Wikipedia on Open Source Business

Chapter 1

Multi-licensing

Multi-licensing is the practice of distributing software under two or more different sets of terms and conditions. This may mean multiple different software licenses or sets of licenses. Prefixes may be used to indicate the number of licenses used, e.g. **dual-licensed** for software licensed under two different licenses.

When software is multi-licensed, recipients can choose the terms under which they want to use or distribute the software. The distributor may or may not apply a fee to either option. The two usual motivations for multi-licensing are license compatibility and market segregation based business models.

1.1 Business models

Multi-licensing is commonly done to support free software business models in a commercial environment. In this scenario, one option is a proprietary software license, which allows the possibility of creating proprietary applications derived from it, while the other license is a copyleft free software/open-source license, thus requiring any derived work to be released under the same license. The copyright holder of the software then typically provides the free version of the software at little or no cost, and profits by selling proprietary licenses to commercial operations looking to incorporate the software into their own business. This model can be compared to shareware.[1][2]

Since in most cases, only the copyright holder can change the licensing terms of a software, multi licensing is mostly used by companies that wholly own the software which they are licensing. Confusion may arise when a person outside the company creates additional source code, using the less restrictive license. Because the company with the official code is not the copyright holder of the additional code, they may not legally include this new work in their more restrictively licensed version. Companies may demand outside developers agree to a contributor license agreement, before accepting their work in the official codebase and source code repositories.[3]

Multi licensing is used by the copyright holders of some free software packages advertising their willingness to distribute using both a copyleft free software license and a non-free software license. The latter license typically offers users the software as proprietary software or offers third parties the source code without copyleft provisions. Copyright holders are exercising the monopoly they're provided under copyright in this scenario, but also use multi licensing to distinguish the rights and freedoms different recipients receive.

Such licensing allows the holder to offer customizations and early releases, generate other derivative works or grant rights to third parties to redistribute proprietary versions all while offering everyone a free version of the software. Sharing the package as copyleft free software can benefit the copyright holder by receiving contributions from users and hackers of the free software community. These contributions can be the support of a dedicated user community, word of mouth marketing or modifications that are made available as stipulated by a copyleft license. However, a copyright holder’s commitment to elude copyleft provisions and advertise proprietary redistributions risks losing confidence and support from free software users.[4][5]

Examples of multi-licensed software include Oracle's NetBeans IDE, MySQL AB's database, Asterisk, Oracle Corporation's Berkeley DB, Modelio, ZeroC's Ice, Magnolia CMS and Qt Software's Qt development toolkit.

Description on one specific example to illustrate multi-licensing: Oracle MySQL comes in various editions: MySQL Enterprise Edition[6] is a commercial edition, hence to be purchased. The license is only offered as a subscription, named MySQL Enterprise Edition Subscription. The same applies for MySQL Standard Edition (MySQL Standard Edition Subscription) and MySQL Cluster CGE (MySQL Cluster Carrier Grade Edition Subscription). The other editions, such as the MySQL Classic Edition or MySQL Community Edition, are free to use with some restrictions. For instance, the MySQL Community

Edition is a freely downloadable version, available under the GPL license and is supported by a community of open source developers.[7]

1.1.1 Single-Vendor Commercial Open Source Business Model

The term *single-vendor commercial open source* was coined by Dirk Riehle in 2010,[8][9] and has later been further popularized by other scholars, such as Simon R. B. Berdal.[10]

According to Riehle:

> *Single-vendor commercial open source firms build their business around an open source software project that they fully control, typically by having developed the software and never having shared control with third parties. This is done by owning the full copyright to the code and related intellectual property such as patents and trademarks... Typically, the free open source form is provided under a reciprocal license like the GPL to drive adoption but stall possible competitors. Paid-for versions of the software are then provided under a commercial license like traditional software vendors do. This is also known as the dual-license strategy of commercial open source.*[8]

In contrast to traditional open source projects, a Single-vendor commercial open source project *is controlled by exactly one stakeholder with the purpose of commercially exploiting it.*[8] In this context, one should note that the open source community is less engaged in the development of core functionality, as they typically are in conventional (pure) open source projects. As the then CEO Mårten Mikos of MySQL himself noted in an interview:

> *The depth of the contributions varies by product and situation. The deeper you go into the core of the database engine, the more difficult it is for somebody to contribute because it takes five years to learn. If you build something on the outskirts of the kernel - some tool or function that you add on top of it - then that is much easier because there's less risk that you will mess up the whole product. But something great can emerge out of many tiny-looking contributions. It's analogous to how, in economic development, microloans can have such a huge impact - each entry is minimal, but when you multiply it by the number of people who are involved, it grows massive. It starts getting a momentum of its own..*[11]

Hence, the community of multi-license software as a rule includes employees of the code-owning firm, as well as strategic partners that have vested interest in the software. As Riehle notes, *In single-vendor open source, almost all of the core product development work is carried out by the commercial firm, with occasional contributions from the community.*[8]

As Berdal notes, the governance of the open source community becomes *a key business management process* in this context: *As such, it needs to be aligned with other business activities. Governance models of dual-licensed OSS editions may therefore display a tendency towards commercial bias. To prevent the community from being provoked or alienated it may therefore seem imperative to balance commercial inclinations against "open" interests.*[10] This is by no means an easy task. As Berdal demonstrated through a case study of SugarCRM, this commercial open source software (COSS) business model can trigger substantial friction points, which can eventually lead to pure open source forks (table adapted from Berdal, Table 3, page 75[10]):

Interestingly, only a few months after these friction points were observed, a new fork (SuiteCRM) of the SugarCRM Community Edition was announced.

1.2 License compatibility

A second use of multi-licensing with free software is for license compatibility, allowing code from differently licensed free software projects to be combined, or to provide users the preference to pick a license.

Examples include the source code of Mozilla Application Suite and previously Mozilla Thunderbird and Mozilla Firefox, that have used tri-licensing under the Mozilla Public License (MPL) 1.1, GNU General Public License (GPL) 2.0 or GNU Lesser General Public License (LGPL) 2.1[12] before the latter upgraded to GPL-compatible MPL 2.0, making the tri-licensing unnecessary;[13] Perl, which is dual-licensed under the GPL or Artistic License;[14] and Ruby, whose license contains explicit GPL dual licensing.

1.3 Market segregation in proprietary software

Multi-licensing is also used by distributors of non-free software. Sometimes this is done to proprietary software to segregate a market. By splitting customers into multiple categories such as home users, professional users, and academic users, copyright holders can set different prices for each group. However, among proprietary software companies, it is more common to release a “home edition” and a “professional edition” of a given product, which differ by the software and software features included, not just the license.

1.4 See also

- Business models for open source software
- Commercial use of copyleft works
- Commercial open source applications
- Professional open-source

1.5 References

[1] Linux News: Tech Buzz: Dual Licensing: Having Your Cake and Eating It Too

[2] Dual-Licensing Open Source Business Models | Linux

[3] Digium Incorporated. “Asterisk Guidelines, The contributor license agreement”. Retrieved 2009-02-10.

[4] Netscape Public License - GNU Project - Free Software Foundation (FSF)

[5] FSF’s Opinion on the Apple Public Source License (APSL) - GNU Project - Free Software Foundation (FSF)

[6] “My SQL Enterprise Edition”. Oracle. Retrieved 25 April 2013.

[7] “MySQL Community Edition”. Oracle, MySQL. Retrieved 25 April 2013.

[8] *The Single-Vendor Commercial Open Source Business Model*, 2010-11-09, retrieved 8 December 2013

[9] Riehle, Dirk (March 2012). “The single-vendor commercial open source business model.” **10** (1). Information Systems and e-Business Management. pp. 5–17. doi:10.1007/s10257-010-0149-x.

[10] Berdal, S.R.B. (January 2013). “Peculiarities of the Commercial Open Source business model: Case study of SugarCRM”. Tronheim, Norway: Norwegian University of Science and Technology, Faculty of Social Sciences and Technology Management, Department of Industrial Economics and Technology Management.

[11] “The Oh-So-Practical Magic of Open-Source Innovation”. *MIT Sloan Management Review* **50** (1). October 1, 2008. Retrieved 8 December 2013.

[12] Mozilla Foundation. “Mozilla Code Licensing”. Retrieved 2007-09-17.

[13] “MPL 2 Upgrade”. Retrieved 2012-08-18.

[14] The Perl Foundation. “Perl Licensing - perl.org”. Retrieved 2007-09-17.

1.6 External links

- Dual Licensing information from OSS Watch
- Article "The Dual-Licensing Model" May 1, 2002 by Don Marti
- Article "Dual Licensing: Having Your Cake and Eating It Too" Nov 16 2004 by Philip H. Albert
- Article "Does dual licensing threaten free software?" Jul 27 2006 by Glyn Moody
- Article "Dual-Licensing Open Source Business Models" by Heather Meeker
- Paper "Dual Licensing in Open Source Software Industry" by Mikko Välimäki
- Dual Licensing Schemes
- Open Source Business Models leveraging multi-licensing
- "Should code be “dual licensed” under the GPL and a permissive license?" by Software Freedom Law Center
- Combining GPL, closed code
- Glossary of Software Licensing

Chapter 2

Professional open source

For a broader coverage related to this topic, see Business models for open-source software.

