

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

Gayathery Sathya
MASTER THESIS

Strategic Open-Sourcing in Companies: Why and How?

Intents, Processes and Best Practices

Submitted on 19. March 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.

Gayathery Sathya

Nuremberg, 19. March 2018

License

This work is licensed under the Creative Commons Attribution 4.0 International license (CC BY 4.0), see <https://creativecommons.org/licenses/by/4.0/>

Gayathery Sathya

Nuremberg, 19. March 2018

Abstract

Software Companies of the past were keen to develop proprietary software and to hold all boundaries of the software within their control. As open-source software gains momentum and acceptance across the industry, companies start understanding the opportunities underlying it and take strategic decisions to open-source some of their software components to gain business value. This paper presents various intents behind which a company can open-source strategically, a best practice workflow and a list of other best practices derived from using a case study research methodology across four global companies. The list of best practices regarding the categories 'People', 'Process', 'Policy', 'Tooling' and 'Artifact' across the domains of Open-source Advocacy, Software Development and Project Management were derived using qualitative data analysis. They are presented as best practice patterns which can be utilized by software companies in dealing with challenges during strategic open-sourcing.

Keywords

Strategic open-sourcing, open-source software, open-source advocacy, open-sourcing process, best practice, software development management, external workforce, innovation

Contents

- 1 Introduction.....5**
 - 1.1 Original Thesis Goals.....5
 - 1.2 Changes to Thesis Goals.....6
- 2 Research Chapter.....7**
 - 2.1 Introduction.....7
 - 2.2 Related Work.....7
 - 2.3 Research Question.....10
 - 2.4 Research Approach.....10
 - 2.5 Used Data Sources.....11
 - 2.6 Research Results.....12
 - 2.6.1 Intent for Strategic Open-Sourcing.....12
 - 2.6.2 Best Practices for Strategic Open-Sourcing.....14
 - 2.6.3 Process of Strategic Open-Sourcing.....16
 - 2.7 Results Discussion.....20
 - 2.8 Limitations.....21
 - 2.9 Conclusion.....22
- 3 Elaboration Chapter.....23**
 - 3.1 Literature Survey.....23
 - 3.1.1 Survey Methodology.....25
 - 3.1.2 Summary of Survey & Discussion.....26
 - 3.2 List of Best Practices.....28
 - 3.3 Additional Research Results.....33
 - 3.3.1 When NOT to Open Source?.....33
 - 3.3.2 Cause-Effect of Strategies.....33
 - 3.4 Acknowledgments.....34
- Appendix A **Case study protocol.....35**
- Appendix B **Interview Questions.....38**
- Appendix C **Researcher's Checklist.....40**
- Appendix D **Code System.....43**
- References.....45**

1 Introduction

1.1 Original Thesis Goals

The original goal of the exploratory embedded multiple-case study was to answer the following research questions: Why and How do software companies strategically open-source software components? What are the best practices to realize strategic open-sourcing in a software company?

The answer to the research question has to be derived from data obtained from case studies of three independent software companies which open-source some of its software components for a strategic intent. The case study was supposed to be carried out with two industry partners from each company, with one being in the role of an Evangelist/Open-Source Advocate and another being a core contributor in the software development team.

The initial research goal was to achieve the following milestones during 6 months.

- Step 1: Identify and Formulate Research Question
- Step 2: Conduct a systematic literature survey in the context of the research question and derive empirical results.
- Step 3: Identify cases (software companies) using theoretical sampling
- Step 4: Design data collection methods and procedures (ex. Interview Questions)
- Step 5: Realize the initial case study research (conduct interviews) and collect data
- Step 6: Analyze the data using qualitative data analysis and create preliminary code-book
- Step 7: Review data collection procedure and fine tune.
- Step 8: Realize other two case studies and execute Step 6 and Step 7 in iteration.
- Step 9 : Systematically analyze all available data
- Step 10: Integrate findings from each case and derive results based on all the cases.
- Step 11: Present the answers to the research questions with references to chain of evidence
- Step 12: Present a set of best practices for open-sourcing strategically along with appropriate references to research data
- Step 13: Present results of research to relevant stakeholders including case study participants.

The focus of this research was oriented towards the domain of strategic open-sourcing which is relatively not much explored topic according to existing literature. Scientific literature relevant to this topic are only scarcely available and some fragments of the research questions were completely unexplored in scientific literature. This was evident from the literature survey and hence the goal was to use case study research methodology to arrive at answers to the research questions.

The goal of the case study methodology was to do a theoretical sampling to identify three companies and then perform semi-structured interviews with a couple of industry partners

from each company. Another goal was to execute the case study based on methods proposed by Yin (2013) in an iterative approach to identify procedure improvements in an early stage.

The final goal was to elicitate the answers to the research questions and to formulate a set of industry best practices which are derived from research results.

1.2 Changes to Thesis Goals

The initial goal was to conduct interviews with two industry partners from each of three identified companies. Due to time constraints of one of the industry partner from one of the companies, it was not possible to conduct an interview. Instead the industry partner shared relevant internal documents which helped to answer the research questions. To overcome this limitation an interview with a fourth company was conducted and the data was included for analysis.

Another change to the original goal was that, instead of exploratory interviews with one of the companies, data collection was carried out by submitting an interview questionnaire and obtaining digital responses for those questions.

2 Research Chapter

2.1 Introduction

Though commercial product development had dominated the software industry in the past, open-source software had slowly risen to be a strong contender (Walli, Gynn & Rotz, 2005). Open Source Software (OSS) development paradigm consisted of community of volunteers working together and collaborating to develop an end-product at no cost (Hippel & Krogh, 2003; Mockus, Fielding & Herbsleb, 2002). Over time, many commercial companies and government institutions incorporated the usage of open-source software in their product development life cycle. The reason behind their usage was because of the characteristics of open-source software development.

OSS boasted unique characteristics such as reduced cost, rapid development cycles, high re-usability and freedom to choose compared to those of proprietary software (Lerner & Tirole, 2002). In recent years, a paradigm shift is observed in the nature of adoption of open-source by commercial companies. Commercial software companies, initially been users of open-source software gradually shifted to becoming providers of open-source software and organizers of OSS development paving way for 'Open-Sourcing'. This term referred to cases of commercially controlled and created software switching partially or fully to open-source licensing. This term had in the last decade taken on a significantly different meaning implying a deeper link with fundamental sourcing options and strategic decisions and outsourcing strategies in particular (Ågerfalk & Fitzgerald, 2006). Shaikh and Cornford (2008) propose that 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 characteristics unwinds an open question to product and process managers of commercial software product companies. The core question that would prevail in the minds of decision makers in these companies is the following "*Why and How do companies strategically open-source software components?*"

This paper focuses on finding the various benefits for which a company could employ open-sourcing strategically. Another goal of this research is to formulate a set of best practices for companies to deploy open-sourcing with a strategic intent. The goals of the research are attained by an initial systematic literature survey followed by data collection using an exploratory case study methodology. Using appropriate tools, an iterative Qualitative Data Analysis (QDA) is carried out on the obtained data to derive at the research results.

2.2 Related Work

Since a very long time software companies had been either users or contributors of open-source software. Lots of scientific research in this area were carried out by academicians and were published. But from our literature survey it is identified that open-sourcing software components for a strategic intent is a relatively new area and is not much addressed by the academic research community. Though a handful of literature which studies the possible strategic reasons for open-sourcing is available, there has not been any research on how to achieve it.

Based on on the characteristics of open-sourcing as identified from studies of Shaikh and Cornford (2008) a systematic literature survey was conducted using the methodology presented by Webster and Watson (2002). The survey yielded twenty one articles which were relevant

for this research. Most of the relevant research articles cited the research work authored by Ägerfalk et al. and had acknowledged that the factors proposed by these authors have a real influence on open-sourcing strategies. Some of the important papers which had the most relevance are summarized.

Ägerfalk and Fitzgerald (2008) in their paper discuss about 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 author describes that open-sourcing reduces the cost of product development and caters to creativity because of the collaboration of a community having a vast diversity in skills. The paper also describes some limitations involved in open-sourcing such as: the company cannot force requirements, timeline and priority to the open community. It also explains some conflicts that can occur between having control over project and allowing it to be open sourced. Also they explain that developers may lose interest when they see it more as a company's project and informs that there is a thin line which the company needs to handle appropriately. To encounter this, the author suggests the company to have outsourcing relationship with some developers of the community and also to incorporate a position like Open-source program director for engaging with the community. This can also help to recruit suitable talent from the community.

To attract many developers to the community and to commodify the product, the authors suggest that the company can provide their expertise of marketing and productization to market and increase the visibility of the product. Their articles express that the nature of licensing of the software components influence developer's preference to contribution and also company's control over the product future. The authors even propose that core components can still be open sourced if there is an appropriate license chosen and convey that even competitors contribute together by co-existing in the OSS community. Thus inherently there exists a set of company obligation and a set of community obligations during open-sourcing. The authors identify that cost cutting, recruitment opportunities, capability to increase innovation, requisite variety and 24/7 value addition to the product are the main strategic reasons why a company should do open-sourcing.

Asundi, Carare and Dogan (2012) analyze the economic trade-offs associated with open-sourcing. They analyze 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 vs closed-sourcing, unilateral open-sourcing where one firm alone does open-sourcing and open source equilibrium where at some stage of the product life cycle each firm choose to do open-sourcing. Based on their analysis they conclude that open-sourcing increases the size of the market and that the product gains market exposure. They also conclude that open-sourcing helps firms to reap more profits through complementary products or through reduced future cost of maintenance. They propose that since the open-source version tend to be more valuable than its closed-source counterpart because of the possibility of customization by customer and it can insulate a firm from pricing strategy of opponent. Another implication of their analysis shows that in order to stay competitive, software firms should open-source their products whenever a competitor chose to do so.

