

Inovação e Liberdade no Desenvolvimento de Software

Fabio Kon, fabio.kon@ime.usp.br, Centro de Competência em Software Livre, IME/USP.

Juliano Maranhão, julianomaranhao@usp.br, Faculdade de Direito, USP.

Nelson Lago, lago@ime.usp.br, Centro de Competência em Software Livre, IME/USP.

Paulo Meirelles, paulormm@ime.usp.br, Centro de Competência em Software Livre, IME/USP.

A propriedade intelectual de software, embora razoavelmente bem definida na legislação, é tema de discussões a respeito de formas de proteção e mecanismos de incentivo à inovação tecnológica. O software livre promove o desenvolvimento de software com base na colaboração e disseminação do conhecimento, e seu sucesso técnico e comercial sugere que se incentive caminhos baseados no livre compartilhamento de ideias. Restrições, como as patentes ou a obrigatoriedade de registro, cumprem um importante papel em outros contextos; no entanto, não são adequadas ao software, além de irem contra essa atual tendência em direção ao conhecimento compartilhado na qual se baseia o software livre.

Um dos componentes fundamentais de qualquer sistema de computação é o software, que efetivamente faz uso do hardware para atingir os mais diversos objetivos. De fato, pode-se dizer que o software expressa a solução abstrata dos mais diversos problemas computacionais, enquanto o hardware é o meio pelo qual o software produz resultados palpáveis. Isso significa que o software traz consigo um amplo corpo de conhecimento relacionado aos mais diversos problemas aos quais a computação costuma ser aplicada. Assim, o software se apresenta cada vez mais como um tema de interesse geral, e não apenas para profissionais da área de computação, na medida em que vários aspectos relacionados a ele vão além de características puramente técnicas.

Boa parte do software atualmente usado e desenvolvido, seja para computadores pessoais ou servidores, seja para uso geral ou específico, é disponibilizado sob licenças restritivas. Essas licenças, em maior ou menor grau, impõem restrições ao seu uso, distribuição ou acesso ao código-fonte. Esse tipo de licenciamento é possível porque o software está sujeito à proteção da lei a respeito dos direitos de autor, que garante ao criador o direito exclusivo de exploração de sua obra. Isso lhe permite autorizar ou não determinadas formas de uso do software por parte dos usuários. Chamamos o software disponibilizado sob licenças que impõem restrições desse tipo de *software restrito, exclusivo ou proprietário*.

Em contraste com o software restrito, o software livre é definido como aquele que é licenciado de forma que permita sua livre utilização, redistribuição e modificação praticamente sem restrições. Essa abordagem para o licenciamento do software tem impacto muito maior que o que se poderia imaginar, pois estabelece uma dinâmica única e potencialmente muito positiva em relação (i) ao processo de desenvolvimento do software; (ii) aos mecanismos econômicos que regem esse desenvolvimento e seu uso; (iii) ao relacionamento entre desenvolvedores, fornecedores e usuários do software e (iv) aos aspectos éticos e legais relacionados ao software.

Benefícios do Software Livre

Em princípio, o que diferencia o software livre do software restrito é apenas a forma de licenciamento; no entanto, essa diferença tem por consequência diversas outras, algumas bastante acentuadas. Várias delas se traduzem em benefícios em relação aos pontos elencados acima. Assim, o software livre se apresenta como uma alternativa vantajosa e de qualidade para o desenvolvimento de sistemas computacionais em diferentes ambientes e contextos, incluindo universidades, empresas, governos e ONGs.

Uma vantagem oferecida pelo software livre em comparação ao software restrito vem do fato que o código-fonte pode ser livremente compartilhado. Esse compartilhamento pode simplificar o desenvolvimento de aplicações personalizadas, que não precisam ser programadas a partir do zero, mas podem se basear em soluções já existentes. Na medida em que o desenvolvimento de aplicações personalizadas é um dos focos do desenvolvimento de software em geral (Ghosh 2006), essa vantagem tem impacto significativo na redução de custos e na diminuição na duplicação de esforços, tirando proveito da característica abstrata do software.

Outra vantagem resultante do compartilhamento do código se refere à possível melhoria na qualidade, mesmo frente aos problemas inerentes à sua complexidade (Raymond 1997). Isso se deve ao maior número de desenvolvedores e usuários envolvidos com o software: de um lado, um número maior de desenvolvedores, com diferentes perspectivas e necessidades, é capaz de identificar e corrigir mais *bugs* em menos tempo; de outro, um número maior de usuários gera situações de uso e necessidades mais variadas, o que se traduz em um maior número de *bugs* identificados e mais sugestões de melhorias.

