The Right to a Contribution: An Exploratory Survey on How Organizations Address It

Germán Poo-Caamaño, Daniel German

The Right To a Contribution: An Exploratory Survey On How Organizations Address It

Germán Poo-Caamaño and Daniel M. German

University of Victoria, Canada
gpoo,dmg@uvic.ca

Abstract. Free and Open Source Software (FOSS) projects are characterized by the opportunity to attract external contributors, where contributions can be in any form of copyrightable material, such as code or documentation. In most of them it is understood that contributions would be licensed in similar or compatible terms than the project's license. Some projects require a copyright transfer from the contributor to an organization for the work contributed to a project, such documents are known as copyright assignment agreements. In a way, it is similar to the copyright transfer than some researchers grant to a publisher. In this work we present an exploratory survey of the multiple visions of copyright assignments, and aggregate them in a work that researchers and practitioners could use to get informed of the alternatives available in the literature. We expect that our findings help inform practitioners on legal concerns when receiving external contributions.

1 Introduction

In FOSS projects, contributions can come from anybody. When a project receives a contribution, how can it guarantee that it has the rights to incorporate such contribution into the project?

In this paper, we present findings from our exploratory literature survey on Copyright Assignments and Contribution License Agreements in FOSS projects. The available literature is scattered, no comprehensive body of research has been forthcoming to help practitioners and researchers cope with the different perspectives on the topic of these type of agreements. Reports on proliferation of such agreements in FOSS projects [2,21] motivated us to better understand their differences, the rationale behind their decisions, and the relevant literature. We were also motivated to study the impact of such agreements in the software development practices, and possible consequences for FOSS projects.

The contributions of this paper are an integrated overview of the available material; first, by reviewing the relevant literature on copyright assignments; and, second, by suggesting areas of future research. We expect this document to help FOSS developers both those who control and those who contribute regarding the legal concerns of accepting and submitting contribution.

2 Methodology

We approached this exploratory study as a literature survey on the current research and practices. This study is based on the original guidelines for systematic review in

software engineering by Kitchenham [18]. To find the current practices, we extended our search beyond the scope of journals and conferences. The steps in the systematic review method used are: formulate research questions, carry out a search process, define inclusion and exclusion criteria, and analyze data.

The research questions addressed by this study are:

$Q_{1.}$ How is the term "Copyright Assignment" defined?
$Q_{2.}$ What research topics are addressed in the Copyright Assignments literature?
$Q_{3.}$ What projects and organizations require an agreement?

The search process was a manual search on conference proceedings, journal papers, and projects' websites. We searched peer reviewed papers, essays, copyright assignment and contribution license agreement forms, and FOSS projects' policies on copyright assignments. We searched for *'copyright assignment'* and *'contribution license agreement'* in combination with *'free software'* and *'open source software'*.

The selected literature body should commit to a set of inclusion criteria: *a*) the literature should address the area of contributions agreements, either main or secondary; *b*) be either an agreement form with legal implications (in any format) or have a document body in the form of research paper, essay, or book; and, *c*) be written in English.

We classified the agreements according the their clauses, and grouped them depending on whether they would require transferring the copyright of the contribution or not.

3 Results

In this section, we present the analysis of the literature and discuss the answers to our research questions.

3.1 How is the term "Copyright Assignment" defined?

The term Copyright Assignment has been used in other contexts than FOSS, such as the publishing and music industry. Wu [34] defines Copyright Assignment as "*the transfer of the rights of the author to another party [...] For example, large record labels usually demand that composers assign all of their copyrights to the label*".

In the context of FOSS, we found that few authors define Copyright Assignment. In those cases, Copyright Assignment is defined together with Contribution License Agreement, both of them as a subclass of Contributor Agreement. We, therefore, present the definitions of Contributor Agreement, Copyright Assignment and Contributor License Agreement.

Guadamuz and Rens [14] define Contributor Agreements as "*written contracts in which the contributor either assigns copyright to the project organisers, or grant them a license*". Depending on how the document is drafted, the resulting contract would be either a Copyright Assignment Agreement or Contributor License Agreement. When the contributor assigns copyright—transfer the copyright ownership— to the project owner, it is a Copyright Assignment Agreement. When the contributor grants and irrevocable license—which may be exclusive or non–exclusive— to the project owner to use the contribution, then it is a Contributor License Agreement.

