Open Source Software Governance: A Case Study Evaluation of Supply Chain Management Best Practices

Nikolay Harutyunyan
Computer Science Department
Friedrich-Alexander University Erlangen Nürnberg
nikolay.harutyunyan@fau.de

Dirk Riehle
Computer Science Department
Friedrich-Alexander University Erlangen Nürnberg
dirk@riehle.org

Abstract

Corporate open source governance aims to manage the increasing use of free/libre and open source software (FLOSS) in companies. To avoid the risks of the ungoverned use, companies need to establish processes addressing license compliance, component approval, and supply chain management (SCM).

We proposed a set of industry-inspired best practices for supply chain management organized into a handbook. To evaluate the handbook, we ran a one-year case study at a large enterprise software company, where we performed semi-structured interviews, workshops, and direct observations. We assessed the initial situation of open source governance, the implementation of the proposed SCM best practices, and the resulting impact.

We report the results of this study by demonstrating and discussing the artifacts created while the case study company implemented the SCM-focused governance process. The evaluation case study enabled the real-life application and the improvement of the proposed best practices.

Keywords

Best Practice, Case Study, Corporate Open Source Governance, Open Source Software, OSS, FLOSS, Supply Chain Management, Software Supply Chains

1. Introduction

Modern software products are built using open source software components, which are either incorporated directly by the software developers or come with supplied code. To address the former, we proposed industry best practices dealing with the component approval [17] and reuse [19] as an essential part of inbound open source software governance. As for the latter scenario, we recognized the importance of supply chain management (SCM) in open source governance, which led to our recent publications [14] [15] on this topic including a number of proposed best practices for dealing with software supply chains, individual suppliers, prevention and correction of FLOSS governance-related issues, bill-of-materials (BOM) management, and compliance.

Software supply chains are responsible for the majority of the open source code that ends up in software products. Surprisingly, it's also the more overlooked part by the corporate users of open source software, which often regulate the direct incorporation of open source components by their own developers, but not that of their suppliers. Even advanced companies in terms of FLOSS governance often focus on searching and selecting open source components rather than on supply chain management. For example, Google's internal guidelines for open source use governance explicitly ban the use of AGPL-licensed software in their products (Google's Internal Guidelines for Open Source Use Governance - https://opensource.google/docs/using/agpl-policy/), but don't mention one major source of potential violations of this rule - supplied code that could include AGPL-licensed code. Addressing this particular issue in our previous work, we recommend setting up a preventive SCM process alongside a BOM management process that helps companies review the supplier code for any potential FLOSS governance issues, including license compliance and copyright infringement [14] [15].

Our proposed SCM best practices stem from a large-scale study of open source governance experts in industry through a qualitative survey following Jansen [21]. The latter was the research method used in our exploratory work eliciting the best practices. In that study, we run a qualitative analysis [9] of 20 company guidelines and 21 expert interviews. Our data was collected from a diverse sample of companies with an advanced understanding of FLOSS governance, such as Google, Intel, Qualcomm, BMW, SUSE, and others. The result was the publication of proposed industry best practices [17] [19], which we made easily applicable by other companies with less experience in FLOSS governance. To achieve this, we used the actionable format of best practice patterns [10] [29]. When referring to best practices in this paper, we talk about problem-context-solution patterns for dealing with open source governance we discovered through

industry-based qualitative studies at expert companies. You can see an example of supply chain management best practice in Table 1.

This paper presents the next stage of our research focusing on the evaluation of the suggested best practices in a real-life context by having them implemented in a production project at an enterprise software company with no prior SCM-related governance in place. In this study, we chose a subset of the proposed SCM best practices and shared them with a partner company willing to implement them, which allowed us to observe the implementation process as part of a case study.

In this evaluation case study, we studied the trustworthiness of our results, following Guba's [12] [26] criteria for the trustworthiness of qualitative studies. Namely, we considered the credibility (the degree to which we can establish confidence in the truth of our findings in the context of the inquiry), dependability (the degree of consistency of the findings and traceability from the data to the results), confirmability (the degree to which the authors are neutral towards the inquiry and their potential bias effect on the findings), and transferability (the degree to which the findings of our study hold validity in other contexts). While the former three were addressed in the original exploratory study, the transferability couldn't have been, as we needed to evaluate our findings in a different context, which led to this research. Transferability is the degree to which the findings of our study hold validity in other contexts. To evaluate the transferability, we looked at how our findings could be generalized and applied at companies with limited SCM-related corporate open source governance in place. This strategy has been recommended by researchers evaluating the trustworthiness of qualitative research projects [28] [31] [32]. Previously, we conducted a similar study for a different topic of open source governance - how companies get started with managing their open source use [16]. This study follows a similar case study design, but at a different company with more experience in FLOSS governance though lagging in the area of supply chain management (while the previous study focused on a company with no open source governance in place).

After finding a fitting company for our evaluation study, we asked the following research question:

***RQ**: How transferable are the proposed open source governance best practices for supply chain management in the context of companies with limited governance in place?*

The research question aims to evaluate the previously proposed best practices on the open source software integration into software supply chains and the standardization of open source use and adoption in commercial software.

We answered this question through a one-year case study informed by Yin [36] at a multinational enterprise software company based in Germany. The company already had some limited experience with open source governance, such as component approval and reuse, but was only a beginner when it came to supply chain management. They didn't have a unified way of dealing with BOMs or open source components originating in the supplied code. This was fitting for our study, as we planned to implement proposed best practices to address these issues.

