage por ampliarem a biodiversidade do ambiente. Aos novos membros do GEOPI, que chegaram quando eu estava fora, Paule, Fernanda, Fernando, Edilson, Marcos, Rodrigo, Carol, Josi e Rafa e Fabrício (estes dois últimos de amizade de longa data) pela responsabilidade de continuar com a história. E como é de praxe no Geopi, uma mão lava a outra e as duas juntas dão um banho...

Aos professores do DPCT pelos ensinamentos multidisciplinares. À Adriana, Val, Edinalva e demais técnicos do IG de serviço nota 1000 plus +.

Agradeço à CAPES e ao CNPq pelo apoio financeiro no começo e no fim do doutorado.

Aos colegas da SOFTEX pelos tempos de trabalho e aprendizado no Observatório. Em especial ao Gian e à Carol pela solidariedade, amizade, companheirismo, paciência com meu “gênio forte”, agora estamos em outras praias, mas nossos barquinhos continuam navegando em mares próximos. Também agradeço aos estagiários com que tive o prazer de trabalhar: Tainá, Taise, Ana Serino e Murilo.

Aos amigos especialistas que ajudaram na discussão da tese em algum momento: Cássia Mendes em PI; Gian e Roselino na indústria de software; De Lucca, Angela e Rogério em TI.

Aos Sergios, membros da banca de qualificação e da defesa. Aos professores Suzigan e Paulo Tigre pela participação na banca da defesa.

À “equipe médica” – Cida, Carlos, Tânia, Claudia, Josés. À Elizangela e marido pela ajuda com a casa.

Às meninas do EMI pela diversão garantida a cada 15 dias, com a boa companhia, pautas para discussão e eventos lúdicos.

À Claudete e Martha pela amizade de longa data que sobrevive a distâncias e preenche aquele vazio que a gente sente de vez em quando...

Aos amigos que trabalham e moram em outras paragens, mas cuja amizade continua inabalável: Rodrigo Pups, Rogério Robinho, Ana (Maria) Pina, Marilis, Luciana, Goretti, Ronize e Andréia.

Às famílias que consegui em Campinas: Angela, Astor e Sandra; Eduardo, Tati, Atílio e Eorly e toda família; Mario e Celia – pelo carinho, conforto, aconchego em todas as horas, pelas alegrias, pelos momentos não tão alegres, afinal o que passamos como famílias. Em especial à Angela, amiga do peito, pelos tantos papéis que tem desempenhado na minha vida, cabelereira, conselheira, emprestadora de livros, etc. etc.

À minha família que sempre mora no meu coração, mesmo estando em três das cinco regiões do país: a meu pai e a minha mãe por todos os sacrifícios que fizeram para que uma filha virasse doutora, a minha irmã Ana Lúcia e a meu irmão Manoel sempre amorosos e ao Miro, Tânia e Decon, agora mais próximos.

Aos meus amarecos, Michel, Catarata e Spike pela convivência diária. Ao Michel que várias vezes

tentou me ajudar a terminar a tese fjlçekur,vmrue,r,fekrfj2147 (sai Michel!) pisando no teclado e querendo atenção.

“A Menina que roubava livros – última linha
Odiei as palavras e as amei,
e espero tê-las usado direito”

(Markus Zusak)

Sumário

Introdução	1
1. Proteção da propriedade intelectual e apropriação dos resultados da atividade inovativa	11
1.1 Regimes tecnológicos e de apropriabilidade.....	11
1.1.1 Funções da propriedade intelectual	19
1.1.2 Uso estratégico da propriedade intelectual	24
1.2 Apropriação na Economia do Conhecimento	33
1.2.1 Novas variáveis no jogo da apropriação	35
2. A Evolução da Indústria de Software Pré-Internet	41
2.1 Evolução da indústria de software e de seus modelos de negócio até 1994	44
2.1.1 Software como atividade complementar ao hardware	45
2.1.2 Software customizado e primórdios do software-produto	47
2.1.3 A revolução do PC: hardware torna-se commodity para o software pacote de massa.....	50
2.2 Sistematização dos modelos de negócio da indústria de software.....	56
2.3 A formação da proteção jurídica da propriedade intelectual para software.....	61
2.3.1 Copyright vs. Patente?	62
2.3.2 Apropriação por modelo de negócio	72
3. A indústria de Software Pós Internet.....	85
3.1 Software torna-se commodity e suporte para serviços	86
3.1.1 A indústria de software se consolida internacionalmente	88
3.1.2 A bolha da Internet e o crescimento do mercado de serviços	89
3.1.3 Software livre e de Código Aberto: da “economia da dívida” ao mainstream da indústria	99
3.1.4 Web 2.0 e Serviços na web	114
3.2 A corrida patentária e os custos de transação	118
3.2.1 A corrida de patentes de software exógena à indústria de software	118
3.2.2 A corrida patentária endógena à indústria de software	123
3.3 Propriedade Intelectual e os modelos de negócio - novos e recauchutados – da indústria de software	127
4. Propriedade intelectual e reuso de software	135
4.1 Engenharia de Software baseada em componentes e economias de escopo.....	139
4.1.1 Cadeia de valor da componentização	143
4.1.2 A oferta de componentes	148
4.2 Proteção da Propriedade Intelectual e custos de transação no desenvolvimento com componentes	153
5. Considerações Finais.....	165
ANEXO 1 – Iniciativas para lidar com a corrida patentária	175
Anexo 2 - Exercício de monitoramento tecnológico	182

Índice de gráficos

<i>Gráfico 2.1 - Participação no mercado dos Sistemas Operacionais, em 1998</i>	59
<i>Gráfico 3.1 – Valor das fusões e aquisições nas indústrias de TICs e outras indústrias, 1995</i>	92
<i>Gráfico 3.3 – Faturamento da IBM por segmento de mercado, 1992-2002</i>	96
<i>Gráfico 3.4 – Patentes concedidas nos EUA (total e patentes de software) – 1971 - 2006</i>	120
<i>Gráfico 3.5 - Distribuição das patentes concedidas das empresas selecionadas (1976 – 2004)</i>	124
<i>Gráfico 3.6 - Distribuição das patentes concedidas e depositadas das empresas selecionadas segundo posicionamento quanto ao patenteamento de software em 1994 (1976 – 2006)</i>	125
<i>Gráfico 3.7 – Patentes depositadas de empresas selecionadas (1998 – 2005)</i>	126
<i>Gráfico 4.1 – Estimativas sobre o mercado mundial de componentes (1999 – 2005)</i>	149

Índice de figuras

<i>Figura 1.1 - Estratégias tecnológicas em vários regimes tecnológicos</i>	18
<i>Figura 1.2 - Comportamento das empresas em oportunidades pervasivas e base de conhecimento complexa</i>	19
<i>Figura 2.1 – Regimes tecnológicos na indústria de software</i>	42
<i>Figura 2.2 – Desenvolvimento dos modelos de negócio nos regimes tecnológicos da indústria de software</i>	43
<i>Figura 2.3 – Esquema dos modelos de negócio na indústria de software</i>	59
<i>Figura 3.1 – Modelos de negócio da indústria de software ordenados segundo grau de especificidade e nível de apropriabilidade via código fechado</i>	113
<i>Figura 3.2 – Faturamento da propaganda on-line nos EUA em milhões de dólares, 1997- 2005</i>	115
<i>Figura 4.1 – Trajetória de reuso de software transversal aos regimes tecnológicos</i>	135
<i>Figura 4.2 – Cadeia de valor da indústria de software com ESBC madura</i>	145

Índice de Quadros

<i>Quadro 2.1 – Características dos principais modelos de negócio da indústria de software</i>	60
<i>Quadro 2.2 - Principais formas de apropriação dos resultados da atividade econômica dos principais modelos de negócio da indústria de software</i>	83
<i>Quadro 3.1 – Licenças de Software de Código Aberto ordenadas por categorias</i>	104
<i>Quadro 3.2 - Relações entre modelos de negócios específicos para SL/CA e da indústria de software</i>	106
<i>Quadro 3.3 - Alterações nos modelos da indústria de software</i>	128
<i>Quadro 3.4 - Novos modelos da indústria de software</i>	130
<i>Quadro 3.5 - Principais formas de apropriação dos resultados da atividade econômica dos principais modelos de negócio da indústria de software</i>	132
<i>Quadro 4.1 – Valor e detalhamento dos tipos de licença do componente Leadtools Document Imaging v15.0</i>	156

<i>Quadro 5.1 – Principais características e modelos de negócio dos regimes tecnológicos da Indústria de Software</i>	169
---	-----

Índice de tabelas

<i>Tabela 3.1 - Receitas das Principais Ofertantes, 2003-2004 (em US\$ bilhões)</i>	98
<i>Tabela 3.2 - Propaganda online nos EUA por categoria, 2000-2004</i>	115
<i>Tabela 3.3 – Patentes concedidas para aplicações de software por indústria (1994-1997)</i>	121
<i>Tabela 4.1 – Distribuição do número de sites e componentes segundo tipo de ofertante em 1999, 2000 e 2006</i>	151

Siglas e Abreviaturas

ADP: Automatic Data Processing
ADR: Applied Data Research
AOL: America OnLine
APIs: Application Programming Interface
ASP: Application Service Provider (provedor de serviço de aplicação)
BPO: Business Process Outsourcing
BSD: Berkeley Software Distribution
CA: Computer Associates
CAFC: Corte de Apelações do Circuito Federal
CBMA: Computer and Business Equipment Manufacturing Association
CenPRA: Centro de Pesquisas Renato Archer
CEO: Chief Executive Officer
CMMI: Capability Maturity Model Integration
CNAE: Classificação Nacional de Atividades Econômicas
CPU: Unidade Central de Processamento
CSC: Computer Sciences
CUP: Convenção da União de Paris
CUST: Produto Customizável
DEC: Digital Equipment Corporation
DIFELE: Divisão de Patentes de Física, Eletrônica e Eletricidade
DIRPA: Diretoria de Patentes
DPI: Direitos de Propriedade Intelectual
EDS: Electronic Data Systems
EFF: Electronic Frontier Foundation
EMB: Embarcado
EPO: Escritório Europeu de Patentes
EPR: Exxon Production Research Company
ERP: Software de Sistema Integrado de Gestão Empresarial
ESBC: Engenharia de Software Baseada em Componentes
FFII: Foundation for a Free Information Infrastructure
FSF: Free Software Foundation
GATT: Acordo Geral sobre Tarifas e Comércio
GE: General Electric Company
GEOPI: Grupo de Estudos sobre Organização da Pesquisa e da Inovação
GPL: General Public License
HP: Hewlett-Packard
IDC: International Data Corporation
INPI: Instituto Nacional de Propriedade Industrial
IP: Internet Protocol
IPC: Classificação Internacional de Patentes
ISVs: Independent Software Vendors (primeiros vendedores de software independentes)
ITO: Information Technology Outsourcing
LPF: League for Programming Freedom

MIT: Massachusetts Institute of Technology
NASA: National Aeronautics and Space Administration
NASDAQ: North American Securities Dealers Automated Quotation System
NDA: Non Disclosure Agreements
OIN: Open Invention Network
OMC: Organização Mundial do Comércio
OMPI: Organização Mundial de Propriedade Intelectual
OSDL: Open Source Development Labs
OSI: Open Source Initiative
OTAN: Organização do Tratado do Atlântico Norte
P&D: Pesquisa e Desenvolvimento
PAC: Pacote
PC: Computador Pessoal
PI: Propriedade Intelectual
PMEs: Pequenas e Médias Empresas
PUBPAT: Public Patent Foundation
PWC: PricewaterhouseCoopers
RCA: Radio Corporation of America
SAP AG: Systeme, Anwendungen und Produkte - Sistemas, Aplicações e Produtos
SaS: Software as a Service
SAV: Serviço de alto valor
SBV: Serviço de baixo valor
SEI: Instituto de Engenharia de Software da Universidade Carnegie Mellon
SIC: Standard International Classification (classificação industrial internacional)
SL/CA: Software Livre e de Código Aberto
SOA: Arquitetura Orientada a Serviços
SOFTEX: Associação para Promoção da Excelência do Software Brasileiro
SPI: Software Patent Institute
SWOT – Strengths, Weaknesses, Opportunities, Threats
TCP-IP: Transmission Control Protocol / Internet Protocol
TI: Tecnologia de Informação
TICs: tecnologias de informação e comunicação
TRIPs: Acordo sobre Aspectos de Propriedade Intelectual Relacionados ao Comércio
USPTO: Escritório Norte-Americano de Marcas e Patentes



UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE GEOCIÊNCIAS

PÓS-GRADUAÇÃO EM POLÍTICA CIENTÍFICA E TECNOLÓGICA

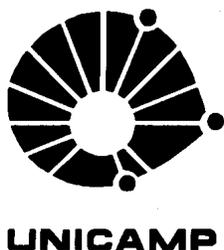
Proteção de ativos na indústria de software: estratégias e tendências de propriedade intelectual

RESUMO

Tese de Doutorado

Ana Maria Carneiro

A tese trata da apropriação dos resultados das inovações geradas na indústria de software e da proteção dos ativos tecnológicos por parte de suas empresas, no contexto do aumento da importância do conhecimento na geração de riqueza, à medida que a economia tem se desmaterializado, e do aumento da importância da propriedade intelectual na apropriação dos retornos do esforço inovativo. O software representa um dos ícones da Economia do Conhecimento, tanto devido à sua participação direta na composição da riqueza nas nações como por sua pervasividade e transversalidade que dinamiza as outras indústrias. Para isso, analisa-se a co-evolução da indústria de software e do processo de regulação da proteção da propriedade intelectual em torno de três regimes tecnológicos relevantes na indústria de software, cada qual formado por um peso diferente entre hardware, software e serviços: grandes computadores (*mainframes*), computador pessoal e serviços na web. Para uma indústria tão heterogênea, certamente não é possível apenas uma solução única em termos de instrumentos para proteção dos direitos de propriedade intelectual, sejam formais ou informais. Dada a evolução histórica da indústria de software, analisam-se como as empresas procuraram proteger seus heterogêneos ativos intangíveis, sendo que para cada modelo de negócio há uma necessidade diferente de proteção, considerando ainda que os mecanismos de proteção variam também de acordo com o país. Na década de 90, assistiu-se a dois movimentos aparentemente contraditórios. Por um lado, ao fortalecimento dos direitos de propriedade intelectual, como o grande crescimento de patentes de software, um tipo de instrumento que até então não era muito usual entre as empresas de software. Por outro lado, no mesmo período, o software livre e de código aberto (SL/CA), que é tanto fruto quanto impulsionador da migração da indústria de software para os modelos de negócio baseados em serviços, ganhou corpo e introduziu formas alternativas de desenvolvimento de software e de disponibilização e proteção dos conhecimentos gerados, baseadas em licenças. Desta forma são analisadas as formas de apropriação utilizadas pelas empresas contextualizadas nos respectivos modelos de negócio que vêm sendo estabelecidos da indústria.



UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE GEOCIÊNCIAS

PÓS-GRADUAÇÃO EM POLÍTICA CIENTÍFICA E TECNOLÓGICA

Proteção de ativos na indústria de software: estratégias e tendências de propriedade intelectual

ABSTRACT

DOCTORATE THESIS

Ana Maria Carneiro

The thesis concerns the appropriation of the results of innovations generated in software industry and the protection of the technological assets by the firms, in the context of the raising importance of knowledge to value generation with the economy dematerialization and also the raising importance of intellectual property in the appropriation of the innovation efforts returns. Software represents one of the icons of Knowledge Economy, by its direct contribution in nation's value generation, but also by its pervasiveness and its transversely character that helps other industries dynamics. It analyze the co-evolution of software industry and of the process of regulation of intellectual property protection focusing three relevant technological regimes of this industry, each one with a different balance between hardware, software and services: big computers (mainframes), personal computers and web services. For such a heterogeneous industry, it is certainly not possible just one solution concerning tools for the protection of intellectual property rights, being them formal or informal. Given the historical evolution of software industry, it analysis how firms tried to protect their intangible assets, considering that each business model has its own needs in terms of protection and that the protection mechanisms are diverse in different countries. At the 90th decade, two apparently contradictory movements appeared. By one side, the strengthening of intellectual property rights, with the raising of software patents, not so usual since then at software firms. In the other hand, at the same period, the free and open code software, that is a result and a stimulus of software industry migration to service based business models, reinforced itself and introduced alternatives ways of software development and of making available and protected the generated knowledge, based on licenses. In this way, the appropriation forms used by firms contextualized in respective business models that have been established in industry are analyzed.

INTRODUÇÃO

Esta tese trata da apropriação dos resultados das inovações geradas na indústria de software e da proteção dos ativos tecnológicos por parte de suas empresas. Analisa a evolução e as tendências de proteção da propriedade intelectual, assim como as estratégias utilizadas pelas empresas da indústria de software nos diferentes modelos de negócio.

A discussão da propriedade intelectual na indústria de software tem como pano de fundo dois fenômenos interligados: o aumento da importância do conhecimento como gerador de valor na economia e o conseqüente aumento da importância da propriedade intelectual como uma das formas de apropriação deste tipo de ativo.

O tema da tese situa-se, então, no contexto das transformações da economia mundial, no qual o conhecimento torna-se um ativo valorizado cada vez mais pelas empresas para fazer frente ao processo de competição. O desenvolvimento econômico tem passado cada vez mais pela capacidade dos atores econômicos gerarem, apropriarem-se e aplicarem o conhecimento na geração e distribuição de riquezas, tendo estas cada vez mais o caráter de bem intangível. Disso resulta a própria denominação dada ao período histórico atual, Economia baseada no Conhecimento ou Era da Informação (Castells, 1999)¹.

O conhecimento como ativo altamente valorizado no período atual possui, então, duas grandes dimensões: a dimensão codificada e a dimensão tácita. Esta diferença é importante no que se refere à valorização e gestão dos ativos intangíveis: a dimensão tácita é necessária para o aprendizado e é difícil de ser transferida ou comercializada, o que, por um lado, dificulta sua proteção por mecanismos jurídicos (que necessitam de registro codificado) e, por outro, por não ser encontrada facilmente no mercado, pode tornar-se fonte de vantagem competitiva (Teece, 1986).

¹ O conceito de Economia da Informação é menos abrangente que Economia do Conhecimento, pois é centrado em setores produtores de informações, deixando de lado os demais setores que também são modificados pelo uso intensivo de informações. Além disso, o conceito de conhecimento difere do de informação, pois este último refere-se aos dados codificados, enquanto o conhecimento envolve, além da dimensão codificada, a dimensão tácita.

Trata-se, portanto, de uma economia tecnologicamente dinâmica, amparada em bens intangíveis com um lastro que não corresponde ao tamanho e volume de negócio, que ainda está sendo compreendida. Como entender que o Google valha bilhões de dólares e tenha poucos bens físicos frente a outros empreendimentos que valem menos e têm bilhões investidos em bens físicos, como terras, máquinas, equipamentos, prédios etc?

A própria introdução da Tecnologia de Informação (TI) tornou a informação vital como recurso e como fator de produção à medida que mais e mais indústrias têm sua base na informação e que bens intangíveis e serviços proliferam na economia. TI tem tido impacto significativo nos empregos, na formação de capital, nas habilidades humanas e na criação da riqueza. Desta forma, a introdução de TI acelerou o crescimento dos bens intangíveis e dos serviços na economia. E também possibilitou o aumento da capacidade de codificação do conhecimento.

Na Economia do Conhecimento, a mudança na forma de criação e distribuição da riqueza, com a valorização maior dos ativos intangíveis do que dos tangíveis, coloca desafios relativos à proteção devido às diferenças entre os dois tipos de ativos. Segundo Teece (2000), os ativos intangíveis do conhecimento diferem dos ativos tangíveis numa série de requisitos: publicidade, depreciação, custos de transferência, direitos de propriedade e mecanismos de garantia destes direitos. Uma das mais importantes diferenças é que os ativos tangíveis são geralmente mais bem protegidos, pois é fácil definir sua propriedade bem como suas fronteiras. Já os bens intangíveis são de difícil definição e delimitação de suas fronteiras, pois para isso necessitam ser codificados, o que é um processo com limites.

Assim sendo, assiste-se à crescente valorização dos ativos intangíveis em relação aos tangíveis, carregando aqueles a maior parte do valor agregado e muitas vezes servindo estes últimos de mero suporte (Buainain *et al*, 2005). Como será visto ao longo da tese, a própria história da indústria de software mostra a crescente valorização do software em relação ao hardware: o software nasce como atividade complementar à produção do hardware, torna-se a principal fonte de valor quando é transformado em produto (e o hardware em *commodity*), até, por fim, tornar-se também atividade suporte para o oferecimento de serviços, portanto também *commodity*, na era dos serviços oferecidos via web (*web services*).

Além disso, as idéias não precisam estar incorporadas em produtos ou serviços para ter valor, pois podem ser licenciadas, vendidas ou trocadas no formato de direitos de propriedade intelectual. Desta forma, à medida que a economia se desmaterializa, cresce a importância da propriedade intelectual como instituição que facilita o controle, a valorização econômica e a circulação de ativos baseados em inovações.

Segundo alguns autores, vivemos uma Era Pró-Patente (Tang *et al.*, 2001), o que seria comprovado, por um lado, pelo aumento das solicitações de patentes. As empresas estão aproveitando o aumento da capacidade de codificação do conhecimento e a expansão dos estatutos legais de propriedade legal, resultando em uma corrida patentária como tentativa de compensar os crescentes gastos com pesquisa e desenvolvimento (P&D) e apropriá-los sob esta forma jurídica, considerada como uma das mais fortes. Uma das características desta corrida é o aumento da privatização dos domínios e atividades do conhecimento que antes eram públicos (patenteamento de programas de computador, algoritmos matemáticos e genes humanos) (Coriat e Orsi, 2002). Por outro lado, ganharam maior visibilidade as reações das empresas e movimentos contrários ao patenteamento das invenções.

Em consequência, a gestão dos ativos intangíveis ganha maior relevância, sendo que a melhor proteção é dada não pelo máximo de registros possíveis de direitos de propriedade intelectual, mas pela gestão eficiente dos ativos (passíveis de registro ou não) (Buainain e Carvalho, 2000). Assim, a gestão da propriedade intelectual entra na pauta das questões estratégicas das empresas, deixando de ser um assunto meramente legal.

O objeto de análise desta tese é a proteção dos ativos tecnológicos e da apropriação dos resultados econômicos da inovação na indústria de software², entendida aqui como atividade econômica de produção e comercialização de produtos e serviços relacionados ao software por parte de empresas que têm estes tipos de atividades como principal fonte de receita. Entretanto, o software tem um caráter pervasivo na geração da riqueza atual. Além das empresas de software

² A denominação indústria de software segue o entendimento de indústria como atividade econômica, pois as atividades de software são denominadas setor de software, sendo que o termo indústria é reservado apenas para as indústrias de manufatura e indústrias extrativas nas classificações oficiais (*Standard International Classification – SIC e Classificação Nacional de Atividades Econômicas – CNAE*).

propriamente, muitas outras empresas produzem software como parte de suas atividades, o que não pode ser ignorado na discussão sobre propriedade intelectual. Desta forma, nos momentos oportunos serão consideradas também as demais empresas de produção de equipamentos de informática e comunicação, bem como as empresas de serviços de comunicação que formam o conjunto das tecnologias de informação e comunicação (TICs)³. Eventualmente são consideradas ainda outras empresas de outros setores que têm no software um importante insumo produtivo.

O objetivo desta tese é analisar as tendências de proteção da propriedade intelectual na indústria de software e as estratégias de apropriação dos resultados da inovação. São objetivos secundários:

1. analisar a co-evolução do desenvolvimento tecnológico e das tendências de proteção do conhecimento na indústria de software. Desta forma, procura-se mostrar a co-evolução dos regimes tecnológicos da indústria de software e dos regimes de apropriabilidade nos diferentes modelos de negócios;
2. analisar as implicações das tendências de proteção identificadas na trajetória tecnológica do reuso, com foco na tecnologia de componentes de software;

A tese procura trabalhar conjuntamente aspectos jurídicos, tecnológicos e econômicos para identificar e discutir as estratégias de apropriação. Algumas hipóteses de trabalho são:

- as formas predominantes de proteção em cada modelo de negócio tendem a variar, afetando a atração de novos investimentos e o acesso aos mercados, criando diferentes barreiras. Foram utilizados os modelos de negócios concebidos na pesquisa realizada pela SOFTEX em 2002 sobre a indústria brasileira de software numa perspectiva comparada com as indústrias chinesa e indiana (SOFTEX, 2003). Estes modelos foram atualizados e complementados com outras fontes, como o trabalho de Roselino (2006);
- a discussão sobre Proteção de PI (propriedade intelectual) na indústria de software tem

³ Em algumas passagens será utilizada a denominação de indústria de computadores ou indústria de TI, o que envolve hardware, software e serviços relacionados.

sido focada na proteção dos elementos codificáveis, isto é, na discussão sobre se o software deve ser protegido por copyright ou por patente. Assim, muita atenção foi dispensada na discussão sobre qual a forma mais adequada de proteção e pouca na análise sobre como as empresas usam os diferentes tipos de PI e outras formas de proteção não passíveis de registro. Assim, procurou-se ir além e observar o conjunto das estratégias de apropriação das empresas contextualizadas nos regimes tecnológicos e nos modelos de negócio;

- o fortalecimento dos DPIs, visível na corrida ao patenteamento que ocorreu e ainda ocorre na indústria de software, tem implicado no aumento dos custos de transação, o que tem tido impactos negativos na trajetória tecnológica de reutilização de partes de código (componentes de software), o que permitiria, por sua vez, economias de escopo e aumento da produtividade.

Para tratar do tema em questão, realizou-se, inicialmente, uma revisão e releitura de cunho mais conceitual da bibliografia sobre as temáticas envolvidas com os assuntos da tese, bibliografia que é fortemente multidisciplinar, incluindo pelo menos quatro disciplinas: direito, computação, administração/gestão e economia. Atenção também foi dada para aqueles estudos empíricos que utilizaram as patentes de software, complementados com informações sobre as empresas detentoras. Em segundo lugar, procurou-se conhecer a opinião dos agentes econômicos sobre o assunto, manifestada em conferências organizadas por escritórios nacionais de patentes, sites, notícias da mídia especializada, entre outros.

Em terceiro lugar, a tese utiliza materiais advindos de três pesquisas que a autora participou como pesquisadora associada do Grupo de Estudos sobre Organização da Pesquisa e da Inovação (GEOPI)⁴ em 2004 e como gerente de pesquisas do Observatório Digital SOFTEX⁵, entre março de 2005 e março de 2007. As pesquisas foram:

⁴ O GEOPI (Grupo de Estudos sobre Organização da Pesquisa e da Inovação) é um Grupo do Departamento de Política Científica e Tecnológica (DPCT/IG/Unicamp). Mais informações em www.ige.unicamp.br/geopi

⁵ Unidade de pesquisas da Associação para Promoção da Excelência do Software Brasileiro. Mais informações, bem como as publicações das pesquisas, em www.softex.br/observatorio

Pesquisa Impacto do Software Livre na Indústria de Software do Brasil: primeiro levantamento sobre as formas de organização técnica e econômica do SL/CA (Software Livre e de Código Aberto) no Brasil, realizado entre fevereiro a dezembro de 2004. Teve como objetivos identificar os principais mercados e modelos de negócio, relacionar especialistas (empresas e profissionais) em SL/CA no país, dimensionar o mercado consumidor, identificar as condições de apropriação do conhecimento, assim como o potencial de desenvolvimento e uso em determinadas áreas e mercados. A pesquisa teve quatro fontes de dados: um painel que reuniu 50 especialistas da academia, das empresas desenvolvedoras e usuárias, do governo e da comunidade de SL/CA; uma enquete eletrônica com 3.657 respondentes (a maior já realizada dentro de um único país); um conjunto de entrevistas com empresas desenvolvedoras e usuárias de SL/CA; e um levantamento de informações secundárias sobre empresas que lidam com SL/CA no Brasil, sendo pelo menos 364 empresas desenvolvedoras e 154 usuárias. A pesquisa, encomendada pelo Ministério de Ciência e Tecnologia, teve a coordenação executiva de Giancarlo Stefanuto (SOFTEX) e a coordenação científica de Sergio Salles-Filho (DPCT/Unicamp)⁶.

Perspectivas de desenvolvimento e uso de componentes na Indústria Brasileira de Software e Serviços: o objetivo geral do estudo, realizado em 2006 pela SOFTEX em parceria com o Departamento de Política Científica e Tecnológica da Unicamp e com apoio do Ministério de Ciência e Tecnologia, foi avaliar as contribuições que a Engenharia de Software Baseada em Componentes (ESBC) pode trazer para a Indústria Brasileira de Software e Serviços em termos de sua qualificação técnica e de sua inserção econômica no mercado global de software e serviços relacionados. A pesquisa teve a coordenação executiva de Giancarlo Stefanuto (SOFTEX) e a coordenação científica de Sergio Salles-Filho (DPCT/Unicamp)⁷. A pesquisa utilizou o seguinte conjunto de metodologias: levantamento bibliográfico sobre o estado da arte do tema componentização, Painel Estruturado de Especialistas e Exercício de Cenários Futuros

⁶ A equipe completa foi composta por Adrián S. De Witt B. e Carolina Mattos (SOFTEX), Ana Maria Carneiro e Rogério da Veiga (GEOPI/DPCT/ UNICAMP), Angela Maria Alves (CenPRA - Centro de Pesquisas Renato Archer), José Eduardo de Lucca (UFSC) e Fernando Colugnati (UNIFESP).

⁷ A equipe completa foi composta por Ana Maria Carneiro, Adrián S. De Witt B., Tainakã Fuentes Tacca e Ana Serino de Rezende (SOFTEX), Ricardo Minoda e Rogério da Veiga (UNICAMP), Angela Maria Alves (CenPRA), José Eduardo de Lucca (UFSC).

Alternativos. Finalmente, concluiu-se o trabalho com uma análise dos pontos fortes e fracos, ameaças e oportunidades (SWOT – Strengths, Weaknesses, Opportunities, Threats). O estudo contou com a contribuição de cerca de 60 especialistas do governo, academia e de empresas nas diversas reuniões realizadas.

Indústria Brasileira de Software e Novas Modalidades de Outsourcing: o objetivo da pesquisa, realizada pela SOFTEX com o apoio do Ministério de Ciência e Tecnologia, foi avaliar e aprofundar o entendimento sobre os condicionantes para inserção do Brasil no mercado internacional de serviços e potenciais oportunidades para esta inserção com foco nas novas modalidades de outsourcing. A pesquisa teve a coordenação executiva de Giancarlo Stefanuto e Ana Maria Carneiro (SOFTEX) e a coordenação científica de Paulo Tigre (IE/UFRJ)⁸.

Em quarto lugar, foi realizado um exercício de monitoramento tecnológico utilizando patentes para observar padrões e tendências de patenteamento entre empresas de diferentes modelos de negócio. Para isso, foi construída uma amostra com patentes de dezesseis empresas para analisar a distribuição no tempo, entre outras análises. Além disso, foram geradas outras duas amostras utilizando as técnicas de classificação de patente de software dos estudos realizados por Bessen e Hunt (2004a) e Graham e Mowery (2003).

Além desta introdução, a tese possui quatro capítulos e as considerações finais. O capítulo 1 apropria-se do instrumental teórico dos evolucionistas e da economia dos custos de transação para construir o aparato conceitual para as análises. Este instrumental é submetido a algumas adaptações para o tratamento da apropriação dos resultados da inovação na Economia do Conhecimento.

Os capítulos 2 e 3 contam a história da co-evolução da indústria de software (e mais amplamente da produção de software) e dos instrumentos de proteção dos direitos de propriedade intelectual. O capítulo 2 trata do período de 1945 a 1994, período em que se configuraram os principais modelos de negócio da indústria de software antes do advento e disseminação da Internet. O

⁸ A equipe completa foi composta por Solange Corder, Maria Beatriz Bonacelli e Sergio Salles Filho (UNICAMP), Felipe Silveira Marques e Leandro de Jesús (UFRJ), Carolina Mattos e Ana Serino de Rezende (SOFTEX).

capítulo 3 discute o período de 1994 ao momento atual, marcado pela popularização da Internet, fenômeno que implicou em mudanças nos modelos de negócio.

No primeiro período discutido, que vai do nascimento da atividade de software ao início da expansão da Internet comercial, a discussão maior era sobre qual instrumento de proteção jurídica seria mais adequado ao software. No segundo período esta discussão continua, mas com a clareza da utilização dos vários instrumentos complementares. A discussão maior é sobre se é preciso mais proteção formal com um forte coro de que talvez seja preciso menos.

O capítulo 4 procura discutir as formas de proteção da propriedade intelectual e as tendências de apropriação, mergulhando no processo de “fabricação” do software dentro de um regime tecnológico específico, o do reuso de software e de uma das tecnologias deste regime, os componentes de software. Os componentes são tomados como exemplo da complexidade que os aspectos de propriedade intelectual adquiriram na indústria de software.

Dada a evolução histórica da indústria de software, foram analisadas as formas como as empresas procuraram proteger seus heterogêneos ativos intangíveis, sendo que para cada modelo de negócio há uma necessidade diferente de proteção, considerando ainda que os mecanismos de proteção variam também de acordo com o país.

Na década de 90, assistiu-se a dois movimentos aparentemente contraditórios. Por um lado, ao fortalecimento dos direitos de propriedade intelectual, como o grande crescimento de patentes de software, um tipo de instrumento que até então não era muito usual entre as empresas de software. Por outro lado, no mesmo período, o software livre e de código aberto (SL/CA), que é tanto fruto quanto impulsionador da migração da indústria de software para os modelos de negócio baseados em serviços, ganhou corpo e introduziu formas alternativas de desenvolvimento de software e de disponibilização e proteção dos conhecimentos gerados, baseadas em licenças.

A contradição se desfaz ao se contextualizar os dois movimentos. A corrida patentária está relacionada principalmente ao comportamento defensivo de grandes *players* da indústria de software, o que é observado em outras indústrias, que procuram construir grandes portfólios de

patentes para ter o que negociar em acordos de licenciamento cruzado, que podem ser entendidos como acordos de não agressão mútua.

O segundo movimento está ligado à mudanças estruturais na indústria de software em relação aos modelos de negócio mais fortemente baseados em serviços, nos quais os regimes de apropriabilidade são compostos mais por formas de proteção dos ativos complementares, muitas das quais não passam por registros.

Esta tese procura, portanto, avançar o debate, com o desafio de ampliar seu escopo, por meio da contribuição do arcabouço teórico da política científica e tecnológica, principalmente com a visão da propriedade intelectual como instituição não só voltada para os fins de proteção do conhecimento (e exclusão de rivais), mas que também habilita a articulação entre os agentes envolvidos no processo inovativo (e, assim, os efeitos de inclusão).

1. PROTEÇÃO DA PROPRIEDADE INTELECTUAL E APROPRIAÇÃO DOS RESULTADOS DA ATIVIDADE INOVATIVA

Para analisar a co-evolução da indústria de software e da apropriação dos resultados das atividades inovativas é preciso ter em conta que se trata de um processo multicausal. Há vários fatores que influenciam o desenvolvimento de uma indústria, sendo difícil isolar um deles, no caso a proteção da propriedade intelectual (PI), como causa primária. Segundo Merges (1996), a investigação da pergunta – “como o regime de PI afetou o desenvolvimento da indústria de software” – passa pela análise da pergunta inversa – “como a indústria de software afetou o regime legal”⁹. A proteção da propriedade legal deve ser analisada, então, no contexto inovativo da indústria, considerando ainda que os agentes econômicos têm racionalidade limitada, a história importa, a mudança é custosa e as competências das empresas são heterogêneas (Teece, 1986).

O presente capítulo busca estabelecer os conceitos que serão usados para entender como se dá a apropriabilidade na indústria de software, inserida no contexto maior da Economia do Conhecimento. Para isso, o capítulo está dividido em duas seções. A primeira recorre à teoria evolucionista e à economia dos custos de transação para construir o quadro conceitual da tese. A segunda seção procura dar mais algumas pinceladas ao quadro para acrescentar alguns elementos necessários para tratar dos objetos da Economia do Conhecimento.

1.1 Regimes tecnológicos e de apropriabilidade

A apropriação dos resultados do esforço inovativo não acontece no abstrato, pois depende das competências internas da firma, do ambiente externo e das instituições. Em vista disso, o instrumental teórico evolucionista e da economia dos custos de transação parece bastante útil para entender o comportamento das empresas de software em relação à apropriação dos resultados de seus esforços de inovação.

Na abordagem evolucionista, as empresas são entendidas como repositórios de conhecimentos, que são em grande parte incorporados em suas rotinas operacionais e modificados ao longo do

⁹ Em sua análise comparativa sobre a proteção da propriedade intelectual nas indústrias de software dos EUA, Europa e Japão, Merges (1996) apontou que a PI exerce uma pequena influência nas barreiras à entrada na indústria e na própria estrutura da indústria, como será discutido no capítulo 2.

tempo por meio de suas regras de comportamento e estratégias. Segundo Dosi e Marengo (1994), as competências são características relacionadas à solução de problemas de um conjunto particular de interações organizacionais, normas e estratégias explícitas. Assim, competências apresentam um grau significativo de inércia e especificidade da firma. As competências são propriedades coletivas das rotinas de uma organização e, devido em parte ao seu caráter tácito, são difíceis de transferir ou copiar (Coriat e Dosi, 2000).

Ainda no escopo da abordagem evolucionista, o trabalho de Malerba e Orsenigo (1996) relaciona o comportamento das firmas a aspectos mais estruturais da economia. O comportamento da empresa é entendido como intimamente ligado às suas competências, o que co-evolui com o desenvolvimento da tecnologia, da demanda e das instituições, sendo a Propriedade Intelectual uma destas instituições (Malerba e Orsenigo, 1996). A firma, para traçar suas estratégias competitivas e tecnológicas, precisa articular seus ambientes externos e internos (Tigre, 2006).

O acesso a uma determinada tecnologia, longe de ser livre, envolve um aspecto de aprendizado. O aprendizado tecnológico e organizacional dentro de cada firma é, devido ao caráter tácito, em boa parte local e dependente dos caminhos trilhados por ela. Onde uma empresa pode ir depende da sua posição atual, de onde pretende ir e do caminho já percorrido. Assim, as empresas não possuem um leque infundável de opções, pois os investimentos feitos anteriormente e sua história conformam seu comportamento futuro (Teece e Pisano, 1998).

Isso tudo ocorre dentro de um regime tecnológico, noção baseada no conceito de paradigma tecnológico de Nelson e Winter (1982). Os regimes tecnológicos são caracterizados a partir de quatro conceitos relativos à estrutura industrial no que diz respeito à tecnologia: condições de oportunidade, condições de apropriabilidade, graus de cumulatividade do conhecimento tecnológico e complexidade da base de conhecimento. Estes elementos conformam o ambiente no qual a firma opera, definindo as prescrições e *trade-offs* que identificam os mecanismos dinâmicos básicos e os comportamentos viáveis em termos das estratégias tecnológicas básicas e os tipos básicos de organizações das empresas (Malerba e Orsenigo, 1996).

As condições de **oportunidade**, que refletem a facilidade de inovar mediante um determinado montante de investimento feito em busca, são pensadas em termos de nível (alta ou baixa oportunidade) e pervasividade (alta ou baixa pervasividade). Assim, dependendo do estágio de

desenvolvimento da tecnologia, há diferentes potenciais de oportunidades, como atualmente entre a indústria têxtil (baixa) e indústria de biotecnologia (alta). Oportunidades tecnológicas podem ser exógenas, como as pesquisas em ciência básica feitas nas universidades, ou endógenas à indústria, como as oportunidades geradas pela própria atividade inovativa (Teece e Pisano, 1998). A percepção das oportunidades acontece conforme as competências internas das empresas. “Assim, não apenas as empresas de uma mesma indústria encontram menus com diferentes custos associados com escolhas tecnológicas particulares, mas elas também estão olhando para menus que contêm diferentes escolhas” (*id. ibid*, p.204).

As condições de **apropriabilidade** dos resultados da inovação representam uma proteção (teórica) para o inovador contra os imitadores. Estas dependem das características da tecnologia e do sistema legal, sendo que podem incluir desde patentes até o controle dos ativos complementares (este último especialmente importante em condições de baixa apropriabilidade). Desta forma, o fato de alguém ter inventado algo não significa automaticamente que conseguirá leva-lo ao mercado e ainda lucrar com isso (Teece, 1986).

Para Dosi (1984, p.88), as condições de oportunidade e de apropriabilidade juntas representam as condições interligadas para a atividade inovativa em economias de mercado, pois “para cada dada percepção de oportunidade tecnológica, os benefícios econômicos para o inovador (ou imitador rápido) devem ser proporcionais ao grau de apropriabilidade privada das inovações e das externalidades, que com frequência são associadas às atividades inovativas (como expertise, *know-how*, conhecimento)”. Assim, a oportunidade tecnológica e a apropriação privada representam condições interligadas para a atividade inovativa nas economias de mercado. A primeira é uma condição necessária, mas não suficiente, enquanto que a segunda define o grau de compromisso das empresas com a atividade inovativa.

Os graus de **cumulatividade** indicam o processo de acumulação de competências junto com aprendizado (*learning-by-doing*) nas firmas e na indústria. Por fim, a **base de conhecimento** é considerada segundo duas dimensões: a) grau de “*tacitness*”, o quanto um determinado conhecimento é tácito, ou seja, local e específico a uma empresa, ou o quanto pode ser codificado

e mais facilmente acessível¹⁰; e b) grau de complexidade, o quanto requer a integração de diferentes disciplinas científicas e tecnológicas e o quanto as atividades inovativas dependem de uma variedade de competências envolvendo processos de produção, natureza dos mercados, características das demandas, entre outras (Malerba e Orsenigo, 1996, p.46).

Muitas vezes a discussão sobre as condições de apropriabilidade tem sido reduzida aos direitos de propriedade intelectual, mas esta é uma dimensão de um conjunto mais amplo (Dosi *et al*, 2006, p.1111). Como visto, as condições de apropriabilidade estão interligadas com as demais condições dos regimes tecnológicos e, a partir deste conjunto de variáveis, as empresas conformam suas estratégias tecnológicas. Antes de tratar destas estratégias a partir do trabalho de Malerba e Orsenigo (1996), cabe aprofundar um pouco mais a questão da apropriabilidade, utilizando a noção de regimes de apropriabilidade (Teece, 1986; 2000; Teece e Pisano, 1998).

O **regime de apropriabilidade** refere-se aos fatores do ambiente (excluindo-se a estrutura da firma e do mercado) que governam a habilidade de um inovador de capturar os ganhos gerados pela inovação. Isto envolve duas dimensões: a natureza da tecnologia (se é produto ou processo, se é conhecimento tácito ou codificado) e os mecanismos legais de proteção (Teece, 1986).

Ainda segundo este autor, o regime de apropriabilidade pode ser classificado como forte ou fraco dependendo da natureza da tecnologia e da eficácia do sistema legal em conceder e proteger os direitos de propriedade intelectual (PI). O regime de apropriabilidade está ligado à atividade de imitação, pois quanto mais fácil a cópia, menor a apropriabilidade. Assim, o regime de apropriabilidade é considerado forte quando, ao mesmo tempo, é difícil replicar a inovação (tecnológica, organizacional etc.) e o sistema de PI proporciona barreiras legais à imitação. Já em regimes fracos de apropriabilidade, nos quais a tecnologia é fácil de imitar, os ativos complementares, como marketing, manufatura e suporte pós-vendas, tornam-se críticos e o inovador pode ser levado a integrá-los na estrutura de sua empresa (*id. ibid*). Entretanto, dominar todos os ativos complementares pode ser desnecessário ou ter custo proibitivo.

Desta forma, as empresas são levadas a interagir, negociar e estabelecer parcerias, entre outras

¹⁰ O conhecimento codificado é um ativo de baixo custo de transferência, pois parte dos seus elementos pode se tornar componentes comercializáveis. Já o conhecimento tácito é mais difícil de transferência e comercialização, sendo que o alto custo de transferência torna sua difusão mais lenta. O conhecimento tácito e a competência adquirida se tornam fontes de diferenciação (Carvalho, 2003).

opções, para desenvolverem suas atividades em ambientes de alta densidade tecnológica. Desta forma, Teece insere na discussão da apropriabilidade uma dimensão contratual (Winter, 2006).

Bebendo na economia dos custos de transação (Coase, [1936] 1994; Williamson, 1987), Teece¹¹ trata da decisão da firma de comprar ou desenvolver internamente, na medida em que a fronteira da firma é afetada pelos custos de transação. A essência da empresa é que ela dispensa a organização do mercado, porque em seu interior consegue organizar e coordenar certos tipos de atividades econômicas de formas que não são possíveis usando os mercados (Teece e Pisano, 1998). Isso ocorre não apenas devido aos custos de transação, mas por conta das competências em organizar e fazer com que as coisas sejam feitas de formas que não podem ser coordenadas via sistema de preços. Uma competência difícil de replicar ou de imitar pode ser considerada uma competência distintiva. Assim, a fronteira da empresa é uma variável estratégica para os inovadores, ou seja, até que ponto integrar, e passar a produzir internamente, ou contratar de outros. O limite das atividades que devem ser coordenadas dentro da empresa ou pelo mercado muda com o tempo, bem como as escolhas das empresas (Chesbrough *et al*, 2006).

Isso, no limite, pode resultar em um modelo de empresa (totalmente) integrado ou (totalmente) contratual. Este último é facilitado em um regime de apropriabilidade forte, no qual os ativos complementares são disponibilizados de forma competitiva. Assim, licenciar uma tecnologia é uma boa opção para quem tem o direito de propriedade, conta com regime de apropriabilidade forte e não tem todos os ativos complementares para colocar a inovação no mercado (Teece, 2000). Um sistema de direitos de propriedade intelectual (DPI) forte facilita transações no mercado por *know-how*. Mas há riscos de se basear apenas em formas contratuais de licenciamento (Teece, 2006). Por outro lado, quando uma empresa detém os ativos complementares para produzir a inovação, isto já pode ser suficiente para garantir a apropriação, tendo a proteção legal um papel secundário.

A distribuição dos lucros entre imitadores e inovadores, ainda segundo Teece (1986), é determinada não tanto pela estrutura dos mercados, mas das empresas, especialmente o escopo de suas fronteiras, relacionadas com políticas nacionais de desenvolvimento de ativos

¹¹ Teece (2006) considera seu esquema teórico mais rico que a economia dos custos de transação, pois envolve precificação dos ativos e noções de desequilíbrio, além das questões contratuais e dos custos de transação.

complementares.

Com a valorização dos ativos complementares, Teece liga a inovação às estruturas de mercado, uma forma de apontar que a inovação deve ser vista no contexto no qual se situa, ou seja, a própria definição do que será a inovação tem muito a ver com o leque de estratégias de apropriabilidade disponíveis para o inovador e, para isso, a estrutura do mercado conta (Winter, 2006). Desta forma, os ativos complementares têm papel não apenas na apropriabilidade, mas também na estratégia do empreendimento que está sendo formado (Teece, 2006)

Um exemplo ilustrativo deste aspecto pode ser observado na parceria estratégica que foi estabelecida entre a IBM e a Microsoft em relação ao desenvolvimento do sistema operacional MS-DOS no início da história do computador pessoal (PC), que será aprofundado no capítulo 2. O computador pessoal da IBM é considerado, ao mesmo tempo, um dos maiores sucessos (Teece, 1986) e fracassos na indústria de computadores (Chesbrough *et al*, 2006).

A IBM realizou uma grande inovação na indústria de computadores no início dos anos 80, quando criou seu computador pessoal baseado em uma arquitetura aberta, que permitia a montagem de um computador a partir de componentes padronizados. Em relação ao sistema operacional, software básico de infra-estrutura do computador, a IBM decidiu pela contratação da Microsoft, e esta parceria permitiu acesso a novas tecnologias e aprendizado, sem anos e anos de tentativa e erro (Teece, 1986). Já do lado da Microsoft, empresa menor à época, o contrato com uma empresa maior teve efeito de transbordamento (*spillover*) de reputação, pois lhe deu “credibilidade instantânea” (*id. ibid*).

No longo prazo, entretanto, não controlar os ativos complementares pode acabar cedendo lucro para imitadores ou outros competidores que dominem estes ativos. A arquitetura deu vantagem competitiva para a IBM, pois permitiu que seu computador pessoal desbancasse outros tipos de computadores com arquitetura fechada. Mas como a arquitetura aberta tornou-se padrão, permitiu também que outros fabricantes de hardware pudessem criar clones de seu computador pessoal. Além disso, a IBM não conseguiu prever as conseqüências da sua ação e a importância que o software estava adquirindo, pois ao permitir que a Microsoft licenciasse o sistema operacional para outros fabricantes de hardware, a IBM acabou perdendo a liderança no mercado que ajudou a criar: um computador com arquitetura aberta que podia ser montado usando componentes de

vários produtores (O'Reilly, 2005). O sistema operacional da Microsoft veio a se tornar um padrão, o que permitiu ganhos de escala para outros desenvolvedores de aplicativos, e também para a Microsoft, que entrou neste mercado.

Como apresentado acima, uma vez obtida uma vantagem competitiva, ela não está assegurada para sempre, pois suas fontes vão se dissipando à medida que a tecnologia se dissemina no mercado ou é superada por outras. Estudando as empresas líderes no mercado global, Teece e Pisano (1998) elaboraram o paradigma das **capacidades dinâmicas** para mostrar como uma vantagem competitiva é obtida e mantida, relacionando o ambiente externo (dinâmico) às competências, recursos e habilidades internas que devem ser adaptadas, integradas e reconfiguradas por uma gestão estratégica apropriada. Assim, “os vencedores no mercado global são empresas que conseguem demonstrar reação em tempo e inovação de produto rápida e flexível, envolvida por uma capacidade de gestão para efetivamente coordenar e recriar competências internas e externas” (Teece e Pisano, 1998, p.194).

As capacidades dinâmicas são estratégicas na medida em que são difíceis de negociar no mercado e de replicar ou imitar. Aqui pode ser citado o papel importante dos modelos de negócio das empresas, especialmente aqueles difíceis de replicar e que são constantemente aperfeiçoados (Teece, 2006). Um exemplo é o sucesso do modelo de venda direta da Dell (Rivette e Kline, 2000)¹².

Como dito, quanto mais tácito for o conhecimento produtivo da empresa, mais difícil replicá-lo ou imitá-lo. A imitação também é dificultada pelo conjunto de barreiras do sistema de propriedade intelectual; proteção, entretanto, que não deve ser superestimada: “a proteção da propriedade intelectual não é uniforme em relação a produtos, processos e tecnologias e é melhor entendida como uma ilha em um mar aberto de competição” (Teece e Pisano, 1998, p.207).

Assim, na vigência de um determinado regime, são mais prováveis determinados comportamentos das empresas. Por exemplo, no caso do regime de circuito integrado na indústria de computadores, houve a introdução de novos produtos e consequente melhoria nos anos subsequentes por produtores estabelecidos ou avanços nas competências em engenharia e

¹² Para comprar seu micro, o cliente pode entrar direto no site da Dell e montar como quiser a especificação do seu computador de mesa (*desktop*) ou computador móvel (*laptop*).

produção, altos investimentos em equipamentos de manufatura e grande produção por empresas integradas verticalmente (*id. ibid*).

Malerba e Orsenigo resumem as estratégias prováveis em vários regimes tecnológicos possíveis, a partir das associações das condições de oportunidade e apropriabilidade e graus de cumulatividade, como resumido na Figura 1.1.

Em cada regime tecnológico, as empresas podem lançar mão de estratégias mais adequadas. Em geral, o número de possibilidades de estratégias tecnológicas e organização das atividades inovativas aumenta quanto mais pervasivas forem as oportunidades tecnológicas, maior o grau de cumulatividade do progresso técnico, menor o grau de apropriabilidade e mais complexa a base de conhecimento relevante (quadrantes II, IV e VI) (Malerba e Orsenigo, 1996). Ou seja, nestas condições as empresas têm mais espaço para se mexerem, especialmente se forem as *first-movers* ou não dispuserem dos ativos complementares, pois nestes regimes tecnológicos há espaço para várias estratégias como adquirir os ativos complementares e inovar incrementalmente, dentre outras estratégias.

		Alta oportunidade		Baixa Oportunidade	
		Alta cumulatividade	Baixa cumulatividade	Alta cumulatividade	Baixa cumulatividade
Alta apropriabilidade	I Inovações Radicais Inovações incrementais Sugestões para inovações: <i>Balancear entre inovações radicais e incrementais</i> <i>Não se restringir a tecnologias existentes</i>	III Inovações radicais Sugestões para inovações: <i>Continuar a ser inovativo</i> <i>Monitorar novas tecnologias</i>	V Inovações incrementais Sugestões para inovações: <i>Fortalecer as competências fundamentais</i>	VII Ausência de atividades inovativas sistemáticas	
	II Inovações Radicais Inovações incrementais Sugestões para inovações: <i>Balancear entre inovações radicais e incrementais</i> <i>Não se restringir a tecnologias existentes</i> <i>Manter o timing dos líderes</i> <i>"Move down the learning curve"</i> <i>Adquirir ativos complementares</i>	IV Inovações Radicais Imitação Sugestões para inovações: <i>Continuar a ser inovativo</i> <i>Monitorar novas tecnologias</i> <i>Lucrar rapidamente com as inovações</i> <i>Adquirir ativos complementares</i>	VI Inovações incrementais Imitação Sugestões para inovações: <i>Fortalecer as competências fundamentais</i> <i>Adquirir ativos complementares</i>	VIII Ausência de atividade inovativa	

Figura 1.1 - Estratégias tecnológicas em vários regimes tecnológicos

Fonte: Malerba e Orsenigo (1996, p.61).

A estas estratégias são acrescentadas outras características a partir da consideração da complexidade da base de conhecimento, resultando na Figura 1.2.

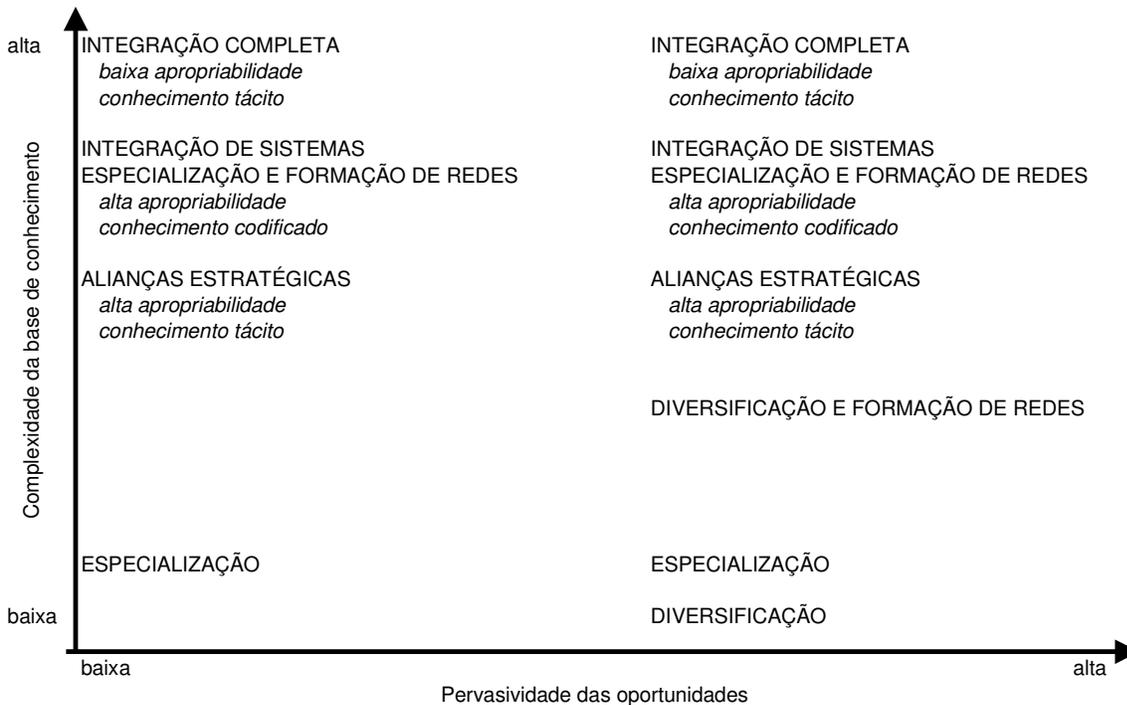


Figura 1.2 - Comportamento das empresas em oportunidades pervasivas e base de conhecimento complexa

Fonte: Malerba e Orsenigo (1996, p.64).

