How Open Source Foundations Handle Conflicting Interests in Company-Started Projects

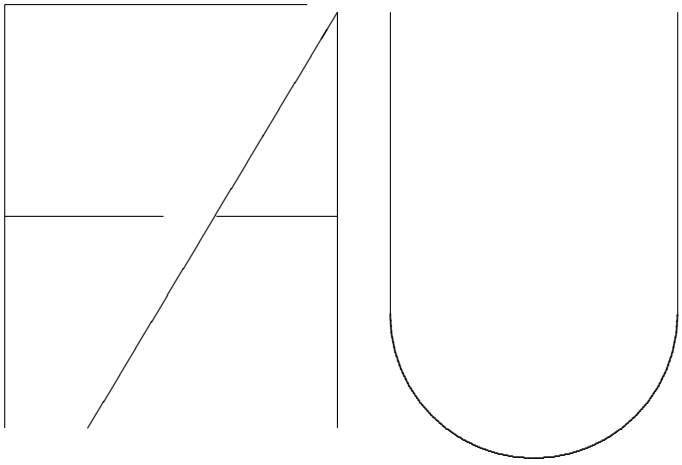
Masterarbeit

Florian W. Weikert

Professur für
Open Source Software

Department Informatik
Technische Fakultät

Friedrich Alexander-
Universität
Erlangen-Nürnberg



How Open Source Foundations Handle Conflicting Interests in Company-Started Projects

Masterarbeit im Fach Informatik

vorgelegt von

Florian W. Weikert

geb. 02.01.1988 in Fürth

angefertigt am

Department Informatik
Professur für Open Source Software
Friedrich-Alexander-Universität Erlangen-Nürnberg

Betreuer: Prof. Dr. Dirk Riehle, M.B.A.

Beginn der Arbeit: 17.02.2014
Abgabe der Arbeit: 17.08.2014

Erklärung zur Selbständigkeit

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

Erlangen, den 17.08.2014

(Florian W. Weikert)

License

Erlangen, den 17.08.2014

(Florian W. Weikert)

Abstract

As Open Source Software has become nearly ubiquitous, more and more companies have started to open up the code of their products. Even more radical, some of them donated their projects to independent nonprofit foundations, thus giving up control while effectively inviting their competitors to participate. Consequently, these projects are subject to numerous commercial interests. This phenomenon raises some questions: How does such a foundation handle conflicting interests of its members, and how does it protect its own interests while doing so? Moreover, how can the donor of the project make sure that his interests are going to be considered after relinquishing control? The present thesis addresses these questions through an exploratory multiple-case study using grounded theory. By examining four real-world cases, a theory about conflicting interests is developed and practical implications thereof are discussed.

Contents

List of Figures

List of Tables

List of Abbreviations

ACS Apache CloudStack

API Application Programming Interface

ASF Apache Software Foundation

AWS Amazon Web Services

CAB Community Advisory Board

CF Cloud Foundry

CSR Case Study Research

EC Eclipse

FLOSS Free/Libre and Open Source Software

GT Grounded Theory

IaaS Infrastructure-as-a-Service

IRS Internal Revenue Service

OSt OpenStack

OSI Open Source Initiative

OSS Open Source Software

PaaS Platform-as-a-Service

PMC Project Management Committee

1. Introduction

This chapter describes the original goals of the thesis and how they have changed over time. A more thorough overview of the research process can be found in Section 2.3 where the rationale behind our decisions is explained in detail.

1.1. Original Thesis Goals

Initially, the present thesis was titled "A Theory of Open Source Developer Foundations" and aimed at understanding why companies donated their own software to nonprofit open source foundations, thus giving up their intellectual property rights. An exploratory multiple-case study based on interviews should be employed to examine two cases, the Eclipse Foundation and the GENIVI Alliance. Moreover, the research results should enable us to make recommendations for two companies facing a similar decision.

However, it became evident that existing literature had already covered this topic exhaustively. Consequently, we decided to change the scope of the thesis.

1.2. Modified Thesis Goals

Since we were still interested in this particular phenomenon, we began looking for appropriate cases and a novel research question in this area. After examining existing foundations and reading relevant literature on this topic, three open source projects emerged as potential cases for the research project: Apache CloudStack, OpenStack and Cloud Foundry. This led to the second version of the thesis description, named "A Comparative Study of Cloud Computing Open Source Foundations". Since we were still exploring potential research questions at that time, the thesis description contained the temporary goal of finding out why there were three separate foundations and how they differed from each other.

1.3. Final Thesis Description

The thesis description had to change once more when we finally discovered the final research questions (see Section 2.1.1). Additionally, we had to add a fourth case (Eclipse Foundation) since we could not find an interview partner at Cloud Foundry.

2. Research Chapter

2.1. Introduction

Open Source Software (OSS) is a phenomenon that "continues to eat the software world" (Black Duck Software, 2014, p. 6), as indicated by the widespread use of popular open source operating systems (Linux, Android), web browsers (Chromium, Firefox) and other applications (Apache HTTP server, Hadoop, MySQL)[1]. Additionally, it has managed to be present on all kinds of platforms, starting on desktop computers (Linux), moving to mobile devices (Android) and to cars (GENIVI alliance)[2]. Open source is ubiquitous, as five out of six developers use or deploy open source software (Rooney, 2012) and the majority of companies is expected to engage with it (Black Duck Software, 2014, p. 25). Moreover, it seems to be "picking up momentum" (Finger, 2014) as the amount of open source code is growing exponentially (Deshpande & Riehle, 2008) and consumer awareness increases (Black Duck Software, 2014, p. 54).

Apparently, companies have recognized this trend as their participation in open source projects is increasing (Robles et al., 2010). For example, Google, Facebook, Apple and Microsoft all rely on open source software[3], with Facebook having nearly 10 million lines of open source code (Pearce, 2014). Moreover, companies have succeeded in building viable business models around open source (Fitzgerald, 2006; Hecker, 1999; Krishnamurthy, 2003; Riehle, 2007, 2009). The increased corporate interest has changed the landscape, resulting in new forms such as "single-vendor open source" (Riehle, 2012). However, it has also forced some of the existing community-managed open source projects to create an independent nonprofit foundation[4] with formal rules and defined structures (Hunter & Walli, 2013). Letellier (2008) describes this evolution as moving from the first generation of open source (loosely coupled individuals) to the second generation (foundations with individual members). More recently, a third generation has emerged: Communities where "companies are first-class citizens" (Gonzalez-Barahona et al., 2013, p. 173).

Some companies take their proprietary software and donate it to such nonprofit foundations – thereby giving up control and their intellectual property rights (Skerrett, 2011) – in order to create open standards and to reduce development costs (Riehle, 2010). Consequently, this approach is more radical than single-vendor open source where the creator remains in control (Riehle, 2012). As a result, the donors have the same privileges as other members of the foundation, who can be individuals or other companies, including their competitors – each of them with its own interests (Gonzalez-Barahona & Robles, 2013).

2.1.1. Research Questions and Contributions

The cases of Apache CloudStack (ACS), Cloud Foundry (CF), Eclipse (EC) and OpenStack (OSt) are recent examples of this trend, thus underlining its economic significance. However, this phenomenon also raises several questions:

- *RQ1*: How do foundations handle conflicting interests of their members?
- *RQ2*: How can foundations ensure that their own interests are not impaired?
- *RQ3*: How do donors protect their interests after giving up control?

This thesis addresses these questions through an exploratory multiple-case study of the aforementioned open source projects (Table 2.1) that were created by companies but donated to nonprofit foundations (Table 2.2)[5].

Table 2.1.: Overview of cases. See Section B.1 for more details.

Case	Project	Donor[6]	Creation date	Donation date
ACS	Apache CloudStack	Citrix Systems, Inc.[7]	2008	2012[8]
CF	Cloud Foundry	Pivotal Software, Inc.	2011	2014[9]
EC	Eclipse	IBM Corporation	2001	2004
OSt	OpenStack	Rackspace US, Inc.[10]	2010	2012

2.1.2. Structure of the Thesis

The structure of this thesis is as follows: Section 2.2 shows existing research on this topic and explains the context of this study. Section 2.3 describes the research process and the data sources that were used. The resulting theory is presented in Sections 2.4 and 2.5, while Section 2.6 discusses potential threads to validity and highlights opportunities for future work. Section 2.7 concludes the research part with a short summary.

Table 2.2.: Overview of foundations. See Section B.1 for more details.

Case	Foundation	Creation date	Legal status[11]	Corporate members
ACS	Apache Software Foundation	1999	501(c)(3)	No
CF	Cloud Foundry Foundation	2014[12]	-[13]	Yes
EC	Eclipse Foundation	2004	501(c)(6)	Yes
OSt	OpenStack Foundation	2012	501(c)(6)	Yes

2.2. Related Work

This section provides an overview of related literature[14]. Table 2.3 shows how the cases in this thesis fit into the existing work of other researchers.

2.2.1. Open Source Foundations

Riehle and Berschneider (2012) cover both the Apache Software Foundation (ASF) and the Eclipse Foundation as part of their classification scheme of open source developer foundations. They highlight differences in legal status, mission, philosophy and governance structures. Prattico (2012) also recognizes differences when examining the distribution of powers. While the executive director holds the most power in the Eclipse Foundation, the power in the ASF lies with the board of directors.

