Industry Best Practices for Corporate Open Sourcing

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

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

Gayathery Sathya
Computer Science Department
Friedrich-Alexander University Erlangen
Nürnberg
gayatherys@gmail.com

Abstract

Companies usually don't share the source code for the software they develop. While this approach is justified in software that constitutes differentiating intellectual property, proprietary development can lead to redundant development and other opportunity costs. In response, companies are increasingly open sourcing some if not all of their non-differentiating software. Given the limited academic research on this emerging topic, we bridge the gap between industry and academia by taking a practice-based approach. We investigate why and how companies engage in corporate open sourcing. We take an exploratory case study approach. Our cases are four companies with multibillion-dollar revenues each: A major e-commerce company based in Germany; a leading social networking service company based in the USA; a cloud computing software company based in the USA; and a manufacturing and media software company based in the USA. We present the resulting theory in an actionable format of state-of-the-art best practice patterns.

Keywords

Best Practice, Case Study, Corporate Open Sourcing, Strategic Open Sourcing, FLOSS, Open Source Governance, Open Source Software, Qualitative Data Analysis, State-of-the-art Practice

1. Introduction

Companies traditionally develop software behind closed doors and source code is rarely shared with other companies or with developers beyond their own organizations. This approach makes sense for the differentiating features of a company's products, because that software constitutes the core intellectual property of a company. However, other software components do not have to be kept closed. Doing so has a high opportunity cost in comparison to open sourcing, which many companies do not recognize. For example, not open sourcing can result in higher maintenance costs for a company using open source components that include certain bugs. If a company developer fixes such a bug without sharing it with the open source community, s/he would end up doing redundant work having to reapply the same fix for new releases of the same component. A better alternative would be for the company to contribute (open source) their bug fixes to the community. As a result, the new releases of the open source software would likely include the company's bug fixes, thus eliminating the extra maintenance effort on the company's part. Beyond mere contributions to open source communities, some companies create their own open source projects, where they share complete software, tools, or components developed internally. The scope of this paper is on such companies, their motivations and practices for corporate open sourcing.

As open source software and open source development gain momentum and acceptance across industries [15, 23], companies also start recognizing the value of potential collaboration across industries. One such opportunity is the collaborative software development of non-differentiating components, which can be developed and used by multiple companies. Without open source software, each such company would be forced to develop or buy the same software component to address internal needs outside of their core competencies, such as the video drivers car manufacturers use in infotainment system.

In recent years, a paradigm shift is observed in the nature of adoption of open source by commercial companies. Commercial software companies, who were initially users of open source software gradually shifted to becoming developers of open source software, paving the way for **corporate open sourcing**. In the last decade this term has taken on a meaning implying a deep link between fundamental sourcing options and strategic decisions and outsourcing strategies in particular [1]. Shaikh and Cornford [22] propose that corporate open sourcing needs to be acknowledged in a global dimension as a means of bringing together diverse and distributed human, cultural and economic resources from across the world. This shift in the nature of open sourcing results in open questions about why and how companies open source. The encompassing research question and more precise sub-questions we asked were:

RQ: Why and how do companies strategically open source software components?

RQ1: What are the motivations, goals and factors for companies to open source the software they develop internally?

RQ2: What are the state-of-the-art practices and processes companies follow when open sourcing the software they develop internally?

We started by reviewing the related literature following the methodology by Webster and Watson [24]. This resulted in the focal concepts of corporate open sourcing from the literature that we contrasted and compared with our findings. We then conducted an exploratory multiple-case case study at four companies chosen through theoretical sampling:

- **Company 1**: e-commerce company
- **Company 2**: social networking service company
- **Company 3**: cloud computing software company
- **Company 4**: engineering, manufacturing, and media software company.

Following the method by Yin [27], we gathered documentation on corporate open sourcing and interviewed employees managing and conducting open sourcing in all four companies. We then analyzed the gathered data by employing a tool for qualitative data analysis developed in our research group - QDAcity[1], which ensured traceability between data and our findings. As a result, we developed a theory of industry best practices for corporate open sourcing. Our practice-based theory identified that companies decide to open source software they develop privately, among other reasons, in order to:

- develop innovative software
- recruit talent
- develop software with better quality
- improve product visibility and branding
- develop business partnership.

Our theory addresses how companies open source theory proprietary software in order to achieve the goals outlined above. We identified state-of-the-art practices for corporate open sourcing in the following three high-level domains:

- open sourcing advocacy and coordination
- software development
- project management.