O comportamento das empresas, assim, pode variar desde a diversificação (em casos de alta pervasividade das oportunidades e baixa complexidade da base de conhecimento), passando por especialização, formação de redes ou alianças estratégicas, integração de sistemas até a integração completa (em casos de alta ou baixa pervasividade das oportunidades e alta complexidade da base de conhecimento) (*id. ibid*).

1.1.1 Funções da propriedade intelectual

Tendo tratado da apropriação no contexto mais amplo dos regimes tecnológicos e das estratégias de inovação das empresas, passa-se agora para uma apresentação das funções e dos estatutos de proteção legal da propriedade intelectual, que, como visto, representam um dos aspectos das

condições de apropriação de importância crescente nas duas últimas décadas (OCDE, 2004).

A apropriação do conhecimento pode ser feita por meio de mecanismos jurídicos de proteção da propriedade intelectual, mas não somente, que variam de acordo com o país e são mais ou menos adequados dependendo da tecnologia.

Propriedade intelectual é definida como a soma dos direitos relativos às várias criações e invenções humanas – tais como obras literárias, artísticas e científicas; interpretações dos artistas; fonogramas e emissões de radiodifusão; invenções em todos os domínios da atividade humana; descobertas científicas, desenhos e modelos industriais; marcas industriais, comerciais e de serviço – e também à proteção contra a concorrência desleal (D. Barbosa, 1997).

A PI é um dos elos entre a geração e a apropriação do conhecimento, que orienta as relações entre os atores públicos e privados, definindo quem participa e quem não participa. Desta forma, é um mecanismo de interação entre os agentes econômicos, de organização de investimentos e pesquisa em inovação, ajudando a organizar mercados, a definir condições para distribuição de benefícios e a orientar a execução de políticas públicas (Salles-Filho, 2007).

A propriedade intelectual é uma instituição historicamente situada e justificada (A. Barbosa, 2005; Biagioli, 2006). A justificativa mais difundida atualmente, mas também duramente criticada, coloca que o objetivo da propriedade intelectual é incentivar as invenções e inovações por meio de um caminho duplo. Tomando-se o sistema de patentes como exemplo, por um lado, este visa proteger os inventores contra a imitação por meio da concessão de um direito de excluir os demais atores econômicos de explorar sua criação ou descoberta por um período determinado de tempo. Por outro lado, visa possibilitar o acesso à tecnologia de forma a gerar benefícios sociais, seja por meio da difusão da invenção no mercado para que a sociedade possa gozar dela, seja por meio da disseminação da informação tecnológica e do conhecimento protegido (Buainain *et al*, 2005).

Este duplo objetivo da propriedade intelectual é também contraditório, posto que envolve dimensões por vezes irreconciliáveis (o interesse do inventor e da sociedade) e é fonte de contínua contestação devido ao desafio de proteger para incentivar, mas evitar que a proteção vire abuso nas mãos dos detentores dos direitos de propriedade intelectual, gerando monopólios

injustificados sem benefício social. Nesta linha coloca-se Chesbrough *et al* (2006, p.1098):

Uma terceira implicação política crítica é a força da proteção de PI e suas causas. Neste assunto específico, ninguém toma a posição de que toda PI deve ser protegida inteiramente. Todavia, o desafio de quanta proteção é socialmente necessária e desejável para motivar, por um lado, o esforço empreendedor, e por outro, para incentivar a difusão ampla, é um balanço difícil de se obter.

Com alta apropriabilidade, apenas uma pequena porção dos benefícios se espalha pelo sistema econômico na forma de aperfeiçoamentos eficientes, aprendizado por meio da imitação e mudanças de preço (Dosi, 1988).

Em resumo, podem ser apontadas quatro funções da propriedade intelectual, a partir de Carvalho (2003). A primeira refere-se à **apropriação privada do esforço de inovação**, por meio da concessão do monopólio temporário de exploração econômica da invenção ou criação, que oferece proteção ao investimento realizado em troca da divulgação do conhecimento. A base desta função é que o processo de inovação – entendido como a busca por (e descoberta) experimentação, desenvolvimento e adoção de novos produtos, processos e formas organizacionais – envolve uma dimensão inerente de incerteza (Dosi, 1988). Desta forma, uma das funções da propriedade intelectual é diminuir a incerteza quanto à apropriação dos resultados do investimento em inovação¹³.

A segunda função é a possibilidade conferida pela propriedade intelectual de **tornar a tecnologia passível de transação econômica** e, assim, transformar inovações em ativos comercializáveis e ampliar a articulação entre agentes econômicos. Desta forma, a propriedade intelectual é não apenas um mecanismo de proteção do conhecimento, mas também um mecanismo de articulação entre agentes. Esta função abre possibilidades de criação de um comércio de DPIs e de captação de capital de risco.

A segunda função possibilita a terceira, qual seja, a **função estratégica da propriedade**

¹³ Como será visto no capítulo 2, esta função não é inteiramente alcançada no período inicial em que novos objetos, como software, passam a ser elegíveis de proteção por um determinado instrumento. Este período inicial é marcado por insegurança se a empresa conseguirá o registro, por exemplo a patente de software, e se este novo tipo de registro de PI será validado perante um tribunal.

intelectual que a torna um fator de barganha para acesso ou abertura de mercados específicos. Neste sentido, a PI funciona como ingresso para entrar no jogo e, uma vez nele, pode servir como escudo para defender-se dos outros jogadores ou como arma de ataque. Esta terceira função está relacionada também com a complexidade do conhecimento e com o custo da inovação em algumas atividades, o que dificulta que uma mesma empresa consiga cobrir todo o espectro de conhecimento necessário para atuar em um setor. As empresas lidam com isso por meio de aquisição e fusões de empresas com ativos intangíveis complementares e por meio de algumas formas de licenciamento, como o licenciamento cruzado ou licenciamento de um pacote de tecnologias. Esta função será retomada no próximo item, dada a magnitude que tem assumido nos últimos tempos.

A quarta função da propriedade intelectual é a **função prospectiva e de divulgação**, que possibilita a difusão do conhecimento protegido e o aproveitamento de oportunidades tecnológicas por terceiros, visto que as tecnologias das invenções patenteadas, por exemplo, não estão no formato mercadológico, sendo ainda necessário esforço de desenvolvimento. Além disso, a divulgação do conhecimento permite que empresas utilizem-no para monitorar a atividade dos concorrentes e construir sua estratégia tecnológica e mercadológica (Carneiro *et al*, 2004).

A propriedade intelectual é regulada por um conjunto de estatutos ou leis que definem os objetos de proteção jurídica. Os marcos de referência são assentados em leis nacionais que devem sempre ser ajustadas aos acordos internacionais dos quais o país é signatário. A propriedade intelectual é dividida em três grandes campos de proteção jurídica (Carvalho, 2003), a saber:

- **Propriedade industrial:** protege o conjunto de atividades relacionadas às invenções, desenho industrial, marcas, indicações geográficas e designação de origem, e concorrência desleal, por meio de patente de invenção e modelo de utilidade, registro de marca, indicação geográfica e designação de origem e segredos de negócio;
- **Direitos de cópia ou direitos de autor:** protegem a atividade de criação intelectual de trabalhos literários, artísticos, musicais, visuais e programas de computador, sendo que estes

últimos podem ter legislação nacional específica¹⁴;

- **Formas *sui generis* de proteção:** protegem cultivares, topografia de circuitos integrados, diversidade biológica e conhecimentos tradicionais a ela associados. Estas formas mesclam características dos outros dois campos e tentam proteger áreas fruto do desenvolvimento científico e tecnológico.

Os instrumentos legais protegem de formas diferentes as tecnologias, possuindo escopo, período e padrões para proteção diferenciados. Por exemplo, *copyright* protege a expressão da idéia e a forma da criação, enquanto a patente visa proteger a invenção e as idéias nela contidas. As condições para obtenção de *copyright* são originalidade e autoria¹⁵. Já as condições para a obtenção de uma patente são cinco: a) a invenção ser uma matéria patenteável; b) ter aplicação industrial (no caso dos EUA, utilidade); c) ter novidade; d) ter atividade inventiva ou não ser óbvia; e e) apresentar a descrição da invenção¹⁶. Além disso, há custos e mecanismos diferentes associados à obtenção e manutenção da proteção e também para fazer valer o direito frente a violadores (Teece, 2000; Merges *et al*, 2006).

Os objetos de proteção variam de acordo com a legislação nacional. Por exemplo, a legislação norte-americana, considerada mais liberal, permite o patenteamento de plantas, enquanto a brasileira não. Esta e outras diferenças devem-se ao fato de que “a concessão de direitos de propriedade intelectual é uma prerrogativa de caráter nacional” (Carvalho, 2003, p.6).

Cabe destacar que, além das diferenças nacionais¹⁷, os campos de proteção são dinâmicos e evoluem ao longo do tempo, acompanhando com certa distância o desenvolvimento científico e tecnológico – a indústria de software, como será visto adiante, é um exemplo interessante deste parâmetro na medida em que seu desenvolvimento inseriu um novo objeto no *copyright* e

¹⁴ Há ainda os direitos conexos relacionados à mediação entre a criação e o consumo da obra protegida (Carvalho, 2003).

¹⁵ Já foi necessário que a obra estivesse fixada num suporte físico. Mas com o avanço das TICs, esta exigência foi modificada para fixação em suportes tangíveis e intangíveis.

¹⁶ Esta última exigência, que permite a difusão do conhecimento da patente, é um ponto chave na discussão sobre se o sistema de patentes cumpre sua função de divulgação, sendo esta a contrapartida exigida em troca da exclusividade de exploração econômica da invenção. No caso das patentes de software, argumenta-se que as descrições não cumprem este papel.

¹⁷ Estas diferenças têm sido cada vez mais harmonizadas internacionalmente, como mostra o advento do Acordo sobre Aspectos de Propriedade Intelectual Relacionados ao Comércio (TRIPs), que será discutido no capítulo 2.

ampliou o escopo deste estatuto legal (Santos, 2003). Por um lado, os detentores das invenções muitas vezes não consideram que o nível de proteção seja suficiente frente aos direitos que consideram justos. Por outro lado, é função dos Estados Nacionais balancear a tensão entre os ganhos privados e os direitos da sociedade em termos de acesso aos bens e serviços protegidos (Carvalho, 2003).

Entre o quê o quadro legal especifica e sua execução há um espaço enorme de manobra para se usar os mecanismos de proteção. Além disso, a proteção nunca é completa, pois a tecnologia pode ser imitada ou superada, e a capacidade de apropriação apresenta variações, conforme as demais condições do regime tecnológico.

O tema da propriedade legal deixa, então, de ser exclusivo do departamento legal para se tornar um assunto estratégico para as empresas (Rivette e Kline, 2000).

1.1.2 Uso estratégico da propriedade intelectual

Segundo Cohen *et al* (2002), são vários os motivos para uma empresa patentear uma invenção além de sua exclusividade de exploração econômica: medir sua performance, faturar com licenciamento para terceiros, usar em negociações, evitar litígio, bloquear competidores, fortalecer sua reputação, entre outros. Aliás, o patenteamento de uma invenção não significa que a empresa irá realizar sua exploração comercial, pois o estágio de desenvolvimento da invenção pode ainda não estar no ponto de levar ao mercado, ou as tecnologias auxiliares e ativos complementares estarem pouco desenvolvidos.

Segundo estes mesmos autores, os produtos ou processos das indústrias podem ser classificados em complexos – cuja proteção está distribuída por uma quantidade grande de patentes, de forma que uma empresa dificilmente pode cobrir todo o espectro – e discretos – cuja proteção é feita por um pequeno número de patentes. Assim, as empresas de produtos complexos possuem direitos sobre tecnologias que outras empresas necessitam, o que cria uma condição de mútua dependência e fomenta licenciamento cruzado, negociações e formas de compartilhamento da informação:

Comparada com as indústrias de produtos discretos, as patentes são usadas menos para reforçar a exclusividade e mais para assegurar mercados (por meio do acesso à

tecnologia-chave possuída por outros) e liberdade de operação (conferida por meio da habilidade de contra-atacar se acusada de infração e assim desencorajando litígios) (Cohen et al, 2002, p.1356).

Estas duas formas de utilização (assegurar mercado e liberdade de operação) são denominadas por Coriat e Orsi (2002) de uso estratégico da PI, pois de certa forma “extrapolam” o uso previsto na justificativa atual da PI, como visto acima.

Coriat e Orsi (2002) apontam ainda que os mercados financeiros¹⁸ possuem um papel importante no novo regime de PI, o que leva a um modelo de inovação dirigido para as oportunidades de negócio (*finance-driven*) e não para estratégias de exclusão de oponentes. No caso das empresas de biotecnologia, por exemplo, foi estabelecido um mercado para comércio de DPIs. Já no caso das empresas de TIC, a situação é mais complicada: ao invés de mercado de DPIs, as empresas fazem o uso estratégico de direitos.

Segundo estes mesmos autores, estes direitos agora foram estendidos e reforçados. Gallini (2002) aponta que nas últimas duas décadas, o sistema de proteção via patentes nos EUA foi fortalecido de três formas. A primeira delas foi a extensão da proteção patentária a novas matérias, através de decisões das Cortes. Os principais casos foram:

- caso *Diamond vs. Chakrabarty* – decisão da Suprema Corte tornou patenteável bactérias modificadas via engenharia genética em 1980;
- caso *Diamond vs. Diehr* – em 1981, a Corte aprovou a patenteabilidade de software puro;
- caso *State Street Bank and Trust vs. Signature* – em 1998, a Corte de Apelações do Circuito Federal validou a decisão que permitiu o patenteamento dos métodos de negócio.

Segundo Coriat e Orsi, estas extensões de objetos de PI representam a privatização de campos antes públicos.

¹⁸ Para isso foram necessárias mudanças na regulação financeira, como as resultantes do *Bayh-Dole Act* (dos EUA) e a possibilidade de empresas, mesmo com prejuízos por vários anos, serem listadas na NASDAQ, pois estas empresas podem incluir uma série de ativos intangíveis no seu prospecto preliminar, sendo o portfólio de patentes o mais importante deles.

A segunda forma foi a criação da Corte de Apelações do Circuito Federal (CAFC), dedicada exclusivamente para lidar com os casos envolvendo violação e validade de patentes. O objetivo era uniformizar o tratamento destas ações, mas, segundo Gallini (2002), o resultado foi o aumento do sucesso dos proprietários de patentes nas disputas legais. Isto revelaria uma inclinação Pró-PI, especialmente pró-patente.

A terceira forma foi a adequação da legislação norte-americana ao TRIPS (Acordo sobre Aspectos de Propriedade Intelectual Relacionados ao Comércio) quanto ao prazo mínimo de validade de patentes de 17 anos, a partir da data de concessão, para 20 anos, a partir da data de depósito¹⁹. Além das sanções econômicas previstas no tratado relacionadas ao comércio.

O resultado deste processo pode ser caracterizado como uma corrida às patentes, com empresas correndo para patentear suas inovações, às vezes por meio de uma inundação de solicitações nos escritórios de patentes do mundo. Os autores indicam duas situações básicas em que é possível observar esta tendência do uso estratégico dos direitos de PI:

i) Junto com os benefícios das externalidades de mercado, a concessão de patentes fortes para programas de computador tem frequentemente reforçado a posição de grandes firmas incumbentes²⁰ (...)

ii) No extremo do espectro, as novas patentes que têm sido concedidas para os programas de computador e/ou modelos de negócios (...) têm frequentemente facilitado a entrada de pequenas firmas especializadas em nichos específicos de mercado. Capitalistas de risco serão especialmente preparados para trabalhar com tais firmas quanto maiores forem as chances de que elas sejam adquiridas no futuro por grandes firmas e se as inovações dessas start-ups forem originalmente protegidas por patentes ou copyright (Coriat e Orsi, 2002, p.1502-3).

O resultado destas estratégias forma uma rede densa de DPIs, com muitas patentes com sobreposições em seus requerimentos, com a qual uma empresa tem que lidar para colocar uma nova tecnologia no mercado. Esta rede densa de DPIs tem sido chamada de “cipoal de patentes”

¹⁹ Gallini (2002) aponta que também foram tomadas medidas para contrabalancear isso, como a publicação após 18 meses do depósito da patente.

(*patent thicket*) (Shapiro, 2001) – novamente, a indústria de software é uma das áreas em que este fenômeno mais tem se manifestado. Este fenômeno está relacionado a produtos que envolvem muitas patentes, os produtos complexos conforme a denominação de Cohen *et al* (2002). Nestes casos, nenhum ator individualmente detém todo o conhecimento e DPIs da tecnologia, o que força a interação e negociação entre atores.

São várias as formas de interação entre empresas para atravessar o cipoal de patentes usadas. Uma das formas de coordenação entre empresas se dá por meio do **licenciamento cruzado**, que acontece quando duas empresas têm patentes que interessam uma a outra. Trata-se de uma forma especializada de licenciamento comum em indústrias de alta tecnologia e em sistemas tecnológicos cumulativos. O principal objetivo é fortalecer a habilidade das firmas de desenvolver inovações utilizando tecnologia de uma ampla variedade de fontes.

Normalmente, o licenciamento cruzado envolve pouca transferência de tecnologia. Ao contrário, “envolve troca de direitos para o uso de tecnologia para garantir que as empresas tenham ‘liberdade para operar’ no desenvolvimento e exploração das inovações, sem o risco de numerosos requerimentos de violações por parte dos detentores das patentes na mesma área tecnológica” (Teece, 2000, p.138-139). Além disso, o licenciamento cruzado pode ser livre de pagamento de *royalties*. Mas também pode ser estabelecida uma balança de pagamentos baseada na importância das patentes envolvidas.

Desta forma, o licenciamento cruzado pode ser visto como um espaço entre os agentes econômicos que facilita a cooperação e pode diminuir os custos de transação associados (Carvalho, 2003). Este tipo de licenciamento pode ser usado para assegurar liberdade de design (*design freedom*) e de manufatura (*freedom-to-manufacture*). Um exemplo clássico desta forma de coordenação foi a constituição da Radio Corporation of America (RCA) em 1919, que reuniu cerca de 2000 patentes envolvidas na produção do rádio, o que tanto possibilitou que a RCA tivesse a liderança técnica do rádio, quanto permitiu que os outros licenciados continuassem o desenvolvimento para uso em outras áreas ou para fornecimento para RCA (Grindley e Teece, 1997).

²⁰ O termo incumbente é utilizado aqui como sinônimo de empresa estabelecida, em oposição aos novos entrantes.

Entretanto, este espaço de cooperação e de diminuição dos custos de transação pode ter um efeito perverso em relação aos novos entrantes, que possuem menos recursos, como apresentam Buainain *et al* (2005, p.19):

“Quando se faz necessária a utilização do licenciamento cruzado de patentes, especialmente nos segmentos onde o custo e o tempo da imitação são altos, a obtenção de titularidade de ativos intangíveis de propriedade intelectual tem sido estratégia que as empresas estabelecidas nesse mercado utilizam para levantar mútuas barreiras à entrada. Para explorar comercialmente a inovação é necessário que os titulares dos ativos intangíveis entrem em acordo. Por outro lado, os novos entrantes se vêem na necessidade de possuírem uma carteira própria de patentes para forçar as demais a negociarem acordos de licenciamento. Um exemplo importante nesse tipo de estratégia é a biotecnologia, onde fusões e licenciamento cruzado desempenharam papéis relevantes na conformação da indústria”.

No caso da indústria de software, como será discutido nos demais capítulos, houve uma corrida inicial ao patenteamento, considerado um dos instrumentos mais fortes de PI, por parte de vários agentes econômicos. Estes buscavam formar um portfólio de patentes com o objetivo de reunirem armas para se defenderem em caso de acusação de violação, ou para terem o que negociar em casos de acordos de licenciamento cruzado. As implicações desta corrida são várias, mas uma que tem sido bastante destacada na literatura é a baixa qualidade das patentes de software, especialmente as que foram concedidas no começo da corrida patentária, muitas das quais não cumpriam os requisitos mínimos de uma patente, ou tinham o escopo muito amplo.

Segundo Shapiro (2001), outras formas de interação entre empresas em torno do uso estratégico da PI são formação de pacote de licenças de patentes (*patent pools* ou *package licenses*), bloqueio de patentes relacionadas com uma área estratégica e estabelecimento de padrões²¹.

O **pacote de licenças de patentes** é um acordo entre detentores de patentes para licenciar um conjunto de suas patentes para terceiros. Isto ocorre quando duas ou mais empresas controlam as patentes necessárias para produzir um determinado produto e uma outra empresa deseja produzir

²¹ Também há casos em que as empresas usam de aquisições ou pagam para empresas saírem do mercado em questão.

este produto, mas não possui nenhuma patente relacionada com esta tecnologia. A forma de solução deste “problema dos complementos” é a reunião das patentes relacionadas com a tecnologia em questão num pacote de licenças. Esta solução resolve bem o problema do licenciado²² que comodamente pode de uma vez adquirir todas as licenças que necessita. Em tese, seriam reunidas as patentes essenciais, complementares e não rivais no pacote, o que seria fomentado inclusive pelas instituições que cuidam da competitividade do mercado. O pacote de licenças pode ser administrado por uma nova entidade criada exclusivamente para este fim.

O **bloqueio de patentes** é realizado por uma empresa por meio do patenteamento de invenções relacionadas a uma invenção *core* patenteada, como se estivesse colocando blocos ao redor desta. Desta forma, ela pode evitar que seu concorrente consiga patentear as invenções ao redor da sua. Entretanto, se outra empresa tiver conseguido patentear partes desta invenção, pode-se criar um impasse, pois cada uma teria uma parte dos blocos necessários para construir o todo. Uma saída é a formação do pacote de licenças, estratégia comum no contexto do **estabelecimento de um padrão de forma cooperativa**, quando duas ou mais empresas estabelecem acordos de não agressão mútua. Segundo Shapiro (2001, p.20) isto ocorreu entre Microsoft e Netscape em torno de uma tecnologia para web:

Até os inimigos mais severos frequentemente se juntam na indústria de software para promover novos padrões. Em 1997, Microsoft e Netscape, duas companhias não conhecidas como parceiras amigáveis, concordaram em incluir versões compatíveis da Virtual Reality Modeling Language (desenvolvida pela Silicon Graphics) em seus navegadores. Esperava-se que este acordo facilitasse a visualização de imagens 3-D na Web pelos consumidores.

Estas estratégias envolvem portfólios de patentes mais do que patentes individuais, baseadas no peso relativo do conjunto de suas patentes relacionadas com uma determinada área. Segundo Bessen (2003), quanto maior o portfólio, maior a possibilidade de uma empresa ganhar, em caso de litígio, e também maior o seu poder de barganha em negociação de licenciamento cruzado.

Este autor entende o patenteamento como uma atividade econômica resultado do *trade off* entre a

²² Nesta tese, serão utilizados os termos licenciante, para indicar o titular do direito, e licenciado, para aquele que recebe a permissão para utilizar este direito.

probabilidade de ganhar uma ação de litígio versus o custo de patentear. Segundo este autor, os padrões de patenteamento afetam o custo de obter uma patente e determinam o custo de construir um portfólio de patentes. Os problemas que resultam destas estratégias geralmente são custos de transação, problemas de *hold-up*²³ e monopólios verticais (Shapiro, 2001). Estes problemas podem reduzir o investimento em P&D e o ritmo da inovação. Contudo, Bessen (2003, p.2) vê acontecendo atualmente redução dos incentivos para investir em P&D em situações em que não há estes problemas:

O cipoal de patentes pode reduzir os incentivos de P&D até mesmo quando não há custos de transação, problemas de holdup ou de monopólio vertical. Há diferentes tipos de estratégias de licenciamento cruzado com implicações normativas fortemente diferentes. De fato, em contraste com a Oracle, que vê o licenciamento cruzado como um último recurso, outras companhias buscam agressivamente construir grandes portafólios de patentes e utilizá-los para extrair benefícios dos competidores. A IBM, por exemplo, adquiriu cerca de cem vezes do número patentes da Oracle durante os anos 1990 embora seu orçamento de P&D fosse apenas cinco vezes maior que o da Oracle.

Segundo este autor, o licenciamento cruzado também pode reduzir o efeito de incentivo das vantagens do *lead time* porque os lucros do inovador acabam sendo incluídos na negociação do licenciamento cruzado e compartilhados na solução barganhada. Além disso, o licenciamento cruzado agressivo das empresas incumbentes leva a um grande compartilhamento dos lucros, que acabam sendo maiores quando os padrões de patenteamento são baixos.

O uso estratégico pode ser coletivamente útil se favorece a circulação da informação na indústria. Cohen *et al* (2002) estudaram a relação entre os spillovers de P&D, os incentivos para inovar e os instrumentos de proteção nos EUA e Japão. Especificamente buscaram estudar o papel e eficiência do sistema de patentes e como as diferenças das leis de patentes dos dois países e a prática administrativa e jurídica afetam o grau de proteção à inovação dado pelas patentes. Em resumo, os principais resultados foram:

²³ O problema de *hold-up* ocorre quando há um acordo entre dois agentes (X e Y), sendo que um deles (X) necessita realizar um investimento considerável e não transferível sem grandes custos (*sunk costs*). O agente X teme um possível comportamento oportunista de Y devido ao poder de barganha deste último durante e após o acordo. Às vezes, a expectativa de X de quebra de contrato, por parte de Y, é tão grande que chega a impossibilitar o acordo.

A despeito do gasto com P&D como um percentual do PIB ser maior no Japão do que nos EUA, os spillovers de informação do P&D intra-indústria parecem ser maiores no Japão para quase todas as indústrias e a apropriabilidade dos resultados da inovação parece ser menor. Em contraste aos EUA, os respondentes japoneses reportaram o segredo como um mecanismo menor de apropriabilidade e patentes como mais efetivas comparativamente a outros mecanismos. Uma razão chave para menos apropriabilidade e maior spillover de P&D intra-indústria no Japão parece ser que a patente difunde informação através dos rivais mais prontamente que nos EUA (Cohen et al, 2002, p.1350).

Os autores sugerem que as diferenças entre os dois países são devidas não apenas às diferenças de políticas (no Japão a função de divulgação do conhecimento é mais reforçada que nos EUA, por exemplo), mas também devido às diferentes formas de uso das patentes. Nos EUA, na indústria de produtos eletrônicos, considerada uma indústria de produtos complexos, as empresas raramente controlam todos os componentes das tecnologias que estão desenvolvendo. O cipoal de patentes cria dependência entre empresas, criando uma condição de dependência mútua que fomenta licenciamentos cruzados extensivos, negociações e compartilhamento de informações. Desta forma, comparada com indústrias de produtos discretos, as patentes são usadas menos para reforçar a exclusividade e mais para assegurar o acesso ao mercado e a liberdade de operação das empresas (Cohen et al, 2002).

Nos dois países, as formas mais utilizadas foram a vantagem de ter saído na frente (*lead time advantage*) e capacidades complementares de manufatura e vendas. Patentes foram posicionadas como últimas na lista de mais efetivas nos EUA e entre as primeiras mais efetivas no Japão. Segundo os autores, as empresas nos EUA baseiam-se mais no segredo e conseguem se apropriar mais dos resultados da inovação. Já no Japão, as empresas utilizam mais as patentes, que funcionam mais como canais de fluxo de informação entre as empresas e menos como instrumentos de apropriação. Assim, observaram diferenças de apropriabilidade entre as indústrias, mas também entre os países devido a fatores como instituições, políticas, culturas e normas diferentes nos dois países:

Nós conjecturamos, desta forma, que encorajando o licenciamento cruzado e o patenteamento agressivo, o sistema de patentes japonês gera mais divulgação e troca de informação. Nós sugerimos que estes impactos nos fluxos de informação de um número

*maior de patentes por produto reforçam os efeitos do sistema japonês da política de priorizar patentes, baseando-se na divulgação e oposição no processo de patenteamento*²⁴ (Cohen *et al*, 2002, p.1362).

Assim, as diferenças entre os dois países devem-se, por exemplo, às diferenças entre seus sistemas de patentes em termos de divulgação e processo de oposição à patente antes do término do exame de concessão, o que fortalece a contribuição das patentes para o fluxo de informação entre as empresas no Japão. Além disso, o número menor de requerimentos e a construção de requerimentos de menor escopo, que favorece a solicitação de um número maior de patentes, fazem com que a dependência mútua aumente entre os rivais, de forma que eles tenham uma grande tendência a usar patentes para bloquear rivais, licenciamento cruzado e negociações, o que favorece a circulação da informação. Este tipo de comportamento, que Cohen *et al* (2002) denominam estratégia do jogador, só é encontrado nos EUA nas indústrias de produtos complexos enquanto no Japão é mais disseminado.

A estratégia do jogador tem objetivos diferentes nos dois países: no Japão é utilizada para assegurar inclusão e liberdade de operação mesmo em mercados de produtos discretos. Já nos EUA, as empresas tendem a usar as patentes mais para fins de exclusão. As patentes no Japão tornam-se armas em interações estratégicas. As empresas sentem-se compelidas a patentear tanto para se defender de ações ou de bloqueios ou porque querem bloquear rivais ou usar as patentes como objetos de barganha em negociações.

A preocupação colocada pelos autores revisados (Coriat e Orsi, 2002; Gallini, 2002; Shapiro, 2001) é que mais patentes não sejam sinônimo de mais inovação. Teme-se que, no afã da corrida de patentes, as empresas estejam depositando patentes que não atendem propriamente aos requisitos mínimos ou têm o escopo muito amplo, o que é agravado pelo fraco exame dos escritórios nacionais de propriedade intelectual devido ao grande número de processos resultando em patentes de baixa qualidade. Isto estaria ocorrendo especialmente nos novos objetos elegíveis para patenteamento devido às mudanças na legislação ou pela decisão dos tribunais.

Este quadro seria agravado em campos de inovação sequencial, como o software. O uso

²⁴ No processo de patenteamento japonês, há um mecanismo que permite que empresas possam manifestar oposição a uma patente que sendo analisada após a divulgação (18 meses a contar do depósito).

estratégico das patentes poderia atrasar o processo de inovação, como aponta Shapiro (2001, p.28):

Nosso atual sistema de patente está causando uma situação potencialmente perigosa em vários campos, que incluem biotecnologia, semicondutores, software e comércio eletrônico, nos quais um empreendedor novato ou inovador pode encontrar uma barreira de ações de violação que deve ser transposta para colocar seu produto ou serviço no mercado. Em outras palavras, nós estamos diante do perigo de enfrentar um aumento significativo dos custos de transação para aqueles que buscam comercializar novas tecnologias baseadas em múltiplas patentes, direitos sobrepostos e problemas de hold-up.

O uso estratégico da PI não é um fenômeno particular da indústria de software, ocorrendo também em outros setores de grande dinamismo, como será discutido no próximo item.

1.2 Apropriação na Economia do Conhecimento

A linguagem bélica tem sido usada com frequência para descrever a atividade de patenteamento nos últimos anos. As patentes são tratadas como bombas inteligentes na guerra dos negócios, que devem ser obtidas pelas empresas com objetivo defensivo ou de ataque, sendo o grande desafio corporativo o desenvolvimento de novas táticas e estratégias para os novos campos de batalha da Economia do Conhecimento (Rivette e Kline, 2000; Warshofsky, 1994).

Descontando-se a linguagem de auto-ajuda empresarial, o que é certo é que houve uma “explosão” no depósito de patentes nas últimas décadas. De uma média de 60 mil depósitos/ano por parte de inventores nacionais na década de 80, o Escritório Norte-Americano de Marcas e Patentes (USPTO) viu a média anual subir para 150 mil na década de 1990 e atingir mais de 200 mil de 2000 a 2006²⁵. Segundo Rivette e Kline (2000), no caso das empresas high-tech, a corrida começou em 1994. Por exemplo, as empresas Sun, Oracle, Dell, Novell e Intel aumentaram seus portfólios mais de 500% entre 1994 e 1998. E junto com a explosão de patentes, houve um boom no faturamento derivado de licenciamento, que aumentou 700% entre 1990 e 1998. Em 2000, os autores consideravam que o mercado de licenciamento estava apenas na infância.

²⁵ As fontes são: Gallini (2002), para décadas de 80 e 90, e cálculo próprio a partir de dados do USPTO para os anos 2000.

A corrida ao patenteamento é um fenômeno mais geral, relacionado com a transformação da economia mundial dada a conformação da Economia do Conhecimento, mas que acontece de forma mais acentuada nas indústrias *high-tech*, como a de software.

Como mencionado, neste período fica mais evidente que a PI é um assunto não só apenas do departamento jurídico, mas um tema estratégico para a empresa. Grindley e Teece (1997) ilustram isto com o processo de decisão em relação ao patenteamento da empresa HP (Hewlett-Packard). Quando uma invenção é alcançada, ela é encaminhada a uma reunião da Coordenação de Patentes, formada por engenheiros, gerentes e pessoas do departamento jurídico. Estes recorrem então às recomendações jurídicas, técnicas e organizacionais além de observar o Guia de Avaliação da empresa. A invenção poderá ter três destinos: 1) publicação de forma a evitar que seja patenteada por outros; 2) ser mantida como segredo de negócio ou 3) requerimento de patente.

Entretanto, a visão do tema como estratégico não é altamente disseminada. Segundo Rivette e Kline (2000), apenas uma minoria das empresas americanas possuíam uma estratégia de proteção da PI e evitavam falar sobre ela, por se tratar de sua “arma secreta”. As mais bem sucedidas empresas – como Intel, Microsoft, Lucent, Xerox e IBM – começaram a usar suas patentes como armas competitivas para capturar e defender mercados, superar rivais e aumentar faturamento. Para estas empresas, a estratégia de patenteamento era vista como a nova competência *core* – gestão e uso estratégico de PI.

Como já apresentado, a apropriação dos resultados da inovação é realizada pelas empresas por meio de um conjunto de possibilidades, que inclui estatutos jurídicos e outras formas que não passam por registro formal. Teece mostrou que para entender como empresas lucram com suas inovações é preciso olhar para os aspectos da organização econômica, a estratégia de negócios, tecnologia e a inovação (Chesbrough *et al*, 2006). A gestão do portfólio de PI não é feita independente da estratégia de negócios e vice-versa (Teece, 2006) e nem das novas variáveis da economia global digitalmente interconectada, que são discutidas a seguir. O foco desta seção estará nas patentes, devido a sua relevância no tempo atual, que pode ser denominado inclusive de mundo pró-patente (Tang e Paré, 2003).

1.2.1 Novas variáveis no jogo da apropriação

Em 2006, a revista *Research Policy* organizou uma edição especial para comemorar os 20 anos do artigo de D. Teece intitulado *Profiting from technological innovation*. Nesta edição, vários autores evolucionistas, bem como autores menos conhecidos, comentaram as contribuições deste importante artigo para a análise da apropriação da inovação, discutiram eventos ocorridos desde o lançamento do trabalho em 1986 e novas variáveis que colocam novos desafios teóricos. Esta discussão é destacada aqui, pois nesse mesmo período histórico a indústria de software cresceu e se consolidou como atividade econômica. Algumas novas variáveis, inclusive, estão relacionadas diretamente à indústria de software.

Uma destas variáveis é que a própria tecnologia torna-se fonte de lucro, com a criação de mercados para startups obterem dividendos por meio de suas inovações, sem a necessidade de investir nos ativos complementares, como apresentado por Chesbrough *et al* (2006, p.1096):

As firmas precisam considerar até que ponto “lucrar a partir da inovação” requer acesso aos ativos complementares de produção e marketing que permitam a elas comercializar seus produtos e até que ponto elas podem lucrar comercializando a tecnologia ou o conhecimento que sustentam os produtos. Estas são escolhas estratégicas bem diferentes.

Exemplos destacados destes fenômenos são as *start-ups* da bolha da Internet, como será apresentado no capítulo 3. Nas palavras de Coriat e Orsi (2002), trata-se do modelo de inovação voltado para as oportunidades de negócios.

Os ativos complementares também adquiriram novos formatos. Um exemplo são as redes de inovação aberta em bases globais, como alianças ou comunidades de desenvolvimento, como as comunidades de software livre, que podem se tornar ativos complementares de uma inovação (Dahlander e Wallin, 2006; Chesbrough *et al*, 2006). A prática de amplo uso de patentes e segredos pode ser contraprodutiva para algumas empresas, pois isso pode erguer barreiras entre a empresa e seus aliados potenciais, o que leva à necessidade de desenvolver uma atitude mais balanceada em relação à apropriabilidade (Chesbrough *et al*, 2006). Além do papel de ativo complementar, as comunidades de software livre e de código aberto (SL/CA) têm um papel importante na conformação da indústria de software e também na discussão sobre a propriedade intelectual, como será discutido nos capítulos 3 e 4.

O próprio Teece publicou um novo artigo discutindo os aspectos que deveriam ser aprofundados em relação ao artigo de 1986. O autor realçou o papel importante das tecnologias complementares para possibilitar o desenvolvimento da inovação. Como exemplo, citou a fotografia digital que teve que esperar o surgimento de memória de baixo custo para se difundir. No caso das tecnologias sistêmicas, há a necessidade de buscar as tecnologias complementares e suas patentes e também o papel relevante exercido pelas instituições de suporte: instituições reguladoras, instituições de ensino, instituições padronizadoras e cortes judiciais (Teece, 2006).

Uma das questões mais comentadas nesta edição especial da *Research Policy* foi a relação entre inovação e apropriação, uma discussão muito frequente também na literatura sobre apropriação dos resultados da inovação na indústria de software. No fundo, os autores discutem se a justificativa apresentada no início deste capítulo – de que a patente é um incentivo ao inovador por seu esforço em troca da disseminação do conhecimento – ainda é válida frente aos desenvolvimentos tecnológicos e científicos, as mudanças na regulação dos DPIs, entre outros fatores.

Segundo Nelson (2006)²⁶, quando Teece em 1986 focou as estratégias e meios de apropriação, o contexto era diferente do atual, pois os direitos de propriedade intelectual não tinham o papel atual de veículos centrais para assegurar apropriação. Então, Nelson volta-se para as questões de política e lei de patente, pois nos anos recentes ganhou força o argumento que direitos de propriedade mais amplos e mais fortes estimulam inovação e, quanto mais forte, melhor para o crescimento econômico. Tenta então levantar questões sobre se o que é bom para o inovador individual é bom para o progresso econômico.

Se se toma as elaborações teóricas como reflexos do espírito do período histórico, pode-se trabalhar com a idéia que, como faz Winter (2006), comparando os fundamentos abstratos do problema da apropriabilidade em Schumpeter, Arrow e Teece, a discussão passou do foco da sociedade para as empresas voltando agora para a sociedade. No seu entender as contribuições dos dois primeiros autores estavam explicitamente voltadas para as implicações da apropriabilidade para a sociedade tomada como um todo. Já o trabalho de Teece de 1986 tratava

²⁶ Nelson (2006) supõe em suas análises que empresas sempre querem um regime forte de apropriabilidade, mas nem sempre é assim, como mostram os exemplos do software livre.

de como as empresas poderiam lucrar com suas inovações, considerando que a sociedade em geral estaria bem servida com empresas inovativas e lucrativas.

Como será discutido no próximo capítulo, em 1986 iniciava-se a Rodada Uruguaí do Acordo Geral sobre Tarifas e Comércio (GATT) – o qual se transformou em Organização Mundial do Comércio (OMC) e imprimiu uma nova dinâmica de discussão em várias frentes de debate. Prova disso é a posição norte-americana de que este era o local adequado para discutir propriedade intelectual, mais que a Organização Mundial de Propriedade Intelectual (OMPI). Este posicionamento devia-se à percepção de perdas de competitividade de sua economia em função da fraca proteção da propriedade intelectual no comércio internacional, o que permitia que novos países industrializados aumentassem sua participação no comércio internacional em grande parte com produtos que imitavam os produtos inovadores de empresas norte-americanas. Assim, havia o pensamento que era necessário reforçar o sistema de PI para assegurar uma maior apropriação dos resultados das atividades inovativas. Isso foi feito em nível internacional, com o TRIPs, e nacional, com a reforma do sistema norte-americano de patentes (Gallini, 2002). Até então, as questões de PI eram tratadas na OMPI com o foco na relação entre o desenvolvimento tecnológico dos países. Com o TRIPs são deslocadas para a esfera do comércio internacional (Carvalho, 2003).

Ao que parece, assiste-se agora à volta das preocupações para o foco mais geral da sociedade, pois o que a literatura aponta é que o fortalecimento do regime de apropriabilidade via regulação da Propriedade Intelectual pode estar obstruindo o desenvolvimento das empresas inovadoras e lucrativas. Esta discussão é particularmente importante em relação à indústria de software dado seu alto grau de oportunidade e pervasividade na economia. A indústria de software é considerada uma das mais inovativas atividades econômicas. A apropriação da inovação passa por vários instrumentos formais, mas também por formas alternativas, como a própria atividade de inovar mais rápido que o concorrente, novas formas de comercialização, marketing e melhorias de processo, entre outras.

Os autores questionam se o crescimento das taxas de patenteamento significa realmente crescimento das taxas de inovação ou, pelo contrário, empecilhos para a inovação. Dosi *et al* (2006) apontam várias hipóteses para a explosão de patentes:

- explosão análoga no aumento das atividades inovativas fruto do crescimento de volume e qualidade do progresso científico e tecnológico;
- mudanças na estrutura legal e institucional e também nas estratégias das empresas (patente como ativo financeiro, uso estratégico para erguer barreiras à entrada e defesa de possíveis litígios, etc);
- expansão dos domínios patenteáveis, como já apresentado, e dos atores que patenteiam (universidades e agências públicas)²⁷.

Vários estudos mostram que a utilização de patentes é restrita e concentrada entre as empresas, isto é, trata-se de um instrumento utilizado por um grupo relativamente pequeno de empresas, na sua maioria grandes empresas (Nelson, 2006; Gallini, 2002) . Segundo Nelson (2006), os *surveys* que realizou com seus colegas na Yale University, em 1986 e em 1994, sobre as condições de apropriabilidade apontaram que as formas de apropriação em geral das empresas não envolvem patentes, com exceção de algumas indústrias como a indústria farmacêutica. Na maioria das indústrias, as firmas utilizam mais frequentemente outros mecanismos como segredo, vantagem dos *first-mover* e exploração de capacidades complementares.

Dosi *et al* (2006) argumentam que várias inovações geraram valor sem serem patenteadas ou que foram patenteadas num regime fraco de apropriabilidade, como o exemplo do transistor. Esta tecnologia foi patenteada pelo *Bell Labs*, o laboratório de P&D da empresa AT&T, mas, em consequência de uma ação de antitruste e pressão do Departamento de Justiça norte-americano, teve que ser licenciada amplamente a preços considerados razoáveis. Os autores (*id. ibid*, p.1113) ressaltam que as demais tecnologias *core* das TICs emergiram em um ambiente fraco de apropriabilidade:

Nós sugerimos que os direitos de propriedade fortes não desempenharam um papel importante na emergência das TICs e nem mesmo na geração de valor. Ao contrário disso,

²⁷ Kortum e Lerner (1999) apostavam mais em fatores internos às empresas, como a mudança na gestão da P&D para explicar o crescimento do patenteamento. Segundo estes autores, as hipóteses de mudanças externas (“tribunal amigável” (*court friendly*) e aumento das oportunidades tecnológicas) não eram suficientes para explicar tal fenômeno. Gallini (2002) aponta que a pesquisa de Kortum e Lerner deve ser complementada com o aumento do patenteamento por parte de empresas estrangeiras no período posterior à observação dos autores, o que daria suporte para a explicação do “tribunal amigável”.

nos primeiros estágios destes setores, deve ter sido o fraco regime de patentes que incentivou seu rápido crescimento. Inversamente, o fortalecimento do regime de PI nos anos recentes (logo após o boom das TICs no final dos anos 80) deve ter sido (em termos de influência política) uma consequência mais do que uma causa para o ritmo rápido no qual o setor de TICs se expandiu.

Há uma forte discussão se os instrumentos formais, principalmente a patente, fomentam ou atrapalham a inovação na indústria de software. Esta discussão acirrou-se com a concessão de patentes de baixa qualidade, que não cumprem os requisitos mínimos, como mostram Dosi *et al* (2006, p.1117):

Enquanto a qualidade das patentes diminui e seus usos estabelecem poucas conexões com os requerimentos de estímulo à produção e difusão do conhecimento, os custos empreendidos para lidar com os conflitos e requerimentos que se sobrepõem em PI aumentam junto com a incerteza sobre a extensão da responsabilidade legal no uso de inputs de conhecimento.

Resumindo o capítulo 1, como apresentado, os mecanismos jurídicos não são auto-suficientes para proteção e têm que ser complementados com outras formas, que envolvem estratégias de valorização dos ativos relevantes e complementares e outras formas de apropriação dinâmica, como mecanismos técnicos, inovar mais rapidamente que o concorrente, entre outros. A PI impacta de forma diferenciada os setores econômicos. É conhecido que a patente é mais eficiente para proteger as invenções da indústria farmacêutica e da biotecnologia que a da indústria metal-mecânica. Por fim, proteger uma dada tecnologia não passa apenas por optar por um formato jurídico, mas por fazer a gestão dos ativos.

O que se quer destacar aqui é que a apropriação dos resultados dos esforços de inovação não passa apenas pela codificação destes resultados para registrá-los sob algum mecanismo formal, como direito de autor, patente, registro de software ou marca, pois:

- a) parte considerável do conhecimento não pode ser codificado e assim registrado;
- b) a despeito dos mecanismos formais poderem ser utilizados por qualquer empresa, isso depende das competências internas das empresas;

- c) além disso, a incorporação de um conhecimento se dá de forma diferenciada entre os diferentes atores do processo de inovação, que deriva da capacitação dinâmica de quem incorpora a tecnologia, o que valoriza as estruturas organizacionais e aponta que “são criados elementos tácitos também no processo de incorporação da tecnologia, tão mais importantes quanto mais a natureza da tecnologia contiver conhecimentos superpostos e complementares” (Carvalho, 2003, p.16);
- d) a posse de uma patente, por exemplo, não significa o final feliz da história, mas provavelmente o começo de uma história da qual não se conhece o fim. Para a empresa e sua patente “viverem felizes para sempre” dependem também as competências da firma e os regimes tecnológicos e de apropriabilidade a que estão submetidas.

Não basta ter os ativos e os direitos de propriedade intelectual, é necessário saber articulá-los com práticas de gestão da PI. Assim, as condições de apropriação variam de acordo com o local (os instrumentos disponíveis em cada país/região), o tipo de tecnologia, o mercado no qual a empresa está inserida, as estratégias das empresas e sua capacidade de utilizar os instrumentos disponíveis.

Neste capítulo além da revisão da bibliografia clássica, procurou-se apontar novas variáveis no jogo da apropriação, como o uso estratégico de patentes (que pode ter vários efeitos), a articulação e cooperação entre agentes econômicos, o reforço de barreiras à entrada, o aumento dos custos de transação, entre outros.

Nos capítulos 2 e 3, a história da indústria de software será revisada para discutir a co-evolução do desenvolvimento tecnológico e da conformação da apropriação dos resultados da inovação por parte das empresas.

2. A EVOLUÇÃO DA INDÚSTRIA DE SOFTWARE PRÉ-INTERNET

O objetivo do presente capítulo é mostrar a interação entre a evolução da tecnologia e das formas de proteção relacionadas com a indústria software no período que compreende o início desta indústria – por volta de 1945 – até meados da década de 90. Pretende-se mostrar a evolução conjunta da indústria e da forma de apropriação neste período e, assim, analisar como a tecnologia evoluiu e como os agentes econômicos se apropriaram dos resultados desta evolução em suas diversas fases.

O modelo evolucionista ajuda a compreender a evolução da indústria de software. Segundo Dosi e Nelson (1994), nos primeiros estágios de uma indústria, as empresas tendem a serem pequenas e a entrada relativamente fácil, refletindo a diversidade de tecnologias empregadas. Entretanto, à medida que um paradigma tecnológico emerge, surgem (espontaneamente ou não) barreiras à entrada à medida que cresce a necessidade de escala e capital para uma produção competitiva. O aprendizado torna-se cumulativo e as firmas incumbentes possuem certas vantagens em relação às entrantes. Entretanto, quando emerge uma nova tecnologia, esta pode substituir a anterior e dar início a um novo paradigma, no qual as incumbentes podem ter dificuldade em adquirir as novas competências necessárias, de forma que as empresas entrantes podem abocanhar uma parte do mercado²⁸.

Os regimes tecnológicos (para usar a terminologia adotada no capítulo 1) podem ser pensados em relação ao papel dos três elementos em torno dos quais a indústria de software gira: hardware, software e serviços. O primeiro regime foi centrado no hardware, no caso os grandes computadores (*mainframes*), tendo o software e os serviços de processamento²⁹ um papel de suporte. Com o surgimento da arquitetura padronizada dos *mainframes*, surgiram os primeiros

²⁸ No caso da indústria de software, em geral a necessidade de capital é relativamente menor e o custo de produção da primeira unidade é alto, mas o custo de reprodução é próximo de zero. Isso varia de acordo com o modelo de negócio, como será visto adiante.

²⁹ Aqui foi adotada uma simplificação, pois na verdade, inicialmente os serviços de processamento fornecidos por empresas especializadas tinham um papel de substituição em relação aos *mainframes* para as empresas usuárias que não tinham condições financeiras de aquisição dos grandes computadores e nem competências internas para o desenvolvimento de software.

vendedores de software independentes (ISVs na sigla em inglês), sendo que o software ganhou um papel maior com a separação da venda do hardware e software, em 1965.

O segundo regime foi o do computador pessoal (PC), com o sistema operacional padronizado (MS-DOS/Windows), sendo o hardware um suporte para o software. O momento atual é o de transição do regime do PC para um novo, marcado por serviços disponíveis na Internet (no estilo do modelo do Google), sendo que o software e o hardware não “importam muito”, servindo como plataforma para oferecimento de serviços. Isto não significa o fim do software como o conhecemos hoje, mas que está se tornando tão comum e disseminado em uma gama de equipamentos que praticamente não é mais notado. Esta é a idéia de *software everywhere* (*desktop*³⁰, PCs mais simples denominados *thin clients*, celulares entre outros dispositivos móveis e mesmo eletroeletrônicos). É importante lembrar que elementos do novo e do velho regime coexistem e é comum que tecnologias ultrapassadas fiquem relegadas a nichos particulares (Teece, 2006). Isto é muito claro na indústria de software, como ilustra a Figura 2.1, em que o regime tecnológico anterior continua fazendo parte do posterior.

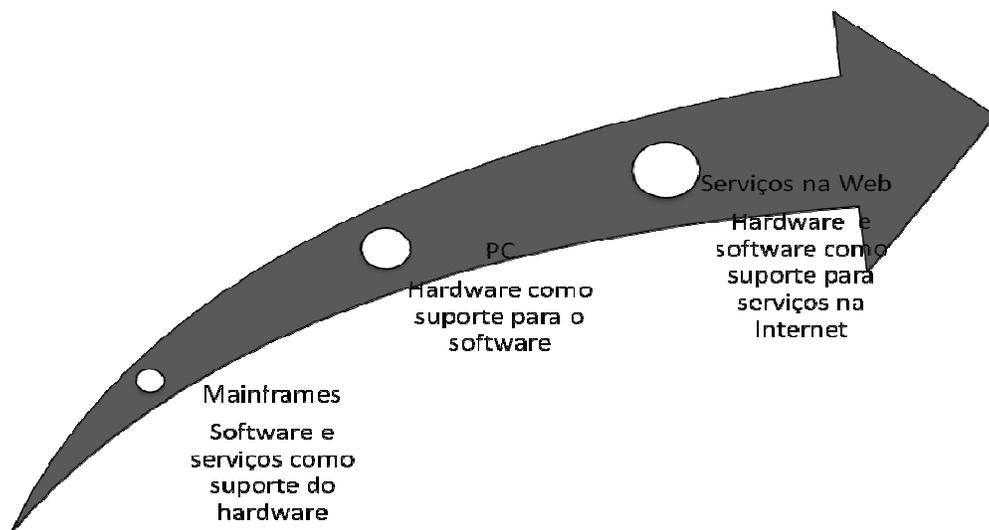


Figura 2.1 – Regimes tecnológicos na indústria de software

Fonte: Elaboração própria.

³⁰ Termo usado para o computador pessoal de mesa em contraponto aos computadores móveis como *laptops* e *notebooks*.

Ao longo dos regimes tecnológicos, as empresas desenvolveram modelos de negócios de acordo com as diferentes condições de oportunidade e de apropriabilidade, graus de cumulatividade do conhecimento tecnológico e complexidade da base de conhecimento. De uma forma muito esquemática, esboçada na Figura 2.2 que será trabalhada ao longo deste capítulo e do próximo, proporcionalmente os modelos de negócio baseados em serviços foram crescendo ao longo do tempo, enquanto os modelos baseados em produtos de software tiveram um boom de crescimento no regime do computador pessoal (PC), ritmo que tem diminuído no regime de transição atual.

