

# Software Livre: Uma Visão Crítica

Hugo Alves Richard

## Resumo

O artigo discutirá algumas características importantes de aplicativos disponibilizados ao público por meio de licenças Open Source; a ideia não é deificar o software livre, mas mostrar, principalmente, as suas fraquezas, apresentando formas de se melhorar o processo de desenvolvimento e, conseqüentemente, o produto final.

O mundo do Software Livre (SL) ainda é inacessível para muitas pessoas. Vários são os motivos que contribuem para isso. Porém, nota-se que o maior dos problemas advém de sua própria Filosofia.

Pode-se dizer, de forma resumida, que Software Livre é todo software que dá ao usuário a liberdade de execução, cópia, distribuição, estudo, modificação e melhoria. Existe um documento disponível no site da Free Software Foundation (FSF) intitulado “The Free Software Definition”, que mostra exatamente quais são os requisitos básicos para que um software seja considerado SL. A ideia principal é que “livre” não se refere a ser gratuito, mas sim a permitir os usuários terem as liberdades citadas anteriormente.

Neste contexto, para facilitar a vida de quem desenvolve e distribui Software Livre, foram criadas as famosas licenças Open Source (OS), responsáveis por reger o modo como o software poderá ser modificado e distribuído por outras pessoas. Como exemplo, pode-se citar: a “GNU General Public License version 2” (GNUv2) e a “GNU General Public License version 3” (GNUv3), que são as licenças OS mais conhecidas, criadas e mantidas pela FSF; a “Apache License version 2”, que é a licença usada na distribuição dos softwares mantidos pela Apache Software Foundation; e a “BSD License”, que foi criada pela Universidade da Califórnia, em Berkley, e é utilizada na distribuição de vários sistemas e aplicativos desenvolvidos por ela.

Grande parte dos aplicativos disponibilizados publicamente, principalmente aqueles que acompanham o sistema operacional GNU/Linux, são distribuídos como Softwares Livres. Nesse sentido, eles seguem a premissa de liberdade mencionada anteriormente, adotam uma licença Open Source para regular esta liberdade e, frequentemente, são disponibilizados gratuitamente na Internet. Até aí, tudo bem. O problema é que se por um lado a liberdade é importante para o mundo do SL, por outro ela é responsável por provocar algumas aberrações.

Aqueles que usam SL a mais tempo, provavelmente já viram sistemas promissores aparecerem e, de uma hora para outra, serem deixados de lado, ficando estagnados até desaparecerem. Ou então, programas que foram lançados com tantos bugs que era quase impossível utilizá-los por falta de confiança. Ou ainda, para piorar, presenciaram a distribuição de sistemas inconsistentes, em que o manual não servia para nada e o usuário tinha que aprender a usá-lo na base do “tapa na cara”.

Essas coisas, infelizmente, ainda acontecem com muita frequência – e, provavelmente, não deixarão de existir. O que as Comunidades de Software Livre podem fazer é minimizar o impacto que esses problemas provocam no usuário leigo que está tentando um primeiro contato com SL.

Vejam bem, tentar iludir os usuários com promessas falsas é a pior estratégia que se pode adotar nesse sentido. O usuário tem que saber que os Softwares Livres em geral não são perfeitos. Apenas dizer, por exemplo, que o GNU/Linux é o melhor sistema operacional e que todas as pessoas deveriam utilizá-lo, pode gerar uma falsa expectativa e atrapalhar bastante a adaptação do usuário ao novo sistema.

As Comunidades de Software Livre também podem investir em estratégias mais técnicas. Por exemplo, a forma distribuída é a mais comum no desenvolvimento de SL – ou seja, pessoas ao redor do mundo tomam parte na tarefa de desenvolver o software. No entanto, este tipo de

desenvolvimento nem sempre é bem organizado. A adoção de modernas técnicas de Engenharia de Software, unidas a uma centralização formal nos indivíduos mais capacitados dentro da equipe de desenvolvimento, resolveriam problemas de sincronização de versões, união de esforços, desenvolvimento dentro de prazos preestabelecidos, sistematização de rotinas de teste e correção de bugs.

Além disso, é de suma importância que as equipes de desenvolvimento tenham integrantes capacitados em Engenharia de Usabilidade. Não é atoa que esta ciência tem crescido bastante nos últimos anos. As equipes precisam entender as vontades dos usuários, padronizar comportamentos e signos, documentar o funcionamento de forma precisa, sempre visando a melhoria contínua das interfaces.

Outra medida importante a ser adotada pelas Comunidades de Software Live é a tentativa de integração de esforços – ao contrário da criação de bifurcações, os famosos forks, que são muito comuns no mundo do SL. Isso porque, quanto mais pessoas interessadas e integradas ao desenvolvimento de uma versão específica de um software, maior é a chance de sucesso. Apesar dos forks terem a sua utilidade em algumas circunstâncias, principalmente na implementação de soluções criativas e na melhoria de alguns aspectos do software, é fato que o excesso deles prejudica bastante o desenvolvimento como um todo. Se todos os projetos de SL bem sucedidos tivessem a ajuda dos desenvolvedores de seus forks, talvez eles fossem bem melhores do que são atualmente e, conseqüentemente, conquistariam um número maior de usuários. Com um número maior de usuários, a tendência de participação de novos indivíduos no processo de desenvolvimento também aumentaria e, com isso, os ciclos de desenvolvimento ficariam cada vez melhores e em maior harmonia com as vontades dos usuários.

Em suma, Softwares Livres possuem fraquezas. As Comunidades precisam atentar para isso e atacar da melhor forma possível o problema. A chave para a solução está no processo de melhoria contínua da qualidade, por meio de processos de desenvolvimento mais organizados e sólidos, preocupação permanente com a Usabilidade, necessidade de integração de esforços e estímulo à contribuição voluntária de todos os usuários.

## **Referências**

Free Software Foundation: <http://www.fsf.org/>

The Free Software Definition: <http://www.gnu.org/philosophy/free-sw.html>

Open Source Initiative – Licences by Name: <http://www.opensource.org/licenses/alphabetical>

BSD Licences: [http://en.wikipedia.org/wiki/BSD\\_licenses](http://en.wikipedia.org/wiki/BSD_licenses)

Thomas, M. P. “Why Free Software has poor usability, and how to improve it”

<http://mpt.net.nz/archive/2008/08/01/free-software-usability>