

Padrões para Negócios baseados em Software de Código Aberto

Rafael de Albuquerque Ribeiro^{1,2}, Romeu de Andrade Guimarães³, Alan Kelon Oliveira de Moraes^{1,2}, Jones Oliveira de Albuquerque⁴, Silvio Lemos Meira^{1,2}

¹C.E.S.A.R. – Centro de Estudos e Sistemas Avançados do Recife, ²Centro de Informática – Universidade Federal de Pernambuco, ³Universidade de Pernambuco - Departamento de Sistemas Computacionais, ⁴Universidade Federal Rural de Pernambuco

rar@cesar.org.br, rag@dsc.upe.br, akom@cin.ufpe.br, joa@ufrpe.br, silvio@cesar.org.br

***Resumo.** Este artigo descreve padrões de negócios e licenciamento de software com a finalidade de facilitar o desenvolvimento de um novo empreendimento baseado em software de código aberto. Os padrões foram identificados baseados em estudos de caso e pesquisas de estratégias de utilização de software de código aberto no mundo corporativo.*

***Abstract.** This paper describes business and licensing patterns aiming to make it easier to develop a new entrepreneurship based on Open Source software. The patterns were identified based on case studies and researches of strategies for using Open Source software in the business world.*

1. Introdução

Após vários exemplos de empresas bem sucedidas baseadas em software de código aberto, por exemplo, Red Hat, MySQL, Trolltech e JBoss (MySQL, 2004; Marson, 2005), o tema passou a atrair atenção considerável da indústria de software (OSBC, 2006; Goth, 2005). O que realmente motivaria o início de um novo empreendimento baseado em software de código aberto e quais seriam as vantagens dessa escolha ganhou a atenção também da academia (Rossi, 2005).

É importante ressaltar que ainda existem algumas dúvidas na utilização comercial de software de código aberto. Destas dúvidas podemos ressaltar duas que merecem uma atenção especial. A primeira – e provavelmente a mais importante – é a percepção dos nichos de mercado onde o software de código aberto é uma solução viável (Karels, 2003; Softex, 2005). A segunda dificuldade – geralmente relacionada aos empreendedores acostumados ao mercado de software tradicional onde a principal fonte de renda é o licenciamento do software e seu código fonte deve ser considerado secreto – é entender como seria possível obter rentabilidade abrindo mão do que era até então considerado matéria prima da companhia (Goldman e Gabriel, 2004). Apesar desses dilemas, as licenças têm papel relevante para o sucesso de um projeto de código aberto (Krill, 2006; Stewart et al., 2005).

Esse trabalho propõe analisar algumas das principais variáveis envolvidas no desenvolvimento de um novo empreendimento baseado em software de código aberto e desenvolver padrões que auxiliem na comunicação dos empreendedores responsáveis pelo desenvolvimento do novo empreendimento.

2. Software de código aberto

Uma das formas de caracterizar o software de código aberto é através da licença de uso utilizada, que deve ter as seguintes premissas (Perens, 1999):

- O software, ao ser adquirido, deve vir acompanhado do código fonte;
- O código fonte do software deve estar disponível para redistribuição sem qualquer restrição ou cobrança de taxas;
- A licença deve permitir a modificação do software e a criação de trabalhos derivados;
- Trabalhos derivados podem ser redistribuídos sob os mesmos termos do trabalho original.

A principal implicação deste licenciamento é a disponibilidade do código fonte para usuários e desenvolvedores. A partir disto, o software é desenvolvido de forma colaborativa e distribuída por qualquer interessado. Assim temos uma alta revisão do código fonte, pois o mesmo está disponível a toda a comunidade, o que permite que falhas e bugs sejam encontrados e corrigidos com maior velocidade que nos softwares convencionais. Além disso, devido a credibilidade associada ao software de código aberto por causa da disponibilidade do código fonte, é possível alcançar um grande número de usuários rapidamente, possibilitando que novas empresas e produtos sejam difundidos com mais rapidez.

No entanto, o excesso de licenças disponíveis¹ pode gerar conflitos devido a restrições diferentes impostas por cada uma. Atualmente há um trabalho de redução do número de licenças existentes desenvolvido pela Open Source Initiative que visa aumentar a compatibilidade entre as licenças de software de código aberto². O grande e crescente número de patentes também é algo preocupante para o software de código aberto, pois é praticamente impossível saber quando um determinado algoritmo está violando alguma patente. Dessa forma grandes empresas detentoras de patentes podem dificultar a utilização de software de código aberto e até atacar desenvolvedores e usuários³.

