Getting Started with Open Source Governance and Compliance in Companies

Nikolay Harutyunyan
Friedrich-Alexander University Erlangen-Nürnberg
Erlangen, Germany
nikolay.harutyunyan@fau.de

Dirk Riehle
Friedrich-Alexander University Erlangen-Nürnberg
Erlangen, Germany
dirk@riehle.org

ABSTRACT

Commercial use of open source software is on the rise as more companies realize the benefits of using FLOSS components in their products. At the same time, the ungoverned use of such components can result in legal, financial, intellectual property, and other risks. To mitigate these risks, companies must govern their use of open source through appropriate processes. This paper presents an initial theory of industry best practices on getting started with open source governance and compliance. Through a qualitative survey, we conducted and analyzed 15 expert interviews in companies with advanced capabilities in open source governance. We also studied practitioner reports on existing practices for introducing FLOSS governance processes. We cast our resulting initial theory in the actionable format of best practice patterns that, when combined, form a practical handbook of getting started with FLOSS governance in companies.

Author Keywords

Commercial Use of Open Source; FLOSS; FOSS; Industry Best Practice; Introduction of FLOSS in Companies; Open Source Software; Open Source Governance; Qualitative Survey.

INTRODUCTION

Companies have been using open source tools for software development for a long time [7, 11, 23, 24, 28], but in recent years more and more companies have been introducing open source components into their products. While beneficial, this carries certain risks if a company has no rules or guidelines for such use of open source components. The ungoverned use of FLOSS components can result in legal problems resulting from inadequate license compliance, operational issues resulting from lengthy release reviews including scanning and documenting the used open source components, financial and intellectual property issues resulting in litigation, cease and desist claims or product recalls [9, 21, 25, 27]. To mitigate these risks companies must govern their use of open source software thought FLOSS governance processes and guidelines.

We define FLOSS governance as the set of processes, best practices, and tools employed by companies to use FLOSS components as parts of their commercial products while minimizing their risks and maximizing their benefit from such use [13]. In the context of this paper, the definition of FLOSS governance should not be confused with other definitions that cover the governance of open source communities or projects, such as the definition by Markus [18]: "[Open source governance is defined as] the means of achieving the direction, control, and coordination of wholly or partially autonomous individuals and organizations on behalf of an OSS development project to which they jointly contribute".

FLOSS governance can apply to the commercial use, contribution or leadership of open source projects. However, we limited the scope of this paper to the commercial use of open source components only, intentionally excluding governance considerations of companies contributing to or leading open source communities or projects. This focus enabled us to build and present a detailed theory covering multiple aspects of getting started with open source governance in companies, a topic of the highest practical relevance to most companies today and novel to FLOSS research [14].

Depending on the maturity of a company's open source governance, it can cover topics such as governance management, open source program office, license compliance, component search, component selection, component approval, component integration, component repository and reuse, software product model, supply chain management, communication, capabilities and more. In this paper we focus on companies that are new to governing their use of open source. Thus, our scope is limited to the governance aspects of getting started with the adoption of FLOSS components. Companies lacking established processes, best practices, or tools must transition towards

OpenSym '19, August 20–22, 2019, Skövde, Sweden

ACM ISBN 978-1-4503-6319-8/19/08…$15.00
https://doi.org/10.1145/3306446.3340815

FLOSS governance to maximize the benefits of using open source while minimizing the risks. To do so they should follow findings from research and industry best practices. Our research focuses on the little researched yet highly industry-relevant inquiry on the specific ways companies undertake this transition. Thus, our research question:

RQ: How should companies using open source components in their products get started with open source governance based on existent industry best practices?

The research method we employed is a qualitative survey [15]. Our paper presents an interview-based qualitative survey exploring the open source governance introduction experiences of 15 software development companies that use open source components in their products and have successfully introduced governance processes. We performed data gathering and analysis using formal semi-structured interviews, researcher notes, and materials review. We interviewed FLOSS governance and compliance experts from 15 diverse companies chosen through theoretical sampling of more than 140 companies.

The contribution of our paper is an initial theory of industry best practices for getting started with open source governance in companies. The theory proposes a number of best practices in the following thematic areas:

- Product Analysis (OSGOV-PROANA)
- Transition Policy (OSGOV-TRAPOL)
- Transition Organization (OSGOV-TRAORG)
- IP-at-Risk Analysis (OSGOV-IPRISK)
- Communication and Capabilities (OSGOV-COMCAP).

Each of these themes covers a subset of industry best practices we identified based on the analysis of the expert interviews and industry materials. We cast the individual best practices in the format of context-problem-solution patterns that, when combined, form a practical handbook of getting started with FLOSS governance in companies. For examples of patterns, see Tables 2, 3 and 4. This format is well structured allowing for interconnection of best practices within and across the thematic subsections. We used such interconnections to illustrate workflows combining subsets of best practice patterns, called process templates. For examples of process templates, see Figures 1, 2, 3 and 4. This presentation format is actionable and highly practice-relevant, as it enables industry to apply the findings of our research by adapting and applying the best practices we identified in their companies. The latter was a priority of ours, as we wanted to increase the potential impact of our work.

