Open source for entrepreneurs

In recent years, open source software has become quite popular. More and more people are using open source software for qualitative, financial, or idealistic reasons. It has also evolved and matured as a business opportunity. For example, Red Hat, the most popular Linux provider for companies, has 328 million dollars in the bank. And it is claimed that HP and IBM make more than a billion dollars annually on open-source-related products and activities. So open source is definitely having an impact on both users and providers.
Because open source is still quite new to most people, however, it is subject to much misunderstanding. This article tries to clear things up. First, this article focuses on the user/consumer side, not only because a company needs to focus on its (potential) customers, but because it too is a customer. The article will then deal with how open source business is different from the traditional software business, answering the number one question: “Where's the money?”

What Is “Open Source”?

Open Source (often abbreviated “OS”) is about open content. This content could be comprised of books, technical blueprints, or other subject matter, but the term OS is normally used for a specific kind of content: the source code of software.

Generally a piece of software is considered 'officially' OS when its copyright or usage license is compatible with a strict definition created by the Open Source Initiative[1]. This definition demands, among other things, that anyone is free to access, modify, and redistribute the software's source code, but that one cannot claim credit for other people's work (i.e. no plagiarism), nor can the license state different rights for different uses or users (i.e. no discrimination).

The basic concept is that by opening the source on a piece of software it evolves, cause it is then open for the world to improve. This doesn't have to be the primary incentive, nor must it always live up to this promise. The advantages and disadvantages differ on a case-to-case basis.

The term OS is often used in a much wider sense, however. People who are developing, advocating, or merely using OS software are often considered part of the so-called OS-community. Also, approaches considered good practice within this community get labeled OS, such as attempts at interoperability using open standards (such as HTML, XML, or SVG[2]) or open data.

However, ignorance or personal interest breed many incorrect claims and faulty generalizations regarding OS. A wide spectrum of examples can be found at slashdot.org, many with a negative slant towards Microsoft. Despite many fine products by MS, and apart from antipathy over Microsoft's market dominance, these comments are often founded in reasons very much related to the nature of OS, and will therefore show up frequently in my sometimes rather technical explanations.

OS Is Free

Free is an ambiguous word. While it is true that there is no usage license to pay for (or concomitant administrative hassle), unless one has the skill to do everything oneself and have a very low valuation of one's time, OS can end up costing money for training, system administration, and other support, not to mention the cost of the actual hardware to run it on. These are all factors in the so-called Total Cost of Ownership. Take into account time lost to learning and productivity aspects and one approaches a more complete financial comparison from a users' point of view.

On the freedom aspect, I'll add that there are laws that affect OS software, such as United States encryption export regulations, and the software patent regulations that are currently under heavy discussion.

Everyone Uses OS

While most people are probably unaware of it, everyone uses OS. Most of the servers that form the infrastructure of the Internet (such as nameservers, routers, etc.) run OS software, as do most webservers. Every time you use Google, you're using a system of about 100,000 computers running Linux. Although only a small percentage of computer users run Linux on their own desktop, that does not mean they are not using OS. Linux is merely the most famous OS software, far from being the only example. Additionally, commercial software often uses pieces of OS under the cover.

People Program For A Reason, Not Necessarily Direct Financial Remuneration

Everyone who helps on a OS program does so for a reason, and there are many such rationales. Programmers generally like solving problems, and if a OS program has enough promise to grow into a solution for a problem the programmer has, the programmer has motivation to help. A company that sells a product or service that uses the OS program, or is related in some other way, might employ a programmer to work on that program, or also offer a financial reward for anybody that implements a certain feature or fixes a certain bug. Or the developer may want to program for exercise, fun, ideals of sharing and openness, experience, status, or job opportunities. Nowadays, being active in open source development is a real advantage for getting a job; MySQL AB, for example, earnestly seeks to hire people active in open source[3].

When a piece of OS software is useful for enough people, programming help abounds. Mozilla, for example, has about a 1000 people making significant contributions in programming alone, and many more helping with translating, testing, and other tasks[4]. Naturally, these efforts need systems for cooperatively developing code, and running and managing these systems involves costs.

