2018

Open Data Standards for Open Source Software Risk Management Routines: An Examination of SPDX

Robin A. Gandhi
University of Nebraska at Omaha, rgandhi@unomaha.edu

Matt Germonprez
University of Nebraska at Omaha, mgermonprez@unomaha.edu

Georg J.P. Link
University of Nebraska at Omaha, glink@unomaha.edu

Open Data Standards for Open Source Software Risk Management Routines: An Examination of SPDX

Robin Gandhi
University of Nebraska at Omaha
Omaha, NE, USA
rgandhi@unomaha.edu

Matt Germonprez
University of Nebraska at Omaha
Omaha, NE, USA
mgermonprez@unomaha.edu

Georg J.P. Link
University of Nebraska at Omaha
Omaha, NE, USA
glink@unomaha.edu

ABSTRACT
As the organizational use of open source software (OSS) increases, it requires the adjustment of organizational routines to manage new OSS risk. These routines may be influenced by community-developed open data standards to explicate, analyze, and report OSS risks. Open data standards are co-created in open communities for unifying the exchange of information. The SPDX® specification is such an open data standard to explicate and share OSS risk information. The development and subsequent adoption of SPDX raises the questions of how organizations make sense of SPDX when improving their own risk management routines, and of how a community benefits from the experiential knowledge that is contributed back by organizational adopters. To explore these questions, we conducted a single case, multi-component field study, connecting with members of organizations that employed SPDX. The results of this study contribute to understanding the development and adoption of open data standards within open source environments.

Author Keywords
Risk Management; Open Source Software; Standardization; Practice Theory; Routines; Case Study; Interviews

ACM Classification Keywords
D.2.3 [Software Engineering]: Coding Tools and Techniques—Standards

INTRODUCTION
Organizations are using open source software (OSS) at increasing rates. This includes use in internal development processes, upstream contributions to open source communities, and redistribution in delivered products and services. While the benefits for engaging with open source communities have been well documented [5,8,11,12], engagement with OSS exposes an organization to a number of legal, intellectual property, and security risks. To manage these complex risk factors, organizations have developed routines that include tracking open source assets throughout an organization, creating cross-functional teams to vet OSS licenses, and partnering with open source foundations to support risk management routines. To assist with the complexities of OSS risk management during software exchange in a supply chain, the Software Package Data Exchange (SPDX®) specification was established by the Linux Foundation's SPDX workgroup. SPDX is a community of organizational members who have co-created and applied the SPDX specification from which OSS risk related routines can be enacted. We refer to these practicing and contributing organizational members in the SPDX workgroup as the "SPDX community."

The SPDX specification is quite simply a specification in the way that HTML or IEEE 802.11g are specifications. SPDX intends to support the supply chains that rely on OSS for seamless exchange of software. It is defined by the community, yet the specification does not detail the distributions and engagements of users that work with it locally. As such, engagement with any specification, including SPDX, takes different forms, depending on local organizational situations. An organization using SPDX prepares "SPDX documents" by examining OSS packages. An SPDX document captures metadata information about a software package and is structured according to the SPDX specification. SPDX documents include fields for the name of the software package, version number, license of the software package, URLs to locate vulnerability announcements, and the relationships of the package to other packages (i.e., is a copy_of or prerequisite_for). Figure 1 illustrates the relationship between the SPDX specification, an OSS package, and the resulting SPDX document. The routines of interest in this paper enact this relationship.

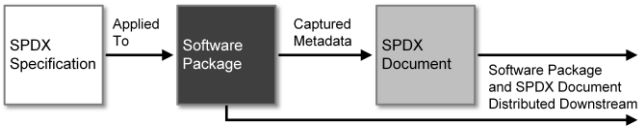

Figure 1. The SPDX specification is applied to a software package to capture its metadata in a standard form in the SPDX document (an instance of the data standard). The SPDX document and the software package are distributed together to downstream users.

GROUP '18, January 7–10, 2018, Sanibel Island, FL, USA

ACM ISBN 978-1-4503-5562-9/18/01.
https://doi.org/10.1145/3148330.3148333

In this paper, we explore interactions in the SPDX community through routines. Specifically:

Locally Structured Routines: In response to the growth of the SPDX community, this research explores how the SPDX specification - one particular artifact produced by the SPDX community - is used to guide improvements to OSS risk management routines in participating organizations. We consider how the SPDX specification serves as both a source of inertia and inflexibility and at the same time offers opportunities for flexibility and change to organizational members considering their own, local OSS risk management routines [7].

Communally Structured Routines: Organizations contribute to the SPDX specification by discussing their own routines and negotiating how these routines will be supported in the SPDX specification. For example, the first version of the SPDX specification was untested and based on assumptions about what OSS risk management routines might look like and how those routines should be captured in a shared specification. After each release, the implementation experience and feedback from organizational members helped improve and evolve the SPDX specification to suit real world OSS risk scenarios.

Routines, such as OSS risk management routines, are dualities [7]. They are, in part, their fixed, organized, and structured aspects. This could include the list of steps to accomplish a particular task, the driving directions between two points, or the instructions for baking a cake. Routines are also, in part, their patterns of behavior when interpreting and enacting the structured instructions. These negotiated aspects are reflected in the task workarounds, the driving shortcuts, and the deflated cake. Both parts inform each other. In this research, we present a single case, multi-component study to understand how OSS risk management routines are advanced through the combination of local interpretation and communal routines, leading to our research questions:

RQ1: How do organizations participating in the SPDX community describe their local interpretations of communally structured OSS risk management routines?

RQ2: How do these local interpretations influence the extent of their SPDX adoption?

RQ3: How do these member organizations seek to guide the advancement of the shared SPDX specification?