We developed a case study protocol following Yin [36] and informed by design science research [20]. In this evaluation case study, the core artifact was the handbook consisting of open source governance best practices. We decided to not directly implement or interfere with the application of our proposed best practice handbook. Instead we took on the role of an observer to ensure the objectivity of the evaluation. We considered an alternative approach of action research, given our previous successful experience with case study research, we chose to use that method. While not directly involved in implementation, we guided the employees in different roles responsible for SCM and open source governance across the company. We answered their questions and helped clarify some points, but left the actual application of our original findings to them in order to decrease bias and to be able to evaluate real-life transferability just as it would be if another company took and implemented our recommendations.

In the course of a year, we tightly worked with our industry partners responsible for the case study company's open source governance and license compliance. We conducted 11 interviews, as well as two workshops, and visited the company site regularly to observe the implementation of the proposed SCM practices and processes. After analyzing the gathered data, we were able to evaluate our original findings and their transferability, as well as to prepare an improved version of the open source governance handbook with more applicable and comprehensive best practices. In this paper, we discuss our evaluation outcomes and present several artifacts created in the course of the implementation, namely the newly introduced supply chain management process for the open source components used across the 5000-employee case study company.

This paper is structured as follows. In section 2, we review some related literature. In section 3, we discuss the research method, including the case study protocol. In section 4, we present the results of the evaluation case study. In section 5, we go over the research limitations. Finally, section 6 concludes the paper.

2. Related work

We define corporate open source governance as a set of processes, best practices, and tools employed by companies to use FLOSS components as part of their commercial products while minimizing their risks and maximizing their benefit from such use [15] [22].

Related literature demonstrates the importance of the FLOSS governance for companies by covering some of the risks caused by the unmanaged use of open source software in products, such as open source license mishandling [8] [30], or not having the most up-to-date and secure versions of open source packages [5] [6]. Companies can address these and many other challenges by implementing corporate open source governance processes that focus on license and copyright compliance, inbound open source governance, component reuse, tool, and more.

In our previous work, we studied how successful companies perform open source governance focusing on getting started with FLOSS governance, industry requirements for the governance tooling, inbound governance including component approval [17] and reuse [19], and supply chain management [14].

As an emerging topic, there is limited research on the specific topic of software supply chains in the context of corporate open source governance. The existing research focuses on the supply chain management policy and process [1] [23], BOM management [11] [24] [33], and supplier standards [35] [7].

Germonprez et al. [11] focus on the compliance in open source supply chains to mitigate FLOSS governance risks. The authors discuss several risks resulting from the lacking open source governance in the software supply chains. The authors go on to propose the use of open source component and license scanning tools (e.g. FOSSology - an open source license compliance software system and toolkit) coupled with the bill of materials provided by the suppliers. The BOMs need to have a structured and standardized format for efficient supply chain management. One proposed format for BOM management is the software package data exchange (SPDX) - a standard format for storing the components, licenses, and copyright metadata associated with open source software packages [27]. In our original study [14] [15], we also found the importance of standards like SPDX, because they can facilitate compliance with free and open source software licenses by standardizing the way license information is shared across the software supply chain. We recommend that companies use a machine-readable format for their suppliers' bills of materials. In this study, we observed that the case study company gave priority to our recommendations for the standardized BOM metadata collection and exchange. Moreover, the SCM responsible employee at the company started planning the company-wide roll-out of an SPDX-compliant SCM process, which we will discuss in the results section.

Kemp [23] discusses operational compliance of open source use in companies, highlighting the importance of a supply chain management policy. The author proposes a SCM policy for open source governance that covers:

- open source compliance training for suppliers
- automated code scanning to facilitate discovery and recognition of OSS in the supplied products
- procedure to prepare a bill of materials with open source software specific metadata.

Furthermore, Kemp recommends using The Linux Foundation's Self-Assessment Checklist (https://www.linuxfoundation.jp/events/2011/05/self-assessment-checklist/) to efficiently assess supplier compliance practices and to engage suppliers in a discussion about compliance. We proposed the best practice *OSGOV-SUCHMA-PREGOV-1.1. Assess open source governance and compliance awareness and maturity* for companies to work with their supply chains to achieve a higher level of license compliance. In this evaluation study, we found that the case study company followed our recommendation, while adapting this best practice to their specific context and making the FLOSS governance maturity of the suppliers option in the current stage. Figure 1 shows the questionnaire designed during the case study to assess the open source governance and license compliance by the current suppliers.

Blecken and Hellingrath [4] researched software supply chains in the domain of humanitarian operations. They defined supply chain management as the integrated process-oriented planning and control of material, information and financial flows along the entire value chain from the customer to the raw material producer. Projecting this traditional definition onto software development, open source components are similar to raw materials used in producing products. In that sense, such components can be incorporated and shipped with software products several times going up the supply chain. In the end, however, they end up in final products, whose producer is responsible for all the components down the supply chain. This requires a systematic corporate approach, which we address in another proposed best practice - *OSGOV-SUCHMA-SCMPOL-1. Establish supply chain management policy*. We observed that the case study company didn't implement a company-wide policy as suggested, deciding instead to have a more operational process that incorporates their take on open source governance, including the issues of the supply chain management. We discuss this process in the results section.

When documenting the industry best practices for supply chain management, we had the future real-life

evaluation in mind, because such exploratory findings are more valuable when coupled with a hands-on evaluation. That's why we cast the proposed best practices in the form of interconnected patterns. Patterns and pattern languages (sets of interconnected patterns) have been used for this purpose in the past, namely by Hannebauer and Gruhn [13], when presenting an overview of the current state of research via 40 open source patterns. In our previous work beyond open source governance, we also used the same format of theory presentation in publications on corporate open sourcing [18] and user and experience design in software product lines.

