Getting Started with Corporate Open Source Governance: A Case Study Evaluation of Industry 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

Open source software usage in companies is on the rise, often resulting in lower development costs, higher quality, and quick availability of code. However, using open source software in products comes with legal, business, and technical risks. Experienced companies prevent and address these risks through corporate open source governance. In our previous work, we studied how top-tier companies got started with corporate open source governance. We proposed a set of industry best practices on the topic, using the practical format of interconnected context-problem-solution patterns. In this study, we put the proposed state-of-the-art practices to the test by evaluating their real-life application in a case study at a Germany-based multibillion-dollar corporation with products in four distinct industries and more than 17000 employees worldwide. In the course of two and a half years, we conducted 35 semi-structured employee interviews and workshops in five divisions of the company to assess the initial situation of open source governance, the process of getting started with governance following our recommendations, and the outcomes. In this paper, we report the results of this longitudinal case study by presenting the artifacts created while getting started with open source governance, as well as the transferability evaluation of the proposed best practices, both individually and collectively.

1. Introduction

Today, virtually all software companies use free/libre and open source software (FLOSS) in their products. FLOSS infrastructure components include cutting-edge operating systems, web servers, database systems, machine learning frameworks, and many more. Wide-spread commercial adoption of open source software brings an estimated value of €114 billion per year directly and up to €399 billion per year overall to the European economy, according to a recent European Commission report.[1] However, while some companies maximize their benefits from using open source software by managing the risks related to the mishandling of FLOSS licenses, copyright, and export restrictions [34, 37], other companies remain vulnerable due to being unaware or ignorant of the risks that come with open source software.

While open source software and open source development have been extensively researched [8, 25], the topic of corporate open source governance, in particular, has been studied to a lesser extent. To address this industry-relevant topic, in our previous work, we studied different aspects of FLOSS governance in companies, such as the potential legal risks of open source use in products [35], and industry requirements for governance tools [23].

In this paper, we focus on corporate FLOSS governance, which consists of industry best practices and processes for dealing with open source use in companies [23]. To collect and publish such state-of-the-art practices from the industry, we performed qualitative data analysis [24, 11] of the 20 primary materials[2] and 21 expert interviews [17, 18, 19, 21]. We derived our findings from the data gathered from a diverse set of companies with an advanced understanding of corporate open source governance, such as Google, Intel, Qualcomm, Microsoft, BMW, and others.

[1] The economic and social impact of software & services on competitiveness and innovation (SMART 2015/0015). European Commission. Luxembourg: Publications Office of the European Union, 2017. Available online from https://op.europa.eu/en/publication-detail/-/publication/480eff53-0495-11e7-8a35-01aa75ed71a1

[2] An example of a primary material in our analysis: Google's Internal Guidelines for Open Source Use Governance - https://opensource.google/docs/using/

URI: https://hdl.handle.net/10125/71376
978-0-9981331-4-0

We found that the first challenge for the companies unfamiliar with open source governance is the question of getting started, identifying the requirements for and the structure of a FLOSS governance policy [18]. In a previous paper published at the 15th International Symposium on Open Collaboration [18], we addressed how companies can get started with governing their use of open source software. We identified state-of-the-art practices for corporate open source governance in the following categories: product analysis, transition policy, transition organization, IP-at-risk analysis, communication, and capabilities.

While publishing sets of best practices in different research outlets [18, 19, 21], we always emphasized the importance of the practical applicability of our research. We cast our findings in an actionable format of best practice patterns [13] and processes. By best practices in this context we mean the state- of-the-art practices in the companies with expertise in FLOSS governance we studied. In this paper, we use the terms "best practice" and "state-of-the-art practice" interchangeably. Table 1 illustrates a previously unpublished example of our industry best practices for getting started with open source governance. To evaluate our previously proposed theory for getting started with FLOSS governance [18], we studied its trustworthiness to ascertain the quality of our exploratory research and findings. We followed Guba [15, 30] identifying the criteria for trustworthiness of qualitative studies, as our exploratory study was conducted using a qualitative survey [24]. Thus, we have to consider 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).

Credibility, dependability, and confirmability were dealt in our original study, based on our research design and process, and resulted in our initial theory. However, we were not able to evaluate the transferability of our theory in the same way during the original study. Transferability is the degree to which the findings of our study hold validity in other contexts. To evaluate the transferability, we had to look at how our theory can be generalized and applied at companies with no or little corporate open source governance in place. Such an evaluation strategy has been recommended by researchers studying the trustworthiness in qualitative research projects [33, 39].

In this study, we focused mainly on evaluating the transferability of our theory. As the evaluation focus was on assessing our initial theory's external validity, we asked the following overarching research question:

RQ: How transferable are the proposed industry best practices for getting started with corporate open source governance in the context of companies with no or little governance in place?

