International Research Workshop on IT Project Management 2012

International Research Workshop on IT Project Management (IRWITPM)

12-15-2012

# Risk Mitigation in Corporate Participation with Open Source Communities: Protection and Compliance in an Open Source Supply Chain

Matt Germonprez

Brett Young

Lars Mathiassen

Julie E. Kendall

Ken E. Kendall

***See next page for additional authors***

**Authors**
Matt Germonprez, Brett Young, Lars Mathiassen, Julie E. Kendall, Ken E. Kendall, and Brian Warner

# Mitigation in Corporate Management of Open Source Community Participation: Protection and Compliance in an Open Source Supply Chain[5]

**Matt Germonprez**
University of Nebraska at Omaha

**Brett Young**
Georgia Gwinnett College

**Brian Warner**
Linux Foundation

**Julie Kendall**
Rutgers University

**Ken Kendall**
Rutgers University

**Lars Mathiassen**
Georgia State University

**Liang Cao**
University of Nebraska at Omaha

## ABSTRACT

Open source communities exist in large part through increasing participation from for-profit corporations. The balance between the seemingly conflicting ideals of open source communities and corporations creates a number of complex challenges for both. In this paper, we focus on corporate risk mitigation and the mandates on corporate participation in open source communities in light of open source license requirements. In response to these challenges, we aim to understand risk mitigation options within the dialectic of corporate participation with open source communities. Rather than emphasizing risk mitigation as ad hoc and emergent process focused on bottom lines and shareholder interests, our interest is in formalized instruments and project management processes that can help corporations mitigate risks associated with participation in open source communities through shared IT projects. Accordingly, we identify two key risk domains that corporations must be attendant to: property protection and compliance. In addition, we discuss risk mitigation sourcing, arguing that tools and processes for mitigating open source project risk do not stem solely from a corporation or solely from an open source community. Instead they originate from the interface between the two and can be paired in a complementary fashion in an overall project management process of risk mitigation.

### Keywords

Open source community, corporate participation, risk mitigation, project management, licenses, compliance.

## INTRODUCTION

Corporate participation with open source communities has become a viable business model in the development of IT artifacts that contain both corporate and community characteristics. Within this, neither a corporation nor an open source community is given preference over the other, as the design and development of a shared artifact is an activity of all involved. Open source design and development has moved beyond the image of the basement hacker to include Fortune 500 corporations leveraging, differentiating, and contributing for reasons of corporate value and community maintenance. Within this ecosystem of open source design and development, the importance of shared artifacts is derived from the distributed, yet communal, efforts from otherwise competitive corporations (Dahlander, 2007; Germonprez et al., 2011).

The Linux kernel is an open source artifact, freely available for public consumption and not owned by any corporation. The Apache web server is an open source artifact, used and modified without charge yet critical to the

[5] This project is funded under the NSF VOSS program as award #1255426 Organizational Participation in Open Communities

success of many business practices. In both cases, the artifact is well recognized, clearly defined, and resides within an open source community. The artifact is designed and continues to evolve through both community and corporate ideals. Without shared participation, open source artifacts would not exist in their current forms. Linux and Apache illustrate high profile examples of open source and are not indicative of all open source projects. They are also not necessarily representative of the majority of open source projects that corporations are engaged in.

Open source software has a broad reach, tied to software packages used both internally and externally to a corporation. Unless sourced from a proprietary vendor, software packages are often comprised of source files licensed under a variety of different licensing conditions. A scan of Enyo 2.0,[6] a JavaScript framework, reveals 1,400 files regulated by ten different open source licenses. In the case of Enyo, the implications of using the software package as part of a larger corporate offering has implications beyond simple attribution to the originating design of the Enyo package (Cheliotis, 2009). As is the case with Enyo and other open source packages, consuming and distributing multi-licensed open source software packages can affect corporate risk project management in open source community participation. For example, a software package that contains files licensed under the General Public License V2[7] requires provision of all source code that is connected to the licensed files for three years, when an open source package is used within a corporate product for sale. So, as Cisco modifies the Linux kernel in their Internet routers, they must post the Cisco-developed source code that interacts with the Linux kernel. In some situations, these requirements are well understood by a corporation and the release of necessary source code to an open source community is considered a cost of doing business. However, the Open Source Initiative recognizes over 60 open source licenses,[8] all with varying degrees of requirements, responsibilities, and risks incurred by a user of open source software. It is therefore not unlikely that a corporation is unaware of the particular licensing requirements within a software package.

In this paper we explore the implications of open source software packages on corporate risk. We demonstrate how risks are inherent, yet can be mitigated in corporate participation with an open source community and we argue that risk mitigation originates from both within participating corporations and within open source communities. In particular, we address the following research question:

1) *How is risk mitigation manifest with open source communities?*

Corporate participation with open source communities is a practiced business approach in the design, development, and deployment of software packages. It is not a 'one size fits all' consideration, used by all corporations in all circumstances. However, it is a consideration that has gained increasing traction in corporations for leveraging an open source community to extend the design and development capacity within a corporation. Open source software has become a suitable option for corporations looking to expand their design and development options in fast-paced and highly competitive markets. It is against this backdrop we respond to the research question with the goal to develop a risk mitigation approach that can help alleviate the potential pitfalls associated with corporate participation with open source communities. As a result, we provide perspectives on the perplexing situation of how risk can be mitigated as a necessary part of using, developing, and managing open source projects as part of commercial product releases.