THE SPDX COMMUNITY

Since 2010, SPDX has become a community of diverse organizational members – software, systems and tool vendors, foundations, and systems integrators – who collaborate in developing the SPDX specification. The history of the SPDX community dates back to 2007, when the original founders raised the issue of software pedigree and authenticity associated with the exchange of OSS. The SPDX community is currently supported by the Linux Foundation, as one of its core workgroups aimed at advancing the use and distribution of OSS. Similar to other projects at the Linux Foundation, SPDX development work is shared among the organizations volunteering their expertise and who have the interest and capacity in using the specification in their own risk related OSS routines.

To manage different activities in the SPDX community, teams are organized to share responsibilities. The Technical Team develops the SPDX specification, documentation, templates, samples, and tools. The Legal Team manages the SPDX License List, a subset of the full SPDX specification that provides a standardized short identifier for OSS licenses. The Outreach Team coordinates public appearances and promotion of SPDX, including participation in events and maintaining the website. The activities of all teams are coordinated at the monthly SPDX General Meeting via a conference call. Within this structure, organizations participate in the SPDX community and contribute their individual experience and expertise where they best can.

EXCHANGING ORGANIZATIONAL ROUTINES

Routines are sets of actions executed repeatedly with reliable outcomes and routines have both fixed and negotiated aspects [7,22]. Fixed aspects are embodied in artifacts, workflows described in references, standardized forms, or other tools used for executing the routines. The fixed aspects of routines can be explicitly stored, shape expectations for behavior, and allow multiple people to carry out actions in coordinated, repetitive, and recognizable patterns [22]. However, routines are constantly adapted and negotiated to circumstances – slightly differently each time [22].

Organizations can exchange routines that were developed elsewhere and thus not have to invent their own routines [23]. In such exchanges, routines are often transferred in a codified form such as handbooks, software, and proprietary standards. The encoding is influenced by the originating organization and its specific context, culture, and understanding. Organizations must overcome the knowledge boundary resulting from differences in organizational contexts and backgrounds before integrating external routines [19,23]. Challenges also exist for implementing off-the-shelf routines (e.g., embedded in commodity software) from vendors where the organization needs to unpack the codified knowledge and integrate it with existing organizational knowledge [21]. Knowledge embedded in artifacts will likely be misunderstood [22,23] and employees will have difficulty applying the exchanged routine [19].

Creating Shared Routines through Shared Standards

An alternative to adopting external routines is to create shared routines that accommodate the organizational needs of all involved [21]. In the case of creating shared routines, accommodating the broad needs of all members is necessary and builds communal support and shared understanding of those routines [19]. Yet, even as routines are created in a shared setting, fixed and negotiated aspects remain present.

Industries create shared routines to achieve compatibility of practices or save costs in the exchange of products or data. For example, the act of sharing data between organizations requires a standardized format and shared understanding to ensure that a receiver can accurately interpret encoded data. Before an industry agrees on a standard way of expressing routines, a negotiation for standardization occurs in which participants engage in complex negotiations [1] over which aspects of technologies and practices are included in the jointly created standard. This negotiation extends beyond the participants involved in the standardization and includes downstream users who engage the published standard in their own meaningful ways, which can inform future versions of the standard [6].

Standards represent fixed aspects of routines that are considered uniform across adopting organizations but the differing local contexts and backgrounds may lead to unexpected implementations due to deviating interpretations [2]. Adoption of standards often depends on the cultural fit [2] and whether organizations can develop compliant local routines associated with the standard [18]. The adoption of standards is an internal process to organizations and unless audited and certified, business partners can often not judge whether local implementations are uniform [17].

Organizations can benefit from investing and engaging in standardization processes [15]. Benefits arise from coupling internal product development with shared standard development to ensure future conformance by adjusting product development or by influencing standards based on a product strategy. Further, participants of the standardization process can express organizational expectations for a standard and through the contact with other experts learn to apply the standard in more effective and productive ways [15]. Specifically, organizations engaged in the standardization process benefit from the expertise gained by employees in the negotiation with other organizations which helps to overcome knowledge boundaries [19,21].

Standards can be developed within open source communities [25] which provide platforms for new forms of shared innovation, particularly for technologies that can benefit all involved participants [11]. Standard development in open source communities enhances the process through early implementation, testing, and experience-based evaluation and refinement [25]. Issues associated with formal standards are mitigated in communally developed standards, including lack of clarity of the specification, licensing and patent issues, and deviating implementations [9].

When developing standards communally, organizational engagement varies [5]. One approach uses communal standards internally but does not interact with the community in their development. This approach is encouraged, since some users might later decide to contribute back, spread the word, or contribute in invisible ways, e.g. educate others on their use [3]. Another approach provides direct engagement with the community through bug submission, new feature requests, and descriptions of how the standard has been implemented locally. This often entails dedicating employees who participate in the community, to engage in operational and strategic discussions, and to even provide resources to the community such as hardware or funding [5].

To adopt standards, organizations might have to change their own practices, find a way to work around the limitations of a standard to support local routines, or seek guidance from the standards community directly. We focus our research to understand how members of an open data standard community play a role in the interpretations of communally defined routines, how these interpretations influence the adoption of routines, and finally, how organizations guide the advancement of the routines within a community – specifically in the context of SPDX.

RESEARCH DESIGN AND METHODS

This case study is part of a four-year, qualitative field study regarding organizational engagement with open source communities. Research team members actively engaged with the SPDX community and were contributing members to the development of the SPDX specification for over two years. Additionally, members from the research team presented and discussed their SPDX community development work at ten Linux Foundation conferences, and ran focus groups at three Fortune 500 companies on organizational engagement with open source communities. Finally, the research team hosts open source tooling related to the deployment and use of the SPDX specification. As such, we leveraged our longstanding direct engagement with the SPDX community members to construct an assurance case design approach [10] that we used to define our interview questions.

Assurance Case Design Approach

Stemming from our direct engagement, we identified recurring claims regarding engagement with the SPDX specification and community. The researcher-identified claims did not determine the answers to our research questions. Instead, the claims provided a logical starting point from which to construct our structured argumentation method based on Goal-structuring Notation (GSN) and derive our interview questions [16].