To answer the research question, we conducted a longitudinal case study following Yin [43] at a multinational company based in Germany with software-intensive products in several industries, such as aerospace, internet of things, metering, and electronic assemblies. The company had no to little (depending on the division) experience with open source governance, while already using FLOSS components in different parts of the company. Working closely with the partner company over the course of two and a half years, we first assessed the initial situation of FLOSS use and ad-hoc governance or lack thereof. We then introduced our proposed best practices and guided their implementation. Having developed a case study protocol following Yin [43], we did not directly implement or interfere with the application of our theory, but rather guided different actors in the company who were responsible for getting started with open source governance across the company. Therefore, we did not directly influence the study subject, but observed it throughout the whole process.

In the course of our case study, we worked with five divisions of the company, interviewing 35 employees based in different parts of Germany, as well as in China, Mexico, and Poland. We interviewed stakeholders with different levels of seniority from software developers to IT officers to C-level managers. We also organized several workshops discussing the initial situation of FLOSS governance, the challenges faced by the company, as well as the possible paths to implement the proposed best practices and process for the transition to open source governance.

In the course of the case study, the overarching research question was operationalized using specific quality criteria. We chose several criteria by looking at academic research from various disciplines, where qualitative theories were evaluated. As a result, we used the following criteria to assess the transferability of our proposed best practices, such as completeness, variability, comprehension, understandability, significance, and more.

Following Yin [43], we used pattern matching to compare and contrast the proposed industry best practices with their actual application in the case study company. The result of our study was a critical review of the original theory as a whole, as well as specific practices using the above-mentioned evaluation criteria.

Going beyond the description of the case study results, we also discuss how the proposed practices were adjusted to better support software companies in getting started with open source governance.

This paper is structured as follows. In section 2 of this paper, we present a review of related work and literature. In section 3, we present the research method, including the case study protocol. In section 4, we present the results of the evaluation case study. In section 5, we discuss research limitations, including threats to internal validity and external validity, followed by section 6, which concludes the paper.

2. Related work

Traditionally, researchers studied open source governance in the context of managing open source communities built around specific open source projects, covering aspects of creating and running such projects [29, 31, 32, 40], as well as managing the code contributors [2]. In contrast, our study focused on a different type of FLOSS governance, namely corporate open source governance, which takes the perspective of companies using existing open source code in their products. 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 [23].

Related literature covers some of the risks caused by the unmanaged use of open source software in products, such as mishandling open source licenses [10, 37], or missing on critical updates and maintenance provided by open source projects [1, 6, 7]. These challenges can be managed through open source governance, which addresses, among other issues, license compliance management [14], and related tooling [26]. In our initial situation assessment at the case study company, we identified license-related issues and risks associated with the unauthorized use of open source software by developers, often in critical features of company products.

In parallel to better understanding and managing the risks of using open source software, companies increasingly realize the benefits of such use in their products, going beyond the commonplace use of FLOSS development tools [12, 28], such as open source software components being quickly available, of high quality, and low cost, as well as the fact that open source components and standards are widely accepted, thoroughly tested, highly secure and well maintained by professional communities. In the course of this study, we found that our case study company got to experience the above- mentioned benefits of using open source software with little risk thanks to the implementation of the proposed industry best practices for open source governance.

In our previous research, we proposed a set of industry best practices for corporate open source governance based on a qualitative survey of industry experts and primary materials [11, 24]. We covered the following key aspects of FLOSS governance in companies: getting started with open source governance [18], inbound governance [19, 21], supplier management [17], outbound governance, and general governance.

In the course of this case study, we evaluated the industry best practices for getting started with open source governance in companies [18]. We identified state-of-the-art practices in the following subcategories:

- Product Analysis - 8 best practices
- Transition Organization - 8 best practices
- Transition Policy - 3 best practices
- IP-at-Risk Analysis - 9 best practices
- Communication and Capabilities - 5 best practices.

Anticipating the need for real-life evaluation of our theory, we ensured the practical applicability of the proposed practices by casting them as patterns, which can be easily implemented by companies. Patterns and pattern languages have been used in the past to present different concepts of open source use, development, and governance. Among others, Hannebauer and Gruhn [16] presented an overview of the current state of research on OSS patterns, including 40 published patterns, their key topics, and relationships between them. In our previous work beyond open source governance, we also used the same format of theory presentation in publications on corporate open sourcing [20] user and experience design in software product lines [22]. We formalized this method in a paper that can serve as a guide for other researchers interested in presenting their theories in a similarly applicable manner [36].

Table 1. Example best practice OSGOV-IPRISK.1.2. Use standard license interpretation

ID: OSGOV-IPRISK.1.2.
Name: Use standard license interpretation
Actor: Developers
Context: Software developers need legal advice on open source licenses before using given components in company's products in order to ensure legally compliant use of open source software. Your company's lawyer → *developed standard license interpretation* and shared them with developers across the company.
Problem: Who should use the standard license interpretations and how?
Solution: Developers must use and follow company's standard license interpretations when adding open source components into company's products.

An example practice from the IP-at-Risk Analysis category covered open source license compliance, namely recommending the use of standard license interpretation across the company establishing open source governance. Table 1 presents the proposed practice OSGOV- IPRISK.1.2. Use standard license interpretation, which we evaluated in this case study.