Table 1. Example best practice OSGOV-SUCHMA-PREGOV-1.1. Assess open source governance and compliance awareness and maturity

ID: OSGOV-SUCHMA-PREGOV-1.1
Name: Assess open source governance and compliance awareness and maturity
Actor: Roles responsible for supply chain management, IT department, Procurement department
Context: Companies use supplied software components in their products, but choosing the wrong supplier in terms of open source governance and compliance maturity can cause potential financial and legal risks.
Problem: To avoid governance and compliance risks caused by your supply chain you → *choose the right supplier*. How can you do that if you have many suppliers?
Solution: You need to assess open source governance and compliance awareness and maturity of the potential suppliers to avoid potential risks of license violations or other governance issues. Companies can demonstrate their knowledge and experience in FLOSS governance by demonstrating their internal governance process, by providing detailed bill of materials with highlighted data on the used open source components and their metadata, as well as through → *governance and compliance certification*. Make sure to add a clause about governance awareness and maturity assessment, when → *designing supplier contracts*. Document the assessment results for the new suppliers in a centralized company-wide database that can be used by other divisions, which will help make decisions about contracting certain suppliers. A systematic and consistent awareness and maturity assessment is the best way to prevent future issues with open source license compliance. This can save financial and legal resources that would otherwise be spent on corrective governance if issues are identified regarding suppliers' use of certain open source components as the final responsibility for all the components lies with the final client (e.g. OEM) at the end of the supply chain.

Table 1 presents one of the proposed SCM best practices focused on preventive governance. *OSGOV-SUCHMA-PREGOV-1.1. Assess open source governance and compliance awareness and maturity* is one of the patterns implemented by our partners at the case study company. We discuss the outcomes in the results section.

We formalized this method in a paper that can serve as a guide for other researchers interested in presenting their theories using a similar approach [29]. Beyond the theory presentation, this proposed research method demonstrates that best practice patterns can be used in theory evaluation, especially through case studies. We used this technique when evaluating another subset of FLOSS governance best practices focused on getting started with corporate open source governance [16]. The current study has a similar setup and research methodology, but focuses on a different and more advanced aspect of FLOSS governance, namely supply chain management.

3. Research method

To answer our research question and to evaluate a set of SCM best practices, we followed the research methodology outlined in detail in our method paper [29]. We also used design science techniques to study our central artifact - the best practice handbook on SCM, which was part of our case study design. Having collected industry best practices for SCM in FLOSS governance, we chose a company willing to implement our recommendations, which we observed and used to evaluate the transferability of our proposed best practices. We chose the research method of a single-case case study following Yin [36], because the complex phenomenon of supply chain management could be best evaluated in a real-life context employing techniques, such as pattern matching [36]. The latter allows the comparison of the proposed theory and the actual approach undertaken by the case study company. This is possible given the context sections we included in each best practice pattern such as that in Table 1. These context sections were abstracted from the industry experts who contributed to the original recommendations. In the course of our evaluation case study, we compared the above-mentioned contexts with the context and the outcomes observed at the case study company.

In the first stage of our case study, we looked for potential companies that would fit the following profile: software product companies that already have some basic open source governance in place, but lack a process for supply chain management. In addition to this, the case study company had to commit to a one-year collaboration, during which we would work with partners at the company to observe their implementation of the SCM practices, conduct interviews, and workshops. We found two suitable Germany-based companies from our network - an

automotive company and an enterprise software company. The former was involved in a different case study, so we decided to pick the former as the subject of this case study.

The case study company, based in the Hessen region of Germany, is a large company operating internationally in the enterprise software industry both in the B2B and B2C domains. The company had begun regulating its open source use and set up the basic open source governance processes focusing mainly on FLOSS license compliance. However, they recognized their lack of governance when it came to supply chain management, which they wanted to address. This led to a synergic collaboration in which the company established a SCM process and we supported them, while gathering data for this study. We anonymized the company per their request.

From June 2018 to May 2019, we studied the open source use and governance of the case study. We focused our study on the centralized team responsible for open source governance and compliance with the company. This team worked company-wide on the issues of open source compliance, which was a sign of the early FLOSS governance maturity (as was planned during case study sampling). We conducted 11 one- to two-hour interviews with managers, developers, procurement, and compliance officers (in different locations of the company) using the interview questionnaire designed for the evaluation of the proposed SCM best practices.

Our goal was to evaluate the key SCM best practices in the context of a production-grade project at the chosen company. We focused on the proposed recommendations covering preventive and corrective FLOSS governance, as well as managing bills of materials. Given the case study company's initial governance maturity, we were able to evaluate this more advanced part of open source governance in a real-life setting.

In the course of the case study, we operationalized the overarching research question into specific evaluation criteria, chose relevant research techniques suggested by Yin [36], outlined a case study design, developed a case study protocol, selected a subject company through theoretical sampling, iteratively collected data, refined the study design, analyzed the gathered data, derived, and presented the results.

To guide our study and to ensure rigorous results, we developed a case study protocol ahead of the study and followed it throughout. According to the protocol, we planned to guide the implementation of our open source governance handbook (set of best practices), without directly interfering in the details of the implementation. This was an explicit decision, which was one of the reasons to choose the case study research method over other alternatives, such as action research, which requires a more direct involvement by the researchers.

We started the case study by assessing the initial state of SCM governance at the company, which was followed by the implementation of the proposed practices and their evaluation using the following transferability criteria: completeness, variability, structure, comprehension, understandability, applicability, relevance, significance, and usefulness. We drew from the literature in different fields to choose the most appropriate evaluation criteria for qualitative studies. We found that the transferability of a qualitative theory can be evaluated using the measures of applicability, relevance, understandability, and usefulness [31] [2]. Another evaluation criterion of the comprehension was proposed by Bitsch [3]. Finally, some other evaluation criteria included the structure, completeness, and variability of qualitative findings [25] [28].