Professional open source is an open-source software business model where an open-source software vendor generates revenue from paid professional services, maintenance and support provided along with the software.[1] Some open-source software vendors also provide commercial licenses of open-source software or customized versions of open-source software to specific customers[2] .[3] Professional open source is more commonly used in the business environment rather than for individual use.[4] *"The more mission-critical the open source software, the more necessary it is to acquire paid support"*, suggested SUSE's Gerald Pfeifer. *"Individual users will often tough out solving problems through community help forums, but SMB owners and enterprise users more likely will opt for paid support rather than devoting internal resources to support open source software,"* he added.[5]

2.1 History

As open-source software began to become popular in the 1990s with the introduction of Linux, business hesitated to adopt it because of fears that no single entity guaranteed its stability or support.[6] As a consequence, larger businesses would often choose commercially distributed software over a product that was released under an open-source license. However, there has been much growth in the number of professional open-source companies – made popular by companies like Liferay, Inc., eXo Platform, OpenSearchServer, Red Hat, MySQL AB, and JBoss. The business model of these companies tries "to offer open-source software with a free license, while using professional services, maintenance and support for these products to derive revenue."[7]

2.2 Professional Services for Business Use

Under the category of professional open source, there are many different types of support available to users. Some may need assistance with their personal use of particular open source software, in which case there are options such as forums, live chat rooms and video tutorials available for free. In the case of business use however, there are more paid professional services available.

2.2.1 Technical Support

Technical support is one of the professional services available to users of open source software, this type of support can include a wide range of services from bug repairs to simple usage guidance. The case with most major open source software packages for business use is that the vendor of the software gives clients the option of commercial support in addition to the free software.[8] Businesses can pay for this type of service on a contractual basis with regular monthly payments. Alternatively, they can receive technical support through independent firms.[9]

2.2.2 Software training classes and workshops

In order for business operations to run efficiently, it is important for members of staff, who will be using the open source software, to be very familiar with the functionalities of the software. Businesses in this situation are likely to benefit from organising for their staff to be trained before they use the software. This can be carried out by sending staff to training classes and workshops provided by independent firms such as Open Logic.[9] To prevent any interruptions of work, similar training is available online through a virtual learning environment provided by firms such as CBT Nuggets.[10]

2.2.3 Assistance with customisation

A lot of companies using open source software for business are using it due to the convenient ability they have to adapt it to their own business needs. *"A starting point for modifying open source applications is taking an open source solution that comes close to your needs and adding your own modifications from there"*, suggested Mac McConnell, vice president of Bonitasoft. *"Companies that still do their own software have very specific needs. Most companies today want to start with open source foundation and then customize from there."*[11] This may, however, not be possible without the assistance of someone with high technical knowledge of customizing the source code. To overcome this problem, businesses can hire professional assistance from the software developers such as QAT Global.[12]

2.3 Approaches

A dominant factor contributing to the preference of proprietary software as opposed to open-source software is the common misconception that there is more support available for the former than the latter. Developers of open-source software are under no licensing agreement to provide any technical support for their software, nor is any particular organisation responsible for doing so.[13] However, professional open source provides technical support for users through one of the following two approaches and is a prime example of how users can counter this issue.

2.3.1 Part of the package

Certain vendors of open source software such as Red Hat include technical support as part of the package they provide.[14] Staying with Red Hat as an example, they offer three levels of support for their version of enterprise Linux, the first of which is self-support which is simply the open source software with no support for the user. Secondly, they offer a standard level which includes support via the Internet and telephone for an unlimited number of cases during business hours. Red Hat also offer a premium level of support which would be provided at a much higher price but would provide support to the user 24 hours a day, 7 days a week for severe cases of technical difficulties. Red Hat is not the only company providing support as part of the software package, other companies with similar strategies include Canonical, AT&T, Rackspace, SUSE, HP, IBM, Nebula and many more.[15]

2.3.2 Optional extra support

Another approach which is used by open source software vendors when providing users of their software with technical support is to provide it as an optional extra. An example of a company offering this service is Ubuntu, which provides an open source operating system for a range of devices.[16] Customers also have the option of subscribing to Ubuntu Advantage, which is their professional support package.[17]

2.4 Open Source Benefits

As the industry grows, the amount of open source software available is increasing. Of the present day, there are free alternative (open source) options available for almost every type of proprietary software.[18] As well as the wide range of availability, there are several reasons why an individual user or a business model may benefit from using open source software instead of expensive proprietary options.

The number of software bugs found in the source code of open source software is gradually decreasing,[19] and in some cases it has been found that proprietary software contains more bugs than open source software which has been built for the same purpose.[20] A common reason for this is the fact that the source code is editable by anyone who wants access to it, therefore software developers are able to remove any bugs and errors as soon as they come across them.

The obvious advantage of using open source software in business models is the monetary savings that can be made through doing so. As there are no direct costs involved with using open source software, it can be said that businesses can save fortunes through not having to pay large amounts of money for costly proprietary software. In 2008 the Standish Group reported a drop in revenue of $60 billion in the propriety software industry due to the adoption of open source software.[21] This also suggests that users had saved a combined $60 billion through using open source software instead of proprietary software.

A feature of open source software which has made it more favourable than proprietary software is the ability to modify the code to suit the needs of a particular user. An example of this was when Linux was used by NASA for their opsLAN system. *"We migrated key functions from Windows to Linux because we needed an operating system that was stable and reliable – one that would give us in-house control. So if we needed to patch, adjust or adapt, we could,"* said Keith Chuvala of the United Space Alliance.[22]

2.5 Reasons for Favouring Proprietary Software

Alternatively to professional open source software, businesses may look into proprietary versions. There are several reasons why this may be a better option despite the mentioned benefits of open source software.

Open source software systems are not sold to generate profit, therefore they tend not to be particularly well suited for all end users and are often developed to suit highly technical users with similar IT backgrounds as the developer of the software. Therefore factors such as ease of use and cleanliness of the user interface are often overlooked.[23]

External support for open source software often comes at a cost, whereas some proprietary software packages are provided with free technical support. Similar support for open source software is available but very scarce and, when found, likely to require a monetary payment.

Despite the fact that the source code being open to the public means that users can eliminate bugs or generally improve the software, it also means that malicious users can access the code and, in theory, infect the source code with further bugs and errors.

Ultimately the businesses needs and current affairs would need to be assessed in order to determine whether proprietary software or open source software would be the most beneficial.

2.6 Alternative Support

As well as the support offered from the vendors of professional open source software packages, there are alternative methods of receiving support for open source software.

Free communal support is available on-line for most open source software packages. Mozilla Firefox, for instance, offers a service called ‘Army of Awesome’.[24] This support system allows users to post their queries to Mozilla through Twitter, the queries can then be answered by contributors. Other packages such as Linux have on-line forums dedicated to individuals and small businesses. Similarly to ‘Army of Awesome’, Linux users can post their queries to the dedicated forum and wait for a response from another member of the forum.[25]

Independent IT consultant firms can also provide technical support for certain open source software. Some consultant firms, such as Unicon[26] and OpenLogic[27] specialise solely on open source software and can provide assistance with a range of technical difficulties. Some firms will charge a fee for this service whereas others provide the service for free.

2.7 See also

- Dual licensing
- Vertical integration
- Commercial use of copyleft works

2.8 References

[1] Karl M. Popp (2015). *Best Practices for commercial use of open source software*. Norderstedt, Germany: BOD. ISBN 978-3738619096.

[2] Karl M. Popp and Ralf Meyer (2010). *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry*. Norderstedt, Germany: BOD. ISBN 3-8391-6983-6.

[3] Richard T. Watson, Donald Wynn and Marie-Claude Boudreau (September 2005). “JBoss: the evolution of professional open source software”. *MIS Quarterly Executive Vol. 4 No. 3*.

[4] Ivelin Ivanov (May 2015). “The beauty of professional open source software collaboration”. *Telestax, Inc. company blog*. Retrieved 21 May 2015.

[5] Germain, Jack M. “FOSS in the Enterprise: To Pay or Not to Pay?". *Linux Insider*. Linux Insider. Retrieved 21 October 2014.

[6] Osterwalder, Alexander; Pigneur, Yves (23 July 2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. John Wiley & Sons. p. 103. ISBN 978-0-470-90103-8. Retrieved 27 September 2013.

[7] Montalbano, Elizabeth (February 16, 2006). “OSBC: Professional open source grows up”. InfoWorld.

[8] “Services that make the most of Ubuntu”. *Canonical*. Retrieved 21 October 2014.

[9] “Enterprise Technical Support for Open Source Software Packages”. *Open Logic*. Retrieved 21 October 2014.

[10] “Linux”. *CBT Nuggets*. Retrieved 21 October 2014.