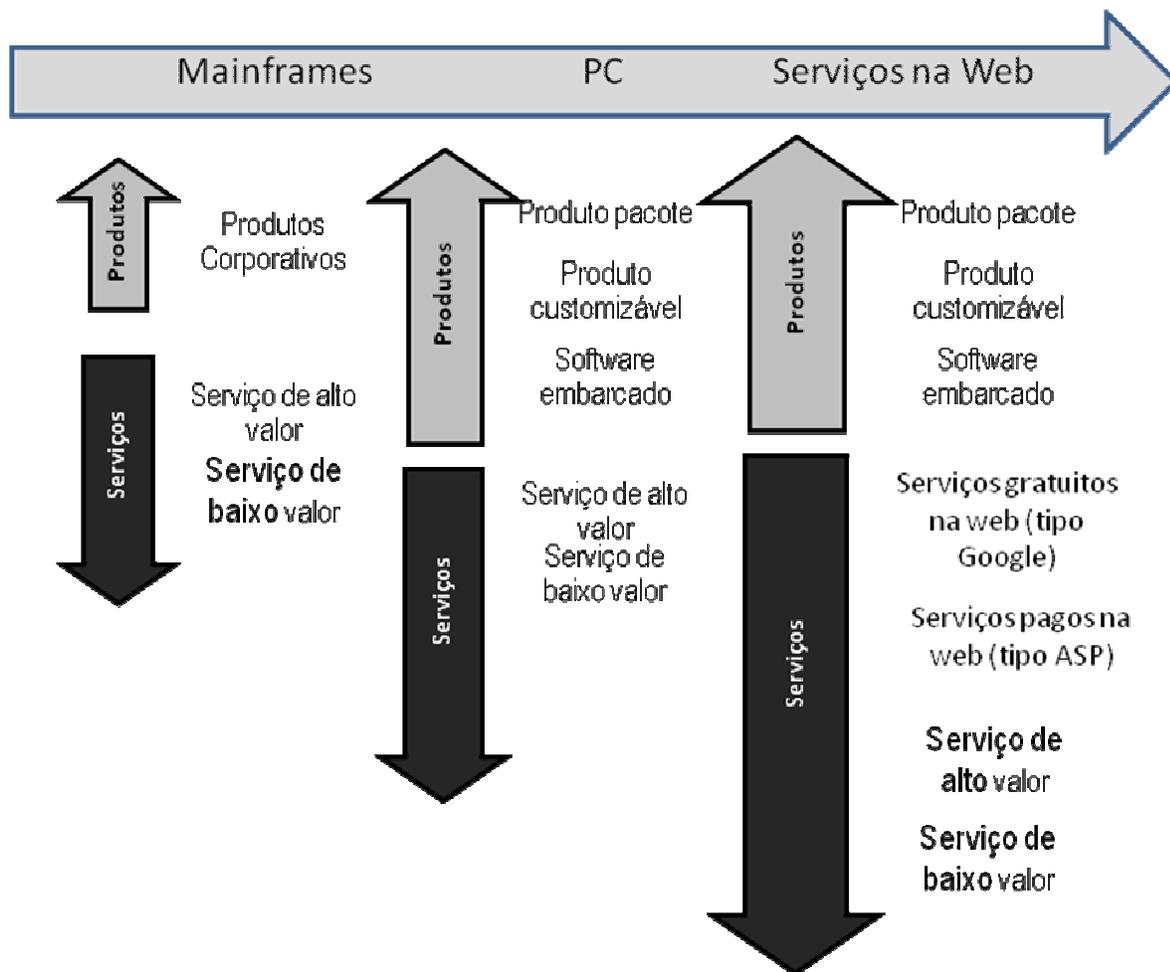


Figura 2.2 – Desenvolvimento dos modelos de negócio nos regimes tecnológicos da indústria de software

Fonte: Elaboração própria.

Para mostrar a co-evolução da indústria e das formas de apropriação dos esforços inovativos, a história é contada em torno dos regimes e dos modelos de negócios:

- software como atividade complementar aos grandes computadores;
- grandes computadores com arquitetura padronizada que possibilitam o surgimento dos modelos de negócio de software customizado e software-produto;
- a revolução do PC que torna o hardware suporte para o software, especialmente para o software produto;
- a “revolução da Internet”, que commoditiza o software e o torna suporte para serviços disponíveis na Internet.

Neste capítulo, num primeiro momento são analisados os três primeiros períodos da história, ficando a análise do quarto período para o capítulo 3. Tendo em mente que a regulação da proteção da propriedade intelectual segue o desenvolvimento tecnológico a alguns passos de distância, a segunda parte do presente capítulo é dedicada à discussão da conformação da proteção jurídica da propriedade intelectual relacionada com o software, bem como dos instrumentos não passíveis de registro utilizados nos diferentes modelos de negócio. A indústria de software apresenta uma grande variedade de atividades no seu interior ao que deve corresponder uma variedade das formas de apropriação do valor gerado pelas inovações.

O capítulo tem como eixo a história da indústria nos EUA devido ao papel de liderança deste país desde a origem da indústria até os dias atuais, construída por fatores históricos, institucionais e econômicos, dentre eles, o fornecimento para o mercado interno, o mais sofisticado do mundo, com vantagens de escala (Roselino, 2006). E também, pelo papel preponderante dos EUA na construção da regulação da proteção da PI internacionalmente.

2.1 Evolução da indústria de software e de seus modelos de negócio até 1994

Segundo Malerba e Orsenigo (1996), o desenvolvimento da indústria de computadores, em seu início, foi caracterizado por um regime tecnológico com condições de alta oportunidade, alta

apropriabilidade, alta cumulatividade e também crescente pervasividade e alta complexidade da base de conhecimento. Esta parte da tese trata justamente do período que vai da fase embrionária do software à popularização da Internet em meados da década de 90.

O limite do recorte temporal em 1994 foi tomado com um fim didático, pois a Internet neste período já possuía mais de 3 milhões de servidores e 13 milhões de usuários³¹, localizados principalmente nos EUA e outros países desenvolvidos. O ano de 1994 é tomado aqui como um marco do início da popularização e disseminação (desigual) nos demais países.

2.1.1 Software como atividade complementar ao hardware

A etapa embrionária da indústria de software iniciou-se com a construção dos primeiros computadores não comerciais na década de 40, por iniciativa do Departamento de Defesa dos EUA e da NASA (National Aeronautics and Space Administration), e estendeu-se até os anos 60 quando houve a introdução de chips reproduzíveis em grande escala, fato que teve papel fundamental no aumento da capacidade de processamento e armazenamento de dados e na diminuição dos custos de produção dos computadores (Roselino, 1998). A indústria de software surgiu, então, umbilicalmente ligada à indústria de hardware, ambas relacionadas a objetivos científicos e militares e não propriamente de negócios (Steinmueller, 1996). O papel do governo dos EUA, por meio de encomendas, foi marcante na conformação da indústria.

As estratégias das empresas produtoras dos primeiros mainframes – IBM, Univac-Sperry, NCR, Control Data, Honeywell, Burroughs, RCA e GE (General Electric Company) – consistiam em introduzir novos produtos que eram então aperfeiçoados por meio de inovações incrementais. Segundo Malerba e Orsenigo (1996, p.54), foi um período de condição alta de apropriabilidade, “devido à presença da proteção legal [no caso, patentes para as invenções do hardware], à relevância do conhecimento tácito no desenvolvimento e produção de computadores e à importância dos ativos complementares, como assistência técnica e redes de distribuição”.

A IBM era a líder mundial na produção de computadores de grande porte, fornecendo, junto com

o equipamento, os serviços de programação e as ferramentas de programação, que eram distribuídas sem cobrança adicional e sem proteção formal, dado que o principal mecanismo de apropriação dos resultados da inovação estava no hardware. Desta forma, o software tinha um papel secundário, sendo entendido como atividade complementar à comercialização dos computadores.

Como era um adicional de “menor” importância, o programa de computador também era produzido pelos próprios usuários, pois não havia disponibilidade de aplicativos para todas as suas necessidades (Campbell-Kelly, 2003). Nesta época, a estrutura favorecia mais a opção pelo “make” do que pelo “buy” no processo de aquisição de software, como apresenta Steinmuller (1996, p. 22): “Produtores de sistemas de computador que ofereceram serviços e softwares para estimular o uso de computadores, usuários que desenvolveram aplicações para seu próprio uso e usuários que cooperaram em troca de rotinas e métodos de programação formaram a organização econômica prévia para o desenvolvimento de atividades de software”.

Os relativamente poucos usuários criaram grupos para cooperar e intercambiar rotinas de programação. Um dos principais grupos foi o de usuários dos computadores da IBM, denominado SHARE, que teve papel importante no processo de estabelecimento de um conjunto de ferramentas agindo como condutor, por meio do qual as necessidades dos usuários eram direcionadas ao departamento de programação da IBM (Campbell-Kelly, 2003). Os produtores de hardware, por sua vez, viam o desenvolvimento deste ativo complementar pelos usuários com bons olhos, incentivando-os e lhes dando suporte.

Estes grupos representam uma característica importante da indústria de software, qual seja, a participação dos usuários finais no desenvolvimento do software de uma forma mais ou menos intensa, como será discutido ainda neste capítulo e no próximo. Isto é interessante porque coloca questões de propriedade, especialmente no momento em que os desenvolvedores individuais têm um papel mais destacado com o advento do software livre e de código aberto (SL/CA).

³¹ Fontes principais: Internet Systems Consortium <<http://www.isc.org/index.pl?/ops/ds/host-count-history.php>> e University of Virginia Webstat <<http://www.virginia.edu/webstats.html>>.

Em 1960, foram lançadas duas linguagens de programação de maior complexidade – Fortran e Cobol. O lançamento destas linguagens facilitou a atividade de desenvolver software pelos usuários, o que atrasou um pouco o crescimento de um mercado externo para software, pois as linguagens possibilitavam ganhos de produtividade por meio do desenvolvimento *in-house* (Steinmueller, 1996, p.23).

As empresas de serviços, seja serviços de processamento ou gerenciamento dos serviços de infraestrutura corporativa, surgiram nos anos 60, sendo que as de maior sucesso foram a Electronic Data Systems (EDS) e a Automatic Data Processing (ADP) (Campbell-Kelly, 2003), que existem ainda hoje. Os serviços oferecidos eram opções alternativas para a compra de computadores e desenvolvimento *in-house* ou para compra de software (desenvolvimento sob encomenda).

2.1.2 Software customizado e primórdios do software-produto

O período de 1965 a 1970 marcou o nascimento da produção de software como atividade autônoma com o surgimento dos primeiros fornecedores independentes de software (*Independent Software Vendors - ISVs*). Segundo Steinmuller (1996), três acontecimentos foram importantes para isso.

O primeiro foi o lançamento pela IBM, em 1965, da primeira família de computadores *mainframes* modulares compatíveis (System/360). Do lado do hardware, a introdução de computadores modulares permitiu a padronização de componentes, ganho de escala na produção de componentes específicos, o que teve como consequência o aumento das barreiras à entrada para novos entrantes no mercado de *mainframes*, mas diminuiu as barreiras para os fornecedores dos novos periféricos (Malerba e Orsenigo, 1996). Já do lado do software, a família de computadores proporcionou a primeira base instalada de computadores com um único sistema operacional. A família System/360 era formada por computadores de menor custo, acessíveis não apenas às grandes empresas, o que provocou uma onda de informatização em empresas de porte médio. Desta forma, os nascentes ISVs tiveram pela primeira vez a oportunidade de vender o mesmo produto de software para uma variedade de usuários.

É interessante fazer um pequeno parêntese aqui para discutir a inovação que representa o produto

de software. Como já mencionado, as atividades de software são classificadas como serviços pelos órgãos estatísticos oficiais. A introdução do conceito de produto em um setor de serviços muda a própria noção de produto.

O segundo acontecimento foi o desenvolvimento da indústria de minicomputadores a partir do lançamento deste tipo de equipamento, em 1965, pela DEC (Digital Equipment Corporation). Os minicomputadores tinham diferentes usos (computador primário, interface de mainframe, sistemas de comunicação de dados, sistemas de controle de processos) e requeriam programas muito diferentes. Assim, a demanda por software para minicomputadores era alta, mas a diversidade das aplicações tornava limitados os mercados para software vendido como um produto, o que restringia as economias de escala. Assim, a estrutura da indústria de software para minicomputadores era semelhante à de *mainframe*, ou seja, muito software produzido sob encomenda, produção in-house e pouco software produto. Isto era muito diferente do mercado de massa que surgiria com o computador pessoal (PC) nos anos 80 (Steinmuller, 1996).

O terceiro acontecimento foi a decisão da IBM de separar a venda do hardware e do software em 1969, o que representou um importante marco na história do software. A decisão aconteceu por uma série de motivos. Por um lado, à medida que os programas de computador tornaram-se maiores e mais complexos, alterou-se a relação de custo entre a produção de hardware e software. O desenvolvimento do software ficou mais custoso em termos de P&D³². Isso influenciou a decisão da IBM de vender separadamente hardware e software, pois levou à reavaliação de seu compromisso de fornecer todas as ferramentas de software que os usuários necessitavam para comprar ou alugar seus computadores (Steinmuller, 1996). A IBM passou, então, a cobrar separadamente pelos seguintes itens além do hardware: engenharia de sistemas, treinamento, serviços de programação e software pacote. Apenas o sistema operacional foi mantido integrado à venda do hardware, o que manteve o monopólio da IBM no mercado de *mainframes* até o fim dos anos 70, quando também o sistema operacional passou a ser vendido separadamente (Campbell-Kelly, 2003).

³² O tortuoso projeto do sistema operacional OS/360 da IBM, que levou muito mais tempo do que o planejado, apontou que produzir software e gerenciar grandes projetos requeriam esforços especiais. Isso levou inclusive, ao surgimento de uma nova disciplina, a engenharia de software, no fim dos anos 60 (Cusumano, 2004).

Por outro lado, em 1967, a IBM foi alvo de investigação pela Divisão Antitruste do Departamento de Justiça dos EUA. Nesta época, a IBM detinha 70% do mercado, mas isso em si não era problema; o problema residia no fato de vender conjuntamente e sem identificação de valor produtos inter-relacionados e necessários para os clientes. Além disso, vendia abaixo do preço de custo (Campbell-Kelly, 2003).

A separação da cobrança do hardware, software e dos serviços criou uma oportunidade ímpar para os ISVs estabelecerem um mercado para produtos de software (Campbell-Kelly, 2003). Até então, o mercado para software produto era muito pequeno. Como a IBM provia software “sem custo” para seus clientes, segundo a própria empresa, era difícil para empreendedores de software estabelecerem mercado (Graham e Mowery, 2003).

Aos poucos se assistiu ao surgimento da atividade autônoma de software. Em 1965, como apontado acima, grande parte dos usuários era também customizador de software, realizado internamente ou por uma empresa de serviços de programação. Em meados dos anos 70, apenas grandes usuários ainda utilizavam com frequência software customizado. As demais empresas, especialmente empresas médias, começaram a usar software produto³³.

A partir da separação, os ISVs começaram a competir em qualidade com produtos da IBM³⁴, com exceção do sistema operacional. Cresceram rapidamente também serviços de programação e de processamento (Steinmuller, 1996).

Com a expansão das vendas de mainframes e de minicomputadores, surgiu demanda de soluções para aplicações de software tanto horizontais como verticais. O software vertical é aquele específico para um determinado setor, como software para o setor financeiro, para transporte, educação etc., o que requer habilidades diferentes dos desenvolvedores, que necessitam conhecer a fundo as características, necessidades e regras de negócio daquele setor. Já o software

³³ Com a multiplicação de empresas especializadas no desenvolvimento e comercialização de software, criou-se um problema de busca, ou seja, achar o melhor produto para uma determinada aplicação. Em função disso, surgiram os catálogos de software, como *Datamation* (Campbell-Kelly, 2003).

³⁴ Após a separação da venda de hardware e software pela IBM, uma série de empresas lançou produtos, a partir de programas pré-existentes, desenvolvidos como ferramentas de programação *in-house* ou de contratos de software customizado (Campbell-Kelly, 2003).

horizontal são as ferramentas de programação, sistemas e aplicativos que não específicos por setor econômico.

Os mercados verticais tinham estrutura fragmentada e ampla distribuição geográfica. O sistema bancário foi um dos primeiros mercados verticais a se informatizar, por meio da utilização de pessoal interno complementado por programadores contratados e software sob encomenda (Campbell-Kelly, 2003).

Desta forma, surgiram as formas iniciais de três importantes modelos de negócio da indústria de software, ainda existentes, com diferentes regimes de apropriabilidade: software produto (também chamado software pacote), software sob encomenda (ou serviços de alto valor) e serviços de baixo valor (ex. programação).

2.1.3 A revolução do PC: hardware torna-se commodity para o software pacote de massa

Nos anos 80, as atividades de software por parte dos fabricantes de computadores caíram dramaticamente, com exceção da IBM, o que refletiu o avanço da autonomia da atividade de software e o amadurecimento dos ISVs. Segundo Steinmuller (1996), ocorreram três grandes acontecimentos neste período:

1. a emergência do mercado de computadores pessoais no começo da década, o que gerou um novo princípio organizador da indústria, qual seja, o software pacote de massa;
2. a introdução, no meio da década, das estações de trabalho (workstations) – equipamentos híbridos entre o PC e o minicomputador – o que levou à criação de uma nova grande classe de aplicações que exploraram a capacidade gráfica e de processamento numérico destas máquinas. Voltada para usuários sofisticados, o principal fabricante era Sun Microsystems, com sistema operacional Unix;
3. o crescimento rápido de empresas de integração de sistemas e terceirização das atividades de processamento de dados e gerenciamento corporativo, devido aos problemas de produtividade e organizacionais das empresas.

Abre-se outro parêntese para um breve relato da história do sistema operacional Unix, dado que foi o único sistema operacional não proprietário de maior significância e longevidade, inclusive para o surgimento do Linux, sistema operacional de código aberto, e da Internet. O Unix foi iniciado em 1969 em um consórcio envolvendo Bell Labs, General Electric e MIT (Massachusetts Institute of Technology) dentro de um projeto maior (MULTICS) que foi um desastre. Seus criadores disponibilizaram o código que foi sendo adaptado em universidades para diversos tipos de computadores – devido a sua natureza que era independente de uma máquina específica, simples e robusta – de forma que a adaptação exigia a reescrita de cerca de 5 a 10 % de código. Ao longo da década de 70, o Unix espalhou-se por universidades. Nos anos 80, sua disponibilidade em um grande número de máquinas diferentes facilitou a interconexão de computadores nos primórdios da Internet. Apesar de o código estar disponível em universidades, os fornecedores de software e hardware licenciavam o código da Unix Software Laboratories Inc. (subsidiária da AT&T que foi fatiada por decisão da comissão anti-truste em 1984) e adaptavam-no para sua plataforma.

Fechado o parêntese, O amadurecimento e a consolidação da indústria de software deram-se finalmente com a criação do mercado de computadores pessoais (PCs). A planilha eletrônica Visicalc transformou a percepção do computador pessoal em uma máquina de negócios. Junto com a planilha, vieram as aplicações de produtividade, como processadores de texto, bases de dados e software de comunicação (Campbell-Kelly, 2003).

A redução de preços e o aumento da capacidade de processamento e de armazenamento possibilitaram a expansão da base instalada de computadores e, assim, do mercado de software. Segundo Steinmueller (1996), o crescente mercado de microcomputadores possibilitou a criação de oportunidades de escala e de lucro para empresas desenvolvedoras de software, devido ao aumento da demanda de software produto, especialmente de aplicações horizontais. E também pelo aumento sem precedentes do mercado homogêneo para sistemas operacionais e aplicações, ou seja, um mesmo software tinha uma base muito maior para ser comercializada, o que abria possibilidades de ganhos de escala e também de externalidades oriundas das economias de rede.

Externalidades de rede estão relacionadas ao grau crescente de adoção de uma solução

tecnológica, ou seja, quanto mais adotada e difundida, mais utilidade tem para seus usuários, o que resulta em mais desenvolvimento e aperfeiçoamento técnico: “a vantagem da empresa *first-mover* tende a se intensificar na medida em que a empresa expande o número de usuários de seu produto” (Roselino, 2006, p.12).

Esta lógica é reforçada no software devido às economias de rede, pois o custo inicial de produção do software é alto, mas o custo de reprodução tende a zero. E também devido à necessidade cada vez mais presente de interatividade, isto é, de troca de dados entre usuários. Desta forma, o processo de seleção do padrão dominante passa muitas vezes por características externas ao software, o que resulta que a melhor solução técnica não seja necessariamente a que se torne o padrão dominante (Roselino, 2006; OCDE, 1997; Tigre, 2006; Gutierrez e Alexandre, 2004). Os efeitos resultantes das externalidades positivas de rede resultam num efeito de *lock-in* de mercados, como apresenta Tigre para as redes virtuais (2006, p.246):

*Os efeitos de rede dão origem ao **feedback positivo**, um processo que fortalece ainda mais as tecnologias ou padrões que se tornaram dominantes no mercado. As redes com poucos usuários tendem a desaparecer, ao passo que aquelas que despontam como vencedoras tendem a dominar todo o mercado. O êxito alimenta a si mesmo, produzindo um círculo virtuoso conhecido como a lógica do ‘vencedor leva tudo’ [grifo do autor].*

O efeito de *feedback* positivo é ampliado devido ao custo de mudança para o usuário de sua opção atual frente a uma tecnologia alternativa, aos custos afundados (*sunk costs*) e aos custos de aprendizado.

Como apontado no capítulo 1, a IBM lançou seu computador pessoal em 1981, numa estratégia de *imitator-seguidor*, pois outras empresas já tinham lançado modelos com o mesmo propósito (Teece, 1986). Entretanto, a IBM tomou uma decisão arriscada, mas que revolucionou a indústria de computadores: terceirizar quase todos os componentes e subsistemas. Em termos tecnológicos, não se tratava do melhor equipamento, mas de um computador com uma arquitetura comum e componentes padrão. Formou-se o que ficou conhecido como tripé Intel-IBM-Microsoft, sendo que cada uma das empresas dominava uma das bases: a IBM era responsável pela arquitetura dos

computadores pessoais, a Intel pelo padrão dos microprocessadores e a Microsoft pelo sistema operacional.

A arquitetura aberta do PC da IBM levou à convergência de padrões tecnológicos e ao surgimento dos clones compatíveis com estes padrões:

O PC da IBM foi desenhado como uma máquina aberta para que os outros fornecedores (não da IBM) pudessem produzir hardware e software adicionais. Na medida em que a máquina ganhou aceitação, fabricantes produziram clones compatíveis com o da IBM em grande quantidade. Por volta de 1983, próximo de um milhão de máquinas compatíveis com o da IBM foram vendidas, 90 por cento delas com cópia do MS-DOS (Campbell-Kelly, 2003, p.207).

O sucesso do PC da IBM não tinha a ver tanto com a tecnologia, mas com os ativos complementares estabelecidos em torno dele, como comentou Teece no seu famoso artigo (1986, p.299):

A IBM poderia ter baseado seu sistema de PC em seu hardware patentado e seu software protegido por copyright. Tal abordagem causaria a co-especialização de produtos complementares, forçando a IBM a produzir periféricos e uma biblioteca abrangente de software em um curto período de tempo. Ao contrário, a IBM adotou o que pode ser chamado de uma abordagem ‘contratual induzida’.

Desta forma, a IBM conseguiu reunir os ativos complementares sem o uso de contratos e de integração. A IBM ajudou, então, a estabelecer um grande novo mercado, mas paradoxalmente não conseguiu aproveitá-lo por muito tempo, pois propositalmente só tinha controle de um dos ativos complementares – a marca IBM. À medida que surgiram outras marcas de PC que licenciaram o sistema operacional DOS da Microsoft, a vantagem competitiva da IBM foi perdida e o tripé Intel-IBM-Microsoft transformou-se no dueto *Mintel*. Como no esquema de Malerba e Orsenigo (1996), em condições de baixa apropriabilidade, o controle dos ativos complementares é importante. Além disso, a IBM não conseguiu perceber o papel relevante que o software vinha adquirindo e continuou em sua estratégia centrada no hardware, o que lhe custou

uma dura jornada, revertida apenas em meados da década de 90 com a migração para os modelos de serviços.

Posteriormente, tanto a Intel quanto a Microsoft cresceram e estabeleceram posições monopolistas. Além disso, a expansão dos desktops com arquitetura aberta criou grande e crescente mercado para aplicações de software durante os anos 80, o que permitiu que novos entrantes, como Microsoft e Adobe Systems ocupassem espaço, enquanto incumbentes, como IBM e Texas Instruments tivessem um papel menor (Graham e Somaya, 2004).

Neste momento foi estabelecida a forma de comercialização do software que tornaria a Microsoft “uma máquina de fazer dinheiro” (Campbell-Kelly, 2003, p.205). Segundo Roselino (1998), o MS-DOS inicialmente era vendido para empresas de hardware mediante um preço fixo, independente da quantidade de máquinas em que fosse instalado. Em 1983, a Microsoft alterou o formato de comercialização passando a associar o valor cobrado ao nível de produção do licenciado e, no final da década de 80, passando a cobrar a licença por CPU (Unidade Central de Processamento).

A disseminação dos PCs levou à proliferação de aplicações e fornecedores independentes nos EUA e Europa. A variedade de aplicações necessárias para a ampla base de PCs possibilitou que os ISVs pudessem se especializar, pois foram criadas janelas de oportunidade a partir da divisão de trabalho entre produtores de software (Torrise, 1998). A demanda que surgiu ultrapassou a capacidade de fornecimento dos produtores estabelecidos, o que foi uma importante fonte de dinamismo e oportunidade para novos entrantes (Graham e Mowery, 2003).

Neste período, surgiram quatro empresas que em pouco tempo conseguiram penetrar em uma grande fatia da base instalada de PCs com um único produto cada, mas que se tornaram padrão para aquele tipo de aplicação devido às externalidades de rede: Lotus para planilha eletrônica (Lotus 1-2-3), Microsoft para sistemas operacionais (MS-DOS), Ashton-Tate para bases de dados (dBase) e WordPerfect para processador de textos (WordPerfect) (Steinmuller, 1996).

Começava aí a situação na qual algumas grandes empresas incumbentes podiam forçar seu produto para o resto do mercado, impondo-os como padrão, usando os benefícios das

externalidades de rede para estabelecer um monopólio, como mostrarão as ações anti-truste contra a Microsoft no final dos anos 90 (Coriat e Orsi, 2002)

As externalidades de rede explicam como um único produto tornou-se padrão para uma determinada classe de aplicação. Segundo Steinmuller (1996, p.35-37), além dos pontos já colocados, a acumulação de habilidades, material de treinamento e produtos adicionais (*add-ons*) facilitam o uso e reforçam a aplicação dominante, que depende também das tecnologias complementares. Por exemplo, no mercado de planilhas eletrônicas, a Lotus apostou no sistema operacional OS/2 da IBM, mas o Windows se tornou padrão e o Excel preencheu o vazio deste tipo de aplicação para este sistema operacional. Aconteceu o mesmo na disputa WordPerfect vs. MS Word.

A Microsoft passou a incorporar em seus produtos as funções de outros programas por meio do desenvolvimento interno ou de aquisições (Campbell-Kelly, 2003). Foi assim com as funcionalidades relacionadas com redes de computadores, por meio das quais passou a competir com a Novell logo mais tarde.

A revolução do PC aumentou também a massa de desenvolvedores individuais, isto é, profissionais com boa formação técnica que desenvolvem software em empresas produtoras e usuárias de software, mas também fora de empresas. Esta característica não possui paralelo em outra indústria. Neste período, o SL/CA começou de forma tímida uma nova trajetória que irá se expandir com a formação das redes de computadores com papel importante dos usuários, como será discutido no capítulo 3.

Ainda neste período, as grandes corporações começaram a ter problemas para integrar diferentes plataformas (minicomputadores, mainframes e PCs) e diferentes aplicações com a finalidade de processamento de dados. Tratava-se de uma tarefa complexa realizada internamente, mas que consumia recursos importantes das empresas usuárias. Empresas que ainda tinham desenvolvimento *in house* então reconsideraram a decisão de “*make-or-buy*”, optando pela compra externa, devido à crescente complexidade das tecnologias de processamento de dados.

Surgiram, então, empresas voltadas para integrar os diversos sistemas e plataformas das

empresas, serviço oferecido conjuntamente com consultoria, treinamento e gerenciamento dos serviços de infra-estrutura corporativa (Steinmuller 1996), que se somaram às empresas já existentes, originalmente estabelecidas como empresas de serviços computacionais, voltando-se para o desenvolvimento de grandes operações de integração de sistemas. Foi criada uma nova forma de fornecer serviços, feita em parceria com o cliente, de forma que a solução era específica para a organização, mas poderia ser adaptada para outros clientes:

A grande organização de serviços de computador é capaz de amortizar os custos fixos do desenvolvimento de software de larga escala e a acumulação de competências técnicas específicas em hardware e software por meio de um grande número de usuários. A organização de serviços de computador pode também ter um impacto significativo no fornecimento de software pacote pela negociação de licenças multi-clientes e requerimento de modificações específicas para satisfazer as necessidades dos clientes (Steinmuller, 1996, p.40).

Retomando a evolução da indústria, quando emergiu o regime dos grandes computadores, as empresas que produziam máquinas “semelhantes” (IBM e Texas Instruments) tiveram vantagens, mas quando emergiu o novo regime do PC, o quadro não se repetiu. A despeito de ter sido uma imitadora rápida, em relação à Xerox e à Apple, e lançado o PC de sucesso fabuloso no início, a IBM não conseguiu desenvolver as habilidades necessárias para se manter na dianteira. Com o surgimento da nova tecnologia, as empresas entrantes conseguiram abocanhar um pedaço do mercado.

2.2 Sistematização dos modelos de negócio da indústria de software

Como apresentado, o mercado de software-produto pode ser dividido em corporativo e de massa. Em geral, o software produto corporativo é comercializado no modelo de negócio de software customizável, ou seja, trata-se de um software pronto, mas que necessita de alguns ajustes para adaptá-lo ao cliente. Já o software produto de massa – também chamado de software pacote e geralmente relacionado com aplicações horizontais fortemente ligadas com a indústria de software para computadores pessoais – é comercializado sob licenças (proprietárias), de acordo

com a quantidade de máquinas onde é instalado, entre outras formas. O software customizável também é licenciado, mas o valor unitário é geralmente muito mais alto que o software pacote, além da cobrança pelos serviços de consultoria (pré-venda), customização, suporte e manutenção pós-venda.

Campbell-Kelly (2003) compara os dois modelos ao mercado de publicações e ao de bens de capital. Enquanto o mercado de massa lembra o mercado de publicações e entretenimento – devido aos altos custos de desenvolvimento da primeira cópia, baixos custos de reprodução e da importância dos canais de distribuição (por meio de distribuidores mais do que venda direta) e suporte – o mercado de software corporativo estaria mais próximo do mercado de bens de capital – devido às capacidades críticas em exploração de vendas, marketing corporativo, qualidade e suporte pré e pós-vendas.

Segundo o mesmo autor, é possível distinguir duas estratégias das empresas no mercado de software corporativo: de um lado, aquelas que buscaram construir portfólios de soluções verticais completas para uma determinada área por meio da integração de produtos desenvolvidos internamente ou adquiridos. Estes grupos de produtos eram vendidos como uma unidade, o que facilitava a compra e instalação. São exemplos a IBM e a Computer Associates (CA). Esta estratégia dos grandes consolidadores de criar grande portfólio de produtos visava prover todo o software necessário para uma organização, o que exigia investimento em P&D e em marca para integrar os pacotes em uma família de produtos.

De outro lado, estavam as empresas que comercializavam um produto único, como Oracle para bases de dados relacionais e a empresa alemã SAP AG (Systeme, Anwendungen und Produkte - Sistemas, Aplicações e Produtos) para software de sistema integrado de gestão empresarial (ERP). O custo de implantar o ERP era de 5 a 10 vezes o custo do software, pois a instalação era complexa e havia o custo da consultoria prévia.

Já no mercado de software pacote de massa, os *first-movers* tiveram vantagens competitivas que foram se intensificando na medida em que estas empresas ampliaram sua base de usuários. Estas vantagens associaram-se às “externalidades positivas da difusão de programas e padrões

tecnológicos no mercado de software, resultando num significativo efeito de *lock-in* dos mercados já ocupados por uma ou poucas empresas” (Roselino, 1998, p.19). O exemplo mais ilustrativo desta situação foi a Microsoft, que em pouco tempo cuidou de fazer o casamento do seu sucesso de vendas, o sistema operacional Windows, com os aplicativos, além da integração entre os aplicativos básicos com lançamento das suítes integradas, como o MS-Office, que reúne processador de texto, planilha eletrônica, base de dados entre outros. Esta estratégia foi seguida por outras empresas, como IBM e Novell, que juntas chegavam a 30% de todo software pacote, segundo estimativas da OCDE (1997).

Esta estratégia foi complementada com uma espécie de pirataria consentida. Segundo Roselino (1998, p.57), esta prática foi utilizada no segmento de software pacote e interativo, ou seja, quando ocorre alguma forma de interação com trocas de dados entre diferentes software e hardware:

A reprodução ilegal não-sistemática, conduzida por iniciativas individuais dos usuários finais pode servir como elemento promotor de um processo de imposição daquele software como padrão dominante em determinados segmentos. Esta prática pode ser funcional para a estratégia competitiva naqueles segmentos de mercado em que esta característica [a interatividade] é determinante da competitividade do software.

A lógica desta prática é renunciar a receitas potenciais no curto prazo visando a realização de lucros advindos de posições monopolistas no futuro. Com estas estratégias, a Microsoft conseguiu atingir, em 1998, quase 80% de participação no mercado dos sistemas operacionais dos computadores pessoais e também das suítes de escritório. O Gráfico 2.1 apresenta a participação no mercado de sistemas operacionais em 1998. Em 2004, esta participação ultrapassou 90% do mercado.

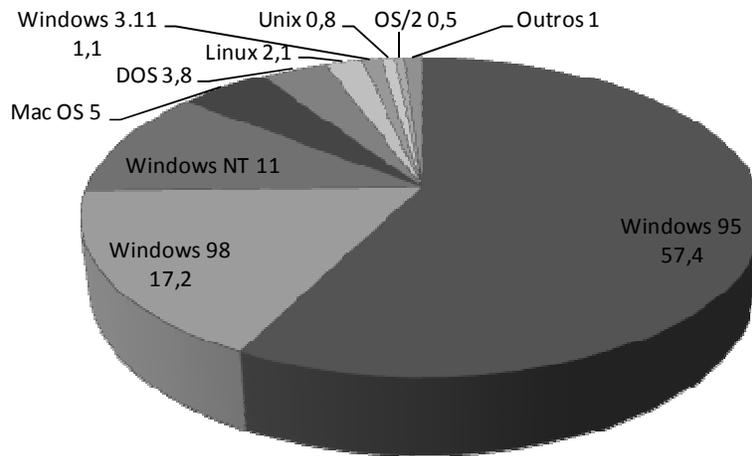


Gráfico 2.1 - Participação no mercado dos Sistemas Operacionais, em 1998

Fonte: IDC (International Data Corporation) disponível em <http://news.com.com/2100-1040-228773.html>

Passa-se agora para a sistematização das características de cada modelo de negócio. Como apresentado, os modelos de negócios podem ser pensados como oscilando entre duas extremidades da escala de serviços a produtos, como proposto por SOFTEX (2003) (figura abaixo). Alguns modelos são mais fortemente baseados em produtos e outros mais em serviços, mas essa fronteira pode ser alterada em cada regime tecnológico. Aqui são adotados os cinco modelos de negócio de SOFTEX (2003), com exceção do modelo de componentes, que será objeto do capítulo 4.

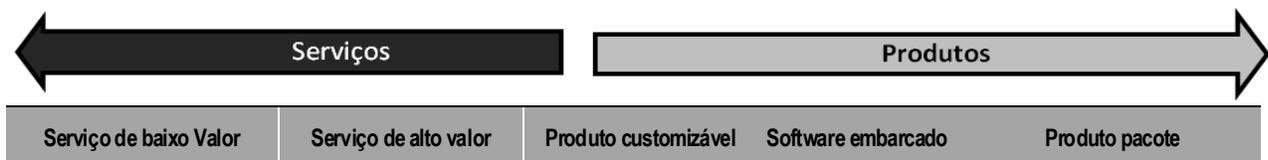


Figura 2.3 – Esquema dos modelos de negócio na indústria de software

A caracterização destes cinco modelos foi enriquecida com o trabalho de Roselino (2006), que trabalhou com três modelos de negócios, dado os objetivos de seu trabalho: serviços em software de baixo valor agregado, serviços em software de alto valor agregado, e software produto.

As principais características dos modelos são apresentadas no Quadro 2.1.

Quadro 2.1 – Características dos principais modelos de negócio da indústria de software

Características	Serviços		Produtos		
	Serviço de baixo Valor	Serviço de alto valor	Produto customizável	Software embarcado	Produto pacote
custo marginal de + 1 venda	virtualmente constante	virtualmente constante	menos que constante	mais que zero	virtualmente zero
Estrutura de mercado	fragmentada com pequenas barreiras à entrada	mercado fragmentado com variedade de nichos e maiores barreiras à entrada	fragmentada com variedade de nichos, mas concentrada em alguns produtos	Fragmentada com variedade de nichos	mercado concentrado
Relação com cliente	um para um	um para um com estreita interação com o cliente	um para vários com estreita interação com o cliente	um para vários	um para vários, interação com cliente é ínfima, produtos-padrão com pouco suporte
Modelo de venda	direto	direto	direto, VARs e varejo	direto	VARs e varejo
Objeto de venda	projeto ou recurso + manutenção e suporte	projeto + manutenção e suporte	licença e serviços de customização + manutenção e suporte	licença e pequena adaptação	licença
Variável chave	custo (mão de obra)	utilização de capacidade	número de clientes		quota de mercado (base de usuários)
Risco do empreendimento	baixo	baixo			alto devido aos custos elevados de desenvolvimento e intenso P&D
capacidade crítica	processo	processo e relação com cliente	análise de requisitos, relação com cliente e tecnologia	relação com cliente e tecnologia	produtos com ciclos de vida curtos e com inovações incrementais
externalidades de rede e economias de escala	pequenos ganhos de escala	reuso do software pode levar a ganhos de escala*			externalidades criam padrões dominantes que reforçam poder de mercado dos incumbentes
Barreira à entrada	competição	reputação	reputação e tecnologia	tecnologia	investimento
Exemplo	integração de sistemas, processamento de banco de dados para terceiros	desenvolvimento customizado	ERP, CRM, produto vertical		processadores de texto
Empresa típica	Tata Services Consulting	IBM	SAP, Oracle	Ericsson	Microsoft

Fontes: Elaboração própria a partir de Roselino (1998; 2006), Allison *et al* (2006), SOFTEX (2002; 2005), Torrissi (1998), Merges (1996).

Nota: *O reuso será discutido no capítulo 4.

Os modelos de serviços podem ser divididos em serviços de alto valor, também conhecidos como software sob encomenda, e serviços de baixo valor. As principais características dos modelos de serviços são a relação direta com o cliente e os serviços oferecidos sob medida, sem grandes possibilidades de ganhos de escala. No serviço de baixo valor, a competição é por custo, enquanto no serviço de alto valor conta bastante a reputação da empresa, a relação com o cliente e a qualidade dos processos.

Já os modelos de software produto são mais intensivos em P&D e são empreendimentos de maior risco, pois o investimento e especificação são de responsabilidade da empresa de software. Além do ativo tecnológico relevante, os ativos complementares de marketing, distribuição e suporte são extremamente importantes.

2.3 A formação da proteção jurídica da propriedade intelectual para software

Para uma indústria tão heterogênea, certamente não será possível ter apenas uma solução única em termos de instrumentos para proteção dos direitos de propriedade intelectual, sejam formais ou informais.

Dada a evolução histórica que culminou na separação da venda de hardware, software e serviços e com a conseqüente autonomia da atividade de software, analisa-se a seguir como diferentes empresas se preocuparam em proteger seus heterogêneos ativos intangíveis. Observa-se que para cada modelo de negócio havia uma necessidade diferente de proteção. Os mecanismos de proteção variam também de acordo com o país.

As diferenças da regulação da propriedade intelectual entre os países dependem, entre outros fatores, **da composição da indústria** em termos de modelos de negócios e das pressões para modificações no regime legal:

Embora a indústria de computador seja global em sua natureza, diferenças significativas restam entre as indústrias de software e o regime de propriedade intelectual associado às economias industriais. Os Estados Unidos, Japão e Europa Ocidental diferem na extensão na qual o consumo e produção doméstica de software são dominados por software pacote. Como resultado, o lobby doméstico para a criação e modificação de regimes legais

cobrando esta relativamente nova forma de propriedade intelectual contribuiu para diferenças no nível e nas características dos direitos de propriedade intelectual para programas de computador entre as grandes economias industriais (Graham e Mowery, 2003, p.7_1-2).

A estrutura e a evolução da indústria de software de um país dependem de eventos históricos e institucionais particulares. Como o software possui custos marginais de reprodução muito baixos ou desprezíveis, a sociedade deveria garantir aos produtores alguns direitos para controlar a reprodução, se se espera que sejam feitos investimentos na criação de software, especialmente em software pacote. Desta forma, a proteção da propriedade intelectual é uma política chave que pode influenciar a estratégia e evolução desta indústria, bem como “a influência política dos fabricantes de hardware, software customizado e provedores de serviços e produtores de software pacote podem gerar modificações na estrutura da proteção da propriedade intelectual relacionada com software” (Steinmuller, 1996, p.17).

Procura-se aqui discutir as controvérsias sobre a proteção legal e apresentar dados de alguns estudos sobre as formas de apropriação (os estatutos legais de copyright, patentes e segredos de negócio, bem como os demais instrumentos não passíveis de registro, como mecanismos técnicos de proteção, inovação rápida, entre outros) segundo os modelos de negócio (produtos e serviços e, quando possível, os modelos mais detalhados – software embarcado, software pacote, produto customizável, serviços de alto valor e serviços de baixo valor).

2.3.1 Copyright vs. Patente?

As inovações na indústria de software demandaram mudanças e adaptações no marco legal da propriedade intelectual. Desta forma, foram sendo criadas novas figuras no sistema jurídico para abarcá-las, bem como as figuras existentes tiveram seu escopo ampliado. O aparelhamento jurídico leva tempo para se adaptar e tentar regular as novas situações (Yamamura, 2006, p.4). A área de software representa um bom laboratório para observar estas mudanças (Graham e Mowery, 2003).

Desde o início da regulamentação da proteção do software, sempre houve polêmica sobre a forma

mais adequada de se garantir a propriedade. Segundo alguns autores, o copyright foi inicialmente adotado sob a regra da dúvida, principalmente por se tratar de um item funcional e utilitário que introduziu certos elementos estranhos a este tipo de regulação (Cerqueira, 1999; Santos, 2003).

Talvez o formato mais adequado tivesse sido um tipo de proteção *sui generis* inteiramente personalizado para o software e suas características tecnológicas. Tal formato chegou a ser sugerido pela Organização Mundial de Propriedade Intelectual (OMPI), como resultado dos trabalhos de um grupo consultivo por ela criado em 1971 para debater a proteção dos programas de computador (Moniz, 2004).

Sempre foi senso comum que o software era fácil de replicar, o que não representava um problema para aquele feito sob encomenda (também chamado customizado) nos primórdios da indústria. Este tipo de software era feito de forma específica para uma organização o que, portanto, dificultava seu uso por outra. Os programas de computador eram objeto de contratos entre partes determinadas, normalmente o vendedor e o comprador do hardware.

A liderança dos EUA em termos de desenvolvimento da indústria de software refletiu-se na liderança em termos do estabelecimento da proteção formal. Em 1964, o Congresso dos EUA aprovou uma lei permitindo proteger o software por copyright, desde que atendesse os requisitos aplicáveis às demais obras intelectuais, como a originalidade e a fixação em forma legível para o ser humano³⁵ (Santos, 2003).

Há uma discussão extensa sobre a adequação da proteção via copyright para o software, pensado de forma mais ampla do que apenas o código-fonte e elementos associados (interface, documentação etc). A lei tradicional de copyright (que protege a expressão e não as idéias) não se aplica totalmente ao software, que tem as características de ser criativo e funcional ao mesmo tempo. Além disso, é difícil estabelecer a distinção entre idéia e expressão no software (Jussawalla, 1992). Segundo Santos (2003, p.393), há um dilema na proteção autoral para o software:

Na verdade, o grande dilema da proteção autoral é que ela reprime a apropriação ilícita da forma de expressão, enquanto o que, no fundo, precisa de tutela legal em matéria de programa de computador é o processo ou método implementado, independente da forma expressiva. Em outras palavras, a questão envolve a proteção da tecnologia e dos investimentos realizados contra atos não só de reprodução, mas acima de tudo de imitação (...)

O maior investimento feito pelas empresas de software reside principalmente no projeto do software e não na codificação, que é relativamente padronizada. Isso explicaria a atividade de patenteamento no software:

Não é por outra razão que, na mesma década em que se harmonizou a opção autoral, consolidou-se a tendência de estender a proteção patentária a programas de computador. A aplicação desse mecanismo legal resulta da constatação prática de que a tutela pelo Direito do Autor atualmente existente é simplesmente insuficiente para assegurar ao autor a consecução do objetivo primordial perseguido. Com efeito, as decisões que estendem a tutela autoral contra a reprodução da funcionalidade ou imitação do comportamento do programa configuram uma extrapolação dos limites naturais do Direito de Autor (Santos, 2003, p.394-393).

Em resumo, segundo Santos (2003), a proteção do software no âmbito do Direito de Autor conseguiu conseqüências opostas: deu uma proteção excessiva considerando-se os limites tradicionais e as exigências da sociedade e, por outro lado, teve o efeito de prover uma proteção insuficiente, dados os objetivos dos autores e a finalidade última de um regime de proteção.

Assim, o copyright provia uma proteção inadequada mesmo para o software pacote de massa, para o qual deveria ser o regime protetivo mais adequado, devido aos problemas de cópia por parte dos usuários (Jussawalla, 1992). O problema da cópia era, e continua sendo, especialmente sensível para a indústria de software, pois a mesma tecnologia que torna fácil e barato acessar,

³⁵ Apesar de soar estranho, o software possui duas formas básicas: o código fonte escrito em uma linguagem de computação, acessível portanto aos capacitados nesta linguagem, e o código binário, uma sequência de “zeros e uns” preparada de forma que uma máquina possa compreender.

armazenar e transmitir dados é utilizada para fazer cópias indevidas. Durante os anos 80 e 90, as decisões judiciais nos EUA trataram da definição do que seria cópia de um programa. Santos (2003) apresenta três gerações de casos:

- 1) na primeira geração, os tribunais trataram da proteção de programas de computador e do que seriam casos de cópias literais do código fonte;
- 2) a segunda geração de casos tratou da definição do âmbito da proteção e dos aspectos de cópia não-literal. Em alguns casos, a proteção foi estendida para a estrutura do programa (caso *Whelan vs. Jaslow*). Entretanto, no caso *Computer Associates vs. Altai*, o tribunal negou a proteção dos aspectos não-literais;
- 3) a terceira geração de casos tratou dos aspectos relacionados ao uso do programa de computador, especialmente a interface, dando origem aos casos relativos de “*look and feel*”.

O caso mais importante da terceira geração foi a ação da *Apple vs. Microsoft*. Quando a Microsoft desenvolveu o Windows 1.0, a Apple alegou que este programa copiava a interface gráfica do Macintosh. Para evitar litígio, assinaram um contrato de licenciamento. Entretanto, quando a Microsoft lançou o Windows 2.0 e o licenciou para a HP, a Apple entrou com uma ação milionária alegando que Microsoft havia copiado o total “*look and feel*” do Macintosh. Este caso foi importante porque mostrou a sensibilidade do tribunal em relação ao desenvolvimento da indústria, pois a facilidade de uso era uma característica importante para um software aumentar sua base de usuários e, assim, permitir ganhos de escala:

Um aspecto importante da decisão foi a sinalização do Tribunal de que a tendência da indústria era pela padronização das interfaces gráficas de usuário, o que decorria de um fator de mercado (‘market factor’) e como forma de assegurar a compatibilidade. Por essa razão, os elementos expressivos da interface tornavam-se padrões comuns, não suscetíveis de proteção. Além disso, a Apple havia dominado o mercado numa época em que os aplicativos utilizavam sua interface padrão. Nesse ínterim, o U.S. Copyright Office expediu

uma decisão administrativa rejeitando o registro de pedidos de 'look and feel' (Santos, 2003, p.275).

Desta forma, tudo indicava que nos EUA a proteção seguiria na linha do copyright. Apesar do estabelecimento da lei de direitos autorais, havia outras formas de proteção disponíveis. Segundo Campbell-Kelly (2003, p.107), a patente foi inicialmente a forma de proteção com maior apelo para software. Mas havia dúvidas se as patentes de software concedidas conseguiriam manter sua validade caso fossem questionadas judicialmente:

A situação de patentes de software era excessivamente complexa, com um backlog de 75 depósitos em 1968 e com cinco recursos em progresso. Além disso, havia assuntos complexos de políticas públicas sobre a validade de patentes de software, já que as patentes eram desenhadas para proteger artefatos tangíveis mais do que 'idéias'. Exatamente que tipo de artefato o software representava era uma questão aberta.

No caso dos dois primeiros produtos de software – Autoflow da ADR (Applied Data Research) e Mark IV da Informatics – tentou-se a proteção via patentes. A ADR conseguiu uma patente de processo. Entretanto, a Informatics não obteve o mesmo feito nos EUA; apenas conseguiu patente na Inglaterra e Canadá. A Informatics precisou então selecionar outra estratégia para proteger seu produto, no qual havia investido meio milhão de dólares³⁶.

A estratégia da Informatics foi utilizar um mix de instrumentos formais e não formais. Como apresentado acima, em 1964 havia uma regulamentação da proteção do software como direito autoral desde que cumpridos os critérios de originalidade e depósito de uma cópia do código fonte do software. Desta forma, uma cópia do código fonte em formato legível para humanos deveria ser depositada no Copyright Office³⁷.

³⁶ Mark IV foi produzido a partir do Mark III adquirido pela Informatics. O investimento pesado foi necessário para transformá-lo em produto, pois grande parte do seu código teve que ser refeito. A aquisição pela Informatics visava mais os 7 anos de aprendizado organizacional e conhecimento do mercado do que propriamente o código. O preço do software (US\$ 30 mil) deixou o mercado assustado, pois nesta época estava acostumado a software gratuito (Campbell-Kelly, 2003).

³⁷ Esta exigência foi eliminada posteriormente, com a revisão da lei Autoral em 1976.

Isto tornou este instrumento inadequado na avaliação da Informatics, pois isto a protegeria de cópias ilegais, mas não protegeria os algoritmos e conhecimento que a expressão literal encapsulava. Abrir o código tornaria relativamente fácil a re-engenharia e a produção de uma cópia perfeitamente legal (Campbell-Kelly, 2003). Neste momento, cópias ilegais não eram o maior problema da Informatics, pois o destino de seu produto era o mercado corporativo, sendo que as empresas tendiam a infringir copyright de forma bem menos freqüente que indivíduos. A Informatics decidiu utilizar então:

1. segredo de negócio, o que requeria assinatura de acordos de confidencialidade (*non disclosure agreements* - *NDA*) junto a clientes e empregados e a entrega apenas do código binário com uma licença perpétua que poderia ser revogada caso o cliente fizesse cópias ilegais;
2. copyright para proteger a documentação;
3. marca para Mark IV e para a expressão visual distintiva do programa;
4. por fim, um investimento pesado em desenvolvimento do produto, vendas, treinamento e gerenciamento dos serviços de infra-estrutura corporativa de forma a proteger sua participação no mercado da imitação de seus produtos (Campbell-Kelly, 2003, p.116-7).

Aqui, vê-se o nascimento de dois instrumentos de apropriação dos resultados da inovação que virão, posteriormente, a ser utilizados pesadamente pelas empresas de software, principalmente de software produto: a entrega apenas do código binário (executável) e o investimento em inovação de produto, marketing, organização etc.

Estes últimos viriam a se tornar uma das principais estratégias das empresas de software na proteção do seu negócio (Roselino, 2006). Os instrumentos formais de proteção, via copyright e registro de marca, eram utilizados pela Informatics de forma auxiliar. Assim, voltando novamente no esquema de Malerba e Orsenigo (1996), em um ambiente de baixa apropriabilidade (facilidade de cópia e legislação “fraca”), como se configurava aquele momento, torna-se mais importante o controle dos ativos complementares do que a proteção formal do ativo tecnológico

principal.

Um dos programas mais relevantes da Apple foi o Visicalc, que praticamente inaugurou o mercado de planilhas eletrônicas para computadores pessoais. Para proteger seu ativo, a Apple tentou conseguir a proteção via patente para o Visicalc ainda em 1979. Entretanto, não conseguiu este tipo de proteção da propriedade intelectual porque a Suprema Corte entendeu que os algoritmos matemáticos dos programas de computador não eram matérias patenteáveis, e nem o software que assimilava o procedimento para solucionar um dado problema matemático (Bessen e Raskind, 1991).

Isso foi alterado com a decisão no caso *Diamond vs. Diehr* em 1981, como apontado no capítulo 1, que deu suporte à idéia de que o software não poderia ser patenteado *per se*, mas quando introduzido num processo poderia requerer patentes se satisfizesse as condições estabelecidas na lei, principalmente o requerimento de utilidade (Coriat e Orsi, 2002).

Desta forma, a jurisprudência abriu espaço para o patenteamento, dado que a cobertura pela Lei de Copyright se mostrou incapaz de proteger todos os tipos de ativos da indústria de software. Segundo Coriat e Orsi (2002), a CAFC em 1982 apontou que a proteção por copyright era muito estreita e restritiva. E ao longo de decisões acabou introduzindo o patenteamento do software por meio da sua jurisprudência, o que resultou em uma mudança progressiva do sistema baseado em copyright para o sistema baseado em patentes.

Desta forma, até os anos 70 havia oposição ao patenteamento nos EUA. Mas a partir de meados dos anos 80, o software se tornou matéria patenteável, suportado tanto pelas vias administrativas quanto judiciais. Isso aconteceu dentro do contexto de maior fortalecimento do direito de patente (Graham e Somaya, 2004).

Assim, a proteção legal para software foi fortalecida nos EUA em meados dos anos 80 com a patente para invenção de software. Um resultado deste fortalecimento foi uma corrida às patentes em quatro áreas: processos controlados por computador, métodos de negócio implementados por meio do uso de computadores, interface homem-computador e algoritmos implementados por computador (Merges, 1996). Desta forma, ficou claro que tinha se tornado possível conseguir

requerer uma patente de um software “puro” ou de partes dele, e não apenas de uma invenção que fosse implementada por software ou que tivesse no software um elemento maior.

Na Europa, a regulação aconteceu mais tardiamente, com a Diretiva 91/250/CEE. Alguns países europeus já tinham promulgado leis nesta direção a partir de 1983, como Hungria, Bulgária, Alemanha, França, Espanha e Inglaterra. Segundo Cerqueira (1999, p.41), a regulamentação na Europa aconteceu sob influência dos EUA, dado que as empresas norte-americanas de software pacote e de serviço dominavam o mercado europeu. A Diretiva de 91 teria sido adotada não só para harmonizar a legislação entre os Estados da Comunidade Européia, mas para colocar a lei européia em conformidade com a norte-americana.

Em 1994, foi implementado o Acordo sobre Aspectos de Propriedade Intelectual Relacionados com o Comércio (TRIPs) de forma compulsória a todos os países integrantes da Organização Mundial de Comércio (OMC), instituição que também foi criada neste ano em substituição ao Acordo Geral de Tarifas e Comércio (GATT), como já apontado. Segundo Carvalho (2003), o TRIPs colocou duas grandes mudanças na proteção da propriedade intelectual. Em primeiro lugar, estabeleceu uma mudança na articulação histórica que havia entre o desenvolvimento tecnológico dos países e a proteção da PI, deslocando o eixo da articulação para o comércio internacional e vinculando o sistema aos padrões adotados em nível internacional sob pena de sanções no comércio internacional.

