Are large software companies bound to adopt the free software model?

Merwan Achibet

Université du Havre

Contents

Abstract

abstract

Keywords

Free software, open source, proprietary software, software companies

Introduction

BLABLABLABLABLA

1 Context

Before pondering about the interests that large software companies might have for the free software model, we have to clearly define it. The concept has a large spectrum of significance and definitions are many, often separated by subtle differences. Nevertheless, they all share the same basis [1]. Firstly, the intrinsec quality of this kind of software comes from its openness: its source code is available for all, and above all, editable by all. There are thus no restriction on the amount of hacking that such a software can endure, which is a good thing, as the next section will demonstrate. Secondly, the available source code is candidate to redistribution, as long as the new software perpetuate the original licence. Lastly, a free software can be used by all and for any means. A differencation must be made between free software and open source. A free software is by definition open-source, but the reciprocal isn't inevitably true. For example, the Android operating system for mobile phone is open-source, everybody can consult its source code and use it as an inspiration. But some part of it are still proprietary.

Free software is not a new trend by any means. In fact, free software was the first distribution model for softwares, even if it didn't have this name yet. In the 50s, when computers where huge contraptions reserved to laboratories, notably academic ones, the sharing of source code was the norm, as the common goal was knowledge discovery and the advancement of science. Two

decades lates, when computers started to appear in corporate environments and later in homes, softwares rapidly became commercial products. Even if free software is still discrete compared to its commercial counterpart, there recently have been a rise in such projects, with the help of code sharing initiatives and the participation of several large companies.

2 The theory: Is free software viable?

A company delivering free softwares automatically benefits from a boost in reputation, as with proprietary software and closedness come a sense of secret. Many companies, Microsoft for example, are labeled as evil because of this model. An other advantage is that FS prompts trust from the users. If they have a doubt about the security of their banking data dor example, or the way their password is encrypted, they can check by themselves. With free softwares, openness is admitted. Thus there is far less interest in patenting the software, nor in budgeting potential legal conflicts. Of course, a major advantage of FS is user contribution. Any programmer around the world can fix any bug he stumbles upon and even add missing features, and all that on a voluntary basis. If we put aside the financial savings that such a practice permits, the major upside comes from the quality boost that a well-controled software could benefit from as the users are very well-placed to know what is wrong with a software and thus improve it.

The shortcoming of all these advantages is that they only concern experimented users, familiar with the computer and its world, whereas laymans will prefer a less intimidating solution. We can see here one of the downfalls

of the free —free as in no money needed —software. Some unexperimented users will choose a proprietary solution because a bigger price tag means more quality ("How can this free stuff be as good as the 200$ one?"). This psychological effect, associated with the public lack of knowledge about free software, is the main brake to free software expansion. The main problem that large software companies reproach to this model is that it does not permit them to hide some parts of a software, especially the more innovative ones. This is the reason why a number of software corporations choose PS over FS: they would have to share their most ingenious algorithms and revolutionizing ideas.

Free software seems to be a smart choice, given the savings and user participation it furnishes. But we have to keep in mind that the goal of a company is to generate revenue and preferably, earnings. Of course the obvious solution is to sell the software, as contrary to popular beliefs, free software can be sold and Richard Stallman even encourage it CITATION NEEDED We present three different businness models compatible with free software.

Companies such as Ubuntu use a support business model. Ther main product, the Ubuntu operating system, is free of charge. They however need infrastructures to ??? 20 million users and money to pay their 400 employees. Browsing through the Ubuntu website, we remark several possible sources of income. The more visible is an online merchandise store which sells mugs and wearable items sporting the Ubuntu logo but this store is more of a marketing tool than a true mean of revenue. No, its true business model is to freely give the tools and to sell the knowledge. Unbuntu staff comprehends

the developer of its products but also a lot of trainers, learning to users the ins and outs of Ubuntu and helping other corporations to deploy Ubuntu. This kind of technical support is not meant for individuals but rather for corporations wanting to install Ubuntu on all their computers and to train employees.

The Mozilla Corporation has chosen another path. Its main product, the Firefox internet browset, is financed by donations but this part of their revenue only accounts for about 5% of their total incomes. The real revenue source is a partnership with Google. In Firefox's interface, a small search field can be seen at the top of the window. Any words the user will input there will be processed through Google Search. There are other available search engines, but by default Google is used. This choice is not random and actually comes from a contract between Mozilla and the search giant. This partnership grants Mozilla with 84% of their annual revenue ().

Another browser example is Chromium, the internet browser from Google. Chromium started as Chrome, an open-source browser which contained a small part of proprietary modules. Following the community pression, Google forked the project and created Chromium, the Chrome free counterpart. An interest fact about Chromium and Chrome is that they generate zero revenue. This is due to the fact that Chrome is not Google's real product: Google sells targetted advertising space and display textual ads related to the search results of an user. By offering a fast and free user, they let people access their real mean of revenue more easily.

3 The pratice: ???

Each one of the largest software companies publicly endorses free software. They all possess one or several pages on their websites to explain how they are dedicated to the free software movement in this era of global sharing. The subtlety being that their definitions differ greatly. We can cast software businesses in three categories.

First, there are the reluctant ones, like Microsoft. Even if they display a public attachment to openness, few are their projects really following the FSF guidelines. At best, one could hope that they would document their software to permit operability with other softwares, potentially free. In general, they don't hesitate to inflict lawsuits to free software equivalents of their products.

Then we have the conflicted ones. We can cite Apple and Oracle as examples. Their products are typically based on free softwares and they participate to the glocal FS effort while defending aggressively their core products. Apple is a good example because while it suffers from a bad reputation in the open source world due to the closedness of their OSs and their hardware, they are surprisingly blablabla

Lastly, some companies chose to embrace the free software movement. Ubuntu and Red Hat respectively offer and sell free software and sustain their financial health by selling support and services. Intel and Google create free softwares related to their main products in order to create a sane and accessible software environement around them.

Conclusion

BLABKLABKLABKABAK

Glossary

Free software a

Open source a

GNU a

Free Software Foundation a

Richard Stallman a

Sources

References

[1] Richard Stallman. *Selected Essays of Richard Stallman.* GNU Press.