[11] Germain, Jack M. (19 November 2013). “Want Customization With That?". Linux Insider. Retrieved 21 October 2014.

[12] “Open Source Software Customization”. *QAT*. Retrieved 21 October 2014.

[13] Czerkawski, Betul (2011). *Free and Open Source Software for E-learning*. p. 29. Retrieved 16 October 2014.

[14] "Production Support Service Level Agreement". *Red Hat*. Retrieved 16 October 2014.

[15] "About Ubuntu". *Ubuntu.com*. Retrieved 16 October 2014.

[16] "Meet Ubuntu". *Ubuntu*. Retrieved 16 October 2014.

[17] "Systems management and support". *Ubuntu*. Retrieved 16 October 2014.

[18] Harvey, Cynthia. "77 Open Source Replacements for Expensive Applications". *Datamation*. Datamation. Retrieved 21 October 2014.

[19] Kanaracus, Chris (19 October 2009). "Fewer Bugs Found in Open-Source Code". Computer World. Retrieved 16 October 2014.

[20] Handy, Alex (24 February 2012). "Study shows open-source code more bug-free than proprietary". SD Times. Retrieved 16 October 2014.

[21] Rothwell, Richard (5 August 2008). "Creating wealth with free software". Free Software Magazine. Retrieved 16 October 2014.

[22] Gunter, Joel (10 May 2013). "International Space Station to boldly go with Linux over Windows". The Telegraph. Retrieved 16 October 2014.

[23] Golden, Bernard (2004). *Succeeding with open source*. p. 31. Retrieved 16 October 2014.

[24] "Mozilla Support". *Support.Mozilla.org*. Retrieved 16 October 2014.

[25] "ToolBox". *Linux Toolbox*. Retrieved 16 October 2014.

[26] "Our Work". *Unicon*. Retrieved 16 October 2014.

[27] "Open Source Support". *Open Logic*. Retrieved 16 October 2014.

Chapter 3

Open-source bounty

An **open-source bounty** is a monetary reward for completing a task in an open-source software project.

3.1 Description

Bounties are usually offered as an incentive for fixing software bugs or implementing minor features. Bounty driven development is one of the Business models for open-source software. The compensation offered for an open-source bounty is usually small. Bountysource is a service that advertises bounties from multiple open-source projects.

3.2 Alternatives

When open-source projects require bigger funds they usually apply for grants or, most recently, launch crowdsourcing or crowdfunding campaigns, typically organized over platforms like Kickstarter[1] or Bountysource (since 2004 also crowdfunding[2]).

3.3 Examples

- Sun MicroSystems (now owned by Oracle Corporation) has offered $1 million in bounties for OpenSolaris, NetBeans, OpenSPARC, Project GlassFish, OpenOffice.org, and OpenJDK.[3]
- Mozilla introduced a Security Bug Bounty Program, offering $500 to anyone who finds a "critical" security bug in Mozilla.[4]
- Artifex Software offers[5] up to $1000 to anyone who fixes some of the issues posted on Ghostscript Bugzilla.
- Two software bounties were completed for the classic Commodore Amiga Motorola 680x0 version of the AROS operating system, producing a free Kickstart ROM replacement for use with the UAE emulator and FPGA Amiga reimplementations, as well as original Amiga hardware.[6][7]
- RISC OS Open bounty scheme to encourage development of RISC OS[8]
- AmiZilla was an over $11,000 bounty to port the Firefox web-browser to AmigaOS, MorphOS & AROS. While the bounty produced little results it inspired many bounty systems in the Amiga community including Timberwolf, Power2people, AROS Bounties, Amigabounty.net and many more.

3.4 See also

- Business models for open-source software
- Reverse bounty
- Crowdfunding

3.5 References

[1] Lunduke, Bryan (2013-08-07). "Open source gets its own crowd-funding site, with bounties included - Bountysource is the crowd-funding site the open source community has been waiting for.". networkworld.com. Retrieved 2013-08-10. *Many open source projects (from phones to programming tools) have taken to crowd-funding sites (such as Kickstarter and indiegogo) in order to raise the cash needed for large-scale development. And, in some cases, this has worked out quite well.*

[2] "Bountysource Raises $1.1 Million for the First Crowdfunding Platform for Open-Source Software Projects". finance.yahoo.com. 2013-07-16. Retrieved 2013-08-08.

[3] *Sun Sponsors Open Source Community $1M Innovation Award*, Sun MicroSystems, archived from the original on 2008-12-19

[4] Leyden, John (2004-08-03), *Mozilla to pay bounty on bugs*, The Register

[5] “Ghostscript: Bug bounty program”. Retrieved 14 July 2015.

[6] “Amiga.org - Kickstart ROM Replacement (Phase II) Assigned”. Retrieved 14 July 2015.

[7] “Kickstart ROM Replacement (Phase II)". Retrieved 14 July 2015.

[8] “RISC OS Open: All bounties”. Retrieved 14 July 2015.

Chapter 4

Open core

This article is about a software business model. For open source hardware cores, see OpenCores.

The term **Open core** is a type of business model for open-source software, i.e., a business model involving the monetization of free and open-source software. The term was coined by Andrew Lampitt in 2008.[1] This model primarily involves offering a “core” or feature limited version of a software product as free and open-source software, while offering “commercial” versions or add-ons as proprietary software, or offering other services for the open source version in a similar manner.[2][3]

The concept of open core software has proven to be controversial, as many developers do not consider the business model to be true open source software. Despite this, open core models are used by a large number of open source software companies.[4]

4.1 Examples

- MySQL (Support and licensing)
- Rapid7, with its Metasploit Pro and Metasploit Express products which are based on the open source Metasploit core framework.
- Talend uses an Open Code license for its data management, data integration and cloud products.

It has been suggested by Bradley M. Kuhn that Canonical Ltd., maintainers of Ubuntu are preparing to go open core, but have not done so,[5] according to the Canonical’s contributor agreement and their Project Harmony which aimed “to assist organisations which use contribution agreements by providing standardised variable templates with clear and concise explanations....”[6]

4.2 Assignment with protections against open core

Some open core models use copyright assignment, but it should be pointed out that some open source projects require assignment of copyright for the sole purpose of defending that copyright, with the promise of retaining (only) open source licensing. For example, by prosecuting modification and binary release of GPLed software without release of the modified source code.

Organisations which see open core as a danger include clauses in their assignment to prohibit open core licensing. One example is Free Software Foundation Europe's (FSFE) Fiduciary Licence Agreement (used by KDE[7]). In this agreement, developers assign copyright to FSFE, but FSFE promises to use a free software licence when distributing the software:

> FSFE shall only exercise the granted rights and licences in accordance with the principles of Free Software as defined by the Free Software Foundations. FSFE guarantees to use the rights and licences transferred in strict accordance with the regulations imposed by Free Software licences, including, but not limited to, the GNU General Public Licence (GPL) or the GNU Lesser General Public Licence (LGPL) respectively. In the event FSFE violates the principles of Free Software, all granted rights and licences shall automatically return to the Beneficiary and the licences granted hereunder shall be terminated and expire.[8]

Other projects that use copyright assignments but which promise not to distribute the software as open core include the GNU Project.[9]

4.3 See also

- Crippleware
- Freemium

4.4 External links

- Bradley M. Kuhn (2009-10-16). ""Open Core" Is the New Shareware".
- Simon Phipps (2010-06-29). "Open Core Is Bad For You". ComputerWorldUK.
- Brian Prentice (2010-03-31). "Open-Core: The Emperor's New Clothes". Gartner.

4.5 References

[1] Phipps, Simon (July 2012). *Open Source Strategies for the Enterprise*. O'Reilly Media. ISBN 978-1-4493-4117-6.

[2] Riehle, Dirk (2009). "The Commercial Open Source Business Model". *Value Creation in e-Business Management*. Springer Verlag. pp. 18–30.

[3] Wasserman, Anthony I. (2011). "How the Internet transformed the software industry". *Journal of Internet Services and Applications* **2** (1): 11–22. doi:10.1007/s13174-011-0019-x. ISSN 1867-4828. Retrieved 25 January 2015. Some companies have only a single version of their software, while others follow an "open core" model, providing a community release of the core version, and offering proprietary premium features us- ing a commercial license.

[4] "Open Core Debate: The Battle for a Business Model". Linux Insider. Retrieved 28 March 2013.

[5] "Canonical, Ltd. Finally On Record: Seeking Open Core - Bradley M. Kuhn (Brad) (bkuhn)". Ebb.org. Retrieved 2012-04-02.

[6] Amanda Brock (2010-06-24). "Project Harmony looks to improve contribution agreements".

[7] "FSFE welcomes KDE's adoption of the Fiduciary Licence Agreement (FLA)". 2008-08-22.

[8] "Fiduciary Licence Agreement (Version 1.2)" (PDF). Retrieved 2010-12-31.

[9] "6.1 Copyright Papers". Retrieved 2011-01-03.

Chapter 5

Open-source economics

Open-source economics is an economic platform based on open collaboration for the production of software, services, or other products.