We cast our findings in an actionable format of best practice patterns and processes. By best practices in this context we mean the current best practices in the industry, that is the state-of-the-art practices. We then summarized the abstract findings of our practice-based study, while presenting some of the key findings in the form of best practices. Our practices are presented as patterns [5] with a Context-Problem-Solution structure at the core. We used a pattern structure to present the identified practices, with patterns as an abstraction from a common solution to a recurring problem in a given context. This format can enable practitioners to benefit from our research, as argued in our previous work on benefits of using design patterns in an industry context [20] and in our previous studies employing this theory presentation format [7-9]. See Table 7 and Table 8 for examples of industry best practices for corporate open sourcing we derived from our data analysis.

In section 2, we present a review of related work and literature, while identifying the key concepts, gaps and open questions. In section 3, we present our research approach and methodology, including case study preparation, case context, data gathering, analysis methods and quality assurance. In section 4, we present the research findings in our theory on industry best practices for corporate open sourcing. We present the summarized results, as well as illustrative practices of our theory. In section 5, we discuss research limitations, including threats to internal validity and external validity. In section 6, we conclude the paper.

2. Related work

Corporate open sourcing is an emerging topic in Information Systems research, which explains the limited academic research on the topic. We carefully collected and systematically reviewed the related work on the topic following the literature review methodology by Webster and Watson [24]. Our goal was to validate the research question; to understand the domain boundaries; and to identify the existing concepts around corporate open sourcing, focused on the reasons and goals behind the phenomenon, as well as the ways in which companies open source their software components. While some literature did address the motivation for open sourcing, very few authors focused on how to do corporate open sourcing. The latter was the significant gap we hope our theory will bridge.

Based on existing literature, Shaikh & Cornford [22] addressed the differences between outsourcing, insourcing, cosourcing, netsourcing, global sourcing and open sourcing. They also listed their take on the core characteristics of open-sourcing, identifying three aspects:

- Process (including communication, control, infrastructure, governance model, maintenance, distribution model, etc.)
- Product (including open source licensing, application types, quality, ownership, architecture, etc.)

[1] QDAcity - qdacity.com, qdacity-app.appspot.com

- Organization (including motivation, contributor profile, level of interest and contribution, mobility of developers, learning and training, etc.).

Shaikh & Cornford [22] suggest that open sourcing is a hybrid form of sourcing, a combination of outsourcing and open source. These characteristics suggest that open-sourcing is mostly relevant to non-core applications and services. The reasons for open sourcing only the non-core, non-competitive or non-differentiating components and services are highlighted by Lindman et al. [17]. The common reasons for avoiding open-sourcing core components are to safeguard a company's intellectual property and to gain an edge in competitive markets, even though some of these problems can be partly overcome by appropriate open source licensing.

We conducted our literature review based on the above-mentioned characteristics of open-sourcing [22] and on our research question. The first step was aimed at identifying related work. It involved conducting a systematic search on Google Scholar, ABI/INFORM Complete[2], and EBSCO's Business Source Complete[3]. Firstly, the search identified relevant documents by the presence of search terms in titles, abstract, subject and keywords. Then for areas which did not yield any result, a full text search was conducted. Table 1 presents the major search terms we used.

Table 1. Search terms used to find related work

Major Search Terms
"open sourcing" AND "strategies"
"open sourcing" AND "innovation"
"open sourcing" AND "product development"
"open sourcing" AND "commercial product"
"open sourcing" AND "outsourcing"
("business" OR "technical") AND ("open sourcing:" OR "open innovation")
("open sourcing" OR "open innovation") AND "impact"
("open sourcing" OR "open innovation") AND "intellectual property"
("open sourcing" OR "open innovation") AND "ROI" "open software development"

The search yielded documents published between the years 1991 and 2016. They included peer-reviewed journal articles, conference papers, and workshop papers. We also identified open-access white papers and essays published by IEEE Computer Society, but we only use peer-reviewed papers in our analysis.

The next step involved the analysis of the resulting papers (based on information in abstract and conclusion) and snowballing (crawling through their references to find more research literature). As a result of this, only nine research articles were identified to be relevant for detailed analysis. Based on the analysis of these relevant articles, new search terms were identified, after which we conducted a new search using the new keywords. For example, *Open Innovation* was identified to have many concepts in common with *Open Sourcing,* and they were often used in literature with a similar meaning. As a result, 17 articles were considered for final analysis. We used the detailed literature analysis to draw parallels with our research findings.

We identified the common corporate motivations and goals for open sourcing found in the surveyed literature, presented in Table 2, where the columns each correspond to an identified motivation for corporate open sourcing:

[A] develop innovative software
[B] recruit talent
[C] develop software with better quality
[D] accelerate pace of development/productivity
[E] incorporate contributions from people belonging to diverse domains and skill set
[F] improve product visibility and branding
[G] develop open standards
[H] improve return on investment (ROI)
[I] create/expand business.