RELATED WORK

Researchers have recognized many benefits of open source software adoption by companies including better interoperability, interconnectivity, trialability, transparency due to the availability of the source code [3, 5, 6, 11, 19, 24]. While some literature exists on FLOSS adoption in industry and on FLOSS governance in general [1, 2, 4, 8, 14, 20, 29] we found little research particularly about industry best practices for getting started with open source governance in companies. Therefore, we also reviewed general FLOSS governance research literature, and compared and contrasted it with our findings on the getting started aspects of the phenomenon.

Ruffin & Ebert [27] talk about possible risks and benefits of using open source software. Besides advantages like saving time and improving security, they point out that companies should be vigilant about the open source components used in their products, preventing possible copyright infringement of third parties and their intellectual-property rights. They also talk about several actions that can be undertaken to mitigate legal exposure, such as governing the use of open source components through well documented processes. We confirmed their findings and identified best practices that help avoid potential risks of the ungoverned use of open source components. Namely, the best practice OSGOV-IPRISK-2 *Analyze risk exposure of using an open source component* covers the potential risks of using open source components and suggests how companies analyze and prevent such risks.

Bonaccorsi & Rossi [2] discuss three key economic problems that arise with the emergence of the commercial use of open source: motivation, coordination, and diffusion. As they explain the different types of open source users, they introduce coordination as a basic alternative to open source governance including having a centralized leadership structure and a clear hierarchical organization or having technical support systems within a company to deal with the use of open source components. They discuss the diffusion of open source in companies. We do not discover industry best practices for clear hierarchical organization when dealing with open source adoption. Instead, we find that the transition towards open source governance should involve stakeholders from all hierarchical levels in a company, guided by a transition policy outlined in OSGOV-TRAPOL-1 *Establish FLOSS governance policy for the transition period*. This best practice also confirms findings by Lerner & Tirole [17] who highlight open source governance policies and internal legal systems as a way to prevent potential risks of ungoverned FLOSS use.

Kemp [16] talks about the operational aspects of the transition towards open source governance in companies. Analyzing the management perspective on the transition, he indicates that the goal for the management undertaking the transition it to install integrated processes across all relevant business functions to manage effective use of FLOSS throughout the organization. He argues that, to get there, an organization should consider disassembling the various pieces into their building block components and threading them together by start point (achievements to date), people

(stakeholders) and the strategic, policy and process aspects. We find industry best practices matching Kemp's findings. Namely, OSGOV-PROANA-3.1 *Run open source use analysis in products* ensures that the management is aware of the various open source components that are currently used in the company, which leads to the best practice OSGOV-PROANA-3.2 *Document current open source use*. As to the governance transition process, we identified the practice OSGOV-TRAORG-6 *Establish the transition process* that covers the operational aspects of setting up the process.

Fendt et al. [9] discuss some critical risks that can arise from the ungoverned use of open source software in products, such as compliance issues when dealing with open source software licenses. They go on to describe a suggested governance process and framework that aim to allow only appropriate FLOSS components into products, to protect internally developed code from potential risks of license non-compliance, and to assure the fulfilment of all license requirements. Furthermore, they state that the process automation and other factors have to be considered when implementing a FLOSS governance process. Our theory touches on some of these issues discovering, for example, an industry best practice for using tools to automate parts of the getting started process in companies - OSGOV-PROANA-1.3 *Select and use governance tools for automation*.

Fitzgerald [11] talks about the specifics of product analysis as he describes the transformation from open source software development to an emerging commercially viable form of open source he calls OSS2.0. He talks about the commercial use of open source software and the challenges of this transition. Our theory addresses some of these challenges providing industry best practices for the initial product analysis in particular. The best practices OSGOV-PROANA-1 *Use a combination of methods for product analysis* covers the methods a company can use for the initial product analysis to identify the previously used yet ungoverned open source components.

The topic of open source governance introduction in companies is of high practical relevance to the industry. Practitioners like Peters [19] analyzed some getting started aspect of open source governance. He highlighted the importance of open source governance policies during transition to FLOSS use in companies. He provided a guide intended to support the creation of a company's open source governance policy. He presented tips and best practices of writing such a policy, intended to regulate the use of open source in corporate environments. These best practices focused on identifying stakeholders, choosing a strategy, and setting the scope. Our theory confirms some of the best practices he identified. Namely the best practice OSGOV-TRANS-1 *Establish a board of stakeholders to organize the transition* sums up the need to identify the stakeholders interesting in the introduction of open source governance processes in the company, and presents the specifics of organizing these stakeholders. Another best practice in our theory OSGOV-TRAORG-4 *Start small, then replicate - define the scope of the transition process* deals with the scope of the transition towards open source governance.