Laat (2004) studies the evolution of different kinds of open networks that had developed over time. He intimates that open-sourcing had changed some of the basic characteristics of open-source development because of involvement of commercial companies. He explains that in the past open-source software was developed collaboratively by the public without any re-

restrictions. Later, when companies started open-sourcing some of their software components for means of cheap labor and with a motive to develop the technology quickly into a stable one and then sell all branded/proprietary applications on top of it. The author discusses that, as open-sourcing evolved, it shrunk the public spectrum of the community involved by limiting access. The author explains that though this is an encouragement to those who plead for OSS because they believe it is better than COTS, this will be different or those open-source enthusiasts who subscribe to it as a movement for the free sharing of software and see it as a way of life.

Other papers which yielded relevant information are discussed in the 'Elaboration Chapter' – 3.0. The summarized results obtained from the literature survey are as follows.

1. It is identified that open-sourcing as a concept had developed since the last 15 years and only in very recent times, more and more companies are making a strategical move towards it because of visible business values.
2. Only an ample amount of research work was carried out in this area
3. Most of the existing literature provides data and research results which portray the reasons and motivation (WHY?) of companies to do open-sourcing and few of them had analyzed as to under what circumstances (WHEN?) a company should move towards open-sourcing.
4. Though there are very few preliminary theoretical models which addresses HOW to achieve some factors by open-sourcing, there is no standard methodology or strategy that were put forward in research literature.
5. Except some articles which suggest licensing methods to safeguard intellectual property, all other research work suggest to open-source only non-differentiating non-competitive software components.
6. A concept matrix was constructed based on the data obtained from the research articles. A consolidated view of concept matrix showed that the motivational factors for strategic open-sourcing can be put under three macro heads, namely: Technical, Business and Marketing. There is also a Legal perspective attached to it but it is mainly a means to achieve these strategies and by themselves are not motivating factors. These macro factors can further be seen to be formed of some micro strategical elements. A detailed view is portrayed in Table 1.

Macro Factors	Micro Factors
Technical	Innovation, Skill & Domain diversity, Quality & Maintenance, Rapid value addition, Productivity
Business	Recruitment (talent acquisition), Cost of Savings (ROI), Demand Creation (Customer Value)
Marketing	Product Visibility, Market creation (intrusion), Open Standards

Table 1. Macro and Micro strategical factors influencing open-sourcing

A deeper look into the factors makes it apparent that the Technical and Business factors are the direct factors and Marketing is an indirect factor of motivation. All the micro elements of Marketing namely product visibility, market creation and open standards are indirectly meant to increase revenue by creating demand which is a direct motivation of the business factor.

In Section 3.1 – 'Literature Survey', the results of literature review are presented in greater detail with references to studies from relevant research works.

2.3 Research Question

The core question of the research is

“Why and How do Companies Strategically Open Source Software Components?” and “What are the Industry Best Practices for Strategically Open-sourcing Software Components”

The goals of the paper is

- To present the major strategic reasons for which companies do open-sourcing.
- To present relevant processes on how companies do open-sourcing.
- To summarize the research results into a set of best practices and present them as abstract guidelines for software companies which open-source software components for a strategic intent.

2.4 Research Approach

This paper employs case study research methodology to arrive at the findings for the research questions. The case study approach was chosen based on studies from Yin (2013) wherein this method is proposed when research questions are in the form of 'How?' and 'Why?' and when investigator do not have control of behavioral events.

The case study methodology is followed similar to the proposal of Yin (2013) by following the steps: identifying the research question, choosing relevant research method, identifying case study design, develop case study protocol, case selection, iterative data collection and design refinement, data analysis using appropriate tools and deriving results.

The research methodology relevant to the leading research question falls in the category of embedded multiple case design described by Yin (2013) wherein the strategic open-sourcing is the context and 'Why?' and 'How' being the unit of analysis. From the literature survey and from the subsequent case studies it was evident that the unit 'How' had further sub-units of analysis. A case study protocol as suggested by Yin (2013) was developed according to the template proposed by Brereton, Kitchenham, Budgen and Li (2008). Appendix A contains the complete case study protocol.

Case selection was achieved based on theoretical sampling formulated depending on the following dimensions

1. Type of software company
2. Type of customer
3. Market position
4. Size of company
5. Maturity of company
6. Maturity of product

These dimensions were specifically chosen to identify companies that have products and processes which are mature enough to be used as an object for research.

Data collection was executed through interviews with relevant industry partners identified through theoretical sampling. Interview questions were restructured after the pilot case and fine tuned using an iterative approach based on review. The used data sources are explained in

section 2.5 and the interview questions are listed in Appendix B. Data analysis was carried out by doing a systematic Qualitative Data Analysis (QDA) using QDAcity tool and then applying further techniques proposed by Yin like pattern matching across cases, explanation building and triangulation. The code book that was used to perform QDA is presented in Appendix-D.

Based on the findings from the data analysis, a list of reasons for which a company strategically open-source software components were identified. Based on the results a handbook of best practices for strategic open-sourcing was developed. These best practices cover various aspects of strategic open-sourcing in the context of people, process, tools and artifacts.

In the motive of assuring quality of the research, a Checklist for Software Engineering Case Study Research (Höst & Runeson, 2007) was filled and is available in Appendix C

2.5 Used Data Sources

The data sources for the research were semi structured interviews with relevant industry partners who were identified using theoretical sampling approach. Interviews were conducted with partners from four companies. All the four companies were either large or medium scale companies. These companies as well as their products were mature and were leaders in their respective domains. These companies had either partial or complete open sourced business model and catered to enterprise and retail customers. Table 2. shows the respective profiles of those companies.

Companies\Dimensions	Customer Type	Market position	Market capitalization	Company/Product Maturity
Company 1	Retail customer	Leader	Large	Mature
Company 2	Retail customer	Leader	Large	Mature
Company 3	Enterprise customer	Leader	Medium	Mature
Company 4	Enterprise and Retail customer	Leader	Large	Mature

Table 2. Industry partner sampling

Other data sources that were used for the research constituted documents, websites and wikis shared by industry partners during (or instead of) interviews. In certain cases where exploratory interview was not possible, data collection was done by submitting a interview questionnaire to the partner and getting responses for them. For each company a open-source evangelist/advocate and a developer were interviewed. Details of data collection are depicted in Table 3.

Company	Data Source Reference	Industry Partner	Data Collection method
Company 1	Data Source1	Developer	Interview
	Data Source2	Evangelist	Reference documents, wiki
Company 2	Data Source3	Open-source advocate	Interview Questionnaire
	Data Source4	Developer	Interview Questionnaire
Company 3	Data Source5	Open-source advocate	Interview, Reference documents
	Data Source6	Developer	Interview
Company 4	Data Source7	Evangelist (also Developer)	Interview

Table 3. Data collection details

The first case was realized through Company 1, which is an electronic commerce company based in Germany. The first interview was with a core software developer of a major product of the company which was open sourced strategically. The second interview which was planned with the evangelist of Company 1 was not feasible because of time constraints of the industry partner but the evangelist shared reference documents and wiki sites which provided responses for most of the intended interview questions.

The second case is with Company 2, which is a global leader in online social networking and is based in USA. Data collection from the lead open-source advocate and core developer of a major open sourced software component was achieved using interview questionnaire.

The third case is with Company 3, which is an American cloud computing software company with major focus on customer relationship management. Data collection from an open-source advocate and a core developer was achieved through exploratory interviews and additional reference documents and web contents were shared by them.

The fourth case is with Company 4, which is a multi-national American software company with products ranging from internet application software development, content and image editing applications to cloud applications. An exploratory interview was conducted with the Open-source director of the company who also had a dual role of a developer.

2.6 Research Results

My research results are presented in three segments in relevance to the research question. The following are the three segments under which the research results are presented.

1. Intent for Strategic Open-Sourcing (Why?)
2. Best practices for Strategic Open-Sourcing (Best Practice Patterns)
3. Process of Strategic Open-Sourcing (How?)

2.6.1 Intent for Strategic Open-Sourcing

From my research, it is derived that software companies open-source some of the software components with a strategic intent. An ample number of strategic open-sourcing motives were identified through literature survey and through data analyzed from case studies. These strategies were prioritized based on their importance described in the case studies and specified in the literature. Additionally, it is also based on frequent mentions of these strategies within the

data that was analyzed for the research. The intents behind strategic open-sourcing are listed in their order of priority in Table 4.

Strategic Intent	References
To develop innovative software	Case Study: Data Source 1,2,3,4; Literature: (Ågerfalk & Fitzgerald, 2008; Asundi, Carare & Dogan, 2012; Ebner et al., 2009; Fagerholm & Guinea, 2014; Hippel & Krogh, 2003; Lindman, Paajanen & Rossi, 2010; Mockus et al., 2002; Morgan & Finnegan, 2014; Santos, Kuk, Kon & Pearson, 2013; Shaikh & Cornford, 2009; West & Gallagher, 2006)
To recruit talent	Case Study: Data Source 1,2,5,6; Literature: (Ågerfalk & Fitzgerald, 2008; Ebner et al., 2009; Fagerholm & Guinea, 2014; Gentleman et al., 2004; Haruvy & Sethi, 2008; Hippel & Krogh, 2003; Lerner & Tirole, 2002, 2005; Morgan & Finnegan, 2014; Shaikh & Cornford, 2008; West & Gallagher, 2006)
To develop software with better quality	Case Study: Data Source 2,3,4; Literature: (Gentleman et al., 2004; Haruvy & Sethi, 2008; Hecker, 1999; Mockus et al., 2002; Morgan & Finnegan, 2014; Shaikh & Cornford, 2009; West, 2003)
To accelerate pace of development/productivity	Case Study: Data Source 3,4,6; Literature: (Fagerholm & Guinea, 2014; Hecker, 1999; Lerner & Tirole, 2005; Mockus et al., 2002; Morgan & Finnegan, 2014)
To incorporate contributions from people belonging to diverse domains and skill set	Case Study: Data Source 1,2,5; Literature: (Ågerfalk & Fitzgerald, 2008; Gentleman et al., 2004; Morgan & Finnegan, 2014; Shaikh & Cornford, 2008)
To improve product visibility and branding	Case Study: Data Source 2,6; Literature: (Asundi et al., 2012; Gentleman et al., 2004; Hecker, 1999; Lerner & Tirole, 2002; West, 2003)
To develop business partnership	Case Study: Data Source 2,6
To develop Open standards	Case Study: Data Source 7; Literature: (de Laat, 2004; Gentleman et al., 2004; Lindman et al., 2010; Morgan & Finnegan, 2014; West, 2003; West & Gallagher, 2006)
To attain Market leadership	Case Study: Data Source 5
To improve Return on Investment (ROI)	Literature: (Ågerfalk & Fitzgerald, 2008; Asundi et al., 2012; Haruvy & Sethi, 2008; Lerner & Tirole, 2005; Morgan & Finnegan, 2014; Shaikh & Cornford, 2009; West & Gallagher, 2006)
To create/expand business	Literature: (Asundi et al., 2012; Haruvy & Sethi, 2008; Hecker, 1999; Lerner & Tirole, 2002; Morgan & Finnegan, 2014; Shaikh & Cornford, 2008; West & Gallagher, 2006)
Continuous code maintenance	Case Study: Data Source 6

Table 4. Intent behind Strategic Open-Sourcing (in order of priority)

From the research, it is derived that the five major intentions behind a company's open-sourcing strategy are to develop innovative software, to recruit talent, to ensure better quality in their products, to accelerate the pace of development and to incorporate contributions from diverse domains and skill sets.

