Friedrich-Alexander-Universität Erlangen-Nürnberg
Technische Fakultät, Department Informatik

GÖZDE HAZER
MASTER THESIS

OPEN SOURCE GOVERNANCE: GETTING STARTED BEST PRACTICES FOR SOFTWARE COMPANIES

Submitted on 19.07.2018

Supervisor: Prof. Dr. Dirk Riehle, M.B.A.
Professur für Open-Source-Software
Department Informatik, Technische Fakultät
Friedrich-Alexander University Erlangen-Nürnberg

Versicherung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

_____

ERLANGEN, [DATE]

License

_____

ERLANGEN, 19.07.2018

Abstract

Open Source Software offers various advantages to software product companies. In order that companies reap the benefits of OSS, organizations should establish an OSS governance program to manage OSS license compliance issues. The introduction of OSS governance could be challenging. However, OSS governance program can be established and maintained properly by assigned teams with clearly defined policies and processes, and enhanced by means of automated tools. This thesis focuses on the main best practices for getting started of open source software governance at companies. In this paper, we conducted a literature review to build a theoretical foundation to derive the research question as well as the interview questions. We used 6 interviews as the main source for our research. Open qualitative survey method is applied to compose the research results. We formulated 25 best practices and categorized under four main categories, namely “Management, Processes, Supplementary and Tools”. We presented the findings in form of best practice patterns as a research result of this paper to be used as a handbook, which could be used at software product companies for building the initial phase of OSS governance program.

Keywords

Open Source Software (OSS), FOSS, FLOSS, Open Source governance, Open Source license compliance, OSS governance getting started, best practice, Qualitative survey method, Qualitative Data Analysis

Abbreviations

FLOSS	Free/Libre and Open Source Software
FOSS	Free Open Source Software
OSS	Open Source Software
OSRB	Open Source Review Board
OSPO	Open Source Program Office
OSEC	Open Source Executive Committee
IP	Intellectual Property

List of Figures

List of Tables

Contents

1 Introduction

1.1 Original Thesis Goals

The original thesis goal is to answer the following research question:

"What are the best practices for getting started with Open Source Governance"

In order to answer the research question, a qualitative survey research method (Jansen, 2010) was used to identify and analyse the patterns. During the analysis phase, we conducted qualitative data analysis to explore, organize and investigate the patterns from the used data sources.

We completed the following steps to achieve the thesis goal:

- Define the research question;
- Conduct the literature review broadly based upon concept-centric approach (Webster & Watson, 2002) (Please see Table 3 "Concept Matrix");
- Prepare interview questions for data collection (Please see Appendix A "Interview Questions");
- Select companies based on the theoretical sampling for data collection (Please see Appendix B "Theoretical Sampling");
- Conduct the first semi-structured interview and collect the data concurrently;
- Identify the main categories and structure the coding system (Please see Appendix C "Coding System")
- Perform coding of the data using the qualitative data analysis tool (QDAcity) to analyse the data;
- Conduct other expert interviews and collect additional data;
- Adjust the existing coding system based on data obtained from each interview and conduct further analysis;
- Identify common best practices;
- Present the results under the best practice patterns.

The purpose of this paper is to derive best practices for getting started with Open Source Governance based on the research findings of expert interviews. The research results would contribute both to science and industry by providing a best practice handbook for software companies for efficient implementation of OSS governance.

1.2 Changes to Thesis Goals

There were 2 changes to the thesis goals. Initially, we planned to conduct case study research method. However, we had to change the method to qualitative survey research due to insufficient data for case study research.

The second change was to increase the number of companies for the interviews to provide more comprehensive research results and derive more wide-ranging best practices. The initial plan was to conduct the interviews with three companies. However, we interviewed 4 companies. Additionally, my thesis supervisor shared 2 expert interviews on the same subject, which were previously conducted by himself, using similar questions as in the current study. Finally, we used total of 6 expert interviews for this study.

2 Research Chapter

2.1 Introduction

The use of Open Source Software (OSS) in commercial products has been increasing significantly over the past two decades due to the technical and business benefits for organizations. According to the survey conducted among worldwide enterprises published by Statista, the main advantages of OSS use and development are mentioned as

- being easier to develop and to deploy IT projects,
- having improved cybersecurity,
- enhanced scalability,
- better interoperability with existing applications and middleware,
- and easier compatibility to customer infrastructure (Statista, 2016).

Another survey published by Black Duck Software reveals that OSS is the foundation for almost all applications, operating systems, cloud computing, databases, big data etc. (Black Duck Software, 2016a). Currently, most of the commercial products on the market contain OSS such as engine control systems, TV screens, cell phones, navigation and entertainment systems etc. (Schöttle & Steger, 2015).

Software products may consist of different kind of software components such as open source, commercial third party and proprietary software (Popp, 2015). Using OSS in commercial products without having in-house OSS governance processes and procedures may lead to common mistakes made by companies such as unawareness of source code used in the respective product, improper documentation of the software development, source code snippets issues, missing license texts and source code (Schöttle & Steger, 2015).

These mistakes may lead to violating OSS license terms of especially strong copyleft licenses, which may result in serious consequences such as being obliged to disclose proprietary source code under OSS license or delay the release of the product until the compliance issues are solved. The relationship between companies and open source community can be damaged because of license infringement issues (Black Duck Software & Bearing Point, 2013) (Haddad, 2016).

Therefore, it is required to have certain mechanisms or procedures to manage the use of OSS. As reported by Black Duck, nearly half of surveyed companies do not have a formal policy for selecting and approving open source code. Additionally, approximately 50% of the companies, which have policies either do not have efficient practices or these policies are neglected (Black Duck Software, 2016a) (Black Duck Software, 2016a). This survey reveals that there is a major gap in dealing with the use of OSS across companies.

Companies need to deliver their products and services on time without interruption as well as establish and amplify the OSS compliance infrastructure inside the company. Thus, it is important to create the compliance infrastructure as a running change during the continuation of the business activities considering the scalability of the products and activities in the future (Haddad, 2016).

We have organized the rest of this paper in the following way: Section 2.2 presents the primary findings by comparing differences and similarities investigated while conducting the literature review and analyzing the expert interviews during the research. Section 2.3 outlines the research question briefly. Section 2.4 describes the methodology used in this research whereas Section 2.5 specifies the used data sources of the thesis results. Section 2.6 displays the research results and Section 2.7 presents the result discussion. Section 2.8 discusses limitations and conclusions. The literature review findings are shown in Section 3.1 and the complete best practice list was presented in Section 3.2. Section 3.3 provides acknowledgements.

2.2 Related Work

Based on the results of the research, four main categories are determined namely: "Management", "Processes", "Supplementary" and "Tools". All the best practices presented in this paper, are studied based on the method "Qualitative Survey Research" obtained from semi-structured interviews conducted with 6 companies. The research methodology is presented in Section 2.4 "Research Approach". 25 best practices are generated under the main categories, which are identified during the analysis phase based on the expert interviews. Each best practice addresses a specific issue and provides a solution practiced by at least one or more companies. Each best practice is presented in a separate best practice table in Section 3.2.

The findings from expert interviews are in line with the findings gathered from literature review. We aggregated strategies and policies, as well as the organizational structure subjects under the category "Management". In the "Processes" category, all processes that implemented successfully by interviewed companies are outlined. The category "Supplementary" includes the topics such as documentation and repository, as well as training and communication programs. Lastly, the category "Tools" presents several tools used by the companies to support the open source governance processes. The article written by Kemp categorizes OSS governance into 4 groups namely: the people context, the strategic context, the policy context and the process context (Kemp, 2010). Additionally, Haddad outlines the areas for establishing OSS management program as well, such as strategies, policies and processes, teams, tools, web presence, education, automation and messaging (Haddad, 2016).

Management

According to the reviewed papers, it is essential to establish a complete strategy as an initial step to determine their objectives for OSS governance and identify key principles (Kemp, 2010) (Haddad, 2016) (Popp, 2015) (Fendt et al., 2016). It is also important to decide if the product is being developed should be licensed under proprietary license or OSS license (Höst et al., 2011). Most of the interviewed companies have OSS strategies and our findings are in line with the literature.

One of the key points for companies is to have a complete OSS compliance policy outlining company principles regarding the OSS use including formal processes and internal rules (Ellis, 2011) (Popp, 2015) (Kemp, 2010) (Fendt et al., 2016) (Höst et al., 2011) (Schöttle et al., 2015).

The research results confirm the statement. However, the form of existence of the strategy and the policy documents can be different depending on the company size and structure.

In addition to the OSS use policy, companies need to embrace also policies for contribution to OSS community (Haddad, 2016) (Kemp, 2010) (Schöttle & Steger, 2015) (Open Chain, 2016) and for component reuse (German & Di Penta, 2012) (Sojer & Henkel, 2011). Our theory supports these requirements. Some of the interviewed companies, especially large enterprises (Company 2, 5, 6) have policies for contribution to OSS community to protect the company intellectual property. As for the reuse policy, companies (Company 2, 4, 6) are aware of the risks of reusing OSS components. The interviewed companies have different practices to manage the reuse of OSS components. Most of them (Company 2, 3, 4, 6) use a central repository where all the approved components as well as the used components are stored to accelerate the license check process as also mentioned in the literature (German & Di Penta, 2012) (Schöttle & Steger, 2015) (Schöttle & Steger, 2015). However, Company 4 is careful with reusing OSS components and checks all OSS components case by case for each project or product.

As mentioned in the reviewed articles, companies should assign a core team also called Open Source Review Board (OSRB) from various departments to obtain and process OSS-related information during the software development lifecycle (Haddad, 2016) (Schöttle & Steger, 2015) (Ellis, 2011) (Kemp, 2010) (Fendt et al., 2016) (Chang et al., 2010). Our theory confirms this approach. Almost all interviewed companies have an assigned team with members such as a product or project manager, an engineering manager, a legal counsel and a compliance manager.

According to Haddad, it is recommended to build an extended team to assist the compliance activities. The extended team consists of members across various departments namely: Documentation, Supply Chain, Corporate Development, IT, Localization, and Open Source Executive Committee (Haddad, 2016). However, the interviewed companies do not have extended teams for OSS governance. Therefore, our research result denies the requirement.

In order to support the OSRB with OSS governance and compliance activities, enterprises should establish an Open Source Program Office (OSPO) (Aniszczyk et al.). According to the literature, it is necessary to have a full-time employed manager with a small team to manage OSS related issues. However, the research findings do not support the statement mentioned in the literature. Interviewed companies especially large corporations have OSPO, which consists of the members of OSRB as well as developers contributing voluntarily or allocating 20% of their working time to deal with open source compliance issues.

As stated in the literature, to manage the OSS use and comply with OSS licenses, it is necessary to understand the legal requirements of OSS. Therefore, organizations need legal support to receive the feedback and the legal advice instantly regarding OSS governance and compliance issues (Haddad, 2016) (Schöttle & Steger, 2015) (Ellis, 2011) (Kemp, 2010). Legal counsel is also one of the representatives of the OSRB (Haddad, 2016) (Schöttle & Steger, 2015). Our findings support the literature. Most of the interviewed companies have in-house legal departments specialized in OSS governance and compliance related issues. Companies which do not have legal departments outsource such services by working with third-party legal experts.

Processes

Most of the reviewed articles (Chang, 2010) (Black Duck Software) (Ellis, 2011) (Fendt et al., 2016) (Haddad, 2016) (Kemp, 2010) (Popp, 2015), illustrate the OS compliance processes with similar phases. According to Popp, there are 6 process phases for successful OS governance, namely:

- Choose,
- Scan,
- Approve,
- Inventory,
- Secure,
- Deliver (Popp, 2015).

The findings of our paper for processes are consistent with the paper written by Popp. According to our results, there are 8 best practices for getting started with OSS governance namely:

- Build an OSS component selection process,
- Integrate OSS into Supply Chain management,
- Discover all the OSS used in the product,
- Establish a process for auditing source code,
- Establish a review process,
- Establish a component approval process,
- Establish a documentation process,
- Ensure license compliance for outgoing products.

Yet, there can occasionally be minor differences in processes in practice. For example as for auditing source code process, the interviewed companies conduct random audits instead of periodically scheduled audits which was recommended by the reviewed articles (Ellis, 2011) (Haddad, 2016) (Schöttle & Steger, 2015).

Supplementary

All the reviewed articles emphasize that the complete documentation is one of the most essential parts of OSS governance. Additionally, it is crucial to create and maintain a bill of materials for each product to review and identify the type of OSS used in each product (Haddad, 2016) (Schöttle & Steger, 2015) (Open Chain, 2016). Our theory supports and captures these requirements.

Creating an approved license list helps companies to facilitate the license compliance process (Lindman et al., 2010) (Ellis, 2011). According to the literature, it is crucial to build a repository for approved components, which can be used and reused (Sojer & Henkel, 2011) (German & Di Penta, 2012) (Kemp, 2010) (Schöttle & Steger, 2015) (German & Di Penta, 2012) (Fendt et al., 2016). Our research findings support these statements.

It is crucial to provide trainings to employees in order to increase the awareness of the OSS policies and strategies, and create a common understanding on OSS licensing issues and risks as well as

improve developers abilities (Fendt et al., 2016) (Haddad, 2016) (Silberman, 2014) (Sojer & Henkel, 2011) (Ellis, 2011) (Open Chain, 2016) (Kemp, 2010) (Chang et al., 2010). Our theory supports these approaches provided by literature with a minor difference. All interviewed companies which conduct trainings, provide only informal trainings to their employees.

In order to mandate third-party suppliers to provide all necessary information concurrently, it is beneficial to formulate a contract with suppliers (Haddad, 2016) (Sojer & Henkel, 2011) (Open Chain, 2016). Most of the interviewed companies (Company 2, 4, 5, 6) include all necessary OSS-related terms into contracts. Furthermore, it is important to build an internal communication system to provide OSS-related information to employees (Haddad, 2016). Our findings support and capture these requirements.

In order to assist the review process it is necessary to build a checklist to warrant that all points are checked during the process (Kemp, 2010). Our research findings are in line with the results of the reviewed literature.

Tools

As stated in the literature, various tools with different features are recommended to support the OSS governance processes such as Discovery and Selection tools, Compliance tools, Dependency checkers, Code Sanitation tool, Security Vulnerability Remediation tools, Supply Chain tools (Popp, 2015) (Haddad, 2016). It becomes complicated to follow communities and receive version update notices for the numerous number of OSS components (Haddad, 2016). Organizations should choose the required OSS tools to handle the use of OSS (Schöttle & Steger, 2015). Our research findings confirm that the component management tools that support SPDX format are used to uphold OSS compliance in a standardized way. Additionally, tools such as Fossology, Black Duck etc. are utilized to scan the source code to meet the license obligations. However, the tools such as Dependency checkers, Code Sanitation, Security Vulnerability Remediation were not mentioned during the interviews.

