

First Results About Motivation and Impact of License Changes in Open Source Projects

Robert Viseur^{1,2(✉)}, and Gregorio Robles^{3(✉)}

¹ CETIC, Rue des Frères Wright, 29/3, B-6041 Charleroi, Belgium
robert.viseur@cetic.be

² University of Mons (Faculty of Engineering), Rue de Houdain, 9, B-7000
Mons, Belgium
robert.viseur@umons.ac.be

³ Universidad Rey Juan Carlos, 28943 Fuenlabrada, Spain
grex@gsync.urjc.es

Abstract. Free and open source software is characterized by the freedoms and criteria that are warranted by specific licenses. These licenses describe the rights and duties of the licensors and licensees. However, a licensing change may be necessary in the life of an open source project to meet legal developments or to allow the implementation of new business models. This paper examines the motivations and impacts of license changes in open source projects. After a state of the art on the subject, a set of case studies where projects changed their license is presented. Then a set of motivations to change licenses, the ways to legally make this change, the problems caused by this change and a set of benefits of the license change are discussed.

Keywords: Open source · Intellectual property · License · Contributor agreement · Business model

1 Introduction

Licenses are thought to be selected at the beginning of the project with no posterior change (Fogel, 2005). They give the rules of the collaboration which everybody agrees to if participating in the project. To some extent, they provide a sort of “constitution” or legal agreement of how the project is developed and distributed. A change in the license is something very complex, not only because the aforementioned rules are changed, but also because it requires all contributors to agree on the license change.

One of the first, and most notorious license changes was performed by Mozilla in the late 90s (www-archive.mozilla.org). Due to incompatibility issues with the GPL noticed by the FSF, Mozilla decided to change from its dual NPL/MPL license to MPL/GPL/LGPL. This meant that all developers that had contributed to Mozilla had to explicitly give their consent to the change. Mozilla's website listed a number of contributors that could not be contacted or were missing. The license change finished when only around a dozen of missing confirmations from developers were still

required; their code was identified and rewritten or removed. Another famous relicensing project is KDE whose purpose was to solve some compatibility issues with GPLv3. The projects leaders were obliged to identify and get agreement from several hundreds of contributors to that major free and open source project.

Despite the importance of the issue for the governance of open source projects, there are few scientific papers dedicated to the motivations and the impact of licenses changes in open source projects. Our paper proposes to contribute to thinking on that issue. It is organized in four sections. The first one includes a brief state of the art about motivation and impact of the license changes. The second one is dedicated to the presentation of the methodology and the analysis of several case studies. The third one consists in a discussion of our first findings. The last one proposes some conclusions and perspectives.

2 Background

Free and open source projects (also said FOSS or FLOSS) are covered by specific licenses that warrant the free and open source nature of software as defined by the Free Software Foundation (fsf.org) and the Open Source Initiative (opensource.org). Those licenses can be divided in two main families (de Laat, 2005; St. Laurent, 2004; Viseur, 2013b). The first one includes the academic or permissive licenses; the second one, the copyleft, reciprocal or restrictive licenses. The permissive licenses allow the use of the source code in proprietary software. The Apache, BSD and MIT licenses are famous examples. Contrariwise, copyleft licenses impose limitations on licensees of any derivative work, such as the conservation of the license or the availability of the source code. The AGPL, CDDL, CPL/EPL, GPL, LGPL, MPL and OSL licenses are famous examples of copyleft licenses.

Ten years ago the copyleft licenses represented more or less 80% of all the open source projects. However the success of permissive licenses has grown over time. For example, Hofmann *et al.* (2013) noted that “*the combined effect of increased commercial investment with the need for competitively differentiated products built on top of that shared investment has lead to an increase of permissively licensed projects*”.

We identified several authors working on the license changes and their impact on the open source projects. The studies are not numerous and are often recent.

