# Choosing Open Source Software License and Corresponding Business Model

*Juho Lindman, Matti Rossi and Anna Paajanen*
*Information Systems Science*
Aalto University School of Economics
Helsinki, Finland
email: [ ]@hse.fi

*Abstract*— **Scores of software producers have turned towards Open Source licenses to improve service for their customers. For these companies, choosing the correct Open Source license determines business success. When the available Open Source stack and licensing options grow, so does the need to understand the interplay between licensing, sourcing decisions and business goals. We propose a model of license choice that emphasizes the different licenses and rationalizes the choice of OSS license. This is crucial for smaller companies and start-ups that do not have the tools and knowledge to perform a thorough investigation of all the consequences of their license choice every time they employ OSS.**

***Keywords; licensing, copyrights, intellectual property***

We tend to agree that lawyer involvement in software business has increased. One notable area of their influence is the Open source software (OSS) stack increasingly used in different kinds of organizations.[1,2] In order to utilize the full potential of a certain software, we need to examine how organizations and users can leverage it. The key legal instrument for this, the software license, is usually chosen by the author of the source code. It defines how the rights to a certain piece of software are divided. A license dictates what users can and cannot do. Thus the license is not only a commercial and technical, but also a juridical (and sometimes even a political) tool. Therefore it is of interest to anyone working with software, not just lawyers. OSS research has focused on explaining licensing issues [3,4] and their interplay with the community contribution. [5,6,7,8,9] But how is a suitable license chosen inside software companies in practise?

We took part in a European industry project (See Box 1) focusing on the commodification of software. During the project, we observed how the partner companies were struggling with the complex interplay of issues related to license terms and obligations as they were conducting their business. They faced even greater difficulties when estimating the future impact of the license choice. We provided our partners with a model of OSS license decision-making as a deliverable of the project and, based on partner feedback, we summarize our findings for a wider audience.

Box 1: Background studies

The content of this paper is based on a systematic review of the OSS literature and a set of interviews conducted in software-intensive organizations in the Finnish legal context. Research took place as part of the ITEA-COSI project (Co-development using Inner and Open source in Software Intensive products). The scope of the project was how companies engage software commodification and OSS. ITEA-COSI lasted from 2006 to 2008 and had industrial and academic partners from several different European countries. The licensing interviews were directed to a rich sub-set of companies operating in the Finnish legal context. These companies were developing software and benefiting from it as part of their offering. The respondents were chosen based on their expertise on OSS licenses in the companies. In small companies the respondent was the CEO and in large companies either the country manager or the Open Source department head and legal advisor. The main criterion was that the company is the one that initially publishes the source code and does not just utilize open source components in some end product.

Other results of the research project are reported in the following papers:
F. Van Der Linden, B. Lundell, and P. Marttiin, "Commodification of Industrial Software: A Case for Open Source," *IEEE Software*, vol 26, no 4, 2009, pp. 77-63.
J. Lindman, J-P Juutilainen and M. Rossi, "Beyond the Business Model: Incentives for Organizations to Publish Software Source Code," in: Proceedings of the 5th International Conference on Open Source Systems(OSS2009), Springer, 2009, pp. 47-56.
J. Lindman, M.Rossi and P. Marttiin, "Applying Open Source Development Practices Inside a Company," in: Proceedings of 4th International Conference on OSS(OSS2008), Springer, 2008, pp. 381-387.
S. Ziemer, Ø. Hauge, T. Østerlie and J. Lindman, "Understanding Open Source in an Industrial Context," in: Proceedings of the 4th IEEE International Conference on Signal-Image Technology & Internet-Based Systems (SITIS2008), IEEE Computer Society, 2008, pp. 539-546.
J. Lindman, A. Paajanen, M. Rossi, "Choosing an Open Source Software License in Commercial Context: A Managerial Perspective," in Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications(SEAA2010), IEEE Computer society, 2010, pp.237-244.
J. Lindman, R. Rajala and M. Rossi, "FLOSS-induced Changes in the Software Business: Insights from the pioneers," in: Proceedings of the 1st International Conference of Software Business (ICSOB2010), 2010, pp. 199-205.