Jakob [15] describes a Contributor Agreement as a way to "*regulate the relationship of the developer with a particular organizational entity*". This agreement could either be a Copyright Assignment Agreement "*whereby the developer transfers and abandons his intellectual property rights in the contribution for the benefit of a project's administration*", or a Contributor License Agreement "*whereby the developer is only required to grant usage rights*".

Maracke [21] describe a Contributor Agreement as a mechanism to "*define and clarify the terms, under which a contribution (code, translation, artwork, etc.) is made to an open source or open content project*". Later, she defines altogether "*Assignment [A]greements require the assignment and therefore transfer of copyright in all contributions to the project owner, while [L]icense [A]greements grant an irrevocable license to allow the project owner to use the contribution"*.

As seen above, Copyright Assignment is usually done via a Copyright Assignment Agreement (CAA). An alternative to the Assignment is a Copyright License Agreement (CLA) where the owner does not transfer the copyright and instead provides a license to the recipient of the contribution.

3.2 What research topics are addressed in the Copyright Assignments literature?

During the analysis of the literature, we could identify five categories or topics covered by the surveyed literature. Table 1 shows a summary of the literature according their topic focus. Below we elaborate on the five categories.

Category	Literature (research papers, essays, books)
Governance	[12,13,24,27,29,30,33]
Community Building	[1,2,13,20,24]
Litigation	[9,13,29,33]
Business Model	[6,13,17,23,29,33]
Proliferation of Agreements	[2,4,7,19,21,25]
Other	[5,28,3,8,10,14,15,16,26,31]

Table 1: Papers according to topics covered.

Governance. Markus [22] defines governance in FOSS as "*the means of achieving the direction, control, and coordination of wholly or partially autonomous individuals and organizations on behalf of an OSS development project to which they jointly contribute*". These papers describe the impact that a CAA or CLA have on the governance of a project.

Wielsch's [33] addresses the challenges that projects face when there are several authors or copyright holders. The legal aspects of licenses and copyright assignment, and the role that some clauses might have in the governance of open collaborative works, not only FOSS projects.

Sometimes the license chosen for a project produces conflicts, such as incompatibilities with other FOSS projects that use other licenses. Such incompatibilities can be an obstacle to reuse code from other projects (either by linking to them, or by copying their source code) [11]. Either changing or adding a small exception to a license may

require a substantial effort to contact and receive consent from all authors (the current copyright holders of the project). Such efforts can be avoided if the copyright is hold by a single entity that can decide the changes unilaterally [13]. Therefore, a single legal owner is a single point of contact and can give flexibility to a project [24], because this also yields power to produce changes to the licensing of the project whenever is needed [13]. The single owner becomes the steward of the project.

Many FOSS projects have informal structures, sometimes without a legal entity or sponsor behind them. The lack of a formal entity creates uncertainty on who has the authority on the project and who takes decisions, particularly, licensing decisions [29]. The legal entity can be a foundation or a company that acts as sponsor of the project. Sponsored projects can be perceived as mature, or at least more than community driven projects. Such maturity can be in the code base, organizational structure of the project, financial backing or a more established base of existing developers [32].

In sponsored projects, the ownership is associated with the degree of responsibility in the project [30], and sometimes, outside contributors can not participate in the design phase that defines the future of the project [32].

Contributors—other than the steward—might consider that this centralization creates uncertainty about the future of the project. The steward may not act predictably according to the interests of the contributors of a project, as organizations depend on the shareholders or management. Although uncommon, there is evidence of organization switching strategies. Caldera, a company known as a contributor of Linux, after being acquired by SCO, switched strategies and litigated copyright infringement in the Linux source code against IBM [24]. After the dissolution of X Consortium, a non-profit organization, the copyright was transferred to its successor organization, The Open Group. The new organization *"changed all the MIT/X Consortium rights to a restrictive copyright"* [12]. Similar concern were raised when Oracle acquired Sun Microsystems; the later had acquired MySQL AB, the original steward of the MySQL project [26].

In a project with disparate ownership each contributor has a valuable ownership right, with different kind of interests in the future of the project. In other words, external contributors, either individuals or companies, may react negatively to sponsored projects perceived as tightly controlled by the copyright holder and not truly participating in collaborative partnerships [24,32].

Changing a license might have a positive outcome, such as facilitate code reuse among different projects. However, the code may also be relicensed in a way that limit its reuse among certain FOSS projects, or even limit the project's ability to reuse FOSS projects. For example, even thought LibreOffice and OpenOffice are both derived from the same code base, each has different licenses; as a consequence LibreOffice can reuse code from OpenOffice, but not the other way around [10].