Several researchers have covered the reasons for creating open source foundations. For example, Xie (2008) claims that foundations are incorporated in order to handle donations and to exploit tax exemptions.

However, Hunter and Walli (2013) argue that open source communities create foundations to deal with corporate interests. This is supported by O'Mahony (2003, 2005) who sees foundations as a protection from corporate appropriation. However, she points out that such formal constructs may conflict with the libertarian values of open source culture and that the affiliations of individuals have to be recognized. This challenge is also addressed by O'Mahony and Bechky (2008, p. 422) who see foundations as an example of "boundary organizations", which enable collaboration between communities and companies.

Similar to O'Mahony (2003), Riehle (2010) describes foundations as legal stewards that can manage and protect projects, intellectual property rights and communities. Moreover, they can carry out tasks such as marketing and day-to-day operations. He also highlights the economic significance of foundations: Companies can participate in the shared development of a standardized platform, thus reducing costs and being able to compete more efficiently. Additionally, the market for complementary products grows as the platform gains traction. Finally, both reputation and visibility of participating companies increase as they create a common good.

Growing such a shared platform, however, requires a diverse community. Skerrett (2011) identifies five best practices, based on the Eclipse Foundation: The use of a permissive license, not requiring copyright assignments, an open development process, a clear trademark policy and a vendor-neutral governance model. However, companies also have to give up control in order to allow for a level playing field where no dominant player can emerge (Skerrett, 2009).

Table 2.3.: Classification of projects. See Section 3.1.1 for more details.

Case	Capiluppi et al.	Letellier	O'Mahony West	Riehle Berschneider	Watson et al.	Wasserman	West Gallagher
ACS	Sponsored	2nd Gen.[15]	Synthetic, autonomous	Foundation, public benefit	Sponsored 1st Gen.	Foundation-based	Spinout
CF	Sponsored	3rd Gen.	Synthetic, autonomous	-[16]	Sponsored 1st Gen.	Foundation-based	Spinout
EC	Sponsored	3rd Gen.	Synthetic, autonomous	Consortium, member benefit	Sponsored 1st Gen.	Foundation-based	Spinout
OSt	Sponsored	3rd Gen.	Synthetic, autonomous	Consortium, member benefit	Sponsored 1st Gen.	Foundation-based	Spinout

2.2.2. Transition From Company-Founded to Community-Managed Projects

West and O'Mahony (2008, p. 4) examine the differences between sponsor-controlled ("sponsored") and community-controlled ("autonomous") projects. Autonomous projects are managed by a community and the governance is independent of companies and employment status. Sponsors face a trade-off between having control and attracting external contributors. This is similar to what West (2003, p. 2) names "essential tension". Since projects may evolve from one state into the other, O'Mahony and West (2005, p. 32) also introduce the categories "synthetic" (sponsor-created, i.e. started as sponsored project) and "organic" (community-created, i.e. started as autonomous project).

If such a project receives its initial resources and code from a commercial entity, it can be classified as a "spinout" (West & O'Mahony, 2005, p. 1). Consequently, spinouts build on an established code base and are usually supported by their creators. They aim at creating de facto standards, enabling the sales of complementary products, delegating support to the community, improving company reputation or abandoning projects that are no longer strategically important (West & Gallagher, 2004, 2006a, 2006b). However, potential contributors face a huge learning effort and do not have any emotional ties to the code since it was developed by someone else, thus reducing their intrinsic motivation. Moreover, building a community around existing projects requires resources and that the sponsors relinquishes control.

Gonzalez-Barahona et al. (2007) point out that this step can be disruptive, as the contributor base of the Mozilla project changed completely when AOL spun it off into the Mozilla Foundation. On the contrary, Gonzalez-Barahona et al. (2013) state that the original creator of OpenStack – Rackspace – is still responsible for the majority of the code base. However, other members account for the majority of new commits, thus increasing diversity and making the project less dependent on Rackspace. Moreover, the recognize the need of formal rules and a neutral foundation to handle tensions between companies and individuals (Gonzalez-Barahona & Robles, 2013).

2.2.3. Conflicts in Open Source Projects

Most researchers trace conflicts back to different interests of corporate and voluntary participants. Freeman and Siltala (2004) and Siltala et al. (2007) examine two cases where corporate sponsors tried to steer the development via financial rewards or wanted to close parts of the code, thus violating the philosophy of Free/Libre and Open Source Software (FLOSS).

Dahlander and Magnusson (2005) also see tensions as a consequence of corporate behavior, which can be classified as symbiotic, commensalistic and parasitic, based on whether companies respect and give back to the community.

Since such tensions may destroy projects, Van Wendel de Joode (2004) argues that conflict management is essential and identifies four mechanisms: Third party intervention through mediators or arbitrators, code modularity to increase independence, parallel development lines

to allow multiple solutions and the option to fork the project.

Elliott (2003); Elliott and Scacchi (2002, 2003, 2004) and Jensen and Scacchi (2005) argue that discussion on persistent and public channels –such as IRC or mailing lists– resolve conflicts. Additionally, projects need shared beliefs and values and should allow newcomers to participate.

However, resolution mechanisms require people to be willing to collaborate, since there are no effective penalties (Raymond, 1998). This is also acknowledged by Lattemann and Stieglitz (2005), who claim that the lack of formal authority and financial incentives forces projects to rely on trust and peer reviews.

2.2.4. Differentiation

Although these papers have covered open source foundations, conflicts or even some of the specific cases, this thesis differs from previous work in several respects. First, most papers on foundations neglect whether these were started by communities or companies. In fact, the specific case of donated projects is often disregarded. Consequently, conflicting interests between the donor and other companies have not been examined in great detail. Finally, conflicts are only recognized within a community of individuals or between a single company and individuals.

2.3. Research Process

This section summarizes the research process which is also shown in Figure 2.1[17].

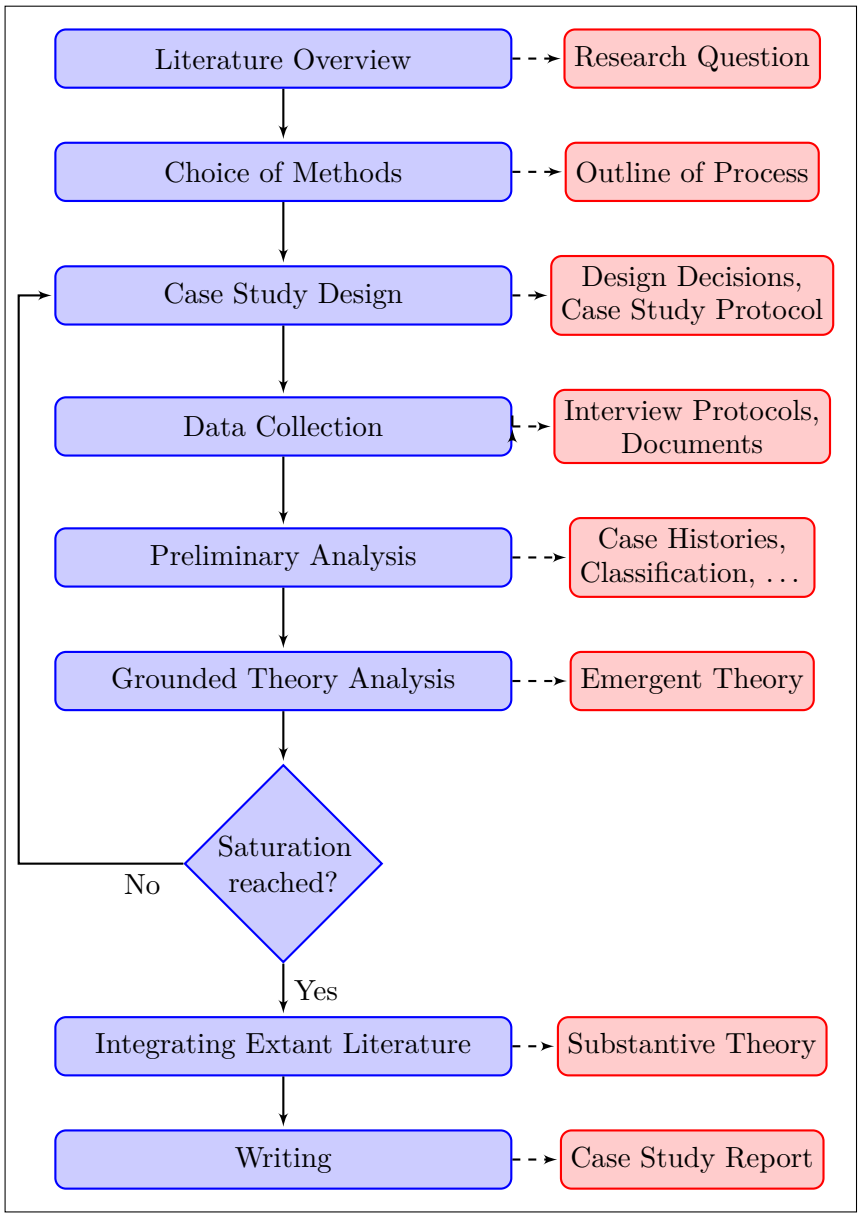


Figure 2.1.: Steps in the research process (blue) and their results (red).