Bonaccorsi et al. [3] talk about companies that use open source software in their products as part of their business strategy. They discuss how using open source influences a company's business model choice. They state that many companies choose to adopt a hybrid business model that comprises proprietary as well as open source products and services. Besides, they also talk about a company's motivation to use open source software in their products and about the factors that influence a company's openness towards using open source. Considering the scope of our paper, we did not investigate the influence of using open source on a company's business model, but rather focused on the reasons and techniques companies follow when getting started with FLOSS use in products. In line with Bonaccorsi's findings, our theory recognizes communication and capability building as central topics of industry best practices for open source governance. Namely we identified a subset of industry practices on the issues including OSGOV-COMCAP-1 *Establish communication channels for open source governance handbook* to OSGOV-COMCAP-5 *Provide employee training,* which talk about setting up internal communication channels, developing and providing employee training.

RESEARCH METHOD

We defined the following research question and sub-questions for our study:

RQ1: How should companies using open source components in their products get started with open source governance?

RQ1.1: How should companies analyze their current use of open source components?

RQ1.2: How should companies transition towards open source governance?

RQ1.3: How should companies analyze and mitigate the risks of ungoverned use of open source components?

To answer these research questions, we conducted a qualitative survey using interviews with industry experts to collect data [10, 15]. Methodologically, qualitative surveys resemble multiple-case case studies [15, 26, 30], in that they both are systems for collecting information from or about people to describe, compare, or explain their knowledge, attitudes, and behavior about a studied phenomenon in a real-life setting [10]. However, while case study research design enables an in-depth analysis of particular cases, the qualitative survey focuses on less specific, yet more comprehensive and all-around

perspective of the subject. As our goal was the latter we used a qualitative survey to answer our research questions.

First we set objectives to collect information, design and plan the study, conduct a theoretical sampling, and choose expert interviews as our main source of data. We then prepared the interview questions that covered different pre-defined aspects or topics of getting started with open source governance. Using semi-structured interviews as our survey instrument, we conducted the interviews in an iterative manner adjusting the questions after each iteration, yet keeping the core topics of the questions intact. We then transcribed and processed the interviews to prepare for data analysis. To analyze survey data, we employed qualitative data analysis (QDA) aided by MaxQDA (a QDA tool) in order to ensure the systematic analysis of the data and the traceability of our theory to the data. Finally, we are reporting our findings as an initial theory of industry best practices in this paper. A best practice is a method reflecting the state-of-the-art as applicable in a particular context [22]. This paper presents some (but not all) of the best practices we developed.

Theoretical Sampling

We chose 15 companies sampled from our industry network of about 140 companies with advanced FLOSS governance practices. The companies in our sample are experienced in FLOSS governance and compliance. We conducted polar theoretical sampling to cover a diverse and representative set of companies, which resulted in a sample with highly varying characteristics [10, 15]. The list of companies and characteristics are presented in Table 1. Company names are anonymized per their request.

Company	Company domain	By size	By type of customer
C1	Consulting	Medium	Enterprise
C2	Automotive	Small	Enterprise
C3	Automotive	Large	Enterprise
C4	Enterprise Software	Medium	Enterprise, retail
C5	Enterprise Software	Medium	Enterprise, retail
C6	Enterprise Software	Large	Enterprise, retail
C7	Enterprise Software	Medium	Enterprise, retail
C8	FLOSS Foundation	Small	Enterprise, retail
C9	Hardware and Software	Large	Enterprise
C10	Legal	Large	Enterprise, government
C11	Enterprise Software	Medium	Enterprise
C12	Consulting, Enterprise Software	Large	Enterprise
C13	Hardware and Software	Large	Enterprise, retail, government
C14	Enterprise Software	Small	Enterprise
C15	Enterprise Software	Large	Enterprise

Table 1. Sample of companies

Data Gathering and Analysis

For data gathering, we used semi-structured interviews conducted by one or two researchers with FLOSS governance experts or responsible coworkers from the sampled companies. In twelve companies we interviewed one expert, in one company we interviewed two experts, and in two companies we interviewed three experts. In total, we conducted 15 interviews. We recorded and transcribed the interviews. In three cases we took notes.

Our interviewees with experience in open source governance can be divided in two groups: software developers, and managers or supporting functions. The latter includes open source managers, technical managers and lawyers. In preparation of the interview guideline and questions, we prepared additional questions only relevant for each of the interviewee groups to collect more in-depth data on specific software development and management aspects of open source governance.