3. Características dos padrões propostos neste trabalho

Esse trabalho propõe padrões para estratégias de negócio de produtos ou serviços de código aberto (Ribeiro et al, 2006). Os padrões descritos seguem o seguinte formato: Problema, Contexto, Forças, Solução, Usos Conhecidos e Padrões Relacionados.

Para facilitar o entendimento, os padrões foram agrupados de acordo com suas finalidades (Gamma et al, 1995). A escolha de dividir por finalidade resultou na identificação de duas principais categorias: 1) Estratégias de Negócio (Koenig, 2004) e 2) Formas de Licenciamento, relacionada à forma como o software utilizado ou produzido é licenciado.

1 Há cerca de 60 licenças disponíveis em <http://www.opensource.org/licenses/>

2 Open source group to streamline licenses. Acesso 25 de fevereiro de 2007 em http://searchopensource.techtarget.com/originalContent/0,289142,sid39_gci1001014,00.html

3 Judge whittles down SCO claims against IBM. Acesso 25 de fevereiro de 2007 em http://news.zdnet.com/2100-3513_22-6090252.html

3.1. Estratégias de Negócio

Koenig (2004) tinha como objetivo descrever sete estratégias de negócio frequentemente utilizadas com softwares de código aberto. A utilização de software de código aberto por empresas como HP, IBM, Oracle, Peoplesoft entre outras foi analisada de forma a identificar as principais estratégias existentes. Foram identificadas estratégias baseadas tanto em produtos como em serviços, o que demonstra que é possível obter lucros das formas mais variadas quando se utiliza software de código aberto.

3.2. Formas de Licenciamento

Algumas licenças de software de código aberto apresentam a característica de impor uma forma de licenciamento a softwares que as utilizem, esta imposição é descrita por alguns autores como contaminação do código que utiliza ou deriva do software original. Em (State Services Commission, 2006), as licenças analisadas são classificadas em diferentes categorias pela capacidade de contaminação: muito contagiosa, pouco contagiosa e permissiva. Das licenças utilizadas, temos a licença GPL como altamente contagiosa; a LGPL e a MPL, pouco contagiosa, onde elas diferem apenas nos critérios de contaminação; e a Apache e BSD, permissivas.

A forma como um software de código aberto é licenciado é um tema tão complexo e delicado que é possível encontrar livros inteiros sobre o tema (Laurent, 2004). Existem fatores como compatibilidade e possibilidades de utilização em produtos derivados que devem ser levados em conta no momento da escolha de uma licença para o desenvolvimento de um software.

4. Padrões propostos

A seguir os padrões identificados neste estudo serão descritos.

4.1. Padrões de Estratégia de Negócio

4.1.1 O Padrão Apadrinhamento

Problema: Ratear os custos de desenvolvimento de um cerne comum para um software especializado com outras empresas do mesmo ramo de atuação.

Contexto: A maioria dos softwares atuais são compostos por várias camadas sem valor comercial e que sobre elas é implementada o diferencial competitivo que é a camada onde situa-se o real valor do software em questão.

Forças: Estimular a comoditização de uma camada de software visando a redução de custos e impor esta como padrão de facto.

Solução: Incentivar o desenvolvimento de um produto de código aberto que apresenta utilidade para outras empresas ou para uma comunidade de desenvolvedores e a partir do produto de código aberto desenvolver seu produto que será comercializado.

Usos conhecidos: A IBM iniciou a Fundação Eclipse⁴ com o intuito de incentivar o desenvolvimento de uma base sólida para dela derivar seus produtos da linha Rational.

Padrões Relacionados: Por normalmente depender da inclusão de código secreto onde será implementado o diferencial competitivo, o padrão Apadrinhamento é normalmente

4 Eclipse – About Us – <http://www.eclipse.org/org/>

utilizado com os padrões Licenciamento Permissivo e Licenciamento Pouco Contagioso.

4.1.2 O Padrão Otimização

Problema: Disponibilizar uma camada de software que funcione de forma otimizada em relação às camadas inferiores.