OS Is Not Difficult

A lot of OS software was built by developers for developers. The “scratch your own itch” attitude of solving problems resulted in many programs only easily used by a few specialists, not by the average user. The programmers were less concerned with the so-called eye-candy for beautiful Graphical User Interfaces than about a solid program to build on. Typing cryptic commands was often needed to operate the programs. Big configuration files full of specialized technological terms often had to be waded through to make a simple obvious change to the settings of these highly configurable programs.

This is slowly changing for the related reasons of *demand pull* and *technology push*. The available base of OS software is becoming more and more complete, so the time has come to make it more user-friendly. As this is recognized by more and more people, their interest in OS grows, and in using it, drive the demand for user-friendliness. Among the ways to achieve usability are using good defaults for the settings in general, making a nice GUI for the most obvious actions and settings, and still have the highly advanced stuff available where the specialist can find it. Another factor which makes OS user-friendly, and aids in its growth, is that you can easily give it a try; it's not an all-or-nothing commitment. You can, for example, install OpenOffice without removing Microsoft Office. In fact, there is quite a lot of quality OS software for Microsoft Windows. You can even try an entire OS operating system without installing it, by using the Knoppix distribution of Linux that runs off a CD instead of your hard drive.

Nowadays, some OS products, Mozilla for example, even get praised for being the most user-friendly product available while still maintaining popularity with developers. These examples are still a bit rare, though.

OS Is Well-Supported

The OS community has a tradition of being helpful. Traditionally, USENET and IRC were the most helpful resources, but today great resources can be found all through the Web. Either the answer to your question is already out there, or you have a good chance of getting an answer soon after you state your question. Of course, you need to be able to find your way on the Internet, but with Google this has become very easy. Quite often this help is quicker and more accurate than the helpdesk of a traditional software company.

However, lots of people and especially companies want contracts to ensure themselves of support. Often these support contracts are included in the purchase price of right-to-use licenses. Normally, though, these contracts guarantee no solution for a possible problem, but only some time-limited help. For OS software. more and more companies and organizations (Red Hat, Mozilla Europe, etc.) have arisen which offer support contracts. And because the source is open, the number of possible experts is not limited up-front.

OS Is Highly Compatible

OS products, or the OS community, favor open formats for reasons of interoperability and re-use. Some traditional vendors use closed, proprietary formats to create “vendor lock-in.” Vendor lock-in is a term meaning the use of a program that puts user data in a closed format that forces that user to keep using (and paying for) the product, thereby being a sure cashcow for the vendor. If the vendor ceases to exist and/or the product gets cancelled, the user loses support and upgrades and their data is orphaned, maybe even inaccessible. Vendor lock-in only works for companies who produce a product with a big enough market-share, like Autodesk's AutoCAD or Microsoft Office. A closed format can mean a definite advantage of interoperability within a suite of products for its vendor, or a disadvantage for its competitors. By reverse engineering closed formats, those formats can still be used outside the originating program. OpenOffice, for example, does a fairly good job at working with Microsoft Office files. And Wine, a program which allows the operation of DOS and MS Windows programs on Linux, makes life easier for non-Windows users. But of course, guessing a format is beat by knowing it.

Formats stand the risk of being hijacked. Microsoft uses the “embrace, extend, exterminate” technique to accomplish this. Its “save as HTML” function, for example, would be better named “save for Microsoft Internet Explorer.”

OS Is Secure

Some say that when the source is open, everybody can see how to abuse the code, therefore it is inherently insecure. Others say, because everybody can see potential security-problems, such problems get fixed quickly.

When looking at the security records of both OS and non-OS products, OS doesn't look inherently insecure. One example is that new viruses, worms, or hacking-exploits often bring down millions of Windows machines; Linux suffers from this a lot less often and a lot less severely. The market share of Windows makes it a popular target for hackers, of course, but this is unlikely to be the only reason for the difference in security-records.