First applied to the open-source software industry,[1] this economic model may be applied to a wide range of enterprises.

Some characteristics of open-source economics may include: work or investment is carried out without express expectation of return; products or services are produced through collaboration between users and developers; there is no direct individual ownership of the enterprise itself.

In the context of open source hardware design, digital designs are shared for free and then anyone with access to digital manufacturing technologies (e.g. RepRap 3-D printers) can replicate the product for the cost of materials. The original sharer gains feedback and potentially improvements on the original design from the peer production community. There is now significant evidence that such sharing creates enormous value.[2]

As of recently there were no known commercial organizations outside of software that employ open-source economics as a structural base.[3] Today there are organizations that provide services and products, or at least instructions for building such services or products, that use an open-source economic model.[4][5]

The structure of open source is based on user participation. “networked environment makes possible a new modality of organizing production: radically decentralized, collaborative, and non-proprietary; based on sharing resources and outputs among widely distributed, loosely connected individuals who cooperate with each other without relying on either market signals or managerial commands.”[6]

5.1 See also

- Free and open-source software
- Business models for open-source software
- Open Source Ecology
- Commercial use of copyleft works
- Commons-based peer production

5.2 References

[1] Josh Lerner, Jean Tirole. “Some Simple Economics of Open Source”. *The Journal of Industrial Economics*. Retrieved 23 February 2012.

[2] Pearce, J.M. (2015) Quantifying the Value of Open Source Hardware Development. Modern Economy, 6, 1-11. doi: 10.4236/me.2015.61001. open access to the full text

[3] Josh Lerner, Jean Tirole. “THE ECONOMICS OF TECHNOLOGY SHARING: OPEN SOURCE AND BEYOND”. *Page 28*. NATIONAL BUREAU OF ECONOMIC RESEARCH. Retrieved 24 February 2012.

[4] Global Village Construction Set. “Open source ecology”. *Industrial Machines*. Retrieved 24 February 2012.

[5] Arduino. “Open-source electronics”. Retrieved 24 February 2012.

[6] Yochai, Benkler. “The Wealth of Networks”. *Chapter 3*. Yale University Press. Retrieved 24 February 2012.

5.3 External links

- Yochai Benkler on the new open-source economics

Chapter 6

Business models for open-source software

Not to be confused with Business applications of open-source software.

Open-source software is widely used both as independent applications and as components in non-open-source applications. Many independent software vendors (ISVs), value-added resellers (VARs), and hardware vendors (OEMs or ODMs) use open-source frameworks, modules, and libraries inside their proprietary, for-profit products and services.[1] From the customer's perspective, the ability to use open-source technology under standard commercial terms and support is valuable. Customers are willing to pay for the legal protection (e.g., indemnification from intellectual property infringement), "commercial-grade QA", and professional support/training/consulting that are typical of commercial software, while also receiving the benefits of fine-grained control and lack of lock-in that come with open source.

6.1 Introduction

Unlike proprietary off-the-shelf software, which comes with restrictive copyright licenses, open source software can be given away for no charge. This means that its creators cannot require each user to pay a license fee to fund development. Instead, a number of alternative models for funding its development have emerged.

Software can be developed as a consulting project for one or more customers. The customers pay to direct the developers' efforts: to have bugs prioritized and fixed or features added. Companies or independent consultants can also charge for training, installation, technical support, or customization of the software.

Another approach to funding is to provide the software freely, but sell licenses to proprietary add-ons such as data libraries. For instance, an open-source CAD program may require parts libraries which are sold on a subscription or flat-fee basis. Open-source software can also promote the sale of specialized hardware that it interoperates with. Some example cases are the Asterisk telephony software developed by a manufacturer of PC telephony hardware Digium, or the Robot Operating System (ROS) robotics platform by Willow Garage and Stanford AI Labs.

Many open source software projects have begun as research projects within universities, as personal projects of students or professors, or as tools to aid scientific research. The influence of universities and research institutions on open source shows in the number of projects named after their host institutions, such as BSD Unix, CMU Common Lisp, or the NCSA HTTPd which evolved into Apache.

Companies may employ developers to work on open-source projects that are useful to the company's infrastructure: in this case, it is developed not as a product to be sold but as a sort of shared public utility. A local bug-fix or solution to a software problem, written by a developer either at a company's request or to make his/her own job easier, can be released as an open-source contribution without costing the company anything.[2] A larger project such as the Linux kernel may have contributors from dozens of companies which use and depend upon it, as well as hobbyist and research developers.

Also, there exists stipends to support the development of open source software like Google's Summer of Code founded 2005.[3]

A new funding approach for open source projects is crowdfunding, organized over web platforms like Kickstarter, Indiegogo, or Bountysource.[4]

6.2 Challenges

Open-source software can be sold and used in general commercially. Also, commercial open-source applications are a part of the software industry for some time.[1][5] Despite that, except for Red Hat and VA Software, no other pure open-source company has gone public on the major stock markets. While commercialization or funding of open-source software projects is possible, it is considered challenging.[6]

Since several open-source licenses stipulate that derived works must distribute their intellectual property under an open-source (copyleft) license, ISVs and VARs have to develop new legal and technical mechanisms to foster their commercial goals,[7] as many traditional mechanisms are not directly applicable anymore.

Traditional business wisdom suggests that a company's methods, assets, and intellectual properties should remain concealed from market competitors as long as possible to maximize the profitable commercialization time of a new product. Open-source software development minimizes the effectiveness of this tactic; development of the product is usually performed in view of the public, allowing competing projects or clones to incorporate new features or improvements as soon as the public code repository is updated, as permitted by most open-source licenses. Also in the computer hardware domain, a hardware producer who provides free and open software drivers reveals the knowledge about hardware implementation details to competitors, who might use this knowledge to catch up.

Therefore, there is considerable debate about whether vendors can make a sustainable business from an open-source strategy. In terms of a traditional software company, this is probably the wrong question to ask. Looking at the landscape of open source applications, many of the larger ones are sponsored (and largely written) by system companies such as IBM who may not have an objective of software license revenues. Other software companies, such as Oracle and Google, have sponsored or delivered significant open-source code bases. These firms' motivation tends to be more strategic, in the sense that they are trying to change the rules of a marketplace and reduce the influence of vendors such as Microsoft. Smaller vendors doing open-source work may be less concerned with immediate revenue growth than developing a large and loyal community, which may be the basis of a corporate valuation at merger time.

A variety of open-source compatible business approaches have gained prominence in recent years; notable examples include dual licensing, software as a service, not charging for the software but for services, freemium, donation-based funding, and crowdfunding (see the Approaches section, below).

The underlying objective of these business models is to harness the size and international scope of the open-source community (typically more than an order of magnitude larger than what would be achieved with closed-source models) for a sustainable commercial venture. The vast majority of commercial open-source companies experience a conversion ratio (as measured by the percentage of downloaders who buy something) well below 1%, so low-cost and highly-scalable marketing and sales functions are key to these firms' profitability.

6.3 Approaches

There are several different types of business models for making profit using open-source software (OSS) or funding the creation. Below are existing and legal commercial business approaches in context of open-source software and open-source licenses.[8] The acceptance of these approaches varies; some of these approaches are recommended (like selling services), others are accepted, while still others are considered controversial or even unethical by the open-source community.

6.3.1 Dual-licensing

Main article: Multi-licensing

Dual licensing offers the software under an open-source license but also under separate proprietary license terms. The proprietary version can be sold to finance the continued development of the free open-source version.[7] Customers can be attracted to a no-cost and open-source edition, then be part of an up-sell to a commercial enterprise edition. Further, customers will learn of open-source software in a company's portfolio and offerings but generate business in other proprietary products and solutions, including commercial technical support contracts and services. A popular example is Oracle's MySQL database which is dual-licensed under a commercial proprietary license as also under the GPLv2.[9] Another example is the Sleepycat License.

6.3.2 Selling professional services

Main article: Professional open source

The financial return of costs on open-source software can also come from selling services, such as training, technical support, or consulting, rather than the software itself.[10][11]

Another possibility is offering open-source software in source code form only, while providing executable binaries to paying customers only, offering the commercial service of compiling and packaging of the software. Also, providing goods like physical installation media (e.g., DVDs) can be a commercial service.

Open-source companies using this business model successfully are for instance RedHat and IBM;[12] a more specialized example is that of Revolution Analytics.

6.3.3 Selling of branded merchandise

Some open-source organizations such as the Mozilla Foundation[13] and the Wikimedia Foundation[14] sell branded merchandise articles like t-shirts and coffee mugs. This can be also seen as an additional service provided to the user community.

6.3.4 Selling of certificates and trademark use

Another financing approach is innovated by Moodle, an open source learning management system and community platform.[15][16] The business model revolves around a network of commercial partners[17] who are certificated and therefore authorised to use the Moodle name and logo,[18] and in turn provide a proportion of revenue to the Moodle Trust, which in turn funds core development.[19]

6.3.5 Selling software as a service

Selling subscriptions for online accounts and server access to customers is a way of making profit based on open-source software. Also, combining desktop software with a service, called software plus services. Providing cloud computing services or software as a service (SaaS) without the release of the open-source software itself, neither in binary nor in source form, conforms with most open-source licenses (with exception of the AGPL).