Table 2. Motivation to open source in literature

Paper	Motivation to Open Source in Companies								
	[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[I]
[1]	X	X			X			X	
[2]	X					X		X	X
[4]	X	X		X					
[12]	X	X							
[17]	X						X		
[18]	X		X	X					
[19]	X	X	X	X	X		X	X	X
[21]	X								
[22]	X	X	X		X			X	X
[26]	X	X					X	X	X

[2] ABI/INFORM Collection - proquest.com/products-services/abi_inform_complete.html

[3] Business Source Complete - ebsco.com/products/research-databases/business-source-complete

[6]		X	X		X	X	X		
[10]		X	X					X	X
[15]		X				X			X
[16]		X		X				X	
[11]			X	X		X			X
[25]			X			X	X		
[14]							X		

Our case study findings confirm some of the corporate open sourcing motivations proposed by the reviewed literature, while suggesting some new ones not found in the analyzed literature.

Beyond the above-mentioned literature analysis, we also present a synthesis of some of the analyzed papers which h

ad the most relevance for our study.

Ägerfalk and Fitzgerald [1] discuss two approaches to open sourcing, namely: the liberation approach to open source a mature software product or component, and the commercialization approach to open source a product or component right from the beginning. The paper also elaborates the reasons, nature and consequences of open sourcing by companies. The authors describe that open sourcing reduces the cost of product development and caters to creativity because it involves collaboration in a community with a diversity of skills. The paper also describes some limitations involved in open sourcing such as a company being unable to force requirements, timeline, and priorities to the open community. It explains that developers may lose interest when they see an open source project as belonging to the company rather than the community and observes that the company must carefully balance its needs and the community's. To encourage this, the authors suggest the company develop an outsourcing relationship with some developers of the community and to create a position of open source program director for engaging with the community. Our best practices confirm this insight, as open source community engagement can help recruit suitable talent from the community.

Asundi, Carare and Dogan [2] analyzed the economic trade-offs associated with open sourcing. They analyzed the incentives for open sourcing by considering a conceptual model of two firms. They compare incentives between open sourcing and commercial off-the-shelf software development and also take competitive factors under consideration. They analyze all combinations like open sourcing versus proprietary development, unilateral open sourcing where one firm alone does open sourcing, and open source equilibrium where at some stage of the product life cycle multiple firms choose to contribute to a project. Based on their analysis they conclude that open sourcing increases the size of the market and that the product gains market exposure. Through our case studies, we could not confirm that open sourcing can directly impact a company's market size.

Santos et al. [21] described that innovation of products and product lines are among the main motivations for corporate open sourcing. They proposed that an open sourced project can catalyze product innovation. Based on their theoretical model, they analyzed various factors like type of license, type of user, application domain, and stage of development, which can influence the attractiveness, effectiveness, activeness, likelihood of task completion, and time to complete the tasks of a project. As a result of their analysis they proposed which types of projects should be open sourced, as well as how companies should coordinate the open sourced projects, what licensing model should be chosen, and also how to increase market visibility. Our theory confirmed that companies decide to open source expecting accelerated innovation and better market visibility.

West [25] explored the indirect benefits of open sourcing by conducting a case study of several open source projects developed by and contributed to by Apple, IBM, Microsoft and others. He identified that these companies considered product maintenance and brand visibility as major reasons to open source their software. Other factors included commoditization of extension components. They also presented their results related to the strategies the companies followed in adopting hybrid business model of whether to open source the commodity software (non-differentiating features) or opening up the technology part which the competitors cannot easily develop on their own. Our theory confirmed the mentioned reasons for open sourcing. We also built upon the identified findings on how companies should open source, casting them as industry best practices.

Gentleman et al. [6] was the only relevant article we identified related to the non-commercial domain of computational biology and bioinformatics research. The authors elicited the importance of open sourcing by researchers. Their reason for open sourcing is that, in case of complicated scientific fields, it would be beneficial to develop software by incorporating contributions from other researchers in the community, which have a diverse and complementary set of skills. Despite the scope of our theory being on the open sourcing by commercial organizations, we recognized that there were similar reasons why commercial and non-commercial organizations decide to open source, though the ways in which they do differ.

3. Research method

3.1. Case study methodology

Our research questions RQ1 and RQ2 can be best answered by studying the concept of corporate open sourcing in its native and real-life context, which dictated our choice of methodology. We followed the case study research methodology informed by Yin [27], which enabled us to study why and how companies open source. We aimed for a practice-based theory with an in-depth analysis and rich insights that can be applied by other companies looking into corporate open sourcing.