From the results it is evident and ensured that these strategic intents can be achieved through open-sourcing. These intents could be guidelines to software companies in deciding whether they need to open-source their software components. Discussions on these results are presented in the Results Discussion in Section 2.7.

2.6.2 Best Practices for Strategic Open-Sourcing

Based on the literature survey and subsequent case study, the following domains were considered for eliciting the best practices.

1. Open-sourcing Advocacy and Coordination
2. Software Development
3. Project Management

The best practices are categorized based on their relevance to different elements of an organization namely: People, Process, Policies, Tooling & Artifacts.

The list of best practices were derived based on Data Source 1,2,3,4,5,6,7 (Refer Table 3) and work of Ägerfalk and Fitzgerald (2008)

The list of best practices which were derived from this research are presented in Table 5.

Ref No	Category	Name
BP01	People	Build Open-Sourcing (OS) 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 and 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

Table 5. List of Best Practices

A typical description of a best practice contains the following elements

1. Reference Number (Ref No) – It is an unique identifier to refer an best practice (eg. BP01)
2. Name – A descriptive name of the best practice
3. Category - The category of the best practice with respect to its characteristics and its relevance to organizational elements. It could be one of the following: People, Process, Policy, Tooling and Artifact

4. Context – It provides the scope of the system environment in which the best practice is applicable
5. Actor – It provides information on stakeholders who are responsible to adhere to the best practice
6. Problem – This parameter provides the questions to which the best practice provides solution to.
7. Solution – The actual description of the best practice.

A couple of best practices are illustrated in Table 6 and Table 7 which shows an example of how a best practice is described. The complete set of derived best practices are presented in detail in Section 3.2 – 'List of Best Practices'.

Ref.No/Name	BP01 - Build Open-Sourcing (OS) Clearing House
Category	People
Context	The company is planning to strategically open-source some of its software components
Actor	Open-source Evangelist / Advocate / Coordinator
Problem	Who will decide whether a software component can be open sourced? Who decides whether open-sourcing a particular component will realize a strategic intent and reap benefits? Who will ensure that different aspects of open-sourcing are in their respective form and structure.
Solution	<p>The Open-source advocate of the company should build a Open-Sourcing Clearing House which is a decision making body. This is a cross-functional team and is responsible to provide a mechanism for distributing governance of OSS development and decision making more broadly across the organization. The stakeholders of the group would consist of members from the following:</p> <ol style="list-style-type: none"> 1. Engineering team 2. Delivery Heads 3. Product security team 4. IT Compliance Team 5. Legal Team 6. Tooling team 7. Product Management Team <p>A recommendation would be to include odd number of members to ensure decisions. The major responsibilities of this team would be the following.</p> <ol style="list-style-type: none"> 1. Take decision on whether to open-source a software component. This decision should be based on a well-defined policy as put forward by best practice BP03 “Create & Use Strategic Decision making policy”. 2. Define action points for a selected component in its incubation phase. 3. Review whether an incubated software component is ready for migration to actual open-sourcing environment. 4. Patent, License and Compliance clearing. 5. Meet on a regular basis to approve/reject projects awaiting public release. 6. Manage priority list of projects to be open sourced. 7. Timely inform policy evolution to respective stakeholders.

Table 6. Best practice – BP01 Build Open-Sourcing (OS) Clearing House

Ref.No/Name	BP03 – Create & Use Strategic Decision making policy
Category	Policy
Context	The company has decided to realize the benefits of open-sourcing and want to formulate processes and guidelines required to implement open-sourcing in some of its potential software components.
Actor	Create - Open-source Evangelist / Advocate / Coordinator; Use – Open-Sourcing Clearing House
Problem	On what basis does the OS clearing house approves a software component for open-sourcing? What are the guidelines that support the clearing house to review an incubated product and approve it for migration to real open-sourcing environment?
Solution	The open-source advocate of the company in consultation with all stake holders of open-sourcing should create a strategic decision making policy. This policy document will serve as guidelines to the OS clearing house (Ref BP01) 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 are to be checked to decide whether a software component can be open sourced? 2. What are the strategic intents that are permissible to open-source a software component? 3. What should be the nature of the software component to consider it 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. What are the architecture, design and implementation concerns that the software component be checked against? 7. What are the business factors that should be considered for approving a project for open-sourcing? (ex. Factors related to competitive features, domain knowledge & Unique Selling Points)

Table 7. Best Practice – BP03 Create & Use Strategic Decision making policy

2.6.3 Process of Strategic Open-Sourcing

From the data collection phase it was identified that other than Company 1, all other companies do not follow a predefined workflow to open-source strategically. But during data analysis through pattern matching and explanation building it is derived that though companies do not follow a predefined workflow, the pattern of process workflow that they follow to open-source is very similar across the cases.

The outcome of the data analysis of my research is a best practice process workflow that each company could follow to efficiently open-source their software components strategically. The best practice process workflow is the Strategic open-sourcing Core Workflow (SOSCW) and is presented in Figure 1.

The SOSCW can be seen to be consisting of three phases. An initiation phase of identifying a software component, analyzing it for its abilities to be open sourced and approving it. The next phase is a preparatory phase wherein the software component is prepared for being deployed in an open-sourcing environment and the third phase is a continuous development phase where in the deployed component encounter planned value addition while being monitored and controlled strategically.

The various processes in the workflow are detailed in Table 8 representing details regarding actors who are responsible for executing the processes, their corresponding actions to complete the process and relevant best practices that need to be followed when performing those actions. The best practices referred in this section are explained in detail in Section 3.2- 'List of Best Practices'

Process	Actors	Actions	Relevant Best Practices (Refer Section 2.6.2)
Identify & Propose	Members of Product Management or Software Development team	<ol style="list-style-type: none"> 1. The responsible members should identify potential software components that can reap a strategic benefit and propose it to the open-source clearing house (Ref-BP01) for approval. 2. It is advisable to propose only components that can provide a strategic benefit and should not be based on personal wishes. 3. The Open-source central team (Ref-BP02) using defined workflows (Ref-BP04) will help the stakeholders to prepare information needed for proposal according to protocols put forth by the clearing house. 	BP01, BP02, BP04
Approve	Open-Source Clearing House	<ol style="list-style-type: none"> 1. In this process the clearing house analyses various aspects of the proposed software component and decides whether it can be open sourced. 2. To achieve this process, the clearing house uses the strategic policy guidelines set forth by the open-source evangelist/coordinator (Ref BP03). 	BP01, BP03
Incubate	open-sourcing Central team	<ol style="list-style-type: none"> 1. In this process the OS central team incubates an approved software component to execute preparatory measures on the component. 2. The incubation is done in a company specific incubation environment (ex. Github Enterprise) 3. The realization of incubation is achieved based on workflows resulting from BP04 	BP02, BP04
Modify	Software Development team with the help of open-sourcing Central Team	<ol style="list-style-type: none"> 1. In this process the software component is modified to adhere to open-sourcing guidelines enforced by BP03. 2. The OS central team helps the stakeholders to achieve the necessary modifications in the component. 3. In this process, the component is detached from their dependencies and altered by applying best practices BP07, BP08, BP11. 4. The outcome of this process would be a software component which is ready for migration to actual open-sourcing environment on approval by the review team. 	BP02, BP03, BP04, BP07, BP08, BP11
Review	Open-Source Clearing House	<ol style="list-style-type: none"> 1. In this process the clearing house based on guidelines put forth by BP03 reviews the modified software component. When the review is satisfactory the component is approved for migration to actual open-sourcing environment or else it is reverted back to 'Modify' process of the workflow. 2. This group reviews the software project with respect to patents, licenses, compliance, contribution model and so on. 3. The review is not only with respect to the artifact but also in regard to all the processes that is expected to interact with the artifact once it is open sourced. For example processes related to bug tracking, contribution tracking, code reviews, code syncing, tools, workflows, communication platform and documentation. 	BP01, BP02
Migrate	Open-sourcing Central team with the help of	<ol style="list-style-type: none"> 1. In this process, the OS central team migrates the approved software component to the actual open-sourcing environment (most frequently 	BP02, BP04

Process	Actors	Actions	Relevant Best Practices (Refer Section 2.6.2)
	stakeholders from development and management teams	<p>www.github.com/company/projectname)</p> <p>2. This process is carried out based on workflows resulting from BP04.</p> <p>3. Migration process not only includes software artifacts but also documentations, license agreements and other relevant policy documents.</p>	
Contribute	Software Development team	<p>1. In this continuous process, value addition is contributed by both internal and external contributors.</p> <p>2. From the perspective of internal contributors, it is important to adhere to best practices “Create Flexible and Extensible Software”, “Abstract differentiating features & Protect Intellectual Property” and “Respect Licence”.</p>	BP07, BP08, BP11
Monitor & Maintain	Project Management team	<p>1. This is a continuous process and is best achieved by deploying a Central coordinator who acts as an central interface between the internal and external environments.</p> <p>2. Decision making in this process needs to be taken strategically considering both internal products of the company and the components that are being open sourced. Corresponding trade-offs need to be done based on strategies.</p>	BP05, BP06, BP09, BP10
Promote	All employees and stakeholders relevant to the Open-Sourcing venture	<p>This process refers to promotion of the open sourced component by the employees in all possible ways. Promoting is achieved by speaking about it in conferences, building communities around the environment, blog posts and social media.</p>	

