License Update and Migration Processes in Open Source Software Projects

Chris Jensen
Institute for Software Research
University of California, Irvine
Irvine, CA USA 92697-3455
Phone: +1 (949) 824-0573
Email: cjensen@ics.uci.edu

Walt Scacchi
Institute for Software Research
University of California, Irvine
Irvine, CA USA 92697-3455
Phone: +1 (949) 824-4130
Email: wscacchi@ics.uci.edu

Abstract

Open source software (OSS) has increasingly been the subject of research efforts. Central to this focus is the nature under which the software can be distributed, used, and modified and the causes and consequent effects on software development, usage, and distribution. At present, we have little understanding of, what happens when these licenses change, what motivates such changes, and how new licenses are created, updated, and deployed. Similarly, little attention has been paid to the agreements under which contributions are made to OSS projects and the impacts of changes to these agreements. We might also ask these same questions regarding the licenses governing how individuals and groups contribute to OSS projects. This paper focuses on addressing these questions with case studies of processes by which the Apache Software Foundation's creation and migration to Version 2.0 of the Apache Software License and the NetBeans project's migration to the Joint Licensing Agreement.

Keywords

Open source, license evolution, process, Apache, NetBeans

Introduction

Software process research has investigated many aspects of open source software (OSS) development in the last several years, including release processes,

communication and collaboration, community joining, and project governance. The central point of Lawrence Lessig's book "Code" is that the hardware and software that make up cyberspace also regulate cyberspace. He argues that code both enables and protects certain freedoms, but also serves as to control cyberspace. Software licenses codify these freedoms and regulations by setting forth the terms and conditions for software use, modification, and distribution of a system and any changes made to it. For that reason, others have suggested that licenses serve as contracts for collaboration. In the case of non-OSS licenses, that contract may indicate no collaboration, but rather strict separation between users and developers.   OSS licenses, by contrast range widely in permissiveness, some granting more rights to the original authors and some granting more rights to consumers of OSS software.   While research has examined OSS licenses to great detail, we are only beginning to understand license evolution. Just as OSS code is not static, neither are the licenses under which it is distributed. Research into license evolution is just beginning. However, when licenses change, so too the contracts for collaboration change.   This paper seeks to provide an incremental step to understanding how changes in software licensing impact software development processes.

Why does understanding license update and migration matter? Companies using OSS software need to know how changes affect their use, modification, and distribution of a software system. License compatibility in OSS has long been a topic of debate. Research is only beginning to provide tools for assistance in resolving software license compatibility [1]. OSS project participants need to understand why changes are being made, whether the changes align with their values and business models (e.g., enabling new avenues of license compatibility offering strategic benefit or opening up new channels of competition). As a project sponsor or host, you may be concerned about how to best protect both the software system and your user community, but also your business model. You typically want a license that will attract a large number of developers to your project [2] while at the same time allowing you to make a profit and stay in business.

While licenses such as the GNU General Public License (GPL), the Berkeley Software Distribution (BSD) license, and the Apache License are well known, we rarely consider another type of license agreement critical to understanding collaboration in OSS projects: individual contributor license agreements (CLAs) and organizational contributor license agreements (OCLAs), for contributos from organized entities. In non-OSS software development, the contract for collaboration is typically an employment contract, often stating that all intellectual property rights pertaining to source code written by an employee are property of the employer. This provides the employer with complete control of the rights granted of licensed software. In OSS development, you have a situation where multiple developers are contributing to a software system. Without copyright assignment or a CLAs, changing a software license requires the consent of every contributor to that system. We observed this situation in the case of the Linux kernel, which suggested that

without a CLA, license evolution can become inhibited or prevented as the number of contributors, each with differing values and objectives, increases. To understand how changes in software licenses affect software development processes, we must also investigate changes in CLAs.

We address these issues with two case studies. The first examines the creation and deployment of the Apache Software License, Version 2.0. The second looks at an update to the contributor license agreement in the NetBeans project.

Background Work

Legal scholars, such as St. Laurent [3] and Larry Rosen [4], former general counsel and secretary for the Open Source Initiative (OSI), have written extensively on license selection. They note that quite often, the choice of license is somewhat outside the control of a particular developer. This is certainly the case for code that is inherited or dependent on code that is either reciprocally licensed, or at the very least, requires a certain license for the sake of compatibility. However, outside such cases, both St. Laurent and Rosen advocate for the use of existing and well-tested, well-understood licenses as opposed to the practice of creating new licenses. Such license proliferation is seen as a source of confusion among users and is often unnecessary given the extensive set of licenses that already exist for a diverse set of purposes. Lerner and Tirole [5] observe specific determinant factors in license selection. Of the 40,000 Sourceforge projects studied, projects geared towards end-users tended towards more restrictive license terms, while projects directed towards software developers tended towards less restrictive licenses. Highly restrictive licenses were also found more common in consumer software (e.g., games) but less common for software on consumer-oriented platforms (e.g., Microsoft Windows) as compared to non-consumer-oriented platforms. Meanwhile, Rosen specifically addresses the issue of relicensing, commenting that license changes made by fiat are likely to fracture the community. This case of relicensing is exactly the focus of our case studies here.

The drafting and release of the GNU General Public License, Version 3.0 was done in a public fashion, inviting many prominent members of the OSS community to participate in the process. In fact, we even see a sort of prescriptive process specification outlining, at a high level, how the new license was to be created. This license revision process is interesting from the perspective that the license in question is not used by one project or one foundation, but rather is an update of the most commonly used open source license in practice. As such the process of its update and impact of its revision on software development is both wide ranging and widely discussed.

Di Penta, et al. [6], examined changes to license headers in source code files in several major open source projects. Their three primary research questions sought to

understand how frequently licensing statements in source code files change, the extent of the changes, and how copyright years change in source code files. Their work shows that most of the changes observed to source code files are small, though even small changes could signify a migration to a different license. The authors also note that little research available speaks to license evolution, pointing to the need for greater understanding in this area.