Community Building. Some organizations decide to release their source code to increase adoption of their product(s), or to encourage the creation of standards around a certain technology, eventually built by them [32]. These papers describe the impact that a CAA or CLA have on community building.

As a copyright holder can license its work any number of times, the steward can license the software under multiple licenses, either FOSS, proprietary (non-FOSS) or

both. Having multiple FOSS licenses can help projects interested in a wide adoption of their code, that is, to be used by other FOSS projects with different licenses.

To build a community, FOSS projects sponsored by an organization can create a tension between the organization and external contributors, because of the sense of control and ownership [32].

For volunteers, it is motivational to have their contributions accepted and merged into the code base. This motivation might decrease if it takes time for the contributions to be incorporated. Although this could happen with or without copyright assignment [1], legal paperwork may undermine the motivation to contribute [13,24,32].

Copyright assignment introduces asymmetry in the relationship between the copyright holder and outside contributors. They are legally unequals [6]. For example, the copyright holder can create a stream of revenue by implementing a dual license for the project. Although a valid approach for business model, it might hurt the ability to nurture a community because the revenue stream is unique to the copyright holder. This inequality creates a barrier to involvement by other contributors, like companies [10,31].

Submitting a contributor agreement can take significant time because those agreements have to be reviewed, approved and signed; and when the contributor is an employee or a company, it is likely to be done by management and legal departments of such organization [2].

Litigation. These papers describe the impact that a CAA or CLA have on litigation.

Rosen [29] covered copyright assignments, dual license and litigation in FOSS projects in his work. He also explains joint work as a way to relicense software and the practice in some FOSS projects to assign ownership to a steward, and some of the risks that a developer could face by assigning the copyright, such as the original authors lose the right to relicense their own work or the perils of assigning the copyright to an informal entity.

License enforcement is often mentioned as the primary reason why organization like the FSF require developers to assign copyright. Under US law, only the owner of a copyright has the right to enforce the license [9,29]. Centralizing the copyright in a single owner simplifies enforcement for the entire code base and, at the same time, releases the contributors of this burden [13]. It also helps in registering copyrights in jurisdictions where it is required [33].

In some cases a copyright assignment is used to circumvent limitations of a FOSS license. For example, to negotiate indemnification when the license in use has patent termination clauses, like MPL1.1 (section 8.2), MPL2.0 (section 5.2), CPL (section 7), OSL/AFL (section 10) or GPLv3 (section 11), among others [29].

License enforcement could be also used to obtain economical advantage. If someone is infringing a GPL code, then the organization holding the copyright could negotiate the terms for a proprietary license. Yet, this might be rarely the desired outcome for developers who are not employees of the organization holding the copyright [13].

Business Model. Licensing the same product under multiple licenses (FOSS and proprietary) might be mechanism used by the licensor to sell proprietary licenses for a fee. These papers describe the impact that a Copyright Assignment has when it is used as a business model.

A licensor could offer two versions of the same product: a commercial one and a FOSS one. The former can be offered with additional features, support, more elaborated warranties or forms of indemnification, or any other negotiation. The latter can be used to promote the software or get external contributions. However, a licensor can only license software for which it holds the copyright or have received permission to license. Therefore, having a single entity holding the copyright of a product can be attractive as a business model [29].

When a single entity holds the copyright of the product, and the product is licensed with a copyleft license, then that entity holds an exclusive power that no one else in the community holds [13]. In this context, the project becomes an asset for the steward. Only the steward can sell and distribute the community's work under creating proprietary licenses (usually adding commercial support, or other services that allow them to monetize their investment in a FOSS project). This asset can be sold total or partially, used to negotiate acquisition, or disposed to the higher bidder in case of bankruptcy" [24]. However, the value might depend on effective ways of verifying the code does not contain "unexpected FOSS or other necessary contractual cover" [17].

However, the circumstance previously discussed does not occur in projects where the license is academic, because in those cases, everyone has equal right to license proprietary versions. This could explain why copyright assignment makes more sense when they are used in projects that use a copyleft licenses [23].

Proliferation of Agreements. There had been a proliferation of copyright and license agreements in FOSS projects, which may share similar goals although with different wording. These papers describe the problem and challenges of proliferation of agreements.

A consequence of having different agreements is that all of them have to be analyzed by lawyers who assist developers, projects, companies and non-profit organizations. Instead, Brock [2] proposed to unify all kind of copyright and license agreements into one, either by using a single modular document with extensive options or multiple documents. However, it raised some criticism because with this initiative the transfer of rights could be perceived as an industry standard [7,19,21].