## RISK IN CORPORATE PARTICIPATION WITH OPEN SOURCE COMMUNITIES

Risks exist in every software development project. Managing and mitigating risk is one key to successful IT project management (Du et al, 2007; Keil et al, 2008). The software development literature largely focuses on internal risks – e.g. the risks that occur inside a corporation. These risks include project scheduling issues, project personnel issues, project culture, control challenges, technical issues, software adoption issues, vendor selection and contracting difficulties, and relationship management problems.

[6] http://www.enyojs.com

[7] http://www.gnu.org/licenses/gpl-2.0.html

[8] http://opensource.org/licenses/alphabetical

Risks involving open source software licensing are somewhat different (Al Marzouq et al., 2005). While it is possible to have a third party assume some responsibility to compensate an injured party after an event occurs, a corporation that sells an artifact that includes improper handling of licensed code remains legally responsible for the event. For example, if a corporation sells a software package using open source code and fails to abide by the licensing agreement for that code, that corporation must defend itself. Here lies open source software risk that must be mitigated. While open source software adoption risks have been a topic for corporations when weighing the benefits and consequences of open source software implementation (e.g. Daniels et al., 2011), risk and risk mitigation typically have not been a focus in the literature on open source exchange and management. In open source software projects, risk is inherent in a number of external places. In the design and development of open source software, external risk is noticeably manifest in software supply chains, necessitating risk mitigation strategies (Gefen and Carmel, 2008). This is evident as software design and development often entails the integration of open source code, which was developed beyond a corporation's boundaries into an internal and corporate-maintained code stream.

Perhaps the most recognized conceptualization of corporate participation with open source software is an *adoption approach* of common, free, and open source software in day-to-day corporate activities (Castelluccio, 2008). This would include the cases for preferring Linux over Windows, Apache in favor of IIS, and MySQL in favor of Oracle. While there may be economic advantages for these cases within a corporate setting, we consider this style of participation to simply be one of adoption, not necessarily an engaged participation. In these cases, an open source community does not need to be engaged by a corporation; instead they contract vendors to provide installation, maintenance, and support much the same as proprietary software vendors. Risk is also evident in the adoption approach of common, free, and open source software as an open community responsible for the design, development, and maintenance of a particular open source project may disband, leaving little in the way of future support or product development. Risk may also manifest through a lack of tooling associated with an open source project in relation to proprietary systems (Yalta and Lucchetti, 2008), burdening an adopting corporation with the responsibility of designing and developing toolsets internally. Many of the risks incurred in the 'adoption' style of participation stem from the infancy or instability of an open source community in providing real value to a corporation often in the light of proprietary options (Ringle, 2004).

As a more active form of engagement, corporations can take a *shared approach* towards open source communities, deciding deliberately and strategically to participate with open source communities. A corporation may choose to participate for reasons of 'upstreaming' or 'franchising' corporate philosophy into an open source community. As an example, a corporation may participate with an open source community in an effort to embed a corporate (and competitive) form of virtualization into an open source operating system. If a corporation is successful in contributing their corporate view of virtualization to an open source community, distributions of the operating system will include and hence circulate this vision. In some cases, corporations may be forced to engage a shared approach in an open source community through current business partnerships that a corporation is engaged in. A corporation can have their 'hand forced' into participation with an open source community as they seek to develop or maintain a competitive position within a selective market. For example, participating with an open source community may be in an effort to broaden silicon chip distribution to include the smartphone and tablet markets. As smartphones and tablets can use a Linux-based operating system, licensed under Apache 2.0 and the General Public License (V2), corporations are required to participate in the respective open source communities as defined in the community licenses. Finally, a corporation may choose to participate in an effort to leverage the collaborative efforts of an open source community (Fitzgerald, 2006).

In a shared approach, participation is generally reciprocal between a corporation and an open source community (Feller et al., 2008). There exists an intention to leverage an open source community for reasons of corporate gain while at the same time contributing back to an open source community in efforts to provide its long term advancement and sustainability. Risks in a shared approach are often internal to a corporation since such time and effort may be lost if a corporation fails to effectively upstream contributions to an open source community. In this scenario, a participating corporation must maintain their own potential open source contributions internally, effectively negating the value in participation (Kogut and Metiu, 2001). Corporations additionally incur risk by potentially (and somewhat inadvertently) releasing intellectual property to an open source community due to open source licensing requirements (McGhee, 2007).

Finally, corporate participation with open source communities may be a *supply chain approach*, stemming from an exchange of open source software packages between corporate partners. In the case of open source supply, participation is not between corporation and an open source community but is instead between two corporations. In this, corporate participation is best understood with an open source software supply chain where software should be assumed to not be the domain of a single corporation, originating from and existing solely within one corporation. Instead, software should be assumed to originate from a variety of sources and exchanged between corporations as part of supplier-buyer chain of relationships. A software supply chain is not exempt from including open source packages and files, regulated under a variety of licenses. In such a software supply chain, attention to open source software and the requisite licenses is paramount in maintaining compliance with the originating open source communities. While this style of participation is between corporations in a supply chain, participation with an open source community remains implicit as a buying corporation is subject to the necessary terms and obligations associated with that software, and unfamiliarity with the community and its regulations is not an exemption.

In a case of a supply chain approach towards an open source community, it is recognized that open source software carries necessary licensing obligations, and corporations inherit the risk of the associated license responsibilities (Al Marzouq et al., 2005). Risk is inherent in a corporation-to-corporation exchange of software to which portions may be open source (Cheliotis, 2009). Risk is incurred via supply chain relationships as the establishment and refinement of supply chain partnerships is critical and valued component in the health of any corporation. Disturbance to these relationships can have complicating effects on an overall software supply chain.

