

# SHARING DIGITAL INFORMATION USING OPEN SOURCE

*Peter Toft*

Skåne Sjælland Linux User Group  
Email contact: [pto@sslug.dk](mailto:pto@sslug.dk)  
Homepage: <http://www.sslug.dk/~pto>

## About the author

The author is the chairman of SSLUG - Skåne Sjælland Linux User Group which is one of the largest Linux groups worldwide with more than 6000 members and a lot of activity going on. The author holds a PhD degree in signal processing, and did his post doc in the field of medical signal processing, and is currently employed at Nokia as a system design specialist. The author has written ten books and articles on various subjects, especially within the area of Linux and Open Source technology.

## Keywords

Open Source, sharing, collaboration in software development, freedom, Internet, Linux, Apache, software bugs, source code, free software, The Cathedral and the Bazaar, KDE, GNOME, Apache.

### ABSTRACT

The Internet has during the last ten years spawned many exiting collaborations between communities of people who do not know each other. Entire operating systems, high quality graphical desktops and thousands of other computer programs have been made in this way. The exiting thing is that the communities often make free software (in the meaning “free as a bird”). This kind of Open Source software is now powering large parts of the Internet infrastructure and for the ordinary computer users it is possible to get computer systems at virtually no cost.

## 1. WHY COLLABORATE?

When people want to work together it is of interest to see what will motivate them. Often an employer will pay a salary to get the person to do certain work. Most often people are paid e.g. 50 euro per hour and they are then happy to do their work. If the employer one day asks the same people to work for e.g. 5 euro per hour then it is most likely that the person will laugh and quit the job.

But in recent years lots of software have been developed by people sharing *Open Source* software legally over the Internet. Most often these people do not get 50 euro per hour, or 5 euro per hour, but do this work for free. Some even work 10-30 hours per week on these projects. What then drives these people to work for free then?

### 1.1. Why collaborate for free?

Most often the highly successful Open Source development projects start off with a good idea, such as ‘Let us make an email program with a nice graphical appearance’. Often a few people start to form either a very sketchy version of the program with very limited functionality, or they make the first written specifications of the program. This basic work is most often announced on an Internet homepage and through public newsgroups on the Internet. *If* the project seems of interest to others, then more people join in with new ideas and features to the program. People start to work together and make the project into reality.

There are often several reasons to get involved in such a project. First of all the project often agrees to license the program under an Open Source license [1], i.e. a software license, which grants the users the right to freely distribute the software as they please, they are also allowed to change the software - this is possible, since Open

Source requires access to the *source code* of the program. The program is written by a software developer in a human-readable language called the source code, and most commercial programs are only sold in binary form, i.e. transformed with a compiler to a non-readable form, which is understood only by the computer.

This principle of open access to the source code of the program and that the project do not have to be profitable most often creates an atmosphere different from the ordinary commercial way of running a company with employed people. In Open Source projects developers only stay if they have interest in the project and the direction of the project. The spirit of the development project is often very open and consensus solutions between the developers often determines the direction of the development. Hence one major reason that many of the Open Source projects are doing so well is the open spirit where the people who are leading the project are those who seem the best to lead, and people in general do what they like to do. It is to be compared with a team of people meeting on a regular basis in order to play e.g. football. If the people feel that there is a goal and a good spirit, then they like to spend the time collaborating with the others achieving a common goal.

Some will participate in Open Source projects because they like to “pay” back for all of the software they got for free made by others. Some will try to show outstanding results, which later can be used to get a well paid job. Others like to earn respect from similar developers, or make friends this way.

Another major reason for doing Open Source development is a common interest to produce high quality software together with the other people in the project while having fun - this is very well presented in the new book [2] by the famous Linus Torvalds (main-author of the computer operating system Linux).

## 1.2. Development cycles in Open Source projects

Most often a commercial software company will face trade-offs between the money spend on the project, the amount of program features, and the software quality. Most often the two last parts are the places to cut. In Open Source projects there are quite a different way to work where quality most often determines the release date.

Most often projects will work with a huge sequential set of beta-versions (i.e. versions that might not be ready for prime-time usage). If a bug is found in the latest beta-version, then a new beta-version is shipped of the software where the bug (presumably) is cured. This beta-track or development-track of the software typically has a very high development pace and it is primarily intended for developers. New versions might come daily. For the projects which have been around for some time (are stabilised) typically you also find a stable track of the software, which is intended for people who just want a version of the software which works. The release-pace of the stable versions are often low (1-10 times per year).

Being an project manager one of the best things you experience in Open Source projects are the contacts between developers and users, most often via email or news-groups on the Internet. Typically the requests you get are very open, thanking for the work you have made, and typically people are very willing to help either with removing bugs or coming up with new good features for the program.

The great thing here is that a user typically can directly contact the actual author of the software and directly exchange ideas and experiences. Especially if the software has a bug (some kind of malfunction), then for the developer it is extremely beneficial for the developer to be in direct contact with the user. Often it can be be hard to recreate a reported problem for the developer, but given the direct contact the developer and the user, bugs are often identified and solved fast.