Following Yin's case study methodology we:

Step 1. identified the research question
Step 2. chose relevant research method
Step 3. identified case study design
Step 4. developed case study protocol
Step 5. selected cases from a theoretical sample
Step 6. iteratively collected data
Step 7. refined the study design
Step 8. analyzed data using appropriate tools
Step 9. derived and presented the results.

In accordance to our identified research questions, we set up an embedded multiple-case case study design wherein the corporate open sourcing is the overarching context. The units of analysis are the motivations for open sourcing (the *"why"*) when answering RQ1, and the state-of-the-art practices (the *"how"*) of open sourcing when answering RQ2. From the literature review and during the case study realization, it became evident that the *"how"* unit had further sub-units of analysis, namely the different aspects of open sourcing best practices. We developed a case study protocol as suggested by Yin [27] and using the template proposed by Brereton et al. [3].

We then selected the companies that would become the cases in our study. We selected four companies from our network of companies with advanced understanding and experience with corporate open sourcing. In order to choose a broad sample of companies, we categorized the companies in our network using the common dimensions of theoretical sampling: country (headquarters), type of customer, market position, size of company, maturity of company.

We then collected data at the selected companies, including documentation and expert interviews. To analyze the collected data we carried out a systematic qualitative data analysis (QDA) using the QDAcity tool, and then applying further techniques proposed by Yin [27] like pattern matching across cases, explanation building, and triangulation.

Based on the findings from the data analysis, a list of reasons for which companies engage in corporate open sourcing were identified. Based on the case study results we derived state-of-the-art practices that form an interconnected set of industry best practices or a handbook for corporate open source governance. These best practices cover various aspects of corporate open sourcing in the context of people, process, tools and artifacts.

3.2. Case context and data sources

The sample of the four companies in our case study includes a mix of companies with similar market positions, size, maturity, but different types of customers and geographic locations. They all are multibillion-dollar revenue companies based either in Germany or in the USA. We anonymized the company names as per their request. Table 3 gives an overview of the companies in our case study.

Table 3. Theoretical sample of case study companies

ID	HQ Country	Customer Type	Market Position	Size	Maturity
C1	Germany	Retail	Leader	Large	Mature
C2	USA	Retail	Leader	Large	In growth
C3	USA	Enterprise	Leader	Large	Mature
C4	USA	Enterprise, Retail	Leader	Large	Mature

Company 1 (C1) is an e-commerce company based in Germany and operating in many EU countries. It is an active user of open source software, and has been actively involved in corporate open sourcing. It encourages corporate open sourcing, and has internal governance structure, rules, and processes.

Company 2 (C2) is an international social networking service company based in the USA. It is an active open source user, contributor, and leader, known for creating and leading several impactful open source projects. It encourages corporate open sourcing, and has an extensive open source governance setup.

Company 3 (C3) is a cloud computing software company based in the USA, and operating globally. It is an active open source user and contributor. It has internal processes and practices for open source governance, and encourages its employees to open source their software.

Company 4 (C4) is an engineering, manufacturing, and media software company based in the USA, and operating internationally. Unlike the other companies, it has a large and diverse product portfolio. The company uses open source and shares some of its software.

To collect data at the selected companies, we conducted semi-structured interviews with expert employees at each company, sent out written questionnaires, and collected documentation on corporate open sourcing. Interview questions were restructured after the pilot case at Company 1 and fine-tuned in an iterative manner. Other data sources that we used in this study constituted internal documentation on open sourcing, websites, and internal wikis shared by case study company employees. For each company we aimed to interview an open source evangelist/advocate and a developer. Open source evangelists/advocates are mainly responsible for coordinating corporate open sourcing, internal communication and training on open source software, development, contribution, and leadership. Employees in this role also encourage open sourcing and disseminate (and sometimes shape) company strategy on open source. On the other hand, developer employees are responsible for the day-to-day aspects of open sourcing, including the development, and maintenance of the company's open sourced components, and community management for the open source projects the company leads. Our interview questions addressed the knowledge of each role. The data we collected is depicted in Table 4.

Table 4. Data sources and details

Company ID	Data Source ID	Expert Employee Role	Data Collection Method
C1	D1	Developer	Interview
C1	D2	Evangelist	Documentation (internal wiki)
C2	D3	Open source advocate	Interview Questionnaire
C2	D4	Developer	Interview Questionnaire
C3	D5	Open source advocate	Interview, Documentation
C3	D6	Developer	Interview
C4	D7	Evangelist, Developer	Interview

Data collection was performed in parallel to qualitative data analysis, which enabled iterative data analysis and collection. After the pilot project and its data analysis, we recognized that some interview questions were out of scope, while some were redundant. This helped us adjust the interview question and improve the collection of the relevant data in the next interviews.