Lindman, et al., [2] examine how companies perceive open source licenses and what major factors contribute to license choice in companies releasing open source software. The study reveals a tight connection between business model, patent potential, the motivation for community members to participate in development, control of project direction, company size, and network externalities (compatibility with other systems) and licensing choice.
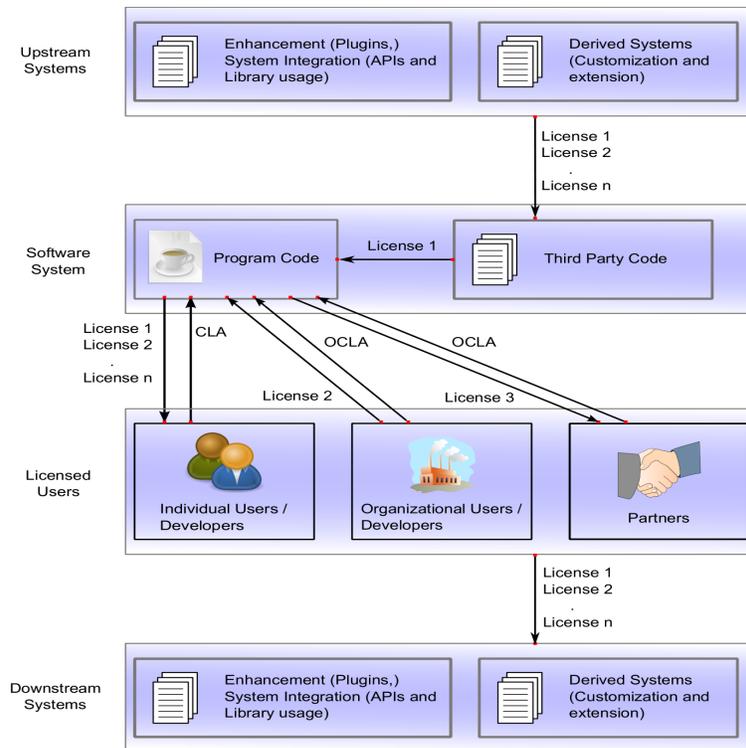

Figure 1. A model of software production and consumption with open source licensing

Lindman, et al., provide a model of a software company, its developers, and users in the context of an OSS system developed and released from a corporate environment [2]. However, few systems are developed in complete isolation.

Rather, they leverage existing libraries, components, and other systems developed by third parties. Moreover, as Goldman and Gabriel point out, open source is more than just source code in a public place released under an OSS license [7]; communities matter. Figure 1 shows the production and consumption of open source software, highlighting the impact of software licenses and contributor license agreements.

Going a step further, Oreizy [8] describes a canonical high-level software customization process for systems and components, highlighting intra-organizational software development processes and resource flow between a system application developer, an add-on developer, a system integrator, and an end user.

Similarly, we have examined such concepts in the context of software ecosystems [9] in the context of process interaction. Software license change can precipitate integrative forms of process interaction in the case of dual and multi-licensing by enabling new opportunities for use of software systems upstream of a project to provide added functionality or support, as well as projects downstream vis a vis use as a library, plugin development, support tool development, and via customization and extension. In such cases, software source becomes a resource flowing between interacting projects. However, license change can also trigger interproject process conflict if new license terms render two systems incompatible. At that point, the resource flow between projects can be cut off, when downstream consumers of software source code no longer receive updates. A more common example with non-OSS is license expiration. License-based interproject process conflicts can also manifest as unmet dependencies in software builds or an inability to fix defects or add enhancements to software, resulting in process breakdown, and failing recover, project failure. OSS licenses, however, guarantee that even when conflict occurs, recovery is possible because the source is available and can be forked.

Methodology

The case studies in this report are part of an ongoing, multi-year research project discovering and modeling open source software processes. Our research methodology is ethnographically informed, applying a grounded theory to the analysis of artifacts found in OSS projects. The primary data sources in this study come from mailing list archives of the Apache and NetBeans projects.

Our primary data sources were mailing list messages. However, we also found supplementary documentation on each project's websites that served to inform our study. These supplementary documents were often, though not always referenced by the messages in the mailing list. Cases regarding the NetBeans project all took place between April and June of 2003, involving over 300 email messages, whereas the Apache cases were spread over several discrete time periods and consisted of more than 350 messages.

Case selection happened in two ways. For NetBeans, the cases arose during our study of requirements and release processes, having stood out as prominent issues facing the community during the time period studied. Although we observed additional incidents appropriate for discussion, the three cases selected fit together nicely as a cohesive story. This approach was also used in the study of the Apache project. However, due to a lower incident frequency, we expanded our study over a longer time period to find incidents that proved substantial. As a testament to the nature of project interaction, issues raised in mailing list discussions proved to be short-lived, either because they were resolved quickly or because the conversation simply ceased. It is possible to suggest this is the normal behavior pattern for both projects. A few issues proved outliers, having more focused discussions, and these were selected for further study. We also observed a tendency for discussions to play out in a series of short-lived discussions sessions. A topic would be raised, receiving little or no attention. Then, at a later time, it would be raised again. The JCA discussion in NetBeans and Subversion migration discussion in the Apache project demonstrated such conversational resurgence. We observed, in general, that discussion topics carry certain conversational momentum. Topics with a high degree of momentum tended to have lengthier discussion periods or frequent discussion sessions until fully resolved or abandoned while topics with a low degree of momentum were addressed quickly or simply died off. The causes and factors affecting changes in momentum were not investigated as they laid too far afield from the focus of this study. We do note that although *consensus by attrition* has been cited in other communities (e.g., [10 and 11]), we did not observe it in effect in any of the cases studied, but rather that the primary participants in discussions remained active in their respective projects for several months following the reported incidents. The creation of the Apache License, version 2.0 was directed to us by a colleague familiar with the project. Data for the Apache licensing case was gathered from email messages sent to a mailing list established for the purpose of discussing the proposed changes.

Considering the difficulties we experienced with building our own search engine to support process discovery, we still faced the challenge of keeping track of process data once we found it as we were building our models. Up until this point, our approach to providing process traceability was simply to include links to project artifacts in our models. However, this strategy did not help us build the models, themselves. We returned the search problem back to the projects, themselves using their own search engines to locate process data, looking for more lightweight support for discovery.

