

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, ³Departamento de Estatística e Informática – Universidade Católica de Pernambuco, ⁴Universidade Federal Rural de Pernambuco

{rar, akom, silvio}@cesar.org.br, romeu.guimaraes@gmail.com, joa@ufrpe.br

Resumo. *Este artigo descreve padrões 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 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 de software de Código Aberto comercialmente. 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, R. e Gabriel, R. 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.

Com isso 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 assim 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 OSI 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

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

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

Este trabalho propõe padrões para estratégias de negócio de produtos ou serviços de código aberto (Ribeiro et al, 2006).

Os padrões descritos neste trabalho seguem o seguinte formato: Problema, Contexto, Forças, Solução, Exemplos e Padrões Relacionados.

Para facilitar o entendimento, agrupamos os padrões 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.

3.1. Estratégias de Negócio

Koenig (2004) tinha como objetivo descrever sete estratégias de negócio freqüentemente utilizadas com softwares de Código Aberto. A utilização de software de Código Aberto por empresas como HP, IBM, Oracle, Peoplesoft entre outras foi analisada afim de 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 utiliza-se software de Código Aberto.

3.2. Formas de Licenciamento

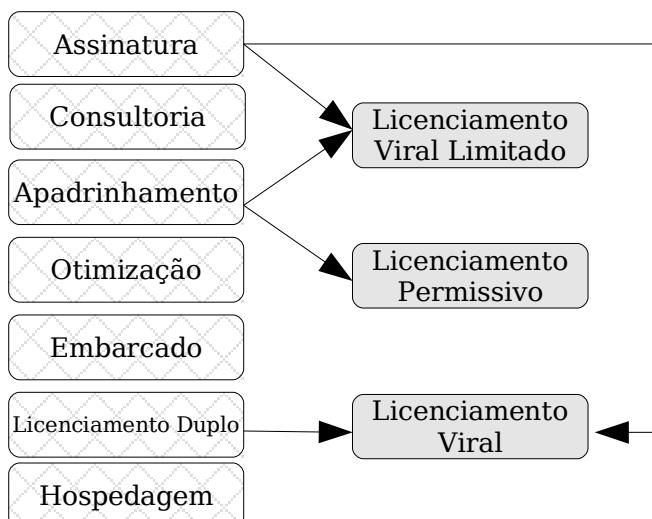
As licenças analisadas são classificadas em diferentes categorias por nível de infecção, de acordo com o proposto por (SSC, 2006), onde as licenças foram classificadas como altamente infecciosa, fracamente infecciosa e permissiva. Das licenças utilizadas temos a GPL como altamente infecciosa, a LGPL e a MPL como fracamente infecciosas onde elas diferem apenas nos critérios de infecção e a Apache e BSD classificadas como 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.

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

4. Padrões propostos

A seguir os padrões identificados neste estudo serão descritos. Para facilitar o relacionamento dos padrões, a figura abaixo representa de forma visual a relação entre os padrões.



Legenda:

Padrões de Estratégia de Negócio

Padrões de Licenciamento

Figura 2 – Relacionamento entre os Padrões

Vale ressaltar que os relacionamentos indicados na figura acima são apenas os casos em que os padrões de Estratégia de Negócio relacionam-se com um número limitado de padrões de Licenciamento. Os casos que não estão indicados são os casos em que os padrões de Estratégia de Negócio podem se relacionar com qualquer padrão de Licenciamento.

4.1. Padrões de Estratégia de Negócio

4.1.1 O Padrão Apadrinhamento

Problema: Compartilhar os custos associados ao desenvolvimento da infra-estrutura comum e fazer uso desse esforço comum em um produto comercializado.

Contexto: Os softwares desenvolvidos para o mercado dos softwares corporativos têm a característica de ter uma base de funcionalidades comuns entre vários clientes mas demandar personalizações para adequar-se a cada cliente.

Forças: O desenvolvimento de um software corporativo normalmente envolve o desenvolvimento de componentes comuns a outros softwares, o que leva a maioria das empresas a consumir parte dos recursos no desenvolvimento dessa infra-estrutura comum.

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.

Exemplos: A IBM iniciou a Fundação Eclipse⁴ com o intuito de incentivar o desenvolvimento de uma base sólida para o desenvolvimento dos 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 utilizado com os padrões Licenciamento Permissivo e Licenciamento Viral Limitado.

4.1.2 O Padrão Otimização

Problema: Ter um produto com desempenho ótimo com o menor custo associado.

Contexto: Um software que soluciona um problema em particular é normalmente composto por vários software sobrepostos por camadas, para que um deles tenha desempenho ótimo, é necessário que as camadas abaixo estejam otimizadas para utilização das camadas dos níveis acima. 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: Maximizar o rendimento do software a ser comercializado fazendo modificações nos softwares que fazem parte da pilha de softwares envolvidos na resolução do problema.

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.

Exemplos: A Oracle oferece seu SGBD ajustado de forma a otimizar o rendimento quando utilizado no Sistema Operacional Linux, dessa 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 Viral só é possível no caso em que o produto roda sobre o Linux, do contrário o comportamento viral do licenciamento afetaria o licenciamento do produto comercial licenciado de forma tradicional.

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

4.1.3 O Padrão Licenciamento Duplo

Problema: Comercializar um software e ao mesmo tempo oferecê-lo livre de cobrança e com o código fonte aberto.