In data analysis, we developed a codebook for QDA, based on the concepts identified during the literature review. We iteratively modified the QDA codebook during the data analysis process and once new data was coded. We explained each code in our code system with a definition. Data analysis enabled us to identify, codify and categorize the key concepts of *why* and *how* companies do corporate open sourcing. It also helped us abstract from our data and consolidate the resulting theory of industry best practices that can be applied by other companies. The final codebook included 4 code categories and 24 codes, where the code categories shaped the resulting state-of-the-art practices. All in all, we have more than 200 coding segments that serve as traces for our theory, some of which we present in the research results in Section 4.

As a quality assurance measure for our QDA, the co-authors of the paper coded parts of the data independently, and discussed their application of the codes from the codebook. This helped us clarify our understanding of the codes in our code system, as well as to adjust and better define them. It also helped us review the controversial codings in the final iteration of the QDA, as well as ensuring that we reached theoretical saturation, when neither requested coding modification or additional codings.

4. Results

Our case study resulted in us answering research questions RQ1 and RQ2, as well as formulating a set of industry best practices in the form of applicable patterns. Addressing RQ1 on *why* companies do corporate open sourcing, we identified 12 Motivations - the main factors motivating companies to open source their software. We detail our findings on why companies open source in Section 4.1, detailing the motivations we found:

[A] develop innovative software
[B] recruit talent
[C] develop software with better quality
[D] accelerate pace of development/productivity
[E] incorporate contributions from people belonging to diverse domains and skill set
[F] improve product visibility and branding
[G] develop open standards
[H] improve return on investment (ROI)
[I] create/expand business
[J] develop business partnership
[K] attain market leadership
[L] continuous code maintenance.

Addressing RQ2 on *how* companies open source their software, we identified three key categories based on our data analysis, covering respective categories of the derived state-of-the-art practices. We detail our findings on these practices, as well as example best practices from our theory in Section 4.2, going beyond the list of the categories:

- **open sourcing advocacy and coordination**
- **software development**

- **project management.**

4.1. Corporate motivation to open source

Answering RQ1, we found that corporate open sourcing is a strategic decision companies take with certain motivations and goals in mind. Two large categories of such goals are the business goals and the technical goals that companies expect to achieve by introducing the IS strategy of open sourcing.

The business goals for corporate open sourcing encompasses expected benefits of recruitment and talent acquisition, cost savings, ROI, demand creation, added customer value, product visibility, market creation/intrusion, and competitive advantage. Open sourcing the non-differentiating software gives companies a chance to set up new projects or to significantly contribute to existing open source projects, which can help recruit talented engineers who are actively contributing to the same projects, thus ensuring that these potential employees have the specialized skills required for a given job. Another reason to share source code is to save development costs, as many companies with similar needs and requirements pull together resources and develop superior software via open source projects in comparison to an alternative any one company would be able to develop on their own.

The technical goals for corporate open sourcing cover the expected benefits of innovation, skill and domain diversity, better code quality, software maintenance, open standards creation, rapid value addition and improved productivity. Open sourcing and developing certain software in an open source project improves code quality as outside developers can notice bugs or other code issues and suggest fixes. Open sourcing is also an efficient way to establish industry-wide standards, such as Android, a mobile operating system actively developed by Google as part of an open source project.

Table 5. Motivation to open source in our theory

ID	Motivation to Open Source	Data Sources
[A]	develop innovative software	D1, D2, D3, D4
[B]	recruit talent	D1, D2, D5, D6
[C]	develop software with better quality	D2, D3, D4
[D]	accelerate pace of development/productivity	D3, D4, D6
[E]	incorporate contributions from people belonging to diverse domains	D1, D2, D5
[F]	improve product visibility and branding	D2, D6
[G]	develop open standards	D7
[H]	improve return on investment	None
[I]	create/expand business	None
[J]	develop business partnership	D2, D6
[K]	attain market leadership	D5
[L]	continuous code maintenance	D6

We give an overview of the industry goals to open source coupled with the data sources from the case study companies they are based on, presented in Table 5.

As mentioned in Section 2 on related work, *Motivations H and I* have been identified in the literature, but not confirmed by our case study, while *Motivations J, K and L* have not been identified in the literature, but were derived by our case study.

4.2. Best practices for corporate open sourcing

Answering RQ2, we derived eleven common state-of-the-art practices during our case study. We developed these practices based on the analysis of data sources from more than one company. Most of these practices are also backed by the literature on the high-level. Using the Context-Problem-Solution patterns we go beyond the high-level presentation of the practices, presenting actionable details as an extension of our theory. Table 6 presents the best practices of our theory and their respective categories.