## LICENSE CHOICE

OSS development activity has certain special legal characteristics. Under normal working life conditions, there is an employment contract between a software-making company and its employees. The contract contains the conditions to make the employment beneficial for both parties. In traditional software development the relationship between the employees and the company is quite formal. The rights are transferred to the employer based on the employment contract. Intellectual property law often protects these rights and contracts usually include very detailed explanations as to who gains what rights to the software. When the company owns these intellectual property rights, it can license the end product for a defined price for the users. This traditional situation is represented in the left part of the Figure 1.

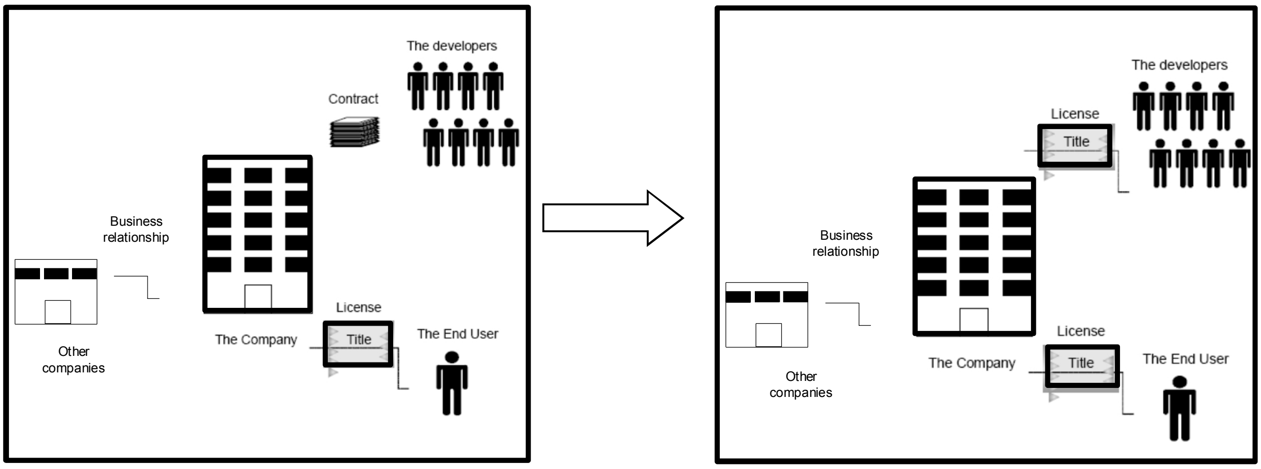

Fig 1 Juridical relationship in a closed source development (left) and OSS community (right)

When a company develops software with an open source community, employment contracts can obviously be used only for the company employees that participate in the project. Companies that choose to publish the source code and to engage in an open development process, hope to gain external contributions from other stakeholders such as customers, partners and voluntary developers and gain their revenue using a wide variety of business models based on service sales and consulting. For other stakeholders, the legal relation is based on the open source license, under which the initial source code is published for the community. The right side of the Figure 1 depicts this situation.

It should be noted that there are also other companies operating in the marketplace such as customers, competitors, vendors, sub-contractors and often some of the work can be contracted out to them. This will make the legal situation more complex and result in chains of rights and contracts that complicate the situation. However, the crucial difference that we want to emphasize in the figure is the relation between the paid developers and external contributors. We do not imply that the community contribution would be free of obligations, but that it cannot be covered by an employment contract (unless the company chooses to employ all the developers). Sometimes it is possible to use contracts other than an employment contract to guarantee contribution, such as a partnership agreement with a partner or a sales contract with a customer. However, the main legal tool is the software license. The license is used both with the open source community - the “employees” - and the end user. Thus one must pay special attention when choosing the license. Wrong choice might result in lost revenue or even the loss of control over the development of the software.

To help with the license decision the Open Source Initiative (OSI) has set ten minimum requirements for an open source license to gain the “OSI Certified” label (www.opensource.org). This certification guarantees acceptability of the license for both the developers and the industry. The OSI defined licenses can be divided into two opposite camps: restrictive (LGPL, GPL, MPL) and permissive licenses (MIT, BSD, Apache). [10,3] The main differences between these licenses and commercial software licenses can be seen in the Table 1.