Our current strategy for providing computer support for process discovery returns to using each project's own search engine to locate process information. We have operationalized the reference model as an OWL ontology with the Protégé ontology editor [12], using only the OWL class and individual constructs to store process concepts and their associated search queries respectively. Secondly, we built a Firefox plugin, Ontology [13], to display the reference model ontology in the Firefox web browser. Next, we enlisted the Zotero citation database Firefox plugin

[14] to store process evidence elicited from project data, integrating the two plugins such that each datum added to the citation database from a project artifact is automatically tagged with the selected reference model entities.

The use of a citation database as a research data repository may seem unintuitive. Zotero, however, has proven well suited for our needs. Like many Firefox plugins, Zotero can create records simply from highlighted sections of a web document, though the creation of arbitrary entries (not gleaned from document text selections) is also possible. It can also save a snapshot of the entire document for later review, which is useful given the high frequency of changes of some web documents- changes that evidence steps in a software processes. The tag, note, and date fields for each entry are useful for recording reference model associations and memos about the entry for use in constructing process steps and ascertaining their order. A screenshot of Zotero with Ontology appears in Figure 2.
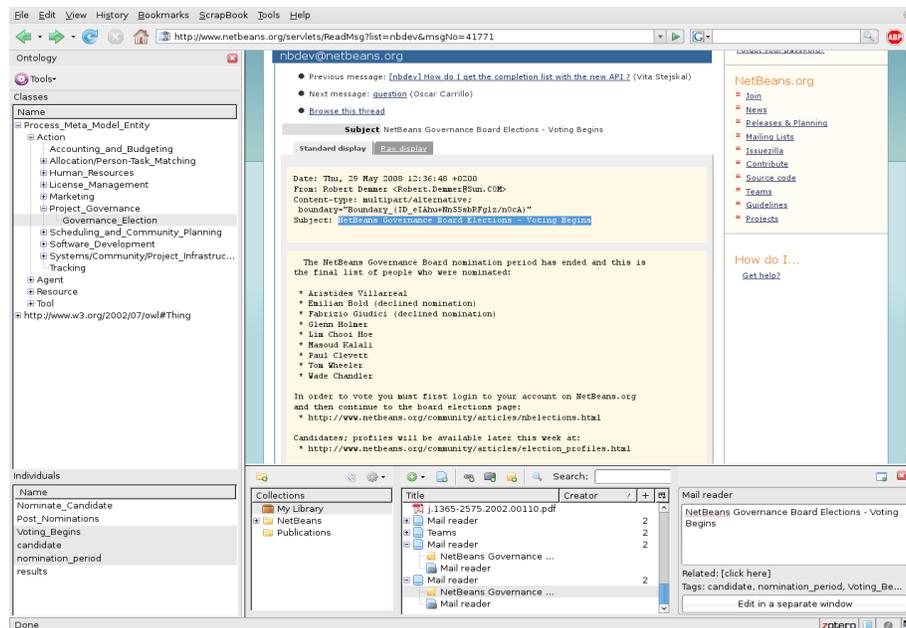

Figure 2. Data capture in Firefox with Zotero and Ontology

The plugin integration greatly facilitates the coding of process evidence and provides traceability from raw research data to analyzed process models. As the tool set is browser-based, it is not limited to analysis of a particular data set, whether local or remote. Moreover, the tool set does not limit users to a single ontology or Zotero database, thereby allowing users to construct research models using multiple ontologies describing other (e.g. non-OSS process) phenomenon and reuse the tool set for analysis of additional data sets. Thus, it may be easily appropriated for grounded theory research in other fields of study.

The elicitation of process evidence is still search driven. Rather than use one highly customized search engine for all examined data repositories, the search task has been shifted back to the organizations of study. This decision has several implications in comparison with the previous approach, both positive and negative. Using an organization's own search engine limits our ability to extract document-type specific metadata, however among the organizations we have studied, their search tools provide greater coverage of document and artifact types than Lucene handled at that time. Furthermore, this approach does not suffer the data set limitations imposed by web crawler exclusion rules. The ability to query the data set in a scripted fashion has been lost, yet some scientists would see this as a gain. The use of computer-assisted qualitative data analysis software (CAQDAS) historically has put into question the validity of both the research method and results [15,16].

This tool was still quite unfinished as we began governance process discovery and modeling. As we added functionality, we had to return to some of our data sources and recapture it. Although we have high hope to use the integrated timeline feature to assist in process activity composition and sequencing, the time and date support within Zotero's native date format was insufficiently precise.   With provisions only for year, month, and day, there is no ability to capture action sequences that happen on the same day. After adding support for greater date and time, we found having to enter the date and time for every piece of data we captured rather tedious. Eventually we have had to prioritize completion of discovery and modeling ahead of computer-support for process discovery, and we had to disable the time and date entry. Unable to utilize Zotero to our intended effect in discovery and modeling, our efforts with Zotero remain in progress, pending usability improvements.
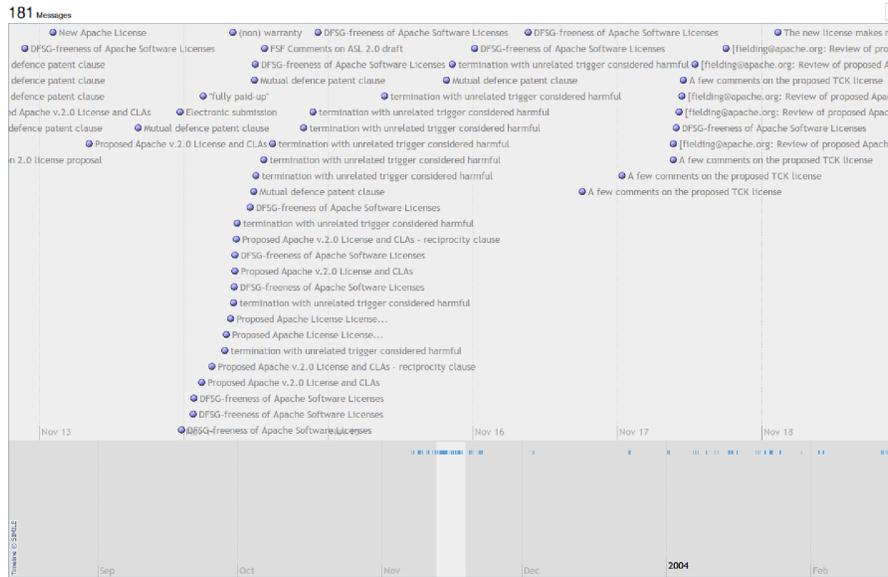
Figure 3. Timeline of the Review and Approval of the Apache License, Version 2.0

Creation and Migration to the Apache License, Version 2.0

The Apache Software Foundation created a new version of their license in the end of 2003 and beginning of 2004. Roy Fielding, then director of the ASF, announced the license proposal on 8 November 2003 [17], inviting review and discussion on a mailing list set up specifically for said purpose. Per Roy's message, the motivations for the proposed license included

- Reducing the number of frequently asked questions about the Apache License.