Table 6. List of industry best practices

ID	Cat.	Best Practice (Name)
BP01	People	Build Open Sourcing Clearing House
BP02	People	Build Open Sourcing Central Team
BP03	Policy	Create & Use Strategic Decision Making Policy
BP04	Process	Create & Use Open Sourcing Realization Workflow
BP05	Policy	Control Strategically
BP06	People	Deploy a Central Coordinator
BP07	Artifact	Create Flexible & Extensible

		Software
BP08	Artifact	Abstract Differentiating Features & Protect Intellectual Property
BP09	Process	Plan Small & Rapid Iterations
BP10	Tooling	Use a Centralized Dashboard
BP11	Policy	Respect License

Open Sourcing Advocacy and Coordination. Best practices *BP01, BP02, BP03, BP04* correspond to this thematic category. They cover the establishment of open source governance processes in companies to deal with the complexities of corporate open sourcing in an efficient manner, while encouraging open source contributions. These practices establish a framework and the rules for the company and its employees to follow. All these practices have the same actor Open Source Evangelist/Advocate/Coordinator. Table 7 presents an example best practice from our handbook on corporate open sourcing (full set of practices).

Table 7. Example best practice BP03

ID: BP03
Name: Create & Use Strategic Decision Making Policy
Context: Your company decided to realize the benefits of open sourcing and wants to formulate processes and guidelines required to implement open sourcing for some of its potential software components.
Problem: On what basis does the → *open source clearing house* approve a software component for open sourcing, while protecting the company's differentiating features and intellectual property? What are the guidelines that support the clearing house to review an incubated product and approve it for migration to the real open source environment?
Solution: The open-source advocate of the company in consultation with all stakeholders of open-sourcing should create a strategic decision making policy. This policy document will serve as guidelines to the → *open source clearing house* to approve and review the open-sourcing project. The enacted policies are highly company dependent and their strategy behind open-sourcing. In addition to various other factors, the policies should contain answers to the following questions: 1. What factors do we need to check to decide whether a software component can be open sourced? 2. What are the strategic motivations to open source software the component? 3. What are the common characteristics of a software component to be considered for open sourcing? 4. What are the factors related to intellectual property that should be considered? 5. Under what conditions should a software component be never open sourced? 6. How should a component be developed to be easily open sourced? What are the architecture, design and implementation concerns that a software component must be checked for? 7. What are the business factors (e.g. related to competitive features, domain knowledge or unique selling point) to consider for approving a project for open sourcing?
Traces in our data: [C1, D2] [C2, D3], [C2, D4], [C3, D5], [C3, D6], [C4, D7] **Example trace in our data**: *"If a project is a good candidate for open source, the team should know from the start that the code or hardware needs to be built in a certain way so it's robust yet easy to use. The software also needs to be able to be decoupled from internal-only infrastructure."* [C2, D4]

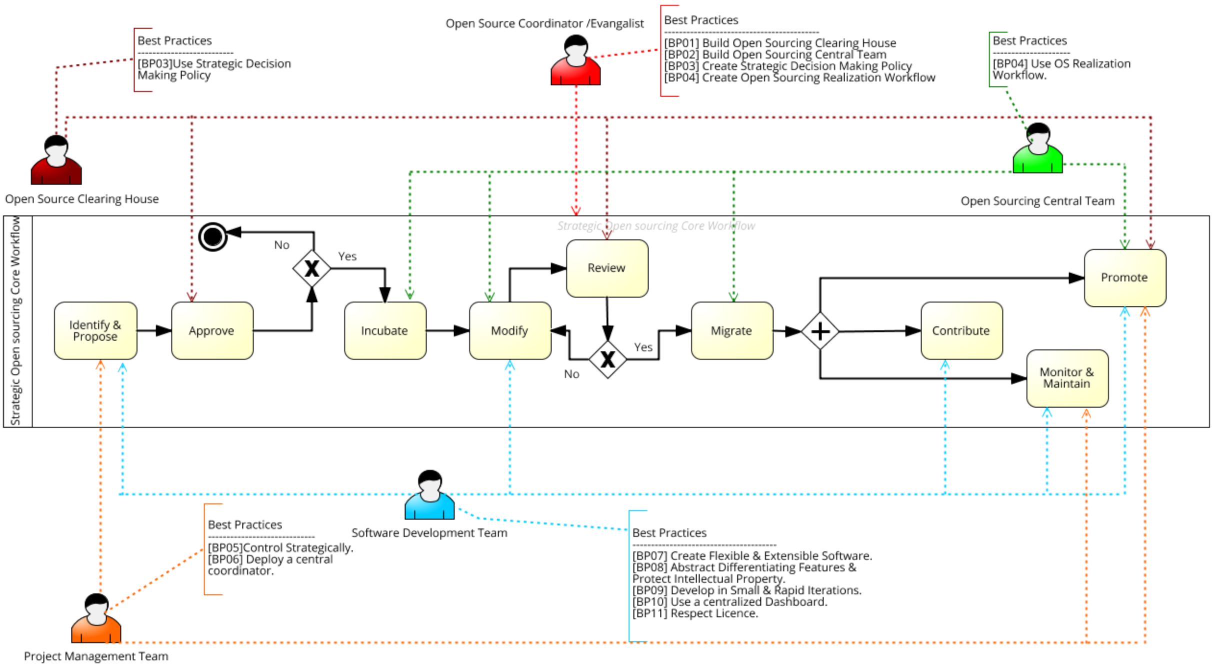
Project Management. Best practices *BP05, BP06, and BP09* correspond to this thematic category. They cover different aspects of project management that operationalize the practices on open source coordination. Companies should follow the proposed guidelines to strategically control the projects they open source, as well as to plan projects with open sourcing in mind.