Developers should be introduced to the standard license interpretation of the major licenses (GPLv2, GPLv3, LGPL, AGPL, MIT, BSD, Modified BSD, etc.) during the → *provided employee training*. When using open source code (eitherdirectly from FLOSS communities or as part of supplied software), developers should consider the license interpretation in a given business case (e.g. using GPLv3 can be acceptable in one use case and not acceptable in another) generally outlined in company's → *established FLOSS governance policy for the transition period*. For the special cases that are not described in the governance policy, developers must consult the transition board or the transition manager, who then review and document their case by case decisions as part of the → *implemented transition process*. The transition manager must use this documentation to → *create license/use case pairs*.

In this study, we evaluated the proposed best practices in the production context of the case study company, both individually and collectively. Continuing the example of the best practice OSGOV-IPRISK.1.2. Use standard license interpretation, we found that the employees responsible for establishing open source governance at the aerospace division of the case study company decided to establish a centralized database for open source licenses and used in the division coupled with their legal and technical interpretations, as well as resulting requirements for the production teams and software developers, in particular.

3. Research method

Once we specified the research question for our theory evaluation, we designed a research approach to answer this question. The clear scope of our evaluation was on the transferability of our proposed best practices for getting started with FLOSS governance in companies. We were able to evaluate the internal validity of our theory during theory building including its credibility, dependability, and confirmability, which we presented together with the published set of best practices [18].

Anticipating the need for the practical implementation and evaluation of the proposed practices, during theory building we presented thorough descriptions of the research context and our underlying assumptions [42]. For each of our proposed industry best practices we presented the context (one of the components of a best practice pattern we used to present our theory) in which we described in detail under which conditions and assumptions a given best practice would apply.

In this study, we aimed at addressing the external validity of our theory, in particular the transferability of the proposed best practices to software companies with little or no understanding of FLOSS governance (as opposed to the expert companies we derived the practices from in the first place).

Despite the complexity of the practical evaluation of a set of practices and processes in a company, we aimed at finding a company willing to implement our recommendations in production context, while enabling us to observe and evaluate how generalizable our findings are. We searched for companies with little or no open source governance processes in place, as such a company would be most interested in the best practices for getting started with open source governance.

We chose case study research as our research method, both to find the appropriate subject company and to design the study. As Yin [43] suggests case study research (in comparison to other strategies such as experiments, surveys, archival analysis, or history) is a fitting research strategy for situations that:

- ask research questions in the form of how and why
- do not require control over behavioral events
- focus on contemporary and complex phenomena that can be studied in real-life context.

Our research question was a how question focused on the transferability of the proposed theory in the context of companies with no or little governance in place. Our evaluation did not require control over behavioral events, nor was such a controlled study possible for a theory so complex and multi- layered (in terms of having an organization-wide impact and hierarchies of stakeholders), which could not realistically be confined to a controllable environment. Finally, our theory focused on the contemporary phenomenon of corporate open source governance, as the topic has been emerging only recently, as demonstrated in the related work section.

We used the case study research method to test a published theory, which corresponded to one of the research purposes a case study could have, as suggested by case study methodology scholars [5, 9, 43]. To guide our study and to ensure its rigor, we developed a case study protocol ahead of the study and followed it throughout.

Our case study was both descriptive and explanatory. It was descriptive in resulting in detailed reports of what the initial state of open source governance at the studied companies was, as well as how companies followed and implemented the proposed

industry best practices from the proposed theory. It was explanatory in presenting the reasons why certain parts of the theory were more or less complete, understandable, applicable, useful, etc., which resulted from analyzing the proposed and the actual implementation patterns of corporate open source governance at the studied companies.

Another characteristic of our research method was it being a longitudinal case study with a holistic design. We studied the implementation and use of the proposed theory at the pilot project teams in five divisions of the case study company. The employees of these teams were our main source of data. Finally, we need to report that the case study company partially funded our research through collaboration with our university, which, however, did not affect the study.

For the evaluation of the proposed best practices, both individually and collectively, we chose several criteria by looking at academic research from various disciplines, where qualitative theories were evaluated. Namely, the applicability, relevance, understandability, and usefulness of a theory could be used to critically appraise the transferability of qualitative research [3, 38]. Bitsch [4] added another evaluation criterion

– the comprehension of the theory. Other evaluation criteria included the structure, completeness, and variability of qualitative theories [27, 33]. As a result, we used the following evaluation criteria in our case study, which also defined the interview questions asked to the relevant stakeholders in the five divisions of the case study company: completeness, variability, structure, comprehension, understandability, applicability, relevance, significance, and usefulness.

3.1. Case study methodology

Our research question could be best answered by studying the concept of corporate open source governance in its real-life context, which dictated our choice of methodology. We followed the case study research methodology by Yin [43]. We aimed for a practice-based theory with an in-depth analysis and rich insights that can be applied by other companies looking into getting started with open source governance.

Following Yin's case study methodology, we identified the research question, chose relevant research methods, identified case study design, developed case study protocol, selected cases from a theoretical sample, iteratively collected data, refined the study design, analyzed data using appropriate tools, and derived and presented the results.