Similar to our previous case study on getting started with open source governance [16], this study is both descriptive and explanatory. It is descriptive because it produced a report of the case study company's initial FLOSS governance state. Additionally, we describe the process of implementing some best practices at the company. The study is also explanatory, because the results provide an analysis of the shift from ungoverned SCM to open source governance including the successful and unsuccessful implementation instances observed.

In the course of the case study, we conducted 11 semi-structured interviews with the stakeholder employees responsible for the implementation of the SCM practices as part of the larger open source governance initiative at the company. These employees included people in supplier management roles at the procurement department, as well as open source license and compliance experts, developers, and the management of the centralized team responsible for the company-wide FLOSS governance. During the interviews, we asked questions about the implementation of specific best practices, and about the general feedback and experiences with the proposed set of best practices.

The data gathered through the interviews completed our notes from the several workshops conducted at the company in the early stages of the case study. During these workshops we introduced different parts of the SCM best practices, provided their context and implementation guidelines. These workshops included people beyond the interviewees and had an educational objective. Our case study was part of the company's overall push towards more regulated open source governance, which brought together people with various roles and from different parts of the company hierarchy. We analyzed the workshops and the Q&As that happened during these workshops to form a more comprehensive picture for our evaluation.

Beyond the interviews and workshops, we also sought employee feedback and questions on the proposed recommendations in the form of documentation, and artifacts created in the process of implementation. We present and discuss some of the collected artifacts in the results section. We analyzed the data from our evaluation interviews, as well as the notes from the direct observation, documentation and artifact reviews, with the goal of evaluating the different criteria outlined earlier.

In the results section, we describe the modifications and the adjustments the company had to make to our recommendations, as well as the successful and failed experiences. As mentioned earlier, a key technique employed in our analysis was pattern matching [34] [36]. This allowed the comprehensive comparison of the originally proposed best practices based on the industry expert assumptions to the actual implementation at the case study company with the specific company context in mind. As a result, we tried to distill the context-specific transferability issues and the more systematic pitfalls of our proposed theory.

4. Results

In the course of our one-year case study, we evaluated the proposed industry best practices for supply chain management in open source governance in the following subcategories:

- Supply Chain Management Policy - 3 best practices
- Supply Chain Management Process - 5 best practices
- Preventive Governance - 4 best practices
- Corrective Governance - 4 best practices
- Bill of Materials Management - 4 best practices
- License Compliance for Supply Chain - 2 best practices.

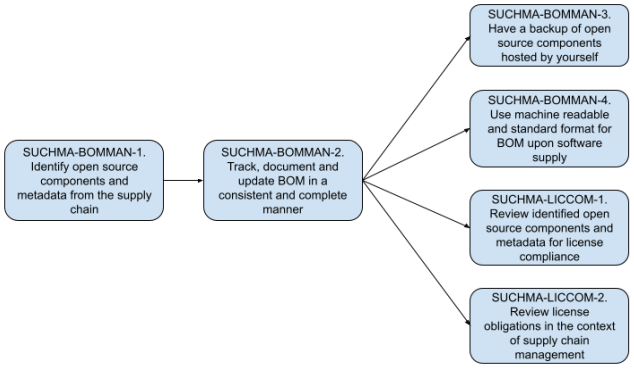

Figure 1. A workflow of interconnected SCM best practices focused on bill of materials management

Table 1 presents an example recommendation from the Preventive Governance category. Tables 2 and 3 present further best practice patterns focused on software supply chain management, which were evaluated at the case study company. Our previous publications provide an overview of the categories [14], as well as the detailed best practices in the Appendix B [15].

Table 2. Example best practice OSGOV-SUCHMA-SCMPRO-5. Use tools to automate supplier management

ID: OSGOV-SUCHMA-SCMPRO-5
Name: Use tools to automate supplier management
Actor: OSPO (Open Source Program Office), Roles responsible for supply chain management
Context: Companies often have hundreds of suppliers. Each supplier provides multiple software deliveries with multiple open source components in each, as well as open source software provided by tier 2 and other suppliers.
Problem: It is not possible to manually deal with the complexity of software supply chains. How can companies deal with this issue in parallel to → *implementing the supply chain management process*.
Solution: Some aspects of supplier management can and should be performed using tools. Tools should be used for preventive governance when you choose a supplier to build and maintain a database of suppliers and their → *assessed maturity of open source governance and compliance*. Other tools can be used for → *governance awareness self-certification by suppliers*. Lawyers can use tools that assist in → *designing supplier contracts with open source governance aspects in mind*. Tools should also be used in corrective governance and license compliance in supply chains. This includes open source code and license scanning tools that automate → *audits of the supplied software*. In bill of materials management tools should be used for → *tracking, documenting and updating bill of materials*, and to → *host a backup of the supplied FLOSS components*. Tools should be used to integrate supplier management processes with other processes and artifacts of this handbook in component approval, component reuse etc.

Table 3. Example best practice SUCHMA-BOMMAN-4. Use machine readable and standard format for BOM upon software supply

ID: SUCHMA-BOMMAN-4
Name: Use machine readable and standard format for BOM upon software supply
Actor: OSPO (Open Source Program Office), Roles responsible for supply chain management

Context: You have used the bill of materials and code scanning of the supplied code to → *identify open source components and metadata from the supply chain*. You have → *tracked, documented and updated BOM in a consistent and complete manner.*
Problem: How can you improve the performance of managing your BOMs?
Solution: Software supply chains are complex and cannot be handled manually. You need to → *use tools* to improve the performance of BOM management. Most importantly you need to establish a machine readable and standard format for BOMs. An example of such a format is called Software Package Data Exchange (SPDX). It enables the documentation and exchange of data and metadata for open source components and BOMs made of such components.

