



Brazil

Open source

Avi Alkalay

O primeiro a propor a idéia de *Open Source Software* (OSS) foi Richard Stallman na década de 1970, que a formalizou, com a ajuda de alguns advogados, definindo a modalidade de licenciamento chamada de *General Public License* (GPL). Ninguém se interessou ou sequer ouviu falar sobre isso, até que em meados da década de 1990, tudo mudou com a grande popularização do Linux, sistema operacional *Open Source*.

A tradução correta para *Open Source Software* é Software de Código Fonte Aberto (e não “software livre”). É importante destacar isso, pois muitas vezes o termo é associado à idéia de software não-proprietário, ou não-comercial. De fato, um software pode ter seu código fonte aberto, mas ser comercial, ou ter código fonte fechado e ser de distribuição gratuita. Código aberto e de distribuição gratuita são características distintas.

A idéia do OSS é simples: um desenvolvedor escreve um programa e outro pode copiá-lo à vontade sem sequer notificar o primeiro. Ele pode, inclusive, modificá-lo e redistribuí-lo, contanto que também mantenha suas modificações abertas e informe a origem e os autores anteriores.

Isso não quer dizer que haverá diversas versões desconexas do mesmo software. Cada modificação passa por um processo estruturado de aceitação, onde boas melhorias retornam à base e são incorporadas à nova versão do software. Atualmente, muitas das contribuições são feitas por empresas de tecnologia (não somente por indivíduos).

É comum – e incorreto – afirmar que OSS represente o fim de todo software de código fechado. Isso não acontece, devido à tendência de que inovações continuem a ser abordadas em um modelo fechado. Imagine um mundo hipotético que ainda não conhece editores de planilhas. É natural que, ao lançar esse produto, seu inventor opte pelo modelo de código fonte fechado, para maximizar seus lucros através do total controle de sua invenção.

Contudo, à medida que essa invenção se populariza e cria um mercado de usuários e potenciais concorrentes, a abordagem em modelo OSS surge como uma das formas – a mais inovadora – para repensá-la. OSS inova ao reimplementar o que outros inventaram e exploraram anteriormente. Recentemente, porém, a indústria começou a lançar alguns

produtos baseados em modelo OSS, justamente pelo seu poder de agregar comunidades e criar ecossistemas.

Outro conceito infundado é acreditar que quando o software em si é gratuito, eliminam-se por completo os gastos. Sempre haverá a necessidade de um suporte confiável. O modelo OSS desloca o eixo do valor agregado do software, movendo-o do software em si (geralmente sem custo de licença), para o serviço de suporte.

No processo de amadurecimento de software, a diferença básica entre OSS e software de código fonte fechado está no fato de que um fornecedor de software no modelo comercial terá que criar estrutura e suporte regional antes de vender seu

produto. Já no OSS, ofertas de suporte só surgirão (espontaneamente) depois de ele atingir uma boa gama de usuários. Entretanto, seja qual for a ordem, um dos principais fatores para garantir maturidade a qualquer software ou produto é um ciclo de desenvolvimento – uso – suporte, que estimule mais desenvolvimento. Somente essa maturidade garante a aceitação do produto em escala corporativa. Exemplos bem conhecidos são OSSs

como Linux, Apache HTTP Server, OpenOffice.org, Samba, e outros que já usufruem desse processo cíclico de uma forma global e corporativa.

Hoje, OSS tem aplicações mais maduras em infra-estrutura e em alguns nichos de middleware. Por sua vez, os softwares de código fonte fechado apresentam maior desenvoltura mercadológica nas funcionalidades de maior valor agregado ao negócio (ERPs, CRMs ou processos empresariais). Isso porque essas funcionalidades têm uma amplitude menor de usuários, o que inviabiliza o surgimento de suporte espontâneo – fator vital para a maturidade do OSS.

Portanto, devemos usar código fonte aberto ou fechado? O ideal é pensar e insistir sempre que possível no uso de padrões abertos – mais do que se preocupar só com o código – e buscar um balanço saudável entre ambos. Os padrões abertos garantem interoperabilidade entre camadas de aplicações, e ampliam as escolhas, liberdade, e por conseqüência, poder de negociação de uma empresa para compor a melhor combinação madura no momento.

