Stakeholder value, usage, needs and obligations from differnet types of F/LOSS licenses

Darren Skidmore
Monash University, Melbourne Australia.
darren.skidmore@infotech.monash.edu.au

Abstract. This paper discusses different types of Stakeholders of F/LOSS, their needs, the value, usage, and obligations that stakeholders have for different types of F/LOSS licenses. Stakeholders include **Developers**: Individuals, Projects, Embedded Systems. **Vendors**: Dominant, Niche, F/LOSS. **Packagers. End Users**. **Organisations:** Disseminators, Internally, Externally Used .

Key words: F/LOSS, Licenses, Stakeholder analysis, Obligations, Value, Usage

1 Introduction

This paper, briefly, explores the perspectives of different stakeholders their requirements / constraints and usage of types of Free / Libre and Open Source Software (F/LOSS) licenses. The increase in the number, the reach and range of F/LOSS applications, as well as the types of stakeholders creates a discussion about the types of licenses which suit different stakeholders, and their needs. This is not a paper about specific licenses, but about broad types of licenses and the generalized aspects of the types, individual licenses still need to be investigated for their specific terms.

2 F/LOSS licenses

The license of the software determines what rights and obligations govern the usage of the source code. The most common F/LOSS licenses in use are the GNU GPL, and GNU LGPL and the BSD licenses[1]. These match to three main general types of licenses: reciprocal, linking and non-reciprocal licenses. Reciprocal license requires

[1] Using data from FLOSSMole project - http://ossmole.sourceforge.net/ . Because of space limitation this cannot be shown, the FLOSSMole has SQL queries that generate the data.

software that uses source code licensed under a reciprocal license must be licensed under the same license [1]. A linking license allows for an application to link to a library or application for functionality without requiring the linkee application to be licensed under the license of the linkor application. A non-reciprocal license does NOT require an application to be relicensed under the license of the original code. Other license types exist such as obligation licenses which require specific conditions or impose limits. Table 1 gives a brief description of a taxonomy of license types drawn from [2].

Table *1.* Different types of F/LOSS licenses from [2]

License Type	Brief Explanation	Example
Traditional	Considered to be the more traditional types of F/LOSS licenses	
Reciprocal	Require derivatives to follow original license	GNU GPL
Non-Reciprocal	NOT required to follow original license	BSD
Linking	Allow other code / applications to link	GNU LGPL
Dual	Different conditions for different types of Use	MySQL
Quasi Open Source	These may or may not have other considerations attached to them	
Obligation	Have an obligation or restriction	Squiz.Net
Morality	Moral conditions restricting license use	HESSLA
Viewable Source	Allows viewing of source code, but not usage	Ms-RL
Membership	Usage of source within a select community.	Avalanche Corp
Support	Not software licenses but assist and support other aspects of F/LOSS	
Content	For documentation and support information.	Creative Commons
Open Standards	For Standards for interoperability	
Public Domain	Anyone can take and use the work.	
Closed Source	No access to source code.	Microsoft EULA

3 Stakeholders

The stakeholders below are explained, in each section. The **incoming** licenses are those that govern the source code and / or applications that the stakeholders use. The **outgoing** licenses are the licenses that stakeholders use for governing the source code / applications they produce for others to use. It is possible that the licenses of the incoming and outgoing source code are the same, different or available under multiple types of licenses.

Table *2.* List of Stakeholders in F/LOSS

Developers
Individual (developers)
Project (developers)
Embedded Systems (developers)
Vendors
Dominant (vendors)
Niche (vendors)
F/LOSS (vendors)
Packagers
End Users
Organisations
Disseminators (organisations)
Internally Used without Development (organisations)
Internally Used with Development (organisations)
External Shared with Development (organisations)

Due to page limitations, the discussion is brief, which is a limitation of the paper. Many of the stakeholders will share issues, but raised here are those important to that stakeholder. Individually a business decision may be made about the license of the source code against alternatives or a specific philosophical or ideological choice may decide the license choice.

3.1 Developers

Developers are historically the users and participants in F/LOSS. In this paper, developers are generally individuals or collections of individuals.

For incoming code developers may want to develop the code on their own, but may procure code to make their own work easier, solve problems, learn, or to provide functionality. Developers might prefer using a non reciprocal style license [3] or Linking licenses.

For outgoing license, the developer might allow anyone to do what they wish, with a non-reciprocal license, or may ensure others share their modifications, or do not profit from their work and use a reciprocal license. Depending upon the incoming licenses used there may be no choice.

Individual (developers)

Individual developers want to use software for their own use or for small number of people. Since they are smaller, they may have to accept impositions of incoming licenses, due to lack of resources.

Project (developers)

The Projects are where developers and others have organised themselves into a project, perhaps large (e.g. KDE, Apache), or small project. The issues for Project developers especially for incoming licenses are similar to that of the Packagers. Some problems are in the engineering of the project, which code is being added, and from whom, or where did the code come. Where all of the code is being developed

from scratch then the authors of the code, are able to license that code as they wish. Otherwise where there is a combination there are issues of management of license mixing.

For outgoing licenses the issues are similar to that of the generic developers. The project might create their own license, to protect or enable aims of the project, eg protect trademarks [4], morality / ethical considerations, [5] or commercial control [6].