For data analysis, we followed the qualitative survey method by Jansen [15], performing iterative and incremental qualitative data analysis (QDA) supported by the MaxQDA tool. We developed a coding system that covered the predefined topics of getting started with open source governance. During the QDA coding process, we iteratively refined the code system. After reaching theoretical saturation, the code system became the basis for our theory. Individual codes correspond to the best practices we identified for the proposed theory. Our code system consists of hierarchical codes. We conducted the QDA process as follows:

- *Open coding*. We created a basic set of codes from which the hierarchy is built. Open codes are direct annotations of primary materials and link to them for data-theory traceability.
- *Axial coding*. We built a code system by deriving more abstract concepts and categories from open codes, thus developing the axes of the code system.

- *Selective coding*. We applied the codes to the gathered data and chose which codes are important and which are not. We adjusted the coding system by removing the irrelevant codes and by adding the ones that emerged during axial codes.

RESEARCH RESULTS

As a result of our study, we answered to the research question and subquestions on getting started with open source governance with an initial theory of industry best practices that emerged from our qualitative survey.

The theory we developed proposes a number of best practices in the following areas of FLOSS governance:

- **Product Analysis** (abbreviated as OSGOV-PROANA) - **8 best practices**
- **Transition Policy** (abbreviated as OSGOV-TRAPOL) - **3 best practices**
- **Transition Organization** (abbreviated as OSGOV-TRAORG) - **8 best practices**
- **IP-at-Risk Analysis** (abbreviated as OSGOV-IPRISK) - **9 best practices**
- **Communication and Capabilities** (abbreviated as OSGOV-COMCAP) - **5 best practices**.

Two specific aspects of open source governance are critical in the getting started phase: analysis of open source software used in products and mitigation of risks resulting from this ungoverned use.

Product Analysis (OSGOV-PROANA)

Answering to RQ1.1, our theory summarizes a number of industry best practices on the scanning of the software product code for license compliance, creating a product architecture model including open source components and their metadata, and documenting open source use analysis in products. Best practices in this category include:

1.	Use a combination of methods for product analysis
1.1.	Use one mandatory survey for initial assessment
1.2.	Establish a process of continuous reporting and assessment
1.3.	Select and use governance tools for automation
2.	Establish and use a product architecture model
2.1.	Create product architecture model
2.2.	Maintain product architecture model
3.	Run use analysis
3.1.	Run open source use analysis in products
3.2.	Document current open source use

Product analysis is a critical part of getting started with open source software. Before setting up open source governance processes, a company must identify and analyze the current use of open source components that have been used but not approved or documented before. Proposed best practices in this category describe methods for analyzing the current use of open source including a mandatory survey for initial situation assessment, and ways to document the identified open source component and their metadata. After the initial assessment, companies should establish a process of continuous reporting and assessment for the OSS components used from that point on, which is described in detail in an example best practice from our theory in Table 2.

ID/ Name	**OSGOV-PROANA-1.2 Establish a process of continuous reporting and assessment**
Context	You already → *used one mandatory survey for initial assessment*. Now you need a process for continuous reporting and assessment of any open source use during the transition.
Problem	The transition needs to prepare the company for fully structured FLOSS governance. However, during the transition how should the process of continuous reporting and assessment look like?

Solution	Establish a process of continuous reporting and assessment that involves defined and easy to follow steps for developers when using a new open source components during the transition. This can be achieved using a product architecture model (a meta-model for all governance related information such as license information, copyright noticed, export restrictions, etc.), bill-of-materials documentation, questionnaires or forms etc. The process should help:
	- continuously report and track new use of open source components
	- continuously assess and approve/reject new use of open source components
	  - assess license compliance
	  - assess copyright notices
	  - assess export restrictions
	  - assess software supply chains
	- store and share the reported data on the used open source components.

Table 2. Best practice OSGOV-PROANA-1.2

Another industry best practice is the creation of a product architecture model to set up and maintain a structured and formalized view of software components used. Companies should define open source component-specific properties within the model to allow collection, tracking, maintenance, and monitoring of metadata including open source license information, export restrictions, known security vulnerabilities, and software dependencies. If possible, the product architecture model should be integrated into the build process or continuous development process to ensure higher automation.

The industry best practices of our theory can be traced to the data from the qualitative survey we performed. Here is an example of such a trace from Company 15's legal

counsel responsible for open source compliance talking about the specifics of establishing a process of continuous reporting and assessment of OSS components:

"When our developers are reporting the open source via [our internal tool], there is always the main file which is also mentioned in the license file which is also computed by GitHub or by the community behind. And with this scan tooling we cross check the whole software, so we definitely see, okay, that's not only the MIT license which is mentioned in the license file but also other licenses, so the GPL files. (Interview at Company 15)

Here is another example of such a trace from Company 2's open source compliance manager:

"We ask the developers to report those kinds of components that have this kind of licenses, and then the license checks the components and the rough context is documented and the system goes to a board, companywide board where we have software developers, the compliance managers, the lawyers and a patent lawyer, a copyright lawyer. A group sits together and then discusses that and makes decisions on the license terms." (Interview at Company 2)