Across each of the three styles, there are inherent risks associated with open source participation. Risks can include the potential release of intellectual property to an open community or partner corporation, resulting from an unsuitable interpretation of an open source license. Risk can also include the failure to provide requisite attribution to an originator of an open source package, resulting in a reduced standing within an open community or amongst partner corporations. Certain risks are more problematic than others and some risks are more evident across certain styles of corporation participation, across different software packages, and within variable open source communities. A review of Aksulu and Wade (2010) reveals little empirical work regarding risk, risk mitigation, and corporate participation in a supply chain, project management approach to open source communities. Of the research that addresses, or mentions risk, there is an unbalanced concentration toward risk and risk mitigation associated with an open source software adoption approach (Aksulu and Wade, 2010). Table 1 illustrates a summary of risk in corporate participation with open source communities, addressing our first research question.

| Participation style | Open source software risk is relative … | Examples of risk | Representative papers |
|---|---|---|---|
| Adoption Approach | to use of publicly available open source software within a corporation | • Collapse of a supporting open community<br>• The lack of community tooling in support of open source software. | Castelluccio, 2008<br>Ringle, 2004<br>Yalta and Lucchetti, 2008 |
| Shared Approach | to the use of software and licenses as defined and understood between corporations and open communities | • Corporate inability to effectively upstream with an open source community<br>• Ineffective license compliance when distributing corporate products imbued with leveraged open source software. | Kogut and Metiv, 2004<br>McGhee, 2007 |
| Supply Chain Approach | to the exchange of open source software packages between partner corporations. | • Ineffective expression of open source software obligations embedded in supply chain code<br>• A failure to integrate open source software obligations | Al Marzouq et al., 2005<br>Cheliotis, 2009 |

**Table 1. Participation Style with Open Source Software and Respective Risks**

In response to the differentiated risks across the various forms of participation, we empirically focus on what is evident to be the most poorly established and least understood area of risk in open source participation: open source participation in a software supply chain approach. In the following sections of the paper, we specifically consider risk in the context of a supply chain approach of corporate participation with open source communities. We understand risk mitigation approaches that exist within both corporations and open source communities, and based on that understanding we suggest that risk mitigation is a shared process between corporate participants in a project management effort to both improve and be improved by open source communities.

## RESEARCH APPROACH

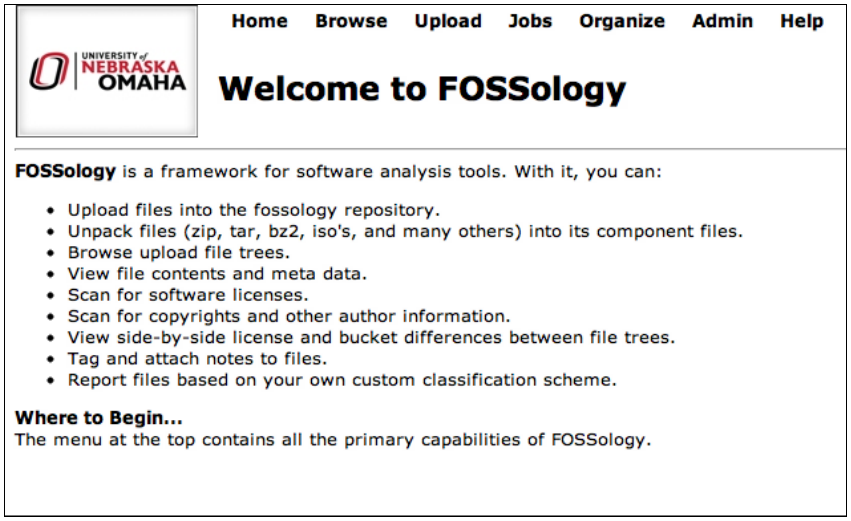
We applied a field study approach in the investigation of risk mitigation in a supply chain approach of corporate management in open source participation. Members of the research team have been investigating corporate participation with open source communities for the past two years, asking questions of why and how participation is manifest. Our investigation of risk mitigation is a progression from our prior work and is in accord with our strong corporate ties and open community understanding.[9]

Our research specifically applied the field study approach in an engagement with corporate participants, the FOSSology community, and the SPDX community as forums for understanding risk mitigation. Over the course of

[9] The field study is part of a larger NSF-funded initiative on open source participation: http://nsf.gov/awardsearch/showAward.do?AwardNumber=1255426&WT.z_pims_id=503256

the project, we have been involved in twelve working group meetings across both communities, conducted and transcribed over 70 semi-structured interviews with corporate participants in open source communities, designed and developed advancements in both the FOSSology and SPDX communities, and we continue to maintain activity in integrating the two risk mitigation tools. Our engagement has been further documented through researcher notes, meeting minutes, and community presentations. Our approach has placed our team within the respective open source communities in an effort to best understand risk mitigation in a supply chain approach through active participation with both corporations and open source communities.

As mentioned, the research team has integrated with open source communities dedicated to risk mitigation, become active participants in their design and development efforts. In particular, the research team identified two open source communities that have mutual trajectories so that team participation and engagement in one can benefit the efforts of the other. The two risk mitigation projects include FOSSology,[10] an open source data analysis tool that scans software packages for open source software and licenses. Figure 1 shows the interface for FOSSology where users can upload and scan software packages, and browse the results.