Table 8. Example best practice BP06

ID: BP06
Name: Deploy a Central Coordinator
Context: The company has open sourced one or many software components and the open sourced software has contributors from both internal and external developers.
Problem: Who will coordinate various aspects of an open source community after open sourcing to manage both internal and external contributions?
Solution: The project management team should deploy a central coordinator who will manage project control, communication and other aspects between the internal and external contributors. This coordinator will enable community building. This role would be advantageous to the company to influence strategic decisions and team building. The coordinator should set up a proper communication channel and should organize regular meetups with internal and external developers to discuss community policy changes, future planning and any challenges identified in the open source community (tooling, processes).
Traces in our data: [C1, D1], [C2, D3], [C3, D5], [C4, D7] **Example trace in our data**: *"So I run our open source program [office] which helps our engineers participate in open source development who contribute projects they built internally to open source projects. We then offer [other developers] to contribute to the projects that are open sourced."* [C3, D5]

Software Development. Best practices *BP07, BP08, BP10, and BP11* correspond to this thematic category. They cover how engineers should develop and open source software, while respecting open source licenses and protecting their company's intellectual property.

To conclude, we present how the best practices are connected forming a process for corporate open sourcing of high industry relevance, presented in Figure 1.


Best Practices

[BP03]Use Strategic Decision Making Policy
Open Source Coordinator /Evangalist
Best Practices

[BP01] Build Open Sourcing Clearing House
[BP02] Build Open Sourcing Central Team
[BP03] Create Strategic Decision Making Policy
[BP04] Create Open Sourcing Realization Workflow
Best Practices

[BP04] Use OS Realization Workflow.
Open Source Clearing House
Open Sourcing Central Team
Strategic Open sourcing Core Workflow
Strategic Open sourcing Core Workflow
No
Yes
Identify & Propose
Approve
Incubate
Modify
Review
Yes
No
Migrate
Contribute
Promote
Monitor & Maintain
Best Practices

[BP05]Control Strategically.
[BP06] Deploy a central coordinator.
Software Development Team
Best Practices

[BP07] Create Flexible & Extensible Software.
[BP08] Abstract Differentiating Features & Protect Intellectual Property.
[BP09] Develop in Small & Rapid Iterations.
[BP10] Use a centralized Dashboard.
[BP11] Respect Licence.
Project Management Team

Figure 1. Process of Corporate Open Sourcing

5. Limitations

The main limitation of this research is that the results are derived based on case studies conducted across four companies only. Confirmatory future research with a widened coverage can further validate the findings presented in this paper. Furthermore, to assure the quality of the research method used, we used the Checklist for Software Engineering Case Study Research [13], following the actual case study.

Our findings regarding the motivation of corporate open sourcing were a consolidated result based on both literature survey and case study. However, the derived best practice patterns were in conjunction with only the case studies since scientific literature in this area was scarce. The best practices presented in this paper does not cover the entire spectrum of corporate open sourcing since it was limited by the scope of the case studies conducted, thus our theory does not claim to cover the topic of open sourcing entirely. Future research could help in finding best practices which can fill the gaps in this research and refine our findings.

Confirmability - the degree to which the authors are neutral towards the inquiry and their potential bias effect on the findings, is another potential limitation. Qualitative data research realized by only one researcher has inherent subjectivity and bias. In our case one co-author performed most of the QDA. Even though we followed the research method constructs carefully, there is potential bias associated with method interpretation and application. To address this, another co-author independently coded parts of the data, after which the co-authors reviewed and discussed their codings.