Transition Policy (OSGOV-TRAPOL)

Partially answering to RQ1.2, we found that most companies establish guidelines for getting started with open source governance. We call these guidelines a transition policy, which must be established, communicated, and continuously adjusted and improved. The transition policy outlines the principles for the transition, but does not cover any operational aspects of the transition, which is done through the transition organization. Policy practices include:

1.	Establish FLOSS governance policy for the transition period
2.	Communicate FLOSS governance policy for the transition period
3.	Adjust and improve FLOSS governance policy for the transition period

FLOSS governance policy for the transition period covers all the critical issues around use of open source components in products, such as license compliance, bill-of-materials management, documentation, and communication. The policy can be stored as a single document or divided into two separate documents. The first explains the intention of the FLOSS use, defines the principles of using FLOSS in products. It outlines what kind of licenses, including their legal assessments and packages are acceptable for use in commercial products, and pairs legal assessments with business use cases for each license. The second establishes a set of standards and tasks for the employees to follow to ensure compliance with FLOSS governance processes. This way, the policy can be implemented across the whole company under identical conditions. Also at larger companies, each division or department can adopt the policy with certain differences.

Transition Organization (OSGOV-TRAORG)

Completing the answer to the RQ1.2 from section 4.2, we identified that a common pattern for organizing the transition to governance follows these best practices:

1.	Establish a board of stakeholders to organize the transition
2.	Designate the transition manager
3.	Define responsibilities and tasks of the transition manager
4.	Start small, then replicate - define the scope of the transition process
5.	Define the transition timeline
6.	Establish the transition process
7.	Communicate the transition process
8.	Implement the transition process

We found that establishing a board of stakeholders to organize the transition is a starting point for getting started with open source governance. These stakeholders include the everyday users and decision makers in regard of open source, including but not limited to senior developers, engineering managers, lawyer, business/product managers, software architect, software procurement officer. The details on establishing a board of stakeholders are captured in the best practice OSGOV-TRAORG-1.1 presented in Table 3. This industry best practice of our theory can be traced to the data from the qualitative survey we performed. Here is an example of such a trace from the interview with Company 10's legal expert talking about the stakeholders of open source governances:

"[talking about the organizers of the transition] I think it should not be legal department, generally what you have is a board and the board ... generally has somebody from legal, from business, and from the technical community. What you want is the board to take a look at risk and say, take a look at technical risk and the business risk. ... Somebody in the legal has to make sure that licenses are compliant. Legal should not be the only people making this decision, because there is more than just a legal issue here." (Interview at Company 10)

Here is another example for a data trace for this best practice by an executive from Company 1:

"It kind of depends on the way the company is run and the people in there. So if you have a lot of technical people who have no real connection to [open source] but you make the decision to use open source, yes then you need somebody or a board to make [governance] decisions because if you have thousands of technical people, you cannot have everyone make their own decisions. Then you have so many different tools, it is going to get tricky. So you have to find some way of reaching a centralized decision." (Interview at Company 1)

ID/ Name	**OSGOV-TRAORG-1.1 Establish a board of stakeholders to organize the transition**
Context	Your company came to recognize the importance of FLOSS governance. You decided to regulate your use of open source software in products using FLOSS governance best practices.
Problem	Before rolling out an overarching FLOSS governance process, you need to review all the existing products that include open source software components. Where and how do you start?
Solution	To start reviewing your existing products and their software components, you need to follow best practices on getting started with FLOSS governance. As a first step, establish a board of stakeholders to organize the transition from ungoverned FLOSS use to structured FLOSS governance. Your transition board should include the current users of open source in the company, decision makers regarding FLOSS use and those to be responsible for FLOSS governance in the future. For the transition board, consider the following employees: - senior developers (known internally for their open source use and competency) - engineering managers (usually de facto decision makers on FLOSS matters) - lawyer (responsible for FLOSS license clearance and related issues) - business/product managers - software architect - software procurement officer. The transition board should be inclusive and transparent, open for any interested stakeholder to join. The board should not require full-time engagement of all the members. However, it's important to → *designate the transition manager* - a responsible role and person for the transition, and to → *define responsibilities and tasks of the transition manager*.

Table 3. Best practice OSGOV-TRAORG-1.1

As to the transition process, we found these common aspects of such a process:

- outlining the motivation behind FLOSS governance
- clarifying the roles of the employees during the transition
- communicating the timeline and scope of the transition
- communicating the steps of the transition (product analysis and risk mitigation) and expected outcomes
- setting up new and structured procedures for governance decision making.