The main study was processed by Di Penta *et al.* (2012). The authors identified several motivations for license changes such as the “*changes in the legal landscape, commercial code licensed as FOSS, or code reused from other FOSS systems*” and the “*evolution of a license per se*” (i.e. GPL v2 → GPL v3). They draw their conclusions from a set of examples and empirical study analyzing, including ArgoUML, Eclipse-JDT, FreeBSD and OpenBSD, Mozilla and Samba. They found a wide variety of licenses, even in source codes that are supposed to be homogeneous, due to a period of transition after the decision to change from a license to another or a version to another one.

Savola and Anttila (2012) published a seminar report dedicated to open source software license changes. They based their seminar report on examples from scientific literature including Di Penta *et al.* (2012). For each example they identified the motivations and the result of the license change. Their selection of license changes includes Netbeans IDE, MySQL, Mono and Java. The report identifies reasons for the license changes (including better defendability, more clear license text, gain more users, better suitability for commercial usage, more paying customers, prevent commercial usage without paying license fee, reduce viral effect, partner's requirement and wider distribution) together with the problems (including license incompatibilities and push-back from the community) and the benefits (including wider distribution, user concerns about copyright and higher money income) after the license change.

Santos *et al.* (2011) worked on projects that had changed their type of license over the years. Their final sample is of 756 free and open source software for a total of 1012 intellectual property policy (IPP) interventions (i.e. changes of license type). They studied the impact of the IPP intervention on FLOSS attractiveness. The authors explored the transitions. They calculated their frequency and their positive or negative impact on the project attractiveness. The results revealed a bias in favor of the confidential Academic Free License. Note that the impact of the license type of open source software success is always a debated topic (Viseur, 2013a).

Viseur (2013b) studied the evolution of business models of open source editors facing cloud computing. The study was focused on e-business software including Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and e-commerce software. The author showed evolutions in license choices. A first pattern is related to companies switching to dual licensing model, often with technical differentiation between community and commercial release of the software (e.g., Compiere). A second pattern is related to companies facing to the rise of business software in SaaS mode and switching to copyleft licenses with network effect (e.g., OpenERP). Broadly speaking, the copyleft licenses with network effect impose the option to download the source code if users communicate with the software interface. Famous examples are the GNU Affero General Public License (AGPL) and the Open Software License (OSL).

3 Methodology and Cases Studies

We list a set of project that encountered at least one license change (see Table 1). The list was manually fed by the authors. Some information collected in the context of a study about forks was also used (Viseur, 2012). The list was completed by new examples that the authors obtained after Twitter requests.

Table 1. Documentation of license changes

Project	License (from)	License (to)
Alfresco	GPL with FLOSS exception	LGPL
ath5k	BSD	BSD + BSD/GPL (fork)
Compiere	MPL	GPL + proprietary
Dimdim	MPL	GPL
ExtJS	LGPL-style	GPL + proprietary
IP Filter	IPFilter (BSD-Like)	IPFilter (modified)
Java	SCSL (hybrid)	GPL with classpath exception
Joomla (framework)	GPL	LGPL
JQuery	GPL/MIT	MIT
Mongoose	MIT	GPL + proprietary
Mono	GPL	MIT/X11
Mozilla	NPL MPL	MPL MPL/LGPL/GPL
MySQL	LGPL	GPL
OpenERP	GPL AGPL	AGPL LGPL
OpenStreetMap	CC-BY-SA	ODBI
SharpDevelop2	GPL	LGPL
Paint.Net	MIT	Proprietary
Squeak	Squeak Apple Public License Apache License	Apple Public License Apache License MIT/Apache License
SugarCRM	SPL (MPL-like) GPL	GPL + proprietary AGPL
Talend	GPL	LGPL
Trolltech	QPL GPL	GPL + proprietary LGPL
VLC	GPL	LGPL
WebM	WebM	WebM (modified)
XFree86	MIT	MIT with credit clause

We analyze each case by focusing on a set of criteria: the application domain, the original and the new license, the projects leaders' motivation to change, the way the license change was conducted and the impact on the open source project. Our work is mainly based on scientific or professional sources such as Bert-Erboul (2013), de Laat (2005), Di Penta and German (2010), Hamerly and Paquin (1999), Välimäki (2003), Viseur (2012), Viseur (2013a), Viseur (2013b) and Willis (2012). The issue of license change has usually not been the main subject of the scientific papers; however, professional sources document some aspects for specific projects. We describe below four representative cases: Mozilla, OpenERP, Trolltech and VLC.