2.3 Research Question

The following research question was investigated in this thesis:

“What are the best practices for getting started with open source governance for software companies?”

In this research paper, we studied how the organizations have been adapting OSS in their products focusing on the recent implementations of the interviewed companies. In order to answer the research question, we identified the main categories of the best practices generated during the comprehensive literature review as well as the expert interviews. According to our findings, there are four main categories namely: Management, Processes, Tools and, Supplementary.

The aim of the research is to identify getting started best practices, outline the fundamental patterns and present them as a handbook for companies which use or plan to use OSS in their commercial products or services.

2.4 Research Approach

In this paper, we conducted a literature search of books, articles and whitepapers of different set of publications in the research area. There are various resources on OSS available in digital databases such as ACM, IEEE, Springer, EBSCO, ABI Inform as well as on Google Scholar. Initially, the research scope was defined. Open source community related topics are determined out of the research scope.

During the research, we have selected 20 papers for the literature review. The concept-centric approach introduced by Webster and Watson is used to analyze and classify the researched information (Webster & Watson, 2002). Referring to the concept matrix table, similar concepts were identified and categorized (Please see Table 3 "Concept Matrix"). As a result of the literature review, the interview questions were prepared beforehand to conduct the interviews more effectively (Please see Appendix A "Interview Questions").

We used the qualitative survey research approach developed by Jansen (Jansen, 2010) to identify the best practices obtained from the semi-structured interviews conducted with 6 companies. The qualitative survey is open (inductive) as the interview transcriptions are main sources for this research to identify relevant topics, different aspects of objects and categories. We followed 4 activities to complete the research namely: Defining knowledge aims, Sampling, Data collection, and Analysis. As an initial step, the research question was determined to define the knowledge aims (Please see the Chapter 2.3). In the sampling stage, theoretical sampling for expert interviews was produced to select the companies with intended characteristics. We considered different metrics for theoretical sampling namely: business model, market position, type of customer, maturity of products and companies, size and market capitalization (Please see Appendix B "Theoretical Sampling").

In the next phase, data collection was performed. The semi-structured expert interviews with the identified companies were the main sources for the data collection. The expert interviews were conducted mostly online with the determined companies using pre-prepared interview questions. The interview questions were structured during the literature review (Please see the Appendix A "Interview Questions"). Each conversation was recorded during the interview and then transcriptions were produced based on the recordings.

The analysis phase consists of 3 levels. In the first-level analysis, the main categories were identified and subsequently the coding system was produced (Please also see Appendix C: "Coding System").

Second-level analysis, the case oriented empirical synthesis was performed by grouping cases on the basis of corresponding combinations of characteristics into several types (Jansen, 2010). In this

level, the qualitative data analysis was conducted using a specific tool namely "QDAcity" to identify, organize and examine the data. The tool helped us to code the data in a structured way. The data for each interview were studied as they became available. In the first stage, categories were selected and then the empirical category combinations were analysed and interpreted in the second stage and lastly the category combinations were selected and labelled.

In the final stage, we generated our findings based on the results of qualitative data analysis. We presented each finding as a best practice in a separate table in the result section (Please see Chapter 3.2). Each best practice table includes the following sections: Name of the best practice, Actor/s, Context, Problem, Solution and References. By means of the best practice tables we could define the particular problem and provide the solution based on our findings in this research.

2.5 Used Data Sources

During the data collection phase, we conducted semi-structured interviews with determined 6 companies based on the theoretical sampling (Please see Appendix B "Theoretical Sampling"). The conversations during the interviews were recorded and transcribed. Below, we described the structure and business model of each interviewed company. The interviewed companies were promised anonymity. Therefore, we will keep the companies and interviewees name confidential and not disclose any company information. We interviewed experts with different roles and responsibilities, which enabled us to learn different aspects of OSS governance.

Company 1:

The first company is a small-sized company in the growth maturity level having 5 offices worldwide. We have conducted the interview online with the Director of OSS. Their OSS strategy is "Building OSS" according to Popp's approach (Popp, 2015). The company commercializes OSS for a direct return. They provide complete communication platforms such as groupware, email, chats, video conferencing etc. to the enterprise customers. According to the article presented by Ellis, the software of the company is in the low-risk software category (Ellis, 2011).

Company 2:

It is a multinational corporation and technology company headquartered in U.S. The interview was conducted online with the senior OSS compliance engineer of the company. OSS strategy of the company is "Building with OSS" providing core functionality by using OSS for its products (Popp, 2015). Their products are in the developed maturity level and leader in the market. The software of the company is in the high-risk software category as the respective software cannot be removed from the product after the shipment (Ellis, 2011). The company uses OSS extensively in their product development and have the established OSS governance and compliance best practices.

Company 3:

The third one is a global IT-consulting company. The company is a large enterprise and the leader of the market. The interview was conducted with the IT consultant of the company who is

responsible of providing most of the information on best practices to the customers which are in the initial phases of OSS governance.

Company 4:

The company is a large enterprise developing embedded systems, such as automotive infotainment systems and digital dashboards to its enterprise customers. The interview was conducted face-to-face with the team manager and the senior legal counsel personal. The company's OSS strategy is "Building with OSS" (Popp, 2015) and its software is in the high-risk software category (Ellis, 2011). They use OSS broadly to develop main functionality for their products.

Company 5:

The company is a medium-sized company developing software and services all as open source and delivering open-source stack for messaging, collaboration and productivity for the service-provider industry. Their OS strategy is "Building OSS" (Popp, 2015) and the software of the company is in the low-risk software category (Ellis, 2011).

Company 6:

The company is a large enterprise and leader in the market developing innovative software for vehicles. The interview was conducted with open source compliance manager. Their software is in the high-risk software category (Ellis, 2011) and its OSS strategy is "Building with OSS" as they use OSS for developing main functionality of their products (Popp, 2015).

2.6 Research Results

As a result of this research paper 25 best practices are derived under four main categories namely:

1) Management,
2) Processes,
3) Supplementary,
4) and, Tools

A complete set of best practice tables were created which can be achieved in form of best practice patterns in Chapter 3.2. Also, the best practice list is presented in the Table 1.

The category "Management" highlights the best practices of establishing OSS compliance strategy and policies as well as defining roles and responsibilities of the compliance teams. The category "Processes" refers to 8 essential processes for establishing successful OSS governance: Component selection, Supply Chain management, Component identification, Auditing source code, Component review, Component approval, Documentation, Distribution. "Supplementary" category outlines all necessary elements for managing the OSS governance processes such as documentation, repository, employee trainings etc. Category "Tools" suggests the necessary tools to support the processes (Please see Table 1: The List of Best Practices).

Category	Best Practice	Actors
Management	1a Establish an open source compliance strategy	Top Management
	1b Establish an open source compliance policy	OSRB
	1c Establish a reuse policy	OSS Compliance Manager (OSCM); Legal Counsel (LC)
	1d Create a policy on how to contribute back to the OS community	OSCM; LC
	1e Define the responsibilities of the OSRB	OSRB
	1f Establish an Open Source Program Office (OSPO)	OSRB, Developers
	1g Structure the in-house legal department	Top Management
Processes	2a Build an OSS component selection process	Engineering manager, Developers
	2b Integrate OSS into Supply Chain management	Supply Chain team, LC, OSCM
	2c Discover all the OSS used in the product	OSCM, LC, Developer
	2d Establish a process for auditing source code	OSRB, Quality engineer
	2e Establish a review process for software components	OSRB, Developer
	2f Establish a component approval process	OSRB, LC
	2g Establish a documentation process	OSRB
	2h Ensure license compliance for outgoing products	OSRB
Supplementary	3a Create a complete documentation	OSCM
	3b Create a bill of materials	OSCM
	3c Create an approved list for licenses	OSRB, LC
	3d Establish a centralized repository for OSS implementation	OSRB, Engineering Manager
	3e Establish an employee training program	OSCM, LC
	3f Build an internal communication system	OSRB, OSCM
	3g Formulate a contract with suppliers	OSCM, LC, Supply Chain Team
	3h Build a checklist for software component review	Engineering manager, Developers
Tools	4a Use a component management tool	OSCM
	4b Use an OS compliance checking tool	Engineering manager, Developers

Table 1. The List of Best Practices

Each best practice pattern consists of 6 subdivisions: Name, Actor/s, Context, Problem, Solution and References. The pattern is demonstrated in Table 2 to show how a best practice is presented. Table 2 illustrates "1a Establish an open source compliance strategy" which is the first best practice in the category "Management".

Name	1a Establish an open source compliance strategy
Actor	Top management
Context	Initially it is critical that the executive board is fully involved with a consensus in building the OS governance strategy. Risks and benefits of OSS use need to be considered by top management. It is very important that the decisions made by top management regarding OSS use are shared with company employees.
Problem	How to establish principles for the OSS governance? How to inform employees about the purpose of the OSS use?
Solution	OSS strategy allows the company management to define the potential risks of OSS use and the internal compromises. The company management should take business, legal and technical decisions on establishing the strategy. It is necessary to define the company's OSS objectives by explaining OSS compliance within the company. By means of the OSS strategy, employees can understand the purpose of the OSS, its practice areas and the areas to avoid.
Sources	Literature: (Kemp, 2010) (Popp, 2015) (Haddad, 2016). Interviewee: Company 1, Company 3, Company 5, Company 6.

Table 2. The Best Practice Pattern

2.7 Results Discussion

All the best practices are categorized into 4 main categories and presented in form of best practice patterns. The best practice patterns interrelate with each other. In order to establish successful OSS governance process at a software product company, it is necessary to identify the sequence of the workflow. Please see the workflow diagram in Figure 1, which is a simple representation of getting started with OSS governance at software product companies.

The first step in getting started with OSS governance is to establish the compliance strategy. The strategy should be defined by the top management (Please see the best practice "1a Establish an open source compliance strategy") to inform employees on aims of OSS governance. Especially large enterprises have a documented OSS compliance policy shared with their employees (Company 3, 6).

In order to establish and manage the processes for effective OSS governance, it is essential to have a core team (Please see "1e Define the responsibilities of the OSRB" and "1f Establish an Open Source Program Office"). The core team usually called Open Source Review Board (OSRB). OSRB is responsible for establishing and maintaining OSS processes as well as policies, reviewing and approving software components, and ensuring OSS license compliance. Almost all interviewed industry partners have an established OSRB with a small group of people. The team usually consists of the members namely product/project team manager, engineering manager, legal counsel and the compliance manager. According to the reviewed literature, Open Source Program

Office (OSPO) is a similar concept to OSRB. Based on the interviews, we used OSPO as an extension of OSRB as the participants of OSPO included also developers in addition to OSRB. Among interviewed companies, only large corporations have an in-house OSPO (Company 2, 6).

The next step is to structure the legal department (Please see "1g Structure the in-house legal department"). It is crucial to understand the legal requirements for OSS license compliance. Therefore, organizations need to receive legal advice on OSS governance and compliance. It is a business decision that should be made by top management to either structure in-house legal department or to outsource legal services (1g).

The decision for creating an approved license list (3c) should be made considering legal and business aspects. It is crucial to identify the type of open source licenses that will be used including the defined use-cases to manage multiple licensing issues and guide the developers. For example, one license might be in the approved list but allowed to be used only in one use-case.

Upon creation of the approved license list, the following steps can be realized concurrently: "3d Establish a centralized repository", "2 Establish processes" and "4a, 4b Embrace tools".

"3d Establish a centralized repository" is one of the next steps before establishing the OS governance policies. The repository enables companies to store all the approved as well as audited OSS components, to reuse the components, and to track the use of OSS.

Another step is to establish OSS governance processes (Please see Figure 2. OSS Governance business process diagram). It is crucial to discover all the OSS used in the product to comply with OSS license liabilities (Please see the "2c Discover all the OSS used in the product").

This process should be maintained by means of an automated tool such as Fossology, Black Duck (Company 2, 3, 6) (Please see "4b Use an OSS compliance checking tool"). To assist this process, organizations should create and maintain a complete BoM to identify the type of OSS used in the product easily (Please see "3b Create a bill of materials").

Organizations should have a component selection process to avoid OSS license infringements (Please see "2a Build an OSS component selection process"). According to the interviewed companies (Company 1, 4, 5, 6), it is necessary to consider several parameters to choose the OSS community before selecting the component e.g. diverse community with a fast reaction on bug reports etc. Sophisticated, most convenient, widely used, well-maintained and functional OSS codes should be preferred. Certainly, while building the selection process, there are different aspects that organizations should consider according to the company requirements and their usage to handle the quality, security, compatibility and maintainability issues. The selection should be made considering the approved licenses in the whitelist (Please see "3c Create an approved list for licenses"). According to the findings based on the interviews (Company 1, 3, 4, 5, 6), organizations also need to set up a centralized repository (Please see the best practice "3d Establish a centralized repository") and a component approval process (Please see "2f Establish a component approval process") to establish a successful component selection process.