- Allowing the license to be usable by any (including non-Apache) projects

- Requiring a patent license on contributions that necessarily infringe the contributor's own patents

- Moving the full text of the license and specific conditions outside the source code

Roy further indicated a desire to have a license compatible with other OSS licenses, notably the GPL.

As you can see from Figure 3, most of the discussion took place in mid November of 2003. In fact, given that the ApacheCon conference that ran from 16-19 November, we can see a high message density in the days leading up to ApacheCon, with a steady rate continuing on for a few days afterward. Beyond this, the frequency becomes sparse. An update to the proposed license was announced on 24 December 2003, after some internal review, a part of the process that is not publicly visible. This update prompted a brief discussion. A second active time period is observable in January 2004, when Fielding announces a final update (20 January 2004) and that the final version of the license has been approved by the board [18 and 19] (21 January 2004).

The primary discussion point of the creation and migration to the 2.0 version of the Apache License centered around a patent clause in the proposed license. According to Brian Behlendorf, who was serving on the ASF board of directors at the time, the ASF's patent-related goals were to "prevent a company from sneaking code into the codebase covered by their own patent and then seeking royalties from either the ASF or end-users" [20]. The clause in question read:

> 5. Reciprocity. If You institute patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to You under this License shall terminate as of the date such litigation is filed. In addition, if You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work itself (excluding combinations of the Work with other software or hardware) infringes Your patent(s), then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed. [21]

Consequences of this clause sparked discussion in a few areas, mainly surrounding the first sentence of the clause regarding license termination. Legal representatives from industry stated objections to losing usage rights for patent litigation regarding any software, even software unrelated to that covered by the license [22], proposing alternative wordings to achieve the stated license goals but restricting the trigger to litigation pertaining to patents covered by the ASF licensed code [23]. Uncertainty regarding the roles of people in the license revision process [24] and proposed changes [25] created additional confusion regarding the patent reciprocity stance.

Eben Moglen, General Counsel for the Free Software Foundation (FSF), adds that the first sentence of the license clause carries great risk for unintended and serious consequences, and is an inappropriate vehicle for protecting free software against patent litigation [26]. As such, the FSF has deemed the clause causes the license to be incompatible with version 2 of the GPL, failing one of the goals of the proposed Apache License.

Brian Carlson reports that the Debian community's consensus is that the proposed license does not meet the criteria for *Free Software Licenses* under the Debian Free Software Guidelines [27]. Consequently, code licensed as such would sandboxed into the non-free archive, and therefore, not automatically built for Debian distributions, nor receive quality assurance attention. Again, the license termination aspect of the reciprocity clause is cited as the critical sticking point [28], with several members of the Debian community arguing that free software licenses should only restrict modification and distribution, but not usage of free software.

The patent reciprocity clause was not entirely rejected. There was support for extending it to provide mutual defense against patent litigation attacks against all open source software [29]. The idea was quickly nixed on the grounds that it could lead to users being attacked and unable to defend themselves if someone were to maliciously violate a user's patent on an unrelated piece of software and create an open source version. In such a scenario, the user would have to choose between using Apache licensed software and losing all their patents [30].

On 18 November, Fielding indicates that there have been "several iterations on the patent sentences, mostly to deal with derivative work" [24], mentioning he will probably include the suggested changes in the patent language recommended by one of the legal representatives from industry. Fielding notes that he has been in contact with representatives from other organizations, among them Apple, Sun, the OSI, Mozilla, and a few independent attorneys, although the details of these portions of the process remain hidden.

The next milestone in the process occurs on 24 December, when Fielding mentions that a second draft, version 1.23, has been prepared after internal review due to extensive changes [31], and has been posted to the proposed licenses website [32] and the mailing list. The new proposed license [33] incorporates many of the proposed changes, including the removal of the contested first sentence of the patent reciprocity clause, leaving the generally agreed upon patent termination condition:

If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

The 1.23 version of the license received little feedback on the license discussion mailing list. Aside from definition clarifications, there was an inquiry about GPL compatibility. Behlendorf commented that Moglen's suggestions had been incorporated to address the two issues with GPL compliance, but he had been contacted earlier in the week to take a look at the current draft [34]. As a result, Behlendorf (on 7 January 2004) offers that the issues presented have been addressed to his satisfaction and is willing to propose the license to the board at the January 2004 meeting [35]. However, before the board meeting, Fielding announces a version 1.24, featuring a change to the definition of "Contributor" [36] and a 1.25

version very shortly thereafter to address the way "Copyright" is represented due to various laws and the use of "(C)" to indicate copyright [37]. Finally, the Apache License, Version 2.0 was approved by the ASF board by a unanimous vote on 20 January 2004 [18] and announced to the mailing list by Fielding the following day [19]. Per the board meeting minutes:

> WHEREAS, the foundation membership has expressed a strong desire for an update to the license under which Apache software is released,
>
> WHEREAS, proposed text for the new license has been reworked and refined for many, many months, based on feedback from the membership and other parties outside the ASF,
>
> NOW, THEREFORE, BE IT RESOLVED, that the proposed license found at http://www.apache.org/licenses/proposed/LICENSE-2.0.txt is officially named the Apache Software License 2.0. To grant a sufficient transition time, this license is to be used for all software releases from the Foundation after the date of March 1st, 2004.