The explicit and logical argumentation structure of GSN combined with defeasible logic [14] produces an assurance case. In our application of an assurance case, a top-level claim regarding engagement with SPDX was created and further refined into sub-claims through a series of rebuttals that can introduce doubts in the top-level claim. The rebuttals were informed by our longstanding direct engagement with the SPDX community.

Through sub-claims, the rebuttals (i.e., doubts) are addressed and eventually substantiated or countered via evidence collected through empirical observations – interviews and a focus group in our case. As sub-claim doubts are eliminated, the assurance in the top-level claim increases [14]. Such induction promotes high assurance by surfacing and

addressing critical issues rather than supporting the top-level claim merely by observing similar repetitions through enumerative induction.

The assurance case design approach is novel. Unlike hypothesis testing, our approach does not develop *a priori* hypotheses and does not evaluate their truth statement. Rather, the assurance case ensured rigor and internal validity in the development of the interview protocol with a top-level question that reflects the intended purpose of the SPDX community. The creation and existence of SPDX is predicated on the fact that it will improve OSS risk management in organizations. This is not a hypothesis that the researchers (us) came up with. The interview protocol was developed to further investigate if this is actually happening based on the SPDX community activities and organizational engagement in those activities.

Structuring the Assurance Case

From the assurance case approach, the how and why research questions were analyzed to derive a top-level claim per the assurance case notation. Our top-level claim captured OSS risk management routines in an organization:

> **Top Claim C0:** Use of the SPDX specification improves OSS risk management routines in an organization.

Sub-claims in the assurance case stem from the top-level claim and direct attention towards the specific characteristics of the research questions. As part of this process, we introduce rebuttals that challenge the top-level and sub-claims. Each rebuttal expresses a reason for doubting that claim. This argumentation continues until a sub-claim can be directly supported by concrete evidence. One branch of this logical argumentation produced these rebuttals and claims:

> **Top Claim C0:** Use of the SPDX specification improves OSS risk management routines in an organization.
>
> **Rebuttal R1:** Unless the SPDX specification is deemed complex for operational needs of local OSS risk management routines.
>
> **Sub-claim C1:** Stakeholders have necessary guidance to correctly interpret the SPDX specification for adopting it in their local OSS risk management routines.
>
> **Rebuttal R1.1:** Unless SPDX adoption into local routines is ad-hoc.
>
> **Sub-claim C1.1:** Stakeholders have access to vetted strategies for SPDX adoption into their local routines.
>
> **Evidence E1.1:** List of strategies to adopt SPDX in local routines.

To develop the interview protocol, each claim and sub-claim is explicitly linked to a question in the interview protocol. For the claims above, the associated interview questions are:

> **Claim C0 → Question Q0:** In the context of software exchange, could you describe your organization's OSS risk management routines?
>
> **Claim C1 → Question Q1:** How did your organization become familiar with or adopt SPDX?
>
> **Claim C1.1 → Question Q1.1:** Can you speak about SPDX adoption strategies in your organization and how those strategies have been informed (i.e., through the SPDX website, discussions in the SPDX community, upstream and downstream vendors, or elsewhere)?

Responses to the interview questions created evidence. All questions were general enough to invite answers that provided insights beyond the evidence we hoped to collect. Through the GSN argumentation structure, the evidence was explicitly linked to the claims that they support or reject. Interviewees, when asked an open-ended question whether they could think of a question we did not ask but should have asked, were satisfied with the breadth and depth of the interview – providing face validity on the interview protocol.

To offset concerns that the assurance case may not be representative of organizational OSS risk management, we performed a preliminary validation with representatives of the Linux Foundation and incorporated their feedback. The full argumentation structure is available online.[1]

Data Collection and Validation

We relied on semi-structured interviews to collect evidence for the assurance case. The interview protocol is available online.[2] All 15 interviewed organizations agreed to be named including, ARM Ltd., Black Duck Software Inc., Dimension Data North America Inc., GitHub Inc., Intel Corporation, Micro Focus International plc, NexB Inc., Palamida Inc., Qualcomm Technologies Inc., Red Hat Inc., Siemens AG, SUSE plc, Texas Instruments Incorporated, and Wind River Systems Inc. We recorded and transcribed a total of 14 interviews, resulting in approximately 10 hours of recording, and had two interviewees decline recording where we relied on copious notes. Immediately after the interviews, interviewers wrote personal debriefs to capture personal perceptions, observations, and thoughts from the interview.

Following the interviews, we created a practitioner-oriented slide deck[3] to present the collected data to SPDX community members. Two members from our research team attended the 2017 Linux Foundation Open Source Leadership Summit and presented the interview data as part of a one-hour focus group as a way to share and collect comments on the data broadly. We presented recurring sentiments gathered from the interview data, without expressing how we, as a research team, understood how the SPDX specification influences or is influenced by organizational risk management routines. A total of 15 SPDX members attended the focus group, some of whom were interviewed in the project earlier. The focus group did not dispute the recurring sentiment outlined in the

[1] https://github.com/SPDX-CaseStudy/files/raw/master/AssuranceCase.png

[2] https://github.com/SPDX-CaseStudy/files/raw/master/InterviewProtocol.docx

[3] https://github.com/SPDX-CaseStudy/files/raw/master/FocusGroup.pptx

presentation, generating discussion, not questions, about the data – providing face validity on the data itself.

Data Analysis

Data analysis was performed by all three members of the research team. The transcribed interviews were imported into NVivo software and recurring themes were coded in-vivo. These themes were the basis of the presentation given to the SPDX members to verify the validity of our data [20]. The presentation included the themes and supporting quotes from the interviews.

As the general analytic strategy for the study we chose to rely on theoretical propositions [26] as manifest in our assurance case. The assurance case builds a bridge between the dualities of routines for OSS risk management centered around SPDX. Through the assurance case, specific patterns of behavior in an organization in interpreting and using the fixed aspects prescribed by the SPDX specification are investigated. Each sub-claim and related interview question in the assurance case were designed to investigate the synergy and breakdowns in the patterns of behavior when enacting an OSS risk management routine.