Home Browse Upload Jobs Organize Admin Help
UNIVERSITY of NEBRASKA OMAHA

Welcome to FOSSology

**FOSSology** is a framework for software analysis tools. With it, you can:

- Upload files into the fossology repository.
- Unpack files (zip, tar, bz2, iso's, and many others) into its component files.
- Browse upload file trees.
- View file contents and meta data.
- Scan for software licenses.
- Scan for copyrights and other author information.
- View side-by-side license and bucket differences between file trees.
- Tag and attach notes to files.
- Report files based on your own custom classification scheme.

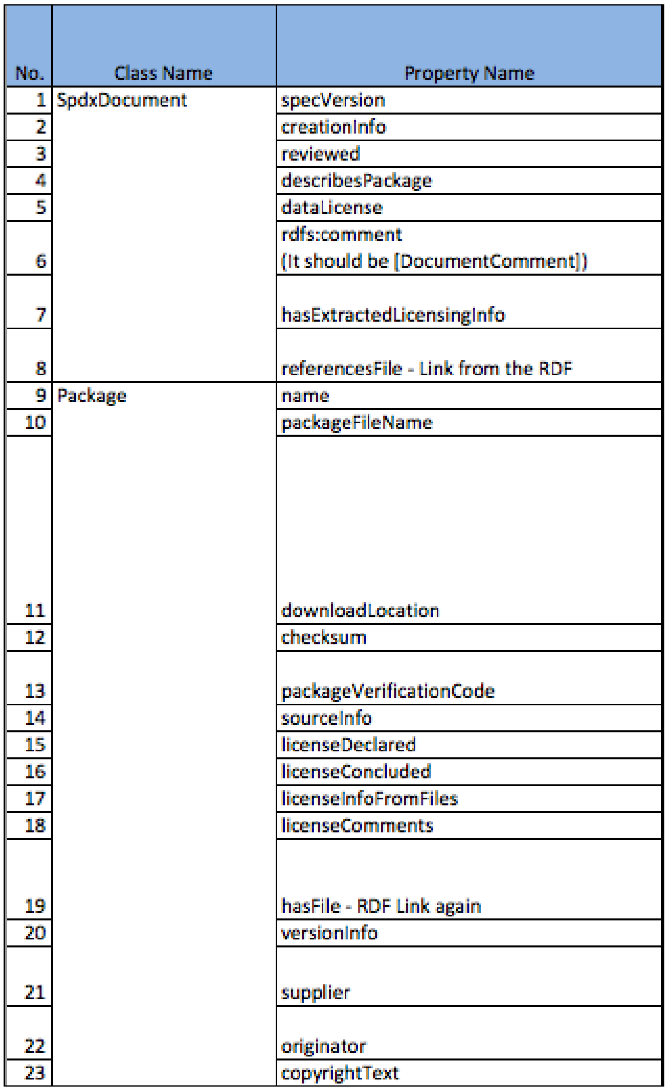
**Where to Begin...**
The menu at the top contains all the primary capabilities of FOSSology.

**Figure 1. Screen Capture of the FOSSology Software Scanning Tool**

The second risk mitigation project is a specification for software package data exchange (SPDX). The tool specifies approximately 70 criteria in the exchange of software between corporations. An open source community associated with the development of the SPDX specification aims to "develop and promote adoption of a specification to enable any party in a software supply chain, from the original author to the final end user, to accurately communicate the licensing information for any piece of copyrightable material that such party may create, alter, combine, pass on, or receive, and to make such information available in a consistent, understandable, and re-usable fashion, with the aim of facilitating license and other policy compliance."[11] Figure 2 shows a sample of the criteria evident in an SPDX document.

[10] http://www.fossology.org

[11] http://spdx.org/content/vison-strategy-execution



| No. | Class Name | Property Name |
|---|---|---|
| 1 | SpdxDocument | specVersion |
| 2 | | creationInfo |
| 3 | | reviewed |
| 4 | | describesPackage |
| 5 | | dataLicense |
| 6 | | rdfs:comment<br>(It should be [DocumentComment]) |
| 7 | | hasExtractedLicensingInfo |
| 8 | | referencesFile - Link from the RDF |
| 9 | Package | name |
| 10 | | packageFileName |
| 11 | | downloadLocation |
| 12 | | checksum |
| 13 | | packageVerificationCode |
| 14 | | sourceInfo |
| 15 | | licenseDeclared |
| 16 | | licenseConcluded |
| 17 | | licenseInfoFromFiles |
| 18 | | licenseComments |
| 19 | | hasFile - RDF Link again |
| 20 | | versionInfo |
| 21 | | supplier |
| 22 | | originator |
| 23 | | copyrightText |

**Figure 2. Sample SPDX Specification Criteria**

With methodological consideration toward risk mitigation within open communities, the efforts of the research team have been both active and engaged. The first author has been an active member with two open source communities engaged in risk mitigation efforts for the last six months and has presented at LinuxCon 2012 on these efforts. The research team hosts a public instance of an open source project on risk mitigation.[12] Finally, the research team includes a member from the largest non-profit consortium dedicated to fostering the growth of Linux, including one of our investigated risk mitigation projects.

