Are Large Software Companies Bound to Adopt the Free Software Model?

Merwan Achibet – Université du Havre

Abstract

With an ever-increasing computer accessibility and a huge available market, large software companies typically release proprietary softwares, but with the growth of the free and open-source movement, we have to ask ourselves if this situation is bound to change. An overview of the advantages of free software and a presentation of several business models compatible with free software shows that it has the capability to attract software companies. Even if free software and proprietary software are meant to cohabit without one model overthrowing the other, we observe a clear increase in the number of companies involved with free projects and we explain that the main brake to this growth comes from the intrinsic openness of free software which does not let companies hide their innovation from competitors.

Keywords

Free software, open source, proprietary software, software companies

Contents

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 [4]. Firstly, the intrinsic 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. Secondly, the available source code is candidate to redistribution, as long as the new software perpetuate the original licence. 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 and thus it does not strictly follow the Free Software Foundation guidelines [5]. It is also important to understant that free software isn't necesarily free of charge; the *free* of *free software* rather conveys an idea of freedom and sharing.

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 the discovery of new knowledge and the advancement of science. Two decades later, when computers started to appear in corporate environments and later in homes, softwares rapidly became commercial

products. Even if free software remains relatively unnoticeable compared to its commercial counterpart, notably because of a lack of advertising, 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?

We are going to analyze the advantages of disadvantages of free software from the perspective of a software company.

Firstly, 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 they favour proprietary software. An other advantage is that FS prompts trust from 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 by exploring the source code. These two advantages are related to the users, and they obviously have a positive impact on a software company as happy users means more users.

There also are financial advantages that come with free software. As openness is admittd, there is less need in patenting softwares. In the same way, less money will be spent in potential legal conflicts [1]. This does not mean that free software garantees legal security but due to less restrictive licences, lawsuits are less enclined to happen.

Of course, a major advantage of free software 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 gain as users are well-placed to know what is wrong with a given software and thus they know how to improve it.

Free software is not perfect and suffers some disadvantages. The shortcoming of all the previous facts is that they only relate to 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 free software. Some unexperimented users will choose a proprietary solution because a bigger price tag means more quality. For example, a user having to pick a new operating operating system and being suggested Ubuntu, a free OS costing zero dollar, and Windows 7, a proprietary one costing around 200 dollar, is more likely to chose Windows because a higher price gives the idea that the expensive product can only be better. This psychological effect, associated with the public lack of knowledge about free software, is a significant brake to the expansion of free software. The main issue 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. The real matter is that their code is visible to everyone, especially competitors. 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 features.

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, if possible, profits. Of course the obvious

solution is to sell the software (contrary to popular beliefs, free software can be sold and Richard Stallman even encourage it [6]). We present here three different businness models compatible with free software.

Companies such as Ubuntu use a support business model. Its main product, the Ubuntu operating system, is free of charge. It however needs infrastructures to distribute it to 20 million users and money to pay its 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 a mere marketing tool and cannot guarantee financial security. No, its true business model is to freely give the product 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 services is not targetted to individuals but rather for corporations wanting to install Ubuntu on all their computers in order to save money.

The Mozilla Corporation has chosen another path. Its main product, the Firefox internet browser, 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 that a user will put there is to be processed through the Google search engine. 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 granted Mozilla with 84% of their annual revenue of

approximately 103M dollar in 2009 [2].

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 demands, Google forked the project and created Chromium, the free counterpart to Chrome. An interesting 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 internet browser to users, they let people access their true source of revenue more easily [3].

3 The pratice: What is really happening

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 [7]. 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 active behind the scene. They are for example the main contributors to the Webkit project, a rendering engine used in their Safari browser, but also in Chrome since it was released as a free software.

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

We cannot realistically expect free software to fully overthrow proprietary software as some large companies are still agressively reluctant to give a try to this model. They chose this path not because of an imaginary corporate evil imagery (???) but rather because they refuse to display their innovation at the sight of all, notably at the sight of competitors.

Even if believing that FS will become ??? remains utopic, it is important to note that a growing number of free projects come to light and that several large companies are following this trend by involving users in the making of their softwares.

Glossary

Free software

Software that everyone is free to copy, redistribute and modify. That implies free software must be available as source code, hence "free open source software" - "FOSS". It is usually also free of charge, though anyone can sell free software so long as they don't impose any new restrictions on its redistribution or use — dictionnary.reference.com

Logiciel que chacun est libre de copier, redistribuer et modifier. Cela signifie qu'un logiciel libre doit être disponible sous forme de code source, d'où l'appellation de *logiciel libre et open source*. Il est habituellement gratuit, cependant n'importe qui peut vendre un logiciel libre tant qu'aucune nouvelle restriction d'utilisation et de redistribution ne sont imposées.

Open source

Generically, open source refers to a program in which the source code is available to the general public for use and/or modification from its original design free of charge — Webopedia.com

Généralement, le terme open source fait référence à un logiciel donc le source code est délivré au public pour l'utiliser et/ou le modifier et ce, de façon gratuite.

Proprietary software

Refers to any computer software that has restrictions on any combination

of the usage, modification, copying or distributing modified versions of the software. Proprietary software may also be called closed-source software — Webopedia.com

Fait référence à tout logiciel informatique soumis à des restrictions quant à son usage, sa modification, la copie ou la distribution de versions modifiées ou toute combinaison de ces actions. Les logiciels propriétaires peuvent aussi être appelés *closed source* (en opposition à *open source*).

Operating system

Software that controls the operation of a computer and directs the processing of programs (as by assigning storage space in memory and controlling input and output functions) — Merriam-Webster

Logiciel contrôlant les opérations d'un ordinateur et dirigeant les calculs nécessaires aux programmes (en assignant de l'espace de stockage en mémoire et en contrôlant les fonctions d'entrée et de sortie par exemple).

Browser

A computer program used for accessing sites or information on a network (as the World Wide Web) — Merriam-Webster

Un programme informatique utilisé pour accéder à des sites ou des information sur un réseau (le réseau Internet par exemple).

Sources

References

[1] Anonymous. Advantages of free software in embedded systems. http://free-electrons.com/doc/reasons.pdf, sep 2009.

[2] Mozilla Foundation. The state of mozilla. http://www.mozilla.org/foundation/annualreport/2009, nov 2010.

[3] Mark Shuttleworth. Mark shuttleworth on companies and free software. http://lwn.net/Articles/442782/, may 2011.

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

[5] Richard Stallman. Is android really free software? http://www.guardian.co.uk/technology/2011/sep/19/android-free-software-stallman, sep 2011.

[6] Richard Stallman. Selling free software. http://www.gnu.org/philosophy/selling.html, sep 2011.

[7] David Wheeler. Why open source software / free software (oss/fs, floss, or foss)? http://www.dwheeler.com/ossfswhy.html, apr 2007.