EDITOR **DIRK RIEHLE**
Friedrich Alexander-University of Erlangen Nürnberg;
dirk.riehle@fau.de

Managing Your Open Source Supply Chain—Why and How?

Nikolay Harutyunyan, Friedrich-Alexander University of Erlangen-Nürnberg

More than 90% of software products include open source components, most of which are not directly added by your own developers. Instead, they are an inseparable part of the software supply chains that virtually all companies depend on. This article covers the related risks of ungoverned open source use and provides industry best practices to practitioners.

A recent European Commission report estimated that using free/libre and open source software (FLOSS) saves the European economy roughly €114 billion per year directly and up to €399 billion per year overall.[1] FLOSS components are an essential part of software infrastructure ranging from operating systems and web servers to media players and much more. Companies use such components to address their product requirements for nondifferentiating functionalities while focusing their internal development efforts on core differentiating features. By extension, software supply chains consist of multiple supplier tiers that all feed into each other and thus accumulate open source software that eventually gets into companies that sell complex products, such as original equipment manufacturers (OEMs).

WHY FLOSS GOVERNANCE FOR SOFTWARE SUPPLY CHAINS?

While FLOSS is highly critical and relevant for industry, many companies are unaware that they use open source software, either disregarding it or delegating it to developers. This approach of ungoverned open source use carries a number of the following potential risks for companies:

- legal risks caused by open source license noncompliance or incompatible licensing

Digital Object Identifier 10.1109/MC.2020.2983530
Date of current version: 4 June 2020

FROM THE EDITOR

Welcome back! This month's article on governing the open source software supply chain pulls together many different aspects from past articles into a comprehensive picture. A company needs to have an open source program office to coordinate all open source activities; it needs to understand the risks that stem from using open source, including open source hiding in components sourced from third-party suppliers; and it needs to actively manage those suppliers and their deliveries. Like all articles in this column, this one can stand alone, but I still recommend that you review past articles if you haven't done so yet to get the most out of this month's piece. Happy hacking, and be safe and healthy! — *Dirk Riehle*

- financial risks resulting from preliminary injunctions (sales stop) or supplier replacement costs
- technical risks stemming from the forced replacement of supplied open source software components.

As an example, let's look at the legal risk category. Companies have supplier contracts that provide some protection against the aforementioned legal risks. However, when it comes to open source license compliance, this protection is often overestimated. According to open source governance experts we interviewed, most companies at the end of the supply chain are much larger than their smaller suppliers. Once such a large company faces litigation over license noncompliance or copyright violation associated with FLOSS use[2] that stems from supplied code, it's possible but impractical to shift the legal responsibility to the supplier (or to a company further down the supply chain). If you adopt the latter strategy, you might end up running your smaller supplier out of business, which would create more problems, such as additional costs for replacing the supplied software and technical resources for maintaining it in house (when possible).

FLOSS GOVERNANCE BEST PRACTICES FOR SOFTWARE SUPPLY CHAINS

Our analysis of open source governance expert interviews suggests an industry best practice of working with the supplier from the get-go to ensure open source governance at the supplier's site during software development, as opposed to a checkup upon software delivery. Our data analysis suggests that some companies with an advanced understanding of supply chain FLOSS governance should focus their efforts on preventive steps, such as providing license checking and approval guidance during the development phase.

> FLOSS components are an essential part of software infrastructure ranging from operating systems and web servers to media players.

In a larger study of open source governance, we conducted and analyzed 21 expert interviews and FLOSS governance guidelines to learn about current industry best practices for using open source software in products.[3,4] This article discusses a small subset of our findings, providing insights for the following best practice categories related to supply chain open source governance:

- the supply chain management (SCM) policy and process
- preventive governance
- corrective governance
- bill-of-materials (BOM) management.

SCM POLICY AND PROCESS

SCM in terms of FLOSS governance is a complex and multifaceted task involving in-house software development teams, procurement offices, suppliers, and lawyers. Coordinating these stakeholders and ensuring a company-wide approach to open source governance is essential. Industry experts recommend that companies set up an SCM policy that strategically defines enterprises' FLOSS governance, informing all stakeholders.

Without a comprehensive policy, different parts of the company might apply dissimilar rules (or no rules at all) when dealing with open source software as part of the supplied code. For example, if a third-party software component is purchased to be used in a product, it is rarely checked for open source license compliance, especially in companies with little FLOSS governance awareness. Instead, firms rely on supplier contracts for any potential intellectual property issues, considering license noncompliance as one such concern. However, such clauses cannot guarantee that your products including open source components are license compliant. Instead, they can act only as a corrective measure if an issue is discovered by a customer, and even then, they are not a universal solution to the risks of ungoverned FLOSS use. Open source governance on the topic of SCM goes beyond supplier contracts and compliance checks, requiring a systematic approach and a company-wide policy.

In the course of our study, we found an industry best practice whereby an SCM policy should address governance aspects, such as

- company goals for supplier management
- metrics for efficient supplier management
- recommendations for automating supplier management through tools
- rules for suppliers that use open source components.