A reputação do programador também acaba tornando-se um fator relevante para a qualidade do software livre. Enquanto o código-fonte do software restrito é geralmente secreto, o código-fonte do software livre é público. Como consequência dessa exposição, o orgulho pessoal do programador, que sabe que sua produção será avaliada por outros e possivelmente terá reflexos em sua carreira profissional, o leva a ser mais cuidadoso.

O software livre também traz vantagens do ponto de vista econômico. O mercado de software é, na verdade, um mercado de serviços: 80% do dinheiro que circula em atividades vinculadas ao software não se relaciona à venda de licenças (Gosh, 2006). Fornecedores de soluções comerciais baseadas em software livre, portanto, competem na principal fatia do mercado sem estar sujeitos aos problemas do software restrito. Além disso, diferentemente do que ocorre com o software restrito, o software livre promove o estabelecimento de vários fornecedores que competem entre si com base no mesmo software. Essa competição mais forte entre fornecedores traz vantagens para os usuários, pois dá melhores garantias quanto ao desenvolvimento futuro do sistema e induz a uma redução nos preços.

De forma similar, os fornecedores também se beneficiam do compartilhamento do software livre, pois tanto os custos quanto os riscos associados ao desenvolvimento do software são diluídos entre os diversos concorrentes. Por conta disso e das outras vantagens discutidas acima, ele possibilita e até mesmo incentiva o surgimento de pequenas empresas que podem atender seus mercados locais. Por sua vez, esse incentivo às pequenas empresas e a consequente redução na dependência de empresas estrangeiras são economicamente interessantes para países como o Brasil.

Esses efeitos têm se mostrado extremamente positivos na prática. Empresas como IBM, HP, Oracle, Intel, Nokia, Red Hat e Google investem continuamente grandes somas em diversos projetos de software livre. Algumas dezenas desses projetos, tais como o GNU/Linux, os navegadores Web Firefox e Chrome, o sistema operacional para telefones celulares Android, o servidor Web Apache e a máquina virtual Java são utilizados em milhões de computadores. Ambientes utilizando tecnologias tais como Drupal, JQuery, Tomcat, MySQL e PHP são responsáveis por centenas de milhares de negócios eletrônicos, serviços governamentais e sítios educacionais e informativos disponíveis na Web. Finalmente, boa parte dos maiores portais da Internet, tais como Google, Amazon e Wikipédia tem sua infraestrutura fortemente baseada em software livre. Graças a tudo isso, podemos dizer que o software livre é utilizado diariamente por cerca de 2 bilhões de pessoas no mundo, que é a população que atualmente tem acesso à Internet.

Inadequação das patentes de software

O software possui dimensão dupla. De um lado liga-se à realização de uma função por meio de um hardware, com a produção de resultados práticos; nessa perspectiva, o software é um produto. De

outro, é uma criação que expressa uma ideia de determinada forma; nesse caso, o software é equiparável à expressão artística de uma ideia.

A legislação brasileira vê o software menos como produto e mais como expressão intelectual, prevendo que os direitos de autor são o mecanismo próprio de proteção ao software e excluindo explicitamente patentes como opção (Lei 9609/98 e Lei 9279/96, art. 10). O mecanismo de patentes, mais restritivo, levaria ao extremo oposto, tratando o software como algo incorporado a um produto ou processo industrial. Essa visão retroage aos primórdios da computação em que o software praticamente não tinha autonomia em relação ao hardware. Na medida em que o software passou a ser considerado independente, esse vínculo entre a solução computacional e a máquina deixa de fazer sentido; insistir em amarrar o software à máquina ou processo industrial seria o mesmo que patentear o livro de receitas da Dona Ofélia em conjunto com o fogão elétrico. Essa inadequação da aplicação de patentes ao software e seus malefícios já vêm sendo amplamente discutidos pela comunidade de software livre e por membros de indústria de software (Stallman, 2002; Irlam & Williams, 1994).

A proteção pelo direito de autor é favorável ao compartilhamento e ao software livre, pois permite que várias expressões diferentes de uma mesma ideia coexistam e, assim, possibilita a criação de software livre para tratar qualquer tipo de problema, mesmo que já haja programas restritos similares. As patentes, por outro lado, pressupõem a existência de apenas uma ou algumas poucas soluções baseadas na mesma abordagem, todas formalmente autorizadas pelo detentor da patente, em geral com base em um acordo comercial. Esse tipo de organização evidentemente não se adequa ao software livre, onde pode haver múltiplas implementações e variações de um mesmo código sem que haja uma entidade central responsável.