The conversation continued on, briefly, to address two points. Firstly, a return to the GPL compatibility discussion. Don Armstrong requested verification as to whether Moglen/the FSF has identified the license as GPL compatible (Fielding's announcement claimed it was) [38]. Fielding responds, saying Moglen sent a private communication commenting on the license compatibility, and furthermore, that it was the belief of the ASF that "a derivative work consisting of both Apache Licensed code and GPL code can be distributed under the GPL," and, as such, there wasn't anything further to consider, as far as the ASF was concerned [39]. Incidentally, the FSF standing is that due to the patent issue, the Apache license 2.0 is GPL3 compatible but not GPL2 compatible [40]. Secondly, Vincent Massol requested information about moving his Apache sub-project to the ASL2 license and what file license headers should be used [41], to which Behlendorf responds [42]. A flow graph of the License creation and migration process appears in Figure 4.
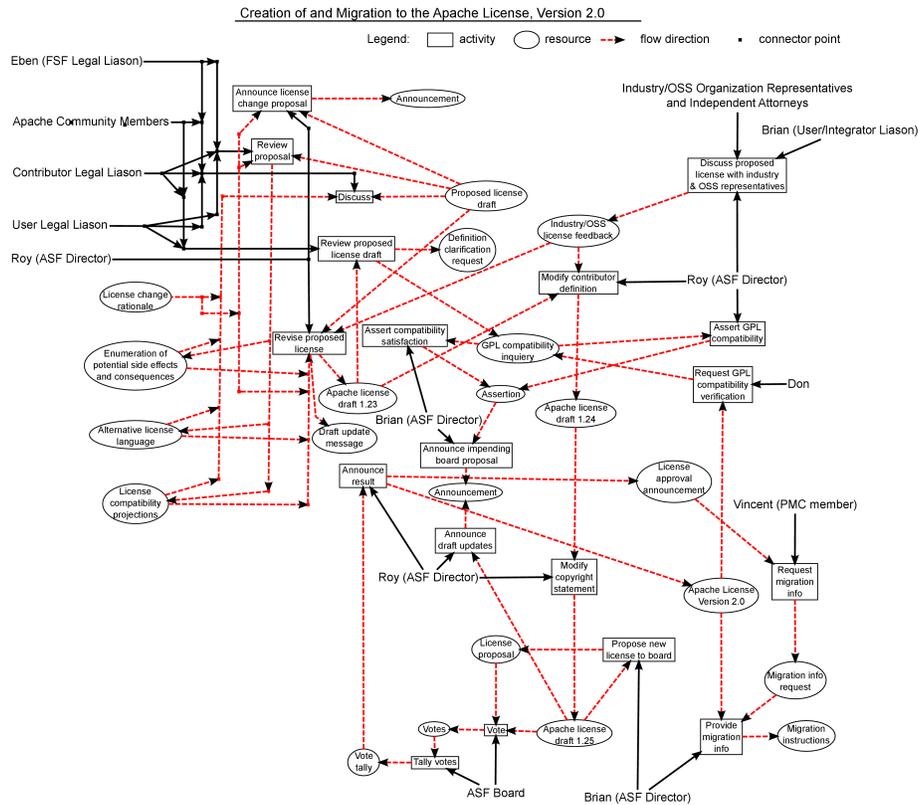
Figure 4. Process flow graph for Apache License Version 2.0 creation

Introduction of the Joint License Agreement

Rosen [4] suggests that copyright assignment is sought for two purposes:

1. So the project can defend itself in court without the participation and approval of its contributors.

2. To give the project (and not the contributor) the right to make licensing decisions, such as relicensing, about the software

The NetBeans case is interesting because it is not simple copyright assignment, but rather affords both the contributor and the project (Sun Microsystems, specifically) equal and independent copyright to contributed source.

The Joint License Agreement (JLA) was introduced to the NetBeans project on 28 April 2003 by Evan Adams, a prominent project participant working for Sun Microsystems [43]. Adams states that the JLA was being introduced in response to

observations by Sun's legal team of Mozilla and other open source projects and believed that Sun required full copyright authority to protect the NetBeans project from legal threats and provide Sun with the flexibility to adapt the NetBeans license over time. Under the proposed agreement, contributors (original authors) would retain all copyrights independently for project contributions and any previous contributions whose authors did not agree to the terms of the JCA would have to be removed from the source tree. The discussion spanned ninety messages from seventeen individuals over nearly two months, with a follow-up discussion consisting of forty six messages from fourteen individuals (eleven of whom participated in the earlier discussion) over a third month. The discussion, which began at the end of April 2003 continued through July (with a few sporadic messages extending out to September), long after the deadline for requiring JLA for project contributions.
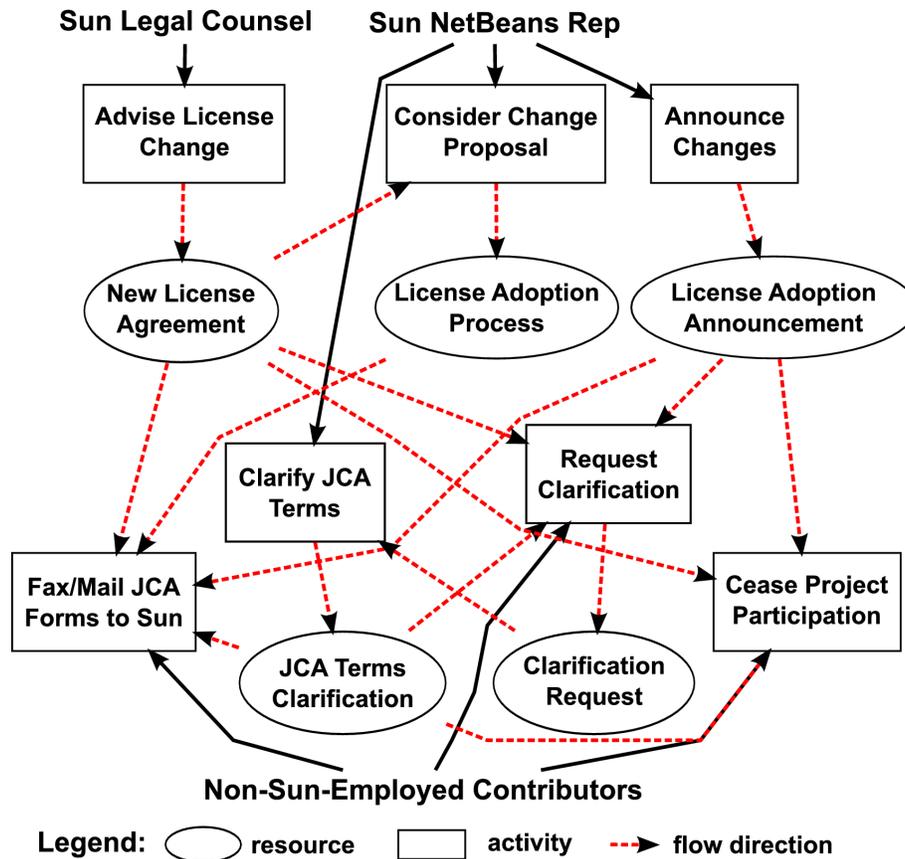
Figure 5. NetBeans JCA introduction process flow graph