OS Is Changing Business

Because OS has a growing role in software, it is steadily changing the software business. Most of the revenue in the software business comes from custom-made software, the rest from so called 'shrink-wrapped' software. Companies selling this 'shrink-wrapped' software will be affected by OS most. When there are a lot of customers for a certain product, it means there are a lot of people interested in an OS-alternative, meaning high chances for its development. Companies that now have a monopoly position will have an added risk of getting competition that they are not used to. Unhappy customers previously with no other options are anxious to switch to alternative products, thus programmers have extra motivation to develop it. Microsoft, with its monopoly of Windows, Office, and Internet Explorer, and its questionable business methods, is a major 'target'.

Most people use Microsoft Windows, often because everybody does; sometimes, they are unaware that alternatives exist, since Windows usually comes pre-installed on their machine. MS has allocated loads of money to marketing, including spreading Fear, Uncertainty and Doubt about competitors' products. Microsoft is even deliberately breaking the law nowadays to maintain their monopoly position, as it seems to outweigh the enormous amounts of money it pays for lawyers and fines. Microsoft has already been fined for overpricing, trying to hijack Java, and many more offenses. Microsoft is starting to be called to task more often lately, however, as the next 2 examples show. Ernie Ball, the world's leading maker of premium guitar strings, and previously a very happy MS customer, felt mistreated and swore off MS[5], to find out that went a lot easier than he expected. Complete governments about to move from MS to OS products, were offered drastically lowered prices by MS to stay with their products, with mixed results (so even fans of MS products can profit in a way).

Where do all these changes leave the software entrepreneur?

Entrepreneuring is about being pro-active, getting things done, all to reach a certain goal.

This goal is usually money, just as MS, but doesn't have to be. Particularly in the software business, one must be aware of possible competition from not-for-profit or non-profit organizations.

So far, among many other things, this article has addressed some possible expenses and savings related with OS. But we also need income, so finally we've come to the number one question.....

Where's The Money?

Most code doesn't have real value in itself. The business built around software focuses on the added value. Most of the revenue from conventional software comes from surrounding services and custom-made software. Open source gets basically all the revenue from these, with the surrounding services maybe being even more diverse.

It might sound strange that giving away the 'crown jewels' can actually be a smart thing to do from a business perspective. But often it is, working as a marketing tool, for example. When people can try out your product for free, they might like it. The company could get more users, and with that more customers. These customers could be potential revenue sources through support services, co-developed or custom-made adjustments, or other leveraged solutions. Using standards to provide interoperability with other software is likely to encourage the effect.

Some people are afraid the competition will steal their product or the associated business when they go open source. These risks are often very slim, however. The original developer of a product is usually considered the natural leader for an open source project. This is quite logical, as this developer knows the code, and so could easily make adjustments, give support, write manuals, etc. It would be very difficult for a competitor to become as fluent with the latest code, and --on top of that-- have the community follow. If a company has proven to be reliable, can be trusted, and is an expert, these facts are valued by its users and customers, prompting loyalty. The very fact of code being open sourced doesn't mean it's unprotected. You can't just copy it and call it your own; that would be plagiarism. And with the most common license, the General Public License, everyone making adjustments is allowed to distribute the changed software, but only under the same license (sometimes called “copyleft,” to contrast it with “copyright”). This helps the original product, and the business around it.

Both companies and individuals actually donate to open source projects. They give equipment, server-space, bandwidth, human resources or just plain money. They do this for either sheer appreciation or for strategic reasons. The simplest is of course that you like the product and want to ensure the continuity, that it will be further developed. Another is that by providing an open source substitute for a needed complement of your product you enable yourself to raise your prices, cause customers care about the total combined price. Another is of course to build a business around it or at least remove or prevent power from competitors.