Contexto: Soluções de software são normalmente compostas por várias outras camadas sobrepostas de software. Para que o software mais acima da pilha tenha desempenho ótimo, ele precisa ser otimizado para os softwares que estão abaixo dele. Esta otimização normalmente depende do conhecimento do comportamento do software abaixo. Tal conhecimento pode ser facilitado caso o código fonte das camadas inferiores do programa sejam acessíveis. O padrão Otimização é baseado na *Lei da Conservação de Modularidade*, de Clayton Christensen (2004), que diz que ou o sistema integrado ou o subsistema necessita ser modular afim de maximizar a performance do outro sistema.

Forças: Necessidade de aumentar a participação no mercado onde o desempenho de softwares concorrentes seja um ponto que ponha em risco a rentabilidade do produto.

Solução: Ajustar o software de código aberto do qual seu produto depende afim de incentivar as vendas de seu produto por ter um desempenho ótimo no software que foi ajustado.

Usos conhecidos: A Oracle oferece seu SGBD ajustado de forma a otimizar o rendimento quando utilizado no sistema operacional Linux, desta forma, ela pode aumentar a margem de lucro sem sacrificar o desempenho.

Padrões Relacionados: Qualquer padrão de licenciamento. Vale ressaltar que o relacionamento com o padrão de Licenciamento Muito Contagioso só é possível no caso em que o produto roda sobre o Linux, do contrário o comportamento contagioso do licenciamento afetaria o licenciamento do produto comercial licenciado de forma tradicional.

4.1.3 O Padrão Licenciamento Duplo

Problema: Aumentar o número de usuários do software e manter a receita proveniente da venda de licenças.

Contexto: O licenciamento duplo é utilizado por empresas detentoras do controle de todos os direitos do software ou controle suficiente para ditar a forma de licenciamento do mesmo. Desta maneira todo desenvolvimento de novos trechos de código que venham a ser incorporados ao software devem ter os direitos devidamente passados à empresa. Assim garante-se que não haja contradições entre as licenças escolhidas.

Forças: Comercializar de forma diferente o mesmo software baseado no perfil ou forma de uso.

Solução: Oferecer o software com duas possibilidades de licenciamento, uma baseada em uma licença para software de código aberto e uma baseada numa licença tradicional de software.

Exemplos: A MySQL AB oferece o SGBD MySQL como dois produtos distintos

dependendo da utilização⁵, quando o MySQL é utilizado com produtos GPL, é possível utilizar a versão grátis licenciada sob GPL, quando é necessário utilizar em um produto comercial de código fechado, é necessário adquirir a licença da versão comercial que utiliza uma licença tradicional.

Padrões Relacionados: Uma vez que o Licenciamento Duplo está normalmente associado a uma licença restritiva para uso não comercial, é mais indicada a utilização do padrão associado ao padrão Licenciamento Muito Contagioso.

4.1.4 O Padrão Embarcado

Problema: Desenvolver um produto embarcado multi-plataforma com o menor custo possível.

Contexto: O desenvolvimento de software para soluções embarcadas demanda recursos para a produção de uma infraestrutura comum a soluções embarcadas. Exemplos de componentes comuns a soluções embarcadas vão desde as partes integrantes do cerne do sistema operacional até componentes de alto nível como serviços de FTP.

Forças: Reduzir o custo associado ao desenvolvimento de uma solução embarcada tentando reutilizar ao máximo soluções existentes.

Solução: Utilizar um software de código aberto como base de um produto embarcado, normalmente utiliza-se o GNU/Linux ou algum software desenvolvido para ele.

Usos Conhecidos: A Cyclades desenvolve sua linha de produtos para gerenciamento de redes desenvolvidas sobre softwares de código aberto.

Padrões Relacionados: O padrão de Licenciamento dependerá do software utilizado como base para o software do dispositivo, podendo ser possível utilizar em conjunto com qualquer padrão de Licenciamento. Vale mencionar um aspecto interessante na utilização do padrão de Licenciamento Muito Contagioso, só existe imposição na divulgação do código para os clientes do produto, não há nada que obrigue a divulgação pública do código fonte do produto.

4.1.5 O Padrão Assinatura

Problema: Adequar-se ao novo modelo de negócios da indústria de software onde a renda é predominantemente baseada em serviços.

Contexto: A fonte primária de renda das empresas do mercado de software vem mudando de licenças para serviços associados ao software (Culpepper, 2002). De acordo com a pesquisa, a distribuição da renda entre licenças e serviços que em 1998 era de 56% e 44% respectivamente passou a ser, em 2001, 46% e 54%.