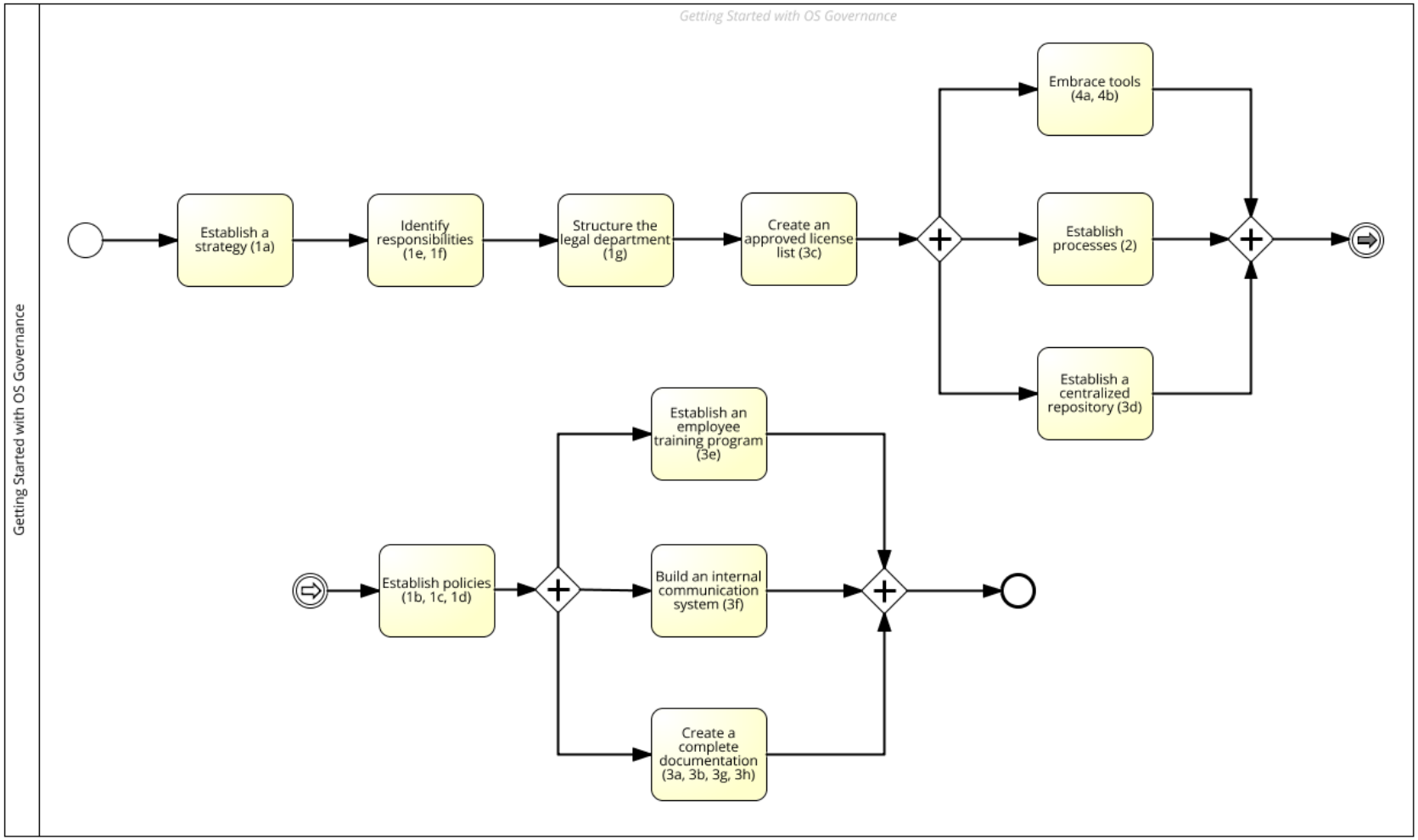
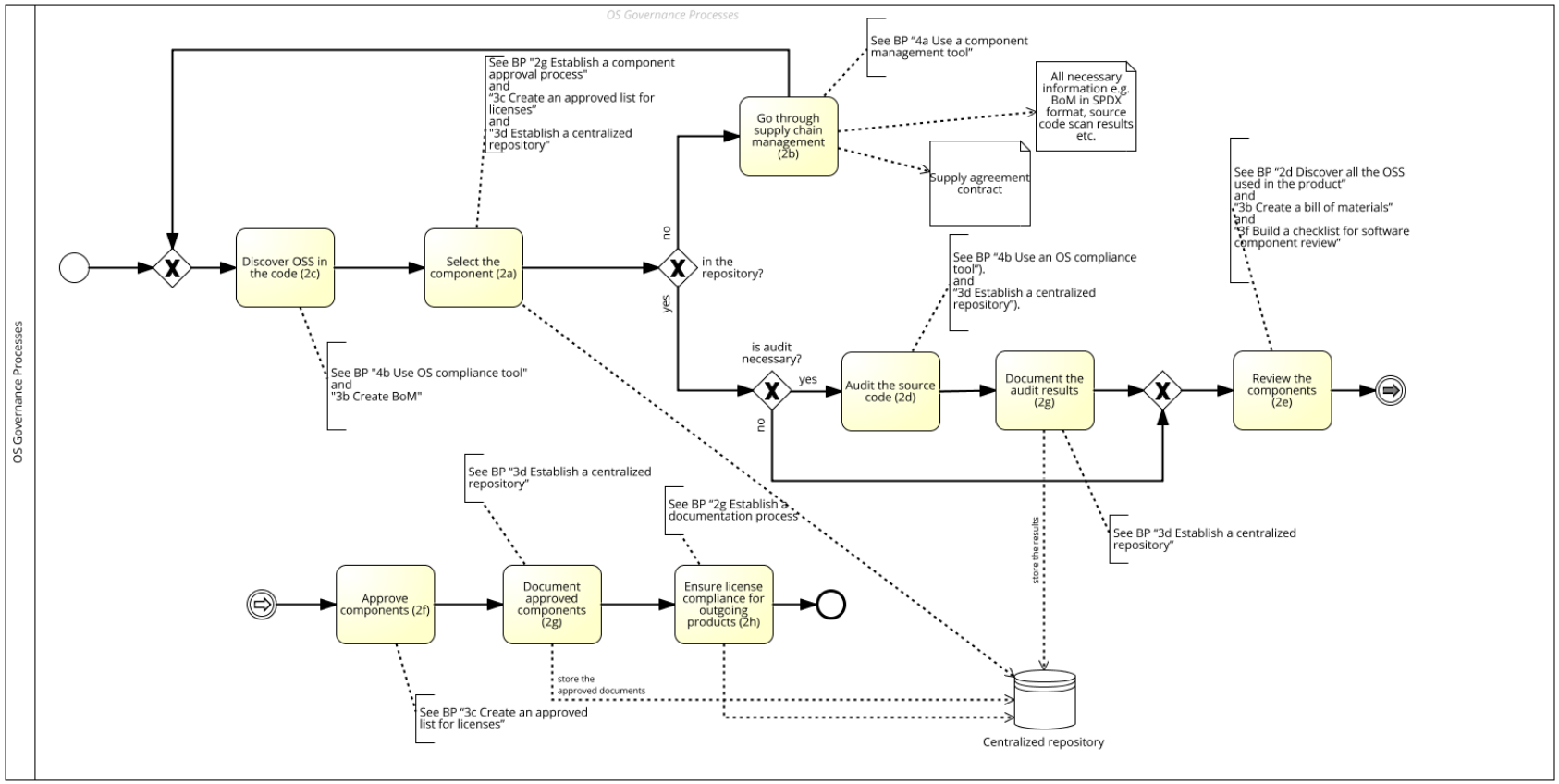


Figure 1. Getting Started with OSS Governance workflow



* BP – Best Practice

Figure 2. OSS Governance business process diagram

It is essential to integrate OSS into supply chain management (2b). This process should be improved by means of a tool (Please see "4a Use a component management tool") due to the variety of OSS components and the follow-ups for all necessary requirements of each component. Mature companies among the interviewed enterprises (Company 2, 4, 6) use a tool that supports SPDX format. In order to ensure that all the necessary information and documents are provided subsequently and the obligations regarding the distributed product are fulfilled, organizations should mandate their suppliers to conclude a contract (Please see "3g Formulate a contract with suppliers"). Interviewed companies (Company 2, 4 and 6) have supply agreement contract with their suppliers including all necessary purchasing terms and deliverable software documentation.

The next step in the governance processes is to establish a process for auditing source code to mitigate IP-bleeding risks. The audit process is performed not periodically but randomly. Therefore, we drew it in the process diagram as an optional process (Please see Figure 2. OSS Governance business process diagram). Organizations should embrace a source code scanning tool to facilitate the process (Please see "4b Use an OSS compliance tool"). Another critical aspect for audits revealed from the interviews (Company 2, 3, 4, 6) is to store all the audit results of components in the repository (Please see "3d Establish a centralized repository" and "2g Establish a documentation process").

In the review process, third-party software and open source software can be treated differently or similarly depending on the technical decision of the company. In order to establish a review process for software components, it is necessary to know what kind of OSS they use as well as to have a complete bill of materials (Please see "2c Discover all the OSS used in the product" and "3b Create a bill of materials"). This process can be enhanced by using a checklist to ensure that all aspects are covered during the review process (Please see "3h Build a checklist for software component review"). Almost all interviewed companies have an established review process (Company 1, 2, 3, 4, 6). However, only mature companies use checklists for the review process (Company 2, 4, 6).

After the review process is completed successfully, the approval process should be performed (Please see "2f Establish a component approval process"). Mature enterprises have the approval process (Company 2, 4, 6). Once the product or project is approved, it should be stored in the centralized repository to track and monitor the used OSS (2g Establish a documentation process). Most of the interviewed companies have their own repository to keep all necessary data regarding the software product (Company 2, 3, 4, 6).

The last process is "2h ensure license compliance for outgoing products". Once the approval process is finished it is necessary to include all necessary notices in the product. The legal department should prepare the disclaimer including copyrights and licenses found in the source code. However, the interviewed companies do not use an automated tool for outbound compliance.

To support and automate the OSS governance processes, it is important to use necessary tools. Therefore, it is important to embrace necessary tools in parallel with processes. According to the interviewed companies, there are 2 different tools: the first tool is "4a Use a component

management tool" that supports SPDX format to standardize the documentation. The second one is "4b Use an OS compliance checking tool" enabling companies to scan the code automatically.

After completing these steps (2, 3d, 4a, 4b) the next step is to "establish an OSS compliance policy (1b)" to inform which licenses and packages are acceptable for use in products (3c Create an approved list for licenses), clarify governance processes (all best practices under "Processes") and provide information regarding roles and responsibilities for OSS related issues ("1e Define the responsibilities of the OSRB", "1f Establish an Open Source Program Office (OSPO)" and "1g Structure the in-house legal department"). Each interviewed company has its own policy model. For instance, the Company 2 publishes policies on wiki-like or blog-like platforms in small pieces of information to minimize search time for employees and keep the previous replies not to spend time on answering the same question. Whereas, Company 6 has a companywide policy that explains the intention OSS use and all the steps to be followed including the legal information and license assessments in a single document. In addition to the general policy, each department has a detailed but separate policy including complete implementation of the processes. Moreover, employees should receive training to understand the principles of the OSS policy. Therefore, we referred to the best practice "3e Establish an employee training program".

Another central aspect for setting up OSS governance is to "establish a reuse policy (1c)". The policy should be defined by the OSS compliance manager and the legal counsel. Among the interviewed companies, the mature enterprises (Company 2, 6) have a reuse policy to mitigate the risks that could arise from OSS license incompatibility. Additionally, organizations need to have a centralized repository to track OSS components for appropriate reuse (Please see "3d Establish a centralized repository").

As aforementioned, OSS community related topics are determined out of the research scope. However, we generated the best practice "1d Create a policy on how to contribute back to the OS community" in term of a policy aspect. Almost all interviewed companies contribute back to the community. However, only 3 companies (Company 2, 5, 6) have a policy to protect the company IP by checking the contribution in terms of legal and technical perspectives.

The following steps can be established parallelly as well: "3e Establish an employee training program", "3f Build an internal communication system", and "Create a complete documentation (3a, 3b, 3g, 3h)". Employees should receive training to have a common understanding of OSS policies and licensing issues as well as the technical aspects (Please see "3e Establish an employee training program"). The interviewed companies do not provide instructor-led courses to their staff but conduct web-trainings or company-wide talks to different employee groups depending on their company structure.

One of the next steps is "Build an internal communication system (3f)" to share all the necessary information and documents with the respective employee. It is a challenge for global companies having offices throughout the world to share the best practices across the company.

In order to maintain OSS governance processes successfully, organizations need to create comprehensive documentation, store them in the repository and share with the employees. The documentation phase includes several best practices ("3a Create a complete documentation", "3b

Create a bill of materials", "3g Formulate a contract with suppliers", and "3h Build a checklist for software component review"). The complete documentation refers to all OSS policies, training materials, all the assessed license lists that should be generated and shared with the team. Additionally, BoM should be documented in the company repository. Interviewed companies (Company 2, 4) generate BoM for each product to identify software used. All the assessed OSS components should be stored. This enables companies to use and reuse the components as well as report and track the use of OSS.

Moreover, "3g Formulate a contract with suppliers" allows companies to ensure that their suppliers fulfil the OSS license obligations and compliance for the delivered products. The interviewed companies (Company 2, 4, 5, 6) mandate their suppliers to provide all necessary information regarding OSS, which is mentioned in the supply agreement contracts. "3h Build a checklist for software component review" improves the review process. By means of a checklist, OSRB ensures that all the aspects are covered and checked during the review process.

2.8 Limitations and Conclusions

Although the research has achieved its goal, there were some limitations. The limited number of interviewed companies for data collection of the qualitative survey research was the first limitation. Another limitation was the limited time availability of the industry partners, which allowed only approximately one hour for each interview. More interviews from different industries are required to generalize the findings and obtain further best practices.

This paper provides a handbook that includes 25 best practices for getting started with OSS governance for software product companies. These best practices are derived from the interviews performed with 6 industrial partners which have advanced OSS governance practices. In order to develop a theory for getting started with OSS governance, we conducted literature review and used qualitative survey research as a research methodology.

The study of OSS community governance is out of the scope of this thesis. Further research can focus on this concept to examine the requirements for OSS community governance.

3 Elaboration Chapter

3.1 Literature Review

Software Licenses

Proprietary licenses grant users a freedom only to use single copy of the product. The software is preserved under license terms accepted by the end users which do not grant them access to the source code (Muffatto, 2006).

Open Source Software (OSS) is software available for free, either sharing or restricting the rights under the OSS license terms. OSS gives users the freedom to use, modify and distribute the software and redistribute derivative works (Muffatto, 2006).

OSS licenses are divided into two main categories: Restrictive licenses (copyleft or reciprocal) and Permissive licenses (non-copyleft). Restrictive Licenses are also classified under 2 categories namely weak copyleft and strong copyleft. Weak copyleft licenses are LGPL, MPL, Mozilla PL, Ms-PL. If the source codes are combined, the newly derived work should be relicensed in case there was no previous license, which ties the licensee to its terms. The newly combined work should be released under the same copyleft license even if it was a relatively weak one such as LGPL or MPL (Yu, 2013).

Strong Copyleft such as GPL and AGPL do not allow users to distribute the open source code under a proprietary license. The software as a derivative or combined work should sustain the license terms of the original software under a strong copyleft license. Specifically, the software should provide the same freedom as the original one to access to the source code, use, copy, modify and distribute it (Yu, 2013).

Permissive licenses such as BSD, Apache, X11/MIT (Copenhaver et al., 2013) (Lindman et al., 2010) allow both licensor and licensee to use the software for commercial purposes. Developers are able to produce a proprietary product using permissive licenses, which allow the source code distribution for derivative works (Lindman et al., 2010).