Table 8. Strategic Open-Sourcing Core Workflow (SOSCW) description

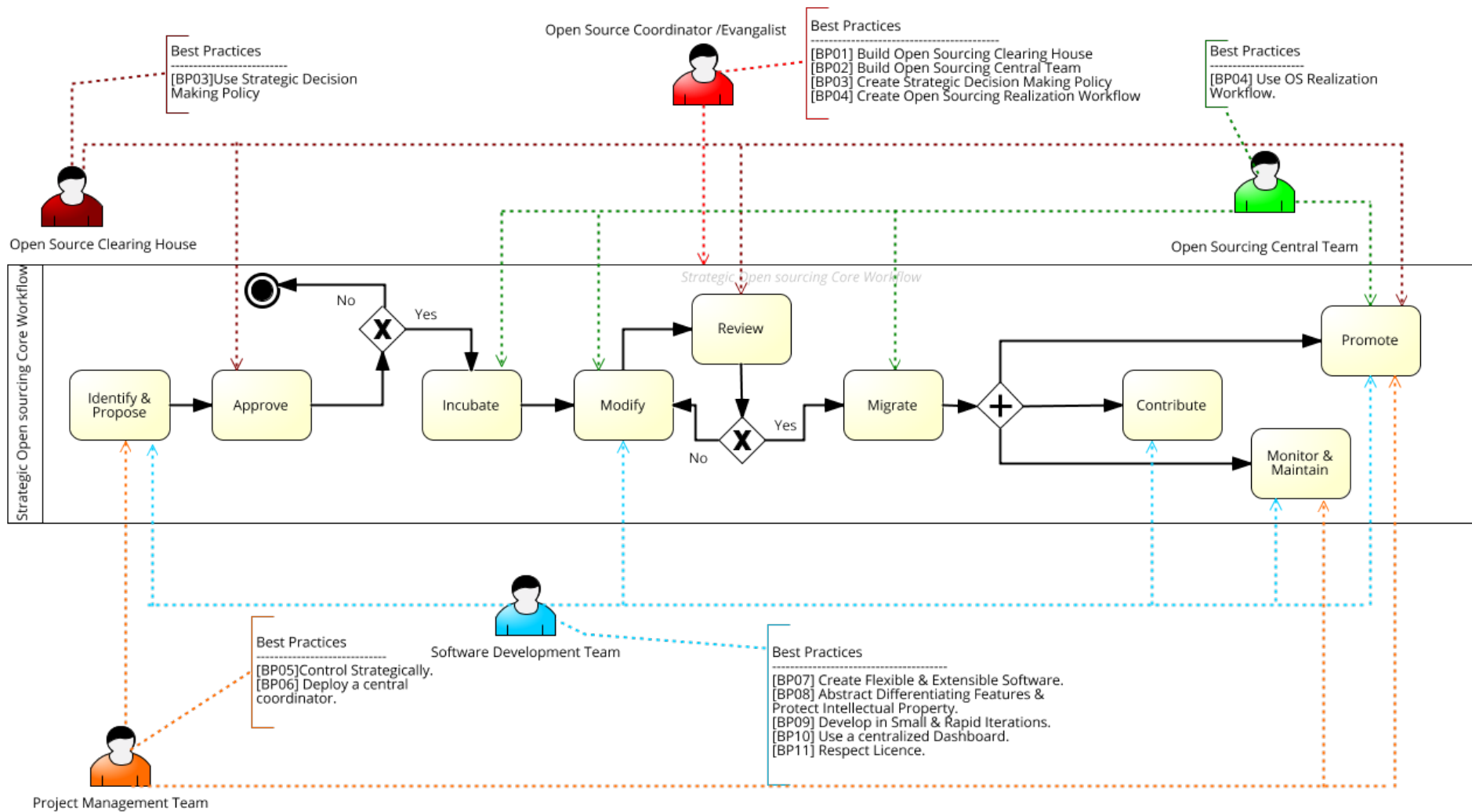


Figure 1. Strategic Open-Sourcing Core Workflow

2.7 Results Discussion

The various intents derived from the research can be categorized as the ones which can be quantified and the others that cannot be quantified. For example, strategic intents like 'to recruit talent', 'to accelerate productivity', 'to attain market leadership' and 'to improve ROI' can be quantified and hence can be compared with expected results or outcomes that were achieved without open-sourcing. But other intents like 'to develop innovative software', 'to develop software with better quality' are qualitative intents and hence would be difficult for a company to measure the outcomes of open-sourcing.

From the research, it is identified that the most widely incorporated strategy is to develop innovative software which lies in the qualitative domain. The extent of achievement of this strategy (or for this basis, any of the strategy) was not measured by any of the companies but were inherently felt and accepted by all the industry partners. In case of intent of recruiting talent which is the second most widely incorporated strategy as per the findings, the realized benefits are quantifiable. Though companies did not have a system to measure the realized benefits, the industry partners were able to provide some numbers related to recruitment from external contributors of the open-sourced component.

Because of the above reasoning, it could be possible that 'to recruit talent' is the most widely used strategy. But, since this reasoning is based on perception, the intent to innovate is still presented in this thesis as the top priority for open-sourcing. When future researches yield a proper approach to measure these outcomes, their relative importance can be identified exactly.

Another finding from the research is that except for few of the intents identified, the remaining intents are directly or indirectly pegged with improving business and thereby improving return on investments

The best practice workflow presented in this paper is a generic representation of processes followed by all the companies included in the case study. It is important to note that all the companies were open-sourcing or interested to open-source components which were already in production. From literature it was identified that both approaches of open-sourcing from beginning and open-sourcing the ones already in production could be followed (Ägerfalk & Fitzgerald, 2008).

But the approach of open-sourcing components right from beginning was not practiced by any of the industry partners in my case studies. This finding could have also been a result of having only four companies in our case study. A future research having a wider coverage of companies could derive the existence of both approaches too.

In all the case studies, the companies chose GitHub as the platform to do open-sourcing. The companies used GitHub Enterprise to incubate the software component within the company and then apply required modifications and later moved it to GitHub to make it open. Some of the reasons behind their usage of GitHub were identified to be the following: Forking-which enabled to manage internal and external component deliveries, communication platform for contributors, presence of various tools for collaboration, monitoring and tracking, better maintenance of documentation and wikis.

The other key points derived from the research are:

1. Most of the industry partners believe that innovative ideas can be implemented in their software by opening out the product to the public. Further, it is evident from the re-

search that software developed through open-sourcing is more innovative than being developed in-house.

2. Though one of the strategies of some of the companies in case study is to recruit talent through open-sourcing, they do not have a formal approach to recruit from the pool of external contributors. The recruitment happens through an informal approach with the internal team identifying unique and extraordinary talent from the external pool and then recruiting them officially. The reverse approach of external contributors expressing their willingness to join the company and then being recruited by the company is also evident.
3. None of the companies had an automated workflow for open-sourcing but the stakeholders believed that open-sourcing could be more efficient if relevant workflows are automated.
4. Companies strategically open-source to gain some business or technical value. But, the companies do not have a system in place to measure the realized benefits of open-sourcing. Case studies also reveal that though a company open-source for a particular strategy, they also reap other benefits outside the scope of their strategy.

As a whole, the case studies revealed the following;

- open-sourcing for a strategic purpose yields both expected and unplanned benefits
- Though open-sourcing decision involves company-wide modifications the advantages that the company can leverage are high
- Adhering to best practices and automation of workflows can make open-sourcing efficient while mitigating the corresponding risks.

Additional results obtained through this research are presented in Section 3.3

2.8 Limitations

The main limitation of this research is that the results are derived based on case studies conducted across four companies. A confirmatory future research with a widened coverage can further validate the findings presented in this paper.

The findings regarding the intents of strategic 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 are scarce. The best practices presented in this paper does not cover the entire spectrum of strategic open-sourcing since it was limited by the scope of the case studies conducted. Future research could help in finding best practices which can fill the gaps in this research and could also refine the existing findings.

2.9 Conclusion

The strategic intents derived from the case studies can be driving factors for software development companies to identify and open-source their non-differentiating software components. Further, the Strategic Open-Sourcing Core workflow presented in this thesis can be used by these software companies as guidelines to set up their open-sourcing environment. The eleven best practices presented in this paper were derived from the case studies on companies which had already deployed open-sourcing. Hence these best practices refer to common industry practices and can be adhered by software companies to efficiently handle challenges emanating from open-sourcing.

From the research it is also evident that companies need a systematic approach to measure the realized vs expected benefits of open-sourcing. A methodology to measure such strategic benefits of open-sourcing could be a scope for future research.

3 Elaboration Chapter

3.1 Literature Survey

A systematic detailed literature survey was conducted to validate the research question and to understand the domain boundaries and to search for relevant answers. To obtain a deeper insight to the research question, one needs to understand the boundaries of open-sourcing and also its differentiating characteristics with other forms of sourcing. (Shaikh & Cornford, 2008) had from existing literature accumulated these differences between Out-sourcing, In-sourcing, co-sourcing, Net-sourcing, Global-Sourcing and Open-Sourcing. They also have listed out the core characteristics of open-sourcing which would be of importance in the quest for answer to the research question.

The core characteristics are focused on three aspects namely: Process, Product and Organization.

Table 9 re-lists those characteristics as described in their paper.