Metzger [25] studied the duration and territoriality of copyright and license agreements. Engelhardt [4] compared their legal consequences in different jurisdictions, and how they differentiate the wording applied in these agreements. Thus, the enforcement might require adjustments in the agreements to match different local copyright laws. However, such effort can augment the proliferation of agreements, and challenge the initiatives that try to standardize them.

3.3 What projects and organizations require Copyright Assignment?

Because of the proliferation of agreements [2], we searched for organizations that require that a contribution is accompanied by its corresponding copyright agreement (including assignments)[1] before such contribution is accepted and merged. We classified

[1] The agreements forms collected can be found at https://github.com/blindr/contributor-assignments.

them according to their intent, either as CAA or CLA. We examined every agreement (we wanted to avoid cases in which a Copyright Assignment was labelled as a Contributor License Agreement, and the other way around). To enable further research, we included historical agreements that organization does not require anymore, such as OpenOffice, Evolution, Clutter and cogl. Table 2 comprises the agreements separated by type of organizations (non–profit and for–profit), as it has been reported to make a difference [19,24,30].

Type	Projects (Organization)
Non–profit organizations	
CLA	Apache Software Foundation (*HTTPS Server, Tomcat, harmony, ...*), Diaspora (*Diaspora*), Django Software Foundation (*Django*), Eclipse Foundation (*Eclipse*), Mozilla Foundation (*Firefox, ...*), OuterCurve (*NuGet, ASP.NET Ajax Library, ...*), Perl Foundation (*Perl*), The PHP Group (*PHP-PDO*)
CAA	Free Software Foundation (*gcc, emacs, glibc, ...*), Open Source Matters (*Joomla*), The Mambo Foundation (*Mambo*)
For–profit organizations	
CLA	Canonical (*Unity, Bazaar, Launchpad, Upstart, ...*), Google (*Android, Chromium, GWT, V8, Go, ...*), Joyen (*Node.js*), Phrabicator (*Phacility*), Red Hat (*Fedora*), Zend Technologies (*Zend*)
CAA	ArtofCode (*Ghostscript, libart, ...*), Intel (*Clutter, cogl*), Nokia/Digia/Qt Company (*Qt, S60/Symbian, ...*), Novell/Xamarin (*Mono*), Openfiler (UK) Ltd (*Openfiler*), Oracle (*Java, MySQL, OpenOffice.org*), Red Hat (*Cygwin*), Rich Hickey (*Clojure*), VMware (*Zimbra, Open VM tools*), Ximian/Novell (*Evolution*)

Table 2: Organizations (and their respective projects) that require either Copyright Assignment Agreements (CAA) or Contribution. License Agreements (CLA)

4 Discussion

It is dangerous for a FOSS project to assume that it has the rights to distribute an external contribution, in particular if such contribution is significant. As it has been described above, the organizations (and individuals) behind FOSS projects have taken two different approaches to address this issue: assigning copyright (copyright assignment agreements–CAA) or requesting a license to distribute the contribution (copyright licensing agreement–CLA). Each model has its advantages and disadvantages.

CAA centralize the ownership of the project. The steward (its owner) is capable of relicensing the code, either by changing its FOSS license, or relicensing under a proprietary license. This model seems to be preferred by organizations that use copyleft licenses (such as the GPL). Examples of such organizations are: the Free Software Foundation, who requests copyright in order to be able to enforce its copyright (on behalf of its authors); for-profit companies such as Nokia, Sun Microsystems, Oracle and Red Hat request a CAA as part of their business model, which allows them to maintain full ownership of their software, and sell proprietary licenses to it.

On the other hand, projects that use permissive (academic) licenses are more likely to require CLAs. The argument can be made that, because anybody can use the software

to create other proprietary and FOSS software, the copyright does not give them any strategic advantage. As shown in Table 2, most non-profit use CLAs rather than CAAs.

In both models, we can find variations in the wording and the way the agreement must be filled and/or signed. Depending of the jurisdiction, a "technical signature" (for example, click on a button) is sufficient for most cases. In others, a paper based signature is required [14]. Chooding a model might depend of the interests of the project, and the alternatives available in the lost jurisdiction.

It is possible that CAAs reduce the number of potential contributors, since those not willing to assign copyright would not participate. It is also known that forks have been created when the copyright of significant contributions was not assigned (as is the case with xemacs, a fork of emacs). One can argue that organizations that use CLAs instead of CAAs create a more egalitarian environment. Further research is required to fully understand the impact of the choice of agreement.