Restrictive licenses have become less common in use for open source projects over the last years (Black Duck Software, 2016b). According to the data recently provided by BlackDuck, permissive licenses (%52) are utilized more frequently than restrictive licenses in OS projects. The most common open source licenses are MIT, GPL 2.0, Apache 2.0, GPL 3.0, BSD and their rankings are %32, %18, %14, %7, and %6 respectively (Black Duck Software, 2018).

According to Lindman et al., the license selection should be made by considering different aspects such as business model, patenting, motivation creation, leadership, externalities, company size in addition to knowing the different licenses and their requirements (Lindman et al., 2010).

The interviewed companies use different kind of OSS licenses for different use-cases.

Open Source Software compliance failures

OSS compliance failures may occur during software development lifecycle due to mistakes and constraints in the processes such as failure to deliver a proper attribution notice, neglection to provide a license notice, omission of a copyright notice, failure to deliver the source code or a modification notice etc. There are three main compliance failure types namely intellectual property failures, process compliance failures and license compliance problems (Haddad, 2016).

The intellectual property failures may force companies to release the proprietary source code under an open source license. Such failures may result in losing intellectual property rights (such as delaying the shipment until the compliance issue is resolved, distributing the source code under OSS license, losing time for re-engineering, damaging the brand/company reputation) as well as losing the ability to differentiate the product in the market (Haddad, 2016).

Our findings confirm that organizations should embrace OS governance processes (Please see Section 3.2.2).

OSS Selection, License Compliance and License Compatibility

Violation of OSS license terms poses a serious risk for companies to lose the copyrights. Such companies may renounce the benefits of the respective work (Laurent, 2004).

Due to various benefits provided by OS, companies started to combine OSS with their commercial software, correspondingly a multi-source development model is developed (Haddad, 2016). According to Haddad, the best practice for companies is to have a well-defined and stable compliance program with active incorporation of a multi-source development model. By means of such compliance program, companies commonly gain a technical advantage, as compliant software portfolios are easier to service, test, upgrade and maintain. Moreover, such compliance programs can also allow organizations to identify crucial pieces of Open Source that are in use across multiple products and parts of an organization (Haddad, 2016).

Additionally, code-snippets should also comply with the respective OSS license of the component (Schöttle & Steger, 2015). Scanning the software product is a key to ensure compliance with the license obligations in order to avoid the risks of license non-compliance including third-party components, nested components, and code snippets issues (Schöttle & Steger, 2015). Therefore, it is highly important to perform binary and source code scans for all software in the products (Ellis, 2011).

Another issue arises from multiple licensing. Usually, developers need to use different programs under different OS licenses to develop a work that may cause a violation of one of the licenses. Therefore, it is crucial to analyse the works licensed under different type of licenses (Laurent, 2004).

Every license has its own rules and regulations which set specific boundaries with users clarifying what is tolerated related to use of the software (Lindman et al., 2010). Therefore, it is highly important to decide which open source licenses will be used by the company considering the license obligations.

Our research results cover these issues (Please see best practices “3c Create an approved list for licenses” and “2a Build an OSS component selection process”).

Business Models

It is important to establish business strategies to improve returns on investment (ROI) in open source (Popp, 2015). There are four main OSS business models. The first one is “Building OSS” where an enterprise develops and distributes OSS for direct return such as establishing platforms, middleware or applications for smart gadgets, computers data centres etc. as open source. “Building with OSS” is the second model that original equipment manufacturers (OEMs) and software vendors implement OSS for main capabilities of their own products or services such as developing network tools with embedded Linux and smart devices with Android. The third model is “Building for OSS” where companies provide services such as employee training, governance tools, documentation and different type of support for OSS. Some examples can be listed as Black Duck Software, Red Hat, etc. The last model is “Building on OSS”, which means a company runs the business with OSS for its strategic operations such as CRM, accounting, engineering, marketing etc. (Popp, 2015).

The most commonly used business model among the interviewed companies (Company 2, 3, 4, 6) is “Building with OSS”.

Software Risk Categorization

The initial step of the OSS compliance for organizations requires analysis of risks and requirements of OSS use. According to John Ellis, it is important to practice the risk categorization system in OSS governance program divided into 3 parts namely High – Medium – Low. Every company uses different software for different purposes and has different implementation. If the functionality provided by the software is compulsory for the product to operate and the respective software cannot be removed from the product, it is in the high-risk software category. Companies use the software in this category, should require the source code scan results rather than binary from their software vendors to achieve the full data of the software code and licenses to mitigate compliance risks. In this category, OEM companies have the highest risk as they cannot take the software out of the product after the product is launched (Ellis, 2011).

If the software is a highly important for the product but it can be changed by an identical software that is in the medium-risk software category. Therefore, software vendor may provide source code or binary code when they are providing software products to companies in this category. OEM companies may replace the software and do not have to stop the product shipment therefore the risk for OEM products is medium (Ellis, 2011).

The software is in the low-risk software category, if the company installs the software on the product after the product shipment such as applications from app-stores of Apple or Android and downloadable add-ons. The risk for OEM products is lowest in this category. There is a risk if the software hosted by the OEM as the OEM is interpreted as a distributor. Therefore, additional liabilities under OSS licenses emerge for OEMs. Additionally, the OEM should conceive the

content of the software if the company makes recommendations on the respective software (Ellis, 2011).

Our findings are in line with the literature. The interviewed companies (Company 2, 4, 6) which are in high-risk software category ask their vendors to provide source code scan results instead of binary.

Open Source Strategy

It is fundamental for enterprises to determine their objectives for OSS compliance and governance, and identify key principles. OSS governance objectives should be adapted by all companies particularly concerning the high-level approach and strategies, including daily operational processes (Kemp, 2010).

It is a strategic decision that the company should make, if the product is going to be developed and distributed under proprietary license or open source license (Höst et al., 2011).

According to Kemp, organizations should identify their complete strategy in accordance with other units such as risk management, IP strategy etc. about the purpose of the OSS, its practice areas and the areas to avoid. All stakeholders are able to define the internal compromise by means of the OSS strategy statement (Kemp, 2010).

The OSS Strategy Statement consists of 5 outlines namely defining company's OSS objectives, explaining OSS compliance, clarifying OSS governance within the company, informing about the mixed software environment (using OSS, and proprietary software), and further details (Kemp, 2010).

Additionally, Haddad examines the strategy based on 2 aspects namely compliance strategy and inquiry response strategy. Haddad emphasises that the strategy should define an established process for the approval, supply chain, and open source use, and a practice for releasing software that includes open source or that is licensed under an open source license (Haddad, 2016).

It is also important to have a process mentioned in the strategy to deal with OSS compliance inquires when the company receives negative feedback or negative notices. This process should define how to handle with incoming inquiries, confirm the receipt of an inquiry, provide an information on a realistic response date (Haddad, 2016).

Our findings confirm these approaches (Please see the best practice "1a Establish an open source compliance strategy").

Open Source Policy and Guidance

According to Statista, in 2017, only 37% of the surveyed companies have OSS acquisition and usage policy in their organizations (Statista, 2017b). There might be some handicaps for organizations who do not have any OSS policies.

According to Popp, organizations should consider 2 factors which bring the success in OSS governance application. The first one is that the integration of the procedures with the current software development processes at the company should be done appropriately. The second factor

is that these procedures need to provide maximum efficiency to mitigate delays and expenses. These can be achieved by means of the automation of OSS governance procedures (Popp, 2015).

Organizations should have a published policy clarifying which licenses and packages are acceptable for use in products including license combinations considering the software risk category (Ellis, 2011). It should be "clear and brief" (Kemp, 2010), maximum 5 pages (Popp, 2015). Other characteristics of the OSS policy are event-driven", "establishing criteria and decision points for OSS use" and "presenting the information to be collected and tracked" (Kemp, 2010).

OSS governance policy should also include acceptable and non-acceptable outcomes according to the proprietary licenses such as the type of representation, assurance, indemnity in case of infringement (Ellis, 2011).

The OSS policy proposed by Richard Kemp consists of 3 main categories. The first one is "Scope and rationale", the second one is "Roles, responsibilities, training, and awareness", and the last one is "OSS Policy for inbound transactions (OSS procurement policies, OSS in Mergers & Acquisitions), in-house development (describing the authorization mechanism, OSS license approval, policy on contribution to OSS projects) and outbound transactions (template agreement for sales and make assigned people available to discuss on outbound transaction issues)" (Kemp, 2010).

Our research findings are in line with the literature. (Please see best practices "1b Establish an open source compliance policy", "1c Establish a reuse policy" and "1d Create a policy on how to contribute back to the OS community")

According to Popp, large organizations usually split OSS policy into 2 separate documents: "A simple, high-level policy statement suitable for compliance verification", and "a guidelines document containing the more detailed rules for use and management of OSS" (Popp, 2015). Our theory supports this approach which is reflected in the best practice "3f Build an internal communication system" in Section 3.2.

Track & Reuse OSS & Repository

OSS Management Database is a very important system to collect and provide necessary information about OSS components and their versions, related licenses and their correlation and historical information. The database stores the date when the copy of the code in the OSS component is made and respective license was gained. By means of the tool, it is possible to achieve the bill of materials of the OSS for each product and respective licenses. Additionally, this approach helps users identify OSS license incompatibility issues (Schöttle & Steger, 2015).

It is highly necessary to produce a repository of components pre-approved for reuse (Sojer & Henkel, 2011) (German & Di Penta, 2012). Additionally, defining procedures is important to document and verify how reused components are being incorporated into products (German & Di Penta, 2012). As German et al. mentioned in their article, all organizations should follow reuse policies by retaining legal advice and creating policies regarding the reuse of open source.

Legislators should assign a responsibility to an Open Source manager to solve the potential issues who inspect the reuse of open source within the company (German & Di Penta, 2012).

Our findings support the requirements mentioned in the reviewed literature (Please see the best practices "2g Establish a documentation process", "1c Establish a reuse policy" and "3d Establish a centralized repository for OSS implementation").

Identifying Responsibilities

According to the survey published by Statista, only 12% of the companies who responded have an Open Source Review Board (OSRB). 28% of the surveyed companies say that their engineering team is responsible for OSS and IP compliance. 12% of the companies have a legal team who is responsible for OSS and IP compliance (Statista, 2017a).

For successful OSS governance, organizations need to assign a core team from various departments to gather and process OSS-related information during the software development lifecycle (Schöttle & Steger, 2015) (Chang et al., 2010) (Haddad, 2016). Therefore, it is crucial to build a core team called an open source review board that consists of members from legal department, business department, product development department (Haddad, 2016) (Schöttle & Steger, 2015) (Ellis, 2011), quality management, supply chain management (Schöttle & Steger, 2015) and compliance officer (Haddad, 2016). OSS Core Team reviews all OSS components, scan reports for the products, questionnaires and OSS licenses used in the OSS portfolio (Schöttle & Steger, 2015) ensures compliance with third-party software as well as OSS (Haddad, 2016) and determines the approval (Ellis, 2011). The team also coordinates the contribution to the OSS community (Schöttle & Steger, 2015).

As mentioned in the article written by Schöttle et al., OSS Agent is responsible for gathering all information about OSS from third-party developers and other vendors for the respective project, and submitting all the data to the OSS management database and tools to ensure that OSS questionnaires are completed by product developers and source code are submitted to the OSS scanning process. Subsequently, OSS Core team review the scan results and questionnaires (Schöttle & Steger, 2015).

Haddad additionally offers an extended team called the Open Source Executive Committee (OSEC) that consists of representatives from IT, engineering, product marketing and compliance officer. OSEC is in charge of establishing OS strategy, inspection and handling of IP releases giving approvals to release the source code under a specific license. There are several teams under the OSEC namely Documentation team (takes care of written offers all respective OS notice in the product documentation), Supply Chain team (mandates 3rd party software and hardware suppliers disclose all OS used in the purchased components and confirm that they meet the OS license obligations), Corporate Development team (is responsible for OS compliance in merges and acquisitions, and outsourced development), IT team (provide support and maintenance for the tools and automation infrastructure used by the compliance program), Localization team (is in charge of translating main information to target language to provide respective information to users about the product or software) (Haddad, 2016).

Richard Kemp offers a complete (internal and external) stakeholder list explaining their roles and objectives. These stakeholders are leadership team (ensures effective use of OSS), finance team (analyses and handles the OSS benefits and risks), technical team (delivers OSS components & developments in time and on budget, deals with technical parts of OSS governance program), developers, supervisory board (ensures that the company adopts suitable OSS governance according to the company's strategy), OSS compliance officer (establishes and realizes OSS governance and ensures compliance), OSS working party (provides respective information to stakeholders), HR team, legal team (mitigates legal risks and increases the benefits in OSS governance), sales & marketing team, shareholders, customers and suppliers (Kemp, 2010).

Most of the interviewed companies have only core teams for OSS compliance and governance (Please see "1e Define the responsibilities of the OSRB"). Our research findings deny the approach of the extended team which is mentioned in the literature.

Open Source Office Program

There are numerous functions of Open Source Program Office (OSPO) at companies. One of the main roles is that OSPO enables companies to fulfil legal liabilities when they combine their own proprietary codes with OSS codes and third-party codes. OSPO also assists organizations to centralize policies about OSS related matters such as OSS use and distribution (Aniszczyk et al.).

OSPO is a means of communication inside and outside the company. By means of OSPO, organizations are able to create and implement their open source strategies precisely as well as to govern OSS successfully within their operations. OSPO can be adapted in different departments based on the organizational structure of the company. For instance, engineering-driven companies fit OSPO within their engineering department or other companies maintain OSPO in their legal department because they have broad IP portfolio (Aniszczyk et al.).

In order to establish an OSPO, corporations should follow several steps. The initial step is to find the right leader with a full-time job to lead the development of the new program office and to run OSPO within the company. The potential leader should understand the company's business objectives of OSS and have the knowledge on how OSS works with technical details (Aniszczyk et al.).

The next step is to establish the budget, tools, and identify necessary employees with clear responsibilities for establishment of OSPO activities. OSPO should be supported by assigned members such as program manager, legal team, compliance team (OSRB and OSEC) etc. According to Aniszczyk et al., OSPO should provide direction to OSS users and contributors, and enable them to make decisions. Lastly, all participants should provide all inputs and feedbacks correctly to sustain a successful OSPO (Aniszczyk et al.).

According to our findings, the large companies have OSPO that supports OSRB. As a matter of fact, OSRB and OSPO are similar concepts as mentioned in the literature. However, we differentiate them in terms of their activities. The members of OSPO are OSRB and developers.

OSRB makes decisions on OSS governance and compliance issues, and developers supports OSRB with OSS-related issues.