Forças: Viabilizar um empreendimento com renda baseada em serviços para uma linha de produtos.

Solução: Distribuir o software de forma gratuita e oferecer serviços agregados como atualizações automáticas do software.

Usos Conhecidos: A Red Hat, produtora de um sistema operacional baseado no GNU/Linux, oferece serviço de atualizações e suporte mediante assinatura.

Padrões Relacionados: Para estimular e facilitar a colaboração é comum utilizar os

5 MySQL AB – MySQL Licensing Policy – <http://www.mysql.com/company/legal/licensing/>

padrões Licenciamento Muito Contagioso e Licenciamento Pouco Contagioso para produtos que utilizam o padrão Assinatura.

4.1.6 O Padrão Hospedagem

Problema: Ter uma plataforma de serviços que independa da plataforma do cliente com o menor custo possível.

Contexto: A Internet possibilitou que serviços fossem oferecidos no mundo inteiro sem custo de deslocamento, tornando bastante atrativo o mercado de provisionamento de serviços.

Forças: Oferecer uma plataforma de serviços especializados onde o cliente não necessite se preocupar com o custo e mão de obra associados a instalação e manutenção do software.

Solução: Implementar os serviços a serem oferecidos utilizando como base softwares de código aberto, dessa forma, é possível reduzir os custos, facilitar o desenvolvimento de aplicações portáteis e independentes de plataforma. E em alguns casos, a vantagem mais importante é não ter a obrigação de revelar os códigos personalizados nem se for utilizado software com o padrão Licenciamento Muito Contagioso.

Usos Conhecidos: O Google e a Amazon são exemplos de empresas que disponibilizam APIs mediante um pequeno custo para o cliente que acessa serviços como busca, localização em mapas entre outros.

Padrões Relacionados: Qualquer padrão de licenciamento. O padrão Hospedagem é outro padrão em que o Licenciamento Muito Contagioso merece uma análise mais cuidadosa.

4.1.7 O Padrão Consultoria

Problema: Adequar-se ao novo formato de renda imposto por um mercado onde os investimentos em licenças de software são baixos.

Contexto: De acordo com um estudo interno da consultoria McKinsey em 1999, das soluções desenvolvidas nas grandes corporações, apenas 30% dos custos eram com licenças de software e os outros 70% com mão de obra. Outra pesquisa do Departamento de Comércio dos EUA mostrou que desde 1962, os investimentos em software de pacotes não passaram de 30% do total de investimentos em software.

Forças: Oferecer serviços de consultoria para grandes empresas interessadas em migrar de plataformas proprietárias de alto custo para plataformas abertas de custo reduzido.

Solução: Fazer parcerias com desenvolvedores de software de código aberto para prestar consultoria a empresas que desejam migrar de produtos proprietários para produtos de código aberto.

Usos Conhecidos: A Covalent oferece serviços de consultoria para empresas que desejam migrar de soluções proprietárias para a linha de produtos web da Apache⁶.

Padrões Relacionados: Qualquer padrão de licenciamento, o padrão vai depender do licenciamento do produto que será prestado consultoria.

6 Covalent – Migration Services – <http://www.covalent.net/services/consulting/migration.html>

4.2. Padrões de Licenciamento

4.2.1 O Padrão Licenciamento Muito Contagioso

Problema: Como evitar o que empresas criem produtos baseados em softwares de código aberto melhorados e não retornem à comunidade ou empresa desenvolvedora do software as melhorias.

Contexto: A distribuição do software com o código poderia ser um atrativo para que se desenvolvesse produtos derivados do software sem que as melhorias fossem divulgadas.

Forças: Incentivar a publicação de software de código aberto.

Solução: Licenciado o software sob a licença GPL da GNU (GNU, 2006).

Usos Conhecidos: A MySQL e a Cyclades utilizam licença GPL em seus produtos.

Padrões Relacionados: Não identificado.

4.2.2 O Padrão Licenciamento Pouco Contagioso

Problema: Como estimular o retorno das melhorias de um software de código aberto sem as restrições da GPL.

Contexto: As restrições impostas na GPL poderiam ser uma barreira muito grande para o desenvolvimento de alguns softwares derivados de software de código aberto, uma vez que certos softwares dependem da implementação de código secreto.

Forças: Permitir que software de código aberto seja utilizado na construção de soluções maiores mas com a obrigação apenas de que as melhorias desenvolvidas no software de código aberto sejam retornadas à comunidade.