Mozilla

Domain - The Mozilla Foundation develops several tools for the World Wide Web. The most famous are the discontinued Mozilla suite and the lighted Firefox Web browser.

License - The project was published in 1998 under NPL license. A conflict with the community led to the publication under the first version of MPL license. The LGPL and the GPL later appeared in the development (triple license scheme), in order to address problems of incompatibilities with third-party projects. The project also published an evolution of the MPL license that solved incompatibility issues with the GPL license.

Scope - The license changes impacted the whole project.

How - There is no copyright assignment. The license changes needed contributor agreement. The Mozilla Foundation asked for the agreement of committers by choosing the acceptable licenses.

Impact - Simplifying reuse in some third-party products is assumed.

OpenERP

Domain - OpenERP (now Odoo) is an open source Enterprise Resource Planning (ERP) software including Customer Relationship Management (CRM) features.

License - The software moved from GPL to AGPL in 2009 in order to avoid the use in SaaS mode without publishing the modifications. The software editor proposes its own SaaS offer based on OpenERP. In 2011 the company introduced a new license, an AGPL with Private Use License (dual licensing) allowing to keep the modifications private in case of SaaS use. A new license change is planned (from AGPL to LGPL).

Scope - The license change affects all the software except the OpenERP Web Client under MPL license.

How - The process of the license change is not clearly documented.

Impact - The OpenERP company evolved its business model and was able to respond to new market trends. However the second license change was associated with the emergence of tensions within the community.

Trolltech (Qt)

Domain - The Qt software is a cross-platform application framework.

License - The software was published under QPL license. Then a dual licensing scheme (QPL + GPL) was applied under the pressure of the KDE community that used Qt intensively. Finally, after Nokia bought Trolltech, the decision to publish the software under

the LGPL license in order to spread the technology was taken. The project is currently managed by another company, Digia, and “*some modules are available under LGPLv2.1, LGPLv3 or GPL v3 and some other modules under LGPLv3 only*” (qt.io).

Scope - The whole library was impacted by the successive license changes. However the actual release licensing scheme is heterogeneous.

How - The company was copyright owner.

Impact - The publication under GPL satisfied the community and the Harmony fork was abandoned. The impact of the publication under LGPL has not been quantified. The current license schema is complex.

VLC

Main - The VLC project offered a cross-platform media player and a streaming media server. The project was initiated by students from Ecole Centrale de Paris.

License - The project has been covered by GPL v2 (or higher) since 2001. The license was changed for LGPL v2.1 (or higher) in 2011.

Scope - The license change was implemented for modules impacting third party vendors.

Motivations - There were several motivations, including “*making VLC compatible with various gadget-vendor application stores (e.g., Apple’s)*” (Willis, 2012).

How - Two hundred and thirty contributors had to be contacted. It was difficult to contact them and get response. With 150 responses, the project owned an agreement for 99.99% of the code for libvlc and libVLCcore. It was more difficult for other parts, for which it was necessary to delete, reimplement, refactor into separate files or drop some code. The code was kept under GPL if there was no impact on third party vendors.

Impact - Some modules had to be reimplemented (less than 5). The impact on Apple is unknown (opacity).