Note the "→"s in the presented best practices that refer to other interconnected patterns within the category of SCM best practices. Subsets of the proposed best practices form workflow templates that the case study company followed with major modifications. See Figure 1 for one of the workflow templates. For instance, SUCHMA-BOMMAN-4's solution refers to the OSGOV-SUCHMA-SCMPRO-5 pattern from Table 2 when it comes to the specifics of using SCM tools.

As the result of our evaluation case study, we observed that the proposed simplistic workflows only partially fulfilled the needs of the case study company. As an alternative, the software architect at the case study company designed a border process focused on supplier management and open source governance in general. The first draft of the process we observed in the early stage of the implementation is presented in Figure 2, while the final draft with major changes is in Figure 4.

Before discussing the specifics of the evaluation artifacts created in the course of the case study, we present the initial situation assessment in terms of SCM and open source governance at the case study company as follows.

4.1. Situation assessment

Confirming our sampling criteria for the case study, we found that the company had basic open source governance in place, especially focused on inbound (including component approval and reuse) and outbound (including an open source software license compliance process) governance. The company had a formalized FLOSS governance process in place, which was managed by a centralized team called *Technical Compliance Department*. This team was in close contact with the R&D department, company lawyers, procurement office, as well as production teams and developers.

At the time of the study, the department was going through a rebranding as part of an overall reorganization of the company after a change in the top management. We observed that this restructuring had a mixed effect on the implementation of our proposed best practices with the more local ones being given priority and the more large-scale changes, such as establishing a company-wide SCM policy, being put on hold. The department had existed for about 20 years dealing with various aspects of technical compliance related to software (not only open source software), dealing with open source and proprietary (commercial) software licenses, compliance process and automation, as well as other aspects of inbound and outbound governance.

The Technical Compliance Department created and maintained several centralized processes, such as the open source license clearance process for the components used by the company developers. However, we found that the topic of supplier management had been largely untouched, delegated to the procurement & IT departments and the production teams directly. This created significant disparities between the governance awareness and compliance across different parts of the company.

The teams that had employees with an advanced understanding of open source governance would reach out to the Technical Compliance Department experts for their guidance on the open source components received with the supplied code, while most of the other teams ignored the fact. This created some vulnerabilities, which were not assessed or managed by the people responsible for the overall open source governance. Given this initial situation, the team at the Technical Compliance Department was eager to collaborate with us and to implement our recommendations in order to kick start a shift towards a formalized supply chain management in the context of FLOSS governance.

In the initial situation assessment, we found a number of strengths when it came to open source governance at the company. Namely, the Technical Compliance Department did not assume that open source software used was correctly reported, tracked, or audited, instead they aimed at educating and managing the governance realistically. The core team at the department automated knowledge management covering various topics of FLOSS governance and related guidelines in company-wide consistent wikis.

They also had introduced a number of company-internal tools (wikis, license scanning tools, component management tools, etc.) that made open source governance less of a chore for the developers and production teams. Additionally, the key stakeholder employees were often asked for feedback on the existing open source compliance processes and their overhead. Finally, the open source governance team was in constant contact with external experts and organizations sharing knowledge and experience in corporate governance.

Alongside the strengths, we identified several weaknesses, which we hoped to address through the proposed SCM best practices. One weakness we found was that the Technical Compliance Department's process for open source compliance (as part of inbound governance) was not well enforced, which led to possible workarounds by the developers under time pressure or unaware of the inbound governance process. The team was trying to mitigate this by making compliance and governance easier and more integrated into the development processes, but it didn't always work as planned.

Another key challenge was that most code from third-party suppliers was not reviewed as thoroughly as the open source components added directly by the production teams, even though the risks of license non-compliance, for example, are equally harmful to the company. Moreover, the inbound governance processes did not apply for the third-party supplied code as the company relied only on contractual safeguards for potential open source compliance issues, which changed in the course of the case study. Finally, the responsibilities in the governance process were not always clear.