As previous studies on open source foundations did not address the specific situation of conflicts in company-started projects, we decided for a theory-building approach by using an exploratory multiple-case study and grounded theory analysis (Eisenhardt, 1989; Yin, 2013; A. L. Strauss & Corbin, 2008).

2.3.1. Case Study Design

Conducting a multiple-case study allowed us to employ a replication logic to generalize case findings (Yin, 2013). The primary unit of analysis was a nonprofit foundation that hosted an open source project donated by a company[18]. The embedded units were the foundation itself and the participating companies, with a focus on the donor, that is, the former owner of the project.

We selected four cases based on their unique characteristics, thus using "theoretical sampling" (Glaser & Strauss, 2009, p. 45) of "polar types" (Eisenhardt, Graebner, Huberman, & Miles, 2007, p. 27). We considered the date of the donation, whether the respective foundation had existed before and whether it accepted corporate members[19]. All projects had been donated within the last decade and their respective donors are still active.

We did not start with a preexisting theory, but followed Yin (2013) by using a case study protocol and a database.

2.3.2. Data Sources

As can be seen in Table 2.4, we used multiple data sources to allow triangulation (Yin, 2011). Data collection and analysis happened simultaneously until theoretical saturation was reached (Charmaz, 1996)[20].

Table 2.4.: Data sources by case and step of analysis[22].

	Preliminary	**Grounded Theory**			
Case	**Documents**	**Interviews[23]**	**Documents**	**Podcasts**	**Videos**
ACS	24	1	20	4	3
CF	69	-	17	1	-
EC	15	1	10	-	3
OSt	81	1	26	5	4

Interviews. For each case, we contacted potential interview partners who had been active in the project for several years and who were (former) employees of the respective donor of the project[24]. With the exception of Cloud Foundry, we had one semi-structured interview per foundation. The interviews lasted between 45 and 90 minutes and were conducted via Skype, recorded and then transcribed. We refined the questions after each interview to incorporate new insights[25].

Governance documents, meeting minutes. Foundation bylaws, meeting minutes of board meetings and mailing list discussions allowed us to investigate the structures, rules and history of the foundations and their projects.

Blogs and announcements. We examined blog posts, press releases and announcements that were created on behalf of the foundations or their members. Additionally, we considered reports by media outlets.

Videos and podcasts. We partially transcribed public recordings of interviews with foundation representatives and conference videos.

2.3.3. Data Analysis

This process consisted of two steps: preliminary analysis and grounded theory analysis. The whole process was iterative since we re-visited previously collected data after new insights had emerged.

Preliminary analysis. Before conducting the interviews, we created a chronology of the most important events in the histories of the foundations[26]. Additionally, we classified the foundations based on a scheme created by Riehle and Berschneider (2012)[27] and identified governance structures as well as the most important entities within the foundation. Moreover, we created an overview of participating companies and tracked the affiliations of contributors and board members[28].

Grounded theory analysis. We used MAXQDA[29] to analyze official blog posts and announcements, the interview transcripts and the partial transcripts of the videos and podcasts while following the approach of Corbin and Strauss (1990)[30]. First, we coded the data by assigning labels to text fragments (open coding). By constantly comparing codes to other codes and text fragments, we combined similar codes into categories. Additionally, we established connections between categories and their subcategories, as well as between different categories (axial coding). Finally, we focused on a set of categories around a "core category" (Esteves et al., 2002, p. 130) and developed their relationships (selective coding). We wrote analytic memos throughout analysis (Charmaz, 2008).

After a theory had emerged, we compared its findings to existing literature. The present thesis uses a "theory-building logic" (Yin, 2013, p. 189) where individual case reports are omitted in favor of the overarching theory.

2.4. Theory

The present theory provides the following answers to the research questions:

- *RQ1*: While all foundations employed conflict resolution mechanisms (Section 2.4.3), more effort was spent on conflict prevention by prohibiting bad behavior of their participants (Section 2.4.2).
- *RQ2*: Foundations also relied on these preventive mechanisms to protect their own interests.
- *RQ3*: Although the donors were also subject to these mechanisms, they had several possibilities to protect their interests after giving up control (Section 2.4.5).

Section 2.4.1 explains factors that could lead to conflicts.

2.4.1. Causes: Competition, Different Interests and Bad Behavior

As summarized in Figure 2.2, the following factors encouraged bad behavior, thus leading to conflicts.

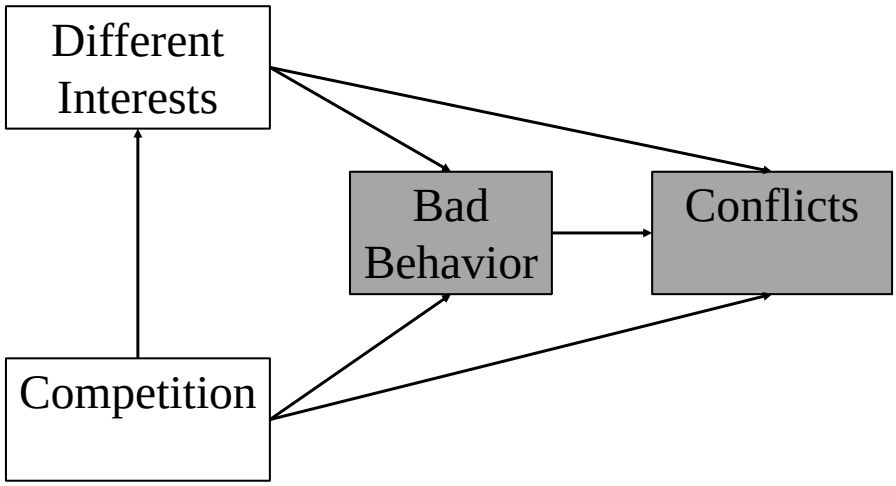


Figure 2.2.: Causes of conflicts.

2.4.1.1. Competition

Donating a project to a foundation attracted more companies, thus potentially reducing the influence of the donor. For example, Rackspace was only the third most active company in regards to the number of commits and mailing list activity for the latest software release[31].

Additionally, new members also included competing companies, such as HP and Borland in the case of IBM and the Eclipse Foundation ($\text{Interviewee}_{\text{EC}}$). The donors had to expect this to happen:

> *"So one of the skill-testing questions we ask folks that want to bring projects to Eclipse is 'Ok, who is your direct competitor? Alright, now imagine they put 20 people on that project and out-code your guys. Are you still ok with this?"'* (Executive Director, Eclipse Foundation)[32]

In fact, donors even welcomed some of their competitors as long as they helped to establish a standard against the most dominant ones:

> *"Having others joining was exactly what should have happened, right? I mean you had in a way 'One enemy' and that one enemy was Microsoft. Everybody else - whether they were a competitor or not for you - was not really it. It was not really an issue."* ($\text{Interviewee}_{\text{EC}}$)

Competition inside a foundation increased if members were targeting the same customers, the market potential was huge or the technology was disruptive. However, $\text{Interviewee}_{\text{ACS}}$ pointed out that fierce competition might damage collaboration ("shark tank"), thus potentially threatening the survival of the project.

2.4.1.2. Different Interests

Besides competing companies, other stakeholders also pursued different interests, as shown in Table 2.5.

2.4.1.3. Bad Behavior: Domination & Takeover

The interview partners suggested that bad behavior of foundation members was a likely cause of conflicts. In this context, bad behavior is defined as a set of actions executed by a member to enforce his own interests at the expense of other members and the foundation. For example, he might try to dominate or even take over completely, thus motivating other members to leave the foundation[36]. As a result, foundations had to balance the interests of their members and make sure that those get along with each other: "Yeah, we will behave nicely." ($\text{Interviewee}_{\text{EC}}$).

2.4.2. Prevention

As can be seen Figure 2.3, foundations employed rules, strategies and processes to prevent bad behavior of their participants, while also relying on values and common interests.

2.4.2.1. Common Interests

Common interests and similar motivation of participating companies made them behave in a collaborative way, thus preventing conflicts. These interests could be observed on both the technical and the business level.

Table 2.5.: Examples of conflicting interests between pairs of stakeholders.