For answering our research questions, we composed an effects matrix of direct quotes [20] to display answers to each interview question across our dataset and followed the pattern matching analytic technique [26]. Every company is represented by one row for each evidence in the assurance case with three columns: supporting evidence, additional information, and counter example. The matrix display allowed us to visually validate the prevalence of themes and sentiment towards our claims [20]. The content of the effects matrix directly provides evidence for the assurance case. The case study is presented in the linear-analytic structure [26].

FINDINGS

Stemming from our top-level claim – use of the SPDX specification impacts OSS risk management routines in an organization – we found that the communally developed SPDX specification has impacted the local OSS risk management routines. The organizations we interviewed are engaged in the development of the SPDX specification and are preparing their organizations to be SPDX compliant. Some started providing SPDX documents with their software to customers for learning and educating customers on SPDX. In an effort to further support this top-level claim, we next discuss the five top-level rebuttals that challenge the claim.

> **Top Claim C0:** Use of the SPDX specification improves OSS risk management routines in an organization.
>
> > **Rebuttal R1:** Unless the SPDX specification is deemed complex for operational needs of local OSS risk management routines.

This rebuttal reflects the communal pressure on internal OSS risk management routines. The pressure comes from a large, complex and formal specification to be interpreted and adopted. If the specification is too complex, the goal of achieving compatibility of practice and cost savings might be impeded because local interpretations are made difficult.

When asked about the SPDX specification, organizations referred to the complexity of the specification as a barrier to initial feasibility and adoption. The complexity is perceived in the large number and partially optional fields that the specification supports and the formatting of the SPDX doument which requires tooling to generate and use.

> *Excessive complexity is getting in the way of adoption.*[4]

Despite the discussed complexity of the SPDX specification, organizations that worked with it found the specification straight forward in how it should be used and get support from the SPDX community to overcome knowledge barriers for implementing the external routine locally.

> *[The SPDX specification] is quite a document. It took me awhile to read. Actually, what you need to output is understandable when you get down to it.*

Further, interviewees identified cases where the SPDX specification integrates with their OSS risk management routines. This includes, being able to produce and import SPDX documents.

> *Our business driver was to reduce the cost of distributing license information which we achieved by switching to SPDX documents only.*

In many interviews, we found that a key strategy towards SPDX adoption was the use of the SPDX License List even prior to the ability to produce and import SPDX documents. As a subset of the full SPDX specification, the SPDX License List reduces OSS risk information complexity through short identifiers for open source licenses (e.g., BSD-3-Clause). The short identifiers allow developers to replace long license text in each source file with the SPDX short identifier to indicate the applicable license. Such use of the short identifiers improves the quality of automatically scanned license reports because ambiguity is eliminated. The License List is perceived as highly valuable in simplifying OSS risk management routines, for example in inter-personal communication where the shared understanding of the short identifiers improves clarity, eliminates unnecessary verbosity, and avoids uncertainty.

> *First of all, was adopting the standardized license names and identifiers. We had all [open source license names] in a non-standard way and we said, let's do a mapping of all the different ways to name a license. To standardize, let's switch the names to be the standard license names and surface those short hand identifiers because those are so much easier to communicate.*

To summarize evidence for rebuttal R1 – unless the SPDX specification is deemed complex for operational needs of local OSS risk management routines – we found that the

[4] In most cases, only one representative quote is chosen in our analysis.

complexity of the specification was a significant barrier to adoption upfront. This finding should caution the SPDX community to discuss ways to address specification complexity and bloating. The SPDX specification is well-defined and community support helps with implementation, but does not provide well-defined gradations for organizations that perceive varying levels of OSS risk or are at different levels of maturity with respect to their OSS risk management routines. A full scope SPDX document is going to be onerous for organizations that do not have a large portfolio of OSS exchanges in supply chains or OSS use in mission critical applications. Specification complexity was easy to overcome for organizations that were engaged in the SPDX community or had started to use SPDX short identifiers in their organizational routines. Many of these early adopter organizations also had a clear business driver or opportunity associated with OSS risk management.

Top Claim C0: Use of the SPDX specification improves OSS risk management routines in an organization.

Rebuttal R2: Unless the information recorded in an SPDX document does not support local OSS risk management routines.

This rebuttal reflects the pressure that local routines put on the SPDX specification. If the SPDX document supports local OSS risk management routines, then the shared creation of the standard succeeded. Conversely, an SPDX document that is useless to organizations can indicate that either the shared routines created through SPDX do not meet local needs or that the SPDX specification is an insufficient compromise between divergent local interpretations.

SPDX released version 2.1 early 2017. Many organizations we interviewed were still working with version 1.2 of the SPDX specification. In version 2.1, expression of relationships between package elements was a major addition. Version 2.1 also added the ability to record any known vulnerabilities in the described package. The organizations we interviewed were involved to various degrees in the development of the new versions of SPDX specification. As such, they had insight into the intentions of the new SPDX specification and how it could be applied in the organizational OSS risk management routines.

I would say right now we're kind of just using all of the basic required fields up to the 1.2 spec level. We're not yet using things like relationships or anything like that just because we haven't really grown into it. We see that kind of stuff being useful, especially for our customers in the future.

Some organizations perceived the specification as being too rigorous or sophisticated while others saw value in most information recorded in SPDX documents. The match between the features of the specification and the needs for the local OSS risk management routines are important in the consideration for adopting the SPDX specification. The two representative quotes exemplify the divergent views:

I think it strikes me as being more rigorous than is necessary.

I think most of the information which is required, or what the standard has defined, [is] really necessary.

As such, the data captured in an SPDX document was not universally aligned with local OSS risk management routines. The relationships between SPDX documents and the level of tracking software artifacts varied. The following quotes show again the variety of uses that the SPDX specification supports and that the value some perceive from tracking licenses at the level of code snippets is not seen favorably by others who cannot justify the extra effort.