Embedded Systems (developers)

Embedded systems are generally specially built to carry out a task, such as mobile phones, lift controllers. With outgoing licenses, they may use a non-reciprocal license to increase the adoption, to increase adoption by purchasers of the embedded system; another option might be dual licenses.

3.2 Vendors

A vendor has a commercial focus with a product or range of products. Vendors might be dependent on other applications in the software stack, needing to build a supporting application or rely on third parties products.

Dominant (vendors)

Dominant vendors have control of large sections of a market. F/LOSS can be used in a mixture of tactical and strategic ways (i.e. participation in the community; sharing development costs, bug fixing and innovation). Vendors may participate in F/LOSS activities of software products, to lower their user's costs, while not interrupting their own revenue streams.

The general usage by Dominant vendors seems to be of reciprocal licenses, since this keeps changes and innovation open. There might also be use of some obligation licenses, but this is generally seen in Niche (vendors).

Niche (vendors)

Niche vendors are generally small to medium vendors, who may do work on request, and / or may have a niche product in the market.

They may need incoming functionality to fit into their software stack and so linking licenses might be used. Where they cannot link but must incorporate Non-Reciprocal licenses are probably the preferred choice.

With outgoing licenses, the use of linking or non-reciprocal licenses, might encourage others to use their applications, with the Niche vendor being able to pickup maintenance work, or advertise their expertise. They might also use an obligation license, to assist their organisational aims.

F/LOSS (vendors)

F/LOSS vendors are those vendors who primarily use F/LOSS, as part of their offerings to the market, in general they would be similar to Niche (vendors), but should be more sophisticated in their use of F/LOSS.

3.3 Packagers

Packagers are people or projects which package up FLOSS applications into a package for others to use, e.g. the Linux distributions. Specific issues are ensuring that the incoming licenses of component packages are compatible with each when combined, since some licenses have conditions which are incompatible with other license conditions.

3.4 End Users

End Users just use the application. The incoming licenses will usually not be of concern, since F/LOSS licenses are generally for the ongoing use of the code not the end user. No outgoing licenses should be needed.

3.5 Organisations

Organisations have different needs to end users. F/LOSS is no different from any software, where a business decision must be made about the benefits and constraints of any business artefact used, including the adoption of a software application or suite. A greater issue facing organisations is possibly the ongoing support and maintenance of their software [7], including the patching and possibly roadmaps of the software [8].

Dissemination (Organisations)

Some organisations wish to disseminate the source code. A Government might wish to have an authentication code distributed, and for developers and vendors to incorporated this into their software packages. One method of distribution would be to use a Non-Reciprocal type of license, which would allow any open source project to use the code, but would also allow closed source vendors to incorporate the code with no ongoing obligation. A Linking license might also be used, for greater control.

Internally Used without Development (Organisations)

Internal users of F/LOSS using it without development are possibly more concerned about support costs including training, patching and usability.

Internally Used with Development (Organisations)

For incoming code, with most F/LOSS code, an organisation should be able to take F/LOSS, use it, modify or adapt the source code to fulfil their own needs. If they do not distribute the application, they should not need to reveal the changes to the source code. Although this may differ with some obligation licenses. However there are valid reasons to reveal the changes, in that especially to get the benefits of the continual development of the program to obtain some control and certainly over the ongoing development, maintenance and direction of the codebase [9].

Externally Shared with Development (Organisations)

Some organisations, might share development, this might be part of the mission of the organisation, or to lower costs and risks. Where the organisation wishes to use the application with an external party the choice of incoming license is more important, depending on the outgoing considerations, or business requirements.

Outgoing licenses will depend on the outcomes desired by the organisation, perhaps using a reciprocal license to ensure that the code is open to all, or a membership or obligation license to enable the organisation to keep control over the software and code base.

4 Conclusion

This paper has described multiple types of stakeholders which now exist in the F/LOSS domain, and their different requirements of incoming and outgoing F/LOSS licenses. The purpose has not been to give a prescriptive directive to the matching of a license or type of license to a particular stakeholder but to try and give some background and definition as to the different types of stakeholders and what type of licenses match to their needs. Ultimately the choice of incoming and outgoing licenses should be a decision to fit with the aims and constraints of the individual project.

References

[1] L. Rosen, Open Source Licensing Software Freedom and Intellectual Property Law. Prentice Hall, 2004.
[2] D. Skidmore, "Free / Libre and Open Source Software: Describing Some Legal, and Software Engineering Terms, and a Taxonomy for Classifying Licenses," in Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives, K. St.Amant and B. Still, Eds. Idea Group, 2007, Chapter 31.
[3] J. Michaelson, "There's no such thing as a Free (software) lunch," ACM Queue, vol. 2, 2004.
[4] Apache Software Foundation, "Apache License, Version 2.0," 2004.
[5] Hacktivismo, "The Hacktivismo Enhanced-Source Software License Agreement," 2005.
[6] Squiz.net, "Squiz.Net Open Source Licence Agreement (Version 1.1),", 2005.
[7] D. L. Parnas, "Software Aging," Proceedings of the 16th international conference on Software engineering, Italy, 1994.
[8] S. Goode, "Something for nothing: management rejection of open source software in Australia's top firms," Information & Management, vol. 42, pp. 669-681, 2005.
[9] K. Edwards, "An economic perspective on software licenses—open source, maintainers and user-developers," Telematics and Informatics, vol. 22, pp. 97-110, 2005.