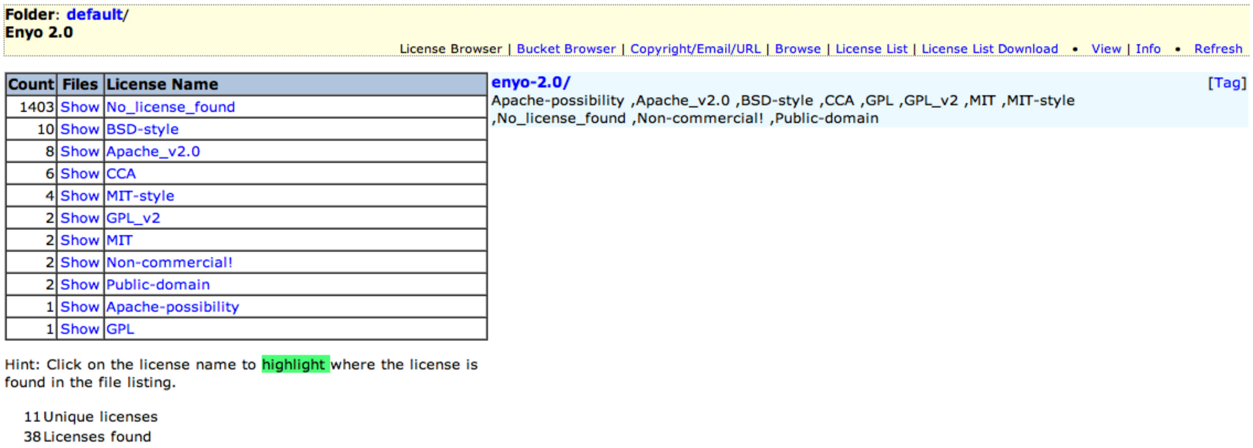
## FINDINGS

Our findings focus on the second of the two presented research questions; specifically how risk mitigation is manifest within open source communities. We found that risk mitigation resides both internally to corporations and also is distributed throughout an open community. Corporate and community approaches can work in combination and provide a broad approach toward risk mitigation. We will present three approaches toward risk mitigation stemming from our fieldwork with corporate participants in open source communities as well as our direct engagement with these communities.

[12] http://fossology.ist.unomaha.edu

**A Case for Software Scanning**. Software scanning is becoming an increasingly relevant activity with respect to risk mitigation in corporate participation with open source communities. There are a number of open source and for-profit offerings associated with software scanning, with several corporations offering these services at an enterprise level. From a more local level within open source communities, tools have emerged to support software-scanning processes to "build a community to facilitate the study of free and open source software by providing free data analysis tools."[13]

Software scanning provides a level of understanding, not decision making, which can be used to support either individual or corporate open source efforts. Software scanning is most commonly used to support processes of license vetting. As part of a larger process of license and property protection decision making within corporations, software scanning tools have emerged as a key part of knowing the inherent risks in an open source package. In the case of the FOSSology, the open source community to which the research team participates, software scanning provides a window into the types and locations of licenses embedded in open source code. Figure 3 provides an image of the results from the earlier mentioned Enyo 2.0 Framework:



**Folder**: **default**/
**Enyo 2.0**

License Browser | Bucket Browser | Copyright/Email/URL | Browse | License List | License List Download • View | Info • Refresh

| Count | Files | License Name |
|---|---|---|
| 1403 | Show | No_license_found |
| 10 | Show | BSD-style |
| 8 | Show | Apache_v2.0 |
| 6 | Show | CCA |
| 4 | Show | MIT-style |
| 2 | Show | GPL_v2 |
| 2 | Show | MIT |
| 2 | Show | Non-commercial! |
| 2 | Show | Public-domain |
| 1 | Show | Apache-possibility |
| 1 | Show | GPL |

**enyo-2.0/** [Tag]
Apache-possibility ,Apache_v2.0 ,BSD-style ,CCA ,GPL ,GPL_v2 ,MIT ,MIT-style ,No_license_found ,Non-commercial! ,Public-domain

Hint: Click on the license name to highlight where the license is found in the file listing.

11 Unique licenses
38 Licenses found

**Figure 3: License Scanning Output for the Enyo 2.0 Framework**

In this example, the Enyo 2.0 repository contains 38 licenses within the software package (10 unique). The licenses range from permissive (MIT) to restrictive (GPL), each carrying different obligations for a corporation using this software package. Understanding the obligations of each license is critical to managing and mitigating risk to a corporation wanting to utilize this package in its own commercially released products. This subsequently leads to the related consideration of what can be done with the scanned software license information.

**A Case for Open Source Program Offices.** With respect to a supply chain approach, corporations have developed internal mechanisms, both to manage the protection of intellectual property and to maintain necessary compliance with open source package licenses. Figure 4 illustrates a management review that an open source program office might conduct when evaluating internal open source projects, in part understood through software scanning results. In this example, a project team first creates a project incorporating open source code. This project is given management approval and is reviewed by a business attorney who offers guidance regarding legal risks. Once legal approval is given, an open source review board reviews the project using internal code evaluation tools to identify areas needing further review and then follows up with the originating project team. If pre-approved licenses are involved, the licensing team reviews the project and offers legal guidance regarding present risks. If no pre-approved licenses are involved, a new license review is performed. Finally, after completing all reviews, the project is approved or denied.