Because of its lack of software freedoms, Richard Stallman calls SaaS “inherently bad” while acknowledging its legality.[20][21] The FSF called the server-side use-case without release of the source-code the *ASP loophole in the GPLv2* and encourage therefore the use of the Affero General Public License which plugged this hole in 2002.[22][23] In 2007 the FSF contemplated including the special provision of AGPLv1 into GPLv3 but ultimately decided to keep the licenses separate.[24]

6.3.6 Partnership with funding organizations

Other financial situations include partnerships with other companies. Governments, universities, companies, and non-governmental organizations may develop internally or hire a contractor for custom in-house modifications, then release that code under an open-source license. Some organizations support the development of open-source software by grants or stipends, like Google’s Summer of Code initiative founded in 2005.[3]

6.3.7 Voluntary donations

Main article: Donationware

Also, there were experiments by Independent developers to fund development of open-source software donation-driven directly by the users, e.g., with the Illumination Software Creator in 2012.[25] SourceForge, for example, allows users to donate money to the projects it hosts that opt to accept donations.[26] Internet micro-payments systems like PayPal, flattr, and Bitcoin help this approach.

Larger donation campaigns also exist. In 2004 the Mozilla Foundation carried out a fundraising campaign to support the launch of the Firefox 1.0 web browser. It placed a two-page ad in the December 16 edition of the *New York Times* listing the names of the thousands who had donated.[27][28]

6.3.8 Bounty driven development

Main article: Open-source bounty

The users of a particular software artifact may come together and pool money into an open-source bounty for the implementation of a desired feature or functionality. Offering bounties as funding has existed for some time. For instance, Bountysource is a web platform which has offered this funding model for open source software since 2003.

Another bounty source is companies or foundations that set up bounty programs for implemented features or bugfixes in open-source software relevant to them. For instance, Mozilla has been paying and funding freelance open-source programmers for security bug hunting and fixing since 2004.[29][30][31]

6.3.9 Pre-order/crowdfunding/reverse-bounty model

A newer funding opportunity for open-source software projects is crowdfunding, which shares similarities with the pre-order or Praenumeration business model, as well as the reverse bounty model. It is typically organized over web platforms like Kickstarter,[32] Indiegogo,[33] or Bountysource[4] (see also comparison of crowd funding services). An example is a successfully funded Indiegogo campaign of Australian programmer Timothy Arceri, who offered for $2,500 to implement in two weeks an OpenGL 4.3 extension for the Mesa library.[33] Arceri delivered the OpenGL extension code, which got merged into Mesa, and continued later his Mesa work with successive crowdfunding campaigns.[34]

6.3.10 Advertising-supported software

In order to commercialize FOSS, many companies (including Google, Mozilla, and Canonical) have moved towards an economic model of advertising-supported software. For instance, the open-source application AdBlock Plus gets paid by Google for letting whitelisted *Acceptable Ads* bypassing the browser ad remover.[35] As another example is SourceForge, an open-source project service provider, has the revenue model of advertising banner sales on their website. In 2006, SourceForge reported quarterly takings of $6.5 million[36] and $23 million in 2009.[37]

6.3.11 Selling of optional proprietary extensions

Main article: Open core

Some companies sell proprietary but optional extensions, modules, plugins or add-ons to an open-source software product. This can be a “license conform” approach with many open-source licenses if done technically sufficiently carefully. For instance, mixing proprietary code and open-source licensed code in statically linked libraries[38] or compiling all source code together in a software product might violate open-source licenses, while keeping them separated by interfaces and dynamic-link libraries might often adhere to license conform.

This approach is a variant of the freemium business model. The proprietary software may be intended to let customers get more value out of their data, infrastructure, or platform, e.g., operate their infrastructure/platform more effectively and efficiently, manage it better, or secure it better. Examples include the IBM proprietary Linux software, where IBM contributes to the Linux open-source ecosystem, but it builds and delivers (to IBM’s paying customers) database software, middleware, and other software that runs on top of the open-source core. Other examples of proprietary products built on open-source software include Red Hat Enterprise Linux and Cloudera's Apache Hadoop-based software. Some companies appear to re-invest a portion of their financial profits from the sale of proprietary software back into the open source infrastructure.[39]

Some companies, such as Digium, sell proprietary but optional digital electronics hardware controlled by an open-source software product.[40]

6.3.12 Selling of required proprietary parts of a software product

A variant of the approach above is the keeping of required data content (for instance a video game's audio, graphic, and other art assets) of a software product proprietary while making the software’s source code open-source. While this approach is completely legitimate and compatible with most open-source licenses, customers have to buy the content to have a complete and working software product.[41] Restrictive licenses can then be applied on the content, which prevents the redistribution or re-selling of the complete software product. An example is Kot-in-Action Creative Artel video game *Steel Storm*, where the engine is licensed as GPLv2 while the artwork is CC-BY-NC-SA 3.0 licensed.[42] Doing so conforms with the FSF and Richard Stallman, who stated that for art or entertainment the software freedoms are not required or important.[43]

The similar product bundling of an open-source software product with a proprietary hardware part is called tivoization and legal with most open-source licenses except GPLv3, which explicitly prohibits this use-case.[44]

6.3.13 Re-licensing under a proprietary license

If a software product uses only own software and open-source software under a permissive free software licence, a company can re-license the resulting software product under a proprietary license and sell the product without the source code or software freedoms. For instance, Apple Inc. is an avid user of this approach by using source code and software from open-source projects. For example, the BSD Unix operating system kernel (under the BSD license) was used in Apple’s Mac PCs that were sold as proprietary products.[45]

6.3.14 Obfuscation of source code

An approach to allow commercialization under some open-source licenses while still protecting crucial business secrets, intellectual property and technical know-how is obfuscation of source code. This approach was used in several cases, for instance by Nvidia in their open-source graphic card device drivers.[46] This practise is used to get the open-source-friendly propaganda without bearing the inconveniences. There has been debate in the free-software/open-source community on whether it is illegal to skirt copyleft software licenses by releasing source code in obfuscated form, such as in cases in which the author is less willing to make the source code available. The general consensus was that while unethical, it was not considered a violation.

The Free Software Foundation, on the other hand, is clearly against this practice.[47] The GNU General Public License since version 2 has defined "source code" as "the preferred form of the work for making modifications to it." This is intended to prevent the release of obfuscated source code.[48]

6.3.15 Delayed open-sourcing

Some companies provide the latest version available only to paying customers. A vendor forks a non-copyleft software project then adds closed-source additions to it and sells the resulting software. After a fixed time period the patches are released back upstream under the same license as the rest of the codebase. This business model is called version lagging or time delaying.[39][49]

An extreme variant of "time-delayed open-sourcing" is a business practice popularized by Id Software[50][51] and 3D Realms,[52][53] which released several software products under a free software license after a long proprietary commercialization time period and the return of investment was achieved. The motivation of companies following this practice of releasing the source code when a software reaches the commercial end-of-life, is to prevent that their software becomes unsupported Abandonware or even get lost due to digital obsolescence.[54] This gives the user communities the chance to continue development and support of the software product themselves as an open-source software project.[55] Many examples from the video game domain are in the list of commercial video games with later released source code.

Popular non-game software examples are the Netscape Communicator which was open-sourced in 1998[56][57] and Sun Microsystems's office suite, StarOffice, which was released in October 2000 at its commercial end of life.[58] Both releases formed the basis of important open-source projects, namely the Mozilla Firefox and OpenOffice.org/LibreOffice.

This approach works only with own source code or with software under specific open-source licenses, namely the permissive licences, as there is no copy-left license available which allows the opening of source code in a defined delayed time-window after distributing or selling of a software product.

6.4 FOSS and economy

Main article: Open-source economics

According to Yochai Benkler, the Berkman Professor for Entrepreneurial Legal Studies at Harvard Law School, free software is the most visible part of a new economy of commons-based peer production of information, knowledge, and culture. As examples, he cites a variety of FOSS projects, including both free software and open source.[59]

This new economy is already under development. In order to commercialize FOSS, many companies, Google being the most successful, are moving towards an economic model of advertising-supported software. In such a model, the only way to increase revenue is to make the advertising more valuable. Facebook has recently come under fire for using novel user tracking methods to accomplish this.[60]

This new economy is not without alternatives. Apple's App Stores have proven very popular with both users and developers. The Free Software Foundation considers Apple's App Stores to be incompatible with its GPL and complained that Apple was infringing on the GPL with its iTunes terms of use. Rather than change those terms to comply with the GPL, Apple removed the GPL-licensed products from its App Stores.[61] The authors of VLC, one of the GPL-licensed programs at the center of those complaints, recently began the process to switch from the GPL to the LGPL.[62]

6.5 Examples

Main article: List of commercial open-source applications and services

Much of the Internet runs on open-source software tools and utilities such as Linux, Apache, MySQL, and PHP, known as the LAMP stack for web servers. Using open source appeals to software developers for three main reasons: low or no cost, access to source code they can tailor themselves, and a shared community that ensures a generally robust code base, with quick fixes for new issues.