Aspect	Open-Sourcing Characteristics
PROCESS	
Communication	Combination of face to-face and online
Control	Combination of control mechanisms used ranging from very centralized to quite distribute and Informal.
Infrastructure	Built on both global internal infrastructure and internet
Governance Model	Combination of hierarchical (and client based Governance structure matching) and open source democratic style.
Maintenance	Combination of clear phase and evolutionary type of development and after sales service.
Distribution Model	Mix of internal closed channel with the use of the Internet (where companies often take from this latter channel but don't pour back their now copyright idea/product).
Total Cost of Ownership (TCO)	Depending on the open-sourcing route taken companies would face varying fuzziness of TCO. TCO is clearer in open-sourcing than OS as some elements must be quantified for a company wishing to sell its product/service.
Marketing	Global marketing strategy but also based on expertise of the company – however also rely on word-of-mouth and forums.
Transparency process	Partly transparent – depends on license. If dual license then good level of transparency in process.
Development model	Combination of planned and clear phase production with agile methods and beta testing.

Aspect	Open-Sourcing Characteristics
PRODUCT	
License	Dual licenses and some only OS or proprietary.
Application Type	Non-core applications and services but open sourcing model used as a strategic device and propped up as strong PR for the Company.
Core/Product/Idea quality	SLA specifies quality level so quite effective yet made even better when both product and process are transparent and open to scrutiny.
Ownership	Partly owned by the collective and some strands of the product/service owned by a company.
Architecture	Early part of the product/service is open but the final product is often closed architecturally.
Level of documentation	Documentation of good quality and detail is a must and this counters a serious problem companies have with OS products/services.
After sales service	Companies often bridge between the community and client and ensure that after sales services are provided through the company but tapping into community expertise.
Reusability of code/idea/product	Good reusability level as company usually releases much of the design or product back into the community. Indeed this is encouraged as a way to build trust between company and community.
COMMUNITY/ORGANIZATION	
Motivation	Community and company link building to retain expertise over time, promotion of product and company, and sustaining the community and its members.
Contributor profile	Global as the OS community it links too is very global but the company members are often mostly from one country, paid employees.
Level of interest and contribution	Good level of interest and contribution may be small but is consistent.
Mobility of developers/creators	Very good mobility between company sites, client sites and OS community.
Access to learning / training	Employees/members have both training and collaborative learning resources. Training is provided by the company and all members have access to a global community of OS experts outside the company walls.
Size of community	A mix of both company employees and access to the global workforce accessible through the OS community.

Aspect	Open-Sourcing Characteristics
Sanctions on rule breaking	Mix of informal and legally enforceable sanctions.
Status of core developer/creator group	It is a position of control and authority yet there is some measure of answerability to the company.
Global distribution	Beneficial combination of resources which gives rise to a considerably globally distributed presence.

Table 9. Characteristics of Open-sourcing (Shaikh & Cornford, 2008)

These characteristics suggest that open-sourcing is a strategic way of 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 and non-competitive components and services are explained and implied in research papers (Lindman 2010; Wen, 2015). Though the common reasons for avoiding open-sourcing core components are to safeguard intellectual property and to gain lead in competition, some of the research articles claim that these problems can be partly overcome by appropriate licensing.

3.1.1 Survey Methodology

Based on the characteristics of open-sourcing and the leading question, a literature survey was conducted following the methodology as described in (Webster & Watson, 2002). The first step was aimed at identifying related work. It involved conducting a systematic search on Google Scholar, ABI Inform, EBSCO and Business source complete. 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. Some of the major search terms used for searching is listed in Table 10.

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”

Table 10. Major search terms used in identifying related work

The search yielded documents published between the years 1991 and 2016. The resultant documents were contributed by journal articles, conference papers, submissions to workshop, open access white papers and essays published by IEEE computer society. The next step involved analysis of the resultant documents (based on information in abstract and conclusion) and crawling through their references to find more research literature. As a result of this, only 9 research articles were identified to be relevant for detailed analysis. Based on analysis of these relevant articles, new search terms were identified and search was conducted again us-

ing those keywords. For example Open Innovation was identified to have so many concepts in common with Open-Sourcing and they were often used in literature under the same meaning. As a result, 21 articles were considered for final analysis.

3.1.2 Summary of Survey & Discussion

Other than the papers discussed in Section 2.2 – 'Related works', the following are the content from academic literature which yielded relevant information.

Hecker (1999) of Netscape in his essay in IEEE software proposes that companies can choose to make source code freely available and still serve its own business interests as a for-profit organization. The author also proposes that when a company makes the right products open-source and chooses an appropriate business model, it can ultimately benefit in ways that more than offset any short-term loss of profits that might stem from no longer being able to sell those products in the traditional way. The author suggests that open-sourcing is cost effective and is useful to improve code quality and with proper licensing strategy can avoid improper code modifications. The author also proposes that open-sourcing can help to sell complementary products and services, widget frosting (selling hardware with open-source software) and low costs for product maintenance.

West and Gallagher (2006) in their paper affirm that open-sourcing assists in revenue generation by creating a market for complementary products by means of Standards and can also provide value to customer. The authors mean that open-sourcing is an option for triggering open innovation in product lines. They also propose a couple of strategies. One of them is to sharing research and development activities with companies having vertical support and the other one being the open-sourcing of internal development which can generate revenue on complementary products. Another major strategy proposed by these authors is to open-source software components that are not yet commercialized and that are not differentiating.

Shaikh & Cornford, (2008) in their paper explore the evolution of different types of sourcing and had shown how the characteristics of this different type of sourcing had over time converged to open-sourcing. They also analyze the characteristics of open-sourcing and suggest when and why companies should do open-sourcing. They propose two business models based on which open-sourcing happens, namely demand (Product) and supply (Process) focused models. Based on these models they identify skill diversity, recruitment, market creation and intrusion as the key reasons for companies to do open-sourcing. They also discuss some problems that could be encountered in various areas when a company shifts towards open-sourcing.

Fagerholm et al. (2014) explains that recruitment, low cost innovation and increased productivity are the major factors that motivates a company to open up. They also analyze data from existing open-sourced projects and conclude that incorporating a mentor during open-sourcing (an experienced developer from within the company) can boost productivity.

Santos et al. (2012) describes that innovation of products and product lines are the main motivators for open-sourcing. They propose that attractiveness of the open sourced project is of major concern for innovation. Based on a theoretical model, they analyze various factors like type of license, type of user, application domain, stage of development which can influence the attractiveness, effectiveness, activeness, likelihood of task completion and time to complete the tasks of the project. In respect to the results of their analysis they propose what kind of projects should be outsourced, how to co-ordinate the project, what licensing should be chosen and also how to increase market visibility.

Morgan and Finnegan (2014) tells that firms should rethink their strategy and processes as there is a recent shift in focus from ownership to openness and collaboration with external parties. They tell that open-sourcing is a value creation process which provides businesses with value through access to knowledge and innovation capacity residing in online communities. Based on a field study of some European firms they explore the creation and capture of business value through strategic open-sourcing. Based on their study they explain that some of the characteristics of open-sourcing that facilitate value creation are cost advantage, quality, escaping from vendor lock-in, increased collaboration, customer service and commodification on non-differentiating part of the software. They explain that quality, reliability, security and performance of open-sourcing are perceived by all the companies as particular characteristics of value creation. They also identify that working as part of a community provides the company with access to code and opportunities for recruitment outside of their own development team. They also discuss some factors which can create conflicts like sharing source code to public and lack of proper road maps.

West (2003) explores the indirect benefits of open-sourcing by doing a case study on different OSS done at Apple, IBM, Microsoft and other companies and identifies product maintenance and visibility to be the major factors. Other factors included commoditization of extension components. They also present their studies related to the strategies the companies followed in adopting hybrid business model of whether to open-source the commodity part (non-differentiating) or opening up the technology part which the competitors cannot easily do.

Haruvy et al. (2008) explore the control decisions regarding the factors for open-sourcing. They put forth a mathematical model and do an analytical investigation and numerical analysis with respect to cost, quality and workforce based on demand and compare these factors between closed sourcing and open-sourcing. They identify that the success to open-sourcing decision depends on product demand and the in-house cost of producing the same. They explain that the model proposed by them can help firms to decide on open-sourcing to gain profit and also lists the main motives of open-sourcing to be cost savings in terms of quality, developer network and labor cost savings.

The article of Gentleman et al. (2004) is the only relevant article identified in the survey that comes from a completely different domain and had elicited the importance of open-sourcing. The authors have unanimously voted for open-sourcing of software components in the field of computational biology and bioinformatics. The reason they provide for open-sourcing and the major motivating factor is that, in case of such complicated scientific fields it is better to develop software by incorporating a wide community diverse in domain skills. Software and data resources in an open-source environment can be read by interested investigators, and can be modified and extended to achieve new functionalities. They also claim that Novices can use the open sources as learning materials and is particularly effective when good documentation protocols are established. They explain that open-source approach thus can aid in recruitment and training of future generations of scientists and software developers.

3.2 List of Best Practices

Ref.No/Name	BP01 - Build Open-Sourcing (OS) Clearing House
Category	People
Context	The company is planning to strategically open-source some of its software components
Actor	Open-source Evangelist / Advocate / Coordinator
Problem	Who will decide whether a software component can be open sourced? Who decides whether open-sourcing a particular component will realize a strategic intent and reap benefits? Who will ensure that different aspects of open-sourcing are in their respective form and structure.
Solution	<p>The Open-source advocate of the company should build a Open-Sourcing Clearing House which is a decision making body. This is a cross-functional team and is responsible to provide a mechanism for distributing governance of OSS development and decision making more broadly across the organization. The stakeholders of the group would consist of members from the following:</p> <ol style="list-style-type: none"> 1. Engineering team 2. Delivery Heads 3. Product security team 4. IT Compliance Team 5. Legal Team 6. Tooling team 7. Product Management Team <p>A recommendation would be to include odd number of members to ensure decisions. The major responsibilities of this team would be the following.</p> <ol style="list-style-type: none"> 1. Take decision on whether to open-source a software component. This decision should be based on a well-defined policy as put forward by best practice BP03 “Create & Use Strategic Decision making policy”. 2. Define action points for a selected component in its incubation phase. 3. Review whether an incubated software component is ready for migration to actual open-sourcing environment. 4. Patent, License and Compliance clearing. 5. Meet on a regular basis to approve/reject projects awaiting public release. 6. Manage priority list of projects to be open sourced. 7. Timely inform policy evolution to respective stakeholders.