IBM adapted Apache to prevent Microsoft from becoming a monopolist in the Web server market. It is adapting Linux to hinder Solaris and Windows. It has started the Eclipse Integrated Development Environment project as a perfect base to sell its advanced developers Rational tools in the form of plug-ins. And it just started a Web-based project to try to eat away at the MS Office monopoly. Some people fear IBM will cash in on its patent portfolio, so its OS products won't be free after all, in a tactic similar to SCO suing companies using Linux now (though it is argued SCO doesn't own the patents).

SourceForge.net, a free webspace for tens of thousands of collaborative open source projects, has recently added the option to donate money to the projects it hosts. This is bound to have its effect on popular projects (just as Mozilla.org, a not-for-profit organization and OS project, raised 100,000 USD in a year without much effort). SourceForge itself is part of the Open Source Developers Network (owned by VA Software) that makes money in several ways: selling the software that runs its Web sites, advertisements on those sites, “stuff for smart masses”-goods at thinkgeek.com, and donations.

Red Hat, sells support with its Linux distribution. Its revenue comes from media distribution, branding, training, consulting, custom development, and post-sales support.

O'Reilly is “accessorizing.” It distributes mainly books, but also T-shirts and other physical items associated with, and supportive of, open-source software.

Companies like Covalent, Lindows and 10X software offer consulting and a (complete) solution stack. Lindows sells a Linux distribution with an update subscription service. Covalent is a LAMP-specialist. LAMP being a popular solution stack for webservers: Linux, Apache, MySQL and PHP. 10X software does system integration.

Just as in closed source, open source software can have a license, and usually does. With OS there is more standardization in licensing, as the earlier mentioned GPL and other OSI-licenses, for example. But one can always design a custom licensing policy, even though it might not be OS anymore by OSI's definition. One can distinguish between certain types of users and uses, such as differentiating the rights given or sold depending on whether or not the user is commercial (to determine by the vendor), or uses a favored operating system. This is called multiple licensing: a business policy to let customers choose between available licenses.

A company might also have more than one product with a large common codebase; one product might be a closed commercial product, probably with support and the most advanced features, and the other a trailing open source version, with improvements added later than the main product. Star Office and Open Office are such a pair; notice that the common codebase doesn't allow Open Office to have the GPL. When the OS version trails too little, no one will buy the commercial version. When it trails too much it doesn't work as a marketing tool for the commercial version. So usually the question is more about when, not if, to open the source, bearing in mind that keeping a secret can be difficult.

As already mentioned, building a brand is important: it's a reputation, a valuation of whether one is trusted, whether one is an expert. It also serves as a channel to potential customers. That's why big names like Sun and HP can sell Linux certification programs.
A strong brand may even be able to earn revenue by allowing others to us its name for a franchise fee.

concluding:

“OS provides some interesting challenges and opportunities to entrepreneurs.”

[1]: quoting its website opensource.org : "OSI is a non-profit corporation dedicated to managing and promoting the Open Source Definition for the good of the community, specifically through the OSI Certified Open Source Software certification mark and program."

[2]: SVG, Scalable Vector Graphics, "the HTML of graphics", open standard by W3C.
The **University of Twente hosts SVGopen2005**, cause she considers SVG very promising:
http://svgopen.steltenpower.com

[3]: http://www.newsforge.com/business/04/05/04/1058254.shtml

[4]: http://www.pctechtalk.com/view.php?id=2908

[5]: http://news.com.com/2008-1082_3-5065859.html?tag=lh

I have read OS-related articles before, and for creating this article i specifically re-read and used:
http://www.dwheeler.com/oss_fs_why.html
http://www.opensource.org/...
http://www.hecker.org/writings/setting-up-shop.html
http://www.itmanagersjournal.com/management/04/05/10/2052216.shtml

- I wrote this article with OpenOffice and investigated the subject by surfing the web with Mozilla FireFox.

- I value good usability (Human-Computer Interaction as you yourself like it) and interoperability (to combine into powerful functionality), both as a user and a programmer and therefore am a fan of OS.

- I would like the University of Twente to stimulate the use and development of OS with an official policy, building on the example:
http://orange.math.buffalo.edu/csc/resolution2_april2003_approved.html