The process for the license format change seems simple. The particulars of the proposed license received early focus in the discussion. As the discussion progressed, concern shifted away from details of the license agreement to the way in which the change was proposed. In the course of discussion, it was revealed that switching to the JLA was an idea proposed by the Sun legal counsel and the decision to adopt it was done internally, unilaterally, and irrevocably by Sun without the involvement of the project, at large. The adoption decision raised questions regarding the decision rights and transparency within the project.

While recognizing that Sun-employed contributors were responsible for a majority of project effort, non-Sun contributors took the lack of transparency and consideration in the decision making process as disenfranchisement. In a follow-up discussion, project members further expressed fears that giving Sun full copyright of contributed code could lead to reclassification of volunteer-contributed code in

objectionable ways. More significantly, they feared the change could impact copyright of projects built upon the NetBeans codebase, but not contributed back to NetBeans source repository.

In time, most of the "corner case" concerns about the license agreement were addressed. However, ultimately non-Sun employed contributors were still in the position of having to trust Sun to act in an acceptable manner with a grant of full copyright. Moreover, the discussion drew out larger concerns regarding Sun's role position of leadership and control of the project, and regarding transparency in decision making. A flow graph of the JCA introduction process appears in Figure 5.

Discussion and Conclusions

The two cases presented are not directly comparable. The Apache study looks at the process of creating a new license, to be used by all projects under the domain of the Apache Software Foundation. The NetBeans study focuses on the adoption of a new license agreement for contributors to the NetBeans IDE and platform. Software source licenses govern the rights and responsibilities of software consumers to (among other things) use, modify, and distribute software. Contributor license agreements (CLAs), on the other hand, govern the rights and responsibilities to (among other things) use, modify, and distribute contributions of the organization to which the contributions are submitted, and those retained by the contributor. The new CLA stated that copyright of project contributions would be jointly owned by the originating contributors, as well as the project's benefactor, Sun Microsystems. Code contribution agreements may not be of interest to end users of software executables. However, the OSS movement is known for its tendency towards user-contributors; that is, users who contribute to the development of the software and developers who use their own software.

If we consider, specifically, the license changes in the Apache and NetBeans projects, both were introduced as inevitable changes by persons of authority in each project (founder Roy Fielding of Apache and Evan Adams of Sun Microsystems for NetBeans). The initiators of the discussion both presented the rationale for making the changes. For Apache, the move was motivated by a desire to increase compatibility with other licenses, reduce the number of questions about the Apache license, moving the text outside the source code, and require patent license on contributions where necessary. For NetBeans, the motivations were to protect the project from legal threats and provide Sun the ability to change the license in the future. In the Apache case, the motivations for making the changes went unquestioned. The discussion focused on what objectives to achieve with the change and how best to achieve them. The former had to do with a (minority) subset of participants who saw the license change as an opportunity to affect software development culture, altering the direction of the software ecosystem as a means of governance on a macro level. The latter had to do with making sure the verbiage of

the license achieved the intended objectives of the license without unintended consequences (such as those whose nature was of the former). In the NetBeans case, the discussion focused on the differences between the licenses and their affect on non-sponsoring-organization participants (meso-level project governance) of the license. Given the context of the surrounded cases, the structural and procedural governance of the project was also questioned.

The area of the NetBeans license change that received the greatest push-back was granting the sponsoring organization the right to change the license unilaterally at any point in the future. This right was similarly granted to the ASF in the Apache contributor license agreement (CLA) [44], a point that was not lost on participants in the NetBeans license change discussions [45]. Why did this issue receive push-back in NetBeans and not Apache? West and O'Mahony [46] suggest caution that, unlike community-initiated projects, sponsored OSS projects must achieve a balance between establishing pre-emptive governance design (as we saw here) and establishing boundaries between commercial and community ownership and control. The surrounding cases served to create an atmosphere of distrust within the project. The distrust led to fears that contributions from the sponsoring organization would become closed off from the community, perhaps saved for the organization's commercial version of the product, leaving the sponsoring organization as free-riders [47 and 48] profiting off of the efforts of others without giving back [49] or otherwise limit what project participants can do with project code.

Perhaps the most striking difference in the way the two license changes were introduced is that the Apache case invited project participants (as well as the software ecosystem and the public, at large) to be a part of the change, whereas the NetBeans case did not. Participants in the NetBeans project were left without a sense of *transparency in the decision-making process* in that the change was put on them without any warning before the decision was made. Moreover, they were left without *representation in the decision-making process* in that they did not participate in determining the outcome of a decision that had a large impact on them. This is not to say that the Apache case was entirely transparent. There are clear indications from the messages on the list that conversations were held off-list. Likewise, there were misconceptions over what roles participants played and participant affiliation. However, the process was not questioned, nor the result.