| License | Type | Derivative | Bundling |
|---|---|---|---|
| MIT, BSD | Permissive | No restrictions | No restrictions |
| Apache | Permissive | Apache name not allowed marketing | Apache name not allowed marketing |
| MPL | Restrictive | GPL | No restrictions |
| LGPL | Restrictive | GPL or LGPL | No restrictions |
| GPL | Restrictive | GPL | Only GPL |
| Proprietary licenses | Restrictive | Not allowed | Restricted |

Table 1: Some common licenses and their characteristics

Permissive licenses (MIT, BSD, and Apache) have academic roots. They allow, but do not require, the distribution of source code for derivative works. [11, 3] These licenses can also be used to create a proprietary good and the code contributed by someone in the community does not have to be attributed back. The MIT license is the simplest popular license in use. [11] MIT and BSD licenses permit commercial use of the software for the licensor as well as for the licensee. The only restriction is that the name of the license, the names of the code creators and the warranty clause cannot be removed from the software. The Apache license resembles university licenses in many aspects, but the Apache 2.0 is juridically more detailed and the implicit assumptions of the MIT and the BSD licenses have now been written down. [11] What differentiates the Apache License 2.0 is the option to use different licenses for the Apache code in derivative works. These licenses permit the commercial use of the final work or the derivative work. Thus they offer a possibility for a company to open source code and then close it again later for licensing purposes. There remains a risk that if a university license is used and the project is not well governed, some competitor may fork the code to produce a competing solution. GPL, LGPL and MPL do not allow this to happen. These licenses are also far more complicated.

The General Public License (GPL) is evidently the most used OSS license. It was generated by Richard Stallman, the founder of the Free Software Foundation. GPL follows the OSI definitions, but it has a component that also affects derivative works. This means that if the company uses the GPL license to open the development, it will have to publish the source code of the final product, as well as any derivative, under the same GPL license. The Lesser (before Library) General Public License, or the LGPL, is a little less restrictive than the GPL. The origins of this license are still in the Free Software Foundation: even though the direct modifications done to the LGPL software must be licensed under the same license or GPL, the combination can be licensed under some other license, even proprietary.[3] The Mozilla Public License (MPL) was crafted in 1998 when Netscape decided to open the source code of its Internet browser. [11] Since then MPL has been used in other corporate open source projects as a starting point of what has then been further modified to suit the particular company needs. The license has a reciprocity stipulation just like the GPL forcing the contributors to give all the source code modifications back to the community. GPL also has different versions that differ slightly, but we limit them outside the scope of this study, as we also do with double licensed software.

The existing OSI certified licenses differ from each other materially. Knowing the different licenses and their characteristics is essential for the OSS in company usage, but this knowledge is not enough. The interplay of the chosen business model, [7,4] and project management need to be considered when making the license choice.[5,6] Larger software companies sometimes have the possibility to alter some of the existing license terms to meet their specific need and thus create a new license. [12]

## LICENSE AND BUSINESS MODEL

This article is not about “OSS business models” as there already exist several thorough overviews on the topic (for example [13,14,15]), but about linking the license and the business model. Based on our empirical research, we claim that the business goals and the selected business models and licensing decisions are closely interrelated. Our research shows the different motivations that can, and should, be taken into account when making the licensing decisions. The companies in our study took the motivational aspect carefully into account. We found three types of motivational aspects: externalities, creating developer motivation and leadership in the community. In addition, company size had an effect on decision-making. Our derived framework is depicted in Figure 2.