A Rodada do Uruguai do GATT, que resultou no TRIPs, não começou com o objetivo de discutir PI e comércio internacional, mas isso foi colocado na agenda por pressão dos EUA, como também já visto acima. Os países europeus mantiveram uma posição dúbia quanto a este tema, passando a endossar a posição norte-americana na década de 90, enquanto os países em desenvolvimento achavam que o local para discutir isso era a OMPI, por meio da revisão dos vários tratados em vigência (Convenções de Paris, Berna, Roma e Tratado sobre a Propriedade Intelectual em matéria de Circuitos Integrados)³⁸. A posição norte-americana devia-se às perdas de competitividade de sua economia em função da proteção considerada inadequada para o

³⁸ Para maiores detalhes sobre os tratados, ver Carvalho (2003).

comércio internacional e do aumento da participação dos novos países industrializados no comércio internacional, cujos produtos eram em grande parte fruto de inovação imitativa dos produtos dos norte-americanos:

O Acordo TRIPs representa uma iniciativa dos países desenvolvidos no sentido de ampliar a proteção à propriedade intelectual. Essa iniciativa se deu num contexto de ampliação do comércio internacional e do conteúdo tecnológico dessas exportações, assim como de consolidação de uma nova lógica de produção global, na qual o controle da tecnologia ganha uma dimensão qualitativa diferenciada em relação ao ambiente no qual firmou-se a CUP [Convenção da União de Paris] e as demais revisões. (Carvalho, 2003, p.54)

Em segundo lugar, houve “o estabelecimento de padrões mínimos de proteção, que alteraram o prazo de proteção e a imposição de reconhecimento de proteção para fármacos, alimentos e plantas” (Carvalho, 2003, p.42). O estabelecimento destes padrões mínimos não correspondia à maturidade destas indústrias em grande parte dos países que não reconheciam patentes para estas áreas.

Mas qual o tratamento que o TRIPs oferece ao software? Para início de conversa, o tratamento é controverso³⁹, porque, segundo algumas interpretações, haveria uma contradição entre o texto do artigo 10 parágrafo 1 e o artigo 27 parágrafo 1. Mas há também interpretações que colocam essa pretensa contradição como uma falácia. Vejam-se os textos. O artigo 10 trata da proteção aos Programas de Computador e Compilações de Dados. Em seu parágrafo 1º diz: “Programas de computador, em código fonte ou objeto, serão protegidos como obras literárias pela Convenção de Berna (1971)”. Já o Artigo 27, que trata da matéria patenteável, apresenta:

1 - Sem prejuízo do disposto nos parágrafos 2º e 3º abaixo⁴⁰, qualquer invenção, de produto ou de processo, em todos os setores tecnológicos, será patenteável, desde que seja

³⁹ Um espaço que mostra a controvérsia desta discussão são os verbetes sobre o TRIPs na Wikipedia, como o verbeito denominado “Software patents under TRIPs Agreement” disponível em <http://en.wikipedia.org/wiki/Software_patents_under_TRIPs_Agreement>

⁴⁰ Os artigos 2 e 3 têm o seguinte texto: “2 - Os Membros podem considerar como não patenteáveis invenções cuja exploração em seu território seja necessário evitar para proteger a ordem pública ou a moralidade, inclusive para

nova, envolva um passo inventivo e seja passível de aplicação industrial. (5) Sem prejuízo do disposto no parágrafo 4º do art.65, no parágrafo 8º do art.70 e no parágrafo 3º deste Artigo, as patentes serão disponíveis e os direitos patentários serão usufruíveis sem discriminação quanto ao local de invenção, quanto a seu setor tecnológico e quanto ao fato de os bens serem importados ou produzidos localmente.

(5) Para os fins deste Artigo, os termos “passo inventivo” “passível de aplicação industrial” podem ser considerados por um Membro como sinônimos aos termos “não óbvio” e “utilizável”.

A interpretação – de que haveria uma contradição entre o artigo 10 e o 27 e que seria possível patentear o software – parte do pressuposto de que o software é uma tecnologia, que há passos inventivos na sua criação e que há aplicação industrial. Desta forma, não poderia ser discriminado e haveria uma complementaridade de proteção entre patente e direito de autor. Já os autores contrários a esta interpretação argumentam que o software é uma criação intelectual, que não há passo inventivo e nem aplicação industrial e que, por isso, o programa de computador deve ser protegido apenas por direito de autor.

Da mesma forma que com relação à adoção do direito autoral para proteger software, Santos (2003, p.404) argumenta que os EUA desempenharam um papel importante:

Sob influência da experiência norte-americana a tendência de se estender a proteção patentária para programas de computador cresceu bastante na década de 1990. Deverá isso substituir o Direito Autoral? Segundo a Comissão Européia, em estudo de fevereiro de 1999, o que se contempla é a “aplicação paralela do direito de autor e do direito

proteger a vida ou a saúde humana, animal ou vegetal ou para evitar sérios prejuízos ao meio ambiente, desde que esta determinação não seja feita apenas por que a exploração é proibida por sua legislação.

3 - Os Membros também podem considerar como não patenteáveis:

- a) métodos diagnósticos, terapêuticos e cirúrgicos para o tratamento de seres humanos ou de animais;
- b) plantas e animais, exceto microorganismos e processos essencialmente biológicos para a produção de plantas ou animais, excetuando-se os processos não biológicos e microbiológicos. Não obstante, os Membros concederão proteção a variedades vegetais, seja por meio de patentes, seja por meio de um sistema "sui generis" eficaz, seja por uma combinação de ambos. O disposto neste subparágrafo será revisto quatro anos após a entrada em vigor do Acordo Constitutivo da OMC.”

patentário”, ou seja, a coexistência dos dois sistemas. Essa orientação é coerente com o Acordo TRIPs, segundo o qual o regime autoral não exclui o patentário.

Acredita-se que os EUA tentaram forçar uma proteção mais ampla para as invenções implementadas por software por meio de patentes, mas encontraram muita resistência. Entretanto, conseguiram que não fosse incluída a proibição da patente para o software *per se* como aparece na Convenção Européia de Patentes. Segundo o grupo Eupat da *Foundation for a Free Information Infrastructure* (FFII), o Escritório Europeu de Patentes (EPO) começou a aceitar patentes de software em 1998, com a criação da figura das invenções implementadas por software. Esta figura daria o caráter técnico ao software, o que seria um critério necessário para o patenteamento de qualquer invenção, segundo o TRIPs⁴¹.

2.3.2 Apropriação por modelo de negócio

Neste item, são apresentados os resultados de duas pesquisas que analisaram o uso dos instrumentos de propriedade intelectual nos diferentes modelos de negócio. A primeira pesquisa (Merges, 1996) comparou os efeitos dos regimes de propriedade intelectual nas indústrias de software norte-americana, japonesa e européia no início da década de 90. A segunda pesquisa (Torrise, 1998) comparou a utilização dos instrumentos legais entre empresas de hardware e de software. Além das duas pesquisas, para aprofundar o entendimento das formas de apropriação dos resultados econômicos da inovação, apresentam-se alguns depoimentos de empresas dos vários modelos de negócio na indústria de software, por ocasião de um evento organizado pelo USPTO em 1994, sobre a adequação dos instrumentos legais e de suas práticas no processo de análise das patentes de software.

Merges (1996) considerou a relação do regime de PI com três características da indústria: tamanho, tamanho relativo do segmento de software customizado/pacote e estrutura da indústria (especialmente o tamanho das empresas). A indústria norte-americana era muito mais expressiva que a japonesa e européia. Além disso, a participação do segmento de software produto era muito maior na indústria norte-americana que nas demais. Estimava-se que cerca de 70% do

faturamento da indústria norte-americana vinha do segmento de software pacote, enquanto que no Japão o segmento de software customizado contribuía com cerca de 90% do faturamento da indústria e na Europa 46%. Parte considerável dos 54% de faturamento com software pacote na Europa vinha de empresas norte-americana.

No Japão, o regime de PI era relativamente fraco e a indústria de software era relativamente pequena se comparada à presença japonesa no mercado de bens complementares da indústria de computadores, como semicondutores. A presença do software customizado era muito maior que a do software pacote. A estrutura da indústria era formada por empresas grandes que eram parte de grandes grupos (*keiretsu*), muitas das quais funcionavam como fábricas de software. Havia muito desenvolvimento de software *in-house* e, em geral, as empresas de software atendiam apenas um cliente (*patron*). Segundo Merges (1996), dada a estreita ligação das empresas de software customizado com um cliente, elas não dependiam muito do regime legal para proteger o software, pois a proteção da PI era realizada através de contratos. Além disso, o regime fraco de PI ainda propiciou que as empresas japonesas se apropriassem de idéias das empresas estrangeiras e lucrassem com elas.

Mas este quadro estava sendo alterado com o crescimento do segmento de software pacote. Segundo Merges (1996, p. 282), a proteção forte da PI era importante para diminuir custos de transação neste segmento:

A indústria japonesa de software ensina algumas lições valiosas sobre o papel dos direitos de propriedade na superação dos custos de transação. Sem a segurança de um direito de propriedade garantido pelo governo, os fornecedores de software no Japão seriam relutantes a deixar a esfera contratual de proteção que eles compartilhavam com seus clientes/patrões usuais. Mas com tais direitos, impostos além do contexto dos contratos individuais (ou seja, um direito que é um 'bem contra o mundo'), estas firmas são livres para vender para outros consumidores. Assim, o estabelecimento do direito de propriedade

⁴¹ <http://eupat.ffii.org/analysis/trips/swpattrips.en.pdf>

reduz os custos de transação para lidar com um grupo de compradores amplamente disperso, tornando tais transações possíveis.

Aqui é interessante notar que hoje a situação parece ter se invertido, pois o fortalecimento dos DPIs na indústria de software tem aumentado os custos de transação, como apontado no capítulo 1 e como será discutido nos próximos capítulos.

A Europa também contava com um regime de PI fraco, com a adoção de copyright num grau modesto. A indústria era relativamente pequena com um mercado de software customizado maior e mais aberto se comparado ao japonês. Para Merges, a explicação do insucesso do segmento de software pacote tinha que ser buscada fora do âmbito da PI. A explicação estava na falta de sucesso da indústria de hardware, que não conseguiu alcançar uma posição importante no cenário internacional em parte porque havia alguns “campeões nacionais” que eram protegidos da competição por seus governos nacionais, o que evitou a difusão do hardware mais avançado norte-americano. Isto impediu a constituição de uma configuração padrão de hardware, como ocorreu nos EUA, o que no caso norte-americano possibilitou a criação de uma grande base para que o software pacote pudesse atingir ganhos de escala e também externalidades positivas de rede. Desta forma, as medidas de proteção de mercado utilizadas pelos governos europeus tiveram um efeito danoso na inovação na indústria de hardware e, conseqüentemente, na de software.

Nos EUA, ao contrário do Japão e Europa, a PI foi um importante instrumento, manipulado pelo governo para o fortalecimento da indústria de software, particularmente do segmento de software pacote. Os DPIs tiveram um importante papel na determinação da estrutura da indústria. Ao contrário do Japão, o sistema norte-americano de apropriabilidade era muito forte, de tal forma que ameaçava condições saudáveis e competitivas de entrada no mercado. Um exemplo citado por Merges foi a intervenção governamental na aquisição da Ashton-Tate, produtora de uma das primeiras bases de dados (dBase) pela Borland. Como condição para a aquisição, o Departamento de Justiça impôs que as características da interface e da estrutura de comandos não poderiam ser fonte de processos de copyright. Esta condição colocou no domínio público estes elementos, o que reduziu o poder de mercado da empresa conjunta.

Para Merges (1996), PI era uma variável chave na indústria de software, assim como eram as capacidades produtivas nas indústrias tradicionais. Acreditava que níveis mais baixos de PI poderiam resultar em mais inovação, uma espécie de pirataria cooperativa. Além disso, a inovação nesta indústria vinha em maior grau das novas empresas entrantes do que do aperfeiçoamento feito pelas incumbentes em seus produtos. Segundo este autor, os padrões de software tinham sido criados quando DPIs eram mais fracos. Como entendia que o software pacote era chave para o crescimento da indústria, previa que a política de PI teria impacto real e que deveria se pensar em moderá-la. Acreditava que ocorreria uma redução geral da PI para criadores de software nos EUA de forma que ficariam mais alinhados com Japão e Europa. Entretanto, passados mais de 10 anos do artigo de Merges, o que se vê foi que ocorreu exatamente o contrário, pois os EUA fortaleceram ainda mais seu regime de PI e os outros países têm seguido nessa direção em maior ou menor grau.

Os resultados do survey realizado por Torrisi (1998), em 1990, com uma amostra de 51 empresas de hardware e de software da Europa⁴², mostrou resultados semelhantes. Os instrumentos legais eram considerados instrumentos fracos de apropriabilidade pelas empresas pesquisadas. Havia diferenças dos mecanismos utilizados entre empresas de hardware e de software:

- as empresas de hardware, que também produziam software, faziam uso de um conjunto mais amplo de instrumentos, a começar pelos instrumentos legais, passando por *lead time*, inovação contínua e alianças com empresas dotadas de recursos complementares;
- já as empresas de software utilizavam quase exclusivamente a apropriabilidade dinâmica: *lead time*, inovação contínua, pessoal qualificado – mostrando fraca habilidade para a utilização da proteção legal, o que podia ser explicado pela recente, na época, extensão da proteção por copyright e patentes para software, mas também pela dificuldade de separar o conhecimento codificável e protegível do conhecimento tácito.

Já as empresas norte-americanas utilizavam com mais frequência os instrumentos legais.

⁴² A amostra de empresas, localizadas na Itália, Reino Unido, França e Alemanha, incluía: as maiores empresas de software européias, como Cap Gemini, Finsiel, Sema Group, SD-Scicon, Logica, Concept e Software AG; e produtores de hardware.

Segundo um estudo realizado pelo MIT em 1989 com empresas norte-americanas de software, 75% das empresas entrevistadas utilizavam segredo de negócio, 25% copyright e 8% patentes (Torrise, 1998).

Para encerrar o capítulo, procura-se abaixo “dar voz” às empresas, a partir de seus depoimentos nas conferências denominadas *Public Hearing on Use of the Patent System to Protect Software-Related Inventions*, organizada pelo USPTO em 1994 com o objetivo de discutir o patenteamento de invenções relacionadas com software (USPTO, 1994).

A inovação constante era o motor principal do sucesso dessa indústria e o USPTO estava preocupado com a disponibilidade de mecanismos adequados para proteção das inovações do software. O USPTO queria com esta discussão aperfeiçoar seus mecanismos de exame das patentes e garantir que as informações relatadas nas solicitações realmente atendessem ao critério de divulgação (*disclosure*), ou seja, que por meio de sua leitura alguém com habilidades na área fosse capaz de recriar a invenção descrita. A seguir, são discutidos os posicionamentos apresentados na conferência de San Jose⁴³ agrupados pelos modelos de negócio, acrescentando algumas empresas que produzem software, mas não são classificadas como empresas de software.

Os representantes de empresas da **indústria de hardware** na época⁴⁴ posicionaram-se favoravelmente ao patenteamento das invenções relacionadas com software. Estas empresas investiam parte substancial de seu faturamento em P&D, sendo que grande parte disso se dirigia às atividades de desenvolvimento de software. Argumentavam que as patentes de software deveriam ser tratadas como as demais patentes, já que a distinção técnica entre hardware e software estava desaparecendo. Segundo eles, discriminar este tipo de patente seria um erro, dado que se tratava de uma tecnologia importante para a prosperidade da economia. A indústria de hardware investia muito em software como forma de diferenciar seus produtos. Assim, em termos práticos, para estas empresas o software não era diferente de hardware; entretanto o USPTO

⁴³ Houve outra conferência em Arlington que tratou dos aspectos de interface.

⁴⁴ Entre eles Intel, StorageTek, *Computer and Business Equipment Manufacturing Association* (CBMA), Sun, Silicon Graphics e IBM.

estava mais bem preparado para lidar com invenções de hardware.

Ademais, as patentes de software estavam integradas nos portfólios destas empresas, que tinham a opinião de que a patente provia o instrumento mais amplo para atrair capital de risco para novos empreendimentos. Copyright protegia apenas a expressão e segredo de negócio provia a proteção necessária em relação à divulgação (*disclosure*) por parte de empregados ou parceiros em situações de *joint venture*. Desta forma, cada instrumento protegeria um aspecto diferente e não seria o caso de optar entre uma coisa ou outra.

Segundo a Intel, o sistema de patentes tinha consistentemente servido como incentivo para os esforços técnicos e financeiros para desenvolver novas tecnologias e isso não era diferente para a área de software. Complementando, a Sun apontou que o sistema tinha problemas que precisavam ser resolvidos, pois a operação do sistema em 1994 trazia incerteza para executivos que, em geral, não gostam de arriscar com base nos resultados do sistema legal.

Em seu depoimento, a IBM afirmou que protegia a expressão de todos os seus produtos de software por meio de copyright. Além disso, aproximadamente de 3% a 5% destes programas possuíam funções novas e não óbvias que eram, então, protegidas por patente. Na visão desta empresa, a atividade de P&D era executada para ganhar competitividade sobre os adversários. Desta forma, se seus frutos não pudessem ser protegidos da cópia por parte dos concorrentes, a luta teria sido perdida. A IBM mostra aqui uma estratégia de proteção diversificada e voltada para os concorrentes e não para seus usuários, como será comum no segmento de software pacote.

Outro grupo é o de empresas de outras áreas econômicas que **não software**, mas cujas atividades exigem **altos níveis de investimento em P&D em software**, como AT&T e Exxon Production Research Company (EPR). Na opinião destas empresas, a proteção via copyright era adotada largamente para proteger software pacote de massa. Entretanto, este instrumento era insuficiente para proteger o investimento em software do tipo que realizavam, porque copyright só protegia a expressão, não protegendo os sistemas, métodos e outras funcionalidades do software. Além disso, as patentes representavam uma garantia para o investidor, facilitavam alianças e

promoviam a divulgação da informação (função de divulgação e prospecção da patente, como visto no capítulo 1).

A EPR era constantemente desafiada nas áreas relacionadas com computação, seja hardware ou software para processar os dados sísmicos ou dos reservatórios e convertê-los em formatos rapidamente analisáveis. Para isso, precisava desenvolver algoritmos matemáticos sofisticados, o que envolvia pesados investimentos em termos de tempo e dinheiro para criar os algoritmos e integrá-los ao software e hardware. No caso desta empresa, a combinação de hardware aliada ao software e algoritmos sofisticados permitiu a ela um salto tecnológico, o que gostaria de poder patentear.

As empresas de **software produto se posicionaram tanto favoravelmente quanto contrariamente ao patenteamento de software**. Para o Multimedia Development Group, num **posicionamento mais abertamente favorável**⁴⁵, a patente era uma ferramenta que encorajava a inovação e ajudava os inventores a se beneficiarem das contribuições dos outros, sendo que isso valia tanto para software como para as invenções relacionadas com tecnologias tangíveis, sejam mecânicas, químicas ou elétricas.

Uma argumentação mais conservadora deste grupo de empresas era, já que existiam patentes para invenções relacionadas com software, que estas fossem aperfeiçoadas. Ou seja, já que estas invenções podiam ser patenteadas, gostariam que as patentes fossem de boa qualidade e claras, especialmente quando tratavam de elementos básicos da indústria de software. A Microsoft pode ser situada neste grupo conservador em 1994, pois acreditava que a proteção via copyright tinha servido bem à indústria, mas que havia um papel crescente para patentes de invenção. O sistema precisava ser aperfeiçoado e inclusive apoiavam a iniciativa do Software Patent Institute (SPI)⁴⁶

⁴⁵ As empresas que se posicionaram favoravelmente, com maior ou menor entusiasmo, foram: Software Entrepreneurs Forum; Ikonix; Taligent Inc.; Microsoft; Realsoftware e Multimedia Development Group.

⁴⁶ SPI é uma corporação sem fins lucrativos criada para fornecer cursos e base de dados para pesquisa de anterioridade (*prior art*) sobre tecnologia de software para aperfeiçoar o processo de patenteamento. Os financiadores do SPI são empresas de grande porte como: Adobe, Apple, IBM, Microsoft, Oracle e Symantec, além de associações de empresas de software e escritórios de advocacia. SPI encontra-se hospedado no Sagamore Institute for Policy Research, um *think tank* sem fins lucrativos localizado em Indianapolis. Mais detalhes em <http://www.spi.org/> e <http://www.sipr.org/>

de criar um banco de dados com informação técnica para facilitar a busca do estado da arte.

Sobre o impacto das patentes de software na indústria, a Microsoft afirmou que não tinha processado os responsáveis por possíveis violações de suas patentes, mas que já tinha tido que se defender em vários casos, que consumiram recursos consideráveis, os quais gostaria de tê-los direcionado para seus esforços de P&D. Apesar disso, estava comprometida com o sistema de patentes como um veículo razoável e responsável para proteger a inovação no software. Segundo esta empresa, a estrutura legal era apropriada (a combinação do uso de copyright, patente e segredo de negócio), tanto que o amadurecimento da indústria comprovava isso.

Durante seu depoimento, o representante da Microsoft foi questionado pelo Comissário Bruce Lehman⁴⁷, que perguntou o que havia de diferente naquele momento, já que a Microsoft tinha crescido e se estabelecido praticamente sem fazer uso da proteção via patentes. O representante respondeu que a Microsoft tinha se tornado consciente do valor de patentear invenções relacionadas com software, pois até então a indústria de software era muito jovem e utilizava fortemente segredo de negócio e copyright. Entretanto, empresas como IBM e AT&T, que atuavam principalmente com hardware, já tinham a cultura de patente implantada na empresa e por isso empregavam mais este instrumento. Já as empresas de software não tinham esta cultura⁴⁸.

Era praticamente um consenso, mesmo entre as empresas abertamente favoráveis, que o sistema de patenteamento favorecia grandes empresas mais do que pequenas empresas. Entendiam que patentes de escopo incerto, principalmente as com escopo muito amplo, dificultavam a tomada de decisão, a atração de capital de risco e desenvolvimento de planos de negócios. E estas patentes eram mais danosas para pequenas empresas do que para grandes, pois as pequenas em geral não conseguiam pagar por uma boa análise legal sobre o escopo e a validade de uma determinada

⁴⁷ Bruce A. Lehman foi Secretário Assistente do Comércio e Comissário de Patentes e Marcas entre 1993 e 1998 e foi responsável por significantes mudanças na lei de patentes dos EUA. Quando saiu do USPTO fundou o Intellectual Property Institute. Lehman foi o responsável pela organização das duas audiências sobre patente de software em 1994. Por volta de março deste mesmo ano, Lehman solicitou o re-exame da patente da Compton, uma das patentes mais polêmicas, que foi então rejeitada. Fontes: Wikipedia, <http://www.interesting-people.org/archives/interesting-people/199403/msg00104.html>

⁴⁸ Por exemplo, em 1994 a Microsoft tinha 6 advogados trabalhando com patentes e dois anos e meio antes não tinha nenhum. A Microsoft possuía apenas cerca de 340 patentes aprovadas em 1994, número que crescerá bastante a partir desta data.

patente caso fossem acusados de violá-la patente em caso de litígio. A impressão destas empresas era que o sistema de patentes à época parecia estimular litígio.

As **empresas de software produto corporativo e software customizável** posicionaram-se contrariamente ao patenteamento das invenções relacionadas com software⁴⁹. Em geral, estas empresas acreditavam que a patente deveria ser excluída como forma de proteção do software, visto que a proteção provida por copyright e segredo de negócio era suficiente.

Quando a Adobe iniciou suas atividades, seu objetivo principal era criar um software para revolucionar o mundo da impressão. Nesta época, acreditavam que não era possível patentear este trabalho. Entretanto, isto não os impediu de criar o software nem de conseguir capital de risco. Além do desperdício de tempo e dinheiro para depositar patente, o argumento da Adobe era que conceder posições de monopólio numa indústria que já era a mais inovativa de todas iria ter um efeito contrário: promover a estagnação ao invés de aumentar a taxa de inovação. Copyright era considerado o veículo adequado para proteger os direitos dos criadores nesta indústria, dado que o software era visto basicamente uma idéia, cujo processo de criação não era caro nem consumia tempo. Além disso, todo programa expressa algoritmos matemáticos, de forma que seria inconsistente, por um lado, colocar o algoritmo como não patenteável e, por outro, possibilitar a patenteabilidade do software.

A argumentação da Oracle foi na mesma linha: o monopólio que era dado por meio da patente em outras áreas (e que incentivava o dispêndio em inovação, design, produção, marketing) não servia para a área de software. Isto porque a inovação e o desenvolvimento dos produtos de software eram diferentes das indústrias de manufatura tanto porque o desenvolvimento e aperfeiçoamento de produtos era mais rápido, quanto porque não havia necessidade de instrumentação e fabricação. Desta forma, o processo de patenteamento era demorado em relação ao ciclo de vida dos produtos, o que não incentivaria as empresas a investir em desenvolvimento, produção e distribuição dos seus produtos.

A Oracle ainda não tinha sido acusada de violar patentes, mas acreditava que seria virtualmente

⁴⁹ As empresas foram Adobe, Oracle, Borland, Autodesk, Exemplar logic e Reasoning Systems.

impossível desenvolver um produto complicado de software sem infringir alguma patente, especialmente as de amplo escopo que foram concedidas inicialmente sem um exame criterioso. Como a validade de várias patentes podia ser, e estava sendo de fato, questionada e como esta empresa possuía os recursos necessários para se defender em caso de litígio, seu representante acreditava que os detentores de patentes seriam reticentes em acusá-la. Além disso, a Oracle adotou uma estratégia defensiva: depositar patentes de invenções selecionadas que representavam as melhores oportunidades para licenciamento cruzado com outras empresas.

Seguindo uma linha de raciocínio muito próxima do grupo de empresas de software produto e software customizável, a empresa de **software embarcado** Wind River Systems⁵⁰ colocou que a patente não permitiria a invenção ao redor, mas conferiria a seu proprietário o monopólio total daquele software. Como exemplo, citou que se o protocolo TCP-IP⁵¹, uma das tecnologias fundamentais da Internet, tivesse sido patenteado, provavelmente a rede mundial de computadores não existiria do jeito como a conhecemos.

De forma radicalmente contrária ao patenteamento de software também se colocou a **League for Programming Freedom - LPF**⁵², tendo como porta-voz Gordon Irlam, representando os **programadores individuais** que, como visto acima, sempre tiveram um papel relevante na indústria de software. Para eles, a patente dificultava a atividade de programação, pois a cada nova idéia o programador deveria chamar o departamento jurídico para ver se esta já tinha sido

⁵⁰ Wind River Systems, Inc. é uma empresa listada em bolsa, com faturamento de US\$ 30 milhões em 1994 que produz software embarcado (para camada middleware) para carro, fax, telefones, robôs, fábricas e eletrônicos, e que exportava cerca de 40% de sua produção na época. Ela é proprietária da versão comercial do sistema operacional BSD e em 2004 estabeleceu uma parceria com Red Hat, que produz uma das mais importantes distribuições do Linux, para produzir uma versão deste sistema operacional para ser embarcada em equipamentos, que foi lançada em 2005. Seus competidores atuais incluem a Microsoft que possui uma linha do Windows para embarcados (Windows CE e Windows NT embedded). Fonte: http://en.wikipedia.org/wiki/Wind_River_Systems

⁵¹ TCP/IP é um protocolo de controle de transmissão da Internet (Transmission Control Protocol / Internet Protocol), criado nos anos 70 por um comitê formado por várias organizações, dentre elas a Internet Engineering Task Force – IETF (<http://www.ietf.org>). O sucesso deste protocolo foi sua inclusão na versão do sistema operacional UNIX da UCLA de Berkeley, que era disponibilizado com o código aberto (Alves Jr., 2001).

⁵² A LPF é uma organização que reúne mais de 600 desenvolvedores de software, empresários, professores e estudantes e que se opõe às patentes de software e à proteção via copyright das interfaces de usuário. Dedicar-se a restaurar a liberdade que existia no desenvolvimento de software antes das patentes de software. Em 1991 produziram um manifesto contra patentes de software, disponível em <http://lpf.ai.mit.edu/Patents/against-software-patents.html>

patenteada ou estava coberta por alguma patente. Segundo Irlam, as patentes tinham efeitos diferentes dependendo da indústria. Na indústria de software tinham um efeito danoso, visto que era uma indústria altamente competitiva e as patentes sufocavam a competição. Na produção do software o mais importante não estava nas novas idéias, mas na resolução dos problemas dos clientes por meio da implementação das idéias e da construção de produtos.

Outro porta-voz dos programadores foi Richard **Stallman**. Falando pela **Free Software Foundation**⁵³, Stallman apontou que, em termos de patenteamento, o software seria o extremo oposto dos fármacos, que exigem uma patente para cada produto. No caso do software, uma patente envolvia muitos programas similares e um mesmo software podia infringir várias patentes dado que um software combina um grande número de técnicas e implementa muitas funções.

No desenvolvimento do software, a dificuldade não residia em criar inovações, mas em incorporá-las em sistemas complexos que fizessem o que era pedido. E não tinha conhecimento da forma de patentear isso. Segundo Stallman, o mercado de software tinha crescido assustadoramente nas últimas duas décadas e não acreditava que isso tivesse ocorrido devido a patentes de software.

Para concluir o capítulo, o quadro dos modelos de negócios é apresentado novamente (Quadro 2.2), agora complementado com as formas de apropriação utilizadas pelas empresas, como apresentado nos estudos e depoimentos. O que se vê é a utilização de um conjunto de instrumentos para proteger os aspectos de cada modelo de negócio e não uma solução única. Na maioria dos casos, a utilização dos instrumentos jurídicos é auxiliar a outras formas de apropriabilidade dinâmica.

⁵³ A FSF será apresentada no capítulo 3 quando for apresentado o software livre.

Quadro 2.2 - Principais formas de apropriação dos resultados da atividade econômica dos principais modelos de negócio da indústria de software

Características	Principais instrumentos de apropriação (ativos principais e complementares)
Serviço de baixo Valor	<ul style="list-style-type: none"> • Relação contratual entre fornecedor e cliente com acordo de nível de serviço (SLA da sigla em inglês) • Preço • Uso insignificante de DPIs
Serviço de alto valor	<ul style="list-style-type: none"> • Relação contratual entre fornecedor e cliente • Imagem da empresa dada por carteira de clientes e/ou certificação • No caso de uso de componentes, copyright e/ou patente destes • Suporte pré e pós vendas • Pequeno uso de DPIs (marca) mas crescente depósito de patentes
Produto customizável	<ul style="list-style-type: none"> • Licença • Disponibilização apenas do programa executável (código binário sem código fonte) • Suporte pré e pós vendas • Uso marginal de DPIs (segredo de negócio e copyright)
Software embarcado	<ul style="list-style-type: none"> • patente de invenção do equipamento e da funcionalidade do software • disponibilização do código binário e, às vezes, do código fonte • especificidade do software limita risco de cópia por concorrentes
Produto pacote	<ul style="list-style-type: none"> • grande base de usuários • Disponibilização apenas do programa executável • Licença • Inovação rápida • Marketing • Rede de distribuição, treinamento e suporte • Uso mais acentuado de DPIs: Copyright, segredo de negócio e patente • Ações de reforço do copyright (luta anti-pirataria)

Fontes: Elaboração própria a partir de Roselino (1998; 2006), Allison *et al* (2006), SOFTEX (2003; 2005), Torrasi (1998), Merges (1996).

Nota: O modelo de negócio *hardware que produz software* não foi considerado aqui.

No caso dos serviços de baixo valor, a apropriação se dá por meio da relação contratual entre cliente e fornecedor, de forma que os instrumentos formais de proteção da PI são pouco utilizados, com exceção talvez da marca. O mesmo ocorre no caso dos serviços de alto valor, mas aqui a imagem da empresa é mais relevante, o que implica na valorização dos ativos complementares de relacionamento com o cliente, certificação em qualidade e suporte pré e pós-vendas.

Nos modelos de software produto, os instrumentos formais de PI possuem uma importância maior. No caso do software customizável, o uso dos DPIs é marginal. Os ativos tecnológicos são

protegidos tecnicamente via entrega da versão executável do software, comercializado via licença. Os ativos complementares devem ser fortalecidos via rede de distribuição e suporte (integrada ou via parceiros). No caso do software pacote, inicialmente quanto mais usuários, melhor, o que pode induzir práticas de “pirataria consentida”, como no caso da Microsoft discutido anteriormente. Desta forma apesar de ser o modelo de negócio cujas empresas mais utilizavam copyright, no caso desta prática, era mais importante ampliar a base de usuários do que o pagamento de royalties perdido devido às cópias piratas. A proteção via copyright pode ter um papel menor.

Já no software embarcado, como trata-se de um software customizado para um determinado equipamento, esta especificidade de certa forma já impede a imitação e é possível inclusive a disponibilização do código binário. Mas é o modelo que pode utilizar a forma mais forte da patente de invenção para o conjunto formado por equipamento e software.

Em resumo, o capítulo 2 procurou traçar a co-evolução da indústria de software, dos seus primórdios até o início comercial da Internet, e da regulação da propriedade intelectual. Um dos pressupostos do capítulo foi que a apropriação dos resultados da inovação varia de acordo com o país e também de acordo com o modelo de negócio da empresa. Por isso, foi realizada uma tentativa de sistematizar as principais características de um conjunto de modelos de negócio existentes na indústria neste período, bem como sistematizar os principais instrumentos de apropriação utilizados em cada modelo de negócios.

O próximo capítulo continuará o esforço narrativo da história da indústria, apresentando alterações nestes modelos de negócio, novos modelos e formas de apropriação desenvolvidas.

3. A INDÚSTRIA DE SOFTWARE PÓS INTERNET

O objetivo deste capítulo é, por meio da continuação da história analisada no capítulo anterior, mostrar os efeitos da co-evolução da indústria de software e dos mecanismos de proteção da propriedade intelectual com os temperos do lado da indústria (advento da Internet e fortalecimento do software de código aberto) e do lado da regulação da propriedade intelectual (corrida patentária).

Assim, algumas variáveis adicionaram complexidade neste caldeirão, como a corrida por patentes de software. Independente da opinião expressa sobre esses aspectos, várias das empresas consideradas no capítulo anterior correram ao patenteamento de suas invenções relacionadas com software, no mínimo, como medida defensiva. A corrida aconteceu com intensidade diferente dependendo do ativo principal de cada empresa.

O software livre e de código aberto (SL/CA) introduziu uma segunda nova variável neste período ao apontar modelos alternativos de desenvolvimento de software e de disponibilização e proteção dos conhecimentos gerados. Isso ocorreu principalmente com o estabelecimento do desenvolvimento compartilhado de software, uma decorrência da junção da difusão do acesso à Internet com o processo de capacitação que ocorreu com a proliferação do computador, criando uma massa de desenvolvedores com um padrão diferente de apropriação, por meio principalmente de serviços, que não privilegiam a proteção do software. Como já apontado, o SL/CA mostra o papel importante dos usuários na inovação desta indústria.

Outra variável importante introduzida pelo software livre foi o desenvolvimento de programas de infra-estrutura, como sistema operacional Linux, o servidor web Apache entre outros. À medida que o software livre foi se tornando código aberto (*Open Source*)⁵⁴ foi também influenciando várias tendências da indústria de software. Assim, houve mudanças no software proprietário, mas também aconteceram várias mudanças no software de código aberto. Num primeiro momento, o software livre procurou ocupar espaços no segmento de infra-estrutura via imitação de software pacote considerado *commodity*, como sistema operacional, pacotes de automação de escritório, entre outros. Mais recentemente, verificam-se tendências de produzir programas mais sofisticados. Mas não se

pode deixar de notar que o SL/CA alterou as posições de todos os atores de alguma forma.

Nos anos 2000, surgiram iniciativas de “*flair play*” frente ao ritmo que a corrida tomou e os possíveis riscos para a indústria em geral e também devido à estratégia de algumas empresas de apostar em padrões de interoperabilidade⁵⁵. Ao mesmo tempo, a discussão sobre qual a melhor forma de proteger as invenções de software se acirrou, a despeito de sempre ter sido “quente” e de ter se tornado prática corrente o patenteamento de software.

Diante do exposto, o presente capítulo se propõe a dar continuidade à análise proposta na tese dando destaque a dois aspectos: às mudanças em curso do regime tecnológico, denominado de serviços na web, e às mudanças nos regimes de apropriabilidade, como o aumento da intensidade do patenteamento e o surgimento de outros formatos para proteção da PI.

Para isso, este capítulo é composto por três partes. A primeira parte continua analisando a história da indústria de software no período de 1994 aos dias atuais. Esta periodização é justificada pelo fato deste período ter sido marcado pela expansão e consolidação da indústria de software no mundo, pelo advento do software livre e de código aberto e a migração mais acentuada dos negócios da indústria de software para o oferecimento de serviços e pela disseminação da Internet. Além disso, a Internet influenciou também a expansão territorial da indústria de software e possibilitou o compartilhamento de trabalho no SL/CA.

A segunda parte trata da corrida patentária endógena e exógena à indústria de software, dos custos de transação relacionados e das estratégias encontradas pelas empresas para lidar com essa situação. A terceira parte sistematiza as mudanças nos cinco modelos de negócios, apresentados no capítulo 2, e apresenta os dois novos modelos de negócio, aqui denominados Serviços Gratuitos na Web (tipo Google) e Serviços Pagos na Web (tipo ASP - *Application Service Provider*).

3.1 Software torna-se commodity e suporte para serviços

O mercado de software pacote expandiu-se com a produção em massa de computadores

⁵⁴ Esta distinção será apresentada com detalhe mais à frente.

⁵⁵ Habilidade de sistemas distribuídos trocarem dados entre si.

padronizados, começando pela camada mais baixa da hierarquia do software⁵⁶, ou seja, os sistemas operacionais e linguagens computacionais, indo para a camada dos aplicativos, cujo exemplo mais conhecido é a suíte de escritório, composta por processador de texto, planilha eletrônica, banco de dados e software de apresentação. Com a difusão da informática por vários setores econômicos e também nos domicílios, surgiram oportunidades de fornecimento de soluções padronizadas em larga escala, o que resultou no fenômeno da *commoditização* (também chamada *commoditificação*)⁵⁷ deste tipo de software.

Além do ganho de escala para o software com a padronização do hardware, a emergência e disseminação do SL/CA também favorecem a *commoditização* do software por ampliarem o efeito das outras características da *commodity*, quais sejam, a existência de múltiplos produtores que oferecem produtos de qualidade homogênea. Isto possibilita a substituição de uma aplicação de software por outra sem maiores problemas, pois possuem funcionalidades e “*look and feel*” semelhantes. A *commoditização*, junto com a difusão do reuso de software⁵⁸, indicariam um sinal de amadurecimento da indústria de software, segundo Roselino (1998). Entretanto, há controvérsias quanto a esta opinião, pois ao mesmo tempo em que estes processos avançam (em ritmo mais ou menos lento, como será discutido no próximo capítulo), há outros sinais de juventude da indústria de software.

Desta forma, o período de 1994 ao presente é marcado por várias transformações na indústria de software. Em termos de modelos de negócio, há um crescimento do software produto, com a expansão da informatização em diversos setores econômicos e nos domicílios. Entretanto, acontece, quase ao mesmo tempo, uma acentuada migração para os modelos baseados em serviços. De forma associada, acontece a consolidação do software livre e de código aberto. Neste período também ocorre o fenômeno que ficou conhecido como a bolha da Internet, que cresce e estoura em março de 2000. E também se pode dizer que a indústria se torna realmente internacional, tanto em termos de comércio (pois praticamente já nasceu internacionalizada) e quanto em termos de produção.

⁵⁶ O software pode ser diferenciado a partir da posição ou camada que ocupa na hierarquia dos sistemas de software, da camada mais baixa (sistema operacional e ferramentas de desenvolvimento), passando por software para a camada *middleware* (gerenciador de rede, gerenciador de sistemas, etc.) e por fim os aplicativos (Roselino, 2006).

⁵⁷ Isso ocorre também em menor grau no software para mercados verticais como automação comercial, contabilidade, finanças, entre outros (Roselino, 1998).

3.1.1 A indústria de software se consolida internacionalmente

Segundo Roselino (2006, p.47), a indústria de software pode ser vista como um mosaico, dotada de um caráter heterogêneo que “resulta num quadro internacional marcado pela convivência de segmentos mais regionalizados com outros nos quais o elevado grau de internacionalização é o traço marcante desde sua constituição original”. Há níveis distintos de internacionalização nos vários segmentos, mas há também uma tendência geral em direção à intensificação do caráter global da atividade de software, em harmonia com o movimento geral da globalização da economia – pelo aumento da comercialização, pela presença de empresas transnacionais nas diversas configurações nacionais, bem como nos novos modelos internacionalizados de desenvolvimento de software. Ainda segundo o mesmo autor, há dois motores da internacionalização desta indústria:

i. A intensificação do processo geral de globalização produtiva, com a descentralização geográfica das funções corporativas nos mais diversos setores produtivos. Nesse novo modelo produtivo as atividades intensivas em TI são freqüentemente terceirizadas para firmas especializadas, amiúde localizadas em outros países;

ii. A reprodução, no âmbito da própria indústria de software, desse mesmo movimento de globalização produtiva ocorrido em diversas atividades manufatureiras, com a decorrente tendência de externalização de etapas do desenvolvimento de soluções em software (serviços e produtos) por parte das grandes empresas do setor (Roselino, 2006, p.48)

O primeiro motor é a internacionalização passiva, dada pela terceirização das atividades de informática das empresas usuárias, cujo maior exemplo é a Índia. O segundo é a internacionalização ativa, dada pela descentralização do desenvolvimento do software, mas mantendo o comando das atividades pela empresa líder, normalmente norte-americana, que controla as funções superiores. Essa externalização do desenvolvimento de software aloca em países não-centrais funções nas quais as vantagens comparativas de custo têm papel fundamental. O exemplo mais conhecido da internacionalização ativa é a Irlanda, que sedia a unidade europeia de “empacotamento” e

⁵⁸ Este tema será tratado mais detalhadamente no capítulo 4.

localização⁵⁹ da Microsoft. Desta forma, o que se nota é a internacionalização da prestação de serviços, seja de funções corporativas, seja dos serviços de baixo valor na cadeia produtiva do software, o que reforça a tendência de migração para os modelos de negócio baseados em serviços.

Em relação à regulação dos direitos de propriedade intelectual, pode ser feito um paralelo com o processo de internacionalização da indústria. A regulação foi se estabelecendo nos vários países ao longo dos anos, mas sob influência clara da experiência norte-americana, conforme apresentado nos capítulos 1 e 2.

3.1.2 A bolha da Internet e o crescimento do mercado de serviços

No final dos anos 80, ocorreu um importante avanço da tecnologia de informação, a interligação dos computadores em redes. O início do desenvolvimento deu-se no ambiente militar e das universidades americanas, ainda no final da década de 60, com a criação da ARPANET. O início comercial deu-se com a formação de redes corporativas (algo próximo do que chamamos hoje de intranet), que foram expandidas posteriormente.

O impulso para a criação das redes veio tanto das aplicações possíveis (e-mail, transferência de arquivos, grupos de trabalho), quanto da importância dos *mainframes* como repositórios de bases de dados, que eram acessados pelos usuários dos PCs. Além disso, as redes permitiam o compartilhamento de software e recursos computacionais e também a manutenção do software dentro da organização (Steinmuller, 1996).

No início da década de 90, surgiu a World Wide Web, um dos serviços da Internet, com a grande novidade do ambiente gráfico e amigável que se tornou certamente o serviço mais popular entre os usuários, e que é confundido como sendo sinônimo da própria Internet, o que será reforçado mais adiante com o crescimento dos *web services*. A arquitetura aberta da Web⁶⁰ foi um dos fatores de sucesso em relação às redes proprietárias (como a rede Microsoft).

A Internet levou à criação de novos mercados para a indústria de software. Segundo Cusumano

⁵⁹ Localizar um software significa não apenas traduzir os menus e documentação para uma determinada língua, mas também adaptá-lo aos diferentes teclados, moedas etc.

⁶⁰ Baseada no protocolos TCP-IP e HTTP e, originalmente, na linguagem html.

(2004), estes novos mercados ocorreram em três estágios:

1. oferecer serviços tradicionais de TI via *web*, como software e serviços de gestão de recursos (ERP);
2. criar novos produtos ou serviços em negócios tradicionais, como por exemplo livrarias (Amazon.com) e leilões (eBay); e
3. usar a Internet para criar serviços totalmente novos, como os serviços de buscas e portais como Yahoo e, mais recentemente, Google e serviços associados.

O primeiro estágio começou a desenvolver-se no auge da bolha, mas não floresceu propriamente devido a problemas de infra-estrutura (como largura de banda de rede e segurança das informações). Mas atualmente voltou a crescer e será a base do modelo de negócio Serviços Pagos na Web, discutido adiante. O segundo estágio foi o que ficou conhecido como a bolha da Internet. E o terceiro iniciou-se também na bolha, mas tem se intensificado no que está sendo chamado mais recentemente de “web 2.0” e de “efeito Google”, como será tratado adiante.

Em termos mais estritos, as atividades econômicas do segundo período não são propriamente de empresas de software, mas junto com os demais criaram uma forte demanda de software para comércio eletrônico, a começar do mercado de software para navegação na web (*browser*).

O mercado de navegador foi liderado inicialmente pelo software da empresa Netscape, visto que a Microsoft não apostou na Internet imediatamente (por apostar em sua rede privada). Entretanto, a Microsoft conseguiu em pouco tempo dominar o mercado de *browser*, ao usar a estratégia de disponibilizar seu navegador (Internet Explorer) junto com o sistema operacional Windows, o que resultou inclusive em uma ação anti-truste movida pelo governo norte-americano. A posição monopolista da Microsoft foi questionada pelo Departamento de Justiça dos EUA devido à venda casada do navegador com o sistema operacional⁶¹. Três fatos indicavam que a Microsoft detinha uma posição monopolista:

⁶¹ A principal ação começou em 1998, mas a empresa já tinha sido processada outras vezes desde 1991.

Primeiramente, a fatia de mercado de sistemas operacionais da Microsoft para PCs compatíveis com [o padrão] Intel era extremamente grande e estável. Segundo, a fatia dominante de mercado da Microsoft era protegida por uma alta barreira à entrada. Em terceiro e fortemente como resultado desta barreira, os clientes da Microsoft careciam de alternativas comerciais viáveis ao Windows (United States District Court for the District of Columbia, 1999, p.16).

O que para a Microsoft era um ciclo de retornos positivos, era para seus competidores um ciclo vicioso. Em 2000 saiu o veredicto de que a Microsoft deveria então ser dividida em duas unidades: uma para produzir o sistema operacional e outra para os demais programas. A Microsoft recorreu e fez um acordo.

Retomando a análise dos mercados da internet, o segmento de mercado não teve tratamento privilegiado no começo da década de 90, tendo sido ocupado por diversas empresas emergentes que atuavam com base em produtos inovadores. A Microsoft penetrou neste mercado por meio de aquisições de empresas emergentes e alianças com grandes empresas e grupos com competências complementares (Roselino, 1998, p.60). Por ter conseguido virar o jogo, a Microsoft é citada por Teece (2000, p.8) como exemplo de empresa que possui as capacidades dinâmicas:

As capacitações dinâmicas são mais fáceis de serem encontradas em firmas altamente empreendedoras, com menos níveis hierárquicos, visão clara, poderosos incentivos e grande autonomia (para garantir capacidade de resposta). A firma deve ser capaz de efetivamente mover-se com rápidos movimentos, como a Microsoft fez quando Gates reconheceu a importância da Internet. A minimização de custos e a otimização estática geram apenas vantagens menos importantes. Planos são frequentemente feitos e então descartados com prontidão. As companhias devem constantemente transformar e retransformar. Uma orientação de ‘missão crítica’ é essencial.

Por outro lado, também surgiram oportunidades para novos serviços de software, devido à necessidade de integração entre plataformas (mainframes, workstations e PCs), sob a forma de terceirização (*outsourcing*).

A **bolha especulativa das empresas ponto-com**, que durou entre 1995 e 2000, foi caracterizada

pela criação de muitas empresas com negócios relacionados com a Internet, pela supervalorização destas empresas e de suas ações na bolsa de valores de empresas de alta tecnologia NASDAQ (North American Securities Dealers Automated Quotation System), e também por grandes fusões, como a compra da empresa Time Warner pela America OnLine (AOL). Este foi um dos períodos mais acentuados de fusões e aquisições de empresa nas indústrias de TICs (ver Gráfico 3.1). Entre 1999 e 2000, as fusões e aquisições de empresas de TICs tiveram um crescimento de quase 2000% em relação a 1995, enquanto que este índice foi cerca de 600% nas outras indústrias.

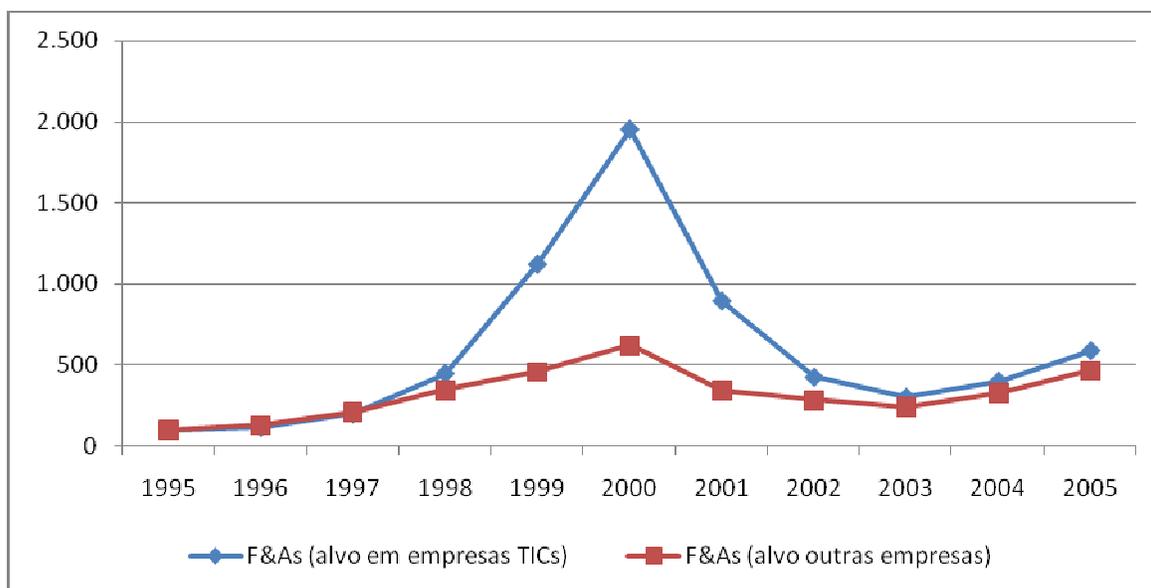


Gráfico 3.1 – Valor das fusões e aquisições nas indústrias de TICs e outras indústrias, 1995

Fonte: OCDE (2006)

Nota: preços correntes, índice 1995=100

Uma empresa ponto-com típica deveria explorar as externalidades de rede construindo uma grande base de usuários de seus produtos numa estratégia de “tornar-se grande rapidamente” (*get big fast*). Houve uma explosão de surgimento de empresas, sendo que muitas conseguiram levantar fundos via abertura do capital. O valor das ações era superestimado e surgiram vários milionários jovens e, geralmente, inexperientes.

Outra importante característica deste período foi a acentuada corrida por patentes devido à pervasividade das aplicações de software para comércio eletrônico baseado na Internet (Tang e Paré, 2003). Segundo Rivette e Kline (2000), as start-ups que buscavam financiamento achavam que suas

chances seriam maiores se possuíssem proteção por patentes para seus empreendimentos de risco. Um dos exemplos mais marcantes foi o da empresa Priceline.com:

A empresa, que conduz um negócio de leilões na web, ganhou notoriedade nacional no verão de 1998 quando conseguiu uma patente para seu método de fazer negócio – uma forma de leilão eletrônico dirigido aos compradores. O domínio desta patente ampla, diz Walker⁶², era crucial não apenas na atração do ex-presidente da Citicorp, Richard Braddock, para ser o CEO [Chief Executive Officer] da firma, mas também na alavancagem de capital de risco para a firma em num total de \$100 milhões – um montante quase sem precedentes para uma nova start-up⁶³.(Rivette e Kline, 2000, p.8).

Além da corrida por patentes, indicativo que o portfólio de patentes estava sendo usado como instrumento de investimento, os mesmos autores mostraram que novos tipos de negócio foram formados em torno da própria questão da propriedade intelectual: empresas para desenvolvimento de patentes, empreendimento para licenciamento de PI, comércio de licenciamento, entre outros. Seriam exemplos:

- Yet2.com – mercado de PI para licenciamento e venda de patentes de tecnologia criado em 1999;
- Real3D – *spin off* da Lockheed-Martin que recebeu US\$65 milhões da Intel e Silicon Graphics.

O exemplo da Priceline.com acima e a criação de empresas específicas para comercializar tecnologia ilustram bem **a função da patente de tornar a tecnologia passível de transação econômica, possibilitando a captação de capital de risco e a criação de mercados específicos para a comercialização destes DPIs**. Este provavelmente foi o período em que esta função ficou mais ressaltada nas indústrias de TI.

A bolha especulativa estourou no início de 2000, com a desvalorização das ações supervalorizadas

⁶² Jay Walker é proprietário da Priceline.com, empresa que atualmente oferece serviços de viagem, como reservas de vôos, hotéis, pacotes turísticos etc.

das empresas ponto-com. Desta forma, a queda vertiginosa ocorrida no índice NASDAQ teve o efeito de uma correção nos valores das ações.

Um dos motivos apontados para a queda foi o investimento pesado realizado em infra-estrutura, principalmente em Internet banda larga, e também para cuidar dos problemas que viriam com o Bug do milênio. Neste último caso, as empresas gastaram com programação e equipamentos em 1999, gasto que então não foi realizado em 2000. Entretanto, o legado de infra-estrutura permaneceu, base importante para os dois novos modelos de negócio. Além disso, ficou também a cultura dos negócios na Internet:

Durante 2001-2002, a maioria das empresas “ponto com” empalideceu ou faliu em função da ausência de consumidores suficientes e custos de infra-estrutura oscilantes. Mas o desenvolvimento de sistemas para habilitar as firmas a fazerem negócios na Internet tornou-se uma prática padrão para firmas da ‘velha-economia’, assim como para as novas. (Cusumano, 2004, p.116)

Dentre os motivos que explicam o estouro da bolha está o novo relacionamento criado entre direitos de PI e capital de risco, como explicam Coriat e Orsi (2002, p.1505):

Mesmo se vários fatores puderem ser usados para explicar a criação e o subsequente aparecimento súbito da bolha da Nasdaq (dado que as novas relações entre DPIs e capital de risco apenas constituíram uma das várias explicações para a bolha), a instabilidade que vai caracterizar o impulso desse mercado significa que deve haver um questionamento renovado sobre a sustentabilidade em longo prazo do modelo de inovação induzido pelas oportunidades de negócio (finance-driven) que estas mudanças recentes ajudaram a criar.

O efeito do estouro da bolha foi sentido fortemente na indústria de software. Empresas emergentes de comércio eletrônico eram vistas como a nova fronteira dos negócios. Depois do estouro da bolha, software deixou de ser “o milagre” que garantiria a competitividade das empresas, crença alimentada na década de 90. O crescimento do gasto com TI caiu e as empresas ficaram mais

⁶³ A patente da Priceline.com foi questionada. Mesmo com o questionamento, a empresa entrou com ação contra a Microsoft por infringir sua patente.

receosas por soluções que implicavam a troca de todo seu software por outra solução. Ou seja, queriam soluções que aproveitassem os legados e também queriam compartilhar riscos com os fornecedores.