Apesar dessas considerações, os advogados desenvolveram habilidade especial para descrever o software como se fosse parte de um processo industrial, o que tem viabilizado nos Estados Unidos (Samuelson, 1990), e mesmo no Brasil, a sua patente, ao arripio do espírito da legislação. Um agravante desse problema é que o escritório de patentes americano tem apresentado níveis de exigência quanto à originalidade e não-trivialidade muito baixos para o registro de patentes de software, o que torna o número de patentes artificialmente grande.

Além das inadequações discutidas acima, esse artifício dificulta a posição das pequenas empresas: enquanto empresas grandes têm acordos mútuos de licenciamento de patentes, as microempresas podem ser consideradas infratoras a qualquer momento. Ao mesmo tempo, dado o grande número de patentes existentes e sua não-especificidade a uma única área (em virtude da grande versatilidade do software), os custos para o levantamento de possíveis patentes em uso chega a ser proibitivo. Isso significa que o risco para a entrada de novas empresas no mercado de software é muito maior que em outras áreas, o que pode levar à concentração e mesmo ao monopólio.

Se esses problemas afetam profundamente o desenvolvimento de software em geral, eles são ainda mais contundentes no contexto do software livre. Como não é viável cobrar *royalties* de programas livres, patentes de software podem impedir totalmente a existência de vários tipos de software livre. O licenciamento mútuo não serve como opção para minimizar esse problema: como o interesse da comunidade é o compartilhamento do conhecimento e os custos de registro são altos, não há vantagem em registrar patentes sobre quaisquer técnicas desenvolvidas no ambiente do software livre. Também não existe uma entidade com poderes para licenciar patentes de terceiros em nome da comunidade para uso em qualquer contexto. Dado que o software livre permite a reutilização do código em diferentes situações, uma única patente pode ser infringida em um sem-número de ambientes de software diferentes por conta de um único trecho de código. E, como raramente há uma entidade central responsável pelo software, não há verificações sobre possíveis violações de patentes, e eventuais processos a respeito colocam toda a comunidade, ao invés de uma única empresa, em xeque.

Assim, as patentes de software não se prestam ao ambiente de compartilhamento do conhecimento típico do software livre. Ao mesmo tempo, se é verdade que o mecanismo de patentes tem funcionado

adequadamente em outras áreas, pode-se observar que o mesmo não se aplica ao software em geral. Patentes não tiveram um papel relevante no desenvolvimento da tecnologia relacionada ao software nos últimos 50 anos, o que mostra que esse mecanismo é um custo desnecessário a pagar.

Registro de Software

Como visto, os programas de computador são protegidos pelo direito autoral. Desse modo, os direitos relativos à propriedade intelectual do software surgem automaticamente com a criação da própria obra: sua proteção independe de qualquer registro e basta a demonstração da autoria para o criador ter seus direitos garantidos. Esses direitos são reconhecidos internacionalmente pelos países signatários do acordo TRIPS (*trade-related aspects of intellectual property rights*) e sua validade é de 50 anos.

A mesma Lei 9609/98 também conferiu ao criador do software a possibilidade de registrar o programa no Instituto Nacional da Propriedade Industrial – INPI – com o intuito de simplificar a identificação da autoria em casos de dúvida ou litígio. No entanto, se, de um lado, o registro não é necessário para a proteção jurídica, de outro, ele também não constitui comprovação absoluta dessa autoria. Há sempre a possibilidade de questionamento judicial sobre quem foi o efetivo autor. A vantagem do registro consiste em permitir que, diante de um conflito causado pela incerteza da autoria de determinado software, exista a presunção de que o titular do certificado é o criador da obra, cabendo àquele que desafiou o registro o ônus de provar o contrário.

Entretanto, vale considerar que a atual dinâmica de produção de software, notadamente em contexto de produção no modelo de compartilhamento, criou mecanismos e portais de confiabilidade para identificação do autor, como, por exemplo, a disponibilização do software no *sourceforge.net*, *github.com* ou *gitorious.org*. Do ponto de vista jurídico, não se trata de registro na autarquia legalmente designada; porém, do ponto de vista prático e fático, essa iniciativa cria reconhecimento de autoria difícil de ser violado, com a vantagem de evitar a burocracia e produção de toda documentação necessária para o registro, que pode levar até 90 dias, sem falar nos custos, que variam de R\$300,00 a R\$2.500,00 ou, para pessoas físicas, pequenas empresas, instituições de ensino ou fundações, de R\$120,00 a R\$1.000,00. Finalmente, o procedimento de registro se revela anacrônico no contexto de criações envolvendo um grande número de autores colaborando internacionalmente de maneira pouco formal, que é exatamente o caso típico do software livre. Assim, a exigência de registro para qualquer atividade relacionada ao software (como ocorre com o Software Público Brasileiro) não se justifica na prática e cria dificuldades, em especial para o software livre.