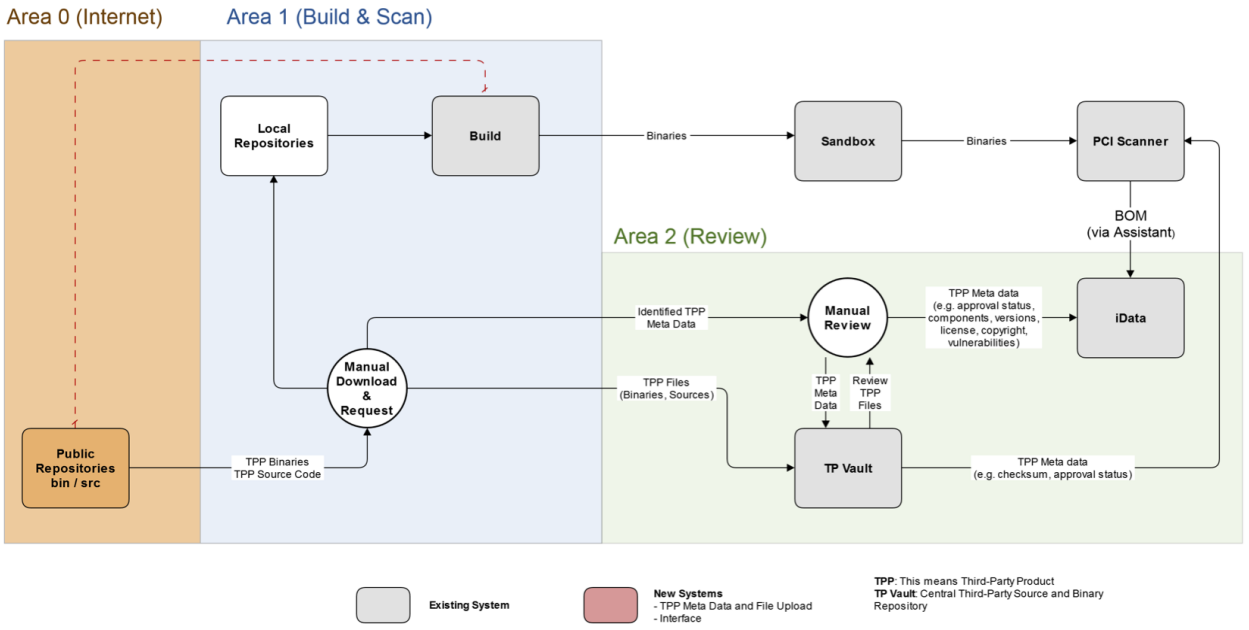

Figure 2. First draft artifact of the SCM process designed by the case study company

4.2. Best practice implementation and evaluation

The trigger to start the best practice implementation was the planned company acquisition which used a significant amount of open source software, but needed to be checked for compliance. Our partner team at the Technical Compliance Department aimed at establishing a supply chain management process that would also be used for this to-be-acquired company. During this early stage of our evaluation, we discovered that our case study partners decided to extend the expected use of the proposed best practices applying them in this manner. Our broader FLOSS governance handbook had a specific process for dealing with mergers and acquisitions, but this didn't match the implementation pattern undertaken by the case study company. This was the first instance where our pattern matching technique demonstrated a disparity between the proposed process and the actual one. While this did not manifest a *relevance* or *usefulness* issue, it demonstrated that the expected *applicability* was different in real-life.

One of the first steps by our partner team was selecting a pilot project and the team that would carry out the implementation of the handbook. After the introductory workshop, we met with the pilot project team that consisted of the technical top manager, a compliance manager, a compliance officer, and a procurement officer. During the guided implementation the technical top manager oversaw the work of the pilot project team and provided strategic input. The compliance manager and the compliance officer dealt with the operational side of the handbook section implementation, as well as its integration into the existing processes at the company. The procurement officer worked directly with the existing and future suppliers following the handbook best practices on ensuring the open source governance in the software

supply chains. This separation of tasks is in line with our recommendations, which demonstrates the *completeness* of our theory under evaluation.

In a follow-up workshop we identified the specific best practices for implementation, after which the pilot project team prioritized the proposed practices based on their needs, as well as on their estimated applicability in the scope of our case study. Given the abstract nature of our best practice, the evaluation demonstrated that some best practice descriptions lacked context and were not readily *understandable* and *applicable* at the company. To address this expected shortcoming, the pilot project team was in close contact with us seeking clarifications and guidance when needed. For instance, we discussed the pattern *OSGOV-SUCHMA-SCMPRO-5. Use tools to automate supplier management,* and provided the necessary details and guidance on choosing the right tooling.

We also discussed the best practices of *OSGOV-SUCHMA-PREGOV-1. Choose the right supplier* and *OSGOV-SUCHMA-PREGOV-1.1. Assess open source governance and compliance awareness and maturity*, which were implemented by the pilot project team with major modifications. The result was a supplier questionnaire with a focus on open source governance prepared by the procurement officer. Figure 3 demonstrates an excerpt from this Open Source Compliance Questionnaire for Suppliers / Licensors, which shows that at that stage minimum requirement to document the BOMs of the supplied products were PDFs, but the more preferred reporting formats were either a machine-readable (debian/copyright file) or in the best case the company-specific SPDX document. The requirements for the last two options were also implemented following our best practices, namely *OSGOV-SUCHMA-BOMMAN-2. Track, document and update BOM in a consistent and complete manner* and *OSGOV-SUCHMA-BOMMAN-4. Use machine readable and standard format for BOM upon software supply*, which shows the *significance* and the *relevance* of our findings. Moreover, these best practice patterns were employed without major adjustments, which shows the *correctness* and the *applicability* of the evaluated practices.

In the course of the pilot project, the questionnaire was sent to many of the recent suppliers of the company, but were optional having an educational objective first and foremost.

The implementation team planned to make such supplier BOM documentation and reporting mandatory over time. Some other best practices were postponed altogether, including *OSGOV-SUCHMA-PREGOV-1.2. Request supplier certification or self-certification*, which was deemed too demanding by the company. This observation during our evaluation hints at a previously implicit dimension of the proposed best practices - the FLOSS governance maturity level required for the implementation of given patterns.

7. Requirements to use Licensor's Products using Free and Open Source Software

a) Licensor shall provide Bill of Material in one of these suitable formats:

	Format	Specification
Minimum Requirement	PDF document	specs to be requested by COMPANY
Standard Requirement	Machine-readable `debian/copyright` file	https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Best/ recommended Requirement	COMPANY specific SPDX Format	

b) Licensor shall provide Source Code of Open Source Components:

Minimum Requirement	where this is required by Open Source License (e.g. GPL family of licenses)
Standard Requirement	all components where Source Code is available
Descriptions of the Location: (FTP, GIT, ...)	