It would be rare for me to think of situations where I would go beyond the file level (one aspect of the specification). - I actually found from experience that if we try to describe package licensing at too detailed a level, we get information that is too complex to be useful.

We found that file level is not enough, that there are often snippets that could have an effect on our file and on the entire package.

Further, organizations pointed out that SPDX documents were not designed to be used for internal OSS risk management routines but that it is an exchange format that is only relevant when providing the information downstream. For internal OSS risk management routines, organizations are using their own data format or databases that aligns best with other operations or data management routines. The SPDX specification combines the many local practices through a process of combining innovations.

[In the SPDX group] we talked about the merits of different fields, how to characterize them, and how to serialize formats.

However, organizations reported that the development and advancement of their internal data structures and routines are influenced by the SPDX specification. The naming of internal data fields was aligned with SPDX fields where appropriate. Ultimately, to produce SPDX documents, the data from internal data structures has to be mapped to SPDX fields. This is done through transformations where needed.

When I hear my guys having modeling discussions, I often say, "look at SPDX, if it's a coin flip what to call this field, let's go with the standard.'

In response to rebuttal R2 – unless the information recorded in an SPDX document does not support local OSS risk management routines – we found that organizations use different subsets of the entire SPDX specification depending on what makes sense in their local routines. Although the use of SPDX documents may not fully be part of internal routines, the information required to create such a document is being recorded in internal artifacts. For organizations that advance their OSS risk management routines, the SPDX specification seems to provide standard information that an organization can record to be compatible.

Top Claim C0: Use of the SPDX specification improves OSS risk management routines in an organization.

Rebuttal R3: Unless the organization does not require SPDX documents upon supply or intake.

This rebuttal reflects the level of adoption across the open source supply-chain ecosystem. Shared routines through standards are evident in the use of standard-compliant artifacts such as SPDX documents, that are transferred and understood between organizations. A lack of exchanging of SPDX documents could indicate that local interpretations of the standard are not aligned across organizations and that local routines are unaffected by the creation of the shared standard.

Organizations did not require SPDX from upstream suppliers. The consensus is that producing SPDX documents requires a tool, is too much work, and, consequently, cannot be expected from open source suppliers which may be mostly communities of volunteers.

> *[We don't require SPDX] from our suppliers, in that outside of open source we don't use a lot of third party content within our products. It's not really relevant from that perspective.*

For many organizations, there was no advantage to being an early adopter. Organizations had reservations for asking SPDX documents from commercial suppliers as the SPDX specification is not yet well understood and the adoption of SPDX is limited.

> *We're not asking them to do it because I don't think we've fully figured it out ourselves and I'm not going to ask a vendor to [provide SPDX documents] until we've got it nailed down and really understand what it means.*

For some organizations, a business driver for SPDX adoption is that they have to provide licensing information about their products to every customer and prior to SPDX there was no standard way to do so. SPDX documents allowed to reduce the work in supplying this information in a unique format for each customer. Customers were educated in the use of SPDX documents and the benefits of switching to the standard format.

> *The cost of distributing license information was our business driver for adopting SPDX.*

Others have started experimenting with SPDX and shipping SPDX documents with a limited set of products. The purpose is to learn how SPDX can be integrated in their OSS risk management routines. These efforts uncover challenges with SPDX, including the ability to produce and consume SPDX documents.

> *Very recently, we've started providing an SPDX summary of those licenses alongside copies of the licenses with one product. I'm not sure we entirely know how we want this stuff formatted ourselves. There's experimentation going on to learn what we want before we start [with] other products. Because once you do that it's really hard to change later.*

In response to rebuttal R3 – unless the organization does not require SPDX documents upon supply or intake – we found organizations experimenting with supplying SPDX documents but that challenges remain. SPDX adoption is not wide spread in software supply chains and when used, the patterns of behavior have yet to crystalize. The process of organizational compliance with SPDX requires organizations to reconsider their OSS risk management routines and make changes within their OSS supply-chain.

> **Top Claim C0:** Use of the SPDX specification improves OSS risk management routines in an organization.
>
> **Rebuttal R4:** Unless SPDX does not integrate well in to organizational training programs.

This rebuttal reflects organizational commitment to the SPDX specification. The local interpretation is influenced using individuals and their understanding of how the standard impacts their routines. Through training, an organization ensures that the local interpretation is consistent across employees, reflects best practices, and is aligned with intended use cases. A lack of training can lead to divergent understandings, inconsistent and non-standard use or avoidance of the SPDX specification, which defeats the purpose of the standard.

We found that the SPDX specification is rarely integrated in developer training. One of the reasons is the limited use of the SPDX specification in software exchanges.

> *Until the day comes when we would attempt to adopt the SPDX specification, I don't see how it would enter into our developer training.*

In many organizations, developers are not required to interact with SPDX documents, because specialized departments are responsible for reviewing license compliance and creating SPDX documents for software package exchanges.

> *[Developers] know about the fields that they have to fill in their request, about license and stuff like that. I'm not sure they are aware of SPDX.*

When SPDX is integrated in developer training, the focus is on the aforementioned license short identifiers and the remaining SPDX specification is only mentioned. Participants often point out that the short identifiers simplify communication and developers are required to use them in their daily work.

> *I definitely mention SPDX as the standard. We don't go through its breakdown, of the fields and the structure.*

In a few organizations, mainly tool vendors that implement SPDX as part of their service, we did find that the SPDX specification is an integral part of developers' training and daily routines. The training is informal and knowledge about SPDX is shared through everyday work routines.

> *Our training is relatively informal so it's mainly when we have weekly [meetings] and our audit of our internal and external work. It's part of just an ongoing discussion. We're members [of the SPDX community], we follow the standards, so it's not a particularly formal training. We use Slack for our business and there's an SPDX chat, and so we're constantly talking about things that are going on in SPDX.*

In response to rebuttal R4 – unless SPDX does not integrate well in to organizational training programs – we found that

the SPDX License List does find its use in training but broadly, developers are not trained on the SPDX specification. Many participants indicated that SPDX was only mentioned in developer trainings.