The policy should be defined by the open source program office to ensure a consistent approach to SCM within the company. It should be maintained, revised, and communicated through time. While the SCM policy defines the company's strategic take, it needs to be translated into the day-to-day processes of product development and software procurement. To achieve this, experts recommend operationalizing the policy through an SCM process.

The SCM process guides product managers, technical product managers, software developers, and others dealing with software supply chains. It also helps procurement managers and IT managers with external tasks related to SCM, such as dealing with supplier requests and contracts. The SCM process covers, among other things, assessing the open source governance maturity of a supplier, requesting supplier certification, and auditing suppliers.

The SCM process should be integrated into the daily workflows of the company. It should be easy to read, and it should be created in collaboration with the stakeholder engineers and managers. The process should be directly related to the tasks of software development and solve problems that engineers and managers face in their work when dealing with FLOSS governance.

PREVENTIVE GOVERNANCE

We found that industry experts recommend focusing on preventive open source governance. Companies should take steps to prevent supplier-related FLOSS governance issues. The initial preventive measure applies to choosing suppliers. We found a best practice to choose the right supplier, taking into account the supplier's open source governance and compliance awareness and maturity. To do so, companies should design supplier contracts with open source governance aspects in mind and consider requesting supplier certification. Such certifications can be conducted internally (self-certification) or using existing standard certification frameworks for FLOSS governance in supply chains. A leading framework on the topic is being developed by the OpenChain Project.[5]

Another best practice for preventive FLOSS governance of software supply chains focuses on supplier contracts. In certain cases, these contracts can include strict provisions, such as specific templates that suppliers must follow before any anticipated use of an open source component in the software development of the to-be-supplied code. A supplier would have to use the template to send open source component requests to the client for approval. The suppliers would also be encouraged to employ a similar practice with their own suppliers, which would, in turn, make the whole supply chain safer in terms of open source compliance.

CORRECTIVE GOVERNANCE

We found a number of industry best practices for addressing the issues of FLOSS governance and compliance caused by software supply chains. Going beyond the preventive measures, companies should also establish corrective open source governance in the context of SCM. Though preventive governance best practices mitigate the potential issues that result from lacking SCM, companies should be ready to address any cases of noncompliance as well as other issues caused by suppliers.

One expert recommendation is to conduct regular and surprise audits of software suppliers and their code to find potential issues, such as unintended open source licenses and missing copyright data. If risks are found, companies should proceed to mitigate them by assessing the threats' criticality and costs as well as by triggering supplier contract clauses and working with suppliers to take care of the issues, when possible. However, industry experts recommend against running suppliers out of business when conducting corrective governance, as the potential losses could increase with the bankruptcy of a small supplier.

> Industry experts recommend against running suppliers out of business when conducting corrective governance.

MANAGING BILLS OF MATERIALS

Most open source components end up in company products through software supply chains. Given the complex dependencies between open source components and libraries, as well as with companies' proprietary code,[6,7] enterprises need to use systematic and consistent instruments to ensure the complete and transferable documentation of open source use that is introduced by their suppliers and their own developers. BOMs are such an instrument; however, they need to be extended beyond the traditional format of merely listing the software components of a product (supplied or own). To address the specifics of open source governance, BOMs must include additional metadata for open source components, such as accurate license information, versions, copyright details, and export-restriction tags.

Leading industry experts recommend using existing standards for BOM documentation and exchange within

software supply chains. The current leading standard is called the *Software Package Data Exchange* (*SPDX*),[8] which is an open standard for communicating software BOM information that enables the specialized documentation of open source component metadata. Using this format can be of high value to an OEM because doing so ensures full transparency when it comes to the open source use in products, including the awareness of FLOSS components originating in the supply chain.

It's an industry best practice to ask your suppliers for their software's BOM in the SPDX format, which can be checked and combined with the BOMs from other suppliers, eventually forming the BOM of an OEM's final product. As a consequence, an OEM would have an updated and ready BOM for its own products if a customer requested it. The experts we interviewed mentioned further benefits of the aforementioned approach, including the use of a machine-readable format compatible with most open source governance and compliance tools as well as the method's industry-wide recognition as a leading standard.

During the course of our research on SCM in terms of FLOSS governance, we identified a number of industry best practices akin to the previously mentioned one. Putting some of these best practices together, we propose workflows or processes that practitioners can adjust and use in their companies. Figure 1 presents an example of such a workflow for BOM management. Starting with identifying the used FLOSS components and their metadata, companies should track and document this use, employing machine-readable exchange formats as well as ensuring the license compliance of and self-hosting backups for the used components.