[13] http://www.fossology.org

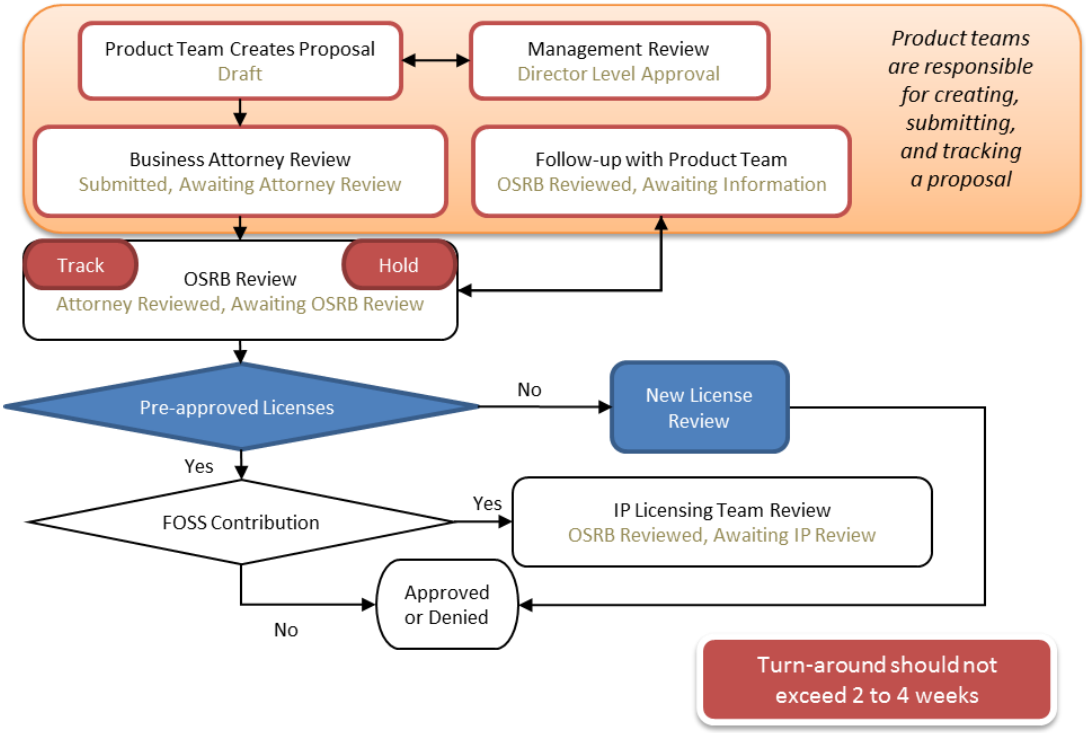

Product Team Creates Proposal
Draft
Management Review
Director Level Approval
Product teams are responsible for creating, submitting, and tracking a proposal
Business Attorney Review
Submitted, Awaiting Attorney Review
Follow-up with Product Team
OSRB Reviewed, Awaiting Information
Track
Hold
OSRB Review
Attorney Reviewed, Awaiting OSRB Review
Pre-approved Licenses
No
New License Review
Yes
FOSS Contribution
Yes
IP Licensing Team Review
OSRB Reviewed, Awaiting IP Review
Approved or Denied
No
Turn-around should not exceed 2 to 4 weeks

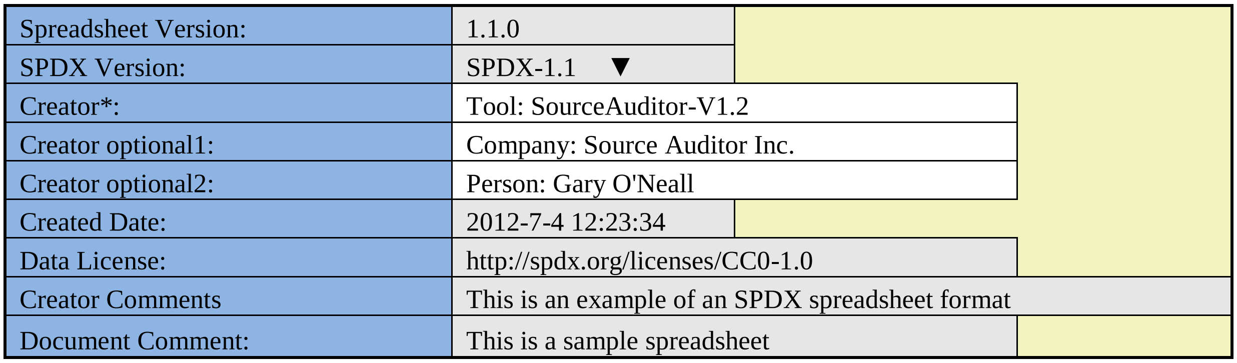
**Figure 4: Management Review Processes of an Open Source Program Office**

Open source review boards are generally comprised of corporate employees familiar with the complexities and nuances associated with open source community participation. Through an open source review board, risk is mitigated through corporate processes as in Figure 4, wherein supply chain participation is undertaken only after license and property concerns are satisfied.

**The Case of Open Source Data Exchange (Community)**. Knowing license requirements with a software package are (see FOSSology) and knowing how to apply those findings (see open source program offices) is key in any risk mitigation agenda. Exchanging data between corporations, per an open source supply chain, represents a final consideration in understanding of risk mitigation in a supply chain approach to corporate participation with open source communities.