IP-at-Risk Analysis (OSGOV-IPRISK)

Answering to the RQ1.3, the getting started theory highlights industry best practices on analyzing potential risks of the ungoverned use of open source and ways to mitigate these risks. An overview of these practices includes:

1.	Run license compliance analysis
1.1.	Develop standard license interpretation
1.2.	Use standard license interpretation
1.3.	Create license-use case pairs
2.	Analyze risk exposure of using an open source component
3.	Mitigate risk to intellectual property
3.1.	Replace problematic components
3.2.	Decouple problematic components
3.3.	Require bill-of-materials for supplied code by 3rd party post-factum
3.4.	Run random audits to identify previously undetected or missed open source components and their metadata
4.	Analyze security risk of using an open source component

Potential risks of the ungoverned FLOSS use include open source license non-compliance, security risks and other risks to a company's intellectual property. Once the risks are identified and analyzed, companies need to mitigate them by replacing or decoupling the problematic components depending on the use case and on the license of the component used. Other mitigation practices suggest requiring detailed bill-of-materials for the supplied code by third parties after the delivery, as well as running random audits to identify the metadata of the previously undetected or missed open source components. Table 4 presents one of the best practices on the topic of IP-at-risk when getting started with open source governance.

This industry best practices of our theory can be traced to the data from the qualitative survey we performed. Here is an example of such a trace from the interview with Company 2's open source compliance manager talking about the specifics of open source license-use case pairs:

"[Our] open source handbook doesn't really present rules in a concrete setup, but what it explains all the interpretations of the licenses that we have [used]. We assess licenses with lawyers, with our internal lawyers, and from these license assessments, we determine certain [company] rules for its usage, modification, and contribution. And these rules for the individual licenses are explained in that document [as license-use case pairs]." (Interview at Company 2)

Here is another example for a data trace for this best practice by a legal expert from Company 10:

"The first thing to recognize is that one size [of license compliance] does not fit all, there are sort of what I view as a couple different use cases, the first most important use case is anything that gets the delivered outside the company, something that gets distributed. Why is that? Because all the copyleft licenses except the AGPL v3 depend on distribution, which is a transfer of copy to trigger the obligation. If you have a total SaaS

infrastructure, you probably have a lot less risks then in a standalone application, particularly with GPL v2, as it's not a distribution. You need to match use cases with open source license interpretations." (Interview at Company 10)

ID/ Name	OSGOV-IPRISK-1.3 Create license-use case pairs
Context	Your company → *developed standard license interpretation* and you are → *using standard license interpretation.* Developers are also consulting company's → *established FLOSS governance policy for the transition period*, and are contacting the transition board or the transition manager for case by case review of special cases of FLOSS use.
Problem	What's the best way to document the case by case decisions on special cases of FLOSS use, reviewed by the transition board or the transition manager?
Solution	In one centrally available document, create license-use case pairs to document the case by case decisions on special cases of FLOSS use. This document should include all the major licenses and company's detailed approach to their use in different business contexts or use cases. For example, it can be acceptable to use a copyleft license for certain (non-differentiating) products, while it might be unacceptable in other cases such as for company's main products (with competitive advantage). Such license-use case pairs should be well structured and documented. In case of a new decision on a special license-use case pair by the transition board, this document must be updated by the transition manager. Developers must consult the document before contacting the transition board or the transition manager with a new review request, because they might be able to find their answer for a specific license-use case pair in the document. Having such a document improves performance and reduces unnecessary redundancy.

Table 4. Best practice OSGOV-PROANA-1.2

Communication and Capabilities (OSGOV-COMCAP)

Beyond the research questions we asked in the study, we identified some meta-level best practices that enable a smooth transition towards open source governance. We group these practices under the category of communication and capabilities of our theory, including:

1.	Establish communication channels for open source governance issues
2.	Assess open source governance capabilities among developers and engineering manager
3.	Develop FLOSS governance and compliance capabilities at the central legal department
4.	Design employee training
5.	Provide employee training

This category of best practices covers the communication channels a company should use when getting started with FLOSS governance, as well as practices on assessing and building open source governance capabilities among developers, managers and support function employees. Building such capabilities includes employee training, as well as learning from academic literature, governance experts and organizations such as Linux Foundation, TODO Group, OpenChain and SPDX working groups etc.

CONCLUSION AND DISCUSSION

The main contribution of this paper is a theory of industry best practices on getting started with open source governance, which we cast as a handbook of best practice patterns. Our proposed theory introduces actionable best practice that emerged from our data analysis. We found that the identified industry best practices are interconnected, which can be illustrated through workflow diagrams, as we call them. Figure 1 and Figure 2 illustrate examples of such process templates for the transition organization (TRAORG).