Documentation and Audit Trail

Patent and copyright activities related to software are increasing. Therefore, companies should maintain comprehensive documentation and audit trail for the software used in their products and development process. It is crucial for a company to demonstrate that there is no IP-bleed between proprietary and OS software including the software provided by third-party vendors (Ellis, 2011). It is important where inbound software is kept within the company. Responsible people who have access to the software should be specified in OSS governance policy (Ellis, 2011).

Organizations should have a well-structured audit trail including authority and reproducibility mechanisms. It is vital to archive the certified results as well as source code and binary objects of the software within the highly developed audit trail. The retention period for the archive is determined considering the product lifecycle in the market and jurisdictional issues on limitation regulations (Ellis, 2011).

Our research findings are in line with the literature (Please see the best practices "2d Establish a process for auditing source code", "2g Establish a documentation process", 3d Establish a centralized repository for OSS implementation).

Supply Chain Management

Organizations are working with many suppliers that provide different type of software components for different products. The complexity in the supply chain is increasing along with the number of software products and combined works (Ellis, 2011).

Acquisition of OSS is not being governed properly by all organizations. Developers have been using OSS codes existing on the internet without following any formal acquisition process (Popp, 2015).

According to the article published by BlackDuck and BearingPoint, companies do not check the third-party codes properly for security, safety, and quality which leads to delayed or lost revenue, delayed or recalled products, security vulnerability issues (Black Duck Software & Bearing Point, 2013).

Organizations should scan not only open source but also proprietary software provided by the third-parties (Ellis, 2011). Such companies need to create a process to analyse all third-party software components to ensure license compliance (Schöttle & Steger, 2015) (Ellis, 2011). Employees in the Supply Chain management should enforce third-party software and hardware vendors to provide all information regarding OSS used in the delivered products and to help with licensing (Haddad, 2016).

Companies can avoid the risk of license breaches by performing a check on the software by themselves or by using qualified vendors. The compliance check might be performed manually or by using software tools depending on the number of software components used in the product (Schöttle & Steger, 2015).

Our research findings are in line with the literature which are demonstrated in the best practices "2b Integrate OSS into Supply Chain management" and "4a Use a component management tool").

Contracts with Suppliers and Customers

Organizations should add the required clauses to their supply contracts to ensure their suppliers comply with all OSS licenses used in the deliverables (Schöttle & Steger, 2015) (Haddad, 2016). Suppliers should deliver to their customers all the materials and information regarding OSS product compliance (Schöttle & Steger, 2015) (Haddad, 2016).

Our research results show that most of the companies have supply agreement contracts with their suppliers including all necessary OS-related terms (Company 2, 3, 4, 6). ("Please see the best practice "3g Formulate a contract with suppliers").

According to the literature, companies should add necessary clauses in the agreement to inform their customers about the OSS licenses and the source codes such as how to obtain the licenses and where the source codes are provided (Schöttle & Steger, 2015). However, we did not find any solid information regarding the OS-related terms added in sales agreements of the interviewed companies.

Open Source Governance Tools

The survey published by Black Duck reveals that 41% of the vulnerabilities are detected and remediated manually while 10% through third-parties and 19% automatically. 30% of companies have no process for identifying, tracking or remediating open source vulnerabilities (Black Duck Software, 2016a).

One of the great advantages of using OSS is the quality of OSS components, which are consistently updated and improved by the open source community. OS community updates versions due to bug fixes, new functionalities, improvements for security vulnerabilities or compatibility issues (Popp, 2015). However, if OSS components are used in the product, the developers should also follow up the updates. It can be challenging for developers to receive the version update notices for high number of components. Missing version updates may have a negative impact on losing the advantages of bug fixing, exposing the products to hacker attacks and having incompatibility issues (Popp, 2015).

The update issue can be automated by means of tools or outsourced services to determine the version updates of components and track them. Organizations can solve the version update issues replacing the manual work with automated solutions (Popp, 2015).

There are different types of OSS governance tools for different purposes available in the market. Discovery/Selection tools are used for finding OSS project code, reviewing attributes, developing catalogues and white lists for approval. Some examples can be mentioned as Black Duck Code Centre, Open Hub (Popp, 2015). Compliance tools specify the code contents such as a bill of materials or software portfolio as well as the names, versions, licensing etc. Additionally, these tools can support to create required documentation. Some examples for these tools are Black Duck Protex, Black Duck Export, Fossology (Popp, 2015). The context for open source integration is

shown by Dependency Checkers such as Black Duck Protex and The Linux Foundation Dependency Checker, indicating where and how open source libraries and components are invoked and connected (Popp, 2015). Code Sanitation tools such as Linux Foundation Code Janitor is used to refine source code from incorrect language or proprietary data (Popp, 2015). Security Vulnerability Remediation tools such as Black Duck Hub are useful for discovering all open source components and identifying risks (Popp, 2015). Supply Chain tools features are scanning, compliance, security as well as standardization and easy communication between parties for OSS (Popp, 2015).

Our research results show that most commonly used tools for OSS governance are component manegement tools and an OSS compliance checking tools. (Please see the best practices "4a Use a component management tool" and "4b Use an OS compliance checking tool")

Support for the Sales Organization

Companies require their suppliers deliver compliance information for the deliverables. However, suppliers do not have to provide all the information about their proprietary codes such as scanning reports, OSS questionnaires or some OSS database information which may include confidential information (Schöttle & Steger, 2015). Therefore, software enterprises should create a policy which assigns authorized employees to communicate information about the use of OSS with the stakeholders outside the company (Schöttle & Steger, 2015).

However, the interviewed companies do not have any policy on the communication regarding OSS issues with external stakeholders. Therefore, our findings deny this requirement.

Discovery of OSS Used in the Code

It is very crucial to review and satisfy license obligations of all the components and all the sub-components (Schöttle & Steger, 2015) (Sojer & Henkel, 2011) (Haddad, 2016) (Kemp, 2010) (Chang et al, 2010) and then investigate if these OSS codes reused in the software infringe the license obligations (Sojer & Henkel, 2011) to avoid potential risks related to OSS license compliance issues (Schöttle & Steger, 2015) (Kemp, 2010). In order to fulfil this approach, the software development requires elaborated, valid and complete documentation (Schöttle & Steger, 2015).

All the information can be obtained by means of a software through scanning and auditing the code. Audits should be performed periodically to provide permanent compliance and governance (Black Duck Software). There are several tools and methods to handle the issue of documenting OSS related information including the origin of the source and OSS license liabilities (Schöttle & Steger, 2015).

The research findings support the approach given in the literature. (Please see the best practice "2c Discover all the OSS used in the product" and "2g Establish a documentation process")

Training Staff & Communication

OSS training helps increase the awareness of the open source policies and strategies, and create a common understanding on OSS licensing issues and risks. Therefore, the employees should be trained about compliance policies, processes, and guidelines of the company (Haddad, 2016).

Employees of most of the software companies do not receive proper training on the OSS license compliance and reuse. Organizations may provide formal (instructor led courses) or informal trainings (online materials, company-wide open source newsletters, discussion forms, presentations, mailing lists, meetings, seminars and orientations in recruitment) to their employees depending on the size of the company and the extent of the open source used in the commercial products (Haddad, 2016).

All engineering members involved in the product development, all personnel work in the business unit including non-engineering members should receive also the training periodically (Ellis, 2011).

Additionally, "Knowing Your FOSS Responsibilities" requirement of Open Chain should also be mentioned as a best practice example to prepare a guidebook to assess full completion of the mandatory OSS training to include the OSS policy and where to find a copy; basics of Intellectual Property law pertaining to OSS and OSS licenses; OSS licensing concepts (including the concepts of permissive and copyleft licenses); OSS project licensing models; Software Staff roles and responsibilities pertaining to OSS compliance specifically and the OSS policy in general; and process for identifying, recording and/or tracking of OSS components contained in Supplied Software. Software Staff should complete OSS training within the last 24 months. That means, all OSS training materials and method of tracking the completion of the training for all Software Staff should exist. At least 85% of the Software Staff are up to date (Open Chain, 2016).

As Ellis states in his article, it is crucial to provide the OSS training not only to the employees but also to the suppliers. For the software vendors, this training should cover policies and procedures of the company related to the software and it should also focus on OSS requirements (Ellis, 2011) (Fendt et al., 2016).

Organizations should also communicate with open source community to inform them regarding the license obligations that the organization needs to fulfil (Haddad, 2016) (Fendt et al., 2016). There are several external communication forms offered by Haddad such as websites dedicated to distributing OSS for the compliance issues, social welfare and support for OS organizations and contributing and participation in OS events (Haddad, 2016) (Fendt et al., 2016).

Our research findings support the approach given in the literature. However, there are some conflicts in the practice. One of the differences is to provide only informal trainings to their employees instead of instructor led courses. The trainings address only the OSS governance team and developers but not non-engineering members.

Checklist for OSS Process

The checklist for OSS process for companies described by Kemp is divided into 4 stages, which are Dependencies, Pre-Implementation, Implementation, and Post-Implementation (Kemp, 2010). Kemp explains the dependencies on policies from other areas, and different aspects and risks should be conformed to OSS governance strategy and policy statements, Patents and other Intellectual Property Rights policies, Relevant stakeholder groups – e.g. architecture group, etc., Source code management, HR policies, Inbound/outbound contract groups, and exit strategy (Kemp, 2010).

The Pre-Implementation stage consists of 5 parts, these are "Project planning, road mapping, timetabling", "Indicator tool implementation", "Initial assessment", "amnesty cases for developers", "pilot project implementation" (Kemp, 2010).

Implementation stage includes approval process for frequently used OSS licenses (identification of the list for favoured licenses in the company and practice for OSS license identification and analysis), approval rules, pre-launch/release compliance check, and identification of service level for OSS related questions of users (Kemp, 2010).

Kemp mentions "arrangements for code and another information repository", "periodical code assessment", "remediation where necessary" and "training and awareness" in the post-implementation stage (Kemp, 2010).

OSS Compliance Process

OSS governance should be supported by OSS processes (Kemp, 2010). Organizations can mitigate the legal, operational and security risks associated with open source issues by automating the governance and compliance processes (Black Duck Software). Companies may harmonize open source with development and deployment of software and services by means of systematic control (Popp, 2015).

According to Haddad, OSS compliance process consists of 10 steps namely identification of OS, auditing source code, resolving issues, reviews, approvals, registration, notices, pre-distribution verifications, distribution and final verifications (Haddad, 2016).

Haddad specifies certain methods in the identification phase to elicit OS used in the product such as requesting the use of specific open source from compliance team, auditing the complete platform to create compliance standards, demanding to disclose all open source components used in the delivered products from software suppliers and letting the compliance team to review them etc. In the auditing phase scanning of the source code by means of automated tools is required to identify the source code origins and the license of the respective source code (Haddad, 2016).

In the phase of "resolving issues", the engineers should solve all the matters discovered in the auditing phase. Different reviews need to be performed by the responsible parties such as Internal package owner, source code auditing personnel, OSRB and OSEC to comprehend the respective licenses that manage the use, modification and distribution of the software. After the required reviews are completed, most of the components are approved by OSRB in the "Approvals" phase.

Subsequently, the registration phase starts and the approved component is added to software inventory that keeps track of open source use and use case (Haddad, 2016).

In the "Notice" phase, companies have to provide all required information to the end users such as how to achieve a copy of the source code in order to meet license obligations, disclosing the use of OS by providing necessary notices (copyright and attribution notices), providing full text of the license agreements for the OSS codes used in the product (Haddad, 2016).

In the pre-distribution verification phase, the method and mode of distribution, type of packages to distribute, and mechanism of distribution should be decided. In distribution phase, organizations should upload the open source packages to the distribution website, marked with the product and version. Organizations should validate that all uploaded packages can be downloaded by external parties without any issues in the "final verification" phase (Haddad, 2016).

According to Popp, OS Logistics automates open source governance which consists of 6 phases namely Choose, Scan, Approve, Inventory, Secure, Deliver (Popp, 2015). Choose (Discovery and Selection) Phase is to select the most convenient, sophisticated and functional OS code according to company requirements. Scanning Phase is to analyse the source code used at the company considering licenses, versions, origin, and potential effects of risks in all Software development stages. Approval Phase is to endorse the open source code considering the defined policies and workflows. Inventory and Tracking Phase is to document all the open source used at the company specifying the integrated and deployed code, and the version. Security Phase is to record vulnerabilities in the code and perform remediation. Delivery Phase: software code and products should be delivered confidently throughout the supply chain to the end user (Popp, 2015).

Our theory confirms the requirements and embraces OS governance processes (Please see best practices "2a Build an OSS component selection process", "2b Integrate OSS into Supply Chain management", "2c Discover all the OSS used in the product", "2d Establish a process for auditing source code", "2e Establish a review process for software components", "2f Establish a component approval process", "2g Establish a documentation process" and "2h Ensure license compliance for outgoing products").

We categorized similar concepts in the literature based upon concept-centric approach (Webster & Watson, 2002) (Please see Table 3 "Concept Matrix").

#	Articles	Concepts										
		Processes					Management			Tools	Supplementary	
		Identification	Supply Chain	Review & Selection	Approval	Distribution	Strategy	Policies	Roles & Responsibilities	OS Gov. Tools	Training & Communication	Documentation
1	Abernathy, C., Aniszczyk, C., Beda, J., Novotny, S., Yehuda,								x			
2	Aniszczyk, C., McAffer, J., Norris, W., Spyker, A.								x			
3	Black Duck Software	x	x	x			x	x	x		x	
4	Chang, S., Lee, J., Yi, W. (2010)	x				x			x		x	
5	Copenhaver, K., Radcliffe, M., Vescuso, P. (2013)	x		x								
6	Ellis, J. (2011)		x			x		x	x		x	x
7	Fendt, O., Jaeger, M., Serrano, R. J. (2016)	x	x		x		x		x	x	x	x
8	German, D., Di Penta, M. (2012)	x						x				x
9	Haddad, I. (2016)	x	x	x	x	x	x	x	x	x	x	x
10	Höst, M., Oručević-Alagić, A., Runeson, P. (Ed.) (2011)			x			x	x				
11	Kemp, R. (2010)	x	x		x	x	x	x	x	x	x	x
12	Laurent, A. (2004)	x										
13	Lindman, J., Paajanen, A., Rossi, M. 2010	x										
14	Muffatto, M. (2006)			x								
15	Open Chain (2016)		x						x		x	x
16	Popp, K. M. (2015)				x	x	x	x		x		
17	Schöttle, H., Steger, U. (2015).	x	x					x	x	x		x
18	Silberman, G. P. (2014)			x	x			x			x	
19	Sojer, M., Henkel, J. (2011)	x						x			x	x
20	Yu, Y. (2013)			x								

Table 3. Concept Matrix

3.2 The Complete Best Practice List

3.2.1 Management

Name	1a Establish an open source compliance strategy
Actor	Top management
Context	Initially it is critical that the executive board is fully involved with a consensus in building the OSS governance strategy. Risks and benefits of OSS use need to be considered by top management. It is very important that the decisions made by top management regarding OSS use are shared with company employees.
Problem	How to establish principles for the open source governance? How to inform employees about the purpose of the OSS use?
Solution	OSS strategy allows the company management to define the potential risks of OSS use and the internal compromises. The company management should take business, legal and technical decisions on establishing the strategy. It is necessary to define the company's OSS objectives by explaining OSS compliance within the company. By means of the OSS strategy, employees can understand the purpose of the OSS, its practice areas and the areas to avoid.
Sources	Literature: (Fendt et al., 2016); (Kemp, 2010); (Popp, 2015); (Haddad, 2016); (Höst et al., 2011). Interviewee: Company 1, 3, 5, 6.