Conclusão

O modelo do software livre propõe uma abordagem de desenvolvimento baseada no compartilhamento e colaboração, e tem tido grande sucesso. No entanto, ainda há aspectos da legislação que podem entrar em conflito com essa abordagem; é preciso identificar os meios de favorecer a evolução dessa legislação para que ela não ofereça obstáculos a um modelo que, além de ser eficiente do ponto de vista do mercado e incentivar o surgimento de pequenas empresas, também promove o senso de comunidade, a inclusão social e a disseminação do conhecimento. Se você também considera esse modelo o mais adequado para promover a inovação tecnológica, em seu próximo projeto de desenvolvimento, opte por uma licença de software livre e compartilhe seu código em um repositório aberto, como sourceforge, gitorious, github, launchpad etc. Para saber mais sobre os aspectos jurídicos e modelos de negócios relacionados ao software livre, consulte o texto disponível em <http://ccsl.ime.usp.br/files/slpi.pdf>.

Recursos

Ghosh, R. A. et al. (2006). Economic impact of open source software on innovation and the competitiveness of the information and communication technologies (ICT) sector in the EU. Relatório de Estudo. Disponível em: <http://www.flossimpact.eu> [Acesso em 28 out 2011].

Irlam, G. & Williams, R. N. (1994). Software Patents: An Industry at Risk. Disponível em: <http://progfree.org/Patents/industry-at-risk.html> [Acesso em 29 out 2011].

Raymond, E. S. (1997). The cathedral and the bazaar. Disponível em: <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar> [Acesso em 29 out 2011].

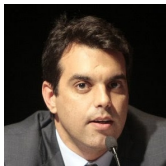
Samuelson, P. (1990). Should program algorithms be patented? Communications of the ACM, vol. 33, num. 8.

Stallman, R. (2002). Software patents: Obstacles to software development. Disponível em: <http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html> [Acesso em 29 out 2011].

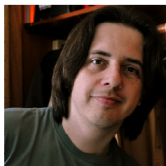
Sobre os autores



Fabio Kon é Professor Titular do Departamento de Ciência da Computação do IME/USP. Atua em projetos de pesquisa e desenvolvimento nas áreas de Software Livre, Coreografias de Serviços Web, Computação em Nuvem entre outras. Fabio é autor de mais de 100 artigos científicos publicados no Brasil e no exterior. Atualmente, em 2011, é o diretor do Centro de Competência em Software Livre (CCSL) do IME/USP, um dos diretores da Open Source Initiative (OSI), Editor-in-Chief do Journal of Internet Services and Applications (JISA) e Program Chair da International Middleware Conference.



Juliano Souza de Albuquerque Maranhão é Professor Livre-Docente da Faculdade de Direito da Universidade de São Paulo e Doutor pela Faculdade de Direito da USP. Tem pós-doutorado pela PUC/SP e pela Universidade de Utrecht, na Holanda. É advogado em São Paulo, coordenador do grupo de estudos “Aspectos Jurídicos do Software Livre” na Faculdade de Direito da USP e assessor do CCSL-USP para aspectos legais do software.



Nelson Lago é Gerente Técnico do Centro de Competência em Software Livre do IME/USP e doutorando em Ciência da Computação, também no IME/USP. Ministrou diversos cursos e palestras sobre linux e software livre, abordando tanto aspectos técnicos quanto conceituais. Tem artigos publicados em eventos no Brasil e no exterior. Participou do processo de criação da ONG “LinuxSP”, onde ofereceu, juntamente com outros voluntários, apoio técnico ao projeto dos telecentros da prefeitura de São Paulo.



Paulo Meirelles é doutorando do Departamento de Ciência da Computação do IME/USP. Desde 2004, é membro do grupo “Projeto Software Livre do Rio Grande do Norte” (PSL-RN), onde colabora na disseminação do software livre em seus aspectos técnicos, sociais e legais. É sócio da ONG “Associação Software Livre” (ASL.org) desde 2006. Em seu doutorado, colabora no desenvolvimento de 3 projetos de software livre: Analizo, Kalibro e Mezuro. Tem artigos publicados em conferências nacionais e internacionais e ministrou cursos e várias palestras na área de software livre em diversos eventos nacionais.