Stakeholders	**Interests**	**Examples**
Foundation Individuals	Turnover Career	"The foundation is interested in having ... some reasonable turn over on committers. What that means is that you know older - or more accomplished committers I should better say - leave the foundation and newer people come in because that actually means that the mind share in the younger developer community is alive." (Interviewee$_{EC}$) "That precedent started at the beginning with the creation of Glance, a project that never should have existed, and the subsequent creation of PTLs. The dynamics of the perceived prestige of a PTL superseded other considerations."[33]
Foundation Companies	Protect brand Use brand	"Citrix and some of the service providers and people that are offering products around CloudStack they find that they want to use the trademark in ways that may or may not be favorable to the Apache Software Foundation." (Interviewee$_{ACS}$)
Foundation Projects	Overall health Project health	"But on the other hand as a foundation do you care about if a particular project dies, if at the same point in time let us say you get four additional projects? No, you don't. So the foundation's interest is to prosper. The interest of individuals' projects is to do very well. Some actually don't do very well or didn't do very well and they died. It is a very natural process." (Interviewee$_{EC}$)
Companies Individuals	Brand Culture	"While individuals may [be] highly motivated by culture they are not as motivated by brand; conversely, corporations are highly motivated by brand and compliance and minimally by culture."[34]
Large companies Small companies	Leverage budget Frugality	"And 250 grand for somebody like SAP, that is not a lot of money, you know, it cuts more through other expenses but it is completely unaffordable for smaller companies." (Interviewee$_{EC}$)
Companies Community	Customer needs Own needs	"Citrix tries to serve their customers while staying in sync with the CloudStack community."[35]

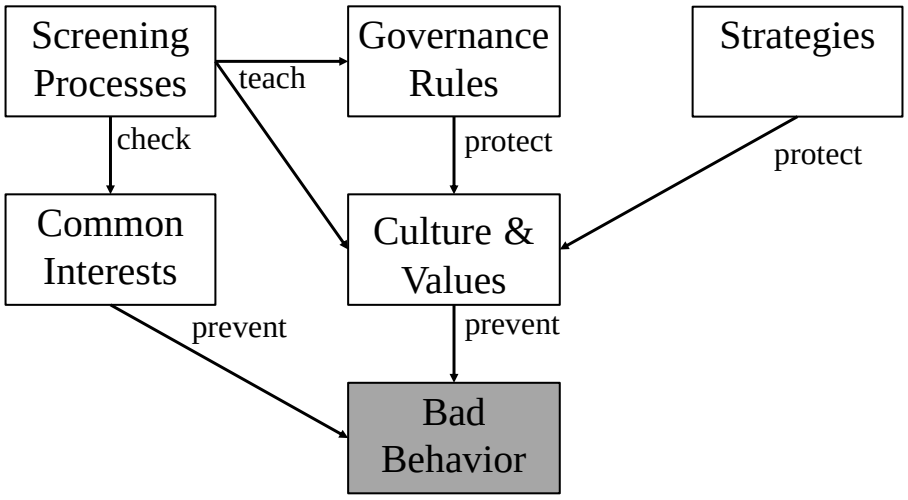


Figure 2.3.: Mechanisms for conflict prevention and their relations.

Technical Level

Interviewee$_{\text{EC}}$ claimed that contributors were "engineers by heart" who valued effective technical solutions more than corporate agendas. Consequently, the outcomes of discussions depended on the technical merit of possible solutions rather than on developers' employment relations.

Business Level

On the business level, companies' participation in open source projects was driven by similar pragmatic interests:

> *"There is nothing altruistic in it or anything like that in none of the open source projects, atleast those that are backed by companies. Those are all crystal clear business decisions."* (Interviewee$_{\text{EC}}$)

Companies only participated in projects on which they had a commercial dependency. For example, IBM, Rackspace and Citrix all spun out their projects to fight their dominant competitors:

> *"You wanted to get unified against Microsoft."* (Interviewee$_{\text{EC}}$)

> *"In order to compete with Amazon they [Rackspace] needed to have software that was like Amazon's. And the only way to get software like Amazon's was to band together every competitor of Amazon and develop that software."* (Interviewee$_{\text{OSt}}$)

> *"CloudStack is Citrix's effort to take on VMware and enlist the rest of the vendor community in doing so."*[37]

Consequently, these companies were interested in growing their former projects to create a unifying standard against these opponents. This implied attracting individual contributors and other companies that shared the same economic motivations. As a result, the donors –and other corporate members– did not do anything that would conflict with their business goals, thus effectively forcing them to behave in a collaborative way. For example, exercising unilateral control would have scared away potential members while damaging the donors' reputation.

Additionally, bad behavior was also prevented because of its costs. Being the dominant member in a foundation meant putting in the most resources, thus rendering this approach ineffective:

> *"So I don't think there is, you know, big conspiracy theories or anything like that around it because money is the big equalizer in this whole game. ... So I guess it all boils down to what I said in the end, the money you spend and the outcome you get out of this is the big equalizer in this game."* ($\text{Interviewee}_{\text{EC}}$)

2.4.2.2. Culture and Values

Culture and its values were important tools for the foundations. For example, $\text{Interviewee}_{\text{ACS}}$ attributed project success to the existence of a good culture. Others agreed with him:

> *"I have seen that culture as a more potent force than the legalese and licenses. While a strong culture reinforces itself, a toxic culture will rot a project like ice cream in the summer."*[38]

Culture was especially important inside the ASF, where it was manifested in the "Apache Way" ($\text{Interviewee}_{\text{ACS}}$). However, cultural values also played an integral part in influencing collaboration in other foundations, as acknowledged by one of the interview partners:

> *"In order to work together you need to have some shared values and goals, otherwise it is not going to happen."* ($\text{Interviewee}_{\text{EC}}$)

Some of the values depended on or influenced each other[39].

Openness

According to the executive director of the Eclipse Foundation, openness was more than publicly available source code: "Openness means being open to everybody who wants to come in and join in"[40]. This also applied to decision-making processes, conferences and product

roadmaps. However, interview partners pointed out that too much openness slowed down decision processes and could scare away potential commercial members. Still, openness was an important value: "Strong communities can best function when work is done openly"[41].

Transparency

Transparency was related to openness and was equally important: "There are expectations of transparency in open source projects"[42]. However, $\text{Interviewee}_{\text{ACS}}$ claimed that transparency could differ between processes. For example, the existence of private mailing lists for legal issues in the ASF made governance less transparent than the completely open development process.

Equality

Especially when foundations accepted corporate members, equality became an issue due to the uneven amount of resources that companies could invest in the project. Consequently, individuals and smaller companies only participated when the foundation provided "equality of opportunity" (Arneson, 2008; Bell, 1971; D. A. Strauss, 1992)[43]:

> *"The main role of the foundation is to make sure there is a level playing field where everybody feels safe."*[44]

As a result, there had to be no single member with special privileges. This was important to $\text{Interviewee}_{\text{OSt}}$, as the resulting inequality would allow members to exploit the project on the expense of others, thus damaging it.

Merit

Especially the ASF acknowledged the value of contributions through merit. Merit was only considered for individual members and did not depend on any employment relations. However, contributions had to be publicly visible so that merit could be recognized.

Neutrality

Another important value was neutrality or vendor-neutrality. Its meaning was not only stressed by $\text{Interviewee}_{\text{ACS}}$, it was also explicitly mentioned in the bylaws of the Eclipse Foundation. Moreover, its executive director explained that "neutrality is probably the single most important thing of all"[45]. In a vendor-neutral environment, no member was treated in a preferred way. Neutrality also created a "safe place" where even competitors could collaborate. It was related to equality and was also an important criterion for attracting new members.

Independence

Foundations and their entities did not want to depend on a single member. As pointed out by the executive director of the Eclipse Foundation, companies assessed the independence of a foundation before joining it[46]. Moreover, the presence of competing corporate members signaled independence:

> *"Having those direct competitors join the fun was a very clear endorsement of the Eclipse Foundation's governance and independence"*[47]

Diversity

A diverse base of members guaranteed the longevity of a foundation since it reduced the dependency on a single member. As expressed by $\text{Interviewee}_{\text{EC}}$:

> *"Absolutely because I mean that [diversity] is what makes your ecosystem survive, right? ... This is an indication of a healthy ecosystem if you have all of those different kinds of players."*

Moreover, a diverse community with different experiences could lead to software that was useful to a wider audience of users.

2.4.2.3. Screening Processes

All cases relied on processes for screening potential members and projects. In the latter case, this was implemented as an incubation process. These processes had the following purposes:

Assessing Motivations and Fit

All foundations checked the motivation of potential members in order to guarantee a good fit with their rules, goals and missions. This applied to individuals, companies and donated projects (Table 2.6).

In the latter case, the foundation also assessed whether the project was a strategic and technical fit. Moreover, it checked whether the donor was interested in long-term support of the project and whether he tolerated external contributors.

Teaching Rules and Values

In the case of the ASF, the incubation process for accepting new projects also aimed at teaching the processes and cultural values of the foundation: "The incubation process is obviously sort of their way to that top level projects as well ... as the developers and participants ... were there to learn the Apache Way" ($\text{Interviewee}_{\text{ACS}}$).

Table 2.6.: Different types of screening processes.

Subject	Example
Committers	"Committed enough for the task and matched the human attitudes required to work well with others"[48]
Companies	"They need to acknowledge that they have in mind the fact that the success of the foundation is the success of their own business" ($\text{Interviewee}_{\text{OSt}}$)
Projects	"The community has learned and demonstrated that it understands the principles and processes laid by the Apache Software Foundation and that it can now operate more autonomously."[49]

$\text{Interviewee}_{\text{ACS}}$ pointed out that the incubation process was a "two way vetting process". Consequently, Citrix could also assess whether the ASF met its needs, thus guaranteeing a better alignment of interests.

2.4.2.4. Governance Structures and Rules

All four foundations had created clear governance structures and rules which were documented in their bylaws. $\text{Interviewee}_{\text{EC}}$ highlighted their importance for companies: "If you think about how you as a company operate you need a clear governance structure around". This opinion was also shared by the executive director of the Eclipse Foundation: "I think governance is extremely important"[50]. Table 2.7 relates these structures to the cases while Table 2.8 shows how values were protected by them.

Transparent Affiliations

Individual members could be forced to disclose their affiliations when joining a foundation. Furthermore, foundations could demand notifications when the employment relations of members changed.

Distributed Decision-Making

The foundations established clear decision-making processes by employing a distributed approach where members could vote on decisions. Depending on the particular issue, voting could be limited to specific foundation entities such as the board of directors, and different types of majority could be required.