Ref.No/Name	BP02 – Build Open-Sourcing Central Team
Category	People
Context	The company is planning to strategically open-source some of its software components
Actor	Open-source Evangelist / Advocate / Coordinator
Problem	Who is responsible for incubating the project that is approved by the open-sourcing clearing house? How will the incubated project transformed to a form that is ready for migration to open-sourcing environment? Who will guide all the teams responsible for open-sourcing an approved product? Who will guide all the teams to identify, propose and submit a potential project for getting approval for open-sourcing?
Solution	<p>The Open-source advocate of the company should build a cross functional central team whose purpose is to help the delivery teams in achieving their milestones. This team is different from the clearing house that is built based on best practice BP01 “Build Open-Sourcing Clearing House”. The contrast is that this team will focus on day to day operational efforts related to open-source development. The stakeholders of the team will be members from the following.</p> <ol style="list-style-type: none"> 1. Development team 2. Architecture team 3. IT compliance team

	<p>4. Tooling team 5. Product security team 6. Legal team 7. Operations team</p> <p>The main responsibilities of this team are the following:</p> <ol style="list-style-type: none"> 1. Follow the open-source realization workflow created as a result of BP04 “Create & Use Open-Sourcing Realization Workflow”. 2. Help delivery teams to prepare OSS projects for Clearing House approval. 3. Establish open-sourcing objectives and key results for the company and appropriate mechanisms to measure the results. 4. Guide & Train all relevant stakeholders in open-sourcing development practices and policies. 5. Establish mechanisms to include open-sourcing activities in performance reviews. 6. Help & provide mechanism for stakeholders to track progress and effectiveness of open-sourcing process.
--	--

Ref.No/Name	BP03 – Create & Use Strategic Decision making policy
Category	Policy
Context	The company has decided to realize the benefits of open-sourcing and want to formulate processes and guidelines required to implement open-sourcing in some of its potential software components.
Actor	Create - Open-source Evangelist / Advocate / Coordinator; Use – Open-Sourcing Clearing House
Problem	On what basis does the OS clearing house approves a software component for open-sourcing? What are the guidelines that support the clearing house to review an incubated product and approve it for migration to real open-sourcing environment?
Solution	<p>The open-source advocate of the company in consultation with all stake holders of open-sourcing should create a strategic decision making policy. This policy document will serve as guidelines to the OS clearing house (Ref BP01) 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 should contain answers to the following questions.</p> <ol style="list-style-type: none"> 1. What factors are to be checked to decide whether a software component can be open sourced? 2. What are the strategic intents that are permissible to open-source a software component? 3. What should be the nature of the software component to consider it 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. What are the architecture, design and implementation concerns that the software component be checked against? 7. What are the business factors that should be considered for approving a project for open-sourcing? (ex. Factors related to competitive features, domain knowledge & Unique Selling Points)

Ref.No/Name	BP04 – Create & Use Open-Sourcing Realization Workflow
Category	Process
Context	The company has decided to open-source some of its software components strategically and requires a well defined workflow to realize and monitor it.
Actor	Create - Open-source Evangelist / Advocate / Coordinator; Use – Open-Sourcing Central team
Problem	What are the steps that the OS central team (Ref. BP02) should follow to realize the open-sourcing of an approved software component? How can the realization of open-sourcing a software component be monitor and tracked by various stakeholders?
Solution	<p>The open-source advocate of the company should define a workflow that the OS central team should follow to realize the open-sourcing of an approved software component. The workflow definition is based on the company structure and hierarchy. It is highly recommended that once this realization workflow is defined, it must be automated to enable efficient functioning. Various tools must be deployed in conjunction with this workflow to help stakeholders monitor and track various work items in the workflow. The workflow should define processes for the following functions.</p> <ol style="list-style-type: none"> 1. Temporary incubation of projects approved by OS clearing house. 2. Realize the process of modification of the software project in line with actions points declared by the clearing house. 3. Realize the migration of the software project from incubator to actual open-sourcing environment. 5. Train stakeholders on policy evolution. 6. Change management. 7. Intellectual components abstraction and approval. <p>This workflow should be integrated with other workflows like tool development, license approval, compliance approval and OS core process.</p>

Ref.No/Name	BP05 – Control Strategically
Category	Policy
Context	The company has open sourced some of its software components and the project management team needs to make decisions on feature planning, prioritization and corresponding milestones.
Actor	Project Management Team
Problem	Since the company had open sourced its software component, there is a high level of influence from contributors external to the company. Hence feature planning, prioritization and milestone planning becomes complicated for the project management.
Solution	<p>The project management should make strategic decisions instead of conventional decision making. The project management should keep the following in mind when making decisions</p> <ol style="list-style-type: none"> 1. The company should reserve the right to make changes to the component though it is open-sourced. 2. Though it is open-sourced, cost-benefit analysis should be done for prioritization of features. A well-defined process should be in place to handle decision making during trade-offs in feature prioritization with respect to community vs company. 3. The dependence of proprietary software on the open sourced component should be considered when making milestone planning of both the internal and open-sourced software. 4. From the perspective of the company, the decision making process should always be strategically oriented to the benefit of the company. If the resultant decision has adverse effects on the external community or the open sourced software, then the decision needs to be revisited.

Ref.No/Name	BP06 – Deploy a Central Coordinator
Category	People
Context	The company has open sourced one or many software components and the open sourced software has contributors from both internal and external to the company.
Actor	Project Management Team
Problem	Who will co-ordinate various aspects in a open-sourcing environment where both contributors internal and external to the company are involved?
Solution	<ol style="list-style-type: none"> 1. The project management team should deploy a central coordinator who will manage project control, communication and other aspects between the internal and external contributors. 2. This coordinator will enable community building. 3. This role would be advantageous to the company to influence strategic decisions and team building. 4. The coordinator should set up a proper communication channel and should organize regular meetups with internal and external team members to discuss on policy changes, future planning and any other pit falls that were identified in open-sourcing environment, tooling, processes or documentation.

Ref.No/Name	BP07 – Create Flexible and Extensible Software
Category	Artifact
Context	An internal software component is approved to be open sourced and need to be modified to be capable of open-sourcing
Actor	Software Development Team
Problem	What are the design modifications that need to be done to the approved software component to make it maintainable even after open-sourcing it?
Solution	<ol style="list-style-type: none"> 1. The interface to the software component should be clearly defined. 2. The component should be loosely coupled with other dependent internal software components. 3. Extension points should be properly defined. 4. Data contracts for inputs and outputs to the software component must be well-defined. 5. Dependencies of the component on other internal components should be completely eliminated.

Ref.No/Name	BP08 – Abstract differentiating features & Protect Intellectual Property
Category	Artifact
Context	An internal software component is approved to be open sourced and need to be modified to be capable of open-sourcing without affecting business interests. Another context is that the software component is already open sourced and is being modified by the internal software development team.
Actor	Software Development Team
Problem	What are the functional modifications that need to be done to the approved software component to avoid conflicts with business interests of the company? What are the considerations that need to be taken care when modifying a software component that was open sourced strategically?
Solution	<ol style="list-style-type: none"> 1. The differentiating features of the software component which contribute to specific business value of the company should be abstracted. This includes business value obtained either because of competition or by a specific unique selling point (USP) 2. Any features that is directly or indirectly related to an intellectual property owned by the

	<p>company should be abstracted from the software component.</p> <p>3. Any feature pertaining to the core domain knowledge of the company should be avoided from open-sourcing and hence must be abstracted.</p> <p>4. If the strategy behind open-sourcing is to create open standards, then the component that is planned for open-sourcing should be modified in such a way that it contains only features that are required for the standard and the remaining features should be abstracted.</p>
--	---

Ref.No/Name	BP09 – Plan Small & Rapid Iterations
Category	Process
Context	A software component of the company is strategically open sourced and modifications to the both the internal and external software components need to be planned
Actor	Project Management Team
Problem	How should the project management team plan modifications to the internal software component and the open sourced software component?
Solution	<p>1. The schedule for feature development and bug fixing of the internal components that are dependent on the open sourced component should be planned in accordance to the plans of the external component and vice versa.</p> <p>2. Rapid iterations are recommended because any breaking changes that happen in the external component can be fixed and adapted at a very fast pace. In contrast, long iterations will make bug fixes in open sourced component to be reflected in the internal components only after a long time.</p> <p>3. Changes to the software must be planned in small work packages for each iteration so that the possibility of failure is reduced and mitigation of risk is easier.</p>

Ref.No/Name	BP10 – Use a Centralized Dashboard
Category	Tooling
Context	The company has open sourced one or many software components and the internal team and external contributors are contributing to the software component in a full fledged manner.
Actor	Software Development Team
Problem	How are tools, processes, artifacts and information managed when the software component is open-sourced?
Solution	<p>1. It is a best practice to have a centralized dashboard which will be an one stop solution for all members involved in the open-sourcing environment.</p> <p>2. This dashboard should provide links to tools and documents that facilitate and ease internal team to communicate and work with the open-source environment.</p> <p>3. The dashboard should include metrics related to the number of open issues, ratio of internal to external contributions for a given period of time, iteration schedule etc.</p> <p>4. The dashboard should give information on health of the project at any given point in time.</p>

Ref.No/Name	BP11 – Respect License
Category	Policy
Context	The company has approved a software component to be open sourced and it is in the migration phase of moving the project from incubator to actual open-sourcing environment
Actor	Software Development Team
Problem	What kind of legal and license criteria that need to be taken care when moving a software component to a open-source environment?
Solution	1. Based on the nature of the project that is open sourced, an appropriate Contributors License Agreement (CLA) must be defined by the Project Management Team in association with OS

	<p>clearing house (Ref BP01).</p> <p>2. A specific open-source license (like MIT or BSD) must be defined to make it clear to the contributors regarding to compliance and licensing issues.</p> <p>3. The license criteria should be well defined in advance and should be included in the approval and review process by the OS clearing house (Ref BP01) and should be held as a guideline by the OS central team (Ref BP02) when following the realization work flow (Ref BP04).</p> <p>4. Contributors should be aware of the boundaries of the licenses and ensure that their contributions are compliant.</p>
--	---

3.3 Additional Research Results

3.3.1 When NOT to Open Source?

One of the goals of the research was to identify various intents behind which a software company can strategically plan to open-source their software components. The quest for those answers also yielded interesting results of what should NOT be open sourced. The results are based on Data Source 1, 2, 5. The following are those that should not be open sourced

1. Components which are tightly coupled to internal systems
2. Projects that require higher security
3. Projects that the company does not plan to maintain.
4. Components that carry domain knowledge
5. Components that contain customer data
6. Unique selling points (USPs)
7. Projects that risk competitive advantage
8. Components that carry intellectual property

Projects which had been stripped off of the above content should be carefully checked before making it public since certain information can be retrieved from artifact's history if migration was not executed properly.