> **Top Claim C0:** Use of the SPDX specification improves OSS risk management routines in an organization.
>
> **Rebuttal R5:** Unless engagement with SPDX community is difficult.

This rebuttal reflects the importance of engagement in a community of practice. Participation in the standards development process is perceived as beneficial for (1) influencing the standard to meet local needs, and (2) learning how to use the standard and reflecting on local interpretations with the community. The former reflects the process of shared innovation and the creation of shared routines through standards. The latter informs how organizations interpret and implement the standard.

Some interviewees were co-founders or long-standing members of the SPDX community and made significant contributions. For these members, the community is a place to meet like-mined people, to exchange best practices, and codify them in a specification.

> *I look at SPDX as, to a certain extent, our primary trade association. So, all of us in the business, little guys like us and the big ones like Black Duck were all there, we all know each other from there.*

Other interviewees had a more "arm's length" perspective. They described themselves as community observers. They are interested in staying up to date with how the industry is shaping up and evaluate for themselves whether or not to use SPDX. Some reported that they have introduced features into the SPDX specification to better support their own OSS risk management routines.

> *I guess, my impact is that I feed stuff into the License List on occasion and give a bit of a review comment on the technical side, on the specification and things that I find ambiguous or don't really know how to implement. It's nice to see some of those fitting into future specifications.*

Additionally, some reported that the development of the SPDX specification is going in the wrong direction or that it was becoming too complex. Some stay silent about their concerns because others appear to derive value from certain feature, while others voice their concerns explicitly.

> *The other thing is that SPDX, and I made this point also in the general SPDX meeting, at least in my opinion - it's evolving in the wrong direction.*

Finally, engagement with the community has changed perspectives in some cases on OSS risk management and helped improve local risk management routines.

> *I've actually adjusted my thinking about what we need to provide. So, we weren't collecting copyright statements before. Seeing that in the SPDX specification has sort of encouraged me to start collecting those. It's helping to push us to a better situation.*

In response to rebuttal R5 – unless engagement with SPDX community is difficult – we found that the organizations who are participating derive value from the conversations and are able to help shape the SPDX specification to support their local OSS risk management routines. The organizations that do not participate in the creation of shared routines but engage as observers stay up to date on the development, arrive at their own interpretation of the specification, and consequently determine how to implement SPDX to support their local routines. Some organizations are comfortable with only proxy representation through consultants engaged in the SPDX community. Their local OSS risk routines are not burdened by limitations or the complexity of SPDX as the translations to local routines is skillfully taken care of by consultants. This strategy may also alleviate some of the concerns mentioned in previous rebuttals. See Table 1 for summary of all rebuttals and what we found.

Rebuttal	Elimination Summary
Rebuttal R1: Unless the SPDX specification is deemed complex for operational needs of local OSS risk management routines.	**Rebuttal R1** is not eliminated for organizations just starting with SPDX. Organizations engaged in the SPDX community for a long time easily address the rebuttal.
Rebuttal R2: Unless the information recorded in an SPDX document does not support local OSS risk management routines.	**Rebuttal R2** is eliminated in most organizations by mapping parts of SPDX to local OSS risk management routines.
Rebuttal R3: Unless the organization does not require SPDX documents upon supply or intake.	**Rebuttal R3** is not eliminated in most organizations as SPDX adoption in OSS supply chains is not widespread. Few organization are starting to use and ship SPDX to customers.
Rebuttal R4: Unless SPDX does not integrate well in to organizational training programs.	**Rebuttal R4** is partially eliminated by the inclusion of License List in developer training and best practices. However, there is only mention of SPDX in formal training.
Rebuttal R5: Unless engagement with SPDX community is difficult.	**Rebuttal R5** is eliminated in organizations that directly participate, observe, or engage through proxy representation in the SPDX community. SPDX community is perceived as open and inviting.

Table 1. Rebuttals and summary of findings.

DISCUSSION

In this research project, we explored questions of (1) how organizations participating in the SPDX community described their local interpretations of communally structured OSS risk management routines, (2) how these local interpretations influenced the extent of their SPDX adoption, and (3) how these member organizations sought to guide the advancement of the shared SPDX specification. Sensibly, organizations described their local interpretation of the SPDX specification differently. The local interpretation

sparked a number of responses, including the full standard used for exchanging licensing information, the standard becoming a guiding influence in the advancement of local OSS risk management routines, and the standard being questioned as too complex for local needs. The most common engagement came from the SPDX License List short identifiers which simplify internal routines and the exchange of information. Even when the SPDX specification was not fully used, it influenced many organizations' thinking, data collection, and governance.

The duality of routines – as both influencing and being influenced by community engagement – was apparent in the ways that SPDX members shared and deployed the specification. The business driver appeared to be a deciding factor for the extent to which an organization engaged with the SPDX specification and aligned its routines. While extant literature treated external routines that are taken into the local context as codified knowledge that is easily misunderstood and difficult to deploy [19,21,22], we found contrary information in open communities. Organizations involved with the SPDX community shared their experiences and interpretations with other community members and negotiated changes to the shared routines by suggesting changes to the SPDX specification itself. Misunderstandings were resolved in the negotiation process. The divergent implementations resulting from different contexts and backgrounds in each organization became a source of innovation that was shared with the community and reflected in updated releases of the specification [6,7]. The challenge that the SPDX specification may now face is to balance which innovations to include [12], while containing the complexity that could impede use by new and existing adopters.

In the case of SPDX, leveraging open source communities for standards development: (1) advances the specification to better align with local routines and (2) improves local routines based on the codified specification. Within this duality, communal negotiation over features exemplified that the specification was a source of flexibility by accommodating the different forms of risk related work by members, while at the same time serving as a source of inflexibility by requiring those engaged with the specification to be attentive to communally agreed upon features.