Name	1b Establish an open source compliance policy
Actor	OSRB
Context	Employees should be informed about company principles regarding the OSS use including formal processes from acquisition to release of the software, internal rules, and potential issues while using OSS.
Problem	How to inform employees about the company's OSS principles? How to comply with Open Source license obligations?

Solution	The OSS Compliance policy can be stored as a single document or divided into 2 separate documents. The first one explains the intention of the use of open source, defines the rules of how to use OSS and proprietary software and clarifies the governance processes. It outlines what kind of licenses including their legal assessments and packages are acceptable for use in commercial products. The second policy document establishes a set of standards broadly about what the employees need to do to ensure OSS compliance. The policy can be implemented and be regulatory across the whole company under identical conditions. Also at larger companies, each division or department can adopt the policy with certain differences. It is necessary to make the policy accessible to the employees. Companies should provide training to ensure their staff understands the significance of OSS compliance and the principles of OSS policy (Please see *"3e Establish an employee training program"*). It is also important to maintain the policy by receiving feedback from the development teams to improve the processes. It is also essential to record violations to understand why certain development teams violate the process to minimize the issues.
Sources	Literature: (Fendt et al., 2016); (Ellis, 2011); (Haddad, 2016); (Kemp, 2010); (Popp, 2015); (Schöttle & Steger, 2015). Interviewee: Company 2, 3, 4, 5, 6.

Name	1c Establish a reuse policy
Actor	OS Compliance Manager; Legal Counsel
Context	Companies reuse OSS components for different purposes such as saving time, reducing the license review process or increasing the quality. However, it is highly important to ensure how reused components are being incorporated into products, because the reuse of an OSS component might generate proper or improper results in different cases. Therefore, organizations should be careful with the reuse of OSS components.
Problem	How to reuse OSS components? How to ensure OSS components are reused appropriately?
Solution	The solution is to create a reuse policy to handle the compliance issues. Companies need to track how OSS functions, where it is being used and re-used. All approved components can be stored in a centralized repository but should be reused by following the reuse policy and defined procedures (Please see *"3d Establish a centralized repository for OSS implementation"*). If the company does not have a repository, it is required to check the components, case by case and approve accordingly based on the declared rules.
Sources	Literature: (Sojer & Henkel, 2011); (Kemp, 2010); (German & Di Penta, 2012). Interview: Company 2, 4, 6

Name	1d Create a policy on how to contribute back to the OS community
Actor	OSS Compliance Manager; Legal Counsel
Context	Companies can gain further benefits by contributing back to the community. The company's own improvements can be enhanced further by the community and used in the future versions. If the company needs to use the newer version of the respective software, further improvements would be implemented. However, negligence may cause to disclose the company's intellectual property. Therefore, companies should be cautious when they are contributing to the community.
Problem	How to contribute back to the community without disclosing company IP?
Solution	The solution is to create a policy to clarify the procedures before contributing. Developers who would contribute to the community should first consult with their project or product team leader and finally the OSS compliance manager. The team leader reviews the changes, the modification and the contribution. It is important to check during the review in terms of legal and technical perspectives. It is critical to assess if the legal applications have been met, if any kind of company IP is disclosed or if the contribution is valuable for the project to improve technical issues. First the contribution should be approved and the scope should be well defined. After the contribution review process is completed, the developer can contribute as a company employee to the project within the scope.
Sources	Literature: (Haddad, 2016) (Kemp, 2010) (Schöttle & Steger, 2015) (Open Chain, 2016). Interviewee: Company 2, 5, 6

Name	1e Define the responsibilities of the OSRB
Actor	OSRB;
Context	It is not possible to establish and maintain a successful OSS governance across the company without an assigned team.
Problem	How to engage people for effective OSS governance inside the company? How to establish and maintain processes and coordinate them? Who will conduct all the compliance issues? To whom can employees address the issues regarding OSS compliance and governance?

Solution	Organizations can achieve efficient OSS governance only when employees are engaged in the assignments with clearly defined roles and responsibilities. There should be a designated team assigned to receiving and solving queries related to OSS issues. It is required to build a core team, usually called Open Source Review Board (OSRB) with clear objectives engaging participants in a multidisciplinary approach from various departments such as product team manager, engineering manager, legal counsel and the compliance manager. OSRB is responsible for: • ensuring OSS license compliance for OS software obtained from third-party suppliers and OS communities, • reviewing and approving components, • establishing and maintaining OSS processes, policies and guidelines, • building internal communication. Additionally, corporations have an established Open Source Program Office (OSPO) for the OSS governance across the company to support to the OSRB (Please see *"1f Establish Open Source Program Office").*
Sources	Literature: (Ellis, 2011); (Kemp, 2010); (Schöttle & Steger, 2015); (Haddad, 2016). Interview: Company 1, 2, 3, 4, 6

Name	1f Establish an Open Source Program Office (OSPO)
Actor	OSRB, Developers
Context	It is crucial to establish as well as to sustain OSS license compliance check procedure and audits. A well-established OSPO supports the use of open source effectively in commercial products/services, and helps organizations train their developers and build the community engagement.
Problem	How to centralize processes and decision-making about OSS related issues?
Solution	The solution is to establish Open Source Program Office to centralize processes and decision-making. It is crucial for large corporations to use OSPO to support OSRB related to OSS governance and compliance assignments and activities. The members of OSPO are members of OSRB and developers. Developers in each team can be assigned to allocate 20% of their working time to deal with open source compliance. Alternatively, the developers can be volunteers to contribute to OSRB for OSS governance and compliance. These volunteers only actively contribute to the tasks depending on their free time and schedule. The communication can be held by means of the internal mailing lists.
Sources	Literature: (Aniszczyk & McAffer.); (Aniszczyk et al.); (Abernathy et al.). Interviewee: Company 2, 6

Name	1g Structure the in-house legal department
Actor	Top management,
Context	It is required to understand legal requirements for OSS to mitigate the risks and enable developers to utilize open source for company benefits. In order to receive feedback and legal advice concurrently, organizations should structure the legal service.
Problem	How to receive a legal advice regarding OSS governance and compliance issues concurrently?
Solution	Legal counsel is responsible for analysing and giving legal advice on licenses, approving OSS use for commercial purposes, reviewing and providing OSS notices and disclaimers and establishing all necessary policies and processes. It is a business decision to build a legal department or hire a legal counsel. Usually, such a decision is made based on the company size and the intensity of OSS use. Essentially, legal counsel is a part of the core team. However, the small-sized companies usually outsource the legal services. Mid-sized companies and corporations have their own legal departments with at least one expert on OSS governance issues. Company legal counsels also consult with the legal experts outside the company, especially for the complicated issues.
Sources	Literature: (Haddad, 2016); (Schöttle & Steger, 2015); (Ellis, 2011); (Kemp, 2010). Interviewee: Company 1, 2, 3, 4, 5, 6

3.2.2 Processes

Name	2a Build an OSS component selection process
Actor	Engineering manager, developers
Context	Various alternatives of OSS components are available on the web. The wrong selection of OSS components may increase the development costs of the product and lead to violate OSS license terms. Therefore, it is crucial to choose the right component.
Problem	To which aspects organizations should pay attention in choosing an open source software component? How to choose the right component among alternatives?
Solution	The component selection process should be established based on the legal and strategic decisions on the portfolio to fulfil the technical needs. Developers should

	select the component considering the approved list for licenses (Please see *"3c Create an approved list for licenses"*). The practice for the component selection process is to create an own repository of OSS components to catalogue, develop and process the usage of OSS (Please see *"3d Establish a centralized repository for OSS implementation"*). OSS components should be only downloaded from the repository by developers where they are documented after the approval process (Please see *"2f Establish a component approval process"*). The component selection can be made for each project by assessing all the necessary components with their licenses, copyright notices, export restrictions, their functionalities, modification, component linkage etc. that are going to be used in the software development, cataloguing them in the company repository and finally by receiving the approval from the OSRB. If the OSRB or executives do not want to limit their developers with the components in the company repository, it is also possible to manage component selection process by using a specific tool. When a developer needs to use an OSS component, he/she should provide all necessary information by filling out a form. After examining the scan results done by the tool, the developer receives the permission or rejection from the OSRB about the request.
Sources	Literature: (Popp, 2015); (Fendt et al., 2016); (Schöttle & Steger, 2015); (Haddad, 2016); (Black Duck Software & Bearing Point, 2013); (Höst et al., 2011). Interviewee: Company 1, 3, 4, 5, 6

Name	2b Integrate OSS into Supply Chain management
Actor	Supply Chain team, Legal Counsel, OSS Compliance Manager,
Context	Third-party components are provided under the main license to their customers. However, the third-party components might include sub-components that are provided by different OS communities or third-party suppliers which might be licensed under different license categories. When organizations combine the third-party codes with their proprietary codes without knowing the exactly content, they might run into incompatibility issues.
Problem	How to ensure the third-party codes supplied by the vendors do not have any incompatibility issues?

Solution	The solution for the incompatibility issues that arise from inbound transactions of OSS is to improve the supply chain process. Organizations should request from third-party suppliers to disclose all OSS used in the purchased components to confirm that they meet the OSS license obligations and to report by means of a tool that supports a SPDX format (Please see *"4a Use a component management tool"*). Suppliers should provide bill of materials and the complete documentation given by the tool including source code scan results. All required documents (e.g. BoM presented in SPDX format, Source code scan results) which will be provided by the suppliers should be added to the Contracts with the suppliers as a requirement mentioning the delivery deadline for the respective documents (Please see *"3g Formulate a contract with suppliers"*). It is also important to crosscheck and scan the third-party software to detect discrepancies and analyse them by means of a tool. If there is non-compliance with the licenses and copyrights, the supplier should check and provide the correct disclaimer for the respective product.
Sources	Literature: (Haddad, 2016); (Schöttle et al., 2015); (Kemp, 2010); (Ellis, 2011); (Black Duck Software & Bearing Point, 2013); (Fendt et al., 2016). Interviewee: Company 2, 3, 4, 6

Name	2c Discover all the OSS used in the product
Actor	OSS Compliance Manager, Legal Counsel, Developer
Context	In order to handle open source license noncompliance issues and prevent the license infringements, organizations should know what exactly they use in their products to take necessary precautions and to protect the IP.
Problem	How to discover what kind of OSS components are used in which product?
Solution	The solution is to determine all OS software components used in the products and ensure the license terms of all the components. It is critical to use an automated tool to scan the software code (Please see *"4b Use an OS compliance checking tool"*). Therefore, it is required to have complete and accurate documentation storing all the necessary information and keep it up-to-date (Please see "*3b Create a bill of materials*").
Sources	Literature: (Schöttle & Steger, 2015); (Sojer & Henkel, 2011); (Haddad, 2016); (Black Duck Software); (Kemp, 2010); (Chang et al., 2010). Interviewee: Company 2, 4,

Name	2d Establish a process for auditing source code
Actor	OSRB, Quality engineer
Context	Organizations should ensure that their commercial products are not subject to intellectual property risks because of OS software use.
Problem	How to ensure that there is no IP-bleeding in the commercial products?
Solution	The solution is to perform random audits. Audits can be conducted on code base by checking millions of lines of codes in each product or project base depending on the business model of the company. For auditing, it is required to have complete documentation to store all approved components or approved projects including respective licenses in the repository *(*Please see *"3d Establish a centralized repository" and "2g Establish a documentation process*"). It is necessary to have randomly scheduled audits for projects or products. The audits should be conducted by quality engineers and OSRB to have a complete organization-oriented audit approach towards quality to compare with IP plan. During the audits, it is necessary to check what was presented, reviewed and approved.
Sources	Literature: (Haddad, 2016); (Ellis, 2011); (Schöttle & Steger, 2015); (Black Duck Software). Interviewee: Company 1, 2, 4

Name	2e Establish a review process for software components
Actor	OSRB, Developer
Context	It is important to manage the use, modification and distribution of open source software. Review process plays an important role to check license compliance and compatibility to avoid security issues.
Problem	How to ensure that the company fulfils the license obligations of all OSS found in the product?

Solution	The review is usually performed by the OSRB. First of all, it is necessary to build a checklist for the review process according to the IP plan (Please see *"3h Build a checklist for software component review"*). One of the main parts of the process is to check the licenses and their obligations for all OS found in the product (Please see "*2c Discover all the OSS used in the product"* and *"3b Create a bill of materials"*). The contexts of the licenses should be checked to identify if there are any discrepancies, if they are modified by the community or if the licenses are mentioned incorrectly. It is important to have a linkage check to understand if there is a problematic code between the OS code and the proprietary code. The architectural diagrams can be checked in complicated cases to understand what is combined with what, and what is running where etc.
Sources	Literature: (Schöttle & Steger, 2015); (Haddad, 2016); (Ellis, 2011); (Yu, 2013); (Silberman, 2014) (Lindman et al., 2010). Interviewee: Company 1, 2, 3, 4, 6

Name	2f Establish a component approval process
Actor	OSRB, Legal counsel
Context	OSS components should be approved according to the results given in the review process.
Problem	How to ensure if the component is compliant or not?
Solution	The OSRB checks the list of all components and their licenses used in the product, and then approves them separately. Once the component is approved it can be used again without approval process, unless there are changes in the bill of materials and the content of the version. Generally, components are chosen by developers among the pre-existing assessed licenses in the whitelist (Please see *"3c Create an approved list for licenses"*). In this case, the checklist is signed by the respective manager without having approval from OSRB. If the company does not have a repository and approved license list where assessed licenses and components exist, it is necessary to approve each open software component. If a company does not approve components but approves projects, then the company should have the expertise to detect the component with an unusual license during the review process and replacing it with a reliable one and approve accordingly.
Sources	Literature: (Haddad, 2016); (Popp, 2015); (Kemp, 2010). Interviewee: Company 2, 4, 6

Name	2g Establish a documentation process
Actor	OSRB
Context	It is necessary to track the use of OSS in the products and enable developers to reuse the components to shorten the compliance check processes. It is also required to document all the OSS after the approval and audit processes, and when the integration is completed.
Problem	How to track where the approved OSS components are used in the product?
Solution	All the approved components should be stored in the company repository to track open source use and use cases (Please see *"3d Establish a centralized repository for OSS implementation"*). Additionally, it is necessary document all the results when the audits have been completed. To monitor all the OSS used in the product or the project, it needs to be documented specifying all necessary information for security issues. The documentation is also important to reuse the components for reducing the license review process.
Sources	Literature: (Popp, 2015); (Kemp, 2010); (Haddad, 2016); (Fendt et al., 2016). Interviewee: Company 2, 3, 4, 6.