The really great thing about Open Source is that both the developer and the user have the same source code for the program available. Often new features will be programmed by the main developer(s), but quite often some exiting features will be developed and programmed by one of the users, who will simply announce that he has made a new cool feature. This open access to the software technology often enables rapid and efficient development of software.

It can also be noted that Open Source projects need not solely be driven by interested users. Some projects are partly or mostly made by a company. Apache [3] is the web server which runs approximately 60% of all web sites on the Internet. It is driven rather professionally and commercial support is available. OST Developer Network [4] is founded and supported by Nokia in order to develop software for a commercial hardware platform. The very fast Open Source database MySQL [5] used many places on the Internet is mainly driven by the company MySQL AB with 30 people employed around the globe.

## 2. WHAT IS OPEN SOURCE?

Open Source [1] is a common denominator for a set of software licenses, which gives the users many more rights compared to ordinary software licenses.

Typically it is illegal to bring a copy of commercial software from your home to your friends home. For Open Source software the user has the right to copy the software as he or she pleases, *without* having to pay any kind of royalty.

The user is also allowed to sell or give the software away (even for free) to other people. No royalty has to be paid to the software producer if it is an open source program. In general the user only has one major obligation to the producer of the Open Source software - that the source code of the program should remain available, typically from a Internet web-site.

Furthermore the user is also granted right by the Open Source license to *modify* the program by changing the source code. Again, if we compare this to ordinary commercial programs, these rights are very powerful, since you normally do not get the source code, you are not able to change the commercial program you get.

### 2.1. Why change the program?

It might sound stupid to change a program, but most often there are several good reasons to want so.

Very often programs contain programming errors. Most of us have seen the infamous “blue screen of death” with Microsoft Windows. Microsoft computer users will hear (way to often) that some new computer virus threatens their system. These problems are due to sloppy or unsafe programming. For ordinary commercial programs typically the user has to accept programming bugs until the next release of the software - which also has to be purchased.

Open Source is typically strong regarding bugs - typically some a programmer gets upset that he or his company cannot work efficiently given that with the program “bug”, and they correct it. Especially if the bug regards network security then a corrected program version is made fast (typically faster than commercial products).

Since everyone will gain by someone fixing the bugs the bug-fixes are typically fed back to the development-project and incorporated into the next release of the software. Typically this process is very efficient and the people who correct the bugs are highly appreciated.

### 2.2. Open Source is a scientific approach to software development

The concept of sharing ideas and testing new concepts by seeking the opinion of peers is very old. For centuries scientists have been proposing new ideas and the joint peer-review have be used to distinguish good from bad.

Open Source development is typically no different from this. Ultimately the development is determined jointly by the developers and users. If many developers push for a certain development direction, then typically this will be chosen. If the majority of the developers support a new idea, then typically the originator of the idea change the source code of the program and implement the changes. Even though every Open Source developer is given the right to develop modified versions of the program it is much more likely that it is the discussions with the peers that determine the progress of Open Source programs. In this way good ideas are typically accepted and bad ones are eventually discarded. In reality you do not find thousands of different versions of the same software (which could be feared) - most people understand the value of improving a common base of the software, hence changes to the software are typically send back to the developers of the software for review and approval as described above.

## 3. DEMOCRATIC SOFTWARE

Traditionally computer software has been developed by commercial companies primarily based on marked demands. This marked structure has existed for many years and has been proved efficient. This does not mean that commercial software need to be the only software development structure. Open Source is clearly a strong alternative in some technical fields of development.

For the users of commercial software it is typically hard to influence the next versions of the software. On the other hand Open Source gives all of us a chance to influence the development. As described above typically it is possible to subscribe and contribute to the mailing lists where developers are very interested to hear from users.

Especially if you want to contribute your voice will be heard and your opinion will matter. The word “contribute” here could mean that you fix bugs, contribute with new parts of the software program, but for the non-programmers

it is actually equally important to translate menus to other languages, to write documentation, and to perform user tests with the program - three extremely important tasks that can be made by most people.

From a democratic viewpoint it is very exiting that the rather idealistic view of community developed software seems to work - and actually work very well. Every day people around the world share software, optimise the software and communicate to improve what has been made jointly.

### 3.1. Can we get software for free?

Using an Open Source license does not guarantee that the software comes for free, however, any user has the right to take the Open Source software and sell the same software of a price less than the original price. This typically means that the software can be downloaded from an Internet web server for free, and it might be that the software provider also offers another version of the software, typically on CD-ROM or DVD, which cost some money, but also comes with value added, such as access to a hot-line, books, installation guide, clip-art CD-ROM's, etc.