Em 2003, Nicholas G. Carr publicou um artigo provocativo intitulado *IT doesn't matter anymore*, em que mostrava que TI estava se tornando *commodity*, no sentido que estava se transformando em mais um custo na produção das empresas, e não tinha mais um valor estratégico para a competitividade do negócio. O mesmo teria acontecido com outras tecnologias amplamente adotadas, como telefonia e energia elétrica. Saindo de uma posição estratégica, estes serviços se tornaram ubíquos, e assim invisíveis, não importando mais. Segundo Carr, os estudos sobre o gasto corporativo com TI consistentemente mostravam que grandes inversões raramente eram traduzidas em resultados financeiros superiores.

Este cenário restritivo em termos de investimento foi agravado com os atentados de 11 de setembro de 2001. Do ponto de vista das empresas usuárias, o momento não era de realizar mais investimentos, mas de meticulosamente gerenciar custos e riscos. Isto impulsionou as tecnologias de reuso de software (componentes de software e arquitetura orientada a serviços – SOA, que serão discutidas no capítulo 4), a terceirização de atividades de TI e da cadeia de produção de software (como visto acima) e a adoção de software de código aberto, que será discutido a seguir. Terceirizando as atividades de TI, as empresas buscavam cortar custos e também mudar o balanço contábil – de investimento e capital, as despesas com software e hardware poderiam passar para gastos correntes.

Os ganhos com licenciamento nos modelos de negócio de software produto ou customizável caíram e aumentaram os ganhos com manutenção e suporte de modelos baseados em serviços. Este aspecto pode ser marcado como o ponto em que os modelos baseados em produtos perdem relativamente espaço para os modelos baseados em serviços, inclusive com a criação de novas formas de licenciamento e comercialização do software, como o pagamento por uso. Desta forma, o crescimento do mercado de serviços foi acompanhado de mudanças estruturais na indústria e nos formatos utilizados pela indústria para assegurar a apropriação dos resultados econômicos.

O exemplo mais nítido do crescimento da importância dos serviços em relação ao software é o da

própria IBM. Oficialmente, a IBM é classificada na categoria Indústrias de Transformação – Divisão Fabricação de Equipamentos de Informática, Produtos Eletrônicos e Ópticos na classificação industrial internacional (SIC 3570). Entretanto, entre 1992 e 2002 o faturamento total pulou de US\$ 64,5 bilhões para US\$ 81,2 bilhões, a uma taxa média de 2,3% ao ano. No mesmo período, a taxa média de crescimento do segmento de serviços foi de 17,3% ao ano (OCDE, 2004). No Gráfico 3.2, é possível ver claramente a tendência de diminuição do percentual do faturamento advindo do hardware (de cerca de 50% para menos de 35%) e o aumento do segmento de serviços (de cerca de 10% para cerca de 50%).

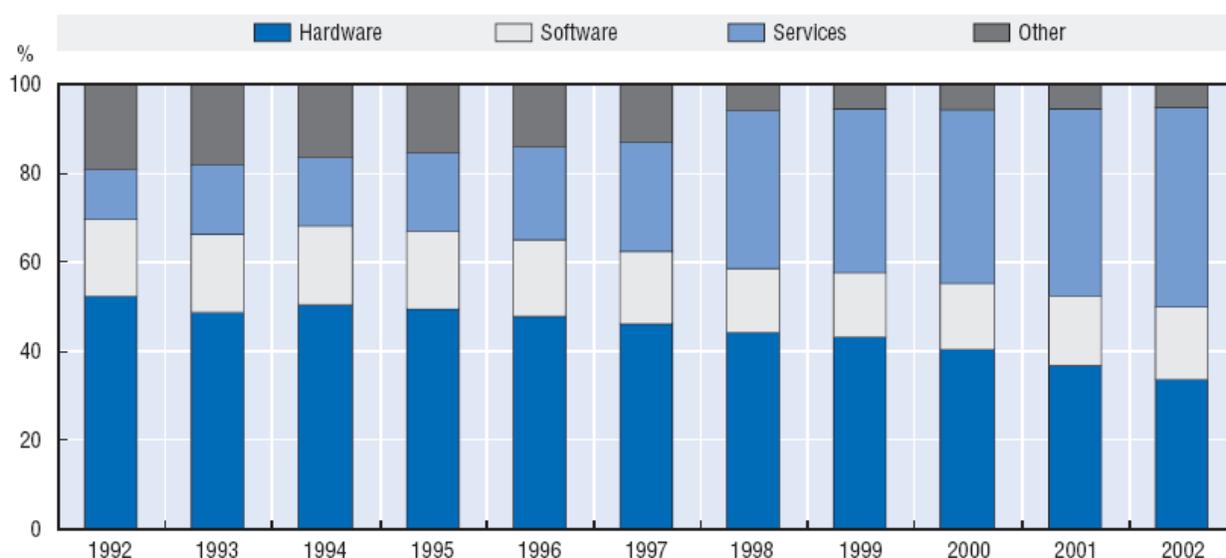


Gráfico 3.2 – Faturamento da IBM por segmento de mercado, 1992-2002

Fonte: OCDE (2004, p.311)

Esta mudança aprofundou-se com a recente venda da sua unidade de fabricação de PCs para a Lenovo, fabricante chinês, e com o rápido crescimento dos serviços globais de *outsourcing* de tecnologia da informação (ITO - Information Technology Outsourcing) e de processo de negócios (BPO - Business Process Outsourcing), que inclui serviços de recursos humanos, finanças, contabilidade etc. (OCDE, 2006; SOFTEX, no prelo). Isto mostra a tendência de focalizar o oferecimento de serviços a partir das unidades de *Global Business Services* descentralizadas no

mundo. Semelhante tendência pode ser encontrada na HP⁶⁴.

A raiz desta mudança na IBM pode ser localizada na criação do seu PC que, como discutido anteriormente, tinha a arquitetura aberta do hardware que permitiu que os concorrentes montassem PCs a partir dos componentes padronizados. Além de perder a batalha no hardware, perdeu também a batalha do sistema operacional para a Microsoft, ou pior ainda, não percebeu que ali se travava uma batalha importante. Restou ficar no nicho dos grandes computadores e depois migrar para a área de serviços:

Depois de perder a guerra do Sistema Operacional para PC, a liderança da IBM ficou confinada àqueles segmentos (sistemas para grandes computadores) que estavam enfrentando um declínio sistêmico. Nos mercados com crescimento mais rápido de Workstations e PC, foi uma derrota. Em resposta, a empresa buscou reposicionar a demanda para mainframes, primeiro por meio da promoção da arquitetura cliente/servidor e mais tarde pelo desenvolvimento de software que os habilitavam a servir como servidores web massivos. Enquanto isso, ela colocou ênfase crescente na venda de software e serviços, ganhando negócios baseada em sua habilidade sem igual de oferecer uma solução completa integrada end-to-end (West, 2003, p.1272).

Louis Gerstner⁶⁵, que se tornou CEO em 1993, criou a IBM *Global Services* para comercializar hardware, software e serviços de consultoria e manutenção, que incluía não só os produtos da IBM, mas produtos dos maiores competidores e também consultoria e integração de software open-source. Além disso, Gerstner decidiu investir em sistemas abertos, como Linux e Apache, e também na Internet, com a visão de *networked computing* para apoiar seus clientes nas aplicações de comércio eletrônico.

A IBM passou também a apostar no conceito de *on-demand computing* (semelhante ao conceito de *serviços na web* que será visto adiante), qual seja, a estratégia de tornar produtos disponíveis nos

⁶⁴ “Os serviços têm um peso cada vez maior na HP. No ano passado, eles representaram 18% da receita mundial da empresa. Pode parecer pouco perto da IBM - na qual responderam por quase 52% do faturamento - ou da EDS, que é uma empresa de serviços pura. Mas, a HP tem uma estrutura muito diferente dessas rivais” (Rosa, 14 ago 2006).

servidores da IBM e cobrar pelo uso, de forma semelhante aos serviços de água e energia elétrica, ao invés de vender produtos de software (Cusumano, 2004, p.108). Como resultado, o faturamento com serviços ultrapassou o da venda de hardware em 2002. A estratégia da IBM, ao abrir mão de trabalhar só com seus produtos, foi, ao invés de prender os clientes com tecnologias proprietárias, fazer o *lock-in* com serviços considerados insubstituíveis.

Esta tendência também tem se estabelecido em outras empresas, num mercado mundial estimado pela Gartner em mais de US\$ 600 milhões, no qual a participação das 20 maiores empresas chega a 39% (Tabela 3.1).

Tabela 3.1 - Receitas das Principais Ofertantes, 2003-2004 (em US\$ milhões)

Companhia	Receitas 2004	Market Share (%) 2004	Receitas 2003	Market Share (%) 2003	Cresc. Receitas (%)
IBM	46.423	7,6	42.635	7,5	8,9
EDS	20.601	3,4	20.610	3,6	0
Fujitsu	16.860	2,8	15.934	2,8	5,8
Hewlett-Packard (HP)	14.193	2,3	13.105	2,3	8,3
Accenture	14.142	2,3	12.150	2,1	16,4
Computer Sciences (CSC)	14.032	2,3	13.005	2,3	7,9
Outras	481.565	79,2	452.173	79,4	6,5
Mercado Total	607.816	100,0	569.612	100,0	6,7

Fonte: Gartner (2005 apud SOFTEX, no prelo).

As consequências deste redirecionamento para os serviços foram, por um lado, o corte no orçamento de P&D, o que fez com que a taxa de P&D sobre as vendas tenha declinado desde 1993. Por outro lado, Gestner iniciou uma estratégia mais agressiva de patenteamento e licenciamento (Bessen e Hunt, 2004a). A IBM é o exemplo citado de empresa que atua de forma mais **ofensiva**. Em 2006, a IBM obteve mais de 3600 registros de patentes, sendo a empresa que mais conquistou patentes nos EUA pelo 14º ano consecutivo⁶⁶. Tal portfólio pode ser usado para vários fins tanto defensivamente quanto para bloquear a competição.

⁶⁶ Louis Gerstner chegou na IBM em um momento de crise existencial e teve um papel importante na guinada para a área de serviços. Ele produziu um livro com a experiência que no Brasil foi traduzido como “Quem Disse que os Elefantes Não Dançam?”.

A IBM inverteu a lógica das externalidades de rede discutida no capítulo anterior. A estratégia das grandes empresas de software foi construir uma grande base de usuários voltada para o consumo de uma mesma solução de software, buscando ganhos de escala e *lock-in* em produtos proprietários (como processadores de texto, planilhas eletrônicas, entre outros).

Ao invés disso, a IBM buscou oferecer serviços em parte customizados (consultoria e integração) e em parte padronizados (suporte, manutenção, serviços de infra-estrutura de TI e BPO)⁶⁷, usando produtos e serviços seus ou de parceiros, buscando ganhos de escala, economias de escopo e *lock-in* agora em serviços considerados “insubstituíveis”. As economias de escala são obtidas com a negociação de contas globais com grandes corporações, sendo que os serviços são oferecidos a partir dos centros estabelecidos em várias partes do mundo conforme a tendência de internacionalização passiva, descrita acima. As economias de escopo são obtidas com a utilização dos vários ativos envolvidos (produtos, serviços, partes de código de software, conhecimento tácito, rede de parcerias, entre outros) para montar o menu de serviços entre os clientes globais, regionais e locais, que são customizados de acordo com as necessidades de cada cliente.

Esta é uma parte importante da estratégia da IBM para proteger seus ativos, retomando a lógica de solução total dos seus primórdios, antes da separação do hardware e software. Naquele tempo, o item mais valorizado era o hardware, que era protegido via patentes, e os demais itens eram “brindes”. Agora, continua a lógica da solução total centrada em serviços balizados com a marca IBM, mas o cliente pode escolher o “brinde”. Este pode ser visto como um caminho de apropriação econômica de ativos utilizados pelas grandes empresas de serviços (EDS, Accenture, Fujitsu, HP, entre outras).

3.1.3 Software livre e de Código Aberto: da “economia da dádiva” ao mainstream da indústria

O software livre e de código aberto (SL/CA) introduziu algumas variáveis novas na indústria de software, especialmente com relação à propriedade intelectual. Uma destas variáveis foi o

⁶⁶ Fonte: < <http://www.ibm.com/news/pt/pt/2007/03/pme.html> >

⁶⁷ A padronização se dá tanto com o reuso de componentes de software e a implementação de SOA para fazer a integração dos ambientes e serviços, quanto com a criação de serviços de TI padronizados.

estabelecimento de comunidades de desenvolvedores de software trabalhando de forma compartilhada via Internet. Com esta massa de gente desenvolvendo software, como se distribui a propriedade dos resultados?

De forma resumida, pode-se dizer que inicialmente a solução dada foi “a propriedade é de todo mundo”. Em geral, o que é de todo mundo, não é de ninguém em particular, o que significa que os comportamentos oportunistas poderiam se manifestar de forma acentuada, ou seja, seria muito fácil colher as contribuições de outros sem contribuir e “privatizá-las”. Para lidar com isso, foram estabelecidas as licenças SL/CA, sendo que a primeira delas tentou garantir que todos fossem beneficiados sem que ninguém se apropriasse mais que os outros. Depois surgiram outras licenças mais “liberais”.

Inicialmente, o software livre teve uma motivação bastante ideológica, motivação esta que ao longo do tempo foi se diversificando para abrigar interesses mais econômicos. Assim como introduziu novidades na indústria de software, o software livre foi mudando, amadurecendo e se profissionalizando, como constata César Taurion (2007), gerente de novas tecnologias aplicadas da IBM, ao observar as mudanças ocorridas no Fórum Internacional de Software Livre⁶⁸:

Quando começamos a debater Open Source com mais intensidade, isto lá pelos idos de 2000/2001, havia quase um consenso que os desenvolvedores que atuavam nos projetos Open Source eram 100% voluntários, o processo de desenvolvimento era anárquico, sem um road map claro e nenhuma preocupação com datas para entrega. Havia também a percepção que os fundamentos econômicos do Open Source eram intangíveis, a chamada gift economy. Hoje se sabe que nos projetos de maior sucesso cerca de 90% dos desenvolvedores mais ativos estão na folha de pagamento da própria indústria de software, existe uma organização clara, com road map definido (os exemplos da Linux Foundation, Eclipse Foundation e Apache Software Foundation são prova disso) e que existem sim, modelos de negócio que podem ser construídos em cima do contexto Open Source (Taurion, 15/04/2007).

Desta forma, o software livre passou da “economia da dádiva”, e de objeto de estudo de

⁶⁸ Evento anual realizado em Porto Alegre que reúne os principais ícones do software livre e de código aberto e que tem atraído cada vez a participação de grandes empresas.

antropólogos que analisavam as motivações não monetárias dos desenvolvedores das comunidades⁶⁹, para um ativo com um importante papel na indústria de software, tornando-se assim um fenômeno investigado por economistas devido à adoção rápida, investimento de empresas e inovação organizacional com o desenvolvimento compartilhado (Federal Trade Commission, 2003).

Indo mais devagar com o andar, o movimento surgiu com a denominação de Free Software em 1985 sob a batuta de Richard Stallman, um programador de inteligência artificial do MIT. Incomodado com a venda de trechos do código do sistema operacional Unix, e conseqüente “fechamento” do software, que tinha ajudado a construir, o que considerava moralmente errado, criou a Fundação do Software Livre (Free Software Foundation - FSF). A FSF tinha por objetivo desenvolver e difundir um mecanismo legal que preservasse o acesso livre aos programas. Como resultado surgiu uma nova licença de software denominada General Public License (GPL)⁷⁰ (West, 2003). A GPL instituiu quatro liberdades aos usuários e desenvolvedores:

- usar o programa, qualquer que seja o propósito;
- modificar o programa de forma a atender às suas necessidades;
- distribuir cópias do programa, gratuitamente ou não; e
- distribuir cópias modificadas do programa, de forma a permitir que outros usufruam das modificações introduzidas.

Apesar de explicitar as liberdades, a GPL é considerada a licença mais restritiva de SL/CA. A FSF também tinha o objetivo de desenvolver um sistema operacional livre baseado no Unix. Para isso, Stallman fundou o projeto GNU, que significa “GNU is Not Unix”, uma sigla recursiva, ou seja, que remete a si mesma infinitamente. Linus Trovalds iniciou, em 1991, a criação de um sistema operacional compatível com Unix para seu PC, solicitando ajuda de outros desenvolvedores. Em 1993, este sistema se tornou o Linux, disponível para download gratuito na Internet (West, 2003).

⁶⁹ Vários estudos foram feitos sobre a motivação dos desenvolvedores em participar das comunidades e contribuir com os projetos de desenvolvimento. Em síntese, as motivações encontradas incluem o aprendizado, diversão, fortalecimento de reputações (efeito vitrine) e satisfação das próprias necessidades (von Hippel, 2003). Mais recentemente foi estabelecida também a remuneração.

⁷⁰ A GPL também é chamada de copyleft, um contraponto ao copyright.

Até 1998, o software livre ficou restrito a certos círculos, em parte devido à ambigüidade do termo *free*, que tem o sentido de livre e de grátis. E também porque grande parte do desenvolvimento de software era direcionada para resolver problemas dos desenvolvedores-usuários. Em função disto, e para destacar o lado comercial deste tipo de software e aproximá-lo mais das empresas, em 1998, Eric Raymond criou a denominação Software de Código Aberto (*Open Source Software*) em seu paper denominado *A Cathedral e o Bazar*. Uma peça importante desta história, que catapultou o software de código aberto, foi a decisão da Netscape de abrir o código de seu navegador para aperfeiçoá-lo e dar-lhe vantagens na disputa que estava travando com o navegador da Microsoft, como já mencionado. Ao mesmo tempo, foi fundada a *Open Source Initiative* (OSI), responsável pela aprovação das licenças de código aberto. Para serem aprovadas, as licenças necessitam cumprir dez critérios⁷¹:

1. liberdade de distribuição, sem restrição de venda ou distribuição gratuita como parte de uma distribuição contendo programas de diferentes fontes, sem pagamento de royalties ou semelhantes;
2. programa deve incluir o código fonte no produto ou a indicação de como obtê-lo em formato que seja possível modificá-lo;
3. permitir modificações e trabalhos derivados;
4. garantir a integridade autoral do código fonte;
5. não discriminar pessoas ou grupos;
6. não discriminar o uso do programa em áreas de conhecimento, setores, atividades;
7. direitos de licença redistribuídos sem necessidade de licenças adicionais pelas partes;
8. a licença não deve ser ligada a um produto específico;
9. a licença não pode restringir que outros softwares sejam divulgados conjuntamente;

⁷¹ Fonte: <http://opensource.org/docs/osd>

10. a licença deve ser neutra em relação a tecnologia e estilo de interface.

Há diferenças nas duas denominações e movimentos, mas código aberto tornou-se mais popular⁷²: “Enquanto as idéias de software livre estão mais vinculadas às questões de garantia e perpetuação das liberdades citadas, as de código aberto estão mais ligadas a questões práticas de produção e negócio, como a agilização do desenvolvimento do software através de comunidades abertas” (SOFTEX, 2005, p.11).

As licenças de SL/CA apresentam diferenças entre si. Gosh (2006) divide as licenças em permissivas e recíprocas. As licenças permissivas estariam próximas do domínio público, pois permitem a seus usuários direitos amplos de usar, estudar, modificar e distribuir o software em quaisquer condições, como as licenças que nasceram em universidades norte-americanas (ex. licença Berkeley BSD (Berkeley Software Distribution) para o sistema operacional FreeBSD) e a licença Apache para o servidor web que se tornou líder de mercado. As licenças recíprocas procuram criar uma “proteção aos comuns”, isto é, garantir que os licenciados repassem os mesmos direitos que receberam de usar, estudar, modificar e distribuir o software por eles modificados. A licença criada por Stallman (GPL) é o maior exemplo de licença recíproca, pois não permite redistribuição com código fechado, limitando o uso comercial (Madanmohan e De’, 2004)⁷³.

Isto coloca uma nova forma de disponibilização do conhecimento gerado via “publicação” dos resultados, pois uma vez publicados estes conhecimentos passam a fazer parte do estado da técnica e não podem (teoricamente) ser registrados e protegidos via instrumentos formais de propriedade intelectual. E esta característica é um importante diferencial da indústria de software em relação a outras indústrias, aproximando-se da noção de ciência aberta (*open science*) (Coriat e Orsi, 2002).

As licenças menos restritivas são importantes na tentativa de se construir padrões e interoperabilidade. Ao longo do tempo, a liberdade de criar licenças resultou na multiplicação de destas, causando certa confusão, pois algumas licenças não funcionam bem juntas e podem causar confusão na hora que o usuário aceita seus termos. A multiplicidade de licenças mobilizou um grupo

⁷² No Brasil, o termo mais comum ainda é software livre, em parte devido à forte comunidade de usuários e desenvolvedores, ao uso por parte do governo e ao Fórum Internacional de Software Livre.

⁷³ Segundo Gosh (2006), a opção por uma licença recíproca pode ser vista como uma escolha “egoísta” no sentido que procura garantir seu próprio acesso aos desenvolvimentos futuros.

de trabalho na OSI que avaliou todas as licenças aprovadas, reordenando-as em categorias para auxiliar a escolha dos desenvolvedores, servindo como uma espécie de guia. Como exposto no Quadro 3.1, as licenças foram reunidas em 7 categorias. Há um pequeno conjunto de licenças mais frequentemente usadas. Mas há também um grande grupo delas que não pode ser reutilizado, pois foram criadas para fins específicos ou só podem ser usadas por determinadas organizações que as rotulam. Neste grupo é interessante observar a presença de grandes empresas de TICs, como Apple, CA, IBM, Nokia e Sun.

Para auxiliar o desenvolvedor a selecionar a licença mais adequada a seus propósitos, há um projeto de criar um serviço automático de ajuda, a partir de uma série de questões, o que ilustra bem a complexidade do assunto.

Quadro 3.1 – Licenças de Software de Código Aberto ordenadas por categorias

Licenças populares e amplamente usadas ou com grandes comunidades (9)	Licenças não reutilizáveis (24)
<ol style="list-style-type: none"> 1. Apache License, 2.0 2. New BSD license 3. GNU General Public License (GPL) 4. GNU Library or "Lesser" General Public License (LGPL) 5. MIT license 6. Mozilla Public License 1.1 (MPL) 7. Common Development and Distribution License 8. Common Public License 1.0 9. Eclipse Public License 	<ol style="list-style-type: none"> 10. Apple Public Source License 11. Computer Associates Trusted Open Source License 1.1 12. CUA Office Public License Version 1.0 13. EU DataGrid Software License 14. Entessa Public License 15. Frameworx License 16. IBM Public License 17. Motosoto License 18. Naumen Public License 19. Nethack General Public License 20. Nokia Open Source License 21. OCLC Research Public License 2.0 22. PHP License 23. Python license (CNRI Python License) 24. Python Software Foundation License 25. RealNetworks Public Source License V1.0 26. Reciprocal Public License 27. Ricoh Source Code Public License 28. Sleepycat License 29. Sun Public License 30. Sybase Open Watcom Public License 1.0 31. Vovida Software License v. 1.0 32. W3C License 33. wxWindows Library License 34. Zope Public License
<p style="text-align: center;">Licenças redundantes com as licenças mais populares (9)</p> <ol style="list-style-type: none"> 35. Academic Free License 36. Attribution Assurance Licenses 37. Eiffel Forum License V2.0 38. Fair License 39. Historical Permission Notice and Disclaimer 40. Lucent Public License Version 1.02 41. University of Illinois/NCSA Open Source License 42. X.Net License 	

<p style="text-align: center;">Licenças com propósitos especiais (3)</p> <p>43. Educational Community License 44. NASA Open Source Agreement 1.3 45. Open Group Test Suite License</p>	<p style="text-align: center;">Licenças que foram voluntariamente aposentadas (4)</p> <p>46. Intel Open Source License 47. Jabber Open Source License 48. MITRE Collaborative Virtual Workspace License (CVW License) 49. Sun Industry Standards Source License (SISSL)</p>
<p style="text-align: center;">Licenças que foram atualizadas e substituídas (4)</p> <p>50. Apache Software License 51. Eiffel Forum License 52. Lucent Public License (Plan9) 53. Mozilla Public License 1.0 (MPL)</p>	<p style="text-align: center;">Outras licenças (5)</p> <p>54. Adaptive Public License 55. Artistic license 56. Open Software License 57. Qt Public License (QPL) 58. zlib/libpng license</p>

Fonte: www.opensource.org

A participação das empresas no desenvolvimento de SL/CA implica algumas vezes no uso de licença dual, ou seja, as empresas possuem uma versão do software licenciada como SL/CA e outra como software proprietário. Valimaki (2003) estudou alguns casos de empresas que tinham feito desenvolvimento internamente, disponibilizavam o código-fonte e aceitavam contribuições externas. Entretanto, estas contribuições eram adaptadas ou reescritas de forma que a empresa pudesse manter o copyright sobre a totalidade do software. Esta prática é utilizada por empresas que desenvolvem componentes de software, como será discutido no capítulo 4, numa forma de aproveitar as idéias, mas não a expressão literal, e de certa forma diminuir os custos de transação deste tipo de desenvolvimento.

O SL/CA é fruto e, ao mesmo tempo, impulsionador da migração da indústria de software para os modelos de negócio baseados em serviços. Segundo O'Reilly (2005, 478-9), o SL/CA, mais do que um conjunto de licenças e práticas de desenvolvimento, está relacionado à padronização e commoditização e ao desenvolvimento de práticas associadas ao software como serviço. Segundo Gosh (2006), o SL/CA saiu do ambiente de pequenas empresas para reformular os modelos de negócio e as estratégias de grandes empresas. Olhando o SL/CA como um universo fechado, os principais “modelos de negócio”⁷⁴, apontados na literatura são:

⁷⁴ As aspas aqui se devem a que estes modelos são mais “formas de fazer dinheiro”, dado que não possuem a mesma complexidade dos modelos de negócios apresentados anteriormente.

- *Serviço integral: negócio baseado na venda do pacote físico (CD, booklets) e na venda de todo tipo de suporte ao software (treinamento, consultoria, pré-venda, desenvolvimento customizado, pós-venda etc.).*
- *Criação de clientela (loss leader): negócio não está baseado no SL/CA especificamente, mas este serve para criar hábitos e preferências que depois serão úteis para a introdução de software comercial proprietário baseado no SL/CA.*
- *Habilitando hardware (widget frosting): uso do software livre para drivers, interfaces ou mesmo sistema operacional visando à redução de custos e de preços do equipamento a ser comercializado.*
- *Acessórios: venda de itens físicos relacionados ao SL/CA (hardware compatível, livros, canecas, imagens etc.).*
- *Oferta on-line: desenvolvimento e oferta de SL/CA em sistemas on line cujo acesso é autorizado mediante pagamento de uma taxa de associação. Além disso, este modelo também apresenta ganhos com propaganda.*
- *Licenciamento de marcas: criam-se e licenciam-se marcas associadas a SL/CA.*
- *Primeiro vender, depois liberar: abertura do código após amortização dos investimentos, criando clientela para novos desenvolvimentos associados ao programa aberto (SOFTEX, 2005, p.59)*

Estes modelos se sobrepõem aos modelos de negócio da indústria de software como apresentado no Quadro 3.2.

Quadro 3.2 - Relações entre modelos de negócios específicos para SL/CA e da indústria de software

Negócios com SL/CA	Fonte de receita*	Modelos de negócios da indústria de software**
Serviço integral	Direta, indireta	SBV; SAV; CUST; PAC
Criação de clientela	Indireta	PAC
Habilitando hw	Indireta	EMB
Acessórios	Indireta	
Oferta on line	Direta, indireta	
Licenciamento de marcas	Indireta	
Vender e liberar	Indireta	PAC; CUST

Fonte: SOFTEX (2005)

Notas: * Fonte de receita direta: rendimento provém da venda de SL/CA; fonte de receita indireta: rendimento provém da venda de serviços ou produtos relacionados ao SL/CA

** SBV - serviço de baixo valor; SAV - serviço de alto valor; CUST - produto customizável; PAC - pacote; EMB - embarcado

A apropriação do retorno dos investimentos feitos no desenvolvimento ocorre de várias formas que envolvem venda de serviços complementares ao software, como consultoria, educação e treinamento, suporte, licenciamento dual (código proprietário adicionado ao aberto), software embarcado (Dahlander, 2004). Como já apontado, o SL/CA começou imitando o software proprietário commoditizado como sistema operacional, suíte de escritório etc. Para se apropriar dos resultados, as empresas de SL/CA tentam agora sair da competição por preço para oferecer distribuição customizada, resposta rápida e construir uma marca forte (O'Reilly, 2005). Este é o caso das distribuições do Linux, como Red Hat e Suse Linux.

Segundo o vice-presidente da divisão de Linux da HP, a discussão sobre propriedade intelectual no software foi acirrada com o advento do software livre, baseado em princípios de compartilhamento, o que contrasta com o segredo de negócio e restrições do software proprietário (Shankland, 2005). Neste sentido, o SL/CA seria a antítese do software proprietário: “ao invés de usar a proteção formal dos direitos de PI para estabelecer fronteiras entre os vendedores e seus competidores e clientes, Open Source alista todos como colaboradores, maximizando adoção por meio da cadeia de valor, mas minimizando as opções para apropriar receitas do software” (West, 2003, p.1264).

Entretanto, o software livre é balizado pela propriedade intelectual no sentido que se baseia no direito de autor:

Ao contrário do que vem sendo difundido, o software livre não questiona o direito de propriedade, e seu uso também se fundamenta no direito autoral. No entanto, apresenta uma nova abordagem sob a perspectiva da propriedade intelectual, pois, enquanto o copyright se baseia em restrições de cópia, distribuição e alteração do programa de computador; no âmbito do software livre, surge o copyleft, como uma alternativa para permitir que qualquer pessoa possa intervir, alterando, reproduzindo, redistribuindo e vendendo o software (Mendes, 2006, p.107).

Assim, o software licenciado como SL/CA pode ser legalmente protegido. Na verdade, quem desenvolveu o software tem direito de propriedade sobre ele (copyright automático e patente, se seguir seus trâmites), mas os cede aos demais usuários nos termos das licenças, como apresentam Graham e Mowery (2005, p.138-139):

Direitos de propriedade intelectual na realidade criam as fundações para a operação da licença de código aberto: direitos de cópia e direitos de patentes – na medida em que este último seja buscado – permanecem com o inventor do software de código aberto que passa estes direitos para outros desenvolvedores, permitindo que estes adotantes voluntários usem os direitos nos termos da licença.

Mas talvez a principal forma de fazer valer os termos das licenças SL/CA seja aquele realizado pela vigilância das comunidades de usuários, que costumam se mobilizar de forma ágil quando detectam “violações” das regras ou ameaças ao desenvolvimento do SL/CA. A mobilização destas comunidades pode ser danosa para imagem da empresa, como ocorreu com a Amazon quando esta empresa obteve uma patente para a invenção da compra com apenas um clique, uma invenção contestada por já fazer parte do estado da técnica⁷⁵.

Observando-se a indústria de software como um todo, o SL/CA mexeu com as posições de todos os atores. O Linux, considerado a maior expressão do SL/CA, alterou condições técnicas e econômicas da indústria em termos de estruturas de mercado e modelos de negócios. O Linux, desta forma, apresenta-se como uma alternativa ao design padrão dominante em sistemas operacionais:

A lógica disso é clara: o padrão proprietário Windows tornou-se um incômodo para alguns e um empecilho para outros. Nunca na história do capitalismo o mercado suportou por tanto tempo e de maneira tão ampla o pagamento de uma taxa de monopólio (virtual) a uma única empresa (Microsoft) para que todas as outras, literalmente, funcionassem. O atual padrão proprietário do mercado de sistemas operacionais, tal como se estruturou nos últimos vinte anos, é uma aberração do ponto de vista do capital e sua tendência é desaparecer (SOFTEX, 2005, p.64).

Desta forma, há várias empresas interessadas em apoiar o desenvolvimento do SL/CA, como o tem feito a IBM, por meio de apoio financeiro (contado em bilhões de dólares⁷⁶), de desenvolvimento de código e de outras iniciativas, e também se beneficiado bastante com isso:

⁷⁵ Outras iniciativas podem ser vistas no Anexo 1.

⁷⁶ Pelo menos US\$ 1 bilhão em 2001 (SOFTEX, 2005).

- A IBM abandonou seu servidor de web e aderiu ao Apache, inclusive contribuindo para melhoria do código com seus programadores;
- Em 1999 criou o *Linux Technology Center* para apoiar, e de certa forma direcionar, o desenvolvimento do Linux;
- Em 2000, junto com outras empresas (HP, CA, Intel e NEC)⁷⁷, fundou a *Open Source Development Labs*, que disponibiliza “recursos modernos e profissionais de computação e teste a desenvolvedores do Linux. Seu objetivo é direcionar e acelerar o desenvolvimento daquele *kernel* para uso corporativo, para o que contribui também com a participação dos usuários junto à comunidade desenvolvedora” (Gutierrez e Alexandre, 2004, p.56);
- Em 2004 doou 500 das suas patentes à *Open Source Initiative* (OSI); e em 2005, junto com NEC, Novell, Philips, Red Hat e Sony, investiu na criação da *Open Invention Network* (OIN), empresa de PI formada para criar um ambiente colaborativo e promover um ecossistema positivo e fértil para Linux⁷⁸. OIN atua comprando ou obtendo acordos de licenciamento cruzado de patentes relacionadas com este sistema operacional, para de certa forma criar uma “zona de conforto” entre as empresas de equipamentos de TICs que desejam utilizar o Linux, mas sem os riscos de possíveis litígios.

Toda esta movimentação da IBM reflete a estratégia de redução de custos, ampliação da participação em mercados e mudança de seus modelos de negócio (Gutierrez e Alexandre, 2004), como tratado anteriormente. West (2003, p.1273-4) reforça esta visão:

A princípio, o Linux providenciou um arcabouço comum de APIs [Application Programming Interface] por meio de sua linha de produção inteira, providendo uma arquitetura unificada para desenvolvedores de software. Em segundo lugar, um sistema operacional comparativamente imaturo (ainda que complexo) requeria serviços de suporte, uma tradicional força da IBM [...] Finalmente, o sistema operacional de código aberto permitiu à IBM fazer mudanças para melhorar sua diferenciação de hardware para seus clientes.

⁷⁷ Depois juntaram-se ao grupo Alcatel, AMD, Cisco, Ericsson, Fujitsu, Hitachi, Mitsubishi, Nokia, Novell, Sun, NTT, Toshiba e Unilever, além de grandes distribuições Linux como a Red Hat.

Como ilustrado pela IBM, o SL/CA provocou várias reações nas grandes empresas de software proprietário. Por um lado, houve a produção de versões (proprietárias) que rodam no Linux ou que trabalham com outros SL/CA, bem como o oferecimento de suporte integral ao Linux para criar “zonas de conforto” para os clientes que optassem por este sistema operacional. Segundo West (2003), a resposta das empresas líderes variou dependendo se usam software como uma fonte de vantagem competitiva ou se conservam outras fontes.

Como apontado no capítulo 1, as comunidades de SL/CA têm funcionado como ativos complementares para algumas empresas. Dahlander e Wallin (2006) estudaram algumas empresas que disponibilizam seus funcionários para ajudar no desenvolvimento de software em comunidades, numa estratégia de inovação aberta (*open innovation*). Esta contribuição profissional ajuda a aperfeiçoar a qualidade do software produzido e estes profissionais tendem a ocupar posições de destaque nestas comunidades, o que legitima a utilização pela empresa dos resultados do trabalho compartilhado, mas também pode ter o efeito de induzir o desenvolvimento do software para um determinado caminho que seja do interesse da empresa patrocinadora.

A interação do SL/CA com as empresas da indústria de software não foi sempre marcada por sinergia, pois houve reações agressivas que foram se abrandando à medida que o SL/CA, especialmente o Linux, se fortaleceu. Por exemplo, a Microsoft, após vários ataques ao Linux⁷⁹, tem procurado criar pontes com o código aberto. Segundo Bill Hilf, Gerente Geral de Estratégia de Plataforma da Microsoft, isto segue o que denomina “modelo donuts”. De forma semelhante ao famoso biscoito, o modelo das empresas de software possui um centro com os produtos principais e uma área complementar, onde são realizadas experimentações que procuram remeter o cliente para o negócio central: “o que a Microsoft quer é seu pedaço no doce, ou seja, abrir oportunidades complementares sem prejudicar seu negócio central, o Windows, do qual o Linux é o principal concorrente” (Fried, 2005).

Assim, mesmo sendo uma empresa que tem seu *core business* no software, a Microsoft não ficou indiferente ao Linux. Por um lado, estudou a ameaça e como competir neste ambiente e, por outro

⁷⁸ <http://www.openinventionnetwork.com/about.php>

⁷⁹ Vide a campanha “Get the facts” com várias estatísticas contrárias à adoção do Linux - www.microsoft.com/windowsserversystem/facts/

lado, estudou como tornar seus produtos compatíveis. Isso significa uma evolução no seu posicionamento em relação ao Linux. Como ele não vai “sumir”, é preciso conviver e aprender com ele.

Em março de 2007, a Microsoft estabeleceu um acordo de desenvolvimento compartilhado da integração do Windows com o Linux com a Novell, empresa especializada em soluções para redes que comercializa a distribuição SuSe Linux. O acordo possui também um caráter de não agressão mútua, pois prevê que a Microsoft não processará os clientes da Novell que, ao usarem Linux, possam violar suas patentes. O acordo inclui o pagamento de *royalties* numa balança “desbalanceada”, segundo Lawsky e Zawadzki (2006):

- A Novell pagará uma soma de US\$ 40 milhões em troca da garantia que a Microsoft não processará seus clientes;
- A Microsoft pagará US\$ 348 milhões em troca do direito de revenda ou distribuição do SuSe Linux para seus clientes.

Entre abril e maio de 2007, a Microsoft selou acordos de não agressão mútua semelhantes com as empresas Dell e SAMSUNG. Estes acordos representam de certa forma um paradoxo. Enquanto há quase um consenso que patentes de software representam uma ameaça ao software livre, por cobrirem invenções básicas do desenvolvimento de soluções criando um terreno minado, o acordo mostrou que pode haver um outro lado:

Enquanto o diabo estará nos detalhes, este acordo é uma boa coisa, e a Red Hat deveria fazer um acordo similar com a Microsoft. O compromisso de interoperabilidade e a proteção bilateral de patentes é uma boa coisa para consumidores que operam o Linux e o Windows (todos grandes consumidores tecnológicos). A proteção de patentes é uma boa coisa para os desenvolvedores de código aberto que estendem os projetos de código aberto cobertos pelos acordos, desde que eles agora recebam mais proteção do que tinham anteriormente. O Linux venceu e é hora de deixar a próxima fase começar. Os dias de kumbaya, quando vendedores davam-se os braços cantando músicas de amor ao código aberto para ‘aumentar o mercado’ por meio de cooperação (co-opetition) terminaram. O negócio de software é um negócio cruel. O Linux é agora tão importante que os vendedores de tecnologia estão lutando por vantagens

competitivas entre seus pares. É feio. É competição. E é bom para os consumidores (Yared, 2006).

Como visto no capítulo 1, este tipo de acordo embora não seja um licenciamento cruzado, tem uma função semelhante de diminuir os custos de transação em áreas de desenvolvimento pré-competitivo, como é o caso do papel do sistema operacional na infra-estrutura não só das TICs, mas de outros setores econômicos, dado o caráter pervasivo do software.

Além disso, o que se vê com o fortalecimento do software de código aberto é que se pode pensar o SL/CA e o software proprietário numa escala contínua em torno da tensão que existe entre adoção e apropriabilidade dos benefícios econômicos de um padrão. Estudando as estratégias das empresas responsáveis por plataformas, West (2003, p.1259) coloca as estratégias proprietárias e de código aberto como os dois extremos desta escala: “na construção da estratégia da plataforma para o século 21, os fabricantes líderes de computador deparam-se com o dilema de o quanto é aberta o suficiente para atrair compradores, enquanto conserva os retornos adequados”.

Desta forma, o SL/CA e o software proprietário não são antagônicos, mas devem coexistir e se mesclarem de acordo com diferentes estratégias. O mesmo raciocínio se aplica aos formatos de PI adotados. Se se cruzam os modelos de negócio da indústria de software com a apropriabilidade via o instrumento técnico de “fechamento do código” acompanhado da respectiva licença, é possível entender melhor estas estratégias. A Figura 3.1 aponta vários aspectos interessantes desta análise.

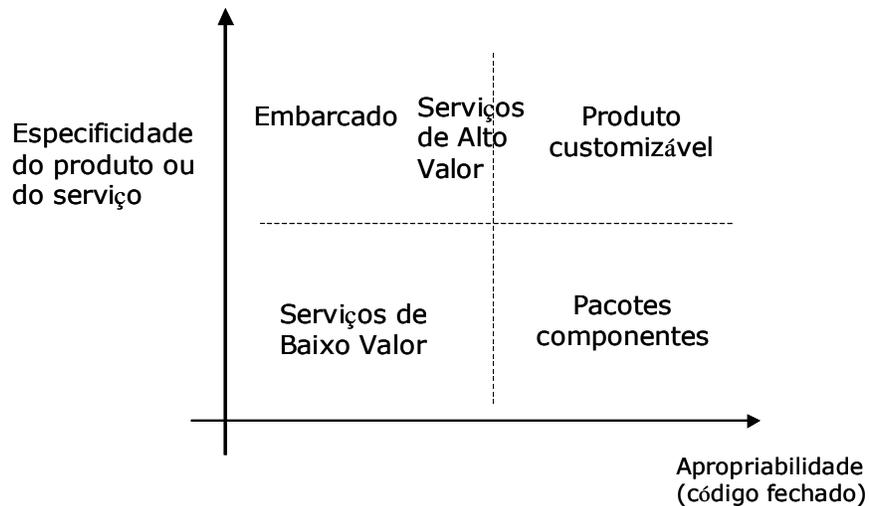


Figura 3.1 – Modelos de negócio da indústria de software ordenados segundo grau de especificidade e nível de apropriabilidade via código fechado

Fonte: SOFTEX (2005).

Assim sendo, pode-se observar:

- o SL/CA tende a ser mais comumente encontrado nos modelos de baixa apropriabilidade e baixa especificidade (serviços de baixo valor) e de baixa apropriabilidade e alta especificidade (software embarcado);
- nos modelos de negócio em que a apropriabilidade é alta, mas a especificidade do produto é baixa, como no software pacote, também é comum encontrar o SL/CA que visa substituir ferramentas de infra-estrutura, como sistema operacional, ferramentas de programação e mesmo suítes de escritório;
- o SL/CA tende a ser mais raramente encontrado nos modelos de alta especificidade e média e alta apropriabilidade, como serviços de alto valor e produto customizável. Mas já se encontram exemplos de estratégias que usam SL/CA nestes modelos, como os casos de ERPs e CRMs das empresas Jboss, SugarCRM e Zimbra

3.1.4 Web 2.0 e Serviços na web

No período mais recente, a tendência de crescimento dos modelos de negócios baseados em serviços se acentuou. Além disso, a web apresenta a tendência de se tornar cada vez mais uma plataforma de serviços, como demonstram os *serviços na web* e os vários serviços reunidos no que está sendo chamado de web 2.0, termo cunhado por Tim O’reilly em 2004, ainda de uso polêmico⁸⁰. Em resumo, as empresas da web 2.0 tentam recriar online as aplicações mais comuns que um usuário tem em seu desktop.

Desta forma, a web 2.0 aproximar-se-ia do software livre em relação à forma de lidar com o conteúdo dos sites, como mostra a Wikipedia, a enciclopédia construída de forma coletiva. Em termos de fenômeno econômico, a web 2.0 tem sido comparada à bolha da internet devido ao número de aquisições milionárias, como por exemplo, a aquisição da Intermix Media (proprietária do MySpace) pela News Corporation; do Skype pelo eBay e do YouTube pelo Google.

Dentre os exemplos, certamente o mais representativo é o Google, que deseja recriar a dominação da Microsoft a partir de um “ecossistema de aplicações”, só que na web (LaMonica, 2005). Seu negócio *core* é a venda de propaganda por meio dos serviços que disponibiliza on-line sem custos para o usuário final. Assim, o usuário é exposto a alguns anúncios quando faz uma busca na web, ou utiliza os outros serviços da corporação, e o Google é remunerado pelo anunciante cada vez que um usuário clica em seu *link*. A grande inovação do Google – que a fez sair do faturamento zero para US\$ 5 bilhões entre 2000 e 2004, abocanhar parte considerável do mercado de busca e tornar-se a marca mais valiosa do mundo⁸¹, foi a criação da busca paga, aliada à alta tecnologia do sistema de buscas (Battelle, 2006). A busca paga é a abordagem focalizada de vender anúncios relacionados com determinadas palavras usadas nas buscas.

O faturamento de propaganda online nos EUA teve um crescimento muito acentuado, até o ano 2000 seguido de uma queda em 2001 e 2002, relacionada com o fenômeno da bolha da Internet

⁸⁰ Segundo Tim Berners-Lee, considerado um dos pais da Web, ninguém sabe o que significa Web 2.0 (<http://arstechnica.com/news.ars/post/20060901-7650.html>). Na realidade, Web 2.0 é uma marca registrada da empresa CMP MEDIA LLC nos EUA.

⁸¹ De acordo com pesquisa realizada pelo jornal Financial Time e a empresa de consultoria Millward Brown, o Google se tornou a marca mais cara do mundo em 2006, avaliada em US\$ 66,3 bilhões, segundo o ranking BrandZ. Desbancou assim a GE, Microsoft e Coca-Cola (Folha Online, 23/04/2007).

como apresentado. Entretanto, após 2003 retomou o crescimento saindo de menos de 6 bilhões de dólares para atingir mais de 12 bilhões em 2005 (estimativa), como apresentado na Figura 3.2. A mudança do curso em 2003 deve-se provavelmente à mudança na composição deste faturamento. Se em 2000, quase metade do faturamento era originada de *banners* inseridos nos sites, em 2003 esta categoria diminuiu para 23% e o faturamento advindo de propaganda em sistemas de busca atingiu 35% do total, sendo que não passava de 1% em 2000 (ver Tabela 3.2).

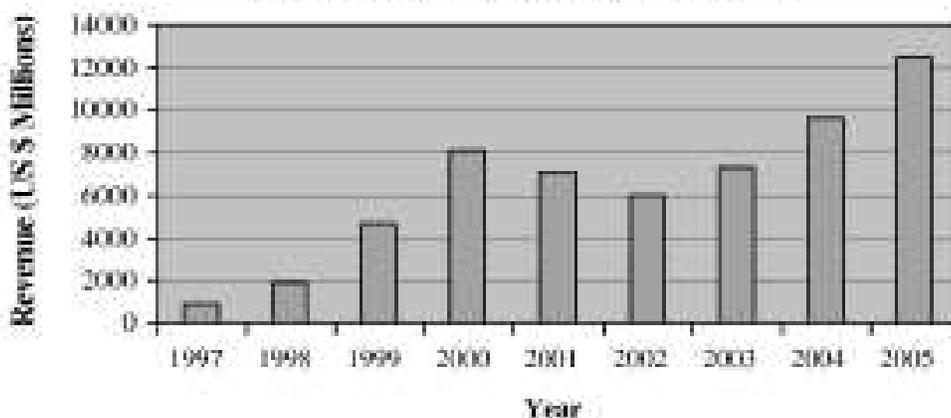


Figura 3.2 – Faturamento da propaganda on-line nos EUA em milhões de dólares, 1997- 2005

Fonte: Laffey (2007) a partir de dados do Interactive Advertising Bureau.

Tabela 3.2 - Propaganda online nos EUA por categoria, 2000-2004

	2000	2001	2002	2003	2004
Banner ads (%)	48	39	29	21	21
Sponsorship (%)	28	26	18	10	8
Search (%)	1	4	15	35	40
Classifields (%)	7	16	15	17	18
Others (%)	16	18	23	17	13

Fonte: Laffey (2007) a partir de Interactive Advertising Bureau.

A alta tecnologia do sistema de buscas é alimentada pelas externalidades de rede, pois quanto mais usuários o sistema consegue atrair, mais aperfeiçoado se torna. Isto porque a tecnologia de busca “aprende” com os usuários. Cada vez que um usuário faz uma busca, o sistema armazena a informação (palavras-chave usadas e links clicados) e cumulativamente vai usando isto nas

próximas buscas. Esta fonte de vantagem competitiva é também usada pelo site de leilões eBay e de venda de livros Amazon. É a idéia de “software com gente dentro”, como apresenta O’Reilly (2005, p.473):

Google tornou-se um caso mais engenhoso da estória do efeito de rede. A inovação inicial do Google foi o algoritmo PageRank, que alavancava as preferências coletivas dos usuários da web, expressas por seus links para sites, para produzir melhores resultados de busca. No caso do Google, a participação do usuário é extrínseca à empresa e a seu produto, e assim pode ser copiado pelos competidores. Se esta análise está correta, o sucesso no longo prazo do Google dependerá da descoberta de formas adicionais para alavancar o valor criado pelos usuários como parte da sua oferta. Serviços tais como Orkut e Gmail sugerem que esta lição foi apreendida.

A segunda grande inovação do Google foi agregar ao núcleo de busca uma série de outros serviços. O Google possui ferramentas específicas para buscar palavras limitando-se o escopo (apenas trabalhos acadêmicos, livros, imagens, blogs) e de localização (num site específico, no computador do usuário). Os serviços oferecidos, além da busca, incluem serviços comuns a outros portais como agenda, blog, e-mail (G-mail), listas de discussão, compartilhamento de fotos (Picasa) e vídeos (YouTube), serviço de rede social (Orkut), mapas e imagens de satélite (Google Earth) e serviços de mensagens instantâneas (Talk). O diferencial é o serviço de processamento de texto e de planilhas eletrônicas de forma compartilhada.

Aqui a estratégia do Google pode ser entendida como uma estratégia de integração, usando o esquema de Malerba e Orsenigo (1996) num contexto de alta pervasividade das oportunidades, alta complexidade da base de conhecimento, mas de baixa apropriabilidade. A tecnologia de busca paga provavelmente é patenteada, segmentada em várias patentes, pois o Google possui 54 patentes concedidas e 47 depositadas. Mas pode ser copiada por outras empresas que inventem ao redor. Por isso, para proteger seus investimentos feitos em inovação tecnológica, a empresa investe não apenas na inovação incremental da tecnologia, mas na agregação de ativos complementares que valorizem a tecnologia *core*. Estes ativos complementares incluem os demais serviços agregados, via aquisição

ou desenvolvimento interno, e certamente a marca valiosa Google⁸².

O sucesso do Google pode ser tomado como um indicativo de que o futuro do desenvolvimento está na web. A idéia é usar a web como plataforma para aplicativos que podem rodar em qualquer *browser*, imitando o ambiente de um PC. Isto tem sido chamado de “Efeito Google” e reflete a tendência de fornecer TI como um serviço, a partir da plataforma web.

O efeito Google está se alastrando também para grandes empresas de software. A Microsoft também está migrando para *oferecer serviços na web*, o que seria uma “revolução silenciosa” na estratégia da empresa. Um indicativo disso é o recente lançamento da plataforma Live, que reúne e-mail, serviço de mensagens, entre outros. Entretanto, ao mesmo tempo, a Microsoft lançou a versão 2007 do Office que implementa um novo formato para seus programas que é incompatível com as versões anteriores⁸³. O movimento da Microsoft é feito com dificuldades porque é uma empresa com uma estratégia forte de software para massa (LaMonica, 30 abr 2007). Ou a empresa pretende manter um pé em cada canoa até que os movimentos se consolidem.

Assim como ela, outros pesos-pesados de software produto para desktop, como Adobe, estão migrando via introdução de serviços online (LaMonica, 7 abr 2007). Isto é reforçado pelo crescimento do SL/CA de infra-estrutura (sistema operacional e servidor web) e em outros segmentos de software commoditizado, tanto em segmentos de software de massa, quanto em segmentos de software que podem rodar via web.

Como já apontado, os *serviços na web* também são marcados pela tendência de tornar a web uma plataforma de desenvolvimento pelo menos de três formas (LaMonica, 16 ago 2005). A primeira é o desenvolvimento compartilhado de software nos repositórios de SL/CA, cujo exemplo mais conhecido é o sourceforge.net. A segunda forma é o desenvolvimento compartilhado de conteúdo como realizado na Wikipedia.

A terceira forma é o desenvolvimento de funcionalidades adicionais (*add-ons*) que agregam valor ao conteúdo oferecido em um determinado site. Estas funcionalidades adicionais, chamadas de *mash-*

⁸² Além da marca registrada, Google já é usado como sinônimo de busca e já foi inserido como entrada em alguns dicionários.

⁸³ Esta versão também permite criar versões compatíveis.

ups, permitem que desenvolvedores independentes combinem informação de diferentes sites e, assim, controlem e customizem a informação. Um exemplo de mash-up são os sites que utilizam as imagens de satélite disponibilizadas pelo Google Earth como base para situar informações sobre imóveis disponíveis para aluguel, localização de restaurantes etc. Os *mash-ups* enriquecem os produtos de empresas que passam a fornecer serviços mais personalizados. No fundo, significa transformar os sites em máquinas programáveis.

Isto coloca questões relativas à propriedade intelectual. Ao mesmo tempo em que as empresas ganham com a personalização de seus produtos, precisam disponibilizar várias ferramentas e códigos para permitir que usuários desenvolvam *mash-ups*, o que mostra que os DPIs podem ter um papel de inclusão (de usuários e colaboradores) e não apenas de exclusão (de concorrentes). Entretanto, os ganhos ainda não são garantidos. Outro exemplo na mesma linha são os acordos firmados pelo site YouTube com produtoras de vídeo e estúdios de cinema para evitar litígio devido à violação de direitos autorais por parte de seus usuários. Os acordos de não-agressão envolvem inclusive a cessão de vídeos em troca do compartilhamento de receitas oriundas da publicidade.

3.2 A corrida patentária e os custos de transação

A indústria de software é muito diversa, considerando apenas as empresas que têm seu *core business* em software. Esta diversidade amplia-se consideravelmente se são observadas todas as indústrias que produzem software como parte de suas atividades (Roselino, 2006). Esta heterogeneidade deve ser levada em conta na hora de analisar as patentes em software. Em vista disso, inicialmente é discutida a corrida patentária que envolve patentes de software de empresas que têm o software como insumo importante, mas não é sua principal fonte de receita, o que aqui é chamado de corrida de patentes de software exógena à indústria de software. Depois, volta-se para a análise da corrida patentária de empresas de software, retomando os acordos de não agressão mútua que estão emergindo, uma forma das empresas lidarem com o aumento dos custos de transação que a corrida patentária pode trazer.

3.2.1 A corrida de patentes de software exógena à indústria de software

Bessen e Hunt (2004a) investigaram as diferenças de patenteamento de software entre os setores

econômicos. Como apresentado no capítulo 1, na década de 90 houve um aumento no número de patentes depositadas em geral e num ritmo mais acelerado de patentes de software. Comparando-se as séries de tempo do total de patentes e das patentes de software, observa-se que as patentes de software cresceram exponencialmente neste período, assim como o número total de patentes, mas não com o mesmo parâmetro. O crescimento das patentes de software foi maior (proporcionalmente) que o crescimento do número total de patentes (Tame, 2007).