Some copyright assignments give back a license to the contribution which grants the author the same benefits as a copyright holder. This includes rights not granted by the project's FOSS licenses, such as permission for proprietary relicensing, except granting exclusive rights (since it was already transferred, it is not exclusive anymore [8]). This will allow the author to contribute the code to another project (under a CLA, but not under another CAA) [29].

Another aspect that is worth considering is, how big a contribution should be before one should worry about the right to use it? For instance, does it require permission to redistribute simple bug fixes? The Free Software Foundation does not require copyright assignment for contributions that involves less than 15 lines of code. If a contributor makes a series of repeated small changes, then it could become a significant contribution, for which it would be required to sign a CAA before accepting further contributions [9].

Another aspect corresponds to the legal consequences of copyright as intellectual property. As such, it is inheritable upon death of the owner, the owner might prefer to assign or transfer the rights to an organization that could manage them instead of risking an involuntary lost [29]. In addition, a contributor assignment might last different depending on the territory, according to the expiration of copyright in each legislation.

5 Limitations of this study

This study is based on the guidelines for systematic review by Kitchenham [18], but it deviates from the guidelines in several ways: *a*) the search was manual, and *b*) the search scope was wider than journal and conferences papers, including books, essays, and legal documents. As a consequence, we may have missed some relevant studies or overlooked the literature. However, the consolidation of agreements and the classification of the literature, can help towards an empirical study of the impact of such agreement requirements in FOSS projects.

The authors are not lawyers and they might have misinterpretations of the consequences of the agreements analyzed. Whenever possible, we have provided references to legal expects supporting our assertions. However, this research would benefit from a review by lawyers expert in FOSS.

6 Conclusions and future work

We surveyed and aggregated multiple visions of copyright assignments. We also collected copyright assignments and contributor license agreements forms from multiple FOSS projects. We learned that accepting contributions pose legal challenges to developers, companies and non-profit organizations involved in FOSS projects. There are several alternatives that require an agreement before accepting contributions, each has its benefits and drawbacks. The expectations of the steward of the project and external contributors might differ, and the right solution will depend on the goals of each project and the specifics of each contribution.

Future work should aim at investigating projects that have stopped requiring the copyright assignment for contributions, in particular, the reasons behind such decision. Also important is to study the rationale of projects that were forked because of the requirement of a copyright assignment. Another aspect that requires further study is whether some individuals might decide not to contribute due to such agreements, and, similarly, what is the impact of such agreements in the dynamics between contributors and the steward of a project. Overall, we expect that our findings help inform practitioners about legal concerns when receiving external contributions, and enable researchers and practitioners to get informed of the alternatives available in the literature.

References

1. Allyn, M.R., Misra, R.B.: Motivation of Open Source Developers. Intl. Journal of Open Source Software and Processes 1(4), 65–81 (2009)
2. Brock, A.: Project Harmony: Inbound transfer of rights in FOSS Projects. Intl. Free and Open Source Software Law Review 2(2), 139–150 (2010)
3. Brock, A.: Understanding Commercial Agreements with Open Source Companies. In: Thoughts on Open Innovation: Essays on Open Innovation from leading thinkers in the field, pp. 119–139. OpenForum Europe LTD for OpenForum Academy (2013)
4. Engelhardt, T.: Drafting Options for Contributor Agreements for FOSS: Assignment, (Non)Exclusive Licence and Legal Consequences. A Comparative Analysis of German and US Law. SCRIPTed 10(2), 149–176 (2013)
5. Fogel, K.: Producing Open Source Software: How to Run a Successful Free Software Project. O'Reilly Media, Inc. (2005)
6. Fontana, R.: Contributor Agreements Considered Harmful. Audio of talk given at OSCON 2011. Online (Jul 2011), http://faif.us/cast/2011/aug/30/0x17/
7. Fontana, R.: The Trouble With Harmony. Online (Jul 2011), http://opensource.com/law/11/7/trouble-harmony-part-1
8. Fontana, R., Kuhn, B.M., Moglen, E., Norwood, M., Ravicher, D.B., Sandler, K., Vasile, J., Williamson, A.: A Legal Issues Primer for Open Source and Free Software Projects. Software Freedom Law Center (2008)
9. Free Software Foundation: Information for maintainers of GNU software. Online (Nov 2012), http://www.gnu.org/prep/maintain/, accessed: 08/12/2012
10. Gamalielsson, J., Lundell, B.: Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved? Journal of Systems and Software 89, 128–145 (2014)
11. German, D.M., Hassan, A.E.: License integration patterns: Addressing license mismatches in component-based development. In: 31st Int. Conf. on Soft. Eng., ICSE. pp. 188–198 (2009)