6. Conclusions

In this study we identified the key motivations for companies to open source based on the case study conducted. Answering the RQ1 of *why* companies should open source, we mapped the motivations to both the related literature and to the data we collected during the study. Most companies can benefit from identifying and open sourcing their non-differentiating software components, but it must be done in following certain practices and processes. Answering the RQ2 of *how* companies should open source, we developed a practice-based theory of state-of-the-art practices that form a handbook on corporate open sourcing. We gave an overview of the best practices and categorized them. We also presented two practice examples in an actionable format of Context-Problem-Solution patterns. To find more best practices we developed, check out this external link to a PDF document[4], where we also presented the appendices to this paper. We also demonstrated that the best practices we derived can be used as part of a unified process, which connects all the actors and practices into one workflow.

Further research we see on this topic can focus on a systematic approach to measure the realized versus expected benefits of corporate open sourcing, a detailed

[4] External PDF with additional results, appendices to this paper - https://faubox.rrze.uni-erlangen.de/dl/fiEzF5fpGoK2fvKdaNvAih2g/Support_HICSS_Paper_Corporate_Open_Sourcing.pdf

study of the challenges of open sourcing, as well as an extension to our theory.

7. Acknowledgments

We would like to acknowledge our case study partners for their collaboration. We would also like to acknowledge Dr. Ann Barcomb and the anonymous reviewers for their valuable feedback that helped us improve the paper significantly.

8. References

[1] Ågerfalk, P., & Fitzgerald, B. (2008). Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy. MIS Quarterly, 32(2), 385.

[2] Asundi, J., Carare, O., & Dogan, K. (2012). Competitive implications of software open sourcing. Decision Support Systems, 54(1), 153-163.

[3] Brereton, P., Kitchenham, B., Budgen, D., & Li,Z. (2008). Using a protocol template for case study planning. In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. University of Bari, Italy.

[4] Fagerholm, F., Sanchez Guinea, A., Borenstein, J., & Munch, J. (2014). Onboarding in Open Source Projects. IEEE Software, 31(6), 54-61.

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

[6] Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., & Dudoit, S. et al. (2004). Bio conductor: open software development for computational biology and bioinformatics. Genome Biology, 5(R80).

[7] Harutyunyan, N., & Riehle, D. (2019, January). User Experience Design in Software Product Lines. In Proceedings of the 52nd Hawaii International Conference on System Sciences.

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

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

[10] Haruvy, E., Sethi, S., & Zhou, J. (2008). Open Source Development with a Commercial Complementary Product or Service. Production And Operations Management, 17(1), 29-43.

[11] Hecker, F. (1999). Setting up shop: The business of open-source software. IEEE Software, 16(1), 45-51.

[12] Hippel, E., & Krogh, G. (2003). Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. Organization Science, 14(2), 209-223.

[13] Höst M. and Runeson P. (2007). Checklists for Software Engineering Case Study Research. In ESEM, 479-481.

[14] Laat, P. (2004). Evolution of Open Source Networks in Industry. The Information Society, 20(4), 291-299.

[15] Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. The Journal Of Industrial Economics, 50(2), 197-234.

[16] Lerner, J., & Tirole, J. (2005). The Economics of Technology Sharing: Open Source and Beyond. Journal Of Economic Perspectives, 19(2), 99-120.

[17] Lindman, J., Paajanen, A., & Rossi, M. (2010). Choosing an Open Source Software License in Commercial Context: A Managerial perspective. In Proceedings of 36th EUROMICRO, 237-244.

[18] Mockus, A., Fielding, R., & Herbsleb, J. (2002). Two case studies of open source software development: Apache and Mozilla. ACM Transactions On Software Engineering And Methodology, 11(3), 309-346.

[19] Morgan, L., & Finnegan, P. (2014). Beyond free software: An exploration of the business value of strategic open source. The Journal Of Strategic Information Systems, 23(3), 226-238.

[20] Riehle. D. (2011). Lessons learned from using design patterns in industry projects. Transactions on Pattern Languages of Programming II. Springer, 1-15.

[21] Santos, C., Kuk, G., Kon, F., & Pearson, J. (2013). The attraction of contributors in free and open source software projects. The Journal Of Strategic Information Systems, 22(1), 26-45.

[22] Shaikh, M., & Cornford, T. (2009). Innovating with Open Sourcing: Governance Concerns for Managers. In Proceedings of AMCIS 2009 (p. 308). San Francisco, California.

[23] Walli, S., Gynn, D. and Rotz, B. (2005). The Growth of Open Source Software in Organizations. Optaros.

[24] Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly, 26(2), 13–23.

[25] West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. Research Policy, 32(7), 1259-1285.

[26] West, J., & Gallagher, S. (2006). Challenges of open innovation: the paradox of firm investment in OSS. R&D Management, 36(3), 319-331.

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