Co-creating Risk Related Best Practices

Observations related to OSS risk and SPDX share parallels with other risk related data exchange standards. We found that organizations attempted to address OSS risks close to delivery. This is also observed with security risk. While it is better to consider security early in the software development lifecycle, it is often done much later and closer to software delivery [13]. Similarly, with OSS risk management, rather than integrating and spreading the responsibility throughout the product lifecycle, it tends to be addressed primarily towards the end – using automated license scanning mechanisms. These automated mechanisms often fall short. An organization that we interviewed had much success by federating the OSS risk responsibility to every developer and every process in product development. Thus, eliminating the need for heavy weight processes closer to software release.

In response, it may be advisable to build a more granular data standard adoption scheme with built-in gradation for different levels of OSS risk management maturity. Most successful security risk frameworks, starting with orange book,[5] have gradation built into them to accommodate different perceived design basis threats. With SPDX, a majority of the fields are optional to allow for gradation in maturity. However, this is not explicitly reflected in the specification. There has been community discussion around a SPDX lite version that reflects this sort of need.[6]

Design in a Responsive and Brokered Engagement

In complex software ecosystems that include both proprietary and OSS, the design of software is responsive to a highly dynamic landscape [12]. Software design is not a solitary experience, accomplished within a single organization. Instead, software design is a shared experience where participants are responsive to the environmental conditions that define choices. Similar to the way a flooded road defines a travel route, risk-related elements (e.g., licenses and vulnerabilities) define software design decisions, along with other elements including intellectual property management, corporate strategy, and community health. The creation of the SPDX specification is an improvement of the road markers that better declare potential risks inherent in OSS.

Interestingly, SPDX not only helps stabilize the complexities inherent in software design by allowing open source participants to respond more appropriately to software risks. SPDX itself entails responsive design as members engage in the duality of routines, informing and being informed by others in the community. The design of the SPDX specification entails a suite of communal responses to the wants and needs of members in mitigating risk-related concerns in OSS design.

To manage the complexity of the many voices and the commercial needs in the design of open source artifacts, neutral brokers such as the Linux Foundation now play important roles [24]. OSS design now readily exists in professional contexts [8], resulting in needs for community governance, codes of conduct, and marketing support. In these brokered engagements, design becomes considerably more structured and considerably less egalitarian [4].

SPDX is one community as part of an intentional collection of such communities. Within the Linux Foundation, other brokered communities include those that manage core

[5] http://csrc.nist.gov/publications/history/dod85.pdf

[6] https://wiki.spdx.org/view/Legal_Team/Minutes/2012-07-25

infrastructure (e.g., Network Time Protocol), provide open source training (e.g., OpenStack Fundamentals), and maintain commercially critical operating systems (e.g., the Linux kernel). Together, one community not only serves its own needs but can support aspects of partner communities (e.g., SPDX providing license declarations for the Linux kernel). As such, design in brokered engagements can include the intrinsic needs of any single community and extrinsic needs of a brokering foundation.

CONCLUSION

This paper makes four contributions. First, this paper contributes to research on routines by uncovering the complexity involved in the development of communal risk related open data standards. We demonstrated how a communal standard codifies aspects of OSS risk management routines deemed as best practices and how organizations engage with the standard to improve their local routines. Organizations engage in the standard development to test their local routines and compare them with other implementations to learn about better ways to accomplish the same goals. Engagement in the SPDX community was essential to ensure that the standard would satisfy organizational needs, inform local interpretations, and codify those interpretations for others to share. The embodiment of the shared routine in the SPDX specification served as a starting point for organizations to adopt the shared routine and engage in negotiation with others about how to interpret and implement the standard.

Second, this paper contributes to open source research by reporting how the SPDX project is changing the open source ecosystem by developing shared routines and encoding their fixed elements in the SPDX specification. The open source ecosystem is often viewed as a collection of communities that build on each other's code but are otherwise independent. Routines often spread through the use of shared tools, such as git, that become shared fixed elements in local routines, or through boundary spanning community members. We found that the SPDX members, through their engagement with the SPDX community, co-create routines that span organizations and open source communities but are not bound to the use of specific tools and rather define the fixed elements collectively.

Third, this paper contributes to standard setting literature by demonstrating how shared practices shape standards. Often, standards precede implementation and serve as fixed aspects of lived routines. We reported a case where the standard responded to the local interpretations, thus introducing a new perspective on the role of standards in routines. The definitions in the SPDX specification provide fixed aspects of local routines but through the community engagement the interpretation was negotiated and adjusted to meet changing local needs. The standard is fully developed in an open source community, not by the rules of a formal standard setting organization.

Fourth, this paper makes a methodological contribution by demonstrating the use of the assurance case driven case study design as proposed by Gandhi and Lee [10]. The assurance case guided the development of the interview questions and provided confidence that we addressed all challenges to the claims. Further, the assurance case facilitated the discussion of the research team, uncovered differing understandings, and ensured that detailed aspects were explored together. The assurance case served as an artifact in our own research routines – as a source of structure and knowledge.

Several questions and avenues for future research remain. Future research can investigate the details by which communally created routines and their embodiment in standards are locally interpreted and implemented. Future research can also investigate how the community driven standard development process compares to the process of standard setting organizations and consequently how these differences affect the local interpretation and adoption. Finally, this study was bound by a focus on SPDX community members, however, we know that SPDX is being adopted and used by organizations that do not participate with the SPDX community. We believe that including such organizations can reveal new lines of inquiry as the specification is deployed across the vast landscape of OSS engagement.

ETHICS

The study was reviewed and approved by our Institutional Review Board.

ACKNOWLEDGEMENT

Authors contributed equally and are listed in alphabetical order.

This project received funding through the National Science Foundation's Virtual Organizations as Sociotechnical Systems and the Innovation and Organizational Sciences Programs [VOSS-IOS: 1122642].