12. Gettys, Jim: Copyright Assignments. Online (Jul 2000), https://mail.gnome.org/archives/foundation-list/2000-July/msg00332.html, accessed: 09/12/2012
13. GNOME: GNOME Foundation Guidelines on Copyright Assignment. Online (Jul 2010), https://live.gnome.org/CopyrightAssignment/Guidelines, accessed: 09/12/2012
14. Guadamuz, A., Rens, A.: Comparative Analysis of copyright assignment and licence formalities for Open Source Contributor Agreements. SCRIPTed 10(2), 207–230 (2013)
15. Jakob, S.F.: A Qualitative Study on the Adoption of Copyright Assignment Agreements (CAA) and Copyright License Agreements (CLA) within Selected FOSS Projects. JIPITEC 5(2), 105–115 (2014)
16. Jensen, C., Scacchi, W.: License Update and Migration Processes in Open Source Software Projects. In: Open Source Systems: Grounding Research - 7th IFIP WG 2.13 Int. Conf., OSS 2011. vol. 365, pp. 177–195. Springer Berlin Heidelberg (2011)
17. Kemp, R.: Current developments in OSS. Computer Law & Security Review 25(6), 569–582 (Nov 2009)
18. Kitchenham, B.: Procedures for performing systematic reviews. Tech. rep., Keele University (TR/SE-0401) and National ICT Australia Ltd. (0400011T.1) (2004)
19. Kuhn, B.: Project Harmony Considered Harmful. Online (Jul 2011), http://ebb.org/bkuhn/blog/2011/07/07/harmony-harmful.html
20. Lerner, J., Tirole, J.: Some Simple Economics of Open Source. The Journal of Industrial Economics 50(2), 197 – 234 (2002)
21. Maracke, C.: Copyright Management for Open Collaborative Projects –Inbound Licensing Models for Open Innovation. SCRIPTed 10(2), 140–148 (2013)
22. Markus, M.L.: The governance of free/open source software projects: monolithic, multidimensional, or configurational? Journal of Management & Governance 11(2), 151–163 (2007)
23. Mcgowan, D.: Legal aspects of FOSS. In: Perspectives on FOSS, chap. 19, pp. 361–391. MIT Press (2007)
24. Meeks, M.: Some thoughts on copyright assignment. Online (Dec 2009), http://people.gnome.org/~michael/blog/copyright-assignment.html, accessed: 27/04/2011
25. Metzger, A.: Internationalisation of FOSS Contributory Copyright Assignments and Licenses: Jurisdiction-Specific or "Unported"? SCRIPTed 10(2), 177–206 (2013)
26. Moglen, E.: Software Freedom Law Center Opinion on the Oracle/Sun Merger (2009), https://softwarefreedom.org/news/2009/dec/04/software-freedom-law-center-submits-opinion-oracle/
27. O'Mahony, S.: Guarding the commons: how community managed software projects protect their work. Research Policy 32(7), 1179–1198 (Jul 2003)
28. O'Mahony, S.: Nonprofit Foundations and Their Role in Community-Firm Software Collaboration. In: Perspectives on FOSS, chap. 20, pp. 393–413. MIT Press (2005)
29. Rosen, L.: Open Source Licensing: Software Freedom and Intellectual Property Law. Prentice Hall (2004)
30. Shuttleworth, Mark: The responsibilities of ownership. Online (Jul 2011), http://www.markshuttleworth.com/archives/687, accessed: 09/12/2012
31. Smith, D., Alshaikh, A., Bojan, R., Kak, A., Kohe, J.M.: Overcoming Barriers to Collaboration in an Open Source Ecosystem. Technology Innovation Management Review 4(January), 18–27 (2014)
32. West, J., O'Mahony, S.: Contrasting Community Building in Sponsored and Community Founded OSS. In: Proc. of the 38th Intl. Conf. on System Sciences. p. 196c. IEEE (2005)
33. Wielsch, D.: Governance of Massive Multiauthor Collaboration. Journal of Intellectual Property, Information Technology and E-Commerce Law 1(2), 96–108 (2010)
34. Wu, T.: On Copyright's Authorship Policy. The University of Chicago Legal Forum (2008), 335–354 (2008)