Como apresentado no Gráfico 3.3 abaixo, utilizando a técnica de classificação de Bessen e Hunt (2004a)⁸⁴, na década de 80 cerca de 1.000 patentes foram concedidas por ano, o que aumentou para cerca de 5.000 por ano na década de 90, sendo que a taxa dobrou em 1996. Em 2002, foram concedidas 25 mil patentes. Trata-se de um crescimento muito mais acentuado que o do patenteamento em geral (Bessen e Hunt, 2004b).

Segundo Bessen e Hunt (2004a), entre 1987 e 1996, o número de patentes de software concedidas aumentou 16% ao ano, um aumento bem mais expressivo que o do gasto em P&D e o das taxas de emprego. Os fatores que podem ter contribuído para este crescimento foram: aumento da atividade de P&D em geral, especialmente da parte da P&D que usa software, maior produtividade no desenvolvimento do software, melhor relação custo-benefício da eficiência das patentes, entre outros fatores. Os autores acreditam que grande parte do crescimento aconteceu como resposta ao fortalecimento das patentes para as invenções de software.

⁸⁴ Na literatura são apresentadas pelo menos três formas de classificar uma patente como patente de software. A mais utilizada é a elaborada por Bessen e Hunt (2004a). No Anexo 2, as três técnicas são apresentadas e discutidas.

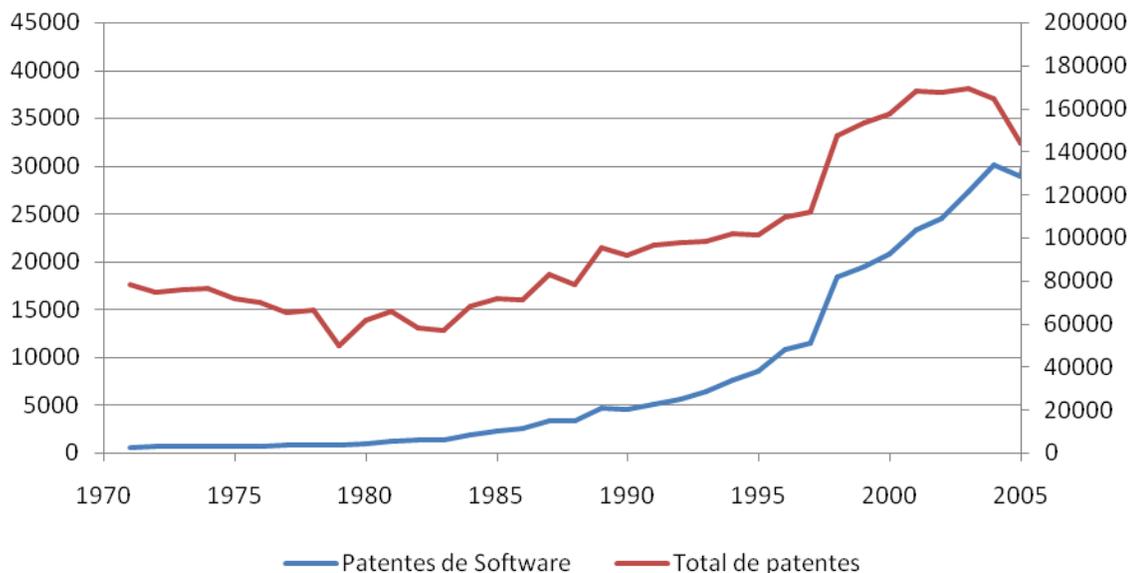


Gráfico 3.3 – Patentes concedidas nos EUA (total e patentes de software) – 1971 a 2006

Fonte: http://en.wikipedia.org/wiki/Bessen/Hunt_technique

Nota: A sequência do total de patentes foi plotada no eixo secundário à direita.

Segundo as análises de Bessen e Hunt (2004b) realizadas com dados do USPTO, a maior parte das patentes classificadas como patentes de software pertence a empresas que não são classificadas como empresas de software. Conforme apresentado na Tabela 3.3, apenas 5% das patentes de software são de empresas classificadas na classe SIC 7372 (equivalente à classe 7220 da CNAE 1.0 – Consultoria em Software: Desenvolvimento e Edição de Software Pronto para o Uso e Software sob Encomenda) e apenas 2% nas outras classes da SIC 73 (equivalente ao grupo 72 da CNAE 1.0). Apenas três setores da indústria de manufatura detêm 2/3 de todas as patentes de software (máquinas e equipamentos; material eletrônico e de aparelhos e equipamentos de comunicações e equipamentos de instrumentação médico-hospitalares, instrumentos de precisão e ópticos, equipamentos para automação industrial, cronômetros e relógios). Estes três setores detêm 54% das patentes em geral. As empresas que mais conseguiram patentes em software estão entre as que mais detêm patentes em geral – nove das dez empresas com mais patentes de software em 1995 estão entre as 20 empresas que mais obtiveram patentes em geral (Bessen e Hunt, 2004b).

Tabela 3.3 – Patentes concedidas para aplicações de software por indústria (1994-1997)

	(1)	(2)	(3)	(4)	(5)	(6)
	Software patents	Share of all		All patents	All Patents/ R&D	Software patent propensity (Table 5.2)
		Prog.	Prog. + Engineers			
<u>Manufacturing</u>	75%	11%	32%	88%	3.8	
Chemicals (SIC 28)	5%	1%	2%	15%	2.5	1.5
Machinery (SIC 35)	24%	3%	7%	17%	4.2	4.4
Electronics (SIC 36)	28%	2%	7%	27%	6.8	9.6
Instruments (SIC 38)	9%	1%	4%	11%	7.1	8.7
Other manu.	9%	5%	13%	18%	2.3	1.9
<u>Non-manufacturing</u>	25%	89%	68%	12%	3.0	
Software publishers (SIC 7372)	5%	} 33%	18%	1%	1.0	1.0 (all SIC 73)
Other software (SIC 737 exc. 7372, IBM)	2%			1%	2.8	
Other non- manufacturing	4%	55%	49%	4%	3.4	3.8
Addendum: IBM	6%	--	--	2%	5.0	

Fonte: Bessen e Hunt (2004a, p.49).

Isto se deve, em parte, à definição ampla de patente de software adotada por Bessen e Hunt, que tende a super-representar patentes que não são de software das empresas de hardware e de outras indústrias produtoras de equipamentos que usam software como produtos ou processos. Isto ocorre segundo Graham e Somaya (2004) devido à técnica de classificação das patentes de uso com o uso das palavras-chave “software” e “programas de computador” no título ou requerimentos das patentes. Esta ressalva não desqualifica a análise destes autores, mas mostra uma característica importante: a corrida patentária ocorre quase toda fora da indústria de software, sendo que no interior da indústria de software é concentrada em grandes players.

Segundo os autores, trata-se de um fenômeno principalmente norte-americano. 70% das patentes de software têm como proprietários empresas com base nos EUA, sendo que este índice é de 51% para as patentes em geral. Além disso, o proprietário típico de uma patente de software é uma empresa relativamente grande e bem estabelecida.

Usando análise de regressão, Bessen e Hunt (2004a) observaram um efeito de substituição da intensidade do P&D por patentes. Isto de certa forma inverte a lógica teórica de que a solicitação de uma patente é uma decorrência da atividade de P&D. E a patente obtida, por sua vez, representa um incentivo para o inovador continuar realizando inovações, ou seja, o mecanismo de recompensa presente na patente realimentaria a atividade de P&D. Entretanto, os autores observaram um efeito de substituição:

De acordo com a teoria tradicional do incentivo, tornar as patentes mais efetivas deveria aumentar a lucratividade das empresas que realizam invenções de software. Isto, por sua vez, induzir-nas-ia a aumentar seu gasto com P&D. Para testar se isto de fato aconteceu, nós usamos uma técnica empírica bem estabelecida de estimar as elasticidades do fator de substituição. Nós mostramos que a hipótese do incentivo pode ser refeita como a hipótese que P&D e patentes são complementos. Isto significa que aumentos de apropriabilidade do software deveriam resultar maior intensidade de P&D. Entretanto, não foi este o caso. As empresas que aumentaram seu patenteamento de software em relação ao total do patenteamento tenderam a diminuir a intensidade de P&D em relação às outras empresas. Este resultado é robusto a uma variedade de considerações econométricas e de outros tipos. Mas, novamente, este efeito é concentrado em indústrias conhecidas pelo patenteamento estratégico⁸⁵ (Bessen e Hunt, 2004a, p.4-5).

Os mesmos autores apontam que as mudanças na legislação da propriedade intelectual (e desta forma nas condições de apropriabilidade) devem implicar poucas mudanças na estrutura da indústria. Em geral, as empresas utilizam como principais mecanismos de apropriabilidade, o *lead-time*, o *learning-by-doing* e os serviços complementares, utilizando patentes de forma secundária:

Qualquer melhoria na apropriabilidade suportada pelas patentes não deve ter um efeito drástico na estrutura de mercado destas indústrias porque as firmas já possuem apropriabilidade significativa. [...] nós achamos que as patentes de software não devem influenciar as margens de lucro significativamente, sugerindo também que a estrutura

⁸⁵ As indústrias conhecidas pelo patenteamento estratégico são indústria de computadores (SIC 35 e IBM), eletrônica (incluindo semicondutores) (SIC 36), instrumentos (SIC 38).

industrial não deve mudar dramaticamente em resposta à disponibilidade de patentes de software (Bessen e Hunt, 2004, p.28).

3.2.2 A corrida patentária endógena à indústria de software

Tendo-se em conta que as empresas do setor de serviços têm uma baixa propensão a patentear, pode-se dizer que algumas empresas de software em pouco tempo adquiriram esta competência. Então, quem não tinha hábito de patentear suas invenções, começou a patenteá-las para futuros usos estratégicos, ofensivos ou defensivos. Mesmo quem possuía um discurso contrário ao patenteamento do software ou das invenções relacionadas com software correu para se proteger, como o caso da Oracle, conforme o testemunho apresentado no capítulo 2. Este tipo de estratégia é comum entre grandes empresas de software:

Como a Oracle, eles não visam fazer valer suas patentes contra seus competidores, nem buscam licenciar todas as patentes que eles poderiam infringir, na esperança que as outras empresas seguirão uma estratégia similar. Tal estratégia de “não agressão mútua” ou “tolerância mútua” foi comum também nos primórdios das indústrias de semicondutores e computadores (Bessen, 2003, p.1).

A patente é mais importante para empresas de software produto, em que parte do conhecimento é codificado, do que as empresas baseadas em modelos de negócio de serviços, nas quais são mais valiosos os conhecimentos tácitos (Allison *et al*, 2006). Entre as empresas de software, a patente é altamente concentrada. Allison *et al* (2006), analisando 646 empresas, encontraram uma média de 4 patentes e 14 depósitos por empresa. Entretanto, mais de $\frac{3}{4}$ das empresas não possuem nenhuma patente ou depósito.

Com o objetivo de observar a obtenção de patentes de algumas empresas, foi realizado um exercício de monitoramento tecnológico utilizando patentes de 16 empresas. As empresas foram selecionadas por terem participado das audiências promovidas pelo USPTO em 1994 ou por representarem tendências mais recentes da indústria de software, como Red Hat, uma das mais importantes distribuições do Linux. Foi utilizado o total de patentes de cada empresa, e não apenas as que se referem às invenções relacionadas com software.

O Gráfico 3.4 apresenta a distribuição das patentes por ano⁸⁶. Claramente, as empresas que atuam com hardware possuem um volume muito maior de patentes, especialmente a IBM como já mencionado, mas também a Intel, Sun e Apple. As empresas de software propriamente têm um volume bem menor de patentes e começaram a patentear bem mais recentemente. Mas já chama atenção o comportamento da Microsoft. Como já visto, na época do auge da bolha da Internet, entre 1999 e 2000, houve uma aceleração do depósito de patentes.

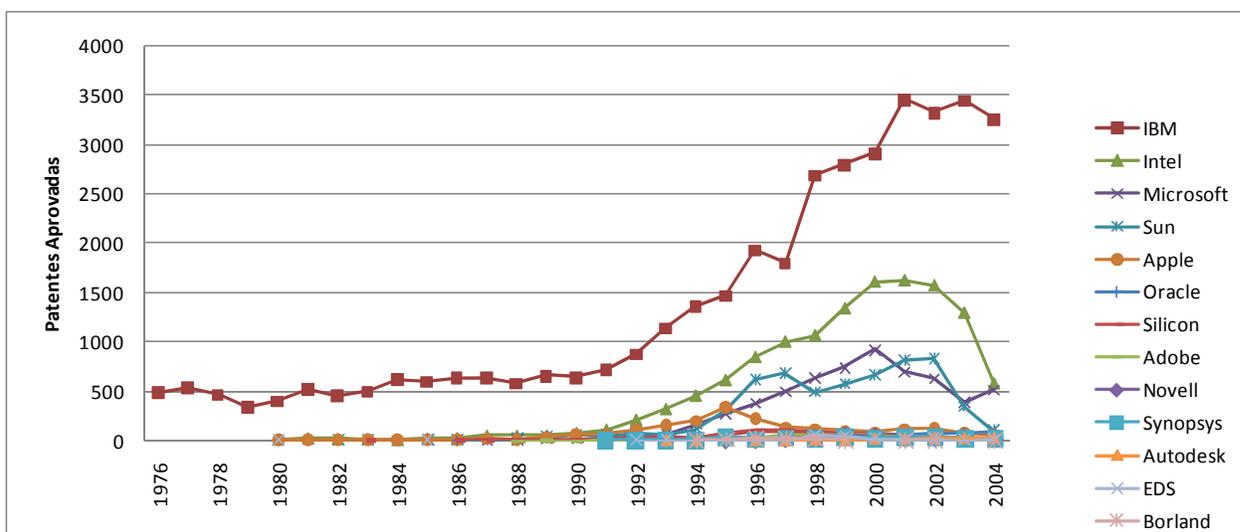


Gráfico 3.4 - Distribuição das patentes concedidas das empresas selecionadas (1976 – 2004)

Fonte: Elaboração própria a partir de dados do USPTO.

Separando as empresas segundo o posicionamento quanto ao patenteamento do software, apresentado na Audiência do USPTO em 1994, percebe-se que as empresas favoráveis à elegibilidade do software como matéria patenteável possuem um volume maior de patentes. Entretanto, numa escala menor, as empresas contrárias ao patenteamento possuem um montante razoável de patentes, considerando que começaram a depositar patentes num período posterior, como pode ser visto no Gráfico 3.5. Em geral, as empresas de software posicionaram-se de forma contrária e possuem relativamente menos patentes. A exceção é a Microsoft que posicionou-se favoravelmente e possui muitas patentes.

⁸⁶ A data da patente utilizada é a do depósito. Desta forma, os últimos 3 anos não são representativos, pois muitas das patentes depositadas neste período ainda encontram-se sob análise.

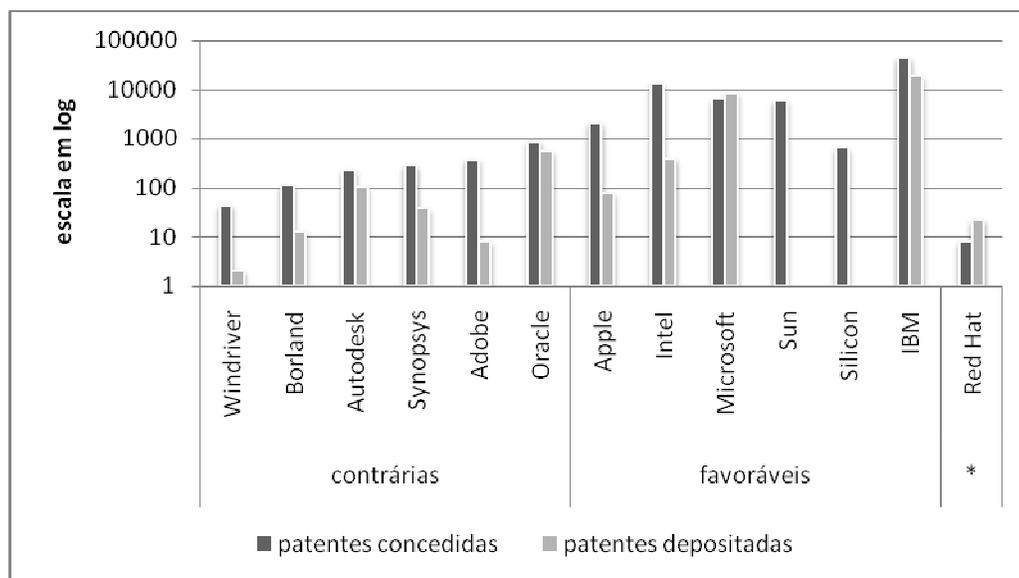


Gráfico 3.5 - Distribuição das patentes concedidas e depositadas das empresas selecionadas segundo posicionamento quanto ao patenteamento de software em 1994 (1976 – 2006)

Fonte: Elaboração própria a partir de dados do USPTO.

Nota: Eixo Y em escala logarítmica; *Red Hat não participou da audiência em 1994.

Em relação às patentes depositadas (ainda em análise), o comportamento da Microsoft é impressionante, aproximando-se a partir de 2003 do número total de depósitos da IBM (contando patentes de software e outras invenções), como pode ser visualizado no Gráfico 3.6. Com um volume muito menor, as empresas de software produto (Oracle, Autodesk) e também de serviços (EDS) têm depositado patentes. Mas se o regime tecnológico aponta para uma migração para serviços que dificilmente são protegidos com patentes, porque as empresas de software produto, especialmente a Microsoft, têm solicitado patentes?

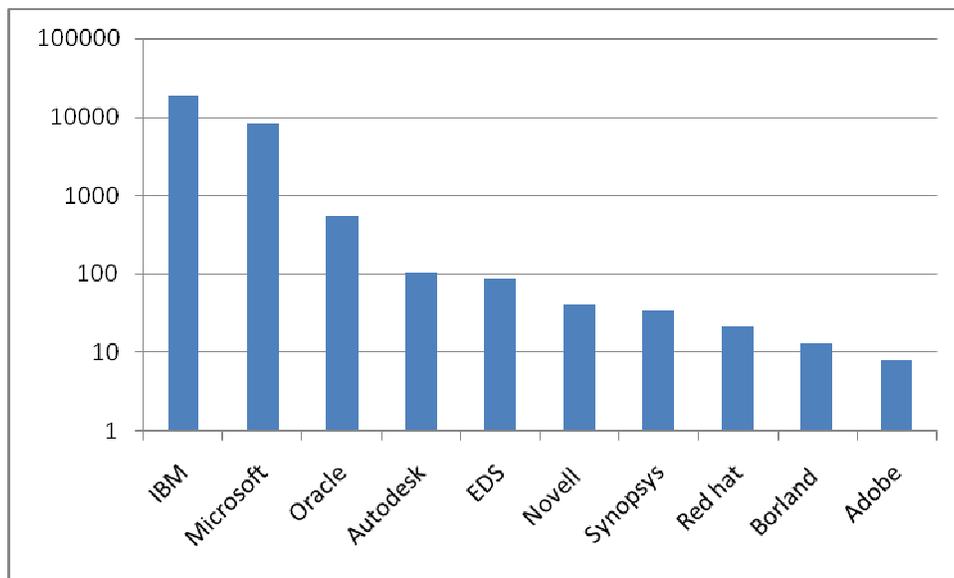


Gráfico 3.6 – Patentes depositadas de empresas selecionadas (1998 – 2005)

Fonte: Elaboração própria a partir de dados do USPTO.

Nota: Eixo Y em escala logarítmica.

Além das explicações para o crescimento no patenteamento do software como reflexo do crescimento dos retornos de investimento e das mudanças institucionais e legais, segundo Graham e Mowery (2005), a motivação para este comportamento é o pateamento defensivo de grandes *players*:

Outra explicação para o crescimento do patenteamento do software é que o aumento no patenteamento, especialmente por grandes empresas tais como Microsoft e IBM, reflete a importância crescente do patenteamento defensivo no software. Empresas rivais podem buscar patentes menos para apoiar o desenvolvimento comercial de invenções do que como meio para evitar os caros processos. “Patenteadores defensivos” requisitam um grande número de patentes para trocá-las em acordos de licenciamento cruzado preservando desta forma sua “liberdade para inovar” (id. ibid, p.151).

3.3 Propriedade Intelectual e os modelos de negócio – novos e recauchutados – da indústria de software

Assim, como o hardware se tornou uma *commodity*, pois a parte mais valorizada da combinação passou a ser o software, assiste-se agora uma migração do “centro de gravidade e de inovação” do software para os serviços, especialmente os serviços oferecidos via Internet. O exemplo mais conhecido hoje é o do “conglomerado” Google, que oferece, via web, serviços de busca de dados (de textos e imagens, de textos acadêmicos, de patentes, entre outros), compartilhamento de vídeos, fotos, suíte de escritório (processador de texto, planilha eletrônica), e-mail, entre outros.

Além dos serviços gratuitos, que são remunerados por meio da venda de propaganda, há também serviços pagos, como já mencionado. Depois de uma pequena experiência na bolha da Internet, tem renascido a utilização de sistemas de software externos à empresa, como o modelo de provedor de serviço de aplicação (ASP), por meio do qual uma empresa utiliza uma aplicação de software localizada em um provedor externo, agregada ou não de serviços ligados ao objetivo final desta aplicação (por exemplo, serviços de contabilidade).

A partir da idéia de Andy Rappaport e Halevi (1991) sobre “*computerless computer company*” (que as empresas de computador venderiam utilidade e não características técnicas melhores), Hornik (2007) sugere a idéia de “*softwareless software company*”, ou seja, as empresas de software venderiam utilidade traduzida em serviços, e não software.

Os acontecimentos apresentados neste período pós Internet implicaram em mudanças nos modelos de negócios (Quadro 3.3), além da criação de novos modelos de negócios (Quadro 3.4). Aqui se faz uma tentativa arriscada de apontar algumas tendências, pois os acontecimentos são ainda muito recentes, numa indústria que é também muito dinâmica.

O advento do SL/CA proporcionou novos modelos de venda em todos os modelos de negócio, inserindo um “componente” de serviços, além do software em si disponibilizado com o código fonte. Além disso, no modelo de serviços de alto valor, a disponibilidade de SL/CA para as camadas de infra-estrutura possibilita o tipo de estratégia da IBM, apresentado acima, que busca oferecer “serviços insubstituíveis” com hardware e software de terceiros, sob a chancela de sua marca e qualidade de processos.

Quadro 3.3 - Alterações nos modelos da indústria de software

Características	Serviço de baixo Valor	Serviço de alto valor	Produto customizável	Software embarcado	Produto pacote
Modelo de venda	direto	direto	direto, VARs e varejo	direto	VARs e varejo
Objeto de venda	projeto ou recurso + manutenção e suporte SL/CA – serviço integral (software + suporte)	projeto + manutenção e suporte SL/CA – serviço integral (software + suporte + consultoria)	licença e serviços de customização + manutenção e suporte SL/CA – serviço integral (software CA + suporte + consultoria + software proprietário)	licença e pequena adaptação SL/CA: habilitando hardware	Licença Pagamento por uso Licença dual SL/CA: Distribuição + serviços
Variável chave	custo (mão de obra)	utilização de capacidade	número de clientes		quota de mercado (base de usuários)
Risco do empreendimento	baixo	baixo			alto devido aos custos elevados de desenvolvimento e intenso P&D possibilidade de economias de rede no desenvolvimento do software
capacidade crítica	processo	processo e relação com cliente	análise de requisitos, relação com cliente e tecnologia	relação com cliente e tecnologia	produtos com ciclos de vida curtos e com inovações incrementais
Externalidades de rede e economias de escala	pequenos ganhos de escala	reuso do software pode levar a ganhos de escala lock-in em serviços insubstituíveis com hardware e software de terceiros			externalidades criam padrões dominantes que reforçam poder de mercado dos incumbentes
Barreira à entrada	competição	reputação	reputação e tecnologia	tecnologia	investimento
Exemplo	integração de sistemas, processamento de banco de dados para terceiros localização e internacionalização de de sw		ERP, CRM, produto vertical		processadores de texto
Empresa típica	Tata Services Consulting	IBM	SAP, Oracle	Ericsson	Microsoft e Red Hat

Fonte: Elaboração própria.

Nota: Alterações estão grifadas em negrito. Algumas características em que não houve alteração foram omitidas.

Como já discutido, além das alterações nos modelos, o SL/CA e os serviços na web fizeram emergir mais dois novos modelos de negócio:

- **Serviços Pagos na web (tipo ASP)** - oferta de serviço contratual para a implementação, hospedagem, gerenciamento e acesso a uma aplicação a partir de um local diferente do site do cliente (SOFTEX, no prelo). Este modelo de negócio já foi tentado antes, mas as condições de infra-estrutura não eram propícias, e foi tomado como uma moda ligada à bolha da internet. Mas recentemente voltou a crescer a partir das heranças da bolha (infra-estrutura, acesso à banda larga em grande escala e necessidade de redução de equipes de TI). O serviço que mais tem crescido é o software para gestão de recursos (ERP) (Ferreira, 2007).
- **Serviços gratuitos na web** (tipo Google) – como descrito acima, trata-se da oferta de serviços relacionados com aplicações horizontais de uma suíte de escritório, agregados a serviços típicos da web, como serviços de buscas.

O primeiro modelo é voltado mais para usuários corporativos e o segundo mais para usuários domésticos e ambos têm como variável crítica a segurança das informações, apesar de este ser um tema que preocupa mais as empresas.

A segurança, na realidade, é um dos termos do problema “*make or buy*”, pois as empresas precisam considerar os custos afundados em TI, o processo de seleção do fornecedor, os riscos, entre outros fatores, porque nunca é demais lembrar que os contratos entre partes são sempre peças incompletas.

Como mencionado, estes dois novos modelos de negócio ainda são restritos, mas tentam puxar o centro de gravidade da indústria para os serviços na web e não mais para software no desktop ou nos servidores das empresas. Como uma indústria marcada pela diversidade, há movimentos de manutenção (Ferreira, 2007).

Quadro 3.4 - Novos modelos da indústria de software

Características	Serviços pagos (tipo ASP)	Serviços gratuitos (tipo Google)
custo marginal de + 1 venda	Menos que constante	Virtualmente zero
Estrutura de mercado	Mercado fragmentado, mas tendendo à consolidação	Mercado fragmentado, mas tendendo à consolidação
Relação com cliente	um para vários com estreita interação com o cliente	Um para vários
Modelo de venda	Direto e VARs SL/CA: Oferta on-line	Direto com os clientes de propaganda
Objeto de venda	Serviços	Propaganda (banners, busca paga etc.)
Variável chave	Reputação/carteira de clientes	Grande base de usuários
Custos de transação	Altos com necessidade de Acordo de nível de Serviço (SLA)	
Capacidade crítica	Segurança das informações, qualidade de processos	
Externalidades de rede e economias de escala		Externalidades criam padrões dominantes que reforçam poder de mercado dos incumbentes
Exemplo	Outsourcing de Gestão Administrativa	“conglomerado” Google, que oferece, via web, serviços de busca de dados, compartilhamento de vídeos, fotos, suite de escritório (processador de texto, planilha eletrônica), e-mail, entre outros
Empresa típica	EDS	Google

Fonte: Elaboração própria.

No limite, os dois modelos podem se tornar iguais, se os serviços tipo ASP alcançarem grandes bases de usuários e, assim, garantir retorno por meio de propaganda como o Google. (Ferreira, 2007).

Também ocorreram alterações nas formas de apropriação dos retornos da inovação (Quadro 3.5). No modelo de serviços de alto valor, aumentou o uso da marca como forma de valorização dos serviços diferenciados oferecidos a partir de software de código aberto e software proprietário. A eventual participação das empresas deste modelo de negócio nas comunidades de desenvolvimento de SL/CA legitimam o uso que fazem do software resultante, bem como pode ter o efeito de induzir o roadmap do desenvolvimento segundo as necessidades da empresa. A

comunidade de SL/CA desta forma torna-se um ativo complementar da empresa.

No software embarcado, a utilização de SL/CA permite a customização do software desenvolvido, o que pode valorizar o produto da empresa. No caso do produto pacote, houve uma corrida patentária acompanhada de acordos de não agressão mútua na tentativa de diminuir custos de transação. Paradoxalmente, os programas desenvolvidos pelas comunidades de SL/CA na sua grande maioria visaram imitar os produtos pacote commoditizados. E um dos efeitos foi o início do lançamento de serviços com estes produtos na web.

No caso dos novos modelos, o ASP é protegido via relação contratual. Já os serviços do tipo Google usam uma variedade maior de DPIs como patente de método de negócio e de software e marca, mas também utilizam como forma de apropriação de seus esforços inovativos a própria inovação incremental, a integração de novos serviços (via aquisições de pequenas empresas inovadoras) e também os desenvolvimentos realizados pelos usuários, como o caso do *mashup* descrito acima, que adicionam valor a seus serviços.

Quadro 3.5 - Principais formas de apropriação dos resultados da atividade econômica dos principais modelos de negócio da indústria de software (com novos modelos)

Características	Principais instrumentos de apropriação (ativos principais e complementares)	Alterações
Serviço de baixo Valor	<ul style="list-style-type: none"> • Relação contratual entre fornecedor e cliente com acordo de nível de serviço (SLA da sigla em inglês) • Preço • Uso insignificante de DPLs 	
Serviço de alto valor	<ul style="list-style-type: none"> • Relação contratual entre fornecedor e cliente • Imagem da empresa dada por carteira de clientes e/ou certificação • No caso de uso de componentes, copyright e/ou patente destes • Suporte pré e pós vendas • Pequeno uso de DPLs (marca) mas crescente depósito de patentes 	<ul style="list-style-type: none"> • Aumento do peso da marca (caso da IBM de oferecer software CA commoditizado + software proprietário + serviços diferenciados) • Participação em comunidades de SL/CA
Produto customizável	<ul style="list-style-type: none"> • Licença • Disponibilização apenas do programa executável (código binário sem código fonte) • Suporte pré e pós vendas • Uso marginal de DPLs (segredo de negócio e copyright) 	
Software embarcado	<ul style="list-style-type: none"> • patente de invenção do equipamento e da funcionalidade do software • disponibilização do código binário e, às vezes, do código fonte • especificidade do software limita risco de cópia por concorrentes 	<ul style="list-style-type: none"> • Customização com SL/CA
Produto pacote	<ul style="list-style-type: none"> • grande base de usuários • Disponibilização apenas do programa executável • Licença • Inovação rápida • Marketing • Rede de distribuição, treinamento e suporte • Uso mais acentuado de DPLs: Copyright, segredo de negócio e patente • Ações de reforço do copyright (luta anti-pirataria) 	<ul style="list-style-type: none"> • Corrida patentária por parte de grandes empresas + Acordos de não agressão mútua • Lançamento do produtos pacotes como serviços na Web • SL/CA: Disponibilização do código fonte
ASP		<ul style="list-style-type: none"> • Marca • Relação contratual e suporte
Serviços Tipo Google		<ul style="list-style-type: none"> • Patente de métodos de negócio + patente de software • Marca • Inovação incremental da tecnologia • Integração de serviços através de aquisições • Desenvolvimentos feitos pelos usuários (mash ups)

Fonte: Elaboração própria.

Resumindo o capítulo 3, à primeira vista parece haver um movimento contraditório, pois de um lado observa-se a migração da indústria para modelos de negócio baseados em serviços, que como visto tendem a utilizar menos intensamente DPIs, além do fenômeno da commoditização do software. Por outro lado, observa-se o fortalecimento do regime de apropriabilidade do software, com a extensão dos estatutos legais para abrigar a proteção do software (copyright e patente), e também o aumento do volume de depósitos e concessões de patentes para invenções relacionadas em software.

O fortalecimento dos DPIs, especialmente das patentes de software, teve como efeito o aumento dos custos de transação na indústria de software. O aumento dos custos econômicos e de transação aconteceu devido:

- ao aumento da necessidade de licenciamento cruzado e outras formas de interação devido ao cipoal de patentes;
- ao aumento dos custos com o depósito de patentes e com a necessidade de monitoramento do ambiente, de percepção de situações oportunistas e da criação de departamento jurídicos e de planejamento e estratégico nas empresas;
- ao aumento dos custos para fazer valer patente (estratégia ofensiva) ou se defender (estratégia defensiva);
- às empresas chegarem mesmo a evitar entrar em uma área já coberta por um cipoal de patentes (muitas e cobrindo áreas sobrepostas).

Como visto ao longo do capítulo, trata-se de um processo ainda em reorganização dos modelos de negócio e atores da indústria. Os grandes players da indústria de software, especialmente o gigante Microsoft, que não tinham hábito de patentear invenções relacionadas com software produto, engrossaram o cordão da corrida patentária, um movimento maior que até então contava principalmente com corredores de outras indústrias, especialmente de equipamentos em que o software tem um papel importante. A motivação para a corrida foi principalmente o patenteamento defensivo, ou seja, a construção de um portfólio de patentes para ter o que

negociar em caso de acordos de licenciamento cruzado ou evitar acusações de litígio.

Outra fonte de corredores para esta maratona foram as empresas que nasceram na bolha da Internet, que procuraram patentear métodos de negócio baseados em software, agora objetos patenteáveis. Com software commodity e métodos protegidos, estas empresas podem oferecer serviços e tentar conseguir grandes bases de usuários, aproveitando as externalidades de rede.

A contradição se desfaz quando se situa o tipo de corrida que ocorreu, e ocorre. Trata-se de uma espécie de corrida de Fórmula 1, entre as grandes empresas que não podem abdicar de lançar mão do patenteamento como forma de proteção de seus ativos tecnológicos, mesmo que com relação à apropriação dos retornos da inovação a patente de software tenha um papel secundário, tendo os ativos complementares um papel mais relevante.

Já em relação às patentes de métodos de fazer negócio, que se tornou um objeto patenteável mais recentemente que o software (1998), o fato de empresas pequenas terem conseguido patentear invenções pode ser explicado, ao menos parcialmente, pela novidade do tema e pouca documentação do estado da técnica, resultando inclusive em patentes de baixa qualidade.

4. PROPRIEDADE INTELECTUAL E REUSO DE SOFTWARE

Este capítulo tem como objetivo analisar as implicações do regime de apropriabilidade no processo de desenvolvimento de software que se utiliza de uma tecnologia particular denominada componentes de software. Componentes são usados principalmente no desenvolvimento customizado e fazem parte de uma trajetória tecnológica transversal aos regimes tecnológicos da indústria de software (ver Figura 4.1) que busca maiores ganhos de produtividade e economias de escopo. Esta trajetória mais ampla é a trajetória do reuso, formada por tecnologias como componentes, orientação a objetos etc..

Isto se explica porque a atividade de software, apesar das diversas tentativas de amadurecimento de processo de software, ainda é uma atividade produtiva bastante artesanal, o que significa que grandes volumes de desenvolvimento de software são feitos partindo do zero, com pouco ou nenhum aproveitamento de desenvolvimentos anteriores.

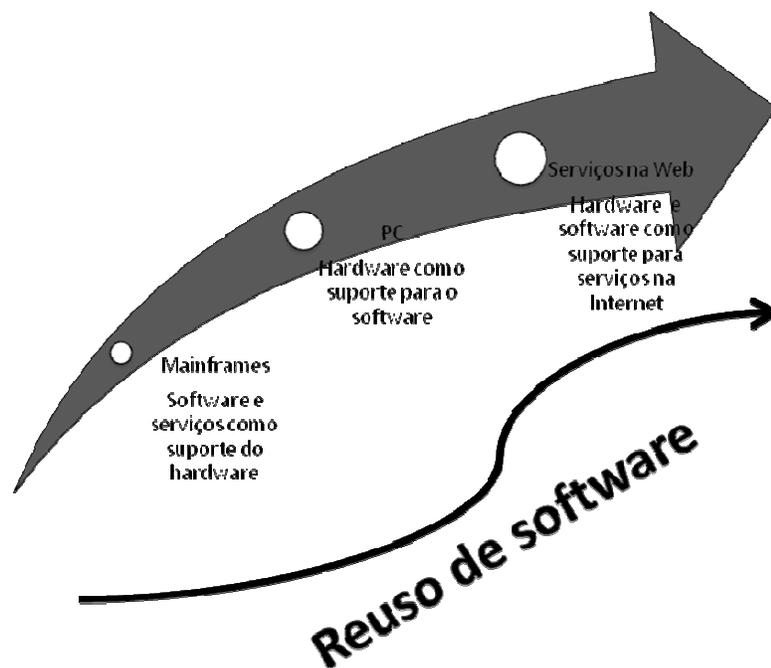


Figura 4.1 – Trajetória de reuso de software transversal aos regimes tecnológicos

Nota: Elaboração própria.

Desta forma, este capítulo tem como objetivo analisar as implicações dos regimes de apropriabilidade quando da “fabricação” do software, especialmente com o uso das tecnologias e arquiteturas de reuso de software. Analisa-se aqui uma dessas tecnologias em particular – componentes de software. Componentização é tomada, então, como caso ilustrativo da complexidade da apropriação na indústria de software, que deve-se a duas dimensões relacionadas. Em primeiro lugar, como indicado nos capítulos 2 e 3, há preocupações de que as patentes de software estejam criando um campo minado para a produção de software. Dificilmente os produtores de software, e mesmo de serviços de software, não infringirão patentes na hora de produzir uma nova solução⁸⁷. Esta preocupação é amplificada com a utilização de componentes de software.

Em segundo lugar, a complexidade advém também das diferentes licenças proprietárias ou de código aberto sob as quais um componente pode ser disponibilizado ou licenciado. As empresas integradoras, que montam as soluções de software a partir dos componentes, vêem-se na difícil tarefa de cuidar para não colocar na mesma solução componentes que tenham termos de licenciamento incompatíveis, bem como remunerar cada fornecedor de componentes de acordo com os respectivos contratos.

Este capítulo baseia-se nos materiais produzidos pela pesquisa *Perspectivas de desenvolvimento e uso de componentes na Indústria Brasileira de Software e Serviços*, realizada entre 2005 e 2006, pela Associação para Promoção da Excelência do Software Brasileiro em parceria com o DPCT/Unicamp sob encomenda do Ministério de Ciência e Tecnologia. Os resultados desta pesquisa chamaram a atenção para o fato que a proteção da propriedade intelectual no desenvolvimento e uso de componentes de software pode estar criando um clima de incerteza para as tecnologias de reuso. Em vista disso, este capítulo procura investigar as formas de apropriabilidade envolvidas com o desenvolvimento e uso de componentes e os direitos de propriedade associados.

A primeira seção do capítulo apresenta a busca de padronização de processos e partes na indústria

⁸⁷ Solução é um sistema de software, geralmente feito sob encomenda, que pode ser montada com o uso das tecnologias de reuso, como por componentes.

de software, que é tão antiga quanto a própria indústria, a trajetória de reuso na indústria de software, bem como a tecnologia de componentes e sua cadeia de valor. A segunda seção trata da complexidade da propriedade intelectual na área de componentes, especialmente dos custos de transação envolvidos.

A evolução das tecnologias de produção de software tem favorecido mudanças significativas na organização desta atividade. A produção de software não se dá de forma linear, mas por meio de etapas muitas vezes circulares. Os modelos maduros de produção de software buscam racionalizar a produção de software por meio de interações constantes com o demandante da solução de software e do uso de ferramentas automatizadas⁸⁸.

A inovação de processo em software tem sido caracterizada desde os primórdios desta atividade pelas tentativas de aplicação de princípios de engenharia da indústria de manufatura para o desenvolvimento de ferramentas de programação. Essa preocupação fez surgir uma disciplina especializada denominada engenharia de software voltada para a aplicação de uma abordagem sistemática ao desenvolvimento, operação e manutenção do software. Um exemplo paradigmático da aplicação da lógica da engenharia à produção de software é a fábrica de software, entendida como um conjunto integrado de ferramentas de desenvolvimento, procedimentos padrão e práticas de gerenciamento e organização em forma de matriz. As primeiras fábricas de software foram estabelecidas nos EUA e Japão no final da década de 60 e início da década de 70, com o objetivo de realizar a gestão estratégica e a integração do desenvolvimento de software para alcançar economias de escopo, aumentar a produtividade e flexibilidade (Torrise, 1998).

Desta forma, a indústria de software tem evoluído de uma produção informal e artesanal para um padrão de divisão de trabalho mais sistemático e científico. Como resultado, a adoção de técnicas de engenharia e aperfeiçoamento na organização do desenvolvimento tem gerado efeitos

⁸⁸ Um exemplo de modelo é o CMMI (Capability Maturity Model Integration), abordagem de melhoria de processo de software que provê os elementos essenciais de processos efetivos para as organizações, o que envolve não apenas o processo de desenvolvimento de software, mas as funções organizacionais também. O CMMI foi desenvolvido pelo Instituto de Engenharia de Software da Universidade Carnegie Mellon (SEI), instituição que também é responsável pela certificação de profissionais e instituições neste modelo. Fonte: <http://www.sei.cmu.edu/cmmi/general/>

positivos na produtividade. Entretanto, o uso das práticas de engenharia de software varia de acordo com o porte da empresa, sendo mais freqüente nas empresas de porte grande:

As experiências descritas são limitadas a grandes empresas de software. Estas possuem recursos e incentivos maiores para investir em engenharia de software e para impor uma disciplina gerencial mais forte a sua produção de software se comparadas com as pequenas empresas. Estas últimas organizam sua produção com base em sistemas de comunicação informais e em um estilo frouxo de gerenciamento. A maioria das pequenas empresas de software é frequentemente descrita como sendo baseadas em atividades artesanais e personalizadas com pouco reuso de ferramentas, métodos e códigos entre os projetos (Torrise, 1998, p.116).

Apesar de ocorrer de forma concentrada na indústria, as práticas de racionalização têm evoluído em direção à **modularização** da produção de software, o que permite fracionar o desenvolvimento de software em partes menores. Isto facilita a produção de sistemas complexos, possibilitando a maior especialização na distribuição das funções e a elaboração concomitante de diferentes partes do software (Roselino, 2006). Estes processos foram enriquecidos com a emergência da trajetória tecnológica de **reuso** na indústria de software.

Reuso é o uso de partes de software ou de conhecimentos existentes para gerar novo software. O objetivo do reuso é aperfeiçoar a qualidade do software e aumentar a produtividade (Frakes e Kang, 2005). A pesquisa de reuso possui mais de 20 anos e inclui diversas tecnologias como bibliotecas de reuso (que possuem subprogramas, códigos, funcionalidades, objetos), métodos e ferramentas de engenharia aplicada a domínios específicos (como finanças, logística, administração etc), programação orientada a objetos, componentes e, mais recentemente, arquitetura orientada a serviços (SOA) e serviços na web.

De uma forma geral, pode-se dizer que as práticas de engenharia de software têm evoluído em busca da padronização de práticas e processos e de economias de escopo no desenvolvimento de software, para não ser preciso a cada momento desenvolver toda a solução demandada a partir do zero. Desta forma, a componentização é um passo a mais na busca por produtividade e

racionalização dos processos. Isto afeta todas as etapas do processo produtivo de software, como será detalhado adiante.

Assim, componentizar é buscar economia de escopo, com diminuição de custos⁸⁹, pois a empresa pode usar várias vezes o mesmo componente. Com isso, aumentam o leque de opções de desenvolvimento e as oportunidades de negócios, o que pode ser caracterizado como um regime de alta oportunidade.

4.1 Engenharia de Software baseada em componentes e economias de escopo⁹⁰

Componente pode ser definido como “um pacote coerente de artefatos de software que pode ser independentemente desenvolvido e distribuído como uma unidade, e que pode ser composto, sem alterações, com outros componentes para construir algo maior” (SOFTEX, 2007, p. 9).

Uma imagem que ajuda a entender o que são e como funcionam os componentes é o brinquedo Lego, formado por várias peças com tamanhos e funcionalidades diferentes. As peças existem de forma independente e possuem especificações claras de interface, no caso, os pinos que mostram como elas se encaixam umas nas outras. Todas são intercambiáveis e podem compor sistemas diferentes. Algumas têm funções mais específicas que outras (como as rodas em comparação com os quadrados básicos). As principais características de um componente são:

- ser intercambiável e reutilizável;
- ter uma função específica e clara;
- possuir uma interface bem definida para interagir com os outros componentes e compor algo maior;
- deve possuir boa documentação que facilite o processo de busca, adaptação e integração e que ateste seu grau de confiabilidade e qualidade (Rossi, 2004).

⁸⁹ Como se verá adiante, a diminuição de custos não ocorre no curto prazo, pois para uma empresa começar a desenvolver soluções com o uso de componentes, é necessário um investimento inicial considerável na adaptação de processos internos e aprendizado.

Em resumo, um componente pode ser pensado como a solução de um problema que pode ser usado para resolver outro problema semelhante (Rossi, 2004). Os componentes visam, então, aumentar a eficiência do desenvolvimento do software e a qualidade⁹¹ dos sistemas de software (Ravichandran e Rothenberger, 2003).

A ESBC pode ser definida como o desenvolvimento que envolve a construção de sistemas de software usando componentes, mas também *frameworks*, arquiteturas e ferramentas adequadas. Tanto os componentes quanto as ferramentas podem ser desenvolvidos internamente ou adquiridos no mercado (Ravichandran e Rothenberger, 2003). A ESBC é um passo a mais no reuso de código, exatamente por possibilitar o reuso do componente “*como ele é*”, ou seja, o componente é reutilizado sem alteração de seu código e implementação, sem custos de desenvolvimento, apenas de “montagem” e parametrização. E dessa forma é também um passo na direção da busca de economias de escopo, ou seja, possibilita produzir mais e diferentes soluções com um mesmo componente.

Comparativamente, a ESBC possui algumas diferenças em relação à engenharia de software tradicional (SOFTEX, 2007). De forma didática e esquemática, a engenharia de software tradicional possui as seguintes etapas, sendo que algumas delas são realimentadas:

- a) análise dos requisitos do cliente;
- b) especificação do sistema a ser desenvolvido;
- c) aprovação da especificação pelo cliente;
- d) projeto da arquitetura do software;
- e) codificação e implementação do programa;

⁹⁰ Esta seção é fortemente baseada em SOFTEX (2007).

⁹¹ O componente ajuda a aumentar a qualidade do software por facilitar a localização dos erros e por já ter sido testado e usado outras vezes.

f) testes⁹²; e

g) implantação do sistema no cliente.

Neste modelo tradicional, pode haver reuso, como na etapa de projeto, na qual alguns desenhos de projeto podem ser reutilizados, ou na etapa de codificação, na qual é possível reutilizar funções já desenvolvidas em outros projetos. Entretanto, o reuso é marginal e a aplicação é desenvolvida quase a partir do zero. Já na ESBC, os passos esquemáticos são:

a) análise dos requisitos do cliente;

b) especificação do sistema a ser desenvolvido;

c) aprovação da especificação pelo cliente;

d) busca e seleção dos componentes que serão utilizados;

e) desenvolvimento das partes que não foram atendidas por componentes já existentes;

f) integração;

g) testes de integração; e

h) implantação do sistema no cliente.

Apesar de existirem passos em comum, as diferenças são significativas, pois desde a especificação do sistema, tudo é feito de forma diferente tendo em mente o uso de componentes: quais componentes estão disponíveis, quais terão que ser desenvolvidos ou adquiridos, como maximizar o reuso dos componentes etc. Componentes mudam a ênfase da programação (construção) para a composição (montagem) (Sauvé, 2004).

Os componentes são classificados quanto ao acesso ao código (*white-box* ou *black-box*) e quanto

⁹² Este modelo linear foi adotado para facilitar o entendimento. Na realidade, a interação com o cliente acontece durante todo o processo, bem como os testes não ocorrem apenas no final. Como exposto, os processos são mais circulares do que lineares.

ao nível de granularidade⁹³ (genéricos, de serviço ou específicos de domínio). Quanto à primeira classificação, os componentes são considerados *white-box* quando é possível ter acesso ao seu código fonte com permissão para adaptá-lo e alterá-lo (Ravichandran e Rothenberger, 2003). Os componentes *white-box* podem ter 3 origens: componentes desenvolvidos internamente à empresa, componentes feitos sob encomenda e componentes de SL/CA. Estes últimos representam uma das mais amplas fontes de componentes disponíveis atualmente, pois a forma de desenvolvimento do software de código aberto em geral é feita de forma modular, o que resulta na produção de muitos componentes.

Já os componentes *black-box* são aqueles em que não é possível acessar seu código. Para utilizá-los, o desenvolvedor precisa saber como eles funcionam e as informações de interface (APIs) referentes à interligação com outros componentes, o que deve ser informado na documentação do componente.

Quanto ao regime de apropriabilidade, o produtor de componentes *white-box* encontra-se em um regime de baixa apropriabilidade, enquanto o de componentes *black-box* situa-se em um regime de alta apropriabilidade.

Ravichandran e Rothenberger (2003) chamam a atenção que a adaptação de um componente *white-box*, embora maximize as oportunidades de reuso, não é tão simples como parece, especialmente se não tiver sido desenvolvido *in-house*, visto que a modificação do código exige um alto grau de familiaridade com os detalhes de implementação e nem sempre há documentação detalhada. Se o componente tiver sido desenvolvido fora, é preciso analisar o custo-benefício de alterá-lo. Dessa forma é compreensível que muitos componentes *white-box* sejam tratados como *black-box*, apesar das vantagens de corrigir problemas e de customização (Weinstock e Hissam, 2005).

Quanto aos níveis de granularidade (ou especificidade), os componentes podem ser classificados como:

⁹³ “Classificação baseada no tamanho e complexidade ou fase em que o componente pode ser reutilizado” (Rossi, 2004, p.62).

- componentes genéricos: são componentes de baixa granularidade e de uso comum em muitos sistemas, tais como os componentes de interface com os usuários (ex. funcionalidades de um formulário a ser preenchido pelos usuários);
- componentes de serviços: são aqueles que fornecem serviços especializados, mas que não são específicos do ponto de vista de domínio de aplicação, como componentes para tratamento de erros em comunicação de dados, geração de gráficos etc.;
- componentes de domínio: são componentes específicos para domínios de aplicação definidos, que implementam regras de negócios, como por exemplo, regras do setor financeiro ou de construção civil. São os componentes de mais alta granularidade.

Há uma relação inversa entre intensidade potencial de reuso e especificidade dos componentes: quanto maior for a especificidade de um componente (componentes de domínio), menor a taxa de reuso e a economia de escopo. É mais provável que os componentes genéricos sejam *white-box* do que os componentes de domínio. Os componentes genéricos podem ser considerados *commodities* entre os componentes, sendo regidos pelo preço ou por serviços associados.

4.1.1 Cadeia de valor da componentização

O regime tecnológico do reuso incentiva a criação de um mercado de componentes com empresas dedicadas a desenvolver, atualizar e dar manutenção em componentes, outras especializadas em desenvolver usando componentes e outras ainda a produzir os elementos de apoio, tais como ferramentas, certificação, treinamento etc.

Segundo SOFTEX (2007), os principais agentes do mercado de componentes são:

- **desenvolvedores de componentes de software:** desenvolvedores de software focados na criação de peças de software caracterizáveis como componentes;
- **integradores:** desenvolvedores de software que usam componentes para montar soluções;
- **clientes/usuários de softwares baseados em componentes:** que usam softwares para

realizar suas atividades de negócio;

- **intermediários (*brokers*):** reúnem a oferta de componentes e apresentam aos potenciais clientes (neste caso, os integradores citados acima) para venda;
- **avaliadores/certificadores de qualidade:** que atestam as funcionalidades e avaliam a qualidade de componentes de terceiros para que os clientes (integradores) tenham informações adequadas para escolha;
- **fornecedores de ferramentas de desenvolvimento:** para desenvolvimento de e com componentes;
- **fornecedores de capacitação técnica:** treinam os demais atores nas práticas de ESBC.

A Figura abaixo apresenta um fluxo da cadeia de valor do mercado de componentes.

(1) O **cliente/usuário** possui uma demanda de software e busca no mercado um **integrador**, empresa que desenvolve soluções de software. O integrador interage com o cliente para conhecer suas necessidades e a partir desta interação especifica os requisitos do sistema, tendo em mente os componentes que possui internamente;

(2) Os componentes que não existem na biblioteca interna da empresa são buscados no mercado. A busca é feita em **repositórios públicos**, em **intermediários (*brokers*)** e diretamente nos **produtores de componentes**;

(3) Os componentes localizados são entregues via alguma forma de licenciamento para o integrador;

(4) Os componentes que não estão disponíveis, ou que o processo de busca não conseguiu localizar, são encomendados a **desenvolvedores de componentes** que os entregam ao integrador (5) e podem também alimentar os repositórios e *brokers* (6). Dependendo da relação de custo-benefício, estes componentes podem ser desenvolvidos internamente também;

(7) Por fim, o integrador monta a solução com os componentes e entrega-a ao cliente.

Nesta cadeia há também as funções de apoio: fornecedor de ferramentas, fornecedor de capacitação técnica e certificador.

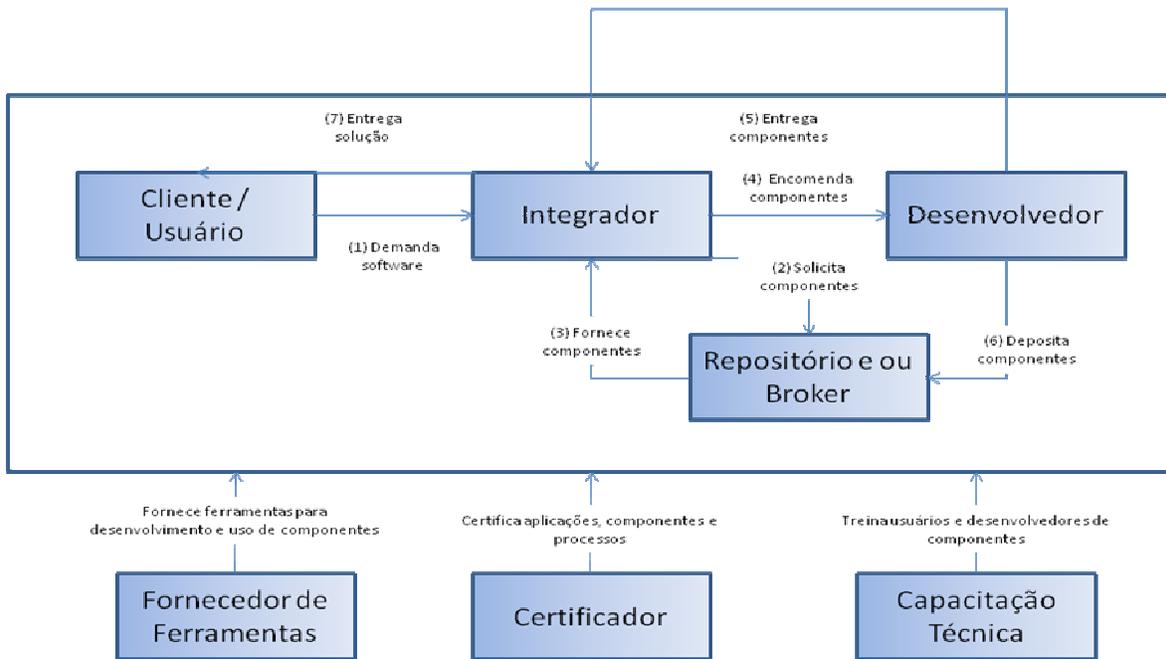


Figura 4.2 – Cadeia de valor da indústria de software com ESBC madura

Fonte: Adaptado a partir de SOFTEX (2007).