Meritocracy

Privileges such as the right to vote and write-access to code repositories were sometimes reserved for members that had contributed a certain amount of work. For example, both

Table 2.7.: Examples of governance structures and rules.

	Cases			
Rule	**ACS**	**EC**	**OSt**	**Example**
Transparent Affiliations		√	√	"I will promptly update any change in my Affiliate status as defined in the Bylaws."[51]
Distributed Decision-Making	√	√	√	"The decisions are made by the vote, as required in our by-laws." ($\text{Interviewee}_{\text{ACS}}$)
Meritocracy	√	√	√	"It is meritocracy and he did a good job that is why he is in the position he is." ($\text{Interviewee}_{\text{ACS}}$)
Decoupling Funding From Control	√	√	√	"You cannot just, you know, shower the foundation in money and then you get all of the power. It is not going to happen." ($\text{Interviewee}_{\text{EC}}$)
Separation of Powers	√	√	√	"There is also very strong separation between the technical decisions and the other things like the management in general, the general management of the foundation." ($\text{Interviewee}_{\text{OSt}}$)
Tiered Membership		√	√	"Tiered structure is exactly to give representation to big companies and to smaller companies and to individuals who are part of a larger free software and open source community who want to care about this project." ($\text{Interviewee}_{\text{OSt}}$)
Representation Limits		√	√	"No more than two directors shall be Affiliated (the 'Director Diversity Requirement')."[52]
Independent Entities	√	√	√	"The Executive Director may not be an employee, officer, director or consultant of any Member of the Eclipse Foundation."[53]

$\text{Interviewee}_{\text{ACS}}$ and $\text{Interviewee}_{\text{OSt}}$ mentioned that even employees of small companies managed to reach high ranks in the respective foundation because of their individual contributions.

Decoupling Funding From Control

While the ASF had limited company influence completely, even the foundations that allowed corporate members had created mechanisms to decouple sponsorship from control. Although the latter offered different membership levels based on the amount of funding, $\text{Interviewee}_{\text{EC}}$ pointed out that there were "safeguards" which prevented complete control by sponsors.

Separation of Powers

Foundations limited the power of specific entities by separating technical from general management issues. For example, the boards of the cases could only make legal and management decisions, while the technical authority was held by separate committees such as Project Management Committee (PMC)'s.

Tiered Membership

In order to address the resource inequality of participating companies and individuals, some foundations offered different tiers of membership, depending on the size and resources of their members. Additionally, each member class could send representatives to the board and other entities.

Representation Limits

When transparent affiliations were enforced, the diversity of the board and other committees could be monitored and representation limits could be established. As a consequence, each corporate member was only able to send a limited number of its employees into such an entity. Moreover, the votes of multiple employees that had the same employer were counted as one.

Independent Entities

Although foundations were independent entities by themselves, they also created independent bodies within their structures and recruited independent staff. Moreover, they monitored whether the participants that voted for a specific decisions were independent of each other.

2.4.2.5. Strategies

In addition to governance structures and rules, foundations employed specific strategies to protect their values and to enforce good behavior. Table 2.9 relates these strategies to the cases while Table 2.10 shows how values were protected by them.

Table 2.8.: How governance structures protected values.

	Values						
Rule	**Transp.**	**Open.**	**Equ.**	**Merit**	**Neutr.**	**Indep.**	**Div.**
Transparent Affiliations[54]	√				√	√	
Distributed Decision-Making		√	√		√	√	
Meritocracy		√	√	√	√		
Decoupling Funding From Control			√		√	√	
Separation of Powers						√	
Tiered Membership			√				√
Representation Limits						√	√
Independent Entities			√		√	√	

Table 2.9.: Examples of prevention strategies.

	Cases			
Strategy	**ACS**	**EC**	**OSt**	**Example**
Monitor Behavior	√	√	√	"The mission of the foundation is to make sure that all the companies and all the groups that are involved into development of the project actually behave."[55]
Allow Community Participation	√	√	√	"Bylaws and legal documents for community review."[56]
Enforce Public Communication	√	√	√	"The Apache mantra is, if it doesn't happen on the list it didn't happen kind of thing. So the mailing list discussion is important." ($\text{Interviewee}_{\text{ACS}}$)
Project-Specific Strategies	√	√	√	"So at this very moment we sent in a few committers to also have the other implementation in that project." ($\text{Interviewee}_{\text{EC}}$)

Monitor Behavior

Due to the transparent processes, foundations were able to monitor the behavior of their members in specific projects and committees. If they noticed bad behavior, they could react by sending in committers (see below). Additionally, both the Eclipse Foundation and the OpenStack Foundation made metrics such as company activity publicly available.

Allow Community Participation

Foundations tried to involve their members in as many decisions and processes as possible. For example, any chances in governance should be made subject of community review, so that members could provide their feedback.

Enforce Public Communication

Foundations had established public and persistent mailing lists where all decisions had to be published and discussed. Additionally, private mailing lists were limited in number and had to be justified.

Project-Specific Strategies

If a specific project was dominated by the employees of a single corporate member, the foundation could send in independent committers, terminate the project or create a competing project.

Table 2.10.: How prevention strategies protected values.

	Values						
Strategy	**Transp.**	**Open.**	**Equ.**	**Merit**	**Neutr.**	**Indep.**	**Div.**
Monitor Behavior[57]	√	√	√	√	√	√	√
Allow Community Participation	√	√					
Enforce Public Communication	√	√					
Project-Specific Strategies						√	√

2.4.3. Resolution

Analysis led to the following three resolution strategies.

2.4.3.1. Building Consensus Through Public Debates

Similar to traditional open source projects[58], conflicts were resolved through public debates and consensus-building: "arguing is essential to solving conflicts"[59]. As a result, members and foundation entities were involved to reach a satisfactory solution. While most of the discussions happened on persistent mailing lists, foundation members recognized that face-to-face meetings such as conferences helped resolve conflicts more effectively. However, reaching consensus was difficult when there were many diverse opinions.

2.4.3.2. Majority Voting

Additionally, conflicts were resolved through voting, thus following the interests of the majority. There were different quora and types of votes –including vetos– to consider issues of different importance. Although this strategy could be used instead of trying to reach consensus, the members and bylaws of the ASF discouraged this approach:

> *"We know how to build consensus and we operate that way. There are *very* few cases where we call a vote to *force* a direction. Votes are used to examine whether consensus is present, rather than to make a decision."* (Director, ASF)[60]

2.4.3.3. Creating Alternative Implementations

Even if no consensus could be reached for a technical issue, the disagreeing parties were free to implement their respective ideas, thus creating alternative implementations.

All cases relied on consensus-building and voting. However, the success of these strategies depended on preventive mechanisms. Debating was only effective when openness and transparency of communication were guaranteed. Additionally, fair voting required the existence of a diverse community with an equal weight of votes.

2.4.4. Influencing Factor: Foundation Type

It became evident that the ASF was a user-led foundation, while the others were vendor-led. This distinction also influenced which of the aforementioned structures and strategies were used. Table 2.11 provides an overview of the relevant differences.

2.4.4.1. Individuals vs. Companies

While vendor-led foundations explicitly recognized companies as sponsors and members, ASF committers were clear on this issue:

Table 2.11.: Two types of foundations.

Type	Creators	Company influence	Affiliations	Emphasis
User-led	Individuals	Limited	Hidden[61]	Rules, meritocracy, culture
Vendor-led	Companies	Encouraged	Transparent	Rules

> *"We take a lot of care to avoid any company influence in our projects. ... For us, at the extreme, I would say that company doesn't exist."*[62]

Consequently, the ASF focused more on individuals, thus limiting the influence of their employers. As $\text{Interviewee}_{\text{ACS}}$ pointed out, this even meant that companies –including the donors– could not promote and sponsor specific projects within the ASF.

2.4.4.2. Affiliations

As a result of its focus on individuals, the ASF had a special approach for handling the affiliations of its members:

> *"Apache very much emphasizes the individual to the extent of hiding the corporate affiliations of its participants"*[63]

Consequently, it did not require the disclosure of member affiliations. Moreover, talking about affiliations was considered an "ASF faux-pas"[64].

As a result, strategies such as monitoring affiliations could not be used.

2.4.4.3. Culture versus Rules

Although foundations of both types relied on governance structures and rules, the ASF also emphasized culture and values. For example, its cultural principles –the Apache Way– and meritocracy were explicitly mentioned in the bylaws of both the foundation and the CloudStack project. Similar behavior could not be observed at vendor-led foundations. Consequently, the role of culture might be a result of the different origins of the foundations[65].

2.4.5. How Did Donors Protect Their Interests?

The interview partners confirmed that the donors irreversibly gave up control and assets. As $\text{Interviewee}_{\text{OSt}}$ put it,

> *"So basically it [Rackspace] demoted itself from steward of the project ... into a player like anybody else where your voice needs to be demonstrated and shown."*

Consequently, donors had to "trade control for influence"[66]. This was accomplished with the following strategies.

Leveraging Project Origin

Since the donors created their projects, they had established the culture and could set initial standards. Moreover, they employed all of the initial contributors, thus being able to steer the technical direction of the project.

Complying with Meritocracy

Similar to other members, donors could make their employees contribute more in order to increase their influence, thus leveraging meritocracy. Moreover, they could try to convince others of their plans.