3.3.2 Cause-Effect of Strategies

From the research a set of intents were derived that cater to the reason behind a company's move towards strategic open-sourcing. From the case studies, though the cause does not seem to have strategic benefit, the effect caters to core business values. For example, from the case studies it was evident that at least two of the companies open sourced some of their components for the sake of 'giving back'. But data from the same cases reveal that this intent is to obtain a benefit of Branding or Product visibility of the company. Here the cause being 'giving back' but the real effect is 'Branding and Product Visibility'

Likewise, it is evident from the research that companies open-source their components for creating inter-operable standards that can benefit the external community. But this intent has an inherent benefit of increasing business value of other products of their company which rely on these standards. Here the cause being 'Open Standards' the real effect is to 'Expand Business'

3.4 Acknowledgments

I would like to acknowledge Prof. Dr. Dirk Riehle and Nikolay Harutyunyan who supported me throughout this entire research by helping and guiding me to find industry partners for the case study, interview organization, conduction and providing access to tools for archiving and analyzing data. I would also like to thank them for continuously guiding, reviewing and providing feedback on my research work.

I would like to thank all the industry partners for providing valuable information through the interviews and for sharing internal documents which were very useful for the research work.

Appendix A

Case study protocol

1. Background

- a) Identify previous research on the topic. A systematic literature survey was conducted based on approach described by Webster and Watson (2002) to obtain data relevant to formulation of research question and design of case study interviews.
- b) Define the main research question being addressed by this study. Why and How do companies strategically open-source software components?
- c) Identify any additional research questions that will be addressed. What are the Industry best practices to open-source software components strategically?

2. Design

- a) Identify whether single case or multiple case and embedded or holistic designs will be used, and show the logical links between these and the research questions. Embedded multiple case study is used since the case study is established across four companies under the context of strategic open-sourcing. 'Why' and 'How' are the major units of analysis and multiple best practices in varying sub-contexts are the sub-units of analysis.
- b) Describe the object of study
- c) Identify any propositions or sub questions derived from each research question and the measures to be used to investigate the propositions. Three different domains were identified during literature survey and also remained valid during the course of the case study namely: Open-source Evangelism, Project Management and Software development. Under these domains, different contexts in the form of People, Process, Policy, Tools and Artifacts were identified. The qualitative data analysis was carried out in these domains considered along with these contexts.

3. Selection

- a) Criteria for case selection. Case selection was done based on a theoretical sampling that was done taking the following dimensions into consideration. Type of Software company, type of customer, market position, size of company, maturity of company and the product under consideration.

4. Procedures and roles

- a) Procedures governing field procedures. Procedures included semi-structured exploratory online interviews with the interviewees with Nikolay Harutyunyan also taking part in the interviews. The list of interview questions is presented in Appendix B.

5. Data Collection

- a) Identify the data to be collected The main source of data are data derived from literature survey, data from semi-structured interviews with relevant industry partners and documents and information shared by those interviewees.
- b) Define a data collection plan. Data collection is achieved in the form of semi-structured interviews, interview questionnaire and by summarizing information from documents and links obtained through interviews.
- c) Define how the data will be stored. All the data is stored digitally. The data from interviews are stored as video & audio conference files or audio files. Formal communication

are stored in email formats and responses to questionnaire in document format. The transcribed data is stored as text files.

6. Analysis

- a) Identify the criteria for interpreting case study findings Interpretation criteria involves domains and contexts derived from literature and cases and applying subsequent data analysis based on these domains and contexts. The process was carried out in iteration deploying pattern matching, explanation building and triangulation.
- b) Identify which data elements are used to address which research question/subquestion/proposition and how the data elements will be combined to answer the question. Each research result is mapped with their data source by providing references and the same is available in Research Results section.
- c) The analysis should take place as the case study task progresses. Unstructured data analysis was carried out after each data collection and further data collection was fine tuned based on that.

7. Plan Validity

- a) General: check plan against Host & Runeson's (2007) checklist items for the design and the data collection plan. Plan was checked with the checklist items and the filled up checklist is available in Appendix C
- b) Construct validity—show that the correct operational measures are planned for the concepts being studied. Tactics for ensuring this include using multiple sources of evidence, establishing chains of evidence, expert reviews of draft protocols and reports. Chain of evidence is assured by coding for qualitative data analysis.
- c) Internal validity—show a causal relationship between outcomes and intervention/treatment (for explanatory or causal studies only). The outcomes were derived from data obtained from coding system of qualitative data analysis and thus ensures internal validity.
- d) External validity—identify the domain to which study finding can be generalized. Tactics include using theory for single-case studies and using multiple-case studies to investigate outcomes in different contexts. Multiple case studies executed on partners from specific domains in relevant companies ensured external data validity. It was possible to identify the answers to the research questions and to formulate a set of industry best practices in strategic out sourcing.

8. Study Limitations

- a) Specify residual validity issues including potential conflicts of interest The major limitations were that some of the interviews needed to be carried out in a non-exploratory manner through questionnaire due to time schedule of industry partners.

9. Reporting

- a) Identify target audience, relationship to larger studies (Yin 2013) The target audience of this research are the companies which want to achieve some strategic benefits by open-sourcing some of its non-differentiating software components. It also targets companies which are already open-sourcing for strategic benefits and need to incorporate industry best practices to open-source efficiently.

10. Schedule

- a) Give time estimates for all of the major steps: planning, data collection, data analysis, reporting. Note data collection and data analysis are not expected to be sequential stages.

Six months was planned for the complete research. One month was dedicated to literature survey and validating the research question. Three months were dedicated to case design, data collection. One month was dedicated for data analysis and one month for reporting.

11. Appendices

- a) Validation: report results of checking plan against Host and Runeson's (2007) checklist items Filled up checklist is available in Appendix C
- b) Divergences: update while conducting the study by noting any divergences from the above steps. There were no major divergences from the protocol.

Appendix B

Interview Questions

Context of Interviewee

1. Could you present yourself, your project, and your role in your company?
2. How is your work connected to open-source and (strategic) open-sourcing?

Why do companies open-source

3. What is the motivation behind your company's initiative to open-source some of its software components under your name (e.g. github.com/company)?
 - a) Strategic motivation for open-source leadership and contribution?
 - b) Mid-range goals for open-source contributions?
 - c) Short-term goals for open-source contributions?
4. When you review expected and realized benefits, do you see a difference?
5. What do you think are the pros and cons of open-sourcing some software components (of your products, development tools etc.)?
6. Who defines company's open-source strategy? Who are the main stakeholders?
7. How does open-sourcing influence your products / product lines, development process?

How do companies open-source

8. Which types of components do you open-source?
 - a) Parts of products
 - b) Development tools
 - c) others
9. How do you decide which components to open-source?
10. Did you open-source existing internal software component? Example?
11. Which procedures did you follow for migrating an existing internal component to open-sourcing environment?
12. Did you open-source a software component from scratch? Example?
13. Which procedures did you follow for open-sourcing a software component from scratch?
14. How are your company's open-sourcing practices different from conventional open-source projects?
15. Do you have best practices you follow when open-sourcing from your company's name? How did they change over time?
 - a) Before open-sourcing
 - b) During / for open-sourcing

- c) After open-sourcing
- 16. Which tools do you use for open-sourcing? Why? How did this change over time?
- 17. What is the level of influence your company has on your open sourced software components?

Other Questions

These set of questions were formulated for the first interview and were later omitted or made as optional based on reviewing the initial interview outcomes.

1. Under what criteria, components in your company's product were chosen to do open-sourcing?
2. What is your Opinion on open-sourcing? Do you prefer to do open-source from the scratch? Or Do you prefer to open up a partially developed component?
3. How does your company market the open sourced component to attract talents to contribute?
4. Does your company really have a competitive edge or cost edge or both by doing open-sourcing?
5. Was there any changes done within the internal team structure or process to adhere with the open-source environment?
6. Do you have a common interface between the internal team and the open sourced team?
7. How is Conflict of interest managed in intra and inter-team?
8. How is the Compliance Managed ?
9. How is Change management propagated? (propagation of changes in requirement based on your company's internal requirements or changes to requirements in dependent components developed internally?)
10. Is there a practice that is followed in your company to check the motivation levels of internal and external contributors?
11. Which platforms do you use for open-sourcing?
12. Are there any other essential components and tools for this launch of open-sourcing ?
13. Who are involved in high level and low level software design?
14. How is configuration management (version control) done in your team?
15. How is quality of code contributed by external contributors ensured?
16. Does your internal team take work based on priority from backlog or is it from individual interest similar to external contributors?
17. Talent of internal contributors vs external contributors. Is external contributors talent visible from inside?
18. How are interfaces between internal components and open sourced components managed?
19. What do you think are the major strategic driving factors that made your company to open-source its products/components?