When designing our research strategy for theory evaluation, we looked at potential industry partners in the professional network of our research group to find companies with no or little open source governance in place that would also be interested in cooperating with us on the topic by allowing a guided implementation of our best practices in some of their production projects. We would like to highlight that our intention was to guide the implementation of our theory, and not to conduct the implementation on our own. We observed how employees at the case study company were using parts of our handbook (the practical artifact that included the proposed best practices), but we did not directly influence them. This ensured a less biased theory evaluation and was in line with our case study research method by Yin [43]. This explicit choice of research design also meant that we could not follow another research method – action research for which we would have to be directly involved in the implementation, rather than being only observers.

As a result of our sampling, we chose a Germany-based multinational company that had no governance processes or practices in place overall (and only little informal governance in place in some divisions), which would enable us to implement the industry best practices for getting started with FLOSS governance. We describe the full profile of the company and its structure in the following subsection.

3.2. Case context and data sources

The company we chose for this study was a large Germany- based company operating internationally in four software- intensive industries, and using open source software in its products (e.g. aerospace systems, IoT devices). We anonymized the company name as per their request. We worked with five divisions of the company, each focused on one industry with the exception of one, which provided internal IT services. The divisions mapped with their industries were:

- Division 1 - Aerospace
- Division 2 - Internet of Things
- Division 3 - Metering
- Division 4 - Electronic Assemblies
- Division 5 - Information Technology (internal)

Over the course of two and a half years in October 2016 – May 2019, we extensively studied open source use and governance across the case study company. Our first and major focus was Division 1 that served as a pilot project in the evaluation case study. We conducted 12 two-hour interviews with managers, software developers and other stakeholders at Division 1. Using Division 1 as a benchmark, we went on to assess open source use and governance situations in other divisions, namely in Divisions 2, 3, 4, and 5. In each of Divisions 2, 3, and 4, we interviewed 7 employees.

Note that Divisions 5 was an internal IT-service provider with no external customers, therefore it was the smallest of all the studied divisions. It collaborated with

the IT departments in the other divisions, while providing centralized support and guidance. As we had interviewed employees with IT roles in Divisions 1, 2, 3, and 4, we decided to interview only two employees at Divisions 5 – the Legal Counsel and the Division CTO.

In the course of the case study, we conducted 35 semi- structured interviews with the stakeholders responsible for the implementation of the governance handbook at the five divisions of the case study company. In addition to interviews, we also collected feedback from employee stakeholders during workshops, reviewed the documentation and artifacts created during handbook implementation, as well as further notes and communication records.

We analyzed the data from our evaluation interviews, as well as the evidence collected through direct observation, document and artifact reviews, in order to identify how the implementation of our theory helped the case study company establish their corporate open source governance in comparison to their initial situation (of no or ad-hoc governance). We described the changes, presented the created artifacts, discussed the successful and failed experiences for different aspects of FLOSS governance. A key technique we employed in theory evaluation was called pattern matching [41, 43], which allowed us to compare the proposed open source governance practices from our theory with the patterns of their actual implementation at the case study company (across different divisions).

As a result of our theory evaluation, we demonstrated how our theory developed based on the expert knowledge at companies with an advanced understanding of FLOSS governance can be transferred to companies with no or limited understanding of open source governance. We also reported what the challenges to transferability could be resulting from the analysis of the pattern matching on different parts of our theory.

In the results section, we summarized our initial situation assessment for the case study company, followed by the evaluation of the proposed theory.

4. Results

As a result of this longitudinal case study, we evaluated a set of proposed industry best practices for getting started with corporate open source governance by applying them in the real- life context of the case study company. As a result of this study, we assessed which practices were applicable right away, which ones had to be adjusted to the company context, and which ones did not meet the company requirements.

4.1. Situation Assessment

As a result of the initial situation assessment, we confirmed our sampling criteria for the case study company. We found that the company and its divisions had no open source governance in place. Some informal governance existed as a way to address key issues of open source use, such as informal processes of clarifying open source license compliance when using open source components or libraries in some teams, but there was no centralized or formalized governance in place. Some employees took on the informal role of open source program office or compliance officers across the company providing support to their colleagues in their teams, divisions, and beyond.

In our initial situation assessment, we found that the company extensively used open source components in its products across all the divisions we studied. Some divisions also contributed back to open source communities (though this was rare). Both open source use and contribution are beneficial, when they are properly governed and regulated. However, the initial situation analysis indicated that FLOSS use was not properly governed or regulated. This unregulated FLOSS use and contribution carried significant threats to the company, including financial risks caused by non-compliance to open source licenses and other risks.