Figure 3. Excerpt from the newly introduced Open Source Compliance Questionnaire for Suppliers / Licensors

Finally, demonstrating the *variability, structure, and comprehension* of the proposed set of SCM best practices, the pilot project implementation resulted in an overarching open source governance framework, which evolved over time (see an early draft in Figure 2 and the final version during the case study in Figure 4). Figure 4 keeps the following components from Figure 2. iData - a master data management system, used to manage the company's product catalog, which contained technical dependencies between different products and their third-party products (TPP) including open source software. PCI Scanner was a tool used to identify requested (known) TPPs and to report the scanning results feeding into the BOMs stored in the iData repository. TP Vault was a repository that contained the requested TPPs (sources and binaries). However, Figure 4 introduces new components such as open source license scanners (e.g. FOSSA) and TPP Fetcher - an internal tool collecting TPP metadata (licenses, copyrights, etc.) from different sources within the build environment. It fetched TPP files (source code and binaries) that belonged to a TPP via package managers. It uploaded TPP metadata and TPP files to the TPP Interface, which would then trigger the TPP review process.

6. Conclusions

To conclude, during the initial assessment, we found the following key processes in place before the case study:

- Open Source Component Approval and Compliance Process
- Open Source License Interpretation Process
- Open Source License-Use Case Pair Documentation Process
- Compliance Automation Process
- Component Reuse Process
- Hiring Process with Focus on Open Source Competencies.

Namely, given the ongoing restructuring and the revolving role of the Technical Compliance Department, the best practice pattern *OSGOV-SUCHMA-SCMPOL-1. Establish supply chain management policy* was not implemented, instead being replaced by a more operational SCM process. This, among others, helped us revise and improve the original best practices making them more transferable thanks to the real-life evaluation through a case study.

This evaluation case study has several limitations. First of all, conducting only one case study limits the generalizability of the evaluation. However, given the choice of the research method, we did not aim for broad generalizability, but rather focused on conducting an in-depth evaluation of the proposed theory in a contemporary and real-life setting. We achieved this as our recommendations were adjusted and implemented in a production level project of a major enterprise software company with 10,000 enterprise customers in over 70 countries.

In the course of the case study, the subject company shiten its focus towards supply chain management designing a new process presented in Figure 4, as well as other artifacts such as a supplier questionnaire to assess their FLOSS governance awareness and maturity presented in Figure 3. These were both directly recommended in the proposed best practices, which demonstrated the usefulness and the transferability of our original findings. At the same time, we found that some of our recommended best practices did not fit the context of the subject company, thus lacking applicability.

Another limitation is the funding received from the company for our consulting services in the scope of this project. To address this potential limitation, we agreed before the start of the project that the research won't be affected by the consulting activities conducted in the course of the study. Moreover, the company did not pay our research institution directly, but rather contributed to a public-private partnership fund, which was then allocated to different research partners within the Software Campus 2.0 project managed by the BMBF's (Federal Ministry of Education and Research).

Finally, to ensure the credibility of our study, we followed the widely accepted case study research methodology by Yin [36], which, among other things, recommended the definition of a case study protocol. In this protocol, we defined the specifics of the evaluation criteria, the interview questions and the data analysis techniques.

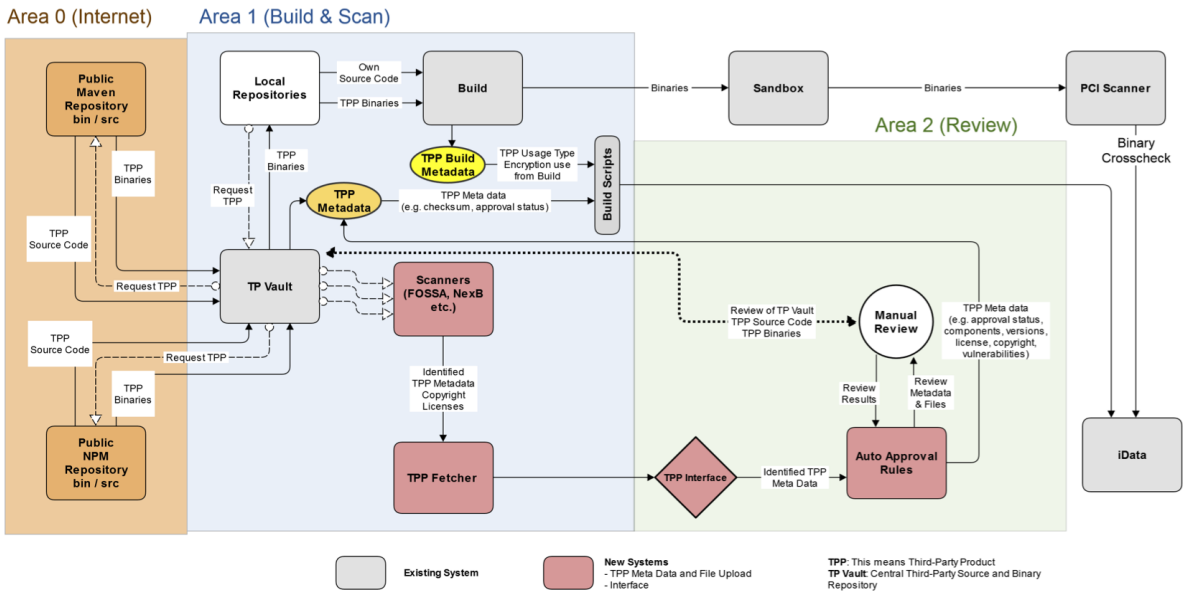

Figure 4. Final artifact of the SCM process designed by the case study company

7. Acknowledgments

This research was funded by BMBF's Software Campus 2.0 project (OSGOV, 01IS17045-17570). We would like to acknowledge Andreas Bauer, who was involved in and contributed to this case study. We also thank our case study partners for their collaboration.

8. References

[1] Alspaugh, T.A., Asuncion, H.U., & Scacchi, W. (2009). Analyzing Software Licenses In Open architecture software systems. In IEEE Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 54–57.

[2] Beck, C. T. (1993). Qualitative research: The evaluation of its credibility, fittingness, and auditability. Western Journal of Nursing Research, 15(2), 263–266.