Name	2h Ensure license compliance for outgoing products
Actor	OSRB
Context	Before shipping the product, organizations should verify that all components in the respective product have been assessed and approved for usage.
Problem	How to ensure license compliance for outgoing products?
Solution	Once the approval has been received, the software is implemented and integrated into the product. Once the integration of the component into the product is completed, it should be documented in the centralized repository to track the use of OSS components (Please see *"2g Establish a documentation process"*). It is necessary to cross-check all the copyrights and licenses to ensure the license compliance before shipment. It is necessary to produce an office file and match all the copyright and licensing in a standard way by means of an automated tool. It is required to provide the source code via CD or on the web. It is also essential to ensure the license compliance according to the distribution and maintenance models. Furthermore, it is required to check if the product is exported and has export restrictions.
Sources	Literature: (Haddad, 2016); (Popp, 2015); (Kemp, 2010); (Ellis, 2011); (Chang et al., 2010). Interviewee: Company 2, 3, 6

3.2.3 Supplementary

Name	3a Create a complete documentation
Actor	OSS Compliance Manager
Context	It is essential to document all kind of information related to OSS to discover, check and track the information flow.
Problem	How to discover, track and maintain OSS related information?
Solution	The solution is to have all OSS policies (e.g. OSS compliance policy, OSS reuse policy, contribution policy to the OS community etc.), training materials, assessed license lists (Black and White list), all approved OSS components including scan results, bill of materials (Please see *"3b Create a bill of materials"*) and all documents of products delivered by vendors catalogued in the company database and shared with the responsible people. It is also necessary to establish a repository to catalogue all OS components with the licenses and their assessments used in the products or projects (Please see *"3d Establish a centralized repository for OSS implementation"*).
Sources	Literature: (Haddad, 2016); (Schöttle & Steger, 2015); (Open Chain, 2016); (Ellis, 2011); (German & Di Penta, 2012); (Popp, 2015). Interviewee: Company 2, 3, 4, 6.

Name	3b Create a bill of materials
Actor	Open Source Compliance Manager
Context	It is highly important to discover all necessary information regarding OSS for successful OSS review, approval and tracking processes
Problem	How to track that there is no IP-bleed between commercial code and OS software code provided by OS community and third-party suppliers? How to go back to see the scan results and responsible person for the specific request?

Solution	The documentation for all the products contains bill of materials for each product to identify what kind of software is being used including the component name, component versions, licenses and license liabilities, copyright information, the link to the software, the linking modification, functionality of the software, where it is used in the product, if it is reused, if it is modified, if it is linked to the company code, the scan ID, the source where the software is obtained and how integrated it is into the product and how it is released. It is crucial to have a complete bill of materials for each product and keep it up-to-date. All the BoMs need to be implemented and reported by means of a tool that supports SPDX format to make both scanning and documentation easy. BoM needs to be updated if any changes happen in the particular product.
Sources	Literature: (Haddad, 2016); (Schöttle & Steger, 2015); (Open Chain, 2016). Interviewee: Company 2, 4,

Name	3c Create an approved list for licenses
Actor	OSRB, Legal counsel
Context	Different programs under various licenses (both commercial and open source) can be combined by developers during the software development. The wide range of license categories complicates the license selection process. Poorly managed license selection process may cause incompatibility issues.
Problem	How to manage multiple licensing? How to reveal potential licensing conflicts? How to solve compatibility issues with the open source software licenses?
Solution	The solution is to create a list for approved licenses to handle the multiple licensing which may cause a violation of one of the licenses used in the product development. The company executives should identify which open source licenses or OS license categories will be used according to the company strategy considering the license obligations (Please see *1a Establish an open source compliance strategy*). Therefore, it is important to make business and legal decisions for the license selection process. The solution is to have an approved license list and use cases that guides developers to use only the components under the licenses in the approved list considering the defined use cases. However, developers should receive a training on OSS licenses and compliance to understand how to handle the issues (Please see *"3e Establish an employee training program"*). If the company does not have a black and white list, it is necessary to specify acceptable licenses by using statements such as "use one of the licenses approved by Open Source Initiative (OSI)". In case a developer wants to use a component under a license which is not approved by OSI, he/she should convince the OSRB to use the

	specific license in question. The license can be selected after receiving an approval from OSRB. Instead of having a complete "black and white license list" or OSI approved license requirements, a company may use a diagnostic tool which stores the complete information and obligations of all the OSS licenses. A license can be selected case by case for each request.
Sources	Literature: (Laurent, 2004); (Kemp, 2010); (Fendt et al., 2016); (Lindman et al., 2010); (Muffatto, 2006); (Silberman, 2014); (Yu, 2013). Interview: Company 1, 2, 3, 4, 6

Name	3d Establish a centralized repository for OSS implementation
Actor	OSRB, Engineering Manager
Context	Centralized repository enables companies to identify how OSS is implemented, to track where it is used and reused, and to categorize.
Problem	How to catalogue and track approved components?
Solution	It is critical to have a list of all the assessed OSS components in the repository to document, report and track them. The list, generally, contains the following information: • the request number, • the name of the software, • the version of the software, • the link to the software, • the linking modification, • functionality of the software, • scan results, • the scan ID, • and use cases. An established repository helps catalogue all the approved OSS components with their licenses and assessments to handle the security issues. Developers are able to download the components from the repository to use them in the product development. The repository facilitates that the companies can give direction to developers about use and reuse of the OSS component accurately. For instance, the component can be marked with a colour in the repository based on the various metrics of the actual quality of the code to provide the developers with further information regarding the OS community (e.g., green colour means that the OS community of the component is vibrant and if you submit it back, it gets fixed). If there is a need to use a new component which is not available in the repository, a developer suggests the new OSS component for the OSRB to consider. The new

	component can be added to the repository after the approval process done by the OSRB (Please see *"2f Establish a component approval process*").
Sources	Literature: (Sojer & Henkel, 2011); (Kemp, 2010); (Fendt et al., 2016); (German & Di Penta, 2012); (Schöttle & Steger, 2015). Interviewee: Company 2, 3, 4, 6

Name	3e Establish an employee training program
Actor	OS Compliance Manager, Legal Counsel
Context	Employee training is intended to increase the awareness of OSS governance inside of the company. Also, training provides a common understanding of strategic and technical implications, and common mindset on OSS compliance.
Problem	How to build a common understanding of OSS licensing issues and risks? How to increase the awareness of the OSS compliance and governance in-house?
Solution	Employee groups who will be involved in the training program should be initially identified e.g. developers and architects, or team managers as well as legal team, supply chain team etc. All the training materials including necessary forms and documentation should be available to the identified group of employees via the corporate intranet, internal Wiki or other platforms. Companies can provide trainings to different group of people. For example, developers and architects receive a web-based training and attend company-wide talks focused on specifically OSS governance and compliance matters as well as technical aspects to improve developer skills. Whereas, company executives may receive only brief information. The trainings can be provided on a regular basis such as once or twice per year to only team managers. The training materials and the information obtained during the training are stored and shared with all the developers who can ask further questions to their team managers (Please see *"3a Create a complete documentation"*).
Sources	Literature: (Kemp, 2010); (Haddad, 2016); (Sojer & Henkel, 2011); (Ellis, 2011); (Open Chain, 2016); (Chang et al., 2010); (Fendt et al., 2016); (Silberman, 2014). Interviewee: Company 2, 3, 4, 6.

Name	3f Build an internal communication system
Actor	OSRB, OSS Compliance Manager
Context	Communication plays an important role to achieve the success of OSS compliance issues. It is a big challenge for global companies having offices throughout the world to share the information and best practices across the company. People may spend plenty amount of time to solve same or similar issues regarding OSS use if there is no knowledge exchange procedure inside company.
Problem	How to build a compliance communication within the company? How to share the knowledge with other teams located in different part of the corporation?
Solution	The solution is to create an effective communication channel inside of the company. There are several methods which are used to increase the awareness of the OSS governance and compliance within the company namely providing trainings (Please see *"3e Establish an employee training program"),* organizing company-wide talks or seminars, and publishing training materials, mailing lists in an online platform, policies, guidelines for OSS use etc. Instead of producing a whole booklet of instructions such as "volume one, volume two", these documents can be published with a small piece of information in a corporate intranet, blog or internal wiki-like platforms that enables users to achieve all the documents easily by using their individual account. It is also necessary to build a procedure for knowledge exchange. Companies should assign a central decision-making management team having general overview of the OSS governance, namely Open Source Review Board for directing people towards the assignments. (Please see *"1e Define the responsibilities of the OSRB".* In case a team has challenging issues, OSRB can direct the team to the people who previously had dealt with or solved same or similar issues and applications.
Sources	Literature: (Haddad, 2016); (Sojer & Henkel, 2011); (Open Chain, 2016); (Fendt et al., 2016). Interviewee: Company 2, 3, 4, 6.

Name	3g Formulate a contract with suppliers
Actor	Legal counsel, OSS Compliance Manager, Supply Chain Team
Context	Organizations should know what kind of licenses are used in the product delivered by the suppliers and ensure license obligations have been satisfied. It is necessary to identify the responsibilities of suppliers and request that they fulfil and demonstrate the OS license obligations and compliance that apply to delivered products.

Problem	How to ensure all the necessary materials, deliverables and information will be provided by the supplier on time? How to mandate the supplier to fulfil OSS license obligations?
Solution	The solution is to formulate a supply agreement contract adding all necessary requirements regarding OSS to mandate suppliers to fulfil their liabilities. Organizations that request OSS license compliance from their suppliers should contractually demand disclosure of all OSS used in the delivered product, provide BoM documentation using SPDX format for OSS, provide source code scan results, and provide all necessary materials and information regarding OSS on time.
Sources	Literature: (Schöttle & Steger, 2015); (Haddad, 2016); (Ellis, 2011). Interviewee: Company 2, 4, 5, 6.

Name	3h Build a checklist for software component review
Actor	OSRB, Developers
Context	One of the central points of the review is to check licensing and fulfil the license obligation for all the open source found in the certain product/project. It is necessary to ensure that all aspects are covered and checked during the review process.
Problem	How to improve the review process for software components?
Solution	When the developers start to work on a project, they should fill all necessary information in the IP plan beforehand. In order to solve the issue, companies should build a checklist in order to perform the review for software components in an appropriate manner. The checklist should include the following information: • The content of the software, • The purpose of the need for the OS component, • What kind of OS is included, • If it has been delivered or not, • If it has been modified or not, • The functionality of the OS component, • Provide BoM, • Provide scan results, • Architectural diagram for the complicated applications, • Applications or approval requests for a single application that uses a certain library • Distribution model • Maintenance model

	The checklist may have different number of checkpoints (e.g. 20/30/40) depending on the project/product because some checkpoints might be not applicable.
Sources	Literature: (Kemp, 2010); (Schöttle & Steger, 2015). Interviewee: Company 2, 4, 6

3.2.4 Tools

Name	4a Use a component management tool
Actor	OSS Compliance Manager
Context	It is a challenge to ensure the license compliance for third-party components manually with the variety and the increased number of used OSS components in the delivered products. OSS compliance teams should manage the issues that arise from OSS noncompliance of third-party components to avoid the risk of license breaches.
Problem	How to manage supply chain?
Solution	The solution is to use a component management tool such as SW360 that supports SPDX format to scan the components for compliance and security issues. The tool also helps to standardize the documents and processes, and provides an easy information flow between suppliers and customers for OSS. By means of the tool, organizations may receive the complete documentation transformed in a common format for bill of materials from their suppliers as well as provide the documentation to their customers in the same format. Software development teams should provide all necessary documentation such as SPDX documents to compose and use them for the license information in the integration phase. The standardized format enables companies to put the bill of materials into the license scanner to check for compliance easily.
Sources	Literature: (Haddad, 2016); (Schöttle & Steger, 2015); (Popp, 2015); (Fendt, 2016). Interviewee: Company 2, 4, 6

Name	4b Use an OSS compliance checking tool
Actor	Engineering manager, Developers
Context	Organizations should know their codes to ensure successful OSS compliance. It is highly recommended to take an advantage of using a tool which would provide a detailed understanding about the code content and assist to create a documentation.

Problem	How to get an insight into the content of a software code to fulfil compliance requirements?
Solution	To solve the issue, companies should utilize a compliance tool to check the bill of materials of the code or software portfolio. By means of the tool, companies can know what kind of software was integrated into their product. The tool enables companies to scan the code automatically and compare with the IP plan to identify if there is a mismatch or not. Widely-used (both commercial and open source) tools are Fossology, Black Duck, etc. The scan is performed before the legal check to give the quick results with a colouring system (Red – critical, Yellow – check it with legal department, Green - ok) to the developers to understand if the OSS contains risks or not. If the developer wants to implement the component with the green colour results, he/she needs to report it in the tool for the further documentation.
Sources	Literature: (Schöttle & Steger, 2015); (Popp, 2015); (Haddad, 2016); (Fendt et al., 2016). Interviewee: Company 2, 3, 6

3.3 Acknowledgements

I would like to express my deep gratitude to my research supervisors Prof. Dr. Dirk Riehle and Nikolay Harutyunyan for providing contacts with the industry partners, helping organize and conduct the interviews, their patient guidance and useful critiques of this research work.

I would also like to extend my thanks to the industry partners for their assistance with the collection of my research data.

Appendix A Interview Questions

// 20 must-ask questions

// Context questions

1. Please present yourself, your role and responsibilities?
2. Please present your company, your market, your products?

// OSS context

3. What kind of business model do you have? (b2b organization OR b2c organization)
4. Which of your products include software?
    a. What kind of open source software do they include?