In conclusion, we have taken a first step to understanding how license change processes impact software development processes by discovering and modeling the update process for the Apache License and the update to the contributor license agreement in the NetBeans project. We observed how differences in the processes in introducing change intent influenced response to the changes. To put these cases into context, NetBeans underwent two license changes since the events described above, neither of which received significant push-back from the community. The first shifted the license to the CDDL. The second was a move to dual license NetBeans under the GPLv2. This second licensing shift was considered by Sun "at the request from the community" [50]. Unlike the introduction of the JCA, the GPL

shift was presented to the community by Sun for feedback (in August 2007) as an added option (rather than a complete relicensing) before the change was made. Thus, we can clearly see further change in the processes used to govern the community in a way that directly addressed the defects in the project's governance processes circa 2003.   Shah [51] echoes these concerns, observing that code ownership by firms creates the possibility that non-firm-employed contributors will be denied future access to project code. In other projects, these threats can lead to forking of the source, as happened when the MySQL corporation was purchased by Sun Microsystems, which, in turn, has recently been acquired by Oracle.

Acknowledgements

References

[1] Scacchi, W.; Alspaugh, T.; and Asuncion, H.   The Role of Software Licenses in Open Architecture Ecosystems, Intern. Workshop on Software Ecosystems, Intern. Conf. Software Reuse, Falls Church, VA, September 2009.

[2] Lindman, J.; Paajanen, A.; and Rossi, M. Choosing an Open Source Software License in Commercial Context: A Managerial Perspective, Software Engineering and Advanced Applications, Euromicro Conference, pp. 237-244, 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 2010.

[3] St. Laurent, A. M. 2004. Understanding Open Source and Free Software Licensing. O'Reilly Media, Inc., Sebastopol, CA.

[4] Rosen, L. 2005. Open Source Licensing: Software Freedom and Intellectual Property Law. Prentice Hall.

[5] Lerner, J. and Tirole, J. 2005. The Scope of Open Source Licensing The Journal of Law, Economics, & Organization, 21(1): 20-56

[6] Di Penta, M.; German, D.; Guéhéneuc, Y.;. and Antoniol, G. 2010. An exploratory study of the evolution of software licensing. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1 (ICSE '10), Vol. 1. ACM, New York, NY, USA, 145-154.

[7] Goldman, R. and Gabriel, R. 2004. Innovation Happens Elsewhere: How and Why a Company should Participate in Open Source. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[8] Oreizy, P. Open Architecture Software: A Flexible Approach to Decentralized Software Evolution.   Ph.D. in Information and Computer Sciences, Irvine, CA, University of California, Irvine, 2000.

[9] Jensen, C. and Scacchi, W. 2005. Process Modeling Across the Web Information Infrastructure. Software Process: Improvement and Practice 10(3):255-272.

[10] Hedhman, N. Mailing list message dated 16 Dec 2004 07:18:55 -0000 "Re: [ANN] Avalon Closed," available online at http://www.mail-archive.com/community@apache.org/msg03889.html, last accessed 15 September 2009

[11] Dailey, D. Mailing list message dated Wed, 02 May 2007 10:38:26 -0400 "Re: Support Existing Content / consensus through attrition?" available online at http://lists.w3.org/Archives/Public/public-html/2007May/0214.html, last accessed 15 September 2009

[12] The Protégé Ontology Editor Project, available online at http://protege.stanford.edu/ [last accessed 23 June, 2008]

[13] The Firefox Ontology Plugin project available online at http://rotterdam.ics.uci.edu/development/padme/browser/ontology [last accessed 23 June, 2008]

[14] The Zotero Project, available online at http://www.zotero.org/ [last accessed 23 June, 2008]

[15] Bringer, J. D.; Johnston, L. H. and Brackenridge, C. H. Using Computer-Assisted Qualitative Data Analysis Software to Develop a Grounded Theory Project Field Methods, 2006, 18(3): 245-266

[16] Kelle, U. Theory Building in Qualitative Research and Computer Programs for the Management of Textual Data Sociological Research Online, 1997, 2(2) available online at http://www.socresonline.org.uk/socresonline/2/2/1.html [last accessed 23 June 2008]

[17] Fielding, R. Message dated Sat, 08 Nov 2003 02:39:09 GMT "Review of proposed Apache License, version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/%3cBAAB287A-1194-11D8-842D-000393753936@apache.org%3e, last acccessed 14 August 2009

[18] Board meeting minutes of The Apache Software Foundation, January 2004, available online at http://apache.org/foundation/records/minutes/2004/board_minutes_2004_01_21.txt, last accessed 13 August 2009.

[19] Fielding, R. Mailing list message dated Sat, 24 Jan 2004 01:34:36 GMT "Apache License, Version 2.0 ," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/%3C781EEF08-4E0D-11D8-915D-000393753936@apache.org%3E, last accessed 13 August 2009

[20] Behlendorf, B. Mailing list message dated Sat, 22 Nov 2003 07:31:40 GMT "RE: termination with unrelated trigger considered harmful," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/%3C20031121232552.X38821@fez.hyperreal.org%3E, last accessed 13 August, 2009

[21] Carlson, B. M. Mailing list message dated Sat, 8 Nov 2003 10:03:55 +0000 "Re: [fielding@apache.org: Review of proposed Apache License, version 2.0]," available online at http://lists.debian.org/debian-legal/2003/11/msg00053.html, last accessed 12 August 2009

[22] Peterson, S.K. Mailing list message dated Fri, 14 Nov 2003 14:52:54 GMT "termination with unrelated trigger considered harmful," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/%3C6D6463F31027B14FB3B1FB094F2C744704A11176@tayexc17.americas.cpqcorp.net%3E, last accessed 13 August 2009

[23] Machovec, J. Mailing list message dated Fri, 14 Nov 2003 16:49:09 GMT "Re: termination with unrelated trigger considered harmful," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/

%3C3FB50785.7010801@golux.com%3E, last accessed 13 August 2009

[24] Fielding, R. Mailing list message dated Tue, 18 Nov 2003 02:10:27 GMT "Re: [fielding@apache.org: Review of proposed Apache License, version 2.0]," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/ %3c60AEF3C1-196C-11D8-A8F4-000393753936@apache.org%3e, last accessed 13 August 2009

[25] Engelfriet, A. Mailing list message dated Mon, 17 Nov 2003 20:59:53 GMT "Re: [fielding@apache.org: Review of proposed Apache License, version 2.0]," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/ %3c20031117205953.GA95846@stack.nl%3e, last accessed 13 August 2009