Appendix C

Researcher's Checklist

Case Study Design

1. What is the object of study? Intent and Process of Strategic open-sourcing in four companies.
2. Is a clear purpose/objective/research question/ hypothesis/proposition defined upfront? Yes, The purpose of the research is to identify the answers for the following research questions: Why & How do companies strategically open-source some of its software components?, and What are the best practices in open-sourcing such components strategically?
3. Is the theoretical basis – relation to existing literature and other cases – defined? Yes, The Related Work and the Literature Survey sections define the theoretical basis for the research. Empirical results are derived from the literature and analyzed in conjunction with the findings from the case study.
4. Are the authors' intentions with the research made clear? The research intention are defined in the Research Question section of this paper.
5. Is the case adequately defined (size, domain, process...)? The case contexts are defined in the Research Approach and Used Data Sources section of this paper.
6. Is a cause-effect relation under study? If yes, is the cause distinguished from other factors? Yes, The best practices derived from the collected and analyzed data and listed in the Research Results section exhibit a cause-effect relationship.
7. Will data be collected from multiple sources? Using multiple methods? Data was collected from semi-structured exploratory interviews, interview questionnaire, documents and internet links shared by the interviewees and project dashboards. Case study methodology for research and Qualitative data analysis for data analysis are the various methods used.
8. Is there a rationale behind the selection of roles, artifacts, viewpoints, etc.? Selection was done based on theoretical sampling and other factors described in Research Approach and Used Data Sources section.
9. Are the case study settings relevant to validly address for the research question? Yes, each case was chosen based on settings which were relevant to the research.
10. Is the integrity of individuals/organization taken into account? Each project corresponding to a strategic open-sourcing case is studied as an integral organization with its own processes and practices.

Preparation for Data Collection

11. Is a protocol for data collection and analysis derived (what, why, how)? Yes. The case study protocol used for data collection and analysis is listed in Appendix A
12. Are multiple data sources and collection methods planned? Yes, multiple data sources in the form of four different companies were chosen for case study. Different collection methods included interviews, information from company's internal documents, wikis and other documents and links shared by interviewees.
13. For quantitative data, are the measurements well defined? The research question is targeted to obtain qualitative results. Hence quantitative methods were not defined for this research.

14. Are the planned methods and measurements sufficient to fulfill the objective of the study? Yes, research through case study methodology, data collection through case interviews and data analysis through qualitative data analysis are sufficient to obtain best practices for a company to initiate and do open-sourcing strategically.

15. Is the study design approved by review board, and has informed consent obtained from individuals and organizations? The study is approved by the appropriate university department and formal informed consent is obtained from relevant individuals and organization.

Collecting Evidence

16. Are data collected according to the protocol? Yes.

17. Is the observed phenomenon correctly implemented (e.g. to what extent is a design method under study actually used)?

18. Are data recorded to enable further analysis? Yes. All data are recorded and stored to enable further analysis.

19. Are sensitive results identified (for individuals, organization, project)? Yes. Sensitive data were identified and appropriately masked to maintain data privacy.

20. Are the data collection procedures well traceable? Yes. Results and data are traceable through coded data generated from qualitative data analysis.

21. Do the collected data provide ability to address the research question? - Yes. The results were able to answer each and every part of the research question.

Analysis of Collected Data

22. Is the analysis of methodology defined, including roles and review procedures? Yes, Qualitative data analysis methodology was used and was done by using QDAcity tool.

23. Is a chain of evidence shown with traceable inferences from data to research questions and existing theory? Research results section contain references that link result data with research data and data from existing theory and link them with research questions.

24. Are alternative perspectives and explanations used in analysis? Yes.

25. Is a cause-effect relation under study? If yes, Is the cause distinguished from other factors? Yes, The best practices derived from the collected and analyzed data and listed in the Research Results section exhibit a cause-effect relationship.

26. Are there clear conclusions from the analysis, including recommendations for practice/further research? Yes, clear conclusions are provided in the Research results sections and open areas of research identified during the case study is provided in the conclusion section.

27. Are threats to validity addresses in a systematic way? The threats are reduced by using proven methods like case study methodology, qualitative data analysis along with using a case study protocol and a research check list.

Reporting

28. Are the case and its context adequately reported? Yes

29. Are the research questions and corresponding answers reported? Yes

30. Are related theory, hypotheses and propositions clearly reported? Yes

31. Are the data collection procedures presented, with relevant motivation? Yes
32. Are sufficient raw data presented? Yes
33. Are the analysis procedures clearly reported? Yes
34. Are threats to validity analysis reported? There were no threats identified.-
35. Are ethical issues reported openly (personal intentions, integrity issues)? Yes
36. Does the report contain conclusions, implications for practice and future research? Yes
37. Does the report give a realistic and credible impression? Yes
38. Is the report suitable for its audience, easy to read and well structured? Yes

Appendix D

Code System

Code System	Code	Description	No. of Segments
Code System			221
Strategy			36
	Technical	This code is relevant to strategies of open-sourcing if motivation behind it is related to innovation, skill & domain diversity, code quality, software maintenance, open standards creation, rapid value addition or productivity.	18
	Business	This code is relevant to strategies of open-sourcing if motivation behind it is related to recruitment, talent acquisition, cost savings, ROI, Demand creation, customer value, product visibility, market creation/intrusion, competition	18
Evangelist Best Practices			118
	Communication	This code is relevant to best practices on communication centric to open source evangelist/advocate/co-ordinator	2
	Decision making	This code is relevant to best practices in making decisions to open source some of the company's software components. It also includes decisions relevant to achieving the goal of open sourcing and decision making influenced by an open source evangelist/advocate/co-ordinator	15
	Documentation	This code is relevant to all documentations relevant to open source evangelism	1
	Licensing	This code is relevant to decision making regarding licensing of open sourced software components	10
	Processes	This code is relevant to all processes that are carried out to open source a software component under a company's name. It also includes processes relevant to modifying existing processes of the company to handle the new environment	30
	Strategy-benefit mapping	This code is relevant to any information from the view of evangelist regarding to cost-benefit value as a result of open sourcing. IT also includes additional side effects incurred due to open sourcing. The side effects could be either a pro or a con to the company	28
	Tools	This code refers to all tools that were used for evangelism or advocacy for open sourcing under a company's name	8
	Training and motivation	This code refers to any training or motivational processes that were initiated/carried out by evangelists or advocates for open sourcing under a company's name	6
	Team	This code refers to hierarchy, structure and responsibilities regarding to open source coordination	18
Project Development Best Practices			49
	External Communication	This code is relevant to all communication between the developers/testers of the open source project inside the	2

Code System	Code	Description	No. of Segments
		company and the developers/testers of the open source project external to the company	
	Documentation	This code is relevant to all technical documentation related to the open source project and also projects that have dependency to the open sourced project	2
	Processes	This code is relevant to all processes in relation to the technical team which is handling the open sourced software component.	9
	SDLC management	This code is relevant to any technical aspects regarding to the software development life cycle of the software component like, design, implementation, testing, code quality, configuration management, code integration, component dependencies, merge processes, code documentation, software hooks, development methodologies, prioritization, feature or increment planning	7
	Tools	This code is relevant to all tools being used by the development team while developing the software component in open source environment.	6
	Team	This code is relevant to any information regarding to management team size, value/skill, hierarchy or structure and responsibilities	12
	Software Component	This code is relevant to any information regarding the management of the open sourced software component and dependencies	11
Project Management Best Practices			18
	External Communication	This code is relevant to all non-technical communications between the people of the open source project internal to the company and with those external to the company	1
	Decision making	This code is relevant to any decision making done to the open sourced software project that are not related to the open source evangelist	5
	Processes	This code is relevant to any managerial processes relevant to the open source project	8
	Tools	This code is relevant to all tools used by project management of the open source project	1
	Training and motivation	This code is relevant to all training and motivational campaigns initiated or organized by/for project management for the open source project team	1
	Team	This code is relevant to any information regarding to contributors team size, value/skill, hierarchy or structure or responsibilities	2

Table 11. Code System

References

- Ågerfalk, P., & Fitzgerald, B. (2008). *Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy*. *MIS Quarterly*, 32(2), 385.
- Asundi, J., Carare, O., & Dogan, K. (2012). *Competitive implications of software open sourcing*. *Decision Support Systems*, 54(1), 153-163.
- 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.
- Fagerholm, F., Sanchez Guinea, A., Borenstein, J., & Munch, J. (2014). *Onboarding in Open Source Projects*. *IEEE Software*, 31(6), 54-61.
- Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., & Dudoit, S. et al. (2004). *Bioconductor: open software development for computational biology and bioinformatics*. *Genome Biology*, 5(R80).
- 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.
- Hecker, F. (1999). *Setting up shop: The business of open-source software*. *IEEE Software*, 16(1), 45-51.
- Hippel, E., & Krogh, G. (2003). *Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science*. *Organization Science*, 14(2), 209-223.
- Höst, M., & Runeson, P. (2007). *Checklists for Software Engineering Case Study Research*. *ESEM*, 479-481.
- Laat, P. (2004). *Evolution of Open Source Networks in Industry*. *The Information Society*, 20(4), 291-299.
- Lerner, J., & Tirole, J. (2002). *Some Simple Economics of Open Source*. *The Journal Of Industrial Economics*, 50(2), 197-234.
- Lerner, J., & Tirole, J. (2005). *The Economics of Technology Sharing: Open Source and Beyond*. *Journal Of Economic Perspectives*, 19(2), 99-120.
- Lindman, J., Paajanen, A., & Rossi, M. (2010). *Choosing an Open Source Software License in Commercial Context: A Managerial perspective*. In *Proceedings of 36th EUROMICRO Conference on Software Engineering and Advanced Applications* (pp. 237-244). Lille, France: IEEE.
- 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.
- 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.
- 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.

- Shaikh, M., & Cornford, T. (2009). *Innovating with Open Sourcing: Governance Concerns for Managers*. In *Proceedings of AMCIS 2009* (p. 308). San Francisco, California.
- Walli, S., Gynn, D. and Rotz, B. (2005). *The Growth of Open Source Software in Organizations*. Optaros.
- Webster, J., & Watson, R. T. (2002). *Analyzing the Past to Prepare for the Future: Writing a Literature Review*. *MIS Quarterly*, 26(2), 13–23.
- West, J. (2003). *How open is open enough? Melding proprietary and open source platform strategies*. *Research Policy*, 32(7), 1259-1285.
- West, J., & Gallagher, S. (2006). *Challenges of open innovation: the paradox of firm investment in open-source software*. *R And D Management*, 36(3), 319-331.
- Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.