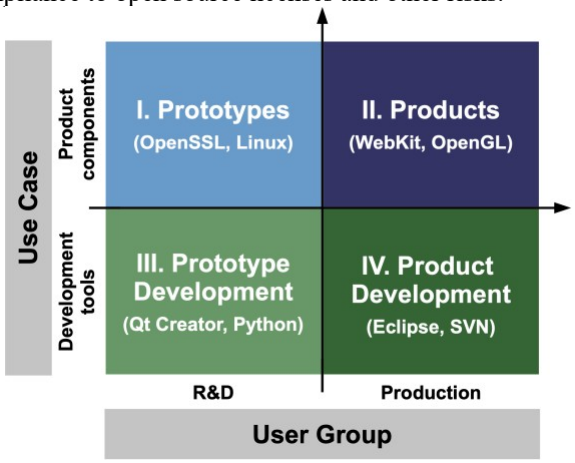

Figure 1. Use of Open Source Software in Division 1

We chose Division 1 for our pilot evaluation, because one of its projects struggled with several issues related to the use of open source components. After this project, the division decided to move towards more software intensive markets, thus anticipating open source use becoming a recurring practice, which needed to be regulated and defined by open source governance processes. In the first phase of our project, we assessed the initial situation of open source governance at Division 1 to identify the governance needs of the division and by extension some of the needs of the case study company. Additionally, starting with a pilot project was prescribed in one of the proposed best practices we were evaluating, namely OSGOV-

TRAORG-4. Start small, then replicate - define the scope of the transition process.

We found open source use both in R&D and in production. Division 1 used OpenSSL and Linux in prototypes, as well as WebKit and OpenGL in products. Figure 1 illustrates some of the open source use in the case study company based on the initial situation assessment in the aerospace division. For example, some open source components such as OpenGL were used in a civil aviation product. Other open source software used at Division 1 included Qt Creator[3] and Yocto[4].

One weakness we identified in using open source for R&D was the lack of structured processes for open source knowledge and competence transfer to other teams in Divisions 1. This resulted in the rough transition from R&D to production due to the limited open source use in production (while R&D prototypes were mainly built using open source components), as well as the initially little attention to open source licenses in prototyping. Our proposed best practices for getting started with open source governance later addressed these issues by successfully guiding the case study company towards a structured governance policy and a derived governance process, namely captured in the best practice OSGOV- TRAPOL-1. Establish FLOSS governance policy for the transition period.

Another noteworthy finding was that the newest use of open source software at Division 1 was in products that were becoming more software-intensive and less-specialized over time. Previously the software components were very specialized for the aerospace industry, which meant that product development teams couldn't find suitable open source components to use.

The other division also used open source software but differed from Division 1 in certain ways.

Division 2 (recently separated from Division 4 and still sharing some business functions, e.g. IT) had an extensive and critical use of open source components in products (also GPL- licensed software with copyleft effect) and in development. Some developers at Division 2 even contributed to open source communities, though rarely and using their private accounts to avoid restrictive company polcies.

Division 3 was similar to Division 2, the key difference being no known use of GPL-licensed software with copyleft effect. The main FLOSS components used in products were certain Java libraries.

Division 4 was more conservative with its open source use (mainly due to the hardware-intensive products with software components with limited features). The division did not use code under copyleft licenses, nor did it contribute to open source projects, unlike previous divisions. Notable FLOSS components used included open source UI components.

Division 5 traditionally used little open source software, instead often procuring third-party software components and systems (given its responsibility for internal IT infrastructure and related security and maintenance). At the same time, internal users demanded more FLOSS components and tools from the division, which resulted in the division providing significant centralized support in FLOSS governance (especially legal support) to other parts of the case study company. Having centralized legal support in the framework of company-wide open source governance matched a recommendation from our theory, namely the practice OSGOV-IPRISK-1.1. Develop standard license interpretation.

4.2. Individual evaluation

After assessing the initial situation of open source governance, we organized a workshop with the pilot project team at Division 1 chosen for the theory evaluation due to its accessibility and urgent need of open source governance practices. The latter was based on the division's recent experience of struggling with the lacking open source governance, as one of the customers had requested a mandatory use of certain open source components (for compatibility reasons). During this workshop we presented the proposed best practices to the stakeholder employees. We highlighted the possible need to adjust and modify the proposed practices and workflows or to create new ones that would fit the company- specific processes and guidelines.

The two software developers working on a Division 1 product and tasked with the implementation of the getting started section best practices started by reading the section and asking any questions they would have to us. For example, one of the questions that was raised concerning best practices OSGOV-GETSTA-PROANA-3.1. Run open source use analysis in products and OSGOV-GETSTA-PROANA-3.2. Document current open source use was about the specific metadata of the used open source components that needed to be documented. Before following the handbook best practices in running open source use analysis in products and documenting the identified open source components in use, the pilot project team wanted to clarify and document the specific metadata for each open source component. Their initial suggestion after reading the handbook was to use the following metadata: license name and version, use case (internal

[3] Qt Project – https://www.qt.io
[4] Yocto Project – https://www.yoctoproject.org/

tool, customer application software, delivered operating system), and restrictions (modifiability, source code publication).

This question indicated to us that this part of our theory was not detailed enough, therefore lacking comprehension and applicability, which corresponded to one of the theory evaluation criteria we had outlined in the case study protocol. To address this, we presented further metadata they could consider based on our theory, including component ID and name, component location, product / project ID and version, multiple licenses (y/n), copyright holder(s), and linkage type to the rest of the (software) product (e.g. dynamic or static).