Investing Resources

In business-friendly foundations with membership tiers, companies increased their influence through sponsorship: "Those board seats are a function of how much you sponsor, sponsorship dollars" ($\text{Interviewee}_{\text{ACS}}$). Alternatively, they raised the number of their contributors by assigning more employees to the project or by hiring existing contributors. However, both approaches were limited by governance structures and the specific foundation type.

2.5. Discussion

This section discusses interesting topics that became apparent during analysis.

2.5.1. Role of Trust

While all interview partners mentioned the effectiveness of their governance and culture, they also acknowledged that the power of foundations was limited. As $\text{Interviewee}_{\text{EC}}$ explained, "You cannot really go and shutdown the project if the project is still operating reasonably well". Moreover, foundation did not have formal authority over their individual members since all of them were volunteers. This was even more important in the ASF as it had to rely on the promise that its members really acted independently of their affiliations.

As a result, foundations had to trust their members. One foundation member also pointed out that especially conflict resolution required trust[67]. Yet, $\text{Interviewee}_{\text{EC}}$ acknowledged the possible presence of "evil spirits"[68].

As a result, it is still unclear how important trust really is, especially since the existing literature on this subject makes opposing claims[69].

2.5.2. Effectiveness of Non-Affiliation

It is also interesting whether non-affiliation solves the problem of commercial interests instead of merely hiding it. $\text{Interviewee}_{\text{EC}}$ pointed out that the underlying business interests still existed: "It doesn't really change the economics of those projects behind the scenes."

Moreover, unlike foundations, companies do have formal authority over their contributors. Consequently, contributors may be forced to follow the agenda of their employer. However, Wagstrom et al. (2010) report cases where employees prioritized community needs over those of their employers.

2.5.3. Trade-Offs

Analysis led to the observation that both foundation members and the foundations themselves had to make trade-offs. For example, foundations had to consider whether they should kill a healthy, yet vendor-dominated project. Moreover, the donors had to relinquish control in order to attract other participants[70]. Finally, the coopetition required in such projects forced competing companies to make compromises.

2.5.4. Relevance of Foundation Type

The ASF was an example of theoretical replication since significant characteristics differed from the three vendor-led foundations, but for explainable reasons (different backgrounds). While several strategies and rules were employed by both types of foundations, $\text{Interviewee}_{\text{ACS}}$

pointed out that the artificial creation of vendor-led foundations ("boy band") provided more challenges in terms of corporate interests and competition.

2.6. Limitations and Future Work

This section discusses possible threats to validity, how they were addressed throughout the research and how they might lead to future work.

2.6.1. Limitations

Choice of methods. Eisenhardt (1989) points out that theory-building based on case studies might result in theory that is too complex and contains features that are unique to the specific cases. Moreover, there is no agreed upon way of doing this kind of research as there are two conflicting versions of grounded theory (Strübing, 2007), whose claims are partially incompatible with the case study process of Yin (2013).

Validity of case study. Yin (2013) mentions four threats to validity: construct, internal and external validity as well as reliability. We considered multiple data sources (triangulation) to guarantee construct validity. Internal validity was enhanced by considering extant literature (Eisenhardt, 1989). The use of a replication logic in this multi-case study increased external validity, while the case study protocol and the case study database improved reliability.

Quality of analysis. Coding was done by one researcher alone and did not follow the advice of Urquhart (2000) and Eisenhardt (1989) to attend coding seminars or to consult grounded theory experts. We solely relied on papers about the grounded theory methodology, thus possibly leading to coding errors and misinterpretation (Fernández, 2004a).

Limited number of cases. We examined only four cases for this thesis. However, Eisenhardt (1989) explicitly recommends between four and ten cases and acknowledges that the number of cases can be limited because of pragmatic reasons. Stoecker (1991) and Creswell (1998) even claim that having more than four cases reduces the contributions of individual cases (as cited by Lauckner et al. (2012)). Finally, Yin (2013) points out that the aim of case study research is analytical generalization which is not based on random sampling.

Limited number of interviews. Limited time and access to potential interview partners led to only three interviews. Consequently, we cannot claim to have reached complete theoretical saturation as the third interview still provided new insights. We compensated for this by analyzing additional documents. Moreover, Eisenhardt (1989) and Willig (2013) state that reaching saturation might not be realistic.

Candidate bias. The four cases overlap in regards to certain characteristics since (a) all foundations and donors are registered in the US, (b) three of the four projects are from the domain

of cloud computing and (c) companies like IBM and HP are involved in multiple foundations. Moreover, the interviews might have provided a biased point of view since all interview partners were still active in the respective foundations and they were not fully involved in the decision processes when the projects were spun out.

2.6.2. How Good Is the Resulting Theory?

Despite the limitations mentioned before, the theory presented in this thesis is a substantive one due to the application of several best practices. For example, the research followed a case study protocol and a clear chain of evidence was maintained by linking the results to evidence in a case study database. Moreover, the use of several data sources allowed triangulation. Relying on multiple cases and a replication logic improved the quality of the findings, thus addressing a possible lack of rigor (Shakir, 2002). Moreover, we employed constant comparison and memoing during analysis to increase theoretical sensitivity (Hallberg, 2010). Additionally, the emerging theory was compared to existing literature, therefore making it a substantive theory (Fernández, 2004b). The review of several grounded theory methodology papers and exemplary case studies provided a better understanding of the methodology. Furthermore, the research process was compared to checklists created by Runeson and Höst (Runeson & Höst, 2008; Höst & Runeson, 2007) and all of its steps and decisions have been discussed in great detail throughout this thesis. Finally, the reader has to keep in mind that the aim of this thesis was theory-building, not confirmation.

2.6.3. Future Work

Future research projects may address the limitations of this study by including more cases from diverse backgrounds. Additionally, additional interviews should be conducted to get the input of more stakeholders, preferably from different positions inside donors and foundations.

Since the aim was theory-building, a validation approach might also warrant future work. Moreover, the issues raised in Section 2.5 should be examined more closely.

2.7. Conclusions

This thesis presented an exploratory multiple-case study of four open source projects that had been created by companies, but were then donated to independent nonprofit foundations. After demonstrating the significance of this phenomenon, we outlined how this research fits into existing literature and identified a research gap regarding conflicting interests of foundation members.

Using grounded theory analysis of interviews and documents, we discovered that foundations created specific mechanisms to resolve conflicts. However, they also relied on a mix of governance structures, strategies, values and common interests to prohibit bad behavior of their members, thus preventing conflicts. Moreover, foundations leveraged these preventive mechanisms to protect their own interests. We also outlined several strategies for the project donor to protect his interests after giving up control.

Next, we discussed the impact of the foundation type on these mechanisms and raised further questions regarding the importance of trust, culture and non-affiliation. In addition to these questions, the limitations of our research and a theory-testing approach might warrant future work on this topic.

Finally, we hope that the contributions of this thesis help companies to better understand the phenomenon of company-started projects in nonprofit open source foundations, especially if they are facing a similar decision as the companies in the four cases did.

2.8. Acknowledgements

First of all, I would like to express my deepest gratitude to my interview partners for sharing their time and expertise with me: Mark Hinkle, Stefano Maffulli and the anonymous interview partner at the Eclipse Foundation. Additionally, I am very grateful to Professor Dr. Dirk Riehle and Ann Barcomb, MSc for their invaluable feedback and guidance throughout the research process.

Research Notes

1. Several articles and surveys highlight their importance (Lomas, 2014; Lynch, 2014; Pingdom, 2009; Shimel, 2012; W3Techs, 2014).

2. Linux Kernel: `https://www.kernel.org/`
   Android: `https://source.android.com/`
   GENIVI: `http://projects.genivi.org/projects`

3. Google: `http://code.google.com/intl/de/opensource/projects.html`
   Facebook: `http://developers.facebook.com/opensource/`
   Apple: `http://www.opensource.apple.com/`
   Microsoft: `http://www.microsoft.com/en-us/openness/default.aspx`

4. For this thesis, the terms "foundation", "nonprofit foundation", "open source foundation" and any combination thereof are used interchangeably to describe a legal entity that (a) complies with either 501(c)(3) or 501(c)(6) of the US Internal Revenue Service (IRS) and (b) owns the copyright of one or more software projects that are covered under an Open Source Initiative (OSI)-approved license.

5. The relation between projects and foundations is explained in Chapter 3.2.6.

6. "Donor" describes the company that owned the project before donating it to an independent nonprofit foundation.

7. The underlying software was created by a startup called VMOps, which latter changed its name to Cloud.com and was finally acquired by Citrix in 2011. More details are provided in Section B.1.

8. CloudStack entered the Apache Incubator in 2012 and graduated to a top-level project in 2013 (see Section B.1).

9. Expected date.

10. In the case of OpenStack, this refers to the corporate sponsor, Rackspace. The original OpenStack project was also created with intellectual property from NASA. However, NASA announced its retreat from the OpenStack project in 2012 (see Section B.1).

11. This field describes the status of the foundations according to US IRS. 501(c)(3) applies to charitable organizations, while 501(c)(6) includes trade associations (US IRS, 2014b, 2014a).

12. Expected date.

13. No details have been released yet.

14. The role of literature review in grounded theory is a controversial topic. More details are explained in Section 3.2.3.