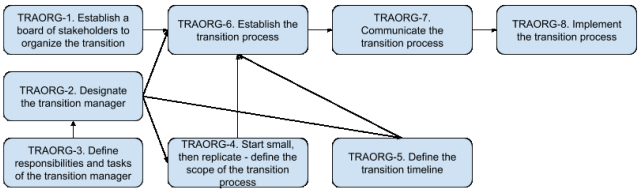

Figure 1. Example 1 of a Workflow Diagram for Transition Organization when Getting Started with FLOSS Governance

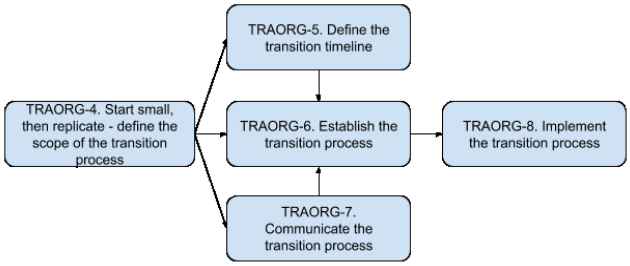

Figure 2. Example 2 of a Workflow Diagram for Transition Organization when Getting Started with FLOSS Governance

Companies getting started with open source governance should start with a transition organization guided by a transition policy. The transition policy helps a company define its principles in regard of open source use and governance. The transition organization then operationalizes the principles defined in the policy, turning them into a process that involves different stakeholders that have been or would be using open source components in products, or making decisions regarding open source governance. The transition organization starts with

establishing a board of stakeholder and a transition manager who oversee and organize the transition that includes defining the transition timeline and scope, as well as defines and implement the transition process.

Figures 3 and Figure 4 illustrate examples of process templates for the product analysis (PROANA) when getting started with open source governance.

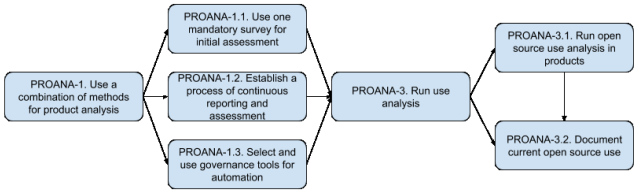

Figure 3. Example 1 of a Workflow Diagram for Product Analysis when Getting Started with FLOSS Governance

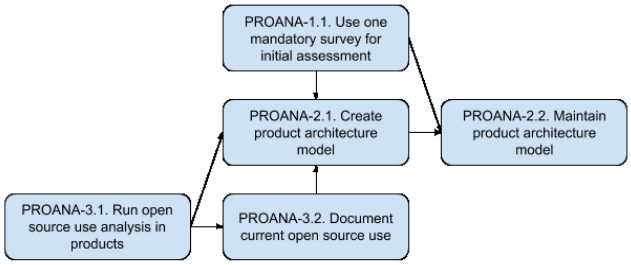

Figure 4. Example 2 of a Workflow Diagram for Product Analysis when Getting Started with FLOSS Governance

Once the transition organization and policy are set up, companies need to analyze their current use of open source components in products. To do so, companies should use a combination of methods for product analysis, including manual surveys and governance tools for automation. This is then followed by the documentation of the identified open source use in products, preferably using a structured and well-maintained product architecture model.

In this exploratory study we do not go into the evaluation of the theory. In our further work, we plan to evaluate this theory using case study research [25, 30], currently running. For the evaluation, we took a subset of the best practice patterns resulting from this research and implemented them in a case study company, currently measuring how mature, complete, correct and comprehensive our theory is. We plan to publish our findings once the case study is completed.

In further work, we also plan to extend our research beyond the focus on getting started with open source governance. We will apply the same research methods to study industry practices for supply chain management and FLOSS governance.

We recognize that our research results while limited in scope are relevant and novel. They present a theory of the issue that can become the groundwork for future studies into FLOSS governance by the authors and other scholars that will hopefully expand the proposed theory.

RESEARCH LIMITATIONS

To address the limitations of our study, we follow [12] in assessing the trustworthiness of our research through the following quality criteria:

Credibility. Credibility is the degree to which we can establish confidence in the truth of our findings in the context of the inquiry. To ensure credibility during data collection we conducted our interviews iteratively, adjusting our semi-structured interview questions based on the company context and on our experience with earlier interviews. We also conducted peer debriefing regarding our findings.

Dependability. Dependability is the degree of consistency of the findings and traceability from the data to the results. We ensured dependability by collecting and saving raw interview data, documenting our qualitative data analysis in different stages of the coding and by documenting our analysis in a manner that allows tracing each requirement in our theory to its origin in our collected data. We included direct references to the expert interviews in the presentation of our research findings.

Confirmability. Confirmability is the degree to which the authors are neutral towards the inquiry and their potential bias effect on the findings. Qualitative data research realized by one researcher has inherent subjectivity and bias. Even though we followed the research method constructs carefully, there is bias associated with method interpretation and application to our specific context. To address this limitation, we had a second coder analyze our data and improve our original QDA coding based on input from the second coder.