Como decorrência, a cadeia de valor pode ser decomposta em vários modelos de negócio (Bass *et al*, 2001; SOFTEX, 2007):

1. **Venda de componentes individuais:** desenvolvimento e comercialização de componentes individuais, por unidade, na forma de mercado de varejo. Exemplos: componentes de interface gráfica e componentes de serviços de média granularidade.
2. **Venda de famílias de componentes:** desenvolvimento e comercialização de linhas (famílias) de componentes na forma de mercado de varejo. Estas famílias constituem linhas completas de componentes dedicadas a um mercado vertical (domínio), como comunicação ou setor financeiro.
3. **Padrões de infra-estrutura:** compreendendo o conjunto de padrões, plataformas e

ferramentas de apoio ao desenvolvimento e gerenciamento de componentes.

4. **Integração/consultoria:** grandes conhecedores dos componentes, das plataformas e ferramentas disponíveis, estes desenvolvedores podem construir aplicações de forma eficiente e rápida para seus clientes, tornando este domínio sua vantagem competitiva.

5. **Intermediação (brokerage):** dedicados a oferecer uma área de compra-venda para componentes de software desenvolvidos por terceiros, criando uma marca consolidada e facilidades para busca e aquisição de componentes.

6. **Certificação:** que oferece aos consumidores de componentes garantias que os mesmos atendem determinados padrões, efetivamente oferecem determinados serviços e/ou apresentam determinados atributos de qualidade.

7. **Treinamento/capacitação:** A adoção da tecnologia de componentes pelas empresas exige profissionais capacitados, criando oportunidades para empresas que forneçam cursos profissionalizantes na área, tanto para empresas quanto para pessoas físicas⁹⁴.

Após esta apresentação dos princípios da componentização, passa-se a apresentar alguns resultados do Painel de Especialistas⁹⁵ realizado em abril de 2005 com o objetivo de sistematizar as principais tendências de desenvolvimento e uso de componentes⁹⁶, que resumem as questões que serão tratadas adiante neste capítulo.

Considerando os **fatores técnicos** que influenciam o desenvolvimento da ESBC, houve a percepção positiva, mas moderada, de que os fatores técnicos representam estímulos para o desenvolvimento e uso de componentes⁹⁷. Dentre os elementos de estímulo mais importantes,

⁹⁴ Modelo acrescentado durante a pesquisa tendo em vista a realidade brasileira.

⁹⁵ O Painel de Especialistas reuniu cerca de 40 participantes em torno de uma agenda estruturada durante um dia. Estes dados são resultados da 1ª questão trabalhada em grupos sobre os fatores que influenciam a tomada de decisão de investimento no desenvolvimento e no uso de componentes de software. Foi apresentada uma lista de 10 fatores técnicos e 10 fatores econômicos, sendo que os participantes acrescentaram mais 3 em cada conjunto.

⁹⁶ A agenda de trabalho do painel possuía questões sobre o Brasil e o mundo. Aqui são utilizados os resultados mais gerais.

⁹⁷ Trecho fortemente baseado em SOFTEX (2007).

foram apontados:

- a) qualidade do software produzido;
- b) manutenibilidade (característica do componente de ser corrigido, melhorado ou adaptado para mudanças em ambiente, e em requisitos e especificações funcionais);
- c) confiabilidade (característica do componente que o faz se manter em um determinado nível de desempenho, quando usado sob condições especificadas); e
- d) evolução da Arquitetura Orientada a Serviços (SOA) (adoção de Arquiteturas Orientadas a Serviços na indústria de software relacionadas com componentes).

O grau de maturidade da tecnologia foi considerado um desestímulo de moderado a alto. Assim, as expectativas são positivas, mas o nível de incerteza sobre os estímulos técnicos ainda é elevado, particularmente pelo grau de maturidade da tecnologia (Vitharana, 2003). Isto significa que apesar de algumas características técnicas serem bem avaliadas, o conjunto da tecnologia de componentes ainda não amadureceu o suficiente para uma adoção mais ampla, especialmente os aspectos relativos à padronização de mecanismos para garantir a interoperabilidade.

Quanto aos **fatores econômicos**, em geral, houve percepção muito positiva sobre os estímulos econômicos, mais do que com relação aos estímulos técnicos. Dentre os mais positivos encontram-se:

- a) produtividade;
- b) *time to market*;
- c) geração de oportunidades de negócios.

A percepção sobre as implicações de custos relacionados ao desenvolvimento e manutenção de software é positiva. Considerou-se a oferta de componentes de SL/CA como um estímulo de moderado a forte. Os custos de migração foram considerados um moderado desestímulo. Outro

desestímulo, este de moderado a forte, foi a disponibilidade de recursos humanos capacitados nesta tecnologia.

Assim, as expectativas quanto aos fatores de estímulo econômicos são positivas, particularmente pelo fato de que componentes contribuem para a ampliação das oportunidades de negócios e para ganhos de produtividade relacionados ao encurtamento do tempo de desenvolvimento e da entrega dos produtos e soluções.

Componentes são vistos como um caminho para ganhos de produtividade e para ampliação de oportunidades de negócios (economia de escopo). **Seu principal gargalo ainda é a incerteza sobre a maturidade da tecnologia em termos de padronização para interoperabilidade.**

Além dos fatores considerados acima, os participantes apontaram um gargalo adicional, não previsto inicialmente na agenda estruturada de trabalho. Este gargalo refere-se ao regime de propriedade intelectual, visto que entre outros aspectos, a gestão dos direitos de propriedade é extremamente complexa, tanto do lado de quem o produz, quanto do lado de quem o usa, e o quadro regulatório é incerto.

O que se depreende da análise dos fatores que influenciam a adoção da tecnologia de componentes é que há dois tipos de incertezas. O primeiro são as incertezas técnicas relacionadas com a maturidade da tecnologia, especialmente com os aspectos de interoperabilidade, como padronização de interfaces e de mecanismos de descrição e busca de componentes. O segundo é o relativo aos mecanismos de apropriabilidade, especialmente com relação aos desenvolvedores de componentes, que arcam com os custos iniciais mais elevados e necessitam de garantias de remuneração para diminuir os riscos elevados do investimento.

4.1.2 A oferta de componentes

Não há estudos recentes com estimativas do tamanho do mercado de componentes, incluindo-se o desenvolvimento e comercialização de componentes e ferramentas e as atividades de certificação, e nem da amplitude da difusão das práticas de desenvolvimento com componentes. As

estimativas disponíveis foram feitas entre 1999 e 2000, portanto antes do estouro da bolha, o que leva a se desconfiar do otimismo quanto ao estabelecimento e crescimento deste mercado, conforme apresentado no Gráfico 4.1. O estudo da PricewaterhouseCoopers (PWC) previa que em 2002 o faturamento mundial com a venda de componentes ultrapassaria US\$ 1 bilhão, chegando a US\$ 10 bilhões em 2005. Já a estimativa da IDC, apesar de partir de números mais altos nos primeiros anos em relação à estimativa da PWC, apostava num crescimento mais lento.

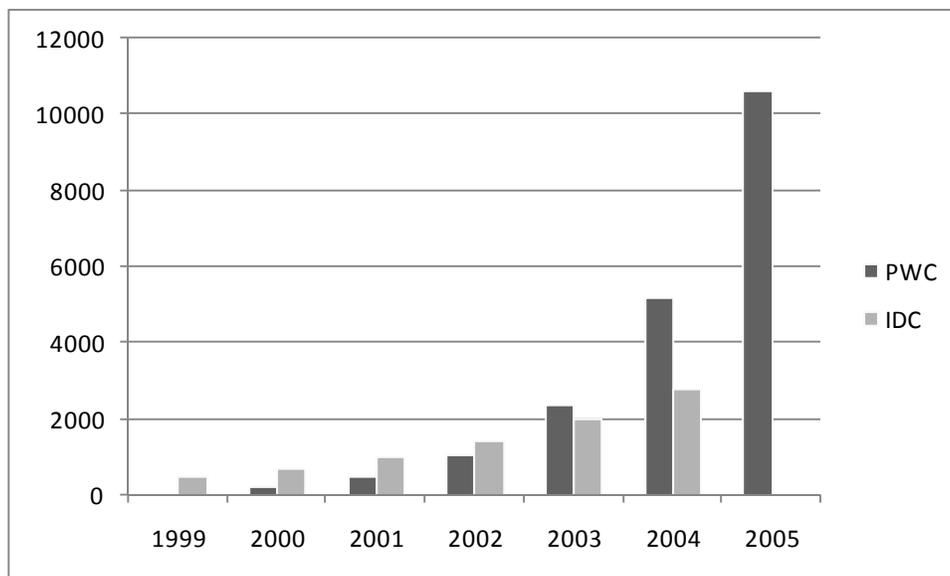


Gráfico 4.1 – Estimativas sobre o mercado mundial de componentes (1999 – 2005)

Fontes: ComponentSource (s/d).

Além das previsões da IDC e do PricewaterhouseCoopers, um estudo do GartnerGroup (ComponentSource (s/d), apontava que em 2003, 70% das novas aplicações seriam desenvolvidas com componentes. Não foram encontrados estudos que mostrem se isso foi comprovado, mas de acordo com as evidências disponíveis, o processo de adoção da tecnologia de componentes parece ter se difundido mais que o mercado de venda de componentes (Bauer, 1 jul. 2001).

Conjectura-se aqui que o mercado de compra e venda de componentes tem crescido mais devagar do que o inicialmente esperado devido à imaturidade técnica da tecnologia, à relativamente pequena oferta de componentes, ao alto investimento necessário para desenvolver com componentes e aos problemas ainda encontrados na apropriação dos retornos deste investimento.

Os principais intermediários de componentes (*ComponentSource* e *IntellectMarket*) começaram a operar em meados da década de 90, mas apesar de mais de 15 anos terem se passado, ainda há dificuldades para encontrar tanto fornecedores como compradores de componentes, pois falta escala em ambas as pontas e também pela dificuldade técnica de descrever e buscar um componente. A descrição de componentes, incluindo suas funcionalidades e restrições, não está plenamente estabelecida. Não há padronização neste sentido. Da mesma forma, a confiabilidade dos componentes via certificação ainda é um mercado incipiente (SOFTEX, 2007).

Estes dois aspectos (descrição e certificação dos componentes) são cruciais para o estabelecimento de ganhos crescentes de escala no mercado de componentes, o que é fundamental para que esta trajetória de reuso se consolide. Esta é a lógica da *commoditização*, com redução de custo e preço, em que o mercado de comercialização de componentes teria que se sustentar. SOFTEX (2007, p.18) apontam que no longo prazo, a produção de componentes poderia seguir duas trajetórias não excludentes:

[1] produção oligopolizada, global e coordenada, resultado da busca por produtividade e de ganhos de escala (assumindo que a padronização dos componentes será crescente e haverá queda de preços nos primórdios deste modelo de negócios); e

[2] produção atomizada, global e não coordenada, resultado da ação de desenvolvedores individuais, empresas de pequeno e médio portes ou de comunidades de prática que colocam os componentes em repositórios públicos ou privados.

O que tem sido constatado é a materialização de uma trajetória mista, com produção atomizada e descentralizada, iniciativas de desenvolvimento pré-competitivo e ainda a formação de uma terceira trajetória, qual seja, a aplicação da tecnologia de componentes por empresas desenvolvedoras de software (como atividade fim ou atividade meio) sem a aquisição ou compartilhamento de componentes externos, ou seja, com a produção *in-house* dos componentes necessários. Desta forma, as empresas estão praticando o reuso de componentes de software sem necessariamente utilizarem fontes externas de fornecimento de componentes. E assim, sem utilizar uma das possibilidades de redução de custos deste modelo.

Heuvel e Smits (2007) realizaram uma pesquisa sobre a estrutura do mercado de componentes de software e de serviços na web (*web services*) em 3 períodos: 1999, 2000 e 2006. Os autores constataram o pequeno crescimento do número de ofertantes (com exceção dos catálogos) e do volume de componentes (Tabela 4.1). A quantidade de componentes divulgados a partir de catálogos impressiona, um salto de 1.037 para 29.865 componentes entre 1999 e 2006, mas é preciso considerar que grande parte refere-se a componentes genéricos de baixa granularidade, o que de certa forma inflaciona a análise. Além disso, os autores constataram que vários problemas persistem como a dificuldade de localizar os componentes e a falta de documentação adequada, inclusive sobre a confiabilidade do componente.

Tabela 4.1 – Distribuição do número de sites e componentes segundo tipo de ofertante em 1999, 2000 e 2006

		Sites	Componentes
1999	Produtores	27	375
	Catálogos	6	1.037
	Intermediários	5	2.697
2000	Produtores	43	865
	Catálogos	7	1.591
	Intermediários	8	482
2006	Produtores	28	2.611
	Catálogos	28	29.865
	Intermediários	8	7.234

Fonte: Heuvel e Smits (2007)

Nota: Catálogos são a coleção de links de sites de vários produtores. Já os intermediários possuem catálogos, mas oferecem serviços complementares, como testes, garantias, suporte, entre outros.

Além disso, os autores perceberam que o mercado sofreu várias turbulências no período, com a ocorrência de fusões e aquisições e também de desaparecimento de empresas:

Desde 1999-2000, três intermediários, quatro catálogos e onze produtores desapareceram. O mercado tem sido turbulento: várias fusões (por exemplo, ComponentSource comprou o competidor Flashline), outros deixaram de existir, especialmente depois do estouro da bolha da Internet. A quantidade de componentes explodiu de 4 mil componentes oferecidos por 38 participantes do mercado em 1999 para 40 mil componentes e 64 participantes em

2006. De fato, este mercado ainda tem muita perspectiva de crescimento dado que novas tecnologias estão emergindo (AJAX) e outras tecnologias estão amadurecendo gradualmente (dotNet e J2EE) (Heuvel e Smits, 2007, s/p).

O *broker* mais conhecido é o ComponentSource, fundado em 1995. Em 2003 adquiriu seu principal competidor, Flashline. Atualmente, oferta 1722 componentes, sendo 1251 componentes e 541 ferramentas para desenvolvimento com componentes. Os componentes estão organizados em 236 categorias e são provenientes de 103 produtores (*publishers*). Os preços dos componentes variam de acordo com a granularidade do componente, a forma de licenciamento e a quantidade de máquinas ou aplicações que rodarão o componente, podendo ultrapassar 30 mil reais⁹⁸. No próximo item é apresentado um exemplo de licença de um componente comercializado no ComponentSource.

Heuvel e Smits (2007) chamam a atenção para a emergência do mercado de componentes de SL/CA, que deve ter impacto no desenvolvimento e uso de componentes. Em seu estudo sobre a utilização de componentes de SL/CA por empresas de software, Madanmohan e De' (2004) perceberam que as empresas levam em conta cinco questões críticas na hora de selecionar componentes de código aberto: custo, requerimentos de customização, características do componente (flexibilidade e consistência do design, estabilidade, confiabilidade e auditabilidade⁹⁹), manutenção e suporte, e licenciamento. A forma de licenciamento de um componente, seja ele de código aberto ou no formato proprietário, afeta sua reusabilidade, como será discutido na próxima seção.

Apesar do crescimento apresentado na tabela acima, o mercado de componentes ainda continua sendo pequeno, como apresentam Ravichandran e Rothenberger (2003, p.111):

[...] mercados de componentes ainda estão em sua infância e são constrangidos pela oferta. Uma implicação do mercado restrito pela oferta são os altos custos de aquisição, que inclui o custo de busca e o preço do componente. O custo da busca é alto devido à falta

⁹⁸ O ComponentSource permite o pagamento dos componentes em moeda nacional, no nosso caso, em Reais.

de capacidades unificadas de busca entre diferentes lugares e esquemas uniformes de especificação de componentes. O preço do componente é alto devido a ineficiências do mercado, levando a uma inabilidade para realizar as economias de escala esperadas da venda de um componente para potencialmente um grande número de usuários.

Além disso, os autores chamam a atenção para a terceira trajetória apontada acima, qual seja, o desenvolvimento e uso de componentes internamente na empresa. Eram esperadas economias de escala, que tornariam os componentes disponíveis no mercado mais baratos que aqueles desenvolvidos *in-house*. Entretanto, isto ainda não tem sido plenamente verificado devido à imaturidade do mercado de compra e venda de componentes, que não permite que as eficiências de escala ocorram. Desta forma, em muitas situações é mais barato desenvolver os componentes internamente do que recorrer ao mercado, uma clara questão de custos de transação.

Procurou-se aqui traçar o panorama da oferta de componentes, focando que a tecnologia ainda é considerada imatura, o mercado de componentes de software imaturo marcado por uma trajetória mista entre produção atomizada e produção internalizada nos integradores. O próximo item tratará do desenvolvimento de software com componentes e também as questões relativas à proteção da propriedade intelectual nesta cadeia de valor.

4.2 Proteção da Propriedade Intelectual e custos de transação no desenvolvimento com componentes

Do ponto de vista dos integradores, que montam soluções usando componentes, a adoção da tecnologia de reuso implica em um investimento considerável em treinamento, reorganização de seus processos e aquisição de ferramentas adequadas de desenvolvimento e gerenciamento dos componentes, como apresentam Sherif e Vinze (2003, p.160):

O custo de construir um programa de reuso é descrito como exorbitante sem retornos no curto prazo. Uma lista extensiva dos custos que uma organização enfrentaria antes de esperar retorno cobre as questões técnicas de desenvolvimento e gerenciamento dos ativos

⁹⁹ “Capacidade do software de verificar a integridade dos dados e de rastrear as atualizações significativas nos dados (quem fez, o que fez e quando fez)” (Bajerski *et al*, 1994).

reutilizáveis, bem como os custos de justificar e sustentar internamente o programa; dentre estes itens estão despesas com o desenvolvimento de uma infra-estrutura, que inclui ferramentas de suporte, treinamento do staff e re-estruturação da organização para adaptar às mudanças no desenvolvimento do software (Sherif e Vinze, 2003, p.160)

Morisio *et al* (2002) apontam um caminho semelhante ao dos autores acima. A partir do estudo que fizeram com 24 projetos de reuso realizados entre 1994 e 1997, os autores notaram que um terço deles fracassou não tanto por fatores técnicos, como implementação de repositórios internos, mas por fatores organizacionais, como não introdução de processos específicos de reuso, não modificação dos demais processos não relacionados com reuso e falta de compromisso da alta gerência com o programa de reuso. Justificar um investimento tão alto e conseguir manter o suporte interno a um programa de reuso é um desafio que os projetos de sucesso conseguiram resolver (Weiler, 2002).

Como já comentado, há várias formas de comercializar um componente de software, que se espelham nas formas de comercialização do software em geral. Basicamente, há quatro formas de ter acesso a um software: 1) desenvolver internamente, 2) encomendar a um fornecedor externo, 3) licenciar ou 4) assinar um serviço, no conceito de software como serviço (SaS) como provedor de serviço de aplicação (ASP) (Messerscmitt e Szyperski, 2001 apud Valimaki, 2003). As formas mais comuns são o desenvolvimento interno e o licenciamento de software desenvolvido externamente à empresa.

Diante disso, segundo Chavez *et al* (1998), uma licença de componente deve minimamente especificar os seguintes itens:

- direitos que o licenciante autoriza o licenciado a exercer no software (garantia da licença);
- pagamentos para o licenciante;
- quem tem a propriedade do componente licenciado bem como de suas modificações;
- os riscos e obrigações que cada parte assume;

- suporte, manutenção e garantias para o componente licenciado; e
- confidencialidade do componente licenciado.

Tratando do licenciamento, há várias formas de cobrar o pagamento de um componente: por cópia, por CPU ou por uso. A licença de um componente é mais complexa que a de um software “normal”, pois “as condições de propriedade nas licenças de componente de software devem tratar não apenas da propriedade do software licenciado, mas também devem contemplar a propriedade da solução final que incorpora o componente licenciado, bem como a propriedade das modificações feitas no componente” (Chavez *et al*, 1998, p.50). Desta forma, se o componente licenciado é *white-box* e se a licença permite modificação do código do componente, isto acrescenta mais um pouco de complexidade no licenciamento e na operacionalização do pagamento.

Algumas licenças exigem o monitoramento do uso do software por parte do integrador em soluções vendidas a usuários finais. Um exemplo é a licença que exige o pagamento de um dado percentual do faturamento com a venda da solução na qual foi integrado o componente que está sendo licenciado. Abaixo é apresentado um caso concreto de licença de componente disponível no *ComponentSource* para ilustrar a complexidade do processo¹⁰⁰.

O componente LEADTOOLS – Ferramenta de Documentação por Imagem¹⁰¹ – requer licença por tempo de execução (*runtime*), baseada na utilização do usuário final da aplicação que for desenvolvida com este componente. São oferecidos vários modelos de licenciamento para permitir flexibilidade para seus clientes e a possibilidade de desenvolver um programa de licenciamento que seja adequado ao modelo de negócio da empresa. O modelo básico de licenciamento requer a compra de uma licença por tempo de execução para cada solução desenvolvida, o que inclui cada cópia da aplicação instalada internamente no cliente e para cada cópia distribuída para os usuários finais, seja pelo cliente seja por distribuidores ou revendedores.

¹⁰⁰ Além do exemplo apresentado, no ComponentSource há vários outros tipos de licença: por desenvolvedor ou por endereço físico (no caso de ferramentas de desenvolvimento de software usando componentes), por servidor etc.

¹⁰¹ Componente da empresa *LEAD Technologies* disponível no Broker ComponentSource em <http://www.componentsource.com/products/leadtools-document-imaging-runtimes/index.html>

Se a solução for instalada numa rede ou servidor, é requerida uma licença para cada computador conectado. No Quadro 4.1 são apresentados os valores de cada tipo de licença.

Quadro 4.1 – Valor e detalhamento dos tipos de licença do componente *Leadtools Document Imaging v15.0*

Tipo de licença	Detalhamento	Valor em R\$***
Non Prepaid Stand Alone Runtime License**	Licença para todas as cópias da solução instaladas, usadas, distribuídas, fornecida sob leasing ou vendida para instalação em computadores (exceto servidores)	R\$ 196 para 1 licença
Non Prepaid Concurrent Runtime License	Idem com a diferença de a aplicação ser instalada em um servidor, sendo exigida a aquisição também de uma licença para cada computador que acessa o servidor	R\$ 393 para 1 licença
Prepaid Stand Alone Runtime License	Uma licença por usuário com a venda de pacotes – instalação feita no computador-cliente	R\$ 98 para cada licença de 50 a 99 licenças até R\$ 12 para cada licença acima de 2500 licenças*
Prepaid Concurrent Runtime License	Uma licença por computador que acessa a aplicação instalada no servidor com a venda de pacotes de licenças	R\$ 196 para cada licença de 50 a 99 licenças até R\$ 24 para cada licença acima de 2500 licenças*
Server Runtime License	Licença para instalação da aplicação em servidores, sendo que não é exigida a aquisição de licenças para os usuários que acessam o servidor	R\$ 9.821 para 1 licença a R\$ 34.372 para 5 licenças

Fonte: ComponentSource em 06/07/2007.

Nota: * Mínimo de 50 licenças; **Indicada para quem possui menos de 50 usuários; *** Valor inclui suporte técnico do ComponentSource e atualização gratuita se uma nova versão for lançada dentro de 30 dias da compra do componente.

O controle da quantidade de utilização do componente em aplicações é feito por meio de relatório a cada 4 meses. A *Lead* oferece ainda licença para quantidade ilimitadas, o que dispensa o envio dos relatórios. Assim, se o licenciado desejar remunerar corretamente o licenciante deverá manter um rígido controle sobre a utilização dos componentes em novas aplicações.

Além das formas de pagamento, a licença também apresenta as condições de propriedade, já que o licenciante tem direito de propriedade sobre o componente, seja através de copyright, patente e/ou segredo de negócio. A empresa *Lead* especifica na licença que é proprietária de todos os

direitos referentes a este componente de software, sendo que sua marca ou nome não podem ser utilizados pelo licenciado na aplicação final. A aplicação deve informar na funcionalidade “Sobre...” o seguinte texto: “Partes do produto foram criadas usando LEADTOOLS ©1991-2006, LEAD Technologies, Inc. TODOS OS DIREITOS RESERVADOS. Partes deste produto são licenciados sob a patente US 5.327.254 e contraparte estrangeira”.

O componente LEADTOOLS Document Imaging é protegido tanto por copyright como por patente. Entretanto, o licenciado deste componente não tem certeza se há violação de outra patente ou copyright ou se contém trechos de código licenciados sob a licença GPL. Para evitar ou minimizar este tipo de risco algumas estratégias foram desenvolvidas.

Algumas licenças prevêm indenização por parte do licenciante caso o licenciado venha a ser acusado de infringir o direito de propriedade intelectual de alguém (o que pode ser replicado no caso do desenvolvedor da solução com componentes em relação ao cliente final). Também há casos de empresas, como JBOSS, Hewlett-Packard, Red Hat e Novell, que têm oferecido indenizações no caso de ações legais que acusem seus clientes de soluções SL/CA de infringir direitos de propriedade intelectual protegidos via copyright ou patente.

Algumas empresas agregaram profissionais com perfil jurídico às equipes de desenvolvimento para estabelecerem políticas apropriadas e procedimentos sobre como incorporar componentes de terceiros (proprietários ou de código aberto) de forma a abranger acesso ao código fonte, modificação, implementação e licenciamento (Tolliver, 2005).

Também foi criado um mercado de ferramentas automáticas que detectam a presença de trechos de código aberto no código de um software ou componente. Uma destas ferramentas é o *IP Amplifier's Dectetor* da Palamida¹⁰², cuja assinatura anual custava, em 2005, entre US\$ 50 mil e US\$ 250 mil, dependendo do tamanho do ambiente de desenvolvimento do cliente. Outra empresa que possui este tipo de ferramenta é a Black Duck com seu produto denominado *transactIP*. Este produto identifica os componentes e suas respectivas licenças e as avalia segundo a política corporativa. O *transactIP* relata os problemas relativos a combinações de

¹⁰² <http://www.palamida.com/>

licenças incompatíveis que necessitam ser remediados pela equipe de desenvolvimento. O valor da assinatura do serviço do *transactIP* varia de acordo com o tamanho do software a ser inspecionado, indo de US\$ 475 para 5 MB a US\$ 7 mil durante 14 dias¹⁰³.

Outro software comercializado pela Black Duck é o *protexIP* que cria um ambiente de trabalho que ajuda as empresas a gerenciarem esta cada vez complexa selva de licenças, com a qual as empresas que desenvolvem com componentes se deparam. Outro software de gestão dos direitos de propriedade intelectual voltado para quem usa componentes é o *Component License Broker™*, como apresentado no site da empresa fabricante, *Systems Integrator Inc.* (2005):

Gerentes de redes corporativas agora têm um meio simples e efetivo de acompanhar e aplicar os acordos – e manter-se legal – em relação a todos os acordos de licença de software incluindo o uso de componentes COM/ActiveX. Component License Broker™ representa o estado da arte em ferramenta para gestão de ativos de software que ativamente monitora e relata o uso de arquivos DLL / OCS em tempo real, assegurando que sua organização está de acordo com os termos das licenças dos componentes adquiridos (desta forma evitando litígio) e que não está pagando por licenças não usadas e geralmente caras. Component License Broker™ quando usado em conjunto com License Broker™ assegura que as organizações possuem controle sobre todos os ativos de software sem as complicações técnicas, overhead administrativo ou sobrecarga financeira.

Não há informações sobre a difusão destas ferramentas, e aparentemente muitas empresas não estão muito preocupadas se estão violando direitos de propriedade intelectual de outrem.

Os projetos de código aberto são uma grande nova fonte de código reutilizável. São componentes *white-box*, pois permitem alteração do código fonte. Embora não exijam pagamento de licença a seus desenvolvedores, as licenças de SL/CA podem colocar outros tipos de restrições. Como já exposto, uma das licenças consideradas mais restritas é a GPL, pois exige que o software resultante deve seguir as mesmas regras da parte inserida, ou seja, continuar sendo regido pela GPL. Isto é chamado de caráter viral da licença GPL. Em relação à modificação no código,

¹⁰³ <https://transactip.blackducksoftware.com/gui4/ondemand/ondemandPurchaseOptions.jsp> acesso em 07/07/2007.

licenças de código fechado restringem totalmente a alteração, enquanto licenças tais como GPL restringem a liberdade dos desenvolvedores especialmente se o código é modificado e se torna trabalho derivativo (Madanmohan e De', 2004).

Mas as empresas já desenvolveram estratégias para lidar com as restrições de algumas licenças, como as da GPL. Uma primeira estratégia foi a identificada por Madanmohan e De' (2004). Os autores relataram o caso de empresas que adotaram a estratégia de desenvolver módulos contendo componentes baseados na licença GPL com APIs (*Application Programming Interface*), expondo estes módulos de forma a evitar a natureza viral da GPL, ou seja, a imposição da GPL de que uma vez introduzido o código licenciado sob a GPL, todo o programa passe a ser regido por esta licença.

Uma segunda estratégia identificada pelos autores em seu estudo sobre o reuso de código aberto por empresas comerciais, foi que na hora de coletar componentes, as empresas investigadas (grandes empresas como Sun, HP, IBM, Motorola India, Philips India, Sancharnet, Wipro e Infosys) preferiam componentes baseados em licenças do tipo BSD e aqueles em que era possível localizar o autor. Apesar das licenças, as empresas procuravam o autor para pedir autorização de uso do componente.

Ainda no estudo de Madanmohan e De' (2004), outra estratégia encontrada foi o fato das empresas estudadas terem que criar rotinas de análise das questões de PI para usar componentes SL/CA. A Motorola criou um processo denominado *Strategic Technology Asset Management Process*. No caso da Philips era denominado *Intellectual Property and Standardization Review*. O código em si não era o principal atributo para escolher um componente, mas se havia uma comunidade grande de desenvolvedores para prover assistência, o que servia como uma garantia de assistência, o que novamente remete à importância dos ativos complementares, como apontado por Teece (1986) no capítulo 1.

Além da complexidade da proteção do software, como colocado, a gestão dos direitos de propriedade no desenvolvimento e uso de componentes é extremamente complexa e depende da posição na cadeia de valor do ator. Voltando à Figura 4.2, um resumo das implicações para cada

ator segue abaixo:

- **Desenvolvedor de componentes:** para quem produz componentes, há a dificuldade de rastreabilidade do mesmo, para cumprimento de contrato que exija o pagamento por parte do integrador relativo a cada utilização em uma nova solução montada. É difícil localizar em um software, entregue sem o seu código-fonte, uma parte específica deste programa, no caso, um componente. Com o código-fonte é mais fácil, mas ainda assim, em softwares de grande porte, a dificuldade é grande. Assim, o cliente, quando adquire a licença de uso de um software, tem dificuldades em verificar se aquele software utiliza componentes que não poderia utilizar, representando um risco. O uso das ferramentas automatizadas facilita esta tarefa, mas representa um custo adicional.
- **Integrador:** para quem os usa para criar soluções, há complicações quanto à responsabilidade jurídica dos componentes, contaminação e pagamento de licenças, entre outros fatores. Estas situações ficam mais complicadas no caso do software, em comparações com outras áreas tecnológicas, devido à recente formação do “cipoal de patentes”¹⁰⁴. As empresas podem usar ferramentas automatizadas para inspecionar um componente e localizar problemas, e também para gerenciar e evitar o uso de licenças concorrentes, com o ônus do custo adicional destas ferramentas. Em alguns casos, as empresas têm preferido reescrever componentes licenciados sob a GPL (von Krogh *et al*, 2005).
- **Cliente/usuário:** para quem é o usuário final, também há o risco de infringir direitos de propriedade intelectual, inclusive sem conhecimento. Para evitar este tipo de risco, alguns usuários negam-se a usar componentes e alguns integradores oferecem seguros contra possíveis infrações.
- **Certificador:** para quem é certificador, o clima de incerteza relacionado com os direitos de propriedade intelectual tem criado um novo mercado, que poderíamos chamar de

¹⁰⁴ Como já apontado, quando o inovador tem que negociar com muitos detentores de patente, os custos de transação são excessivos (Bessen, 2003) e o licenciamento cruzado pode diminui-los (Grindley e Teece, 1997).

“certificadores de propriedade intelectual”, que desenvolveram serviços para gestão de direitos de propriedade intelectual e também de exame das soluções para checar se elas não ferem direitos de propriedade intelectual de outrem, como a empresa Palamida.

Resumindo o capítulo 4, a tecnologia de componentes propõe conciliar as vantagens de duas formas de desenvolver software: sob encomenda e software-produto. A primeira forma oferece a vantagem de grande customização aos desejos do cliente, mas por outro lado há custo e risco elevados, pois se trata de um software relativamente pouco testado. Já o software-produto possui pouca flexibilidade de customização, mas menor probabilidade de problemas e erros e menor custo, dado os possíveis ganhos de escala. A partir da posição do integrador, que desenvolve usando componentes, esta tecnologia oferece as seguintes vantagens:

- redução do tempo de produção de uma solução para chegar ao mercado (*time-to-market*);
- aumento da produtividade do programador;
- sistemas flexíveis que permitem adaptações e mudanças em processos de negócio, pois se pode trocar um componente sem grandes alterações no resto do sistema;
- sistemas confiáveis, seguros e escaláveis¹⁰⁵;
- facilidade de manutenção.

A despeito destas vantagens, há ainda uma série de dúvidas quanto à tecnologia de componentes, a maior parte delas relativa a sua maturidade. Em 2004, Rossi apontou uma série de problemas que ainda estavam, e continuam, em aberto:

- ausência de consenso sobre abordagem de desenvolvimento mais apropriada para desenvolvimento de sistemas baseados em componentes. O surgimento de um padrão dominante geraria uma maior certeza em relação a investimentos e poderia fazer com que

¹⁰⁵ Escalabilidade refere-se à capacidade de ampliação de escala do sistema (SOFTEX, 2007).

a tecnologia se sedimentasse.

- ausência de padrões em relação às tecnologias de plataforma e infra-estrutura para garantir interoperabilidade entre componentes de diferentes plataformas ou ambientes;
- mercado relativamente pequeno de componentes e sem garantia de confiabilidade;
- ausência de abordagem padrão de documentação e catalogação de subsistemas, componentes e outros bens reutilizáveis.

O que se pode apreender da análise tecida neste capítulo é que os direitos de propriedade intelectual têm adicionado um custo para as empresas que produzem software com componentes, seja em relação aos seus próprios direitos, seja em relação aos direitos de terceiros. A empresa desenvolvedora inovadora que deseja proteger sua criação, via patente, deve arcar com os custos do processo de patenteamento e com possíveis litígios. Por outro lado, essa mesma empresa tem que ficar atenta para não infringir patentes, copyright ou termos de licença de outrem. Desta forma, o desenvolvimento e disseminação da tecnologia de componentes têm se deparado com aumento de custos, principalmente de transação.

Como visto, muitas vezes esta vigilância implica na ampliação das equipes, com a contratação de advogados e outros profissionais especialistas em propriedade intelectual, e na aquisição de ferramentas automatizadas para ajudar a manter o software “limpo”, ou seja, de forma tal que não infrinja patentes, que não tenha componentes com licenças incompatíveis entre si e que remunere corretamente os fornecedores de componentes. E também pode implicar em um novo possível custo de seguro para o usuário final contra a violação de direitos de propriedade intelectual.

Como consequência, a decisão da empresa de desenvolver internamente um componente dependerá de uma série de fatores no clássico *trade-off* entre fazer internamente ou comprar fora (*make or buy*). Em teoria, só se vai ao mercado para comprar um componente se os custos de transação são baixos, ou seja, o custo de licenciamento e aprendizado for menor que o de construir intermanente. Além disso, a ida ao mercado também visa buscar suporte, manutenção e garantia. Entretanto, a complexidade dos DPIs e das formas de licenciamento dos componentes

têm tido um peso grande decisão de *make or buy*.

Estas considerações levam à conclusão que ao lado das questões técnicas, organizacionais e econômicas, o regime de apropriabilidade é um elemento decisivo na constituição do mercado de componentes. Uma pré-condição para isto ocorrer é que tanto os licenciantes quanto os licenciados de componentes de software sejam adequadamente compensados por seus esforços (Chávez *et al*, 1998). Assim, se não se constituir um regime minimamente estruturado que garanta remuneração destes esforços, este mercado deve seguir fragmentado e pautado no desenvolvimento *in house*, atomizado ou por comunidades de prática e arranjos similares (SOFTEX, 2007).

5. CONSIDERAÇÕES FINAIS

Esta tese partiu da consideração do aumento da importância do conhecimento na geração de riqueza, à medida que a economia tem se desmaterializado, e do aumento da importância da propriedade intelectual na apropriação dos retornos do esforço inovativo para discutir as tendências de proteção e de apropriação na indústria de software. O software representa um dos ícones da Economia do Conhecimento, tanto devido à sua participação direta na composição da riqueza nas nações como por sua pervasividade e transversalidade que dinamiza as outras indústrias (Roselino, 2006).

O crescimento da importância da propriedade intelectual e dos direitos relacionados tem estabelecido um mundo pró-patente (Tang e Paré, 2003), com a valorização maior de algumas das funções da PI em detrimento de outras. Conforme colocado no capítulo 1, pode-se falar que a PI possui quatro funções básicas: a) mecanismo de apropriação privada via exclusividade de exploração de uma invenção ou criação; b) elemento de transação econômica; c) uso estratégico e) função prospectiva e de divulgação. No mundo pró-patente, as funções b e c têm se destacado principalmente pela corrida patentária que se estabeleceu com o fortalecimento dos direitos de PI, no qual as patentes são usadas não tanto para reforçar a exclusividade de usufruto do conhecimento, mas mais para garantir mercados e liberdade de operação.

Isto tem sido motivo de preocupações porque, como numa corrida de Fórmula 1, as equipes melhor equipadas conseguem assumir a dianteira mais facilmente que as menos equipadas. Na corrida patentária, tanto internamente à indústria de software como externamente, as empresas estabelecidas têm se mantido na dianteira, potencialmente aumentando as barreiras para novos entrantes, que sempre foram uma importante fonte de inovação na indústria de software. A corrida patentária seria potencialmente danosa devido à natureza da indústria de software, marcada por produtos complexos e por inovação seqüencial e cumulativa, e devido ao aumento dos custos de transação.

O quadro de preocupações se agrava mais ainda se se volta a atenção para o processo de exame

das patentes de software. Como o software tornou-se uma matéria patenteável há relativamente pouco tempo, foram concedidas, principalmente nos EUA, muitas patentes questionáveis, no sentido que não cumpriam os requisitos mínimos. Isso deveu-se à falta de boas bibliotecas para a busca do estado da arte, pelo desconhecimento dos analistas na área de software e também ao grande volume de pedidos num curto espaço de tempo.

A patente esquentou uma polêmica que sempre esteve presente na indústria de software, qual seja, a forma mais adequada de se garantir a propriedade intelectual. O copyright foi adotado sob a regra da dúvida inicialmente nos EUA ainda nos anos 60. Talvez o formato mais adequado tivesse sido um tipo de proteção *sui generis* inteiramente personalizado para o software e suas características tecnológicas. Tal formato chegou a ser sugerido pela Organização Mundial de Propriedade Intelectual (OMPI) em 1971, mas a idéia não foi levada adiante. Nos anos 70 e 80, sob pressão das empresas de software-produto norte-americanas, o direito autoral foi expandido para abrigar o software como objeto de proteção (Santos, 2003).

Entretanto, a proteção do software no âmbito do Direito de Autor conseguiu conseqüências opostas: introduziu um objeto estranho, dado seu caráter funcional, neste estatuto e, por outro lado, teve o efeito de prover uma proteção insuficiente, dados os objetivos dos produtores de software. Ao mesmo tempo em que direito autoral se constituiu como a forma adequada e aceita de proteção em grande parte dos países, a patente também passou a ser usada, principalmente por parte das empresas que produziam invenções relacionadas com software, mas cujo negócio estava mais focado no hardware. Quando da criação do Acordo sobre Aspectos de Propriedade Intelectual Relacionados com o Comércio (TRIPs), de certa forma, consolidou-se a coexistência dos instrumentos legais. Nos anos 90, houve a expansão da proteção via patentes, incluindo métodos de fazer negócio. Desta forma, as patentes de software primeiro foram colocadas em dúvida, depois admitidas e finalmente amplamente aceitas (Cohen e Lemley, 2001).

O quadro atual é de superposição e complementaridade de campos de proteção. As empresas e outros atores têm à sua disposição os seguintes instrumentos jurídicos:

- o código do software é protegido sob a lei de direito de autor como um trabalho de autoria

original. No caso brasileiro, é possível fazer o registro do software junto ao Instituto Nacional de Propriedade Industrial - INPI (lei 9609/1998), mas este não é necessário para garantir a proteção¹⁰⁶. Este tipo de proteção permite a seu proprietário o controle da reprodução, adaptação, distribuição e exposição pública do código do software. *Copyright* pode ser usado para evitar que outros copiem o software, através da duplicação direta ou de seus elementos expressivos, mas não pode ser usado para evitar que outros façam uso dos aspectos funcionais do software;

- patentes podem ser usadas para proteger os processos implementados usando software, como equipamentos com software embarcado e outros sistemas. Em alguns países, como os EUA e Japão, também podem ser usadas para proteger o programa em código objeto. Segundo Santos (2003), no Brasil sempre houve a distinção entre “invenções de software”, criações que incluem programas de computador como uma de suas etapas, e os programas de computador em si, que não são enquadrados como invenção¹⁰⁷. Na Europa, há dispositivo semelhante;
- além disso, certos aspectos do software podem ser protegidos por meio do uso de segredo de negócio, acordos contratuais de licenciamento, marcas e desenho industrial (Santos, 2003).

Procurou-se neste trabalho matizar um pouco melhor o debate, muito centrado nos instrumentos de produção, para analisar as formas de proteção a propriedade no contexto dos regimes tecnológicos da indústria e de seus vários modelos de negócio. Como trabalhado durante a tese, entre o que o quadro legal especifica e sua execução há um espaço de manobra para as empresas montarem suas estratégias de apropriação de seus esforços.

No jogo da apropriação, as empresas fazem uso de um mix de instrumentos, dado que os

¹⁰⁶ O registro facultativo é uma forma de assegurar a titularidade do software e comprovar a autoria em caso de litígio. A proteção autoral é iniciada com a colocação da obra em um suporte tangível ou intangível, independente de registro (Mendes, 2006). Segundo Santos, o Brasil é um dos poucos países que implementaram legislação específica, derogatória do direito de autor comum.

¹⁰⁷ Art.10, V da Lei nº 9.279/1996.

mecanismos jurídicos não são auto-suficientes para proteção e têm que ser complementados. Muitas vezes, inclusive, têm um papel secundário na apropriação. Procurou-se, então, traçar sistematizar as principais características de um conjunto de modelos de negócio, bem como sistematizar os principais instrumentos de apropriação utilizados em cada modelo de negócios.

A história da indústria de software foi dividida em três regimes tecnológicos – condições de oportunidade, apropriabilidade, cumulatividade e base do conhecimento – no seio dos quais foram estabelecidos os modelos de negócio:

- O primeiro regime foi centrado nos grandes computadores (*mainframes*), tendo o software e os serviços um papel de suporte;
- O segundo regime foi (e ainda é em grande parte) centrado no computador pessoal (PC), com o sistema operacional padronizado, sendo o hardware um suporte para o software;
- O terceiro regime, em fase de transição, é marcado por serviços disponíveis na Internet, sendo que o software e o hardware não “importam muito”, servindo a web como plataforma para desenvolvimento e oferecimento de serviços.

Os modelos de negócio são resumidos no Quadro 5.1.

Quadro 5.1 – Principais características e modelos de negócio dos regimes tecnológicos da Indústria de Software

Regime tecnológico	Núcleo	Suporte	Modelos de negócio
<i>Mainframes</i>	Hardware	Software e serviços	Produtos Corporativos Serviços de alto valor Serviços de baixo valor
PC	Software	Hardware e serviços	Produto Pacote Software embarcado Produto customizável Serviços de alto valor Serviços de baixo valor
Serviços na Web	serviços	Hardware e software	Produto Pacote Software embarcado Produto customizável Serviços tipo Google ASP Serviços de alto valor Serviços de baixo valor

Fonte: Elaboração própria.

Os modelos de produtos de software fazem uso mais acentuado dos DPIs. Entretanto a indústria de software está cada vez mais migrando para serviços, o que aparentemente indicaria movimentos contrários. O significado disso ainda não está claro, pois se trata de um fenômeno muito recente, e o que se pode fazer aqui são algumas conjecturas para aprofundamento em estudos posteriores.

A corrida patentária endógena à indústria de software pode tratar-se de patenteamento estratégico por parte das empresas que atuam nos modelos de negócio de produtos. Seria um patenteamento defensivo entre grandes players, que também têm criado meios alternativos de lidar com isso, via acordos de não agressão mútua e alianças de desenvolvimento pré-competitivo, como SL/CA mostrou, e também novos modelos de negócio. No caso de produtos complexos, os DPIs não são usados tanto para fins de exclusividade, mas mais para garantir liberdade de interação e inovação. Ou seja, estas estratégias de não agressão mútua visam diminuir os custos de transação, de forma semelhante ao que ocorreu nos primórdios das indústrias de semicondutores e computadores.

A indústria de software é altamente inovadora e competitiva, mesmo nos mercados nos quais as empresas detêm posições monopolistas obtidas pelas externalidades de redes de seus produtos. Um exemplo é a Microsoft que a despeito da condição monopolista possui um comportamento altamente inovador o que é uma das formas mais eficientes de proteção da propriedade intelectual.

Se no caso de produto pacote a inovação rápida continua sendo um importante instrumento de apropriação, pode supor-se que a corrida patentária não deverá diminuir o ritmo da inovação na indústria de software, que, como dito, utiliza um mix de mecanismos de proteção formais e informais dependendo do modelo de negócio.

Entretanto, no caso do desenvolvimento e uso de componentes de software, DPIs fortalecidos podem estar dificultando esta trajetória tecnológica devido ao aumento dos custos de transação, por exemplo, em relação ao cuidado que a empresa que busca o reuso de software deve ter para não infringir patentes, copyright ou termos de licença de outrem. Esta vigilância pode implicar na ampliação das equipes, com a contratação de advogados e outros profissionais especialistas em propriedade intelectual, e na aquisição de ferramentas automatizadas para ajudar a manter seu software “limpo”, além de um novo possível custo de seguro para o usuário final contra a violação de direitos de propriedade.

O uso estratégico de patentes não é bom nem ruim em si, dado que pode ter vários efeitos, sendo usado tanto para articulação e cooperação entre agentes econômicos, quanto para reforçar barreiras à entrada, levando ao aumento dos custos de transação, entre outros. De forma semelhante, enquanto há quase um consenso que as patentes de software representam uma ameaça ao SL/CA, por estarem criando um terreno minado para a circulação dos programadores, os acordos realizados recentemente entre grandes empresas de software proprietário e empresas de SL/CA mostram que pode haver um outro lado, no sentido da cooperação, e que estas últimas nem sempre devem ser vistas como cordeiros em terras de lobos.

Outra peculiaridade da indústria de software é o papel importante dos usuários no desenvolvimento da indústria. Como visto, desde o início os usuários motivados inicialmente

pela resolução de seus próprios problemas colocaram a “mão na massa” e desenvolveram aplicações. Isto ganhou uma dimensão bem maior com o advento do SL/CA.

Na verdade, o desenvolvimento do SL/CA mexeu com todos os atores da indústria de software de alguma forma, bem como com os modelos de negócio. O SL/CA e o software proprietário devem ser pensados não como contrários, mas como pontos de uma escala contínua em torno da tensão que existe entre adoção e apropriabilidade dos benefícios econômicos de um padrão. Trata-se agora de cada ator construir sua estratégia aberta o suficiente para atrair compradores, enquanto conserva os retornos adequados. O SL/CA e o software proprietário não são antagônicos, mas devem coexistir e se mesclar de acordo com diferentes estratégias.

O estudo das estratégias recentes das empresas mostra o uso complementar dos instrumentos de proteção da propriedade intelectual, aliado aos ativos complementares. Por exemplo, a estratégia da IBM, abriu mão de trabalhar apenas suas tecnologias proprietárias, e tentar fazer o *lock-in* através da tecnologia e de seus DPIS, para tentar fazer o *lock-in* com serviços considerados insubstituíveis. Esta é uma parte importante da estratégia da IBM para proteger seus ativos, retomando a lógica de solução total dos seus primórdios (antes da separação do hardware e software), quando o item mais valorizado era o hardware, que era protegido via patentes, e os demais itens eram “brindes”. Agora, continua a lógica da solução total centrada em serviços balizados com a marca IBM, mas o cliente pode escolher o “brinde”. Esta estratégia que mostra que os DPIS têm papel secundário na apropriação dos resultados dos esforços inovativos. Entretanto, a IBM está longe de deixá-los de lado, visto que é a empresa que mais patenteia nos EUA a mais de dez anos. Estratégia semelhante pode ser captada nos outros gigantes do setor de software, como Microsoft.

Desta forma, não há uma receita única para as empresas de software quanto aos DPIS. No caso das “grandes equipes de F1” parece imprescindível correr atrás de todos os direitos, mesmo que não se vá usá-los. No caso das pequenas empresas, isto parece um custo desnecessário e a proteção deve passar por outros elementos.

Da mesma forma, não é possível pensar uma política pública única. Por exemplo, o pacote de

licenças pode ter um lado muito ruim, mas além de ser possível trabalhar com isso, pode-se pensar num exemplo semelhante ao que foi a RCA para algumas tecnologias básicas do setor de software.

De qualquer forma, o estudo os impactos do fortalecimento dos DPIs na indústria de software deve ser aprofundado. À primeira vista parece estranho discutir se patentes em maior número ou mais fortes influenciam o ritmo da inovação, pois a patente inclusive tem sido utilizada nos estudos de indicadores de CT&I como um output (final ou intermediário) do processo de inovação. Mas isto tem sido questionado, especialmente com a observação da corrida patentária.

A tese discute tendências internacionais, mas é possível fazer algumas reflexões sobre o caso brasileiro, à guisa de uma agenda de pesquisa. Como mencionado acima, no Brasil o software é protegido via direito autoral, sendo facultativo o registro de software. Mas pelo menos desde a década de 80 o INPI tem concedido patentes para invenções implementadas por software, nas quais o software altere tecnicamente o funcionamento da invenção, como exposto no Manual de Exame da Diretoria de Patentes (DIRPA) (Abrantes, 2006).

Antônio Abrantes realizou uma pesquisa com 590 pedidos relacionados com software examinados na Divisão de Patentes de Física, Eletrônica e Eletricidade (DIFELE/DIRPA/INPI). Deste total, 74% dos pedidos foram concedidos e 77% já possuíam patentes concedidas nos EUA. Chama a atenção para o grande número de pedidos de não residentes (87%), sendo que 83% deles foram deferidos. Já o número de pedidos de residentes é bem menor e a taxa de deferimento também (17%). Havia poucos pedidos de empresas nacionais.

Segundo Andrade *et al* (2007), a maior liberalidade dos escritórios de patentes dos EUA e da Europa tem levado o INPI a rever sua política de concessão de patentes para software e há um grupo de trabalho específico para isso. Esta é uma iniciativa louvável, à qual deve-se juntar estudos sobre as formas de apropriação utilizadas pelas empresas para chegar no formato mais adequado para o desenvolvimento econômico do setor, dado que grande parte das empresas de software é formada por micro e pequenas empresas. Segundo estudo realizado sobre o uso que as pequenas e médias empresas de todos os setores fazem do sistema de propriedade intelectual no

Brasil, estas empresas acessam muito pouco o sistema de propriedade industrial devido à falta de conhecimento e de capacitação (Salles-Filho *et al*, 2006).

Estudos internacionais confirmam esta tendência. O estudo feito por Tang e Paré (2003) sobre a gestão que as pequenas e médias empresas fazem da proteção da propriedade intelectual corrobora a importância de aprofundar o entendimento desta questão. Na visão destas empresas, as patentes não lhe são muito úteis, pois acham outros instrumentos de proteção mais efetivos, como copyright, sistemas técnicos de proteção (*encryption*), atuação em nichos de mercados, inovação rápida (mais que o concorrente para gozar das vantagens de *first mover*) e segredo de negócio.

Desta forma, encerro esta etapa com uma agenda de pesquisa apontando para novas trilhas que podem ser seguidas para o entendimento sobre as tendências de proteção de ativos na indústria de software no Brasil.

ANEXO 1 – INICIATIVAS PARA LIDAR COM A CORRIDA PATENTÁRIA

A sociedade tem se preocupado com a corrida patentária, e alguns segmentos têm se organizado para questionar e tentar diminuir o ritmo de concessão de patentes, especialmente daquelas consideradas de baixa qualidade. A seguir são listadas algumas destas iniciativas, agrupadas em três conjuntos, segundo o tipo de ator que as deflagaram:

- iniciativas do movimento de software livre, de associações sem fins lucrativos e pequenas empresas;
- iniciativas de grandes empresas;
- iniciativas de universidades.

Veja-se, brevemente, o ponto de vista de cada um dos conjuntos de iniciativas.

Iniciativas do movimento de software livre, de associações sem fins lucrativos e pequenas empresas

As iniciativas envolvem ações de discussão e divulgação, mas também de participação ativa em fóruns oficiais, como discussões promovidas por escritórios nacionais de propriedade intelectual, depoimentos em eventos de órgãos do judiciário, entre outros. Uma das campanhas de mobilização de usuários e desenvolvedores contra mudanças na regulação dos direitos de PI que ficou mais destacada foi a Campanha da Eurolinux Alliance e do site Nosoftwarepatents.com contra a proposta de Diretiva Européia sobre Patenteabilidade das invenções implementadas por programas de computador. Após a realização de um estudo sobre o impacto econômico das patentes de software (Hart *et al*, 2000), a União Européia disponibilizou no final de 2000 uma consulta pública a uma proposta de Diretiva sobre a patenteabilidade das invenções relacionadas com software com vistas a harmonizar a legislação nos países membros. A consulta tratava do escopo da harmonização (se deveria ser adotada uma abordagem mais restritiva ou mais liberal como nos EUA) e também sobre o possível impacto da harmonização (na inovação, sobre as PMEs, sobre o SL/CA e na posição da indústria européia de software na competição global). O

objetivo da consulta era receber opiniões e subsídios que ajudassem a encontrar “o balanço correto entre a promoção da inovação com a possibilidade de conseguir patentes e assegurar a competição adequada no mercado” (PbT Consultants, 2001).

Em agosto de 2001 foi publicado o resultado da consulta. Foram recebidas 1.447 respostas, sendo quase 1.200 ligadas a uma petição organizada pela Eurolinux Alliance, uma associação de mais de 200 empresas de software e associações com o objetivo de promover e proteger o uso de padrões aberto, competição aberta e software de código aberto¹⁰⁸. As respostas foram classificadas em duas proposições distintas conforme apresentado no quadro abaixo.