15. Generation.

16. The legal status of the Cloud Foundry Foundation has not been announced yet.

17. A more detailed explanation of all decisions and the rationale behind them is topic of Section 3.2.

18. In the case of CloudStack, we ignored other projects in the ASF since they are independent of CloudStack. See Chapter 3.2.6 for more details.

19. See Tables 2.1 and 2.2 for the respective values.

20. We did not reach complete saturation, as explained in Section 2.6.1.

21. Data sources were counted multiple times if they were used for several cases or in both steps of analysis. We combined multiple on-line resources into a single document if they contained only a small number of relevant statements. Consequently, the actual number of distinct resources is higher.

22. Data sources were counted multiple times if they were used for several cases or in both steps of analysis. We combined multiple on-line resources into a single document if they contained only a small number of relevant statements. Consequently, the actual number of distinct resources is higher.

23. Although the podcasts and some of the documents also contained interviews, we only counted those that were conducted by us.

24. This allowed us to ask them about the foundation and the decision processes that led to the donation.

25. This was based on the advice of Eisenhardt (1989), Charmaz (1996) and Corbin and Strauss (1990). See Chapter C for the interview questions.

26. See Section B.1.

27. Section 3.1.2 contains the classification of the nonprofit foundations related to the four cases in this thesis.

28. See Chapter B for some of the documents from the case study database.

29. A tool for qualitative data analysis (`http://www.maxqda.com/`).

30. See Section 3.2.7 for a more thorough description of the analysis process.

31. `http://activity.openstack.org/dash/releases/index.html?data_dir=data/icehouse`

32. “Open Source Foundations Considered Helpful” in MAXQDA document “Eclipse Videos”.

33. MAXQDA document “OpenStack A Plea”.

34. MAXQDA document “OpenStack A Plea”.

35. MAXQDA document “Steve Wilson – Help me help you”.

36. For example, Rick Clark, who was one of the founders of the OpenStack project, left it after Rackspace had unilaterally changed the governance structures. However, this had happened before the foundation was created. Another example is Citrix, which left the OpenStack Foundation because of different interests and started to focus on CloudStack (see Section B.1).

37. MAXQDA document “Ecosystems in conflict”.

38. MAXQDA document “Kicking off discussion about OpenStack Core”.

39. See Section 3.3.1.

40. “Open Source Foundations Considered Helpful” in MAXQDA document “Eclipse Videos”.

41. MAXQDA document “An Apache Way Primer”.

42. MAXQDA document “Rick Clark Blog Posts”.

43. Riehle (2011, p. 94) describes this as an “egalitarian approach”.

44. MAXQDA document “The Future of Open Source Foundations”.

45. “Open Source Foundations Considered Helpful” in MAXQDA document “Eclipse Videos”.
46. MAXQDA document “An Exclusive Interview With Mike Milinkovich Of Eclipse Foundation”.
47. MAXQDA document “The Eclipse Foundation”.
48. MAXQDA document “How the ASF works”.
49. MAXQDA document “The CloudStack community by the numbers”.
50. MAXQDA document “Mike Milinkovich”.
51. MAXQDA document “OpenStack - Register as Individual Member”.
52. MAXQDA document “BYLAWS OF THE OPENSTACK FOUNDATION”.
53. MAXQDA document “Eclipse BYLAWS 2011 0815 Final”.
54. See Section 3.3.3.1 for an explanation why specific governance structures protect certain values.
55. MAXQDA document “The Future of Open Source Foundations”.
56. MAXQDA document “OpenStack - websites”.
57. See Section 3.3.3.2 for an explanation why specific strategies protect certain values.
58. “Traditional” refers to community-managed open source projects that are not protected by a foundation. The use of mailing lists is also recognized by Elliott and Scacchi (2004) and Jensen and Scacchi (2005) (see Section 2.2.3).
59. Video “Rackspace Keynote - Just Rebels - Or A Rebel Alliance” in MAXQDA document “RackSpace Videos”.
60. MAXQDA document “Re Requesting clarification in ByLaw text”.
61. Affiliations are only included in the incubation proposal for a new project in order to asses whether there are multiple entities in support of this project. However, even this rule is controversial. See MAXQDA documents “Re Too many licenses” and “A Guide To Proposal Creation - Affiliations”.
62. MAXQDA document “The Future of Open Source Foundations”.
63. MAXQDA document “The Future of Open Source Foundations”.
64. MAXQDA document “Why CloudStack is not a Citrix project”.
65. Section 3.3.4 discusses the different backgrounds of foundations and relates them to existing literature.
66. MAXQDA document “Rick Clark Blog Posts”.
67. Video “Rackspace Keynote - Just Rebels - Or A Rebel Alliance” in MAXQDA document “RackSpace Videos”.
68. O’Mahony and Ferraro (2004, p. 18) describe this problem as “trojan contributors”.
69. Section 3.3.5 discusses this issue in greater detail.
70. Brandenburger and Nalebuff (1996, p. 38) note that “Sometimes the best way to succeed is to let others do well, including your competitiors” (as cited by Loebecke et al. (1999)).

3. Elaboration of Research Chapter

3.1. Classification of Open Source Projects

This section contains a more detailed classification of the single cases.

3.1.1. Comments on the Classification in Related Literature

This section explains the contents of Table 2.3 and justifies the classification.

- Capiluppi, Stol, and Boldyreff (2012, p. 179) subdivide the category of sponsored open source into "industry-led" and "industry-involved". In industry-led projects, one company is in control and is also essential for the survival of the project. The authors state that the Eclipse Foundation is an example for this type of project. However, one might argue whether this particular example is consistent with the definition since there are several active corporate members. In general, the paper fails to address the role of nonprofit foundations since it is not evident whether such projects are either industry-led or industry-involved.

- Letellier (2008) identifies three generation of open source projects: The first one describes a community of individuals without formal structures, the second one refers to a foundation made of individuals, while the third describes a foundation that also accepts corporate members.

- O'Mahony and West (2005) and West and O'Mahony (2008) are discussed in Section 2.2.2.

- Riehle and Berschneider (2012) classify foundations based on their incorporation. The entire classification can be found in Section 3.1.2.

- Wasserman (2013) sees foundations as a subcategory of community-based open source projects.

- Watson et al. (2008, p. 42) describe single-vendor open source companies as "second-generation open source (OSSg2)". He explicitly recognizes projects that are sponsored by nonprofit foundations as part of the first generation of open source.

- According to West and Gallagher (2004, p. 8), a "spinout" project describes a former closed-source software project that was made open source by its original creator: "A sponsor of an internally developed software project releases its code to the public under an open source software license" (West & O'Mahony, 2005, p. 3). However, the creator may retain the copyright (i.e. single-vendor open source) or donate it to a nonprofit foundation.

3.1.2. Classification of Riehle and Berschneider

The classification in Figure 3.1 is based on the work of Riehle and Berschneider (2012) who had already applied their scheme to the ASF and Eclipse. Additional sources were used to cover the cases, as explained in Chapter B.

Several aspects have to be noted:

- We excluded Cloud Foundry since its final governance structures are still being created.
- Instead of committers and PMCs, OpenStack has core reviewers and Project Technical Leads. Although PTLs are individuals instead of committees, we felt that they have similar roles within the foundation.
- Board membership turned out to be more complex than described in (Riehle & Berschneider, 2012). For example, Sustaining Members and Committer Members in the Eclipse Foundation can elect their class representatives, while Strategic Members appoint theirs. Consequently, we see this as a combination of meritocratic and autocratic membership. The OpenStack Foundations allows both classes of individual (democratic) and paying gold members (meritocratic) to elect their respective class representatives, while platinum members appoint their representatives (autocratic).

	Attribute	Values	Apache CloudStack	Eclipse	OpenStack
General	Purpose	Public benefit	PB	-	-
		Member benefit	-	MP	MP
	Incorporation	Foundation	F	-	-
		Consortium	-	C	C
	Members	Natural persons	NP	NP	NP
		Juristic persons	-	JP	JP
Philosophy	Commercial Stance	FSE	-	-	-
		CSA	CSA	CSA	CSA
	Dev. Model	Open	OD	OD	OD
		Closed	-	-	-
IP	License	Reciprocal	-	-	-
		Permissive	PL	PL	PL
	Patents	No grant	-	-	-
		Use	-	UPG	-
		Use + embedding	EPG	-	EPG
	Ownership	No IP required	-	-	-
		Relicensing	RRG	RRG	RRG
		Copyright transfer	-	-	-
Governance	Board membership	Democratic	-	-	DB / MB / AB
		Meritocratic	MB	MB / AB	
		Autocratic	-		
	Project membership	Democratic	-	-	-
		Meritocratic	MP	MP	MP
		Autocratic	-	-	-
	Natural Member Career	UDC	UDC	UDC	UDC
		PMC	PMC	PMC	PMC
		BRD	BRD	BRD	BRD
	Juristic Member Level	Financing	FL	FL	FL
		Developer	-	DL	DL
Financing	Foundation	Fees	-	MF	MF
		Sponsorship	S	S	S
		Gifts	GG	GG	?
		For-profit subsidiary	-	-	-
	Projects	Foundation	PFF	PFF	PFF
		Members	-	-	-
Operations	Infrastructure	Foundation	IFO	IFO	IFO
		Members	-	-	-
	Back Office	Volunteers	BOV	-	-
		Paid Staff	BOE	BOE	BOE

Figure 3.1.: Classification of cases, based on (Riehle & Berschneider, 2012).