Despite doing much business in proprietary software, some companies like Oracle Corporation and IBM participated in developing free and open-source software to deter from monopolies and take a portion of market share for themselves. See Commercial open-source applications for the list of current commercial open-source offerings. Netscape's actions were an example of this, and thus Mozilla Firefox has become more popular, getting market share from Internet Explorer.[63]

- Active Agenda is offered for free, but requires all extensions to be shared back with the world community. The project sells a “Non-Reciprocal Private License” to anyone interested in keeping module extensions private.
- Adobe Systems offers Flex for free, while selling the Flash Builder IDE.
- Apple Inc. offers Darwin for free, while selling Mac OS X.
- Asterisk (PBX), digital electronics hardware controlled by open-source software
- Codeweavers sells CrossOver commercially, deriving it from the free Wine project they also back.
- Canonical Ltd. offers Ubuntu for free, while they sell commercial technical support contracts.
- Cloudera's Apache Hadoop-based software.
- Francisco Burzi offers PHP-Nuke for free, but the latest version is offered commercially.
- DaDaBIK, although following a donationware approach, requires a small, minimum donation fee, to be downloaded.
- IBM proprietary Linux software, where IBM delivers database software, middleware and other software.
- Ingres is offered for free, but services and support are offered as a subscription. The Ingres Icebreaker Appliance is also offered as a commercial database appliance.
- id Software releases their legacy game engines under the GPL, while retaining proprietary ownership on their latest incarnation.
- Mozilla Foundation have a partnership with Google and other companies which provides revenue for inclusion of search engines in Mozilla Firefox.
- MySQL is offered for free, but with the enterprise version includes support and additional features.
- Novell offers openSUSE for free through the openSUSE Project, while selling SUSE Linux Enterprise (SLE).
- OpenSearchServer offers its community edition on SourceForge and an enterprise edition with professional services to enterprises with a paid license
- Oracle - VirtualBox is free and open to anyone, but the VirtualBox extension pack can only be used for free at home, thus requiring payment from business users
- Red Hat sells support subscriptions for Red Hat Enterprise Linux (RHEL) which is an enterprise distribution periodically forked from the community-developed Fedora.
- Sourcefire offers Snort for free, while selling Sourcefire 3D.
- Sun Microsystems (acquired by Oracle in 2010) once offered OpenOffice.org for free, while selling StarOffice
- Untangle provides its Lite Package for free, while selling its Standard and Premium Packages by subscription
- Zend Technologies offers Zend Server CE and Zend Framework for free, but sells Zend Server with support and additional features.

6.6 See also

- Free software business model
- Open Source Development Labs
- Commercial use of copyleft works
- Open business
- Open innovation

6.7 Further reading

- Open Source Business Models, Open Source Business Models at Open source best practices
- Seven Open Source Business Strategies for Commercial Advantage, John C. Koenig (2004) PDF
- Free-Libre / Open Source Software (FLOSS) is Commercial Software, David A. Wheeler
- Selling Free Software, Free Software Foundation
- The Emerging Economic Paradigm of Open Source - Bruce Perens (3 October 2005)
- Economic aspects and business models of Free Software - Free Technology Academy (2010)
- FOSS Business Models in Developing countries - compilation of material and sources (2011, archived)
- From Open Source to long-term sustainability: Review of Business Models and Case studies Chang, Victor, Mills, Hugo and Newhouse, Steven (2007)

- The Business and Economics of Linux and Open Source (2003) by Martin Fink (ISBN 0-13-047677-3)
- The Five Open Source Business Models on informationweek.com
- Opensourcestrategies.org collection of articles
- openbusiness.cc platform to share and develop innovative Open Business ideas (2007, archived)
- Open source business models, Chapter 2 in Succeeding with Open Source, Addison-Wesley, 242 pp. by Bernard Golden (2005)
- Wikinomics: How Mass Collaboration Changes Everything by Don Tapscott and Anthony D. Williams

6.8 References

[1] Karl M. Popp and Ralf Meyer (2010). *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry*. Norderstedt, Germany: BOD. ISBN 3-8391-6983-6.

[2] Holtgrewe, Ursula (2004). "Articulating the Speed(s) of the Internet: The Case of Open Source/Free Software.". *Time & Society* **13**: 129–146. doi:10.1177/0961463X04040750.

[3] Byfield, Bruce (2005-09-21). "Google's Summer of Code concludes (first year)". linux.com. Retrieved 2013-08-08. *Google's Summer of Code (SOC), a program that matched computer science students with free and open source software (FOSS) projects and paid for results, is over.*

[4] "Bountysource Raises $1.1 Million for the First Crowdfunding Platform for Open-Source Software Projects". finance.yahoo.com. 2013-07-16. Retrieved 2013-08-08.

[5] Wheeler, David A. (February 2009). "F/LOSS is Commercial Software". Technology Innovation Management Review. Retrieved 2013-08-08.

[6] "Interview with Richard Stallman". *GNU/LAS s20e10*. Linux action show. 2012-03-11. Retrieved 2014-08-22. RMS: *I'm not gone to claim that I got a way to make it easier to raise money to pay people who write free software. We all know, that to some extent there are ways to do that, but we all know that they are limited, they are not as broad as we would like.*

[7] Karl M. Popp (2015). *Best Practices for commercial use of open source software*. Norderstedt, Germany: BOD. ISBN 978-3738619096.

[8] Karl M. Popp (2015). *Best Practices for commercial use of open source software*. Norderstedt, Germany: BOD. ISBN 978-3738619096.

[9] "Commercial License for OEMs, ISVs and VARs". mysql.com. 2010-07-01. Retrieved 2013-08-10. *Q4: What is Oracle's dual license model for MySQL software? A: Oracle makes its MySQL database server and MySQL Client Libraries available under both the GPL and a commercial license. As a result, developers who use or distribute open source applications under the GPL can use the GPL-licensed MySQL software, and OEMs, ISVs and VARs that do not want to combine or distribute the MySQL software with their own commercial software under a GPL license can purchase a commercial license.*

[10] FOSS in the Enterprise: To Pay or Not to Pay? on linuxinsider.com By Jack M. Germain (Nov 5, 2013)

[11] 6 Reasons to Pay for Open Source Software By Paul Rubens on CIO "Open source software is free to download, modify and use, but that doesn't mean it's not worth paying for sometimes. If you're using open source software in a commercial, enterprise capacity, here are six reasons why you should pay for free software." (Feb 13, 2013)

[12] McMillan, Robert (2012-03-28). "Red Hat Becomes Open Source's First $1 Billion Baby". wired.com. Retrieved 2013-08-12. *Other companies have made big money selling Linux — Intel, IBM, Dell, and others have used it as a way to sell hardware and support services — but Red Hat has managed the tricky business of building a software platform that big businesses will pay for.*

[13] Mozilla Foundation Open Letter Orders Unofficial Mozilla Merchandise Sellers to Stop, Legal Action Hinted (March 16, 2004)

[14] The Wikimedia Shop, *The official online store for Wikipedia and its sister projects.*

[15] Moodle will always be an open source project Posted 06 Oct 2014 by Samantha Gartner on opensource.com

[16] Moodle: a case study in sustainability on oss-watch.ac.uk by Martin Dougiamas, Managing Director, Moodle Pty Ltd on 5 June 2007

[17] How do the Moodle Partners work? on moodle.com (accessed May 31, 2015)

[18] Moodle Trademark Policy on moodle.com (accessed May 31, 2015)

[19] Blackboard's Open-Source Pivot on insidehighered.com by Steve Kolowich (March 27, 2012)

[20] Molla, Rani (2013-08-06). "Hacktivist Richard Stallman takes on proprietary software, SaaS and open source". gigaom.com. Retrieved 2013-08-12. *He also claims software as a service (SaaS) is inherently bad because your information goes through a server beyond your control and that server can add additional software when it likes.*

[21] Johnson, Bobbie (2008-09-29). "Cloud computing is a trap, warns GNU founder Richard Stallman". The Guardian.

Retrieved 2013-08-10. *Web-based programs like Google's Gmail will force people to buy into locked, proprietary systems that will cost more and more over time, according to the free software campaigner*

[22] List of free-software licences on the FSF website: "*We recommend that developers consider using the GNU AGPL for any software which will commonly be run over a network*".

[23] Tiemann, Michael (2007-06-07). "GNU Affero GPL version 3 and the "ASP loophole"". OSI. Retrieved 2013-08-19.