Solução: Licenciado o software sob a licença LGPL da GNU ou MPL da Fundação Mozilla. A LGPL (Lesser General Public Licence) foi criada pela GNU com o intuito de proteger apenas o código original do produto, qualquer código que faça apenas ligação a um código LGPL não precisa ser licenciado sob a mesma (GNU, 2005). Com o mesmo intuito, a Fundação Mozilla desenvolveu a MPL (Mozilla Public License), para utilizá-la nos projetos da Fundação. A MPL protege o código original mas ainda assim permite o uso em produtos maiores (Mozilla, 2006). O critério para contaminação do código da Mozilla é baseado nos arquivos dos fontes de um software, se um arquivo fonte original for modificado, essa modificação tem por obrigação ser publicada, caso seja um arquivo novo, não há nenhuma obrigação legal em publicá-la.

Usos Conhecidos: O OpenOffice.org utiliza licença LGPL no OpenOffice e a Fundação Mozilla utiliza a MPL no Mozilla Firefox e em outros produtos da fundação.

Padrões Relacionados: Não identificado.

4.2.3 O Padrão Licenciamento Permissivo

Problema: Ter uma licença que proteja os direitos autorais dos colaboradores de um projeto de código aberto e obrigar que seja divulgado que o produto é derivado de um projeto de código aberto.

Contexto: O comportamento contagioso presente na licença GPL e a restrição para reutilização do código original presente na LGPL e na MPL seriam imposições muito grandes para alguns projetos de código aberto, inviabilizando o apoio de empresas.

Forças: Permitir um maior incentivo de empresas à projetos de código aberto e ao

mesmo tempo oferecer uma chance de reuso para as empresas que incentivam os projetos.

Solução: Licenciar o software sob a licença APL da Fundação Apache, BSD ou derivada de uma das duas.

Usos Conhecidos: FreeBSD e PostgreSQL utilizam o licenciamento BSD e a Fundação Apache utiliza a APL no servidor HTTP Apache.

Padrões Relacionados: Não identificado.

4.3. Relacionamento entre os padrões

Para facilitar o relacionamento dos padrões, a Figura 1 representa de forma visual a relação entre os padrões.

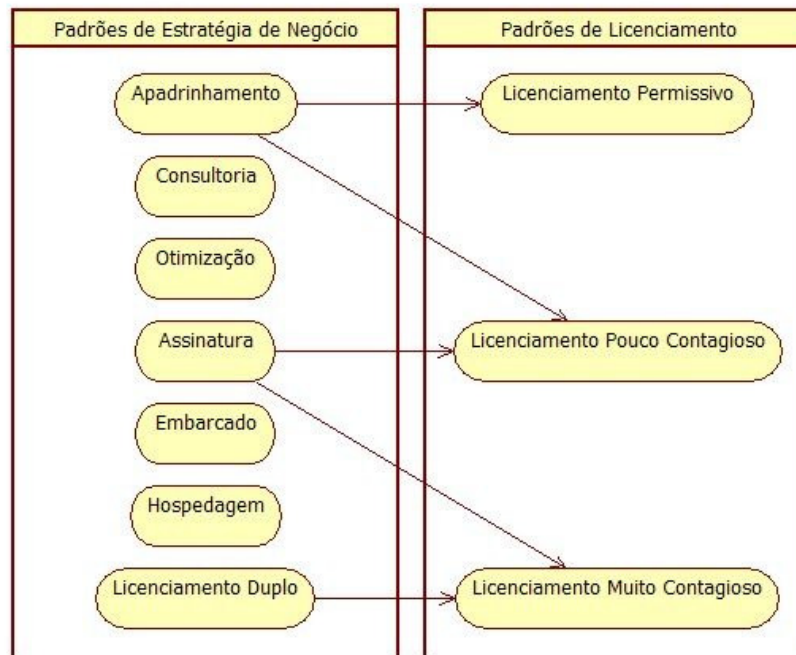


Figura 1 – Relacionamento entre os Padrões

Vale ressaltar que os relacionamentos indicados na figura acima são apenas os casos em que foi identificado um número limitado de padrões de licenciamento relacionados aos padrões de estratégia de negócio. Os casos em que os padrões de estratégia não apresentam nenhum relacionamento são os casos em que eles podem se relacionar com qualquer padrão de licenciamento ou que o padrão de licenciamento relacionado não é tão relevante.