4 Discussion

Beyond the four cases that we have presented so far, we identified a set of motivations for license changes (see Table 2):

1. The license can be changed for compatibility. The open source project can thus be distributed in new packages (e.g., Java), be integrated in wider source code (e.g., WebM) or reduce the license impact on third party add-ons development (e.g., Odoo, VLC or SharpDevelop2).
2. The license can be changed by the editor with the hope that the revenues will increase. This is the case when an editor organizes a transition to a dual licensing scheme (e.g., MySQL).
3. The license can be changed from the GPL license to a more permissive one in order to avoid how “derived work” is defined (e.g., Mono).
4. The license change can be motivated by the simplification of a multiple licensing scheme (e.g., JQuery).
5. The license change can be motivated by the desire to improve the relationships to community and limit the risk of fork (e.g., Trolltech).

6. The license change can be the consequence of an ego-conflict. That reason is often alleged but not proofed (e.g., IP Filter).
7. The license change can be an adaptation to evolutions in the competitive environment. For example, a lot of open source projects are preparing themselves for the cloud computing rise (e.g., OpenERP, now Odoo, or SugarCRM).
8. The license change can be require by a commercial partner (e.g., Mono).
9. The license change can be motivated by legal arguments. It can be to clarify the original license (e.g., IP Filter) or to adopt a more suitable license (e.g., OpenStreetMap).

Table 2. Motivations (the cumulated percentage must be higher than 100% as a license change can be associated to several motivations)

Expected benefit	Case study (#24)	Percentage
#1 Compatibility issue	Alfresco, Java, Joomla, Mono, SharpDevelop2, Squeak, Talend, Trolltech, VLC, WebM	41.7%
#2 Revenue growth	Compiere, ExtJS, Mongoose, MySQL, OpenERP, XFree86	25%
#3 Permissiveness	jQuery, Mono	8.3%
#4 Simplification of license scheme	jQuery	4.2%
#5 Relationships to community	Trolltech	4.2%
#6 Ego-conflict	ath5k, IP Filter, Paint.Net	12.5%
#7 Adaptation to competitive environment	OpenERP, SugarCRM	8.3%
#8 Partner requirement	Dimdim, Mono	8.3%
#9 Legal argument	IP Filter, OpenStreetMap	8.3%

The GPL license is often in the center of the license changes; either the project changes to the GPL for financial or compatibility reason, or the project moves away from GPL to a more permissive one in order to avoid the side effects of the GPL. Note that a license change can also be the result of several motivations at once.

The license change can be hindered by the shared property (among all contributors) of the source code. The issues can be managed in two ways:

1. The editor ensures that he owns the copyright on the source code (by contributors' agreement or by rewriting of contributions) and can change the license. This is the case of MySQL or Qt.
2. The responsible must get agreement from all the contributors (if necessary, by contacting them one by one). This has been the case for Mozilla or VLC.

We identified three problems that the license change can cause:

1. The project license can impact strongly or become incompatible with other projects (e.g., MySQL with PHP). The incompatibilities are often caused by copyleft licenses such as the GPL. An exception clause (e.g., classpath exception, FLOSS or FOSS exception, etc.) allows to reduce the virality effect of the license.
2. The new license, if it is permissive, can allow the appropriation of the source code by the editor, and may be undesired for community members (e.g., Mono).
3. The license change can irritate the community and lead to a fork (e.g., ath5k).

We identified three benefits the license change can bring:

1. The updated project can gain a wider distribution due to the license change (e.g., Squeak or Java). However, the updated project can contrariwise be affected by a smaller distribution due to the license change (e.g., IP Filter).
2. The editor who updated the license can gain more revenue (e.g., MySQL). However there is insufficient data to objectify the effect of license change on the company profitability (e.g., Mono).
3. The license change can lead to the abandonment of a fork (e.g., Trolltech Qt and Harmony) if it is part of community claims.

The case studies showed that the issue of license change was not specific to software. Indeed OpenStreetMap (openstreetmap.org) shows a similar problem in the field of databases. The ODBL license (opendatacommons.org/licenses/odbl/) was adopted because it was considered more suitable compared to CC-BY-SA license (creativecommons.org).