ACKNOWLEDGMENTS

This research was funded by BMBF's (Federal Ministry of Education and Research) Software Campus 2.0 project (OSGOV, 01IS17045-17570). We would like to thank our colleagues and the anonymous reviewers for their feedback. We would also like to thank our industry partners that provided their valuable expertise for this research project.

REFERENCES

1. Aksulu, A., Wade, M.: A comprehensive review and synthesis of open source research. In: Journal of the Association for Information Systems, 11(11), 576-656 (2010)
2. Bonaccorsi, A., Rossi, C.: Why open source software can succeed. In: Research policy, 32(7), 1243-1258 (2003)

3. Bonaccorsi, A., Giannangeli, S., Rossi, C.: Entry strategies under competing standards: Hybrid business models in the open source software industry. In: Management science, 52(7), 1085-1098 (2006)
4. Capra, E., Francalanci, C., Merlo, F.: An empirical study on the relationship between software design quality, development effort and governance in open source projects. In: IEEE Transactions on Software Engineering, 34(6), 765-782 (2008)
5. Chau, P. Y., Tam, K. Y.: Factors affecting the adoption of open systems: an exploratory study. In: MIS quarterly, 1-24 (1997)
6. Dedrick, J., West, J.: An exploratory study into open source platform adoption. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 1-10 (2004)
7. Deshpande, A., Riehle, D.: The total growth of open source. In: Open Source Development, Communities and Quality, 197-209 (2008)
8. Glynn, E., Fitzgerald, B., Exton, C.: Commercial adoption of open source software: an empirical study. In: International Symposium on Empirical Software Engineering, IEEE, 1-10 (2005)
9. Fendt, O., Jaeger, M., Serrano, R. J.: Industrial Experience with Open Source Software Process Management. In: Computer Software and Applications Conference (COMPSAC), IEEE 40(2), 180-185 (2016)
10. Fink, A.: Analysis of qualitative surveys. In: The survey handbook, 61–78. SAGE Publications, California (2003)
11. Fitzgerald, B.: The transformation of open source software. In: MIS Quarterly, 587-598 (2006)
12. Guba, E. G.: Criteria for assessing the trustworthiness of naturalistic inquiries. In: Educational Technology Research and Development, 29(2), 75–91 (1981)
13. Harutyunyan, N., Bauer, A., Riehle, D.: Understanding Industry Requirements for FLOSS Governance Tools. In: IFIP International Conference on Open Source Systems, 151-167 (2018)
14. Hauge, Ø., Ayala, C., Conradi, R.: Adoption of open source software in software-intensive organizations–A systematic literature review. In: Information and Software Technology, 52(11), 1133-1154 (2010)
15. Jansen, H.: 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) (2010)
16. Kemp, R.: Towards Free/Libre Open Source Software (FLOSS) Governance in the Organization. In: IFOSS L. Rev., 1(61) (2009)
17. Lerner, J., Tirole, J.: The economics of technology sharing: Open source and beyond. In: Journal of Economic Perspectives, 19(2), 99-120 (2005)
18. Markus, M. L.: The governance of free/open source software projects: monolithic, multidimensional, or configurational?. In: Journal of Management & Governance, 11(2), 151-163 (2007)
19. Peters, S.: Best Practices for Creating an Open Source Policy. In: OpenLogic (2009)
20. Popp, K. M.: Best Practices for Commercial Use of Open Source Software. In: Business models, processes and tools for managing open source software. BoD–Books on Demand (2015)
21. Radcliffe, M., Odence, P.: The 2017 Open Source Year in Review. In: Black Duck Software, DLA Piper. (self-published presentation) (2017)
22. Riehle, D.: Lessons Learned from Using Design Patterns in Industry Projects. In: Transactions on Pattern Languages of Programming II, LNCS 6510. Springer-Verlag, 1-15 (2011)
23. Riehle, D.: The commercial open source business model. In: Value creation in e-business management (pp. 18-30). Springer, Berlin, Heidelberg (2009)
24. Riehle, D.: The economic motivation of open source software: Stakeholder perspectives. In: Computer, 40(4). (2007)
25. Riehle, D., Lempetzeder, B.: Erfolgsmethoden der Open-Source-Governance und Compliance. In: Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). (2014)
26. Runeson, P., Höst, M.: Guidelines for Conducting and Reporting Case Study Research. Empirical Software Engineering 14(2), 131–164 (2009)
27. Ruffin, C., Ebert, C.: Using open source software in product development: A primer. In: IEEE Software, 21(1), 82-86 (2004)
28. Von Krogh, G., Von Hippel, E.: The promise of research on open source software. In: Management Science, 52(7), 975-983 (2006)
29. Wang, H., Wang, C.: Open source software adoption: A status report. In: IEEE Software, 18(2), 90-95 (2001)
30. Yin R. K.: Case study research: Design and methods. Sage publications (2013)