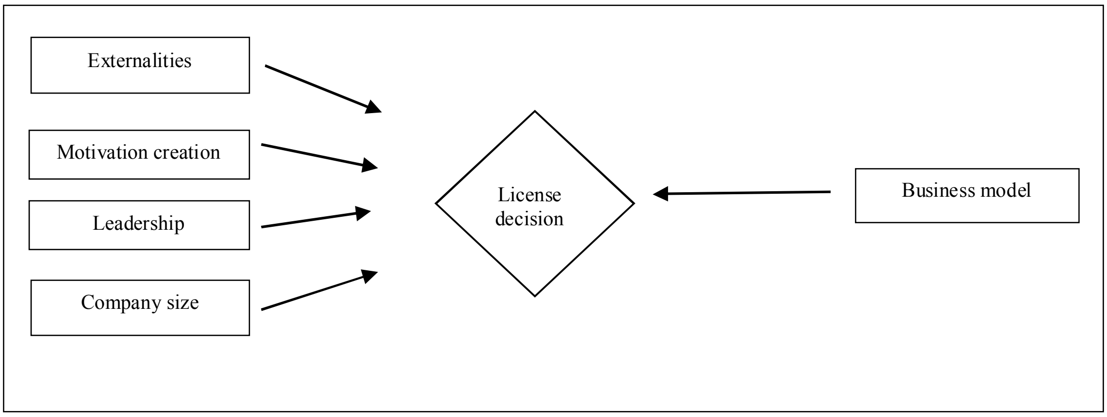

Figure 2: Framework to explain license model choice

*Network externalities*

Network externalities seem to have a strong impact on the license choice. In the context of OSS licenses, the prevalence of one particular open source software license may affect the license choice as choosing a license already in use may imply a more successful project. Existing projects normally have their licensing schemes already in place. A lawyer from a large telecommunications company noted *"In 95% of the cases, the license choice is already made when we join the community. So we have no other option than to take the license they are using. Well, we have negotiated in order to change the license in a community in few cases"*. Often the case shows that the company does not have any other option than to choose a certain license. To give an example, since the GPL includes the reciprocity stipulation, it forces every project it is combined with to be licensed under GPL. If the company wants to use an existing GPL project, or a GPL component for its own use, the resulting code must also be licensed under GPL. [9,5]There are business effects if a product cannot be integrated into other existing systems. Customers have occasionally been lost because their existing systems did not allow for a certain license. Also, some crucial existing components needed to be rejected because they could have caused problems for the end customers.

*Motivation creation*

If the company considered OSS because of competitive advantage, then community developer motivation was something that needed to be addressed. On the other hand, for those companies that built on their "own" communities, the license was a major tool to gain trust among the potential members. GPL or LGPL were often chosen to signal that the company wants this particular software to stay open. One interesting move some software companies are using to motivate their employees is to guarantee, in the employment contract, that all software produced for the company will be released under an OSS license. The attitude towards the community went hand in hand with the company background. As one respondent indicated *"We first thought about how to get our partners (competitors) interested in our product and that drove the choice towards GPL. But we also considered that GPL could motivate other people to join"*. Choosing certain license is of course just part of motivation creation, but license is the foundation onto which the community cooperation is built.

*Leadership and control*

The companies wanted the license to control the development so that the whole software package is also retained and minimum amounts of forking would take place. In addition, to license companies often assigned their own employees to work in the community. One respondent noted *"Our own employees that are involved in the development help to manage the project"*. Some employees were assigned specific roles, like a project manager, but others were solely members of the community, like any other outsider member. License control is in line with earlier literature: to have a successful OSS project, the company must take part in the coding process. *"In the most important projects, we have a strong occupation. In addition we contribute a lot to other communities to show that we are a capable company that should be taken seriously, and so we get merit in the communities"*. If the company chooses a permissive license, this will be even more important since the project may be taken over by someone else if no strong leadership exists.[5] The permissive licenses (BSD and MIT) include the copyright notification of the original source code author, so even if a competitor forks the project, people will know who originally started the work[9]. Software forking, or the possibility of not getting any code returned, is taken into account when deciding which license to use.

*Company size*

It is hardly surprising that company size has a dramatic effect on the decisions concerning licenses and how the decisions were made. Size was often also related to the age of the company or the software. Small size magnified the risks of making wrong licensing choices. On the other hand, some smaller companies were very skillful in their analyses or had acquired the legal expertise from outside the company to avoid unnecessary risk related to licensing. The small companies with fewer resources to use for making the decision often start from a certain community developed product or the business model they want to use. Only when this is already in place, do they consider the license. The large companies have more leeway in comparing the different alternatives and trying to influence the community members in other ways than choosing a certain license. Conversely, smaller companies often benefit more from community goodwill.

### *Business model*

The factors above determine the context of choosing the right license. This must be matched with a business model that is compatible with the chosen license. If the company uses the traditional OSS model of support selling, the license can be very restrictive, but if the plan is to sell a proprietary software asset, there are fewer options. Some early choices can even make this business model invalid. The choice of license has path-dependence for the possible business models. For example, choosing GPL, the most popular license among the developers, and the most restrictive from business point of view limits the available business models. Conversely, choosing a popular license can provide the critical mass of developers for the product and thus the project might have better chances to be successful. The companies first choose whether they will aim to make profit by selling a) support, b) connected hardware or c) commercial software.[13,14] The a) and b) options are similar in the sense that the original license does not change when distribution and profits begin. If the company aims to use the open source software just as a component inside a larger proprietary system (the option c) above) license choice has a crucial role. GPL and LGPL are out of the question due to the reciprocity stipulation. Any other licenses can be used for an open source component without restrictions.