5 Conclusion and Perspectives

As preliminary results, our paper allowed to identify nine motivations, three negative impacts and three positive impacts for license changes.

We intend to continue this study with some future research. First, we plan to add new cases of license changes. For example: KDE (relicensing to be compatible with GPL v3), Samba (adoption of GPL v3) and Wikipedia (move to CC-BY-SA). The discovery of new cases might be facilitated by the use of free and open source projects data. For example, the FLOSSmole datasets (flossmole.org) contain license information (see database schema) that would be useful to automate the search of new interesting cases. Second, we want to study in detail the identified cases. In particular we found there was often a lack of documentation about the license change procedure (even for well known projects such as Qt), and that the impact was rarely objectified. We would like to enrich the cases with facts and figures allowing to better quantify the impact of license changes. So we wish to obtain a deeper understanding of the phenomena. Our final goal is to propose guidelines to open source project managers and entrepreneurs since there is already for license selection.

References

1. Bert-Erboul, C.: VideoLan: un cas d'école dans l'industrie informatique. *Revue Terminal*, pp. 117-133 (2013)
2. de Laat, P.B.: Copyright or copyleft? An analysis of property regimes for software development. *Research Policy* **34**, 1511–1532 (2005)
3. Di Penta, M., et al.: An exploratory study of the evolution of software licensing. In: *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, vol. 1. ACM (2010)
4. Fogel, K.: *Producing open source software: How to run a successful free software project*. O'Reilly Media, Inc. (2005)
5. KDE, *Projects/KDE Relicensing*. https://techbase.kde.org/Projects/KDE_Relicensing (read: January 17, 2015)
6. Hamerly, J., Paquin, T.: *Freeing the source: the story of mozilla*. In: *Open Sources: Voices from the Open Source Revolution*. O'Reilly, January 1999
7. Hofmann, G., Riehle, D., Kolassa, C., Mauerer, W.: A dual model of open source license growth. In: Petrinja, E., Succi, G., El Ioini, N., Sillitti, A. (eds.) *OSS 2013. IFIP AICT*, vol. 404, pp. 245–256. Springer, Heidelberg (2013)
8. Mozilla, *Mozilla Relicensing FAQ - Version 1.1*. <http://www-archive.mozilla.org/MPL/relicensing-faq.html> (read: January 14, 2015)
9. Santos Jr., C.D., et al.: Intellectual property policy and attractiveness: a longitudinal study of free and open source software projects. In: *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*. ACM (2011)
10. Savola, S., Anttila, O.: *Open Source Software License Changes, Open Source Software Development Seminars* (November 22, 2012). https://wiki oulu.fi/download/attachments/28092087/ossd_2012_savola_anttila.pdf (read: January 15, 2015)
11. St. Laurent, A.M.: *Understanding Open Source and Free Software Licensing*. O'Reilly Media (2004)
12. Välimäki, M.: Dual licensing in open source software industry. *Systèmes d'Information et Management* **8**(1), 63–75 (2003)
13. Viseur, R.: Fork impacts and motivations in free open source projects. *International Journal of Advanced Computer Science and Applications* **3**, 117–122 (2012)
14. Viseur, R.: Evolution des stratégies et modèles d'affaires des éditeurs Open Source face au Cloud computing. *Revue Terminal* 113-114
15. Viseur, R.: Identifying success factors for the mozilla project. In: Petrinja, E., Succi, G., El Ioini, N., Sillitti, A. (eds.) *OSS 2013. IFIP AICT*, vol. 404, pp. 45–60. Springer, Heidelberg (2013)
16. Willis, N.: *Relicensing VLC from GPL to LGPL*, LWN.net (November 21, 2012). <http://lwn.net/Articles/525718/> (read: January 15, 2015)