Contexto: Um software pode ser comercializado como dois produtos distintos mesmo compartilhando a mesma base de código, basta que ele tenha duas licenças de uso com termos distintos.

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 gráts 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 Viral.

4.1.4 O Padrão Embarcado

Problema: Reduzir o custo e os riscos de desenvolvimento de uma solução embarcada.

Contexto: O GNU/Linux por ser bastante flexível pode ser encontrado em vários dispositivos embarcados presentes no mercado, isso torna o GNU/Linux uma alternativa bastante atrativa para o desenvolvimento de soluções embarcadas.

Forças: Reutilizar um software de Código Aberto como base para o desenvolvimento de um produto embarcado.

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.

Exemplos: 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

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

com qualquer padrão de Licenciamento. Vale mencionar um aspecto interessante na utilização do padrão de Licenciamento Viral, 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: Maximizar o faturamento proveniente de serviços prestados em cima de um produto que não oferece valor se desacompanhado de um produto mais acima na cadeia de valor.

Contexto: Vários softwares não tem muita utilidade se não forem acompanhados de outro software, contêiners para aplicações J2EE são exemplos dessa categoria de software.

Forças: Empresas que desenvolvem tais softwares podem aumentar a base de usuários distribuindo o software gratuitamente e utilizar de outras formas de faturamento que não sejam cobranças de licença de utilização do software.

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

Exemplos: A Red Hat que produz um sistema operacional baseado no Linux oferece serviço de atualizações e suporte por assinatura para sua linha de sistemas operacionais Linux.

Padrões Relacionados: Para estimular e facilitar a colaboração é comum utilizar os padrões Licenciamento Viral e Licenciamento Viral Limitado para produtos que utilizam o padrão Assinatura.

4.1.6 O Padrão Hospedagem

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

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

Forças: Oferecer serviços de baixo custo a clientes do mundo inteiro através da Internet.

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 portáveis 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 um padrão de Licenciamento Viral.

Exemplos: 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 Viral 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 passou 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.

Exemplos: 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.

4.2. Padrões de Licenciamento

4.2.1 O Padrão Licenciamento Viral

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 característica de distribuir o software com o código era um atrativo para que se utilizasse e não houvesse retorno à comunidade das melhorias implementadas.

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

Para evitar que isso acontecesse, a GNU desenvolveu a licença GPL (GNU, 2005) que obriga que qualquer código que faça ligação a código GPL (General Public License) também seja licenciado sob a GPL.

Forças: Evitar que software de Código Aberto seja utilizado sem que as melhorias sejam retornadas à comunidade.

Solução: Licenciar o software sob a licença GPL da GNU.

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

Padrões Relacionados: Nenhum.

4.2.2 O Padrão Licenciamento Viral Limitado

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 a utilização de vários softwares, para facilitar a utilização do software e ainda assim proteger o código original, a GNU criou a LGPL (Lesser General Public Licence), que só protege o código original do produto, qualquer código que faça apenas ligação não precisa ser licenciado sob a LGPL (GNU, 2006).

Outro licença desta categoria é a MPL (Mozilla Public License), utilizada nos projetos da Fundação Mozilla, a MPL tem o intuito de proteger o código original mas ainda assim permitir o uso em produtos maiores (Mozilla, 2006a). 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.

Forças: Permitir que software de Código Aberto seja utilizado na construção de soluções maiores mas obrigar que as melhorias desenvolvidas no software de Código Aberto sejam retornadas à comunidade.

Solução: Licenciar o software sob a licença LGPL da GNU ou MPL da Fundação Mozilla.

Exemplos: 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: Nenhum.

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 viral presente na licença GPL e a restrição de 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. Para permitir a utilização em projetos comerciais, o projeto Apache utiliza uma licença mais permissiva, a Apache Public License. O mesmo raciocínio foi utilizado pela universidade de Berkeley para desenvolver a licença BSD.

Forças: Permitir um maior incentivo de empresas à projetos de Código Aberto.

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

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

Padrões Relacionados: Nenhum.

5. Conclusões

Os padrões propostos visam facilitar a comunicação entre os atores envolvidos no desenvolvimento de modelos de negócio para empresas baseadas em software de Código Aberto. Abordamos as duas principais variáveis envolvidas no desenvolvimento de empreendimentos baseados em software de Código Aberto: a estratégia de negócio e a forma de licenciamento do software.

Referências

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.

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>
- HAYES, M.; WHEELWRIGHT, S. C. Restouting our competitive edge: competing through manufacturing. New York: Wiley, 1984.
- 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 (2006a) Mozilla Public License. Acessado 15 de setembro de 2006 em <http://www.mozilla.org/MPL/MPL-1.1.html>
- Mozilla Project (2006b) Mozilla Public License FAQ. Acessado 15 de setembro de 2006 em <http://www.mozilla.org/MPL/mpl-faq.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
- O'Reilly, T. (2005) The Open Source Paradigm Shift. In Perspectives on Free and Open Source Software, eds. Feller et al., MIT Press
- 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.
- 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). OSBC (2006) Open Source Business Conference. Acessado 23 de fevereiro de 2007 em <http://www.osbc.com>

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>

OSI. The Open Source Definition. Open Source Initiative. Acessado 25 de fevereiro de 2007 em http://www.opensource.org/docs/definition_plain.php