[3] Bitsch, V. (2005). Qualitative research: A grounded theory example and evaluation criteria. Journal of Agribusiness, 23 (345-2016-15096).

[4] Blecken, A. & Hellingrath, B. (2008). Supply chain management software for humanitarian operations: review and assessment of current tools. Proceedings of the 5th ISCRAM, 342–351.

[5] Chen, W., Li, J., Ma, J., Conradi, R., Ji, J., & Liu, C. (2008). An empirical study on software development with open source components in the chinese software industry. Software Process: Improvement and Practice, 13(1), 89–100.

[6] Conlon, P. & Carew, P. (2005). A risk driven framework for open source information systems development. In the 1st International Conference on Open Source Systems. 200–203.

[7] Coughlan, S., Noda, T., & Tansho, T. (2013). A case study of the collaborative approaches to sustain open source business models. In Proceedings of the 9th International Symposium on Open Collaboration: ACM.

[8] Fendt, O., Jaeger, M., & Serrano, R. J. (2016). Industrial experience with open source software process management. In the Computer Software and Applications Conference, 40(2). IEEE, 180-185.

[9] Fink, A. (2003). Analysis of qualitative surveys. In: The survey handbook. SAGE Publications, 61–78.

[10] Gamma E., Helm R., Johnson R., Vlissides J. (1995). Design Patterns. Addison Wesley.

[11] Germonprez, M., Young, B., Mathiassen, L., Kendall, J.E., Kendall, K.E., Warner, B., & Cao, L. (2012). Risk mitigation in corporate participation with open source communities: protection and compliance in an open source supply chain. Risk, 12, 15–2012.

[12] Guba, E. G. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries. ECTJ, 29(2).

[13] Hannebauer, C. & Gruhn, V. (2019). An open source pattern language. In Transactions on Pattern Languages of Programming IV. Springer, 76–99.

[14] Harutyunyan, N. (2020). Managing Your Open Source Supply Chain-Why and How? IEEE Computer, 53(6), 77–81.

[15] Harutyunyan, N. (2019). Corporate Open Source Governance of Software Supply Chains. PhD Dissertation. Friedrich-Alexander-Universität Erlangen-Nürnberg, available from https://nbn-resolving.org/urn:nbn:de:bvb:29-opus4-122727

[16] Harutyunyan, N., & Riehle, D. (2021, January). Getting Started with Corporate Open Source Governance: A Case Study Evaluation of Industry Best Practices. In Proceedings of the 54th Hawaii International Conference on System Sciences.

[17] Harutyunyan, N., & Riehle, D. (2020, September). Industry best practices for component approval in FLOSS Governance. In Proceedings of the 25th European Conference on Pattern Languages of Programs. ACM.

[18] Harutyunyan, N., & Riehle, D. (2020, January). Industry best practices for corporate open sourcing. In Proceedings of the 53nd Hawaii International Conference on System Sciences.

[19] Harutyunyan, N., & Riehle, D. (2019, July). Industry best practices for FLOSS governance and component reuse. In Proceedings of the 24th European Conference on Pattern Languages of Programs. ACM.

[20] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS quarterly, 75-105.

[21] Jansen, H. (2010). The logic of qualitative survey research and its position in the field of social research methods. In Forum Qualitative Sozialforschung/Forum: Qualitative Social Research, 11 (2).

[22] Kemp, R. (2010). Open source software (OSS) governance in the organisation. Computer Law & Security Review, 26(3), 309-316.

[23] Kemp, R. (2009). Towards free/libre open source software governance in the organization. IFOSS L. Rev., 1.

[24] Koltun, P. (2011). Free And Open Source Software Compliance: Operational Perspective. IFOSS L. Rev., 3.

[25] Krefting, L. (1991). Rigor in qualitative research: the assessment of trustworthiness. The American Journal of Occupational Therapy, 45(3), 214–222.

[26] Lincoln, Y. S., & Guba, E. G. (1985). Establishing Trustworthiness. Naturalistic Inquiry, 289.

[27] Lovejoy, J., Odence, P., & Lamons, S. (2013). Advancing the software package data exchange: An update on spdx. International Free and Open Source Software Law Review, 5(2), 145–152.

[28] Morse, J. M., Barrett, M., Mayan, M., Olson, K., & Spiers, J. (2002). Verification strategies for establishing reliability and validity in qualitative research. International Journal of Qualitative Methods, 1(2), 13–22.

[29] Riehle, D., Harutyunyan, N., & Barcomb, A. (2021). Pattern Discovery and Validation Using Scientific Research Methods. Transactions on Pattern Languages of Programming. *to appear*. Preprint available at: https://dirkriehle.com/wp-content/uploads/2020/03/cs-fau-tr-2020-01.pdf

[30] Ruffin, C., & Ebert, C. (2004). Using open source software in product development: A primer. IEEE Software, 21(1), 82-86.

[31] Russell, C. K. & Gregory, D. M. (2003). Evaluation of qualitative research studies. Evidence-Based Nursing, 6(2), 36–40.

[32] Shenton, A. K. (2004). Strategies for ensuring trustworthiness in qualitative research projects. Education for Information, 22(2), 63–75.

[33] Stewart, K., Odence, P., & Rockett, E. (2010). Software package data exchange specification. IFOSS L. Rev., 2.

[34] Trochim, W. M. (1989). Outcome pattern matching and program theory. Evaluation and program planning, 12(4), 355–366.

[35] West, J. et al. (2007). The economic realities of open standards: Black, white and many shades of gray. Standards and Public Policy, 87.

[36] Yin R. K (2013). Case study research: Design and methods. Sage publications.