Experience shows that licensing decision is tied to the long-term support costs of software. Predicting support costs over the entire life-cycle of software is difficult, especially in the early stages of software life-cycle. However, it can also lead to good payback. Choosing a good OSS package may help to reduce support costs significantly, if compared to building internal software assets and their maintenance. This approach is exemplified by Apple's dramatic business decision to base their operating system (OS X) on top of a Unix-variant (BSD-licensed), scrapping decades of internal software development. This was a risky move, which in Apple's case proved to be successful. Apple's decision becomes especially interesting when it is compared to the role of the OSS stack in the Android-ecosystem or Nokia's Meego.

A certain level of anxiety was observed in the interviews concerning the shift from license based-pricing toward service-based pricing. Especially the companies that had started in the closed source code business and license sales as 100% revenue model are somewhat hesitant to move towards a SAAS (Software As A Service) model. These companies did not want to not exclude the possibility to also gain revenue from license sales. The more additional terms or agreements there are alongside the license, the more complicated these combinations get and the less they will resemble the OSI defined licenses. Even the largest companies have already relinquished their own licenses because these were not considered to enable external contribution. The additional agreements might have a similar effect.

## SUMMARY

The implications of this study are especially important for small software companies that do not have as many resources for use in the decision making as the large companies. There is a fine line between a permissive license with perhaps a smaller community, but secured immaterial property rights and a restrictive license with a larger community, but with no copyright ownership or license sales. The correct license for a project depends on the nature of the end product and the company's intentions. The findings of our study indicate that we are living in a critical junction in software creation. One trend is increasing openness and seeking revenue mainly from selling services. The other direction is back towards closed domain software development.

## REFERENCES

[1] F. Linden, B. Lundell and P. Marttiin, "Commodification of Industrial Software - a Case for Open Source", *IEEE Software*, July/August, 2009.

[2] B. Fitzgerald "The transformation of open source software", MIS Quarterly 30:3, 2006, pp. 587-598.

[3] M. Välimäki, "The Rise of open source licensing: a Challenge to the use of intellectual property in the software industry". Helsinki:Turre Legal, 2005.

[4] E.S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary,* Sebastopol, CA:O'Reilly, 1999.

[5] M. Välimäki and V. Oksanen, "Evaluation of open source licensing models for a company developing mass market software", *Law and Technology*, 2002.

[6] L. Dahlander and M. Magnusson,"Relationship between open source software companies and communities: Observations from Nordic firms", *Research Policy*, vol. 34, no 4, 2005, pp. 481-493.

[7] P. Laat,"Copyright or copyleft? An analysis of property regimes for software development", *Research Policy*, vol 34, no 10, 2005, pp. 1511-1532.

[8] J. Lerner and J. Tirole,"The scope of open source licensing", *Journal of Law, Economics & Organization*, vol 21, no 1, 2005, pp. 20-56.

[9] A. Bonaccorsi and C. Rossi, "Altruistic individuals, selfish firms? The structure of motivation in Open Source software" *First Monday*, vol 9, no 1, 2003.

[10] P. Laat,"Copyright or copyleft? An analysis of property regimes for software development", *Research Policy*, vol 34, no 10, 2005, pp. 1511-1532.

[11] A. St.Laurent, *Understanding open source and free software licensing*, Sebastopol:O'Reilly, 2004.

[12] D. Kennedy, "What lawyers need to know about the open source licenses", *Journal of Internet Law*, 2004 pp. 3-10.

[13] F. Hecker,"Setting up shop: The business of open-source software", *IEEE Software*, vol 16, no 1, 1999, pp. 45-51.

[14] M. Fink, *The business and economics of Linux and open source*, New Jersey: Prentice Hall, 2003

[15] R. Rajala R, J. Nissilä and M. Westerlund, "Revenue models in the open source software business", In *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives* (K St. Amant and B. Still, Eds) IGI Global, Hershey, PA, 2007.