[24] Why did you decide to write the GNU Affero GPLv3 as a separate license? on gnu.org

[25] Sneddon, Joey-Elijah (2012-06-01). "Will You Help Change The Way Open-Source Apps are Funded?". OMGUbuntu. Retrieved 2013-08-08. *Lunduke is pledging to open-source and distribute his portfolio of hitherto paid software – which includes the Linux distro management simulator Linux Tycoon - for free, under the GPL, if he can reach a donation-driven funding goal of $4000/m. Reaching this goal, Lunduke says, 'will provide proof for others, who would also like to move their software businesses to be open source, that it is doable.'*

[26] SourceForge.net Donation System on sourceforge.net

[27] Mozilla Foundation (December 15, 2004). "Mozilla Foundation Places Two-Page Advocacy Ad in the New York Times". Retrieved June 15, 2010.

[28] Marson, Ingrid (2004-12-16). "New York Times runs Firefox ad". cnet.com. Retrieved 2013-08-12. *Fans of the Mozilla Foundation's Firefox browser who funded an advertisement in The New York Times will finally get to see their names in print on Thursday.*

[29] Leyden, John (2004-08-03), *Mozilla to pay bounty on bugs*, The Register, retrieved 2013-08-10

[30] Evers, Joris (July 25, 2005). "Offering a bounty for security bugs". *CNET*. CBS Interactive. Retrieved 12 August 2007.

[31] "Mozilla Foundation Announces Security Bug Bounty Program". *Mozilla Foundation*. Mountain View, California. August 2, 2004. Retrieved 2013-08-10.

[32] Lunduke, Bryan (2013-08-07). "Open source gets its own crowd-funding site, with bounties included - Bountysource is the crowd-funding site the open source community has been waiting for.". networkworld.com. Retrieved 2013-08-10. *Many open source projects (from phones to programming tools) have taken to crowd-funding sites (such as Kickstarter and indiegogo) in order to raise the cash needed for large-scale development. And, in some cases, this has worked out quite well.*

[33] Arceri, Timothy (2013-07-26). "Help improve OpenGL support for the Linux Graphics Drivers". Indiegogo. Retrieved 2013-08-11. *Helping fund the time for me to become a Mesa contributor and document the experience to make it easier for others to understand where to start with the Mesa codebase. Many people have brought up the idea of crowd sourcing open source driver development. This is a small scale experiment to see if it could actually work.*

[34] Crowd-Funding Is Back For Another Mesa Extension by Michael Larabel on Phoronix (12 November 2013)

[35] Callaham, John (2013-06-06). "Report: Google paying AdBlock Plus to not block Google's ads". neowin.com. Retrieved 2013-08-13. *Google is paying money to Eyeo, the company behind AdBlock Plus, so that its ads get through the browser ad remover.*

[36] Hunt, Katherine (2007-05-24). "Sourceforge quarterly profit surges as revenue rises". marketwatch.com. Retrieved 2013-08-13. *Software Corp., late Thursday reported third-quarter net earnings of $6.49 million, or 9 cents a share, up from $997,000, or 2 cents a share, during the year-ago period. Pro forma earnings from continuing operations were $2.1 million, or 3 cents a share, compared with $1.2 million, or 2 cents a share, last year. The Fremont, Calif.-based maker of computer servers and storage systems said revenue for the three months ended April 30 rose to $10.3 million from $7.9 million. Analysts, on average, had forecast a per-share profit of 2 cents on revenue of $12 million.*

[37] "SourceForge Reports Second Quarter Fiscal 2009 Financial Results".

[38] Hustvedt, Eskild (2009-02-08). "Our new way to meet the LGPL". Archived from the original on 2009-02-20. Retrieved 2011-03-09. *You can use a special keyword $ORIGIN to say 'relative to the actual location of the executable'. Suddenly we found we could use -rpath $ORIGIN/lib and it worked. The game was loading the correct libraries, and so was stable and portable, but was also now completely in the spirit of the LGPL as well as the letter!*

[39] Mike Olson (co-founder and CEO of Sleepycat Software and Cloudera), lecture to Stanford University entrepreneurship students, 2013.11.13

[40] Twenty Years of Experience in Developing Software in Silicon Valley, Kim Polese lecture to Stanford University engineering entrepreneurship students, 2005.11.09

[41] "TTimo/doom3.gpl". GitHub. 2012-04-07. Retrieved 2013-08-10. *Doom 3 GPL source release [...] This source release does not contain any game data, the game data is still covered by the original EULA and must be obeyed as usual.*

[42] STEEL STORM EPISODE 1 LIMITED USER SOFTWARE LICENSE AGREEMENT "Steel Storm Episode I EULA". steel-storm.com. Retrieved 2013-08-10. *For the purpose of this Agreement, the Art Assets include pk3 archive inside of "steelstorm/gamedata/" folder that contain two-dimensional and three-dimensional works of graphic art, photographs, prints and art reproductions, maps, charts, diagrams, models, and technical drawings, sound effects and musical arrangements, documentation and tutorial*

videos, and are licensed under Attribution-NonCommercial-ShareAlike 3.0 Unported license. The Engine, which includes Windows, Linux and Mac binaries, and the Engine's source code, are licensed under GNU GPL v2 license.

[43] Stallman, Richard (2012). "On-line education is using a flawed Creative Commons license". stallman.org. Retrieved 2013-08-10. *In my view, nonfree licenses that permit sharing are ok for works of art/entertainment, or that present some party's viewpoint (such as this article itself). Those works aren't meant for doing a practical job, so the argument about the users' control does not apply. Thus, I do not object if they are published with the CC-BY-NC-ND license, which allows only noncommercial redistribution of exact copies.*

[44] "Eben Moglen, speaking about GPLv3 in Barcelona".

[45] Oram, Andy (2011-08-26). "How Free Software Contributed to the Success of Steve Jobs and Apple". radar.oreilly.com. Retrieved 2013-08-10. *the BSD license allowed Apple to keep its changes proprietary*

[46] NVIDIA Drops Their Open-Source Driver, Refers Users To VESA Driver on phoronix *The xf86-video-nv driver has been around that provides very basic 2D acceleration and a crippled set of features besides that (no proper RandR 1.2/1.3, KMS, power management, etc.) while the code has also been obfuscated to try to protect their intellectual property.*

[47] Obfuscated "source code" is not real source code and does not count as source code. - https://www.gnu.org/philosophy/free-sw.html

[48] "Reasoning behind the "preferred form" language in the GPL". lwn.net. 2011-03-07. Retrieved 2013-08-19.

[49] Phoronix - Towards A Real Business Model For Open-Source Software on Phoronix.com

[50] id Software releases Doom 3 source code on h-online.com (3 November 2011)

[51] id Software makes iPhone Wolfenstein open source by Spanner Spencer (March 2009)

[52] Shadow Warrior Source Code Released 3D Realms (1. April 2005)

[53] SOURCE CODE *Selected games have had their source code released by us. These games are: Duke Nukem 3D, Shadow Warrior, Rise of the Triad, Word Whiz, Beyond the Titanic, Supernova, & Kroz. You can obtain these from our downloads page.*

[54] Andersen, John (2011-01-27). "Where Games Go To Sleep: The Game Preservation Crisis, Part 1". Gamasutra. Retrieved 2013-01-10. *The existence of decaying technology, disorganization, and poor storage could in theory put a video game to sleep permanently -- never to be played again. Troubling admissions have surfaced over the years concerning video game preservation. When questions concerning re-releases of certain game titles are brought up during interviews with developers, for example, these developers would reveal issues of game production material being lost or destroyed. Certain game titles could not see a re-release due to various issues. One story began to circulate of source code being lost altogether for a well-known RPG, preventing its re-release on a new console.*

[55] Bell, John (2009-10-01). "Opening the Source of Art". Technology Innovation Management Review. Retrieved 2013-08-09. *[...]that no further patches to the title would be forthcoming. The community was predictably upset. Instead of giving up on the game, users decided that if Activision wasn't going to fix the bugs, they would. They wanted to save the game by getting Activision to open the source so it could be kept alive beyond the point where Activision lost interest. With some help from members of the development team that were active on fan forums, they were eventually able to convince Activision to release Call to Power II's source code in October of 2003.*

[56] "NETSCAPE ANNOUNCES PLANS TO MAKE NEXT-GENERATION COMMUNICATOR SOURCE CODE AVAILABLE FREE ON THE NET". Netscape Communications Corporation. 1998-01-22. Archived from the original on 2007-04-04. Retrieved 2013-08-08. *BOLD MOVE TO HARNESS CREATIVE POWER OF THOUSANDS OF INTERNET DEVELOPERS; COMPANY MAKES NETSCAPE NAVIGATOR AND COMMUNICATOR 4.0 IMMEDIATELY FREE FOR ALL USERS, SEEDING MARKET FOR ENTERPRISE AND NETCENTER BUSINESSES*

[57] "MOUNTAIN VIEW, Calif., April 1 /PRNewswire/ -- Netscape Communications and open source developers are celebrating the first anniversary, March 31, 1999, of the release of Netscape's browser source code to mozilla.org". Netscape Communications. 1999-03-31. Retrieved 2013-01-10. *[...]The organization that manages open source developers working on the next generation of Netscape's browser and communication software. This event marked a historical milestone for the Internet as Netscape became the first major commercial software company to open its source code, a trend that has since been followed by several other corporations. Since the code was first published on the Internet, thousands of individuals and organizations have downloaded it and made hundreds of contributions to the software. Mozilla.org is now celebrating this one-year anniversary with a party Thursday night in San Francisco.*

[58] Proffitt, Brian (2000-10-13). "StarOffice Code Released in Largest Open Source Project". linuxtoday.com. Retrieved 2013-01-10. *Sun's joint effort with CollabNet kicked into high gear on the OpenOffice Web site at 5 a.m. PST this morning with the release of much of the source code for the upcoming 6.0 version of StarOffice. According to Sun, this release of 9 million lines of code under GPL is the beginning of the largest open source software project ever.*

[59] Benkler, Yochai (April 2003). "Freedom in the Commons: Towards a Political Economy of Information". *Duke Law Journal* **52** (6).