// Getting Started - Open Source Governance in general

5. How did your company start using OSS?
    a. when?
    b. in which products?
    c. when did you start keeping track of such use?
    d. was it regulated? how?
6. Did your company start using OSS directly in the product development or was there a phase when OSS was only used for internal purposes? If you used OSS for internal purposes, did you have a governance policy?
    a. OSS as part of product
    b. OSS as tools for product development
    c. OSS as part of R&D
7. Why did you start using OSS? Which benefits has it brought to your company?
    a. any estimated value/cost analysis?
8. Did you have a financial budget to establish OSS governance for the beginning?
    a. Was there any investment strategy in the beginning?
9. How did you decide to use open source components or codes in your product or service? Who pushed for OSS use and/or contribution and/or governance?
    a. developers / middle-managers (bottom-up)
    b. management (top-down)
10. Do you have a documented OSS strategy for your organization?
    a. If yes, since the beginning?
    b. could you please summarize the strategy?
    c. in terms of open source use
        i. OSS as part of product
        ii. OSS as tools for product development
        iii. OSS as part of R&D
    d. in terms of open source contribution
    e. in terms of open source management

f. in terms of open source governance
    i. costs
    ii. processes
    iii. roles
    iv. etc.
g. Did you identify the risks when you created a OSS strategy?
h. Could you please give us access to the documents?

// Getting Started - Open Source Governance in specific best practices

11. Could you please explain your business process for open source software governance?
12. Do you have an open source program office?
    a. If yes, what is the role of the program office?
    b. How did you establish the open source program office?
13. Do you have a mechanism or guidelines for OSS usage?
    a. If yes, when? Since the beginning?
    b. Could you please explain the mechanism?
    c. Do you use any specific software tool?
    d. These documents and guidelines have been accessible by assigned OSS team easily since the beginning?
    e. Could you please give us access to these documents?
14. Do you have a documented OSS governance policy?
    a. What does OSS policy include? Obtain, use, track, maintain?
        i. Do you have OSS approval process?
        ii. Do you have a review process?
        iii. Do you have an audit process? (flowchart)
        iv. (Does your policy include mergers and acquisition terms?)
    b. When did you establish the OSS governance policy in the beginning since you started using OSS in your product?
    c. How did you come up with the idea to have a OSS governance policy?
    d. Did you start using governance policy before you introduced OSS or after having lessons learned issues?
       For example, did you use OSS components only in your internal services or products and decided afterward to distribute your product with OSS components?
    e. Do you have a OSS governance workflow?
    f. How long did it take to establish the OSS policy?
15. Do you reuse OSS components?
    a. How?
    b. Where do you store the metadata?
    c. Do you have a reuse policy?
16. Do you have a OSS policy for the supply chain? (Open Logic - Open Source Policy

Builder)

a. Does your company distinguish between companies that supply OSS and companies that provide proprietary software?
    i. No
    ii. Yes
b. What is the role of software supplied by the vendor?
    i. Is the software the heart of your product?
    ii. Is it considered a key to product experience but can be replaced by other similar software?
    iii. Or do you use this software for apps and consumer downloads?
c. What are the requirements for software delivered to your company from a vendor?
    i. None, it's the responsibility of the supplier to make sure they are adhering to any and all OSS or proprietary licenses
    ii. The supplier must detail all software in their components, including the specific licenses under which the software is being made available
    iii. The supplier must provide a contractual bill of lading that includes a detailed list of software, license(s), and test results from a code scan (e.g., OpenLogic)
d. What are the minimum damages required when dealing with a vendor that supplies software to your organization?
    i. None (no damages; sufficient to cure the breach in an agreed-to timeframe)
    ii. Partial (damages only in actual costs incurred by company to address the breach)
    iii. Full (damages cover all costs including indirect costs — e.g., loss of reputation)
e. Do you check the 3rd party components?
    i. If yes, do you use a special tool to check the 3rd party code?
        1. Binary scan?
        2. Source code scan?

17. How did you assign roles and responsibilities to your employees?
    a. Do you have a documented stakeholder map with clear roles and responsibilities?
    b. Who are the OSS team members?
        i. The technical team,
        ii. developers,
        iii. OSS Review board,
        iv. Legal team,
        v. HR Team,
        vi. OSS Compliance officer
        vii. etc.
    c. Is it documented?

    d. If yes, could you please give us access to your stakeholder map?
    e. Were they employed full time or part time?
    f. Do their roles or responsibilities change for each project?
    g. Who makes decisions on open source related issues?
    h. Do you have OSS Review Board? Who are the main members of the OSS review board?
    i. How do you hire a new developer?
        i. Does the employee contract have any specific terms regarding OSS governance?
        ii. What competencies are important?
        iii. Is it a key attribute to have a skill both in software development and open source community expertise?
18. Do you have a OSS training program for your employees?
    a. When did you start training your employees?
    b. What kind of training do you provide (formal, informal, external support, internal wiki etc)?
    c. The training for which employee groups?
19. Do you have a legal department or legal people in your company for OSS issues, or you outsource the legal service?
    a. If yes, how many legal people work for OSS issues?
    b. How did you handle legal compliance issues in the beginning? Did you outsource the legal service OR did you have internal legal people trained on OSS issues?
20. Do you have OSS contracts with your suppliers and customers with the explained purchasing terms and customer-facing terms?
    a. Disclosure
    b. Approval
    c. Warranties
    d. Indemnification
    e. Scanning/Audit requirements
21. How do you check if you fulfilled license obligations or use of the correct license?
    a. Do you have guidelines or a process to choose the right OS license or check license compliance?
    b. Do you have a repository to archive and list permitted and prohibited licenses?
    c. Could you please give us access to the document?
22. How do you check if the product is ready for distribution?
23. Do you have a OSS scanning software tool to search a source code (both internal and 3rd party codes), get approval for code use and catalog components for reuse and standardization? Such as Software Package Data Exchange (SPDX), BlackDuck KnowledgeBase, BlackDuck OpenHub, etc.
    a. If yes, did you have the tool from the beginning, or did you realize the need for a

tool after a while?

b. If not, how did you know or check how much OSS code or which components you have in your product?

24. Did you get any external support outside the company to establish a OSS governance program?

// more optional questions

25. Do you have your own best practices or lessons learned experiences in the application or OSS governance?
26. Do you have any plans to optimize your OSS governance processes in the near future or long-term?
27. Are your employees allowed to speak or provide information publicly about the use of OSS in your product?
    a. Do you have an established policy which clearly identifies the people within the company who are authorized to communicate the information about the use of OSS? Some information might be confidential and should not be shared with 3rd parties.
    b. If yes, since the beginning?
    c. Could you please give us access to the document?
28. Are your employees allowed to contribute to OSS projects or participate in OSS communities?
    a. Do you have a OSS policy for finding, creating and maintaining with OSS community?
    b. Since beginning?
    c. Do you contribute back to the community?
        i. Why?
        ii. Why not?

// for developers

29. What kind of software development model do you use? (agile, V-model, waterfall etc.)
30. Do you have questionnaires to determine if the OSS component was used in a proper manner and within the restrictions imposed by the relevant OSS license and the company OSS strategy?
31. When you have time restraints or budget restrictions do you still follow all the rules or OSS management process?
    a. Do you have a shortcut (shortest path) to product launch?
32. Who conducted a OSS component search in the beginning? And how?
    a. Now, is there any established process for OSS component search?
    b. Do you have any selection criteria or process?
33. Do you document the OSS codes used in your product?
34. Do you have an approval mechanism for new OSS codes?

Appendix B Theoretical Sampling

Dimensions	Business Model			By type of customer		By market position			By size (employees) / market capitalization			By maturity of company			By maturity of product		
Companies	Closed / Proprietary Software business model	Complete Open sourced business model (vendor or distributor)	Consultancy/ service providers	Enterprise customers	Retail customers	Leader	Also running	Laggard	(Large) Enterprises	Medium-sized businesses	Small Businesses	Mature	Growth	Startup	Developed	Developing	Seeded
Company 1		x		x			x				x		x			x	
Company 2	x			x	x	x			x			x			x		
Company 3	x		x	x		x			x				x				
Company 4	x			x			x		x			x			x		
Company 5		x		x		x				x			x			x	
Company 6	x			x		x			x			x			x		

Appendix C Coding System

Top-level group	Sub group	Code	Total #
TOOLS			**39**
	TOOLS	GENERAL OPEN SOURCE GOVERNANCE TOOLS	19
	TOOLS	OPEN SOURCE LICENSE SCANNING TOOLS	17
	TOOLS	PRODUCT MODEL / ARCHITECTURE TOOLS	3
MANAGEMENT			**127**
	ORGANIZATION	OPEN SOURCE PROGRAM OFFICE	6
	ORGANIZATION	IDENTIFYING INTERNAL RESPONSIBILITIES	42
	ORGANIZATION	BACKGROUND	16
	STRATEGY	COMPLETE OSS STRATEGY	9
	POLICY	COMPLETE OS POLICY	54
PROCESSES			**131**
	IDENTIFICATION	DISCOVERY OF OSS USED IN THE CODE	8
	SUPPLY CHAIN	SUPPLY CHAIN MANAGEMENT	23
	AUDIT	AUDIT SOURCE CODE	7
	REVIEW	REVIEW	35
	APPROVAL	COMPONENT APPROVAL	8
	DISTRIBUTION	ENSURING LICENSE COMPLIANCE FOR OUTGOING PRODUCTS	16
	SELECTION	CODE SELECTION	34
SUPPLEMENTARY			**84**
	DOCUMENTATION	DOCUMENTATION AND REPOSITORY	41
	DOCUMENTATION	CONTRACTS WITH SUPPLIERS AND CUSTOMERS	11
	DOCUMENTATION	OSS QUESTIONNAIRES	6
	EDUCATION	TRAINING STAFF & COMMUNICATION	26
TOTAL CODINGS			**381**

References

Abernathy, C., Aniszczyk, C., Beda, J., Novotny, S. & Yehuda, G. Measuring Your Open Source Program's Success. Retrieved June 01, 2018, from https://github.com/todogroup/guides/blob/master/measuring-your-open-source-program.md.

Aniszczyk, C. & McAffer, J. Tools for Managing Open Source Programs. Retrieved May 15, 2018, from https://github.com/todogroup/guides/blob/master/tools-for-managing-open-source-programs.md.

Aniszczyk, C., McAffer, J., Norris, W. & Spyker, A. *Creating an Open Source Program.* Retrieved March 27, 2018, from https://www.linuxfoundation.org/creating-an-open-source-program/.

Black Duck Software. Introduction to Open Source Governance and Compliance.

Black Duck Software (2016a). *2016 Future of Open Source Survey Results,* from https://www.blackducksoftware.com/2016-future-of-open-source.

Black Duck Software (2016b). *CVEs Jump, Restrictive OS Licenses Fading, & MySQL Vulnerability,* from Copyleft licenses have become less common in use for open source projects over the last years.

Black Duck Software (2018). *Top Open Source Licenses.* Retrieved February 14, 2018, from https://www.blackducksoftware.com/top-open-source-licenses.

Black Duck Software & Bearing Point (2013). *Open Source Governance in Highly Regulated Companies.*

Chang, S., Lee, J. & Yi, W. (2010). A Practical Management Framework for Commercial Software Development with Open Sources.

Copenhaver, K., Radcliffe, M. & Vescuso, P. (2013). An Introduction to Open Source Software and Licensing.

Ellis, J. (2011). Open Source Compliance in the Supply Chain.

Fendt, O., Jaeger, M. & Serrano, R. J. (2016). Industrial Experience with Open Source Software Process Management, from https://doi.org/10.1109/COMPSAC.2016.138.

German, D. & Di Penta, M. (2012). A Method for Open Source License Compliance of Java Applications, *29*(3), 58–63. Retrieved May 2012, from http://ieeexplore.ieee.org/document/6178302/.

Haddad, I. (2016). *Open Source Compliance in the Enterprise.*

Höst, M., Oručević-Alagić, A. & Runeson, P. (Ed.). 2011. *Usage of Open Source in Commercial Software Product Development: Findings from a Focus Group Meeting:* Springer.

Jansen, H. (2010). The Logic of Qualitative Survey Research and its Position in the Field of Social Research Methods, *11*(2).

Kemp, R. (2010). Open Source Software (OSS) governance in the organization, *26(3),* 309–316.

Laurent, A. (2004). *Understanding Open Source and Free Software Licensing: Legal Impacts of Open Source and Free Software Licensing:* O'reilly.

Lindman, J., Paajanen, A. & Rossi, M. (Ed.). 2010. *Choosing an Open Source Software License in Commercial Context: A Managerial Perspective:* IEEE.

Muffatto, M. (2006). *Open Source: A Multidisciplinary Approach* (Vol. 10): Imperial College Press.

Open Chain (2016). *OpenChain Specification.* Retrieved February 01, 2018, from https://wiki.linuxfoundation.org/_media/openchain/openchainspec-1.1.pdf.

Popp, K. M. (2015). *Best Practices for commercial use of open source software.*

Schöttle, H. & Steger, U. (2015). Managing Open Source Software in the Corporate Environment: How to establish an open source license compliance program.

Silberman, G. P. (2014). A Practical Approach to Working with Open Source Software, *26*(6).

Sojer, M. & Henkel, J. (2011). License risks from ad hoc reuse of code from the internet: Software developers' reuse of code from the Internet bears legal and economic risks for their employers., *54*(12), 74–81.

Statista (2016). *Leading benefits of open-source software among enterprises worldwide as of 2016.* Retrieved May 10, 2018, from https://www.statista.com/statistics/629245/worldwide-enterprises-open-source-benefits/.

Statista (2017a). Open source software usage policy ownership distribution in enterprises 2017: Distribution of ownership/responsibility of open source software acquisition and usage policies in enterprises worldwide, as of 2017. Retrieved May 10, 2018, from https://www.statista.com/statistics/788683/worldwide-enterprises-open-source-acquisition-usage-policy-specifics/.

Statista (2017b). Presence of open source software acquisition and usage policy among enterprises worldwide 2017. Retrieved May 10, 2018, from https://www.statista.com/statistics/788668/worldwide-enterprises-open-source-acquisition-usage-policy/.

Webster, J. & Watson, R. T. (2002). Analyzing The Past To Prepare For The Future: Writing a Literature Review, *26*(2).

Yu, Y. (2013). *Software Vendor´s Perspectives on Managing Open Source Software-Involved Endeavors.*