ORCID

Robin Gandhi orcid.org/0000-0002-2632-1692
Matt Germonprez orcid.org/0000-0003-2326-5901
Georg J.P. Link orcid.org/0000-0001-6769-7867

REFERENCES

1. James Backhouse, Carol W. Hsu, and Leiser Silva. 2006. Circuits of power in creating de jure standards: Shaping an international information systems security standard. *MIS Quarterly* 30: 413–438.
2. Anna Canato, Davide Ravasi, and Nelson Phillips. 2013. Coerced practice implementation in cases of low cultural fit: Cultural change and practice adaptation during the implementation of Six Sigma at 3M. *Academy of Management Journal* 56, 6: 1724–1753. https://doi.org/10.5465/amj.2011.0093
3. Jocelyn Cranefield, Pak Yoong, and Sid Huff. 2015. Rethinking lurking: Invisible leading and following in

a knowledge transfer ecosystem. *Journal of the Association for Information Systems* 16, 4: 213–247.

4. Kevin Crowston, Qing Li, Kangning Wei, U. Yeliz Eseryel, and James Howison. 2007. Self-organization of teams for free/libre open source software development. *Information and Software Technology* 49, 6: 564–575. https://doi.org/10.1016/j.infsof.2007.02.004
5. Linus Dahlander and Mats G. Magnusson. 2005. Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy* 34, 4: 481–493. https://doi.org/10.1016/j.respol.2005.02.003
6. Ben Eaton, Silvia Elaluf-Calderwood, Carsten Sørensen, and Youngjin Yoo. 2015. Distributed tuning of boundary resources: The case of Apple's iOS service system. *MIS Quarterly* 39, 1: 217–A12.
7. Martha S. Feldman and Brian T. Pentland. 2003. Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly* 48, 1: 94–118. https://doi.org/10.2307/3556620
8. Joseph Feller, Patrick Finnegan, Brian Fitzgerald, and Jeremy Hayes. 2008. From peer production to productization: A study of socially enabled business exchanges in open source service networks. *Information Systems Research* 19, 4: 475–493. https://doi.org/10.1287/isre.1080.0207
9. J. Gamalielsson and B. Lundell. 2013. Experiences from implementing PDF in open source: Challenges and opportunities for standardisation processes. In *2013 8th International Conference on Standardization and Innovation in Information Technology (SIIT)*, 1–11. https://doi.org/10.1109/SIIT.2013.6774572
10. Robin A. Gandhi and Seok-Won Lee. 2009. Assurance case driven case study design for requirements engineering research. In *Requirements Engineering: Foundation for Software Quality*. Springer Science + Business Media, 190–196. https://doi.org/10.1007/978-3-642-02050-6_16
11. Matt Germonprez, J. P. Allen, Brian Warner, Jamie Hill, and Glenn McClements. 2013. Open source communities of competitors. *ACM Interactions* 20, 6: 54–59. https://doi.org/10.1145/2527191
12. Matt Germonprez, Julie E. Kendall, Kenneth E. Kendall, Lars Mathiassen, Brett Young, and Brian Warner. 2016. A theory of responsive design: A field study of corporate engagement with open source communities. *Information Systems Research* 28, 1: 64–83. https://doi.org/10.1287/isre.2016.0662
13. Karen Mercedes Goertzel. 2013. A twenty-five year perspective. *CrossTalk - The Journal of Defense Software Engineering* 26, 4: 8–15.
14. John B. Goodenough, Charles B. Weinstock, and Ari Z. Klein. 2013. Eliminative induction: A basis for arguing system confidence. In *Proceedings of the 2013 35th International Conference on Software Engineering (ICSE)*, 1161–1164.
15. John Hurd and Jim Isaak. 2005. It standardization: The billion dollar strategy. *International Journal of IT Standards & Standardization Research* 3, 1: 68–74.
16. Tim Kelly and Rob Weaver. 2004. The goal structuring notation – A safety argument notation. In *Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases*.
17. Andrew A. King, Michael J. Lenox, and Ann Terlaak. 2005. The strategic use of decentralized institutions: Exploring certification with the ISO 14001 management standard. *Academy of Management Journal* 48, 6: 1091–1106. https://doi.org/10.5465/AMJ.2005.19573111
18. Tatiana Kostova and Kendall Roth. 2002. Adoption of an organizational practice by subsidiaries of multinational corporations: Institutional and relational effects. *Academy of Management Journal* 45, 1: 215–233. https://doi.org/10.2307/3069293
19. Julia Kotlarsky, Harry Scarbrough, and Ilan Oshri. 2014. Coordinating expertise across knowledge boundaries in offshore-outsourcing projects: The role of codification. *MIS Quarterly* 38, 2: 607–A5.
20. Matthew B. Miles and A. M. Huberman. 1994. *Qualitative data analysis: An expanded sourcebook*. Sage Publications, Thousand Oaks.
21. Jeppe Agger Nielsen, Lars Mathiassen, and Sue Newell. 2014. Theorization and translation in information technology institutionalization: Evidence from Danish home care. *MIS Quarterly* 38, 1: 165–A7.
22. Brian T. Pentland and Martha S. Feldman. 2008. Designing routines: On the folly of designing artifacts, while hoping for patterns of action. *Information and Organization* 18, 4: 235–250. https://doi.org/10.1016/j.infoandorg.2008.08.001
23. Etienne Wenger. 1998. Communities of practice: Learning, meaning, and identity. Cambridge University Press.
24. Joel West and Siobhán O'mahony. 2008. The role of participation architecture in growing sponsored open source communities. *Industry and Innovation* 15, 2: 145–168. https://doi.org/10.1080/13662710801970142
25. S. A. Wright and D. Druta. 2014. Open source and standards: The role of open source in the dialogue between research and standardization. In *2014 IEEE Globecom Workshops (GC Wkshps)*, 650–655. https://doi.org/10.1109/GLOCOMW.2014.7063506
26. Robert K. Yin. 2008. *Case study research: Design and methods*. SAGE Publications, Inc, Los Angeles, Calif.