The pilot project team added the above-mentioned metadata to their initial suggestion of identifying the use case and the usage restrictions of an open source component. They then requested the defined metadata from the developers involved in the pilot project, following the handbook best practice OSGOV-GETSTA-PROANA-1.1. Use one mandatory survey for initial assessment. Following the best practices OSGOV- GETSTA-PROANA-1. Use a combination of methods for product analysis and OSGOV-GETSTA-PROANA-1.2. Establish a process of continuous reporting and assessment, the pilot project team went on to analyze more of the used open source components by scanning several products and starting the establishment of a process of continuous reporting and assessment for future open source component additions.

This resulted in the first automated scan at case study company using an open source tool for FLOSS governance, called FOSSology[5] following the proposed best practice OSGOV-GETSTA-PROANA-1.3. Select and use governance tools for automation. The tools were chosen temporarily for the getting started process, as it did not require a lengthy procurement process necessary for the proprietary tooling alternative. However, the pilot project team was explicit that further tool comparisons would have to be performed before choosing the right long-term tooling of open source governance and compliance used across the company. Running an initial FOSSology scan was aimed at identifying the used but undocumented open source components in the current products at Division 1. As a result, the first implementation artifact was created – a FOSSology report with the identified open source components, their licenses, and other metadata. One of the employees (a manager at the R&D department) tasked with implementing the getting started practices created this artifact, analyzed the results, and started the manual review of the identified open source components in the existing product under review. The company-sensitive data has been anonymized. Some of the identified components and their licenses were masked. Figure 2 illustrates an excerpt from the artifact presenting some of the open source licenses used at Division 1.

FOSSology

OSS Component Clearing Report [Excerpt] at Division A.1, Company A		
Clearing Information	**Department**	FOSSology Generation
	Prepared by	Employee X (employee_x)
	Reviewed by (opt.)	NA
	Report release date	2018/12/18
Component Information	**Community**	NA
	Component	NA
	Version	NA
	Component hash (SHA-1)	D2D346D1B90E4E1E0F7DB22CD92B6649DA7EF1C2
	Release date	NA
	Main license(s)	Main License(s) Not selected.
	Other license(s)	License(s) Not Identified.
	Fossology Upload/Package Link	http://nas02fra:8081/repo/?mod=showjobs&upload=5
	SW360 Portal Link	NA
	Result of License Scan	0BSD, AFL-2.0, AFL-2.1, [redacted], AMD, ATT, Apache, Apache-1.0, Apache-2.0, Artistic-1.0, Artistic-1.0-Perl, Artistic-2.0, Autoconf-exception, BSD, BSD-2-Clause, BSD-2-Clause-FreeBSD, BSD-2-Clause-NetBSD, BSD-3-Clause, BSD-4-Clause, BSD-4-Clause-UC, BSD-possibility, BSD-style, BSL-1.0, Bison-exception, Bison-exception-2.2, CC-BY-NC-SA-3.0, CC-BY-ND-2.0, CC-BY-SA, CC-BY-SA-3.0, CC0-1.0, CMU, CNRI-Python, ClArtistic, Cryptogams, DOC, Dual-license, FSF, FTL, Freeware, [redacted]

Figure 2. Evaluation Artifact – a FOSSology report with the identified open source licenses at Division 1

4.3. Collective evaluation

The previous subsection described the implementation and the individual evaluation of several best practices for getting started with open source governance. This subsection presents the collective evaluation of the proposed best practices for open source governance, using the following dimensions or evaluation criteria.

Completeness was assessed for the getting started practices as a whole, as it evaluated whether the section had an adequate beginning, middle, and end, as well as whether it lacked any practices the case study company needed when applying the handbook. The employees tasked with the implementation reported in follow-up

[5] FLOSS governance and compliance tool FOSSology – https://www.fossology.org/

interviews that the handbook had an adequate level of completeness without any significant gaps or unanswered questions they encountered. However, during our direct observation, we noted that the pilot project faced some completeness related issues, which mainly related to Division 1 specific processes. For example, the development process at the division provided checklists for software developers to fulfill before moving into the next cycle of the development process. We did not take into account this specific need of the case study company (as our theory was developed based on industry best practices at expert companies who did not use such checklists). To complete this gap, the R&D developer tasked with the handbook adoption planned to add some practices to the getting started section before the company- wide roll-out.

Variability was also assessed for the handbook as a whole, as it evaluated whether the section had a balanced mixture of concepts for getting started with corporate open source governance and not overly focused on a single concept. During the case study, we observed that the balanced design of the proposed theory translated into an equal coverage of different getting started concepts, such as transition management, product analysis, and IP-at-risk analysis. This observation was also confirmed by the employees implementing the handbook, who highlighted that no concept was singled out and presented in more detail than others.

Structure was assessed for both the handbook as a while and for the individual industry best practices, as we evaluated how well-structured they were. For the section as a whole, we evaluated whether its different parts were structured in a logical and interconnected manner. The pilot project employees who were implementing the handbook appreciated the interconnecting links between individual best practices within the getting started topic, as such links created workflows that could be made into company processes and were already ingrained into the theory, therefore, making it easier to apply at the company. Using such links between the practices for the section, the R&D developer created a structured overview of the getting started practices and processes. As to the structure of the individual practices, all of the employees involved in the evaluation study noted the value of using the structured presentation format for the industry best practices from our theory – the Context-Problem-Solution pattern format that made the practices more digestible.