Let us also look at what kind of Open Source software are available today. Several computer operating systems have been made, the most well-known being Linux. Linux has been build in the best Open Source traditions on the Internet. The development started in Finland by Linus Torvalds a decade ago as a University-based project, but soon the basis of a revolution was set. Thousands of people have been working together making the best operating system ever. Many thousands of man-years of work has resulted in more that one million lines of source code, and Linux today runs on more than ten different computer processors, spanning from micro-computer devices up to full-blown clusters of thousands of computers as well as main frame computers. For Linux virtually any UNIX-software also exists such as web-, file-, and print-server software. All of the free software from GNU [6], suited for the software developer also exist for Linux.

The desktop is currently the big development area within the Open Source community, where projects like GNOME [7] and KDE [8] "compete" on developing integrated high-quality GUI desktops with all of the software-tools expected on the desktop.

Last year SUN started to release the source code for the StarOffice productivity suite to the OpenOffice project [9]. The Open Office project is currently one of the best candidates for an open user-friendly package of word-processing, spreadsheet, and presentation program (though other candidates such as KOffice [10] also are making progress). Within the next year or so it should be possible to get nearly all of the programs the average computer user needs in daily life on the desktop - for free - as Open Source software. It is amazing to follow!

### 3.2. Where is the catch?

Open Source is *not* some miracle cure for software development. In some areas of software the Open Source development strategy works, and in other areas it is a not used at all. From the famous series of essays from Eric Raymond [11] it is possible to come with some general remarks regarding the situation.

Open Source seems to work very fine in high-interest software areas, where the value of software stability and performance drives the development. On the opposite side Open Source does not seem to work for very specialised areas where people do not agree in general what to develop, or in places where the interest in the program is low, or that the value of the program is predominantly connected so some kind of secret (you cannot hide secrets in Open Source programs, since the source code must be available).

A very fine example is the company "Id Software", who made the famous computer game DOOM some years ago. When DOOM was launched the graphics engine<sup>1</sup> was much better than the competitors could make, hence Id Software had an advantage in keeping their source code closed. As time progressed several other computer games also came with the same (or even better) graphical performance, and then a large part of the company's revenue came from network gaming facilities. At that time the source code for DOOM was released, since it then meant more to have stable game programs than keeping the techniques in the source code a secret. Games likes Quake has later also been released under an Open Source license.

Back to the down-side of Open Source. There might not be an adequate Open Source program for your needs. Currently top-class CAD-programs are not available, Office-suites are under development, and games for Linux are still not quite up the standard. The best advice is to consider your software needs and then try to search for the required software e.g. on Freshmeat [12].

Another problem with Open Source software, e.g. a Linux distribution installed on your PC, is that it may require climbing a high learning curve. The easy path into a Linux system is to get someone to help you install the

---

<sup>1</sup>the part of the program what could generate the stunning graphics at that time

system, and then you should start to read documentation and try thing on your own. New Linux users will soon discover that the system is much more powerful than a Windows computer - especially in the server-area. But it will typically also mean that the user has to edit text-files in order to make a certain program run. This is often very useful for the expert user, but for the new user the start might require a helping hand from experienced users.

#### 4. HOW AND WHERE DO I START?

If you would like to start using Open Source software, then you might want to try a Linux-distribution like Mandrake [13], Red Hat [14], or SuSE [15]. All three are fine collections of the operating system Linux and a tons of Open Source software. Typically the distributions are available for download [16] or bought on CD-ROM rather cheap. Note that community-alternatives like FreeBSD [17] or Debian Linux [18] are also available.

A Linux (or the FreeBSD) distribution is typically installed on a PC, where the user gets network-, email-, web- (browser and server), as well as desktop-software. For the ordinary computer user it is recommended to start by contacting other users in a nearby Linux Linux User Group (LUG) [19] - often there is one available in your area also. Often "install-fests" are held, where it is possible to come with your computer and have a lot of help while the system is installed. Feel free to join in and get started.

#### 5. ACKNOWLEDGEMENTS

Kristian Vilmann, Jacob Sparre Andersen, and especially Jesper Laisen are thanked for corrections to the article. Private correspondence with Per Helge Srensen has also been a source of inspiration.

#### 6. REFERENCES

- [1] <http://www.opensource.org>
- [2] Linus Torvalds and David Diamond *Just for Fun*. HarperBusiness 2001. ISBN: 0066620724.
- [3] <http://www.apache.org>
- [4] <http://www.ostdev.net>
- [5] <http://www.mysql.com>
- [6] <http://www.gnu.org>
- [7] <http://www.gnome.org>
- [8] <http://www.kde.org>
- [9] <http://www.openoffice.org>
- [10] <http://www.koffice.org>
- [11] Eric S. Raymond *The Cathedral and the Bazaar*, *Homesteading the Noosphere*, and *The Magic Cauldron*.  
<http://tuxedo.org/~esr/writings/>
- [12] <http://freshmeat.net>
- [13] <http://www.mandrake.com>
- [14] <http://www.redhat.com>
- [15] <http://www.suse.com>
- [16] <http://www.linuxiso.org>
- [17] <http://www.freebsd.org>
- [18] <http://www.debian.org>
- [19] <http://lugww.counter.li.org/>