[60] Dina ElBoghdady; Hayley Tsukayama (2011-09-30). "Facebook tracking prompts calls for FTC investigation". Washington Post. Retrieved 23 October 2011.

[61] Vaughan-Nichols, Steven. "No GPL Apps for Apple's App Store". Retrieved 23 October 2011.

[62] "Changing the VLC engine license to LGPL". Retrieved 23 October 2011.

[63] Netscape Navigator#The fall of Netscape

6.9 Text and image sources, contributors, and licenses

6.9.1 Text

- **Multi-licensing** *Source:* https://en.wikipedia.org/wiki/Multi-licensing?oldid=675872623 *Contributors:* Edward, GTBacchus, Ntoll, Furrykef, Nv8200pa, UninvitedCompany, Robbot, Bradeos Graphon, Eequor, RossPatterson, Fluzwup, Gronky, Spayrard, Kjkolb, Paul1337, Stuartyeates, Trevie, Robertwharvey, Kesla, Nneonneo, Wongm, YurikBot, RussBot, Category, Mipadi, Kardax, Itub, Petemyers, Chris the speller, NCurse, Guyjohnston, MattEngland~enwiki, Boseko, Joseph Solis in Australia, RekishiEJ, Valodzka, Kozuch, Neustradamus, Knocte, Deadly Serious, Sboden, Fuchsias, Rhwawn, DWorley, Bkraft, Felipe1982, 1mujin22, STBotD, Wxidea, Billinghurst, Xeno8, Redyoshi49q, Altosax456, Logan, SieBot, One Last Time, Alex.muller, Jruderman, ClueBot, Rodhullandemu, Jeshan, Thingg, Azrael Nightwalker, SF007, Jytdog, MystBot, Addbot, Ironholds, Scientus, LaaknorBot, Bluebusy, Jarble, Yobot, Ipatrol, David290, Dinamik-bot, Tomchen1989, ZéroBot, Luca Ghio, BG19bot, Armand3496, ChrisGualtieri, Rotlink, Jodosma, Comp.arch, MargeAtwood, Monkbot, Fench and Anonymous: 59
- **Professional open source** *Source:* https://en.wikipedia.org/wiki/Professional_open_source?oldid=676023170 *Contributors:* Lkesteloot, Chealer, Khalid hassani, Mdd, Toussaint, BD2412, Rjwilmsi, Tony1, Msgilligan, SmackBot, Cydebot, Notopia, Neustradamus, Caorongjin, Keith D, Rodhullandemu, Mild Bill Hiccup, Xp54321, Fgnievinski, Tide rolls, Jarble, Yobot, AnomieBOT, Gilo1969, Mark Schierbecker, LittleWink, John of Reading, JeremyCarroll, JordiGH, Senator2029, Madrenergic, BG19bot, Raphael Perez, Estevezj, Soujak, Benjamin Mestrallet, ArunPatel95 and Anonymous: 19
- **Open-source bounty** *Source:* https://en.wikipedia.org/wiki/Open-source_bounty?oldid=671385311 *Contributors:* Chealer, David Gerard, Kfogel, John Vandenberg, Mequa, Czar, Twin Bird, Tony1, SmackBot, QuimGil, Cydebot, Kozuch, Fences and windows, Addbot, Softy, AnomieBOT, VanishedUser sdu9aya9fasdsopa, FrescoBot, Dewritech, Tisane, SporkBot, ChuispastonBot, Shaddim, BG19bot and Anonymous: 5
- **Open core** *Source:* https://en.wikipedia.org/wiki/Open_core?oldid=650537964 *Contributors:* Zundark, Pnm, Bearcat, David Gerard, Gronky, Ahunt, ViperSnake151, Thumperward, JustAGal, Eurobas, Nemo bis, Genium, Darxus, XLinkBot, Fgnievinski, Yobot, 4th-otaku, AnomieBOT, Jonesey95, Chricho, Babaidas, Ryker-Smith, Eidab, Rickhigh, Editfromwithout, AlexanderRedd and Anonymous: 13
- **Open-source economics** *Source:* https://en.wikipedia.org/wiki/Open-source_economics?oldid=646156015 *Contributors:* Lquilter, Greenrd, Timrollpickering, Beland, Intgr, RussBot, Tony1, MikeCamel, Thomasmeeks, Widefox, DGG, Slogsweep, Fgnievinski, Yobot, AnomieBOT, BattyBot, Metroecho, DoctorKubla, Conpibien, Khazar2, IjonTichyIjonTichy, Joeinwiki, Schwartz1983, Wikiwarrior77 and Anonymous: 2
- **Business models for open-source software** *Source:* https://en.wikipedia.org/wiki/Business_models_for_open-source_software?oldid=676438787 *Contributors:* Damian Yerrick, Edward, Lquilter, Greenrd, David Gerard, Beland, RossPatterson, Bender235, Balthas, Kfogel, BlueNovember, DanShearer, Runarb, Tezh, Czar, J. M., Tony1, Cæruleum, CWenger, SmackBot, Frap, Ashawley, Cybercobra, Danjewell, Vorlion, Kvng, Samuell, Kozuch, Hillshum, Wmasterj, Mchua~enwiki, Falcor84, Kctucker, Comrade Graham, Billinghurst, Mikemoral, Spyhawk, Belorn, JL-Bot, Davodavo, Mild Bill Hiccup, SF007, Nepenthes, Grayfell, Fgnievinski, MrOllie, Jarble, Yobot, Bunnyhop11, AnomieBOT, Monkeybars3000, DSisyphBot, FrescoBot, Haeinous, Skyerise, Dinamik-bot, RjwilmsiBot, Liamzebedee, EmausBot, John of Reading, Tuankiet65, Dewritech, GoingBatty, Tisane, Mfloisand, HMPerson1, Ernstkm, Cdaffara, Mentibot, ClueBot NG, Shaddim, Delusion23, Helpful Pixie Bot, BG19bot, AvocatoBot, Raphael Perez, IjonTichyIjonTichy, Mogism, KevinFixesThings, ScotXW, Monkbot, Algo8 and Anonymous: 42

6.9.2 Images

- **File:Ambox_important.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/b4/Ambox_important.svg *License:* Public domain *Contributors:* Own work, based off of Image:Ambox scales.svg *Original artist:* Dsmurat (talk · contribs)
- **File:Crystal_Clear_app_kedit.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/e8/Crystal_Clear_app_kedit.svg *License:* LGPL *Contributors:* Sabine MINICONI *Original artist:* Sabine MINICONI
- **File:Crystal_Clear_device_cdrom_unmount.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/10/Crystal_Clear_device_cdrom_unmount.png *License:* LGPL *Contributors:* All Crystal Clear icons were posted by the author as LGPL on kde-look; *Original artist:* Everaldo Coelho and YellowIcon;
- **File:Edit-clear.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/f/f2/Edit-clear.svg *License:* Public domain *Contributors:* The *Tango! Desktop Project*. *Original artist:*

  The people from the Tango! project. And according to the meta-data in the file, specifically: "Andreas Nilsson, and Jakub Steiner (although minimally)."
- **File:Emblem-money.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/f3/Emblem-money.svg *License:* GPL *Contributors:* http://www.gnome-look.org/content/show.php/GNOME-colors?content=82562 *Original artist:* perfectska04
- **File:Merge-arrow.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/aa/Merge-arrow.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Mergefrom.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/0f/Mergefrom.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Nuvola_apps_emacs_vector.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/6/67/Nuvola_apps_emacs_vector.svg *License:* LGPL *Contributors:*
- Nuvola_apps_emacs.png *Original artist:* Nuvola_apps_emacs.png: David Vignoni

- **File:Question_book-new.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/9/99/Question_book-new.svg *License:* Cc-by-sa-3.0 *Contributors:*

  Created from scratch in Adobe Illustrator. Based on Image:Question book.png created by User:Equazcion *Original artist:*

  Tkgd2007
- **File:Split-arrows.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a7/Split-arrows.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Text_document_with_red_question_mark.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a4/Text_document_with_red_question_mark.svg *License:* Public domain *Contributors:* Created by bdesham with Inkscape; based upon Text-x-generic.svg from the Tango project. *Original artist:* Benjamin D. Esham (bdesham)

6.9.3 Content license

-