This article presented a snapshot of our larger findings on the topic of open source governance,[3,4] building upon our previous work on managing software supply chains in the context of FLOSS governance.[9] Going beyond the presented best practices from industry, it is crucial to use tools to automate various aspects of open source governance, such as license scanning and compliance checking, documenting open source components as part of product architecture, and managing BOMs. C

REFERENCES

1. European Commission, "The economic and social impact of software & services on competitiveness and innovation (SMART 2015/0015)," Luxembourg: Publications Office of the European Union, 2017. [Online]. Available: https://op.europa.eu/en/publication-detail/-/publication/480eff53-0495-11e7-8a35-01aa75ed71a1
2. H. Schoettle, "Open source license compliance: Why and how?" *Computer*, vol. 52, no. 8, pp. 63–67, 2019. doi: 10.1109/MC.2019.2915690.
3. N. Harutyunyan, "Corporate open source governance of software supply chains,"

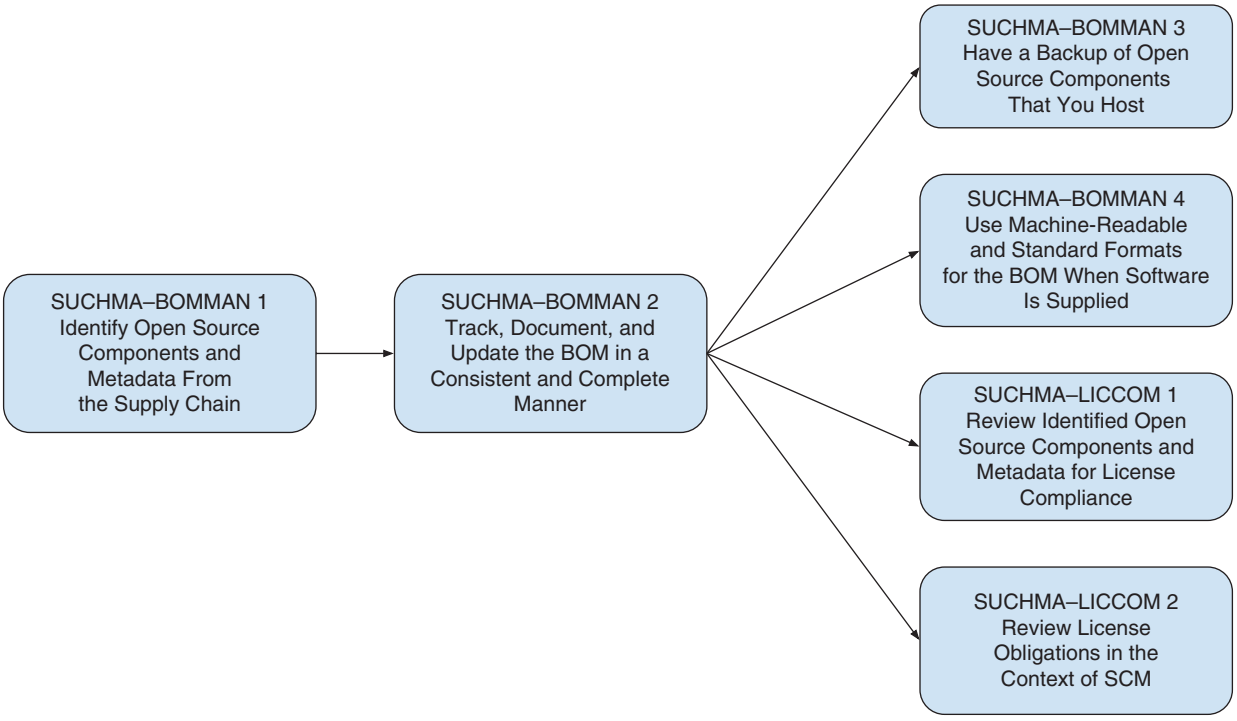

FIGURE 1. A workflow for BOM management.

Ph.D. dissertation. Friedrich-Alexander-Univ. Erlangen-Nürnberg, 2019. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bvb:29-opus4-122727

4. N. Harutyunyan and D. Riehle, "Getting started with open source governance and compliance in companies," in *Proc. 15th Int. Symp. Open Collaboration*, 2019, pp. 1–10. doi: 10.1145/3306446.3340815.
5. OpenChain. Accessed on: Mar. 25, 2020. [Online]. Available: https://www.openchainproject.org/
6. A. Bauer, N. Harutyunyan, D. Riehle, and G.-D. Schwarz, "Challenges of tracking and documenting open source dependencies in products: A case study," in *Proc. 16th Int. Conf. Open Source Systems*, to be published.
7. T. Gustavsson, "Managing the open source dependency," *Computer*, vol. 53, no. 2, pp. 83–87, 2020. doi: 10.1109/MC.2019.2955869.
8. Software Package Data Exchange. Accessed on: Mar. 25, 2020. [Online]. Available: https://spdx.org/
9. D. Riehle and N. Harutyunyan, "Open-source license compliance in software supply chains," in *Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems Impact Sustainability*, B. Fitzgerald, A. Mockus, and M. Zhou, Eds. Singapore: Springer, 2019, pp. 83–95.

NIKOLAY HARUTYUNYAN is an open source researcher and postdoc at Friedrich-Alexander University of Erlangen-Nürnberg, Germany. Contact him at nikolay.harutyunyan@fau.de.

Digital Object Identifier 10.1109/MC.2020.2993516