Quadro A1.1 - Proposições sobre a Diretiva Européia sobre a patenteabilidade das invenções implementadas por computador

Tipo de Proposição	Proposição Restritiva – oposta à maioria das patentes de software	Proposição Liberal – aplicar critério tradicionais de patenteabilidade às invenções implementadas por programa de computador
Membros	Estudantes, acadêmicos, engenheiros e empresas start-ups	Advogados, players estabelecidos da indústria e agências governamentais
Preocupações	<ul style="list-style-type: none"> • ameaças para o movimento SL/CA e para as pequenas e médias empresas, • falta de recursos e conhecimento para patentear, • medo de litígio e impactos negativos nos padrões de interoperabilidade 	<ul style="list-style-type: none"> • proteção ao investimento do desenvolvimento, • igualdade de condições com os EUA • abertura dos mercados globais
Propostas	<ul style="list-style-type: none"> • restringir severamente a patenteabilidade do software • limitar a responsabilidade do software de código aberto por infração • rejeitar todas as patentes de métodos de negócios 	<ul style="list-style-type: none"> • harmonizar a aplicação da prática do Escritório Europeu de Patentes (EPO) • aplicar os critérios de patenteabilidade ao software ligeiramente mais liberais que os propostos no Paper de Consulta da Comissão da Diretiva • tomar extremo cuidado com o patenteamento de métodos de negócios

Fonte: PbT Consultants (2001, p.3-4).

¹⁰⁸ <http://www.eurolinux.org/>

Numericamente, os respondentes da posição restritiva dominaram as respostas dadas (91%), em função da petição organizada pela Eurolinux Alliance. Mas se se toma em conta a importância econômica, haveria uma “maioria econômica” favorável a estas patentes:

Se a conta é feita a partir da musculatura econômica e do número de organizações representadas por respostas da indústria e outras associações, pode-se argumentar que há uma maioria econômica em favor das patentes para invenções implementadas por computador. Por outro lado, aqueles opostos às patentes de software podem requerer que devido ao tamanho e fragilidade das organizações, elas necessitam de suporte. Eles também podem requerer que é apenas o movimento de “código aberto”, por exemplo Linux, que pode efetivamente tirar a “Micro\$oft” deste mundo. Ultimamente, o peso dos dois pontos de vista é uma questão política (PbT Consultants, 2001, p. 4).

Esta colocação do relatório de análise das respostas dadas à consulta sobre a “maioria econômica” fez surgir uma campanha contrária ao patenteamento deste tipo de invenções. A campanha denominada “Economic Majority Against Software Patents”¹⁰⁹, filiada à Foundation for a Free Information Infrastructure (FFII)¹¹⁰, que reúne quase duas mil empresas que empregam cerca de 31 mil pessoas e com faturamento anual de cerca de 3,3 bilhões de euros¹¹¹.

Outra campanha contrária ao patenteamento é o site Nosoftwarepatents.com¹¹², financiado por 5 empresas (RedHat, 1&1, GMX, Schlund Partner e MySQL). O site à época da discussão da Diretiva tinha material, em 17 línguas, sobre a situação atual, os perigos, as não-verdades e as políticas ligadas ao patenteamento de software.

A proposta de Diretiva foi apresentada oficialmente em 2002 e rejeitada em 2005 pelo

¹⁰⁹ <http://www.economic-majority.com/>

¹¹⁰ A FFII é uma associação sem fins lucrativos registrada em 20 países europeus, dedicada ao desenvolvimento dos bens de informação para o benefício público, com base no copyright, competição livre e padrões abertos. Possui mais de 850 membros, 3.500 empresas e 100 mil apoiadores. É um dos grupos mais ativos contra o patenteamento de software porque acreditam que as patentes de software são um desastre em termos de inovação, competição e equilíbrio dos direitos. Também são contrários ao patenteamento de métodos de negócios que, na sua opinião, servem para restringir o comércio. Fonte: <http://eupat.ffii.org/group/>

¹¹¹ Valores referentes ao dia 01/05/2007.

¹¹² <http://www.nosoftwarepatents.com/>

Parlamento Europeu. A notícia da rejeição foi comemorada pelo movimento SL/CA do mundo todo.

Outra iniciativa é através da solicitação de revisão de patentes consideradas inválidas (por não preencherem os requisitos mínimos). Para isso é necessária a reunião de materiais que comprovam anterioridade de patentes consideradas ruins para solicitar a revisão das mesmas, como faz a Public Patent Foundation (PUBPAT), organização sem fins lucrativos fundada em 2003 por Dan Ravicher. A PUBPAT atua em três frentes: representação, educação e advocacia. Quando encontram uma patente que consideram inválida, buscam documentar *prior art* e solicitam que o USPTO revise esta patente¹¹³.

Ravicher tem sido chamado de Robin Hood dos não detentores de patentes do mundo (Kintisch, 2005). Em 2005 deu um testemunho no congresso norte-americano (*Subcommittee on Courts, the Internet, and Intellectual Property*)¹¹⁴ na audiência sobre a reforma do sistema de patentes.. Em fevereiro de 2007, participou de outra audiência do mesmo subcomitê denominado *American Innovation at Risk: The Case for Patent Reform*. Na ocasião argumentou sobre a necessidade de limitar o escopo das patentes.

Outra instituição com atuação semelhante à PubPat é a Electronic Frontier Foundation (EFF)¹¹⁵, organização sem fins lucrativos, que trabalha no interesse público para defender as liberdades civis e a livre expressão no mundo digital. Atua por meio de advogados, técnicos e ativistas para defender a liberdade da expressão, a aplicação apropriada da propriedade intelectual, a privacidade e a inovação tecnológica. Em seu site há uma solicitação de ajuda para que voluntários auxiliem a EFF a reunir material sobre o estado da arte que comprove a anterioridade das invenções de algumas patentes consideradas inválidas. A imagem da campanha lembra os cartazes de procurados pela polícia (figura A1.1).

¹¹³ Já conseguiram invalidar pelo menos três patentes: Forgent JPEG Related Patent, Pfizer Lipitor Patent e Microsoft FAT Patent. Mais informações em <http://www.pubpat.org/Protecting.htm>

¹¹⁴ "Patent Act of 2005", Legislative Hearing, June 9, 2005, U.S. House of Representatives Subcommittee on Courts, the Internet, and Intellectual Property. Também participou do evento *Patent Quality Improvement: Post-Grant Opposition*, Oversight Hearing, June 24, 2004, U.S. House of Representatives Subcommittee on Courts, the Internet, and Intellectual Property.

¹¹⁵ <http://www.eff.org/lang/portuguese/>

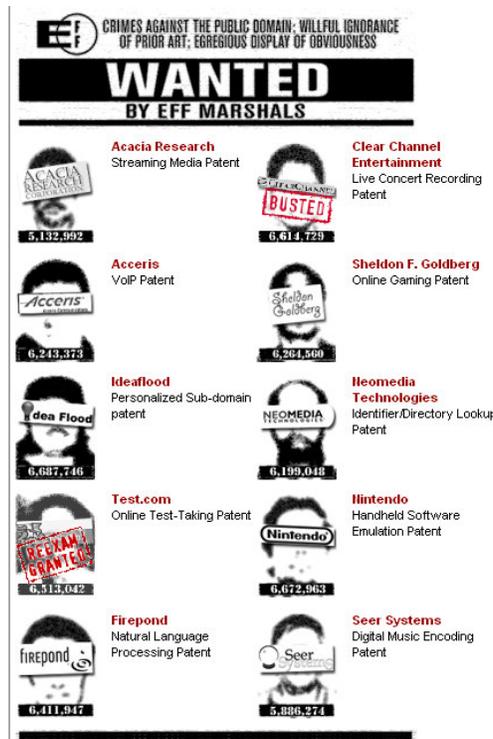


Figura A1.1 - Lista de patentes no alvo da ação da EFF

Do lado das empresas, há iniciativas visando diminuir custos de transação no sentido de formar pacotes de patentes para apoiar o desenvolvimento do SL/CA, o que segundo Samuelson (2006) representa uma virada no uso dos DPIs na indústria de software. Outras iniciativas incluem atividades de inovação aberta (Chesbrough, 2006):

Doação de patentes da IBM para OSI - Em 2004, como já dito acima, a IBM doou 500 patentes para OSI como “parte de uma nova política de propriedade intelectual interna, motivada pela globalização e pela pressão dos custos” (Mendes, 2006, p. 217). Segundo a IBM, enquanto a propriedade é essencial, é possível um balanço entre proteção e compartilhamento. A inovação técnica depende do conhecimento compartilhado e da inovação colaborativa. A nova política marca uma nova era em que a propriedade intelectual irá beneficiar seus clientes e parceiros. As patentes incluem tecnologias de processamento da linguagem natural, interfaces de usuários, sistemas operacionais, interoperabilidade de bases de dados, protocolos de exportação de arquivos, entre outras. Segundo a empresa, não se trata de patentes que não usa mais (Wagner, 2005).

Criação da Open Invention Network (OIN) pelas empresas IBM, NEC, Novell, Philips, Red Hat e Sony para adquirir patentes relacionadas com o Linux e torná-las acessíveis sem a necessidade de pagamento de *royalties* para proteger os fornecedores e compradores do Linux. A OIN está sendo considerada a OTAN (Organização do Tratado do Atlântico Norte) das patentes de software (Techweb, 10 nov 2005).

Dois projetos visam reunir materiais para documentar o estado da técnica e aperfeiçoar o trabalho dos analistas de patentes. São eles:

- **Projeto Patent commons** – projeto dedicado a documentar as fronteiras dos Comuns (The Commons) – um lugar preservado onde os desenvolvedores e usuários de software podem inovar, colaborar e acessar os recursos das patentes num ambiente seguro. A documentação é reunida numa biblioteca disponível em <http://www.patentcommons.org/>. A iniciativa é apoiada pelas empresas CA, Novell, IBM, Red Hat e Sun, além da OSDL (Open Source Development Labs).
- **Open Source Software as Prior Art** – projeto que visa estabelecer um repositório de códigos fonte de SL/CA para servir como base de dados para busca de anterioridade na análise das patentes. OSDL, IBM, Novell, Red Hat e VA Software's SourceForge.net são responsáveis por desenvolver o sistema para prover o acesso online¹¹⁶.

É interessante perceber que o grupo de empresas que participa destas iniciativas é semelhante e que as iniciativas estão relacionadas com o linux e outros programas de infra-estrutura. O que reforça o argumento colocado que, no fundo, trata-se de acabar com uma taxa de monopólio incômoda a todas as empresas (SOFTEX, 2005).

Há ainda as **iniciativas de universidades com o USPTO:**

- **Revisão Aberta de Patente (Open Patent Review)** – um programa que visa estabelecer uma comunidade aberta e colaborativa dentro do processo de exame para aperfeiçoá-lo. Este programa permite que os visitantes do site do USPTO possam escolher assuntos sobre os

quais gostariam de receber informações sobre novos depósitos de patentes. Estabelecido em conjunto com o USPTO, este programa encorajará comunidades a rever patentes em análise e a aperfeiçoar os materiais sobre o estado da técnica¹¹⁷.

- **A comunidade de revisão de patente (The Community Patent Review)** é um projeto piloto do New York Law School Institute for Information Law & Policy em colaboração com o USPTO para aperfeiçoar a qualidade das patentes concedidas possibilitando o acesso do examinador de patentes a uma melhor informação por meio de uma rede aberta para uma comunidade de revisão por pares das patentes depositadas. Os membros da comunidade pertencem a um grupo de empresas: GE, HP, IBM, Microsoft e Red Hat.
- **Índice de Qualidade de Patente** – uma iniciativa que vai criar um índice numérico unificado para avaliar a qualidade das patentes e das aplicações das patentes. O esforço será dirigido pelo Professor R. Polk Wagner da Universidade da Pennsylvania com suporte da IBM e outros e será um recurso aberto e público para o sistema de patentes. Este índice será construído com um *input* extensivo da comunidade, suportado por pesquisa estatística e se tornará uma ferramenta dinâmica e evolutiva com larga aplicabilidade para inventores, participantes no mercado e para a USPTO. Informação sobre o Índice de Qualidade de Patente está disponível em <http://www.patentqualityindex.org/>.

¹¹⁶ Mais informações em <http://developer.osdl.org/dev/priorart/>

¹¹⁷ Mais informações em <http://dotank.nyls.edu/communitypatent>

ANEXO 2 - EXERCÍCIO DE MONITORAMENTO TECNOLÓGICO

De acordo com a literatura, há basicamente duas formas de buscar patentes de software: através de algumas classes tecnológicas da Classificação Internacional de Patentes (IPC na sigla em inglês) e por meio de palavras-chaves na especificação da patente. A primeira forma foi elaborada por Graham e Mowery (2003) e a segunda forma foi elaborada por Bessen e Hunt (2004a)¹¹⁸. A técnica elaborada por Bessen e Hunt (2004a) tem sido utilizada pelas instituições *Public Patent Foundation e Eletronic Frontier Foundation* para monitorar o comportamento das patentes de software. No quadro abaixo é apresentado um resumo dos estudos empíricos.

Quadro A2.1. – Quadro Sinótico dos principais estudos empíricos de patentes de software (continua)

Autores e Áreas	Stuart J. H. Graham – College of Management – Georgia Institute of Technology David C. Mowery - Haas School of Business - University of California
objetivo do estudo	Estudo da proteção da propriedade intelectual na indústria de software norte-americana para examinar as tendências no patenteamento na indústria de software dos EUA nas décadas de 1980 e de 1990
classificação de patente de software	Utilizam 11 grupos da Classificação Internacional de Patentes (IPC), identificados através do exame da atividade de patenteamento total dos 6 maiores produtores de software para computadores pessoais dos EUA entre 1984 e 1995 - Microsoft, Novell, Adobe Systems, Autodesk, Intuit e Symantec. Os grupos são apresentados adiante. Esta conceituação foi inspirada no trabalho de Kortum e Lerner (1999) e tornada mais restrita.
Fontes dos dados	<ul style="list-style-type: none">• Patentes e dados das empresas de software pacote constantes na lista anual da revista <i>Softletter</i>, para os anos de 1985, 1990, 1995 e 1997.• Dados divulgação de invenções, patentes e licenciamento das universidades Stanford University e Columbia University
Variáveis analisadas	<ul style="list-style-type: none">• Patentes de empresas de software pacote: propensão ao patenteamento como % do gasto de P&D, propensão ao patenteamento entre incumbentes (fundadas antes de 1985) e entrantes (fundadas depois de 1985), importância das patentes (medida pela proporção de citação das patentes destas empresas nas classes IPC com a citação total de patentes nestas classes; propensão a copyright• Patentes das empresas de eletrônica: patentes de software em relação ao total de patentes das empresas (IBM, NEC, DEC, Compaq, Hitachi, Fujitsu, National Semiconductor, TI, HP, Rockwell International, Intel e Toshiba)• Patentes de software de universidades (Stanford University e Columbia University): patentes de software em relação ao total, citações das patentes de software como percentual do total de citações das universidades

¹¹⁸ Além destas duas técnicas, há uma terceira elaborada por Allison *et al* (2006), mas como envolve a leitura extensiva de patentes não foi possível replicá-la.

Quadro A2.1. – Quadro Sinótico dos principais estudos empíricos de patentes de software (continuação)

Autores e áreas	James Bessen - Boston University School of Law and Research on Innovation Robert M. Hunt - Federal Reserve Bank of Philadelphia
Objetivo do estudo	Explorar o patenteamento de software nas últimas duas décadas, especialmente o efeito do crescimento das patentes no P&D
Classificação de patente de software	A partir da leitura de uma amostra de patentes, classificadas pelos autores como patentes de software, identificaram algumas características comuns. Com base nestas características, construíram o algoritmo de busca com palavras chaves: <i>((("software" in specification) OR ("computer" AND "program" in specification)) AND (utility patent excluding reissues) ANDNOT ("chip" OR "semiconductor" OR "bus" OR "circuit" OR "circuitry" in title) ANDNOT ("antigen" OR "antigenic" OR "chromatography" in specification))</i> Para validar os resultados (130.650 patentes concedidas entre 1976 e 1999) fizeram uma comparação com a amostra inicial: taxa de falso positivo de 16% e falso negativo de 22%. Também realizaram outros testes baseados na literatura.
Fontes de dados	Patentes concedidas no USPTO Dados de gastos com P&D de 1999 da base Compustat. Dados de citações das patentes do NBER Patent Citations Data File price indices do BLS Multifactor Productivity series.
Variáveis analisadas	Tipo de proprietário das patentes; distribuição através das indústrias; propensão ao patenteamento em função do gasto com P&D, emprego, data de fundação, tipo de indústria, relação do patenteamento com copyright, produtividade dos desenvolvedores de software, mudança no custo/efetividade do patenteamento

Fonte: Elaboração própria.

Com o objetivo de observar a obtenção de patentes de algumas empresas tratadas de software, foi realizado um exercício de monitoramento utilizando as patentes de 16 empresas. As empresas foram selecionadas por terem participado das audiências promovidas pelo USPTO em 1994 ou por representarem tendências mais recentes da indústria de software, como Red Hat, uma das mais importantes distribuições do Linux. As empresas selecionadas são apresentadas na tabela abaixo.

Com o apoio da empresa Elabora Innovation Analysis¹¹⁹, especialmente por meio do trabalho de André Drummond, e das bolsistas Ana Serino de Rezende e Paola Tame do GEOPI foram coletadas 3 conjuntos de patentes:

1. o total de patentes das empresas da amostra, usando as variações dos nomes das empresas no campo depositante (campo ASSIGNEE);
2. usando a técnica de Graham e Mowery (2003) via IPC; e
3. usando a técnica de palavras-chave de Bessen e Hunt (2004a).

A busca foi feita no site da base de patentes americanas USPTO (www.uspto.gov), para o período de 1980-2006, na base de patentes concedidas e na base de patentes depositadas. O quadro abaixo apresenta as queries utilizadas na busca do total de patentes da amostra, a frequência de patentes concedidas e de patentes depositadas:

Quadro A2.2 – Empresas que compõem a amostra

Empresa	Query	Patentes concedidas	Patentes depositadas
Adobe Systems Incorporated*	An/("Adobe Systems" or "Adobe System")	377	8
Apple Computer	An/("Apple Computer" or "Apple Computers")	2098	77
Intel Corporation*	An/("intel" and corp\$)	13217	386
Microsoft Corporation*	An/microsoft	6406	8266
Wind River Systems*	An/"wind river"	43	2
Synopsys*	An/("synopsys" or "synopsis")	281	37
Sun Microsystems, Inc.*	An/("sun" and micros\$)	5888	0
Silicon Graphics, Inc.*	An/"silicon graphics"	651	0
Open Invention Network	An/"open invention network"	3	0
Borland International, Inc.*	An/("borland international" or "Borland Software Corporation" or "borland")	111	13
Autodesk, Inc.*	An/("autodesk inc " or "autodesk canada")	229	104
IBM*	An/("International business machines" or "ibm" or "International business machine")	43974	18881
Novell	An/"novell"	284	41
Red Hat	An/"red hat"	8	22
EDS	an/"Electronic Data Systems"	223	89
Oracle Corporation*	An/("Oracle International Corporation" or "oracle corporation")	787	537

Fonte: Pesquisa de campo.

Nota: * Empresas que participaram da Audiência do USPTO.

Usando as duas formas, surgem os seguintes resultados:

¹¹⁹ <http://www.elabsis.com.br/pt/index.php>

- A busca com classes IPC de Graham e Mowery (2003) traz como resultados menos da metade do número de patentes que a query elaborada por Bessen e Hunt (2004a);
- Apesar da disparidade de valores absolutos, ambas mostram a mesma tendência, qual seja, que o total de patentes concedidas (entre 1976 e abril de 2007) é próximo do total de patentes depositadas, mas que ainda encontram-se em processo de análise. Este seria um indicativo de uma nova aceleração da corrida patentária.

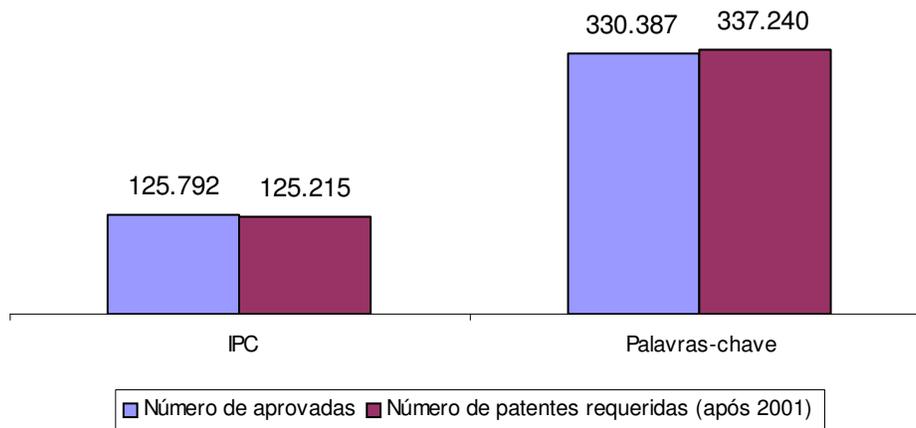


Gráfico A2.1 – Resultados das buscas por patentes concedidas ou publicadas (em processo de análise) no USPTO segundo as estratégias de busca de Graham e Mowery (2003) e Bessen e Hunt (2004a) – abril de 2007

Fonte: Elaboração própria a partir de buscas no site USPTO <www.uspto.gov>

Os 11 grupos selecionados pelo estudo como os principais para caracterização da indústria de software se apresentam abaixo:

- G06F 3/ (Input arrangements for transferring data to be processed into a form capable of being handled by the computer.)
- G06F 5/ (Methods or arrangements for data conversion without changing the order or content of the data handled.)
- G06F 7/ (Methods or arrangements for processing data by operating upon the order or content of the data handled.)
- G06F 9/ (Arrangements for programme control.)

- G06F 11/ Error detection; Error correction; Monitoring.
- G06F 12/ Accessing, addressing or allocating within memory systems or architectures.
- G06F 13/ Interconnection of, or transfer of information or other signals between, memories, input/output devices or central processing units.
- G06F 15/ Digital computers in general.
- G06K 9/ Methods or arrangements for reading or recognizing printed or written characters or for recognizing patterns.
- G06K 15/ Arrangements for producing a permanent visual presentation of the output data.
- H04L 9/ Arrangements for secret or secure communication.

A metodologia foi testada a partir da análise da classificação IPC das patentes de duas empresas da base gerada: a Microsoft Corporation, uma empresa de produtos e a Sun Microsystems, uma empresa de serviços. Como cada patente pode ser classificada em mais de um grupo a porcentagem utilizada na análise foi em termos da quantidade de classificações geradas e não em relação à quantidade de patentes da empresa. O gráfico abaixo apresenta as porcentagens dos grupos do estudo de Graham e Mowery (2003):

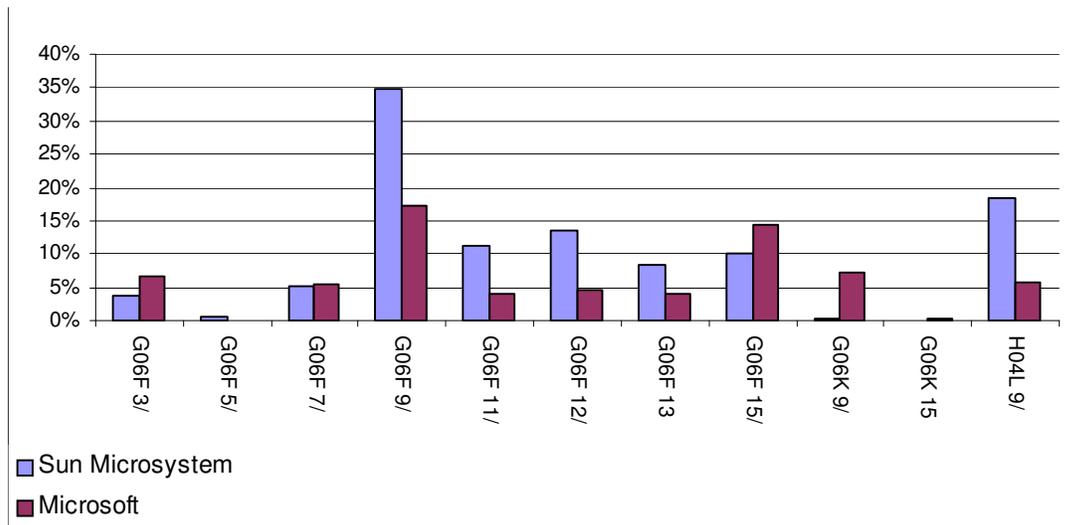


Gráfico A2.2 – Percentual de patentes classificadas nos IPCs da query de Graham e Mowery (2003) da Sun e Microsoft

Fonte: Pesquisa de campo.

Analisando as classificações mais frequentes das patentes das duas empresas algumas novas

classificações, não citadas no estudo de Graham e Mowery (2003), se destacaram. Abaixo se encontram as principais classificações das duas empresas no nível de subclasses, um nível acima da classificação proposta no estudo e o qual o significado de cada uma.

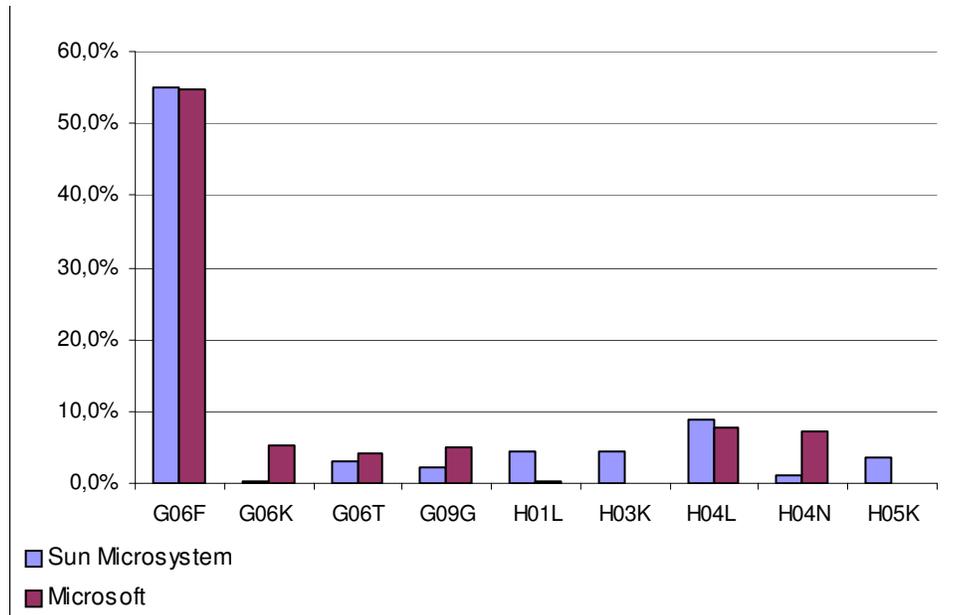


Gráfico A2.3 – Percentual de patentes classificadas nos IPCs mais frequentes da Sun e Microsoft

Fonte: Pesquisa de campo.

- 1- G06F (ELECTRIC DIGITAL DATA PROCESSING)
- 2- G06K (RECOGNITION OF DATA; PRESENTATION OF DATA; RECORD CARRIERS; HANDLING RECORD CARRIERS)
- 3- G06T (IMAGE DATA PROCESSING OR GENERATION, IN GENERAL)
- 4- G09G (ARRANGEMENTS OR CIRCUITS FOR CONTROL OF INDICATING DEVICES USING STATIC MEANS TO PRESENT VARIABLE INFORMATION)
- 5- H01L (SEMICONDUCTOR DEVICES; ELECTRIC SOLID STATE DEVICES NOT OTHERWISE PROVIDED FOR)
- 6- H03K (PULSE TECHNIQUE)
- 7- H04L (TRANSMISSION OF DIGITAL INFORMATION, e.g. TELEGRAPHIC COMMUNICATION)
- 8- H04N (PICTORIAL COMMUNICATION, e.g. TELEVISION)
- 9- H05K (PRINTED CIRCUITS; CASINGS OR CONSTRUCTIONAL DETAILS OF ELECTRIC APPARATUS; MANUFACTURE OF ASSEMBLAGES OF ELECTRICAL COMPONENTS)

É possível observar uma diferença na classificação das duas empresas, possivelmente devido à diferença em seus modelos de negócio e de uma diferença na composição de classificação em

relação à proposta no estudo, ainda que as classificações mais frequentes encontram-se na classe G06F, contemplada no estudo. Como exemplo de novas classes, destaca-se a classe H04N (pictorial communication, e.g. television) para Microsoft e H04L (transmission of digital information, e.g. telegraphic communication) para as duas empresas.

BIBLIOGRAFIA

- ABRANTES, A. C. S. Patentes de software no Brasil. In: *A Propriedade Intelectual no Mundo Digital - A Importância para o Desenvolvimento Brasileiro – ABPI e PFF*. São Paulo, abril de 2006. Disponível em <<http://www.pff.org/digitalamerica/presentations/saopaulo/AntonioCarlosAbrantes11042006.pdf>>, (15/04/2006).
- ALLISON, J. R. *et al.* Patents and business models for software firms. *ExpressO Preprint Series*, n. 1239, 2006 (draft). Disponível em <http://law.bepress.com/expresso/eps/1239>. Acessado em: 05/07/2007.
- ANDRADE, E. *et al.* Propriedade Intelectual em Software: o que podemos apreender da experiência internacional? *Revista Brasileira de Inovação*, v.6, n.1, jan-jun 2007, p.31 - 94
- ALVES Jr., Nilton. *PROTOCOLOS TCP/IP*. CBPF-NT-004/2000, (04/04/01), Disponível em: <<http://mesonpi.cat.cbpf.br/naj/tcpipf.pdf>>, (06/07/2007).
- BAJERSKI C. D. *et al*, Qualidade de Software – Usabilidade. *BateByte*, Edição 38 (Novembro-Dezembro/1994), <<http://www.pr.gov.br/batebyte/edicoes/1994/bb38/qualidad.htm>> (22/05/2007).
- BAKELS, R. E HUGENHOLTZ, P. B. *The patentability of computer programs: discussion of European legislation in the field of patents for software (provisional publication)* (Legal Affairs Series, JURI 1xx EN, 04-2002). European Parliament, 2002. Disponível em <<http://www.europarl.europa.eu/meetdocs/committees/juri/20020619/SoftwarePatent.pub.pdf>>, (11/11/2006).
- BARBOSA, A. L. F. Patentes: crítica à racionalidade, em busca da racionalidade. *Cadernos de Estudos Avançados*, Rio de Janeiro, 2005
- BARBOSA, D. B. *Uma Introdução à Propriedade Intelectual*, v. I. Rio de Janeiro: Lumen Juris, 1997. Disponível em <http://nbb.com.br/public/download.html>. Acessado em: 23/05/2007.
- BASS, L, *et al.* Market Assessment of Component-Based Software Engineering. Technical Note CMU/SEI-2001-TN007 (Volume I), Carnegie Mellon University: EUA, 2001. Disponível em <<http://www.sei.cmu.edu/pub/documents/01.reports/pdf/01tn007.pdf>>, (02/05/2007).
- BATTELLE, J. A Busca: como o Google e seus competidores reinventaram os negócios e estão transformando nossas vidas. Rio de Janeiro: Elsevier, 2006.
- BAUER, Claude J. Meet the component brokers: services, evangelism, and management tools bolster online component marketplace. *developerWorks*, (01/07/2001), Disponível em <www.ibm.com/developerworks/webservices/library/co-broke>, (04/07/2007).

- BESSEN, J. *Patent Thickets: Strategic Patenting of Complex Technologies*. 2003. Disponível em <<http://www.researchoninnovation.org/thicket.pdf>>, (25/04/2007).
- BESSEN, J. e HUNT, R. M. *An empirical look at software patents*. Working Paper n. 03-17/R, 2004a. Disponível em <<http://www.researchoninnovation.org/swpat.pdf>>, (25/04/2007).
- BESSEN, J. e HUNT, R. M. *The software patent experiment*. 2004b. Disponível em <<http://www.researchoninnovation.org/softpat.pdf>>, (05/07/2007).
- BESSEN, S. e MASKIN, E. *Sequential Innovation, Patents, And Imitation*. Working Paper Department of Economics, MIT No. 00-01 January 2000, Disponível em <<http://www.researchoninnovation.org/patent.pdf>>, (10/03/2007).
- BIAGIOLI, Mario. Patent Republic: representing inventions, constructing rights and authors. *Social Research*, v. 73, n. 4, 2006.
- BLACKDUCK. *Best Practices for Government: Managing Software Intellectual Property Assets*, 2007. Disponível em <http://www.carahsoft.com/resources/BlackDuck/BlackDuck_ManagingSWIPAssetsGovernment.pdf>. (05/07/2007).
- BUAINAIN, A. M. e CARVALHO, S. M. P. Propriedade Intelectual em mundo globalizado. *Parcerias Estratégicas*, Brasília, n.9, p.145-153, 2000.
- BUAINAIN, A. M. *et al. Propriedade Intelectual e Inovação Tecnológica: algumas questões para o debate atual*, 2005, Disponível em <http://www.desenvolvimento.gov.br/arquivo/sti/publicacoes/futAmaDilOportunidades/futIndustria_2_00.pdf>, (05/02/2006).
- CAMPBELL-KELLY, M. *From airline reservations to sonic the hedghog: a history of the software industry*. Cambridge: The MIT Press, 2003.
- CARNEIRO, A. M. A.; ZACKIEWICZ, M.; BONACELLI, M. B. M. Monitoramento de Ciência e Tecnologia na Gestão da Inovação. In: *XXIII Simpósio de Gestão da Inovação Tecnológica*. 2004, Curitiba.
- CARR, N. G., It doesn't matter. *Harvard Business Review*, v. 81, n. 5, p.41-49, 2003.
- CARVALHO, S. M. P. *Propriedade intelectual na agricultura*. 2003. Tese (Doutorado em Política Científica e Tecnológica) – Instituto de Geociências, Universidade Estadual de Campinas, Campinas, 2003.
- CASTELLS, M. *A sociedade em rede*. São Paulo: Paz e Terra, 1999. (A Era da Informação: Economia, Sociedade e Cultura; 1).
- CERQUEIRA, T. Q. *A study of conflicts in software protection in the European Community*. Rio de Janeiro: Ed. Esplanada, 1999.

- CHAVEZ *et al.* Software component licensing: a primer. *IEEE Software*, p. 47-53, 1998.
- CHESBROUGH, H. *et al.* Introduction to the Research Policy: 20th anniversary special issue of the publication of “Profiting from Innovation” by David J. Teece. *Research Policy*, v. 35, n.8, p. 1091–1099, 2006.
- COASE, R. H. *La empresa, el mercado y la ley*. Madrid: Alianza Editorial, 1994.
- COHEN, J. E. e LEMLEY, M. A. Patent scope and innovation in the software industry, *California Law Review*, v. 89, n. 1, p.4-57, 2001.
- COHEN, W. M. *et al.* R&D spillovers, patents and the incentives to innovate in Japan and the United States. *Research Policy*, v.31, n.8-9, 2002.
- COMPONENTSOURCE. *Making Reuse reality: eight years of building components supply, aiding components, adoption e driving reuse.* (s/d). Disponível em <<http://www.componentsource.com/services/aboutcomponentsource.asp>>, (19/06/2007).
- CORIAT, B. e DOSI, G. Learning how to govern and learning how to solve problems: on the co-evolution of competences, conflicts and organizational routines. In: DOSI, G. *Innovation, Organization and Economic Dynamics: Selected Essays*. Cheltenham: Edward Elgar, 2000. p.294-324.
- CORIAT, B. e ORSI, F. Establishing a new intellectual property rights regime in the United States: Origins, content and problems, *Research Policy*, v. 31, n. 8-9, p. 1491-1507, 2002.
- CUSUMANO, M. A. *The business of software: what every manager, programmer, and entrepreneur must know to thrive and survive in good times and bad*. New York: Free Press, 2004.
- DAHLANDER, L. Appropriating returns from open innovation processes: a multiple case study of small firms in open source software. 2004. Disponível em <<http://opensource.mit.edu/papers/dahlander.pdf>>, (18/06/2007).
- DAHLANDER, L. e WALLIN, M. W. A man on the inside: unlocking communities as complementary assets. *Research Policy*, v. 35, n.8. p. 1243–1259, 2006.
- DAVIS, J. e HARRISON, S. S. *Edison in the Boardroom: how leading companies realize value from their intellectual assets*. New York: John Wiley & Sons, Inc. 2001.
- Dosi, G. e Marengo, G. Some Elements of an Evolutionary Theory of Organizational Competences. In: ENGLAND, R. (ed.) *Evolutionary Concepts in Contemporary Economics*. Ann Arbor: University of Michigan Press. p.157-178. 1994.
- DOSI, G. e NELSON, R. R. An introduction to evolutionary theories in economics. *Evolutionary Economics*, v.4, n.3, p.153-172, 1994.

- DOSI, G. *et al.*, Towards a theory of Corporate Coherence: a preliminary remarks. In: DOSI, G. *et al* (eds.), *Technology and Enterprise in a Historical Perspective*. Oxford: Clarendon Press, 1992.
- DOSI, G. Institutions and market in a dynamic world. *The Manchester School*, v. LVI, n. 2, p.119-146, 1988.
- DOSI, G. *Technical Change and Industrial Transformation - the Theory and an Application to the Semiconductor Industry*. Londres: Macmillan, 1984.
- DOSI, G.*et al*, How much should society fuel the greed of innovators? On the relations between appropriability, opportunities and rates of innovation. *Research Policy*, v. 35, n.8, p. 1110–1121, 2006.
- FEDERAL TRADE COMMISSION. *Promote innovation: the proper balance of competition and patent law and policy*. A Report by the Federal Trade Commission. 2003. Disponível em <<http://www.ftc.gov/os/2003/10/innovationrpt.pdf>> Acessado em: 05/07/2007.
- FERREIRA, Claudio. O ASP não morreu. *TI Inside*, jul 207, p.40-42.
- FOLHA ONLINE, Google é marca mais valiosa do mundo, indica pesquisa da Folha Online. (24/04/2007). Disponível em <<http://www1.folha.uol.com.br/folha/dinheiro/ult91u116375.shtml>>, (23/04/2007).
- FRAKES, W. B. e KANG, K. Software Reuse Research: Status and Future. *IEEE Transactions on Software Engineering*, v. 31, n. 7, jul 2005.
- FRIED, I. At Microsoft, the yin and yang of Linux. *CNET News.com*, (12/08/2005). Disponível em < http://news.com.com/At+Microsoft%2C+the+yin+and+yang+of+Linux /2100-7344_3-5829512.htm>, (Acessado 24/04/2007).
- GALLINI, N. T. The economics of patents: lessons from recent U.S. patent reform. *The Journal of Economic Perspectives*, v.16, n.2, p.131-154, 2002.
- GOSH, R. *Study on the: Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU: Final report*. Maastricht: Merit. 2006. Disponível em < <http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf>>.
- GRAHAM S. J. H. & MOWERY D. C., Intellectual Property Protection in the U.S. Software Industry. In: COHEN, W. M. e MERRILL, S. A. (eds.) *Patents in the knowledge-based economy*, 2003. Disponível em: <http://faculty.haas.berkeley.edu/graham/unix/swconf.pdf>. (16/06/2007).
- GRAHAM, S. e SOMAYA, D. The Use of Patents, Copyrights and Trademarks in Software: Evidence from Litigation, In: *Patents Innovation and Economic Performance*, proceedings of

- the OECD conference on IPR, Innovation and Economic Performance, 28-29 August 2003, 2004.
- GRAHAM, Stuart J. H. e MOWERY, David C. The Use of USPTO “Continuation” Applications in the Patenting of Software: Implications for Free and Open Source. *LAW & POLICY*, v. 27, n. 1, Jan 2005.
- GRINDLEY, P. C. and TEECE, D. J. (1997) Managing intellectual capital: licensing in semiconductors and electronics, *California Management Review*, 39(2), pp. 8-41.
- GUTIERREZ, R. M.; ALEXANDRE, P. A. M. *Complexo eletrônico: introdução ao software. BNDES Setorial*, Rio de Janeiro, no. 20, p. 3-76, set. 2004.
- HALL, B. H. Business Method Patents, Innovation, and Policy. *UC Berkeley Department of Economics Working Paper*, Berkeley, n. E03-331, 2003. Disponível em <<http://repositories.cdlib.org/iber/econ/E03-331>>, (28/04/2007).
- HART, R. *et al.* The Economic Impact of Patentability of Computer Programs: Report to the European Commission. Study Contract ETD/99/B5-3000/E/106. London: Intellectual Property Institute, 2000. Disponível em <<http://www.ivir.nl/publications/other/softwarepatent.pdf>>, (24/04/2007).
- HEUVEL, Willem-Jan van den e SMITS, Martin. Transformation of the software components and web services market. *20th Bled eConference eMergence: Merging and Emerging Technologies, Processes, and Institutions* June 4 - 6, 2007, Bled, Slovenia. Disponível em <[http://domino.fov.uni-mb.si/proceedings.nsf/Proceedings/5995F3F1CBD9DE97C12572EE0077C67B/\\$File/Paper67.pdf](http://domino.fov.uni-mb.si/proceedings.nsf/Proceedings/5995F3F1CBD9DE97C12572EE0077C67B/$File/Paper67.pdf)> (15/07/2007).
- HORNIK, D. The Softwareless Software Company. *VentureBlog* (09/04/2007), Disponível em <<http://p6.hostingprod.com/@www.ventureblog.com/articles/indiv/2007/001276.html>>, (22/05/2007).
- JUSSAWALLA, M. *The Economics of Intellectual Property in a World without Frontiers: a Study of Computer Software*. New York: Greenwood Press, 1992. (Contributions in economics and economic history, 131).
- KINTISCH, E. Profile Dan Ravicher. *SCIENCE*, v. 1320, p.309, 26/08/2005.
- KORTUM, S. e LERNER, J. What is behind the recent surge in patenting? *Research Policy*, v. 28, n.1, p. 1-22, 1999.
- LAMONICA, M. Google aims for Web developers' hearts and minds: latest products hint at company's efforts to create a platform on which to build add-on products and services. *CNET News.com*, (26/08/2006), Disponível em

<http://news.com.com/Google+aims+for+Web+developers+hearts+and+minds/2100-1007_3-5843402.html?tag=st.prev>, (18/06/2007).

LAMONICA, Martin. From Web page to Web platform, *CNET News.com* (16/08/2005), Disponível em <http://news.com.com/From+Web+page+to+Web+platform/2100-7345_3-5833940.html>, (28/04/2007).

LAMONICA, Martin. Is the 'Web OS' just a geek's dream?, *CNET News.com*, (09/04/2007), Disponível em <http://news.com.com/Is+the+Web+OS+just+a+geeks+dream/2100-7345_3-6174111.html?tag=st.prev%20%EF%83%A0%20idem>, (28/04/2007).

LAMONICA, Martin. Newsmaker: Ozzie's quiet revolution at Microsoft, *CNET News.com*, (30/04/2007), Disponível em <http://news.com.com/Ozzies+quiet+revolution+at+Microsoft/2008-1012_36180428.html?tag=item>, (28/07/2007).

LAWSKY, D. e ZAWADZKI, S. Microsoft to face challenge over Linux licenses. *Reuters*, 21/11/2006.

MACINNES I. Compatibility standards and monopoly incentives: the impact of service-based software licensing, *Int. J. Services and Standards*, v. 1, n. 3, p.255–270, 2005.

MADANMOHAN, T. R. and DE', R. Open Source Reuse in Commercial Firms. *IEEE Software*, v.21, n.6, 2004, p.62-69.

MALERBA, F. e ORSENIGO, L. Schumpeterian patterns of innovation are technology-specific. *Research Policy*, v. 25, p. 451-478, 1996.

MENDES, C. I. C.. *Software Livre e Inovação Tecnológica: Uma Análise sob a Perspectiva da Propriedade Intelectual*. 2006. Dissertação (Mestrado em Economia). Instituto de Economia, Universidade Estadual de Campinas, Campinas: 2006.

MERGES, R. P *et al.* *A Intellectual Property in the New Technological Age*. 3 ed. New York: Aspen, 2006.

MERGES, R.P. A Comparative Look at Intellectual Property Rights and the Software Industry. In: MOWERY, D.C (eds.) *The international computer software industry: a comparative study of industry evolution and structure*. New York: Oxford University Press, 1996.

MONIZ, P. de P. Patenteabilidade de métodos de fazer negócios implementados por software: da perspectiva externa ao ordenamento jurídico pátrio. In: COSTA, A. M. de A. *et al.*(eds.), *Aspectos polêmicos da Propriedade Intelectual*. Rio de Janeiro: Lumens Júris Editora, p.153-203. 2004 (Col. Propriedade Intelectual – Org. Denis Borges Barbosa).

MORISIO, M. *et al.* Success and failure factors in software reuse. *IEEE Transactions on software Engineering*, v. 28, n. 4, p.340-357, 2002.

- NELSON, R. R. Reflections of David Teece's "Profiting from technological innovation..." *Research Policy*, v. 35, p. 1107–1109, 2006.
- NELSON, R., WINTER, S. An evolutionary theory of economic change. Cambridge: Belknap Press, 1982.
- O'REILLY, Tim. The Open Source Paradigm Shift. In: FELLER, J. *Perspectives on Free and Open Source Software*. Cambridge, Massachusetts: The MIT Press, 2005. p.461-482.
- OCDE, OECD Information Technology Outlook: Information and Communications Technologies, Directorate for Science Technology and Industry, 2004.
- OCDE, OECD Information Technology Outlook: Information and Communications Technologies, Directorate for Science Technology and Industry, 2006.
- OCDE, OECD Information Technology Outlook: Information and Communications Technologies, Directorate for Science Technology and Industry, 1997.
- PbT Consultants The results of the European commission consultation exercise on the patentability of computer implemented inventions. 2001. Disponível em <http://ec.europa.eu/internal_market/indprop/docs/comp/softanalyse_en.pdf>, (12/07/2007).
- PISANO, G. Profiting from innovation and the intellectual property revolution. *Research Policy*, v.35, n.8, p.1122-1130, 2006.
- RAPPAPORT, A.S. e HALEVI, S. The computerless computer company. *Harvard Business Review*, v. 69, n.4, Jul-Ago 1991, p.69-80.
- RAVICHANDRAN, T e ROTHENBERGER, M. A. Software reuse strategies and Component markets, *COMMUNICATIONS OF THE ACM*, v. 46, n. 8, 2003. p.109-114.
- RICCIUTI, M. SAP-Microsoft talks underscore harsh market reality. *CnetNews.com 2004*, Disponível em <http://news.com.com/Eyes+on+enterprise/2009-1001_3-5250319.html>, (09/07/2007).
- RIVETTE K. G. e KLINE, D. *Rembrandts in the Attic: unlocking the Hidden Value of Patents*. Cambridge: Harvard Business School Press. 2000.
- ROSA, João Luiz. HP reforça a área de serviços e tenta incomodar a concorrência. *Valor Econômico*, 14/08/2006.
- ROSELINO, J. E. S. *Uma análise das potencialidades da atividade de software no Brasil à luz das práticas concorrencias no setor*. Dissertação (mestrado em Economia). Instituto de Economia, Universidade Estadual de Campinas – Unicamp. Campinas: 1998.

- ROSELINO, J.E.S. *A indústria de software: o “modelo brasileiro” em perspectiva comparada*. Tese (doutorado em Economia). Instituto de Economia, Universidade Estadual de Campinas – Unicamp. Campinas: 2006.
- ROSSI, A. C. 2004. *Representação do componente de software na FARCSOFT: ferramenta de apoio à reutilização de componentes de software*. Dissertação (mestrado em Engenharia). Escola Politécnica – Universidade de São Paulo, 2006.
- SALLES-FILHO, S. *A apropriação do conhecimento e pesquisa pública*. In: FÓRUM PERMANENTE DE INOVAÇÃO: Propriedade Intelectual, Pesquisa Acadêmica e Políticas de Inovação. Campinas: Unicamp, 2007.
- SALLES-FILHO, S. L. M *et al.* Sistema de Propriedade Intelectual e as Pequenas e Médias Empresas no Brasil. Genebra, Suíça: OMPI, 2006.
- SAMUELSON, P. 2006. IBM’s pragmatic embrace of open source: open source has changed the intellectual property landscape of the software industry. *Communications of the ACM*, v. 49, n. 10, 2006, p.21-25.
- SANTOS, M. J.P. Objeto e limites da proteção autoral de programas de Computador. Tese (doutorado em Direito). Faculdade de Direito da USP, São Paulo, 2003.
- SAUVÉ, J. P. Software Baseado em Componentes. In: *Projeto de Software Orientado d Objeto, 2004*, Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/comp/oque.htm>>, (06/07/2007).
- SHANKLAND, Stephen. HP: Don’t like software patents? *CNET News.com*, (15/02/2005), Disponível em <http://news.com.com/HP+Dont+like+software+patents+Learn+to+deal/2100-7344_3-5577604.html >, (28/12/2006).
- SHAPIRO, C. Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard-Setting. *Innovation Policy and the Economy*, 2001, Disponível em <<http://www.haas.berkeley.edu/~shapiro/thicket.pdf>>, (13/03/2006).
- SHERIF, K. e VINZE, A. Barriers to adoption of software reuse: a qualitative study. *Information & Management*, v. 41, p. 159–175, 2003.
- SOFTEX. *A Indústria de Software no Brasil – 2002 – Fortalecendo a Economia do Conhecimento*. Campinas: SOFTEX, 2003.
- SOFTEX. *Indústria de Software e novas modalidades de outsourcing*. Campinas: SOFTEX, no prelo.
- SOFTEX. *O impacto do software livre e de código aberto na indústria de software do Brasil*. Campinas: SOFTEX, 2005.

- SOFTEX. *Perspectivas de desenvolvimento e uso de componentes na indústria brasileira de software e serviços*. Campinas: SOFTEX. 2007.
- STEINMUELLER, W. E. The U.S. software industry: an analysis and interpretative history. In: MOWERY, D. C. *The international computer software industry: a comparative study of industry evolution and structure*. Oxford: Oxford Univ., 1996.
- SYSTEMS INTEGRATOR INC. *Component License Broker™* - Component metering tool for intranet, 2005. Disponível em <<http://www.sintegrators.com/clb/product.htm>>, (Acessado em 07/07/2007).
- TAME, P. *Análise gráfica do crescimento das patentes de software*. Comunicação por e-mail. 2007.
- TANG, P. e PARE, D. *Gathering the foam: Are business method patents a deterrent to software innovation and commercialization? International Review of Law, Computers & Technology*, Abingdon, v. 17, n. 2, p. 127, 2003.
- TANG, P. *et al.* *Patent protection of computer programmes*. ECSC-EC-EAEC, Brussels-Luxembourg (Final Report), 2001.
- TAURION, C. Blog. Disponível em: <<http://www-03.ibm.com/developerworks/blogs/page/ctaurion>>, (18/06/2007).
- TECHWEB. Industry Group Forms To Buy Linux Patents, *TechWeb*, (10/11/2005) <<http://www.techweb.com/showArticle.jhtml;jsessionid=UWISF4WG1T4CYQSNLDRCKH0CJUNN2JVN?articleID=173601498>>, (11/03/2007).
- TEECE, D. (2000), *Managing Intellectual Capital*, Oxford University Press, New York, NY, .
- TEECE, D. e PISANO, G. The Dynamic Capabilities of Firms: an Introduction. In: DOSI, G., TEECE, D. e CHYTRY (eds) *Technology, Organisation and Competitiveness. Perspectives on Industrial and Corporate Change*, Oxford University Press, 1998.
- TEECE, D. Profiting from technological innovation: implications for integration, collaboration, licensing and public policy. *Research Policy*, v.15, n.6, 1986, p.285-305.
- TEECE, D. Reflections on “Profiting from Innovation”, *Research Policy*, v.35, n.8, 2006, p.1131-1146.
- TIGRE, Paulo Bastos. *Gestão da Inovação: A economia da tecnologia no Brasil*. Rio de Janeiro: Elsevier, 2006.
- TOLLIVER, M. Do you own everything in your software portfolio? *CnetNews.com*, 2005, disponível em <http://news.com.com/Do+you+own+everything+in+your+software+portfolio/2010-1071_3-5794276.html>, (22/11/2006).

- TORRISI, S. *Industrial Organisation and Innovation: an international study of the software industry*. Cheltenham: E. Elgar: 1998.
- UNITED STATES DISTRICT COURT FOR THE DISTRICT OF COLUMBIA CIVIL ACTION No. 98-1232 (TPJ) and Civil Action No. 98-1233 (TPJ) *Disponível em* <<http://www.usdoj.gov/atr/cases/f3800/msjudge.pdf>, , 1999, p.16>, (11/06/2007).
- USPTO (United States Patent and Trademark Office), *Public Hearing on Use of the Patent System to Protect Software-Related Inventions*, before the United States Patent and Trademark Office, San Jose California, 26 a 27/01/1994, *Disponível em* <<http://www.uspto.gov/web/offices/com/hearings/software/sanjose/sjhrng.text.>>, (25/02/2007).
- VALIMAKI, M. Dual Licensing in Open Source Software Industry. *Systemes d'Information et Management*, v. 8, n. 1, p. 63-75, 2003.
- VITHARANA,P. Risks and challenges of component-based software development. *Communications of the ACM*, v. 46, n. 8, 2003.p.67-72.
- von HIPPEL, G. e von KROGH, E. Special issue on open source software development. *Research Policy*, 32:1149-1157, 2003.
- von KROGH *ET AL*, Knowledge reuse in open source software: an exploratory study of 15 open source projects. *Proceedings of the 38th Hawaii International Conference on System Sciences*. 2005.
- WAGNER, Jim. IBM Pledges Patents to Open Source, *InternetNews.Com*, (11/01/2005), *Disponível em* <<http://www.Internetnews.com/dev-news/article.php/3457381>>, (15/03/2007).
- WARSHOFSKY, F. *The patent wars: the battle to own the world's technology*. New York: John Wiley & Sons, 1994.
- WEILER, R. K. Decision Support: overcoming obstacles to asset reuse. *InformationWeek*, 2002. *Disponível em* <www.informationweek.com>, (04/07/2007).
- WEINSTOCK, C. B. e HISSAM, S. A. *Open Source Software: The Other Commercial Software*, 2005, *Disponível em* <<http://opensource.ucc.ie/icse2001/hissamweinstock.pdf>>, (19/03/2007).
- WEST, Joel. How open is open enough?: Melding proprietary and open source platform strategies, *Research Policy*, Volume 32, Issue 7, July 2003, Pages 1259-1285.
- WILLIAMSON, O. E. *Las instituciones económicas del capitalismo*. Mexico: Fondo de Cultura Económica, 1987.

WINTER, S. G. The logic of appropriability: From Schumpeter to Arrow to Teece. *Research Policy*, v.35, n.8, p.1100-1106, 2006.

YAMAMURA, Simone. *Plantas Transgênicas e Propriedade Intelectual: Ciência, tecnologia e Inovação no Brasil frente aos Marcos Regulatórios*. Campinas: Dissertação (Mestrado em Política Científica e Tecnológica). Instituto de Geociências, Universidade Estadual de Campinas – Unicamp.2006.

YARED, Peter. Perspective: Microsoft has OS patents; Linux has none. (04/12/2006)
Disponível em <http://news.com.com/Microsoft+has+OS+patents+Linux+has+none/2010-7344_3-6140341.html>, (18/06/2007).