Comprehension evaluated how well the theory answered the problems companies with little to no governance would have, as well as whether the proposed best practices went into enough detail on their respective issues. We found that some of the workflows made of several best practices were confusing to the users of the handbook in the pilot project team. Moreover, we identified that some of the workflows that were giving an overview of the getting started handbook did not comprehensively capture all the interlinked best practices in the section. To address this (together with the above-mentioned issue related to the structure of the section) the case study company put together a comprehensive overview of the handbook to be used in the company-wide implementation after the pilot project.

Understandability focused on assessing both the intentions and the specifics of the proposed handbook. We found that the pilot project employees had to read the section carefully, attentively, and completely to ensure the full understanding. This required a significant amount of time (in average two months per employee) given that implementing the governance handbook was not a full-time task for the pilot project employees. The pilot project team recognized that the users of the handbook (e.g. developers, middle managers) would not read the getting started handbook in full, which could potentially lead to understandability challenges. To prevent such issues, the pilot project team set out to integrate the getting started handbook into the existing software development process at the company. The pilot project team planned to create employee training (for new hires and old employees), e- books, and other educational materials covering the highlights from the handbook in an easily digestible and understandable way.

Applicability helped evaluate how well our theory could be applied to a company with a different context from that at the expert companies involved in theory building. We evaluated how generalizable the getting started handbook was, as well as how much the evaluated best practices needed to be adjusted to become applicable at the case study company. Evaluating the section as a whole, we found that the biggest challenge for the applicability was the lack of a customized process. By design, our theory presented only general industry best practices on the topic, not customizing them for a ready implementation at one specific company. As a solution, the pilot project defined a company-internal guideline.

Relevance assessed how important our proposed practices were to the case study company in terms of addressing the company's needs of getting started with the corporate open source governance. We found that the employees in the middle management of Division 1 clearly recognized their needs for getting started with governance. Further confirming the relevance of the proposed handbook, we observed that company lawyers, developers, and technical managers found that the handbook section answered their questions around open source governance, clarifying the key concepts and

providing actionable advice of dealing with challenges of getting started with governance. Case study company employees also referred to the potential risks of the ungoverned open source use (also captured and presented in the initial situation assessment earlier) matching these risks to the relevant solutions from our recommendations.

Significance was used to evaluate the level of impact our theory had on the case company. We could not fully assess how significant the impact of our proposed theory would be after the full roll-out across the whole company. As to the pilot project, we recognized that the previous efforts at the company could not address all the needs for governance, while the getting started handbook provided significant guidance and support for the transition towards governance in the scope of the pilot project. In this limited evaluation, we observed some challenges, such as the lack of examples in the handbook. We noted that the most significant aspect of our theory to the pilot project employees was that they could use our handbook as a strong argument in front of the top management demonstrating the significance of corporate open source governance at the company.

Usefulness helped evaluate how much value the handbook added to the case company in solving the key issues of getting started with FLOSS governance, as well as whether it enhanced the employee knowledge on these issues and their solutions. Similar to the evaluation criteria of significance, we could not assess the usefulness of our theory to the whole company during this case study. We found that the main issue making the handbook less useful was its abstract nature. We agreed with this observation but highlighted that the handbook was abstract by design and required in-company adjustment. Nonetheless, we considered this as a limitation to the usefulness of our theory for the companies unwilling to perform the required customization of the handbook.

5. Limitations

The main limitation of our study is that the results are derived based on a single-case case study. To mitigate this limitation, we conducted a broad (across five divisions) and deep (longitudinal for two and a half year) evaluation study. Another limitation is the large number of confounding factors when evaluating the complex phenomenon of getting started with open source governance. Recognizing this limitation, we nonetheless settled on the research method of case study in order to test the proposed practices in a real-life context.

To ensure the internal validity of our case study, we made sure to follow the research methodology by Yin [43] rigorously, following the predefined case study protocol throughout the study. In this protocol, we addressed the specifics of the evaluation criteria, the interview questions and format, data collection and analysis.

As to the external validity, generalizability cannot be proved, but our work shows that a transfer (of best practice implementation) is possible to a company with no prior understanding of open source governance. Our findings let anyone decide whether a transfer of results is possible to their company.

6. Conclusions

Having proposed a handbook of industry best practices for getting started with corporate open source governance, we conducted an evaluation case study spanning two and a half years. We assessed the initial state of FLOSS governance at a Germany-based multinational company active in several distinct industries. We then guided the implementation of the handbook at five divisions of the company starting with Division 1, where a pilot project team followed our recommendations and allowed us to observe their progress and the created artifacts.

In this paper, we reported the key results of the case study, covering the initial governance situation and the open source use in the company, the evaluation of individual best practices we had proposed, as well as the overall evaluation of the getting started handbook. For the latter, we used the interdisciplinary evaluation criteria of completeness, comprehension, understandability, significance, and more. Having discovered some of the shortcomings of our recommendations, we also found that our handbook addressed most of the company concerns in regard to establishing formal open source governance.