[26] Moglen, E. Mailing list message dated Fri, 14 Nov 2003 21:28:32 GMT "FSF Comments on ASL 2.0 draft," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/%3c16309.18688.540989.283163@new.law.columbia.edu%3e, last accessed 13 August 2009

[27] Carlson, B. M. Mailing list message dated Thu, 13 Nov 2003 05:39:49 GMT "DFSG-freeness of Apache Software Licenses," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/ %3c20031113053949.GD23250@stonewall%3e, last accessed 13 August 2009

[28] Armstrong, D. Mailing list message dated Fri, 14 Nov 2003 04:39:50 GMT "Re: DFSG-freeness of Apache Software Licenses," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/ %3C20031114043950.GM2707@donarmstrong.com%3E, last accessed 13 August 2009

[29] Johnson, P. Mailing list message dated Wed, 12 Nov 2003 02:09:14 GMT "Mutual defence patent clause," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/ %3C003d01c3a8c1$f9b55170$c6ba400c@protocol.com%3E, last accessed 12 August 2009

[30] Behlendorf, B. Mailing list message dated Wed, 12 Nov 2003 21:09:32 GMT "Re: Mutual defence patent clause," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200311.mbox/ %3C20031112130508.H497@fez.hyperreal.org%3E, last accessed 13 August 2009

[31] Fielding, R. Mailing list message dated 12/24/2003 04:16 AM "Re: Review of proposed Apache License, version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200312.mbox/%3c464B4006-3604-11D8-9A9F-000393753936@apache.org%3e, last accessed 12 August 2009

[32] Apache License Proposal Website, available online at http://www.apache.org/licenses/proposed/, last accessed 13 August 2009

[33] Apache License, Version 1.23, available online at http://mail-archives.apache.org/mod_mbox/archive-license/200312.mbox, last accessed 13 August 2009

[34] Behlendorf, B. Mailing list message datedFri, 09 Jan 2004 22:42:52 GMT "Re: Review of proposed Apache License, version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/ %3c20040109143803.G31301@fez.hyperreal.org%3e, last accessed 13 August 2009

[35] Behlendorf, B. Mailing list message dated Wed, 07 Jan 2004 22:16:36 GMT "Re: Review of proposed Apache License, version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/ %3c20040107140658.A23429@fez.hyperreal.org%3e, last accessed 13 August 2009

[36] Fielding, R. Mailing list message dated Wed, 14 Jan 2004 20:25:50 GMT "Re: Review of proposed Apache License, version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/%3cD81EA136-46CF-11D8-B08A-000393753936@apache.org%3e, last accessed 13 August 2009

[37] Fielding, R. Mailing list message dated Wed, 14 Jan 2004 20:54:26 GMT "Re: Review of proposed Apache License, version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/%3cD6DB9454-46D3-11D8-B08A-000393753936@apache.org%3e, last accessed 13 August 2009

[38] Armstrong, D. Mailing list message dated Sat, 24 Jan 2004 02:13:50 GMT "Re: Apache License, Version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/%3C20040124021350.GG3060@archimedes.ucr.edu%3E, last accessed 13 August 2009

[39] Fielding, R. Mailing list message dated Sat, 24 Jan 2004 02:29:29 GMT "Re: Apache License, Version 2.0," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/%3C23385101-4E15-11D8-915D-000393753936@apache.org%3E, last accessed 13 August 2009

[40] Free Software Foundation Licenses webpage, available online at http://www.fsf.org/licensing/licenses/index_html#GPLCompatibleLicenses, last accessed 14 August 2009

[41] Massol, V. Mailing list message dated Sun, 25 Jan 2004 16:01:19 GMT "How to use the 2.0 license?," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/%3C012f01c3e35c$78e229d0$2502a8c0@vma%3E, last accessed 13 August 2009

[42] Behlendorf, B. Mailing list message dated Sun, 25 Jan 2004 20:17:06 GMT "Re: How to use the 2.0 license?," available online at http://mail-archives.apache.org/mod_mbox/archive-license/200401.mbox/%3C20040125121456.H396@fez.hyperreal.org%3E, last accessed 13 August 2009

[43] Adams, E. NBDiscuss mailing list message: "Joint Copyright Assignment," available online at http://www.netbeans.org/servlets/ReadMsg?list=nbdiscuss&msgNo=2228, last accessed 6 August, 2009

[44] The Apache Software Foundation Individual Contributor License Agreement, Version 2.0 available online at http://www.apache.org/licenses/icla.txt, last accessed 20 October 2009.

[45] Brabant, V. mailing list message dated Tue, 15 Jul 2003 18:52:36 +0200 "[nbdiscuss] Re: licenses and trees," available onlinet at http://www.netbeans.org/servlets/ReadMsg?listName=nbdiscuss&msgNo=2547, last accessed 20 October 2009.

[46] West, J. and O'Mahony, S. 2005. Contrasting Community Building in Sponsored and Community Founded Open Source Projects. In Proceedings of the Proceedings of the 38th Annual Hawaii international Conference on System Sciences - Volume 07 (January 03 - 06, 2005). HICSS. IEEE Computer Society, Washington, DC, 196.3.

[47] Lerner, J. and J. Tirole. 2000. The simple economics of open source. NBER Working paper series, WP 7600, Harvard University, Cambridge, MA.

[48] von Hippel, E. and von Krogh, G. 2003. Open source software and the 'private-collective' innovation model: Issues for organizational science. Organization Science, 14(2):209-223.

[49] Hedhman, N. mailing list message dated Sun, 29 Jun 2003 13:31:48 +0800 "[nbdiscuss] Re: licenses and trees (was: Anti-Sun Animosity)," available online at http://www.netbeans.org/servlets/ReadMsg?listName=nbdiscuss&msgNo=2578, last accessed 21 October 2009.

[50] NBDiscuss   mailing   list   message.   Available   online   at
       http://www.netbeans.org/servlets/ReadMsg?list=nbdiscuss&msgNo=3784 last accessed 28
       February 2009
[51] Shah, S.K. 2006. Motivation, governance and the viability of hybrid forms in open
       source software development, Management Science, 52(7), 1000-1014.