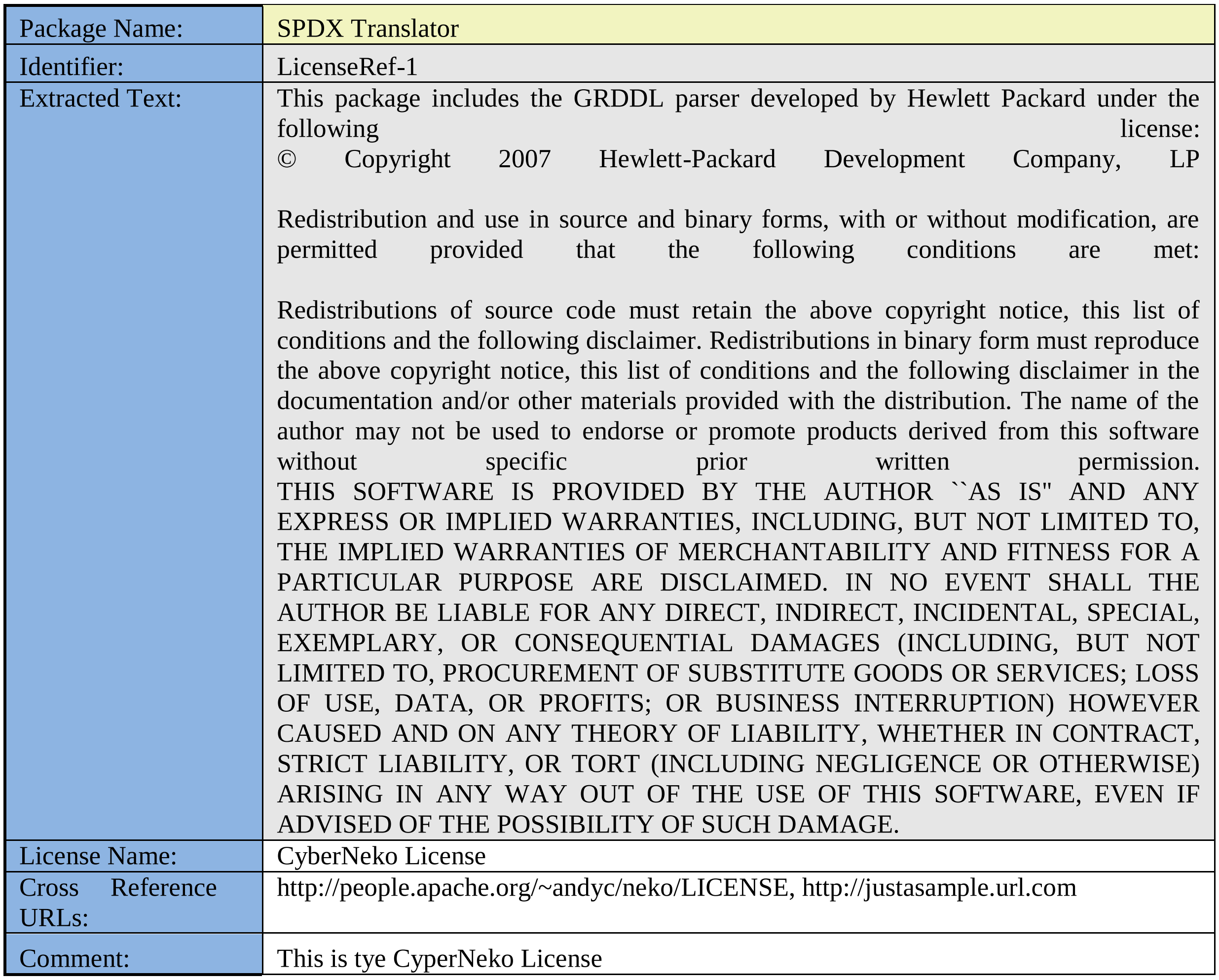
Emergent as risk mitigation tool from within open communities, the software package data exchange (SPDX) specification is "a standard format for communicating the components, licenses and copyrights associated with a software package. The SPDX standard helps facilitate compliance with free and open source software licenses by standardizing the way license information is shared across the software supply chain."[14] The goal of SPDX is to mitigate risk inherent between corporations in a supply chain of open source code. SPDX is analogous to a bill of materials describing which parts and components are included during the product's manufacture. Similarly, SPDX articulates the material within a software package, including the evident licenses, insurance of SPDX validity, and SPDX author credentials. Figure 5 and Figure 6 illustrate SPDX output exchanged between corporations in a software supply chain.

[14] http://www.spdx.org



| | |
|---|---|
| Spreadsheet Version: | 1.1.0 |
| SPDX Version: | SPDX-1.1 ▼ |
| Creator*: | Tool: SourceAuditor-V1.2 |
| Creator optional1: | Company: Source Auditor Inc. |
| Creator optional2: | Person: Gary O'Neall |
| Created Date: | 2012-7-4 12:23:34 |
| Data License: | http://spdx.org/licenses/CC0-1.0 |
| Creator Comments | This is an example of an SPDX spreadsheet format |
| Document Comment: | This is a sample spreadsheet |

**Figure 5: SPDX Authorship Information**



| | |
|---|---|
| Package Name: | SPDX Translator |
| Identifier: | LicenseRef-1 |
| Extracted Text: | This package includes the GRDDL parser developed by Hewlett Packard under the following license:<br>© Copyright 2007 Hewlett-Packard Development Company, LP<br><br>Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:<br><br>Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.<br>THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. |
| License Name: | CyberNeko License |
| Cross Reference URLs: | http://people.apache.org/~andyc/neko/LICENSE, http://justasample.url.com |
| Comment: | This is tye CyperNeko License |

**Figure 6: SPDX Extracted License Information**

In each figure, information is provided to articulate the delivery and receipt of software in a software supply chain. These figures only illustrate 15 of 70 fields in an SPDX document but demonstrate the nature and style of information exchanged in an effort to mitigate risk between corporations participating with open source communities.