We found that most practices were well-structured, comprehensive, and applicable. However, some practices, such as OSGOV-GETSTA-TRAORG-4. Start small, then replicate - define the scope of the transition process lacked understandability, while some others lacked usefulness in the context of the case company, such as OSGOV-GETSTA- TRAORG-1. Establish a board of stakeholders to organize the transition. The feedback collected during the evaluation study helped us improve our theory and make the handbook for getting started with open source governance more applicable and industry-relevant.

7. Acknowledgments

We would like to acknowledge our case study partners for their collaboration. We would also like to

acknowledge the anonymous reviewers for their valuable feedback that helped us improve the paper significantly.

8. References

[1] Ågerfalk, P. J., Deverell, A., Fitzgerald, B., & Morgan, L. (2006). State of the art and practice of open source component integration. In the 32nd Conference on Software Engineering and Advanced Applications. IEEE, 170–177.

[2] Barcomb, A., Stol, K. J., Fitzgerald, B., & Riehle, D. (2020). Managing Episodic Volunteers in Free/Libre/Open Source Software Communities. IEEE Transactions on Software Engineering.

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

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

[5] Cavaye, A. L. (1996). Case study research: a multi-faceted research approach for is. Information Systems Journal, 6(3), 227–242.

[6] 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.

[7] 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.

[8] Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/libre open-source software development: What we know and what we do not know. ACM Computing Surveys, 44(2).

[9] Eisenhardt, K. M. (1989). Building theories from case study research. Academy of Management Review, 14(4), 532–550.

[10] 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.

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

[12] Fitzgerald, B. (2006). The Transformation of Open Source Software, MIS Quarterly, 30 (3), 587.

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

[14] Gangadharan, G., D'Andrea, V., DePaoli, S., & Weiss, M. (2012). Managing license compliance in free and open source software development. Information Systems Frontiers, 14(2), 143–154.

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

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

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

[18] Harutyunyan, N., & Riehle, D. (2019, August). Getting started with FLOSS governance and compliance: A theory of industry best practices. In Proceedings of the 15th International Symposium on Open Collaboration.

[19] 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.

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

[21] 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.

[22] Harutyunyan, N., & Riehle, D. (2019, January). User experience design in software product lines. In Proceedings of the 52nd Hawaii International Conference on System Sciences.

[23] Harutyunyan, N., Bauer, A., & Riehle, D. (2019). Industry requirements for FLOSS governance tools to facilitate the use of open source software in commercial products. Journal of Systems and Software, 158, 110390.

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

[25] Jiang, Q., Qin, J., & Kang, L. (2015). A literature review for open source software studies. In International Conference on HCI in Business. Springer, 699–707.

[26] Kapitsaki, G. M., Tselikas, N. D., & Foukarakis, I. E. (2015). An insight into license tools for open source software systems. Journal of Systems and Software, 102, 72–87.

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

[28] von Krogh, G., & von Hippel, K. (2006). The Promise of Research on Open Source Software, Management Science 52 (7), 975–983.

[29] Li, Y., Tan, C.-H., & Teo, H.-H. (2012). Leadership characteristics and developers motivation in open source software development. Information & Management, 49(5), 257–267.

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

[31] O'Mahony, S. (2007). The governance of open source initiatives: what does it mean to be community managed?. Journal of Management & Governance, 11(2), 139–150.

[32] O'Mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. Academy of Management Journal, 50(5), 1079–1106.

[33] 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.
[34] Radcliffe, M., Odence, P. (2017): The 2017 Open Source Year in Review. In: Black Duck Software, DLA Piper. (self- published presentation)
[35] Riehle, D., & Harutyunyan, N. (2019) Open-source license compliance in software supply chains. In Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability. Springer. 83-95
[36] Riehle, D., Harutyunyan, N., & Barcomb, A. (2020). Pattern Discovery and Validation Using Scientific Research Methods. Friedrich-Alexander-Universität Erlangen-Nürnberg. Technical Report, CS-2020-01
[37] Ruffin, C., & Ebert, C. (2004). Using open source software in product development: A primer. IEEE Software, 21(1), 82-86.
[38] Russell, C. K. & Gregory, D. M. (2003). Evaluation of qualitative research studies. Evidence-Based Nursing, 6(2),36– 40.
[39] Shenton, A. K. (2004). Strategies for ensuring trustworthiness in qualitative research projects. Education for Information, 22(2), 63–75.
[40] Schwab, B., Riehle, D., Barcomb, A., & Harutyunyan, N. (2020). The Ecosystem of openKONSEQUENZ, A User-Led Open Source Foundation. In IFIP International Conference on Open Source Systems. Springer, 1–13.
[41] Trochim, W. M. (1989). Outcome pattern matching and program theory. Evaluation and program planning, 12(4), 355– 366.
[42] Trochim, W. M. (2006). Qualitative measures. Research measure knowledge base, 361, 29–31.
[43] Yin R. K (2013). Case study research: Design and methods. Sage publications.