5. Conclusões

Os padrões propostos visam facilitar a comunicação entre os atores envolvidos no desenvolvimento de novos negócios para empresas baseadas em software de código aberto. Duas das principais variáveis envolvidas no desenvolvimento de empreendimentos baseados em software de código aberto foram abordadas: a estratégia de negócio e a forma de licenciamento do software.

Referências

- Culpepper (2002) Software Revenues Continue Shift from Licenses to Services. Acessado em 10 de junho de 2007 em <http://www.culpepper.com/eBulletin/2002/SeptemberRatiosArticle.asp>
- Christensen, C (2004) Clayton Christensen on Disruptive Technologies and Open Source. Paper presented at Open Source Business Conference 2004, San Francisco, CA. Acessado 15 de setembro de 2006 em <http://www.windley.com/archives/2004/03/16.shtml>
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Design. Addison-Wesley, 1995.
- Goldman, R. and Gabriel, R. 2004 Innovation Happens Elsewhere: How and Why a Company Should Participate in Open Source. Morgan Kaufmann Publishers Inc.
- Goth, G. 2005. Open Source Meets Venture Capital. IEEE Distributed Systems Online pp 2, 6(6).
- GNU (2005) GNU Project – GNU Lesser General Public License. Acessado 15 de setembro de 2006 em <http://www.gnu.org/copyleft/lgpl.html>
- GNU (2006) GNU Project – GNU General Public License. Acessado 15 de setembro de 2006 em <http://www.gnu.org/copyleft/gpl.html>
- Karels, M. J. 2003. Commercializing Open Source Software. Queue 1, 5 (Jul. 2003), 46-55.
- Koenig, J, 2004. Seven open source business strategies for competitive advantage, ITManagers. Acessado 16 de outubro de 2005 em <http://pascal.case.unibz.it/retrieve/1537/OpenSourceBusinessModels.pdf>
- Krill, P. 2006. Sun CEO: Open source Java is "momentous". Acessado em 02 de janeiro de 2006 em http://www.infoworld.com/article/06/11/13/HNjavaopenfollow_1.html
- Laurent, Andrew M. St. 2004. Understanding Open Source and Free Software Licensing. O'Reilly Media, Inc.
- Marson I, 2005. JBoss hints at financial success, ZDNET UK. Acessado 9 de outubro de 2005 em <http://news.zdnet.co.uk/software/applications/0,39020384,39189951,00.html>
- Mozilla Project (2006) Mozilla Public License. Acessado 15 de setembro de 2006 em <http://www.mozilla.org/MPL/MPL-1.1.html>
- MySQL, 2004. Leading Open Source Software Companies MySQL AB, Sleepycat Software and Trolltech AS Prove Strength of Dual-License Model. Acessado 9 de outubro de 2005 em http://www.mysql.com/news-and-events/press-release/release_2004_10.html
- OSBC (2006) Open Source Business Conference. Acessado 23 de fevereiro de 2007 em <http://www.osbc.com>
- O'Reilly, T. (2005) The Open Source Paradigm Shift. In Perspectives on Free and Open Source Software, eds. Feller et al., MIT Press
- Perens, B. (1999) The Open Source Definition. Open Source Initiative. Acessado 25 de

fevereiro de 2007 em <http://perens.com/Articles/OSD.html>

Ribeiro, R. A. ; Silva, F. Q. B. ; Moraes, A. K. O. ; Albuquerque, J. O. ; Meira, S. R. L. .
Towards a software licensing guide for Open Source. In: First International Conference on Open Source Collaborative Development Platforms, 2006, Paris. First International Conference on Open Source Collaborative Development Platforms, 2006.

Rossi, C. and Bonaccorsi, A. 2005. Why profit-oriented companies enter the OS field?: intrinsic vs. extrinsic incentives. In Proceedings of the Fifth Workshop on Open Source Software Engineering (St. Louis, Missouri, May 17 - 17, 2005). 5-WOSSE. ACM Press, New York, NY, 1-5.

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

State Services Commission (2006) Guide to Legal Issues in Using Open Source Software. New Zealand Government Policy. Disponível em <http://www.e.govt.nz/policy/open-source/open-source-legal/guide-to-legal-issues-in-using-open-source-software.pdf>

Stewart et al. (2005) A Preliminary Analysis of the Influences of Licensing and Organizational Sponsorship on Success in Open Source Projects. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05).