We found that risk mitigation in an open source supply chain entails both community developed tools (software scanning and data exchange) and corporate processes (open source review board). These three approaches can be considered in relation to each other as a connected process for risk mitigation as seen in Figure 7:

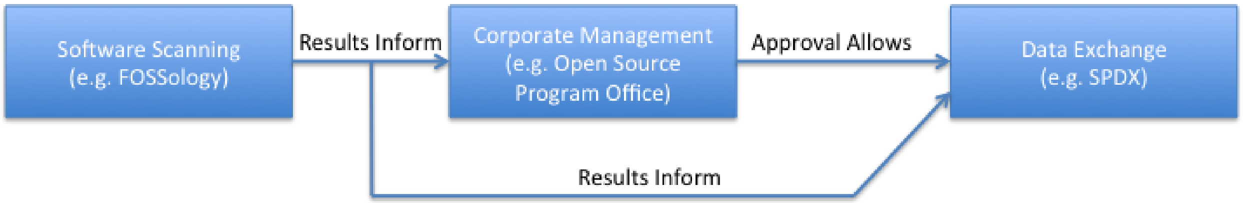


**Figure 7: Risk Mitigation Process in Open Source Supply Chains**

In Figure 7, a software package scan would produce results that can be used per an open source program office review. Upon approval, the original software scanning results can be used to generate the data exchange documents. Across all, risk is driven out of the process by exposing, understanding, and expressing the obligations inherent in an open source software package. Continued work remains as the research team continues to be involved in the process model expressed in Figure 7, specifically focusing on how the results from software scanning can be expressed such that they can easily merge with data exchange.

## CONCLUSIONS

As risk mitigation in an open source supply chain has origins in both a corporate and communal setting, it can be best understood as a shared process for open source project management. The efforts benefit both parties as corporations seek to leverage open communities in efforts to increase design and development capacities and communities seek to leverage corporations for improved market share and distribution. As both sides are inherently dependent on the other, developing approaches toward risk mitigation represents practical, business-driven solutions to develop long-term, productive relationships for all.

Drawing on the literature on risks and risk mitigation within the software discipline it is our intension to further develop this line of research. We have outlined the contours of some key risks and the nature of risk mitigation tactics in this paper. In future iterations, we will include a richer understanding of the risk domains that corporations face when engaging with open source communities in joint IT projects, of the different sources of mitigation that are available, and the heuristics by which managers apply this understanding to manage corporate projects in open source communities. In developing such insights into tools and processes for management, we also need to consider the appropriate form under which risks may be related to mitigation tactics (cf. Iversen et al. 2004).

## REFERENCES

Aksulu, A. and Wade, M. (2010). A Comprehensive Review and Synthesis of Open Source Research, *Journal of the Association for Information Systems*, 11, pp. 576-656.

Al Marzouq, M. L. Zheng, G. Rong and V. Grover (2005) “Open Source: Concepts, Benefits, and Challenges”, *Communications of the Association for Information Systems*, 16, Article 37.

Castelluccio, M. (2008) “Enterprise Open Source Adoption”, *Strategic Finance*, 90(5), pp. 57-58.

Cheliotis, G. (2009) “From open source to open content: Organization, licensing and decision processes in open cultural production”, *Decision Support Systems*, 47(3), pp. 229-244.

Dahlander, L. (2007) “Penguin in a new suit: a tale of how de novo entrants emerged to harness free and open source software communities”, *Industrial and Corporate Change*, 16(5), pp. 913-943.

Daniels, S., Maruping, L., Cataldo, M., Herbsleb, J. (2011). When cultures clash: Participation in open source communities and its implication for organizational commitment, Available at: http://reports-archive.adm.cs.cmu.edu/anon/isr2011/CMU-ISR-11-104.pdf.

Du, S., Keil, M., Mathiassen, L., Shen, Y., and Tiwana, A. (2007). “Attention-Shaping Tools, Expertise, and Perceived Control in IT Project Risk Assessment,” *Decision Support Systems*, 43(1), pp. 267-283.

Feller, J., P. Finnegan, B. Fitzgerald and J. Hayes (2008) “From Peer Production to Productization: A Study of Socially Enabled Business Exchanges in Open Source Service Networks”, *Information Systems Research*, 19(4), pp. 475-493.

Fitzgerald, B. (2006) “The transformation of open source software”, *MIS Quarterly*, 30(3), pp. 587-598.

Gefen, D. and Carmel, E. (2008). Is the world really flat? A look at offshoring at an online programming marketplace, *MIS Quarterly*, 32(2), pp. 367-384.

Iversen, L. Mathiassen and P. A. Nielsen. (2004). “Managing Risks in Software Process Improvement: An Action Research Approach,” *MIS Quarterly*, 28(3), pp. 395-433.

Keil, M., Li, L., Mathiassen, L., and Zheng, G. (2008). “The Influence of Checklists and Roles on Software Practitioner Risk Perception and Decision-Making,” *The Journal of Systems and Software*, 81(6), pp. 908-919.

Kogut, B. and A. Metiu (2001) “Open-source software development and distributed innovation”, *Oxford Review of Economic Policy*, 17(2), pp. 248-264.

McGhee, D.D. (2007) “Free and open source software licenses: Benefits, risks, and steps toward ensuring compliance”, *Intellectual Property & Technology Law Journal*, 19(11), pp. 5-9

Ringle, M. (2004) “Can Collaboration Rescue Imperiled It Budgets”, *EDUCAUSE Review*, 39(6), pp. 38-46.

Yalta, A.T. and R. Lucchetti (2008) “The GNU/Linux platform and freedom respecting software for economists”, *Journal of Applied Econometrics*, 23(2), pp. 279-286.