3.2. Elaboration on Research Process

This section discusses the steps of the research process in more detail.

3.2.1. First Literature Review on Open Source Foundations

We had no specific research question in mind when we started the research project. After reading papers on open source developer foundations and looking at some of the existing ones, we became interested in those that were build around a company-founded project. This led to the research questions in Section 2.1.1. Following the advice of several grounded theory researchers, a more in-depth literature review was postponed after the theory had emerged (see Section 3.2.3).

3.2.2. Literature Review on Case Study Research and Grounded Theory

Since no existing literature addressed the specific research questions, we chose an exploratory, theory-building approach that employed Case Study Research (CSR) combined with the Grounded Theory (GT) methodology. This seemed to be an adequate choice for the following reasons:

- Both case study research and grounded theory are good for exploratory theory-building when there is little existing theory (Urquhart, 2000; Benbasat et al., 1987; Eisenhardt et al., 2007).
- Case study research is suitable for answering "how" and "why" questions (Benbasat et al., 1987; Eisenhardt et al., 2007; Garg & Eisenhardt, 2012; Yin, 2013).
- Moreover, case study research is also applicable when exploring contemporary events that are not under the control of the researcher (Yin, 2013; Benbasat et al., 1987). This is especially true when those events are examined in their "real-world context" (Crowe et al., 2011; Zainal, 2007; Yin, 2011, p. 5).
- Finally, several papers have reported successful applications of the combination of both approaches (Eisenhardt, 1989; Eisenhardt et al., 2007; Fernández, 2004a; Halaweh et al., 2008), with one even describing it as a "synergistically combination" (Arshad & Ahlan, 2013, p. 84). Allan (2003) points out that weaknesses of case study research can be addressed by combining it with grounded theory.

The process of case study research was inspired by Yin (2011, 2013), Eisenhardt (1989) and Eisenhardt et al. (2007), while we choose the "evolved" grounded theory approach of A. L. Strauss and Corbin (2008) over the "traditional" version of Glaser (Mills et al., 2006, p. 3). However, as pointed out by Fernandez, the case study approach has to be modified in

order to be compatible with grounded theory (Fernández et al., 2002; Fernández, 2004a). This had an impact on (a) the extent and time of literature review (Section 3.2.3), (b) the role of theory in case study design and (c) the upfront selection of cases (both in Section 3.2.4).

3.2.3. The Role of Literature in Grounded Theory

The role of literature review in grounded theory is a controversial topic (Eisenhardt, 1989; McGhee, Marland, & Atkinson, 2007; Dunne, 2011).

Some researchers believe that no upfront literature review must take place in order to avoid "preconveived notions" (Charmaz, 1996; McGhee et al., 2007, p. 3). Dunne (2011) and Steenhuis and Bruijn (2006) point out that the fathers of grounded theory, Glaser and Strauss, were also against an early literature review. Fernández (2004b, 2004a, p. 52) even claims that a traditional review is "incongruent with grounded theory and methodologically unsound".

On the contrary, others have recognized that an upfront literature review is needed to find out whether the study has been done before (Hallberg, 2010; Dunne, 2011) and to learn from the mistakes of previous studies (McGhee et al., 2007). Moreover, researchers such as Heath and Cowley (2004) state that the requirement of "no previous knowledge" is not realistic. Finally, Fernández (2004a) clarifies that the researcher should simply not have an existing theory when starting research.

For this thesis, we followed an approach inspired by Dunne (2011) by conducting two separate literature reviews: The first one was an overview and aimed at identifying novel research questions. It resulted in a "contextualization" (Dunne, 2011, p. 121) of this work and is presented in Section 2.2.

In the second review, we performed an in-depth review based on the concepts that emerged from analysis. Consequently, its results are integrated into the final theory in Sections 2.4 and 2.5.

3.2.4. Case Study Design

The design of the case study followed Yin (2013).

The role of theory. Although Yin (2013) suggests the upfront development of theoretical propositions, we refused to do so, since any preexisting theory would be against the principles of grounded theory (Fernández, 2004a; Eisenhardt, 1989).

Single- vs. multiple-case study. We decided for a multiple-case study since having several cases increases the rigor of the study (Shakir, 2002), produces a more robust theory (Eisenhardt et al., 2007), allows better generalization than single-case studies (Zainal, 2007), and is generally more suitable for theory-building (Benbasat et al., 1987).

Other design decisions. Yin (2011) identifies additional components of a case study design that are covered in other parts of this thesis: The research questions are part of Section 2.1.1. Data sources are described in Section 2.3.2 while details of the data analysis process can be found in Section 2.3.3. It is important to note that in grounded theory, data collection and analysis happen simultaneously until theoretical saturation is reached (Charmaz, 1996), thus leading to an iterative process (Urquhart et al., 2009). We followed a case study protocol, while a case study database helped organize evidence and data sources. These documents can be found in Chapters A and B.

3.2.5. Case Selection

Since we could not find an interview partner at Cloud Foundry, we decided to add the case of Eclipse in order to reach saturation. As discussed in Section 3.2.5.1, Eclipse was a suitable candidate since both Cloud Foundry and OpenStack had looked at the Eclipse Foundation as a blueprint and we had access to an interview partner.

This was a contradiction to the original thesis description because it –arbitrarily– limited the cases to projects from the domain of cloud computing. However, Section 3.2.5.2 lays out why this constraint turned out to be of little importance. Consequently, it was dropped and Eclipse could be added as an additional case from a different domain.

3.2.5.1. Why Eclipse?

In addition to having a potential interview partner at the Eclipse Foundation, two reasons were crucial for choosing it as the fourth case.

First of all, the Eclipse Foundation was created prior to the OpenStack Foundation and the Cloud Foundry Foundation. Moreover, documents suggest that it served as an inspiration for those two:

> *"At the start of the process, Jonathan Bryce and I spent the first couple of months learning as much as we could about successful open source foundations, like the ASF, Eclipse, and the Linux Foundation, reading foundation meeting minutes into the wee hours of the morning."* (Mark Collier, COO OpenStack Foundation)[1]

A member of the Cloud Foundry Community Advisory Board (CAB) explicitly acknowledged the influence of Eclipse when describing the governance structure as "inspired by Eclipse" (Whelan, 2014).

Secondly, Section B.2 shows that the original creator of Eclipse, IBM, also participates in the other three foundations: It helped found the ASF and is also a founding platinum member of both the OpenStack Foundation and the Cloud Foundry Foundation. Consequently, it is likely that IBM draws on its experience regarding the creation and operation of foundations.

3.2.5.2. The Specifics of Cloud Computing

The initial three cases (Apache CloudStack, OpenStack and Cloud Foundry) were all from the domain of cloud computing. Consequently, we examined what characteristics were unique to them and what the impact of the domain was:

- The dominance of Amazon Web Services (AWS) in the cloud space turned out to be a motivating factor for establishing open standards. Moreover, there were ongoing debates whether the Application Programming Interface (API)'s of AWS should be supported by the software[2].
- Additionally, there were debates which hypervisors should be supported. Two of the biggest players in this market are Citrix and VMware, with the first donating CloudStack to the Apache Software Foundation while the second one spun out Pivotal, the initial sponsor of Cloud Foundry (see Section B.1).
- There is some technical overlap between the projects. While OpenStack and CloudStack compete in the Infrastructure-as-a-Service (IaaS) business, Cloud Foundry is a Platform-as-a-Service (PaaS) offering which can run on top of OpenStack. However, OpenStack tries to offer PaaS capabilities through Project Solum (see Section B.1).
- Additionally, some companies are participating in several of these project. IBM is a founding member of both the OpenStack and the Cloud Foundry foundation. Additionally, its acquisition of SoftLayer provided it with a software offering that is based on CloudStack. It also helped found the ASF. Rackspace did not only create OpenStack but is also a founding member of the Cloud Foundry Foundation. Citrix was a founding member of the OpenStack Foundation, but ultimately left it after acquiring CloudStack and donating it to the Apache Software Foundation in 2012.

However, on a more abstract level, these issues are similar to problems faced by Eclipse:

- The donation of Eclipse was also motivated by the presence of an overly powerful competitor (Microsoft), who at that time dominated the tooling space.
- The debates about the support of specific hypervisors and the AWS API can be abstracted to the issue of compatibility, which was also an issue for Eclipse (hence its pluggable and extendable software architecture).

Consequently, we decided that the present thesis did not have to focus on cloud computing projects alone.

3.2.6. Unit of Analysis: Projects versus Foundations

Because of the different histories of the cases, the definition of the unit of analysis in terms of "project" and "foundation" is complicated and differs between cases.

Apache CloudStack. It started as a single project and remains a separate top-level project inside the ASF. Consequently, it has its own PMC and is independent of other Apache projects, while still being subject to the bylaws and the board of the overall foundation. As a result, the case focuses on the CloudStack project and its relation to the overall rules, structures and entities of the ASF. Other projects inside the ASF are not considered relevant.

Cloud Foundry. Since Pivotal is creating a foundation specifically for its Cloud Foundry project, the unit of analysis is the entire foundation. This is contrary to the ASF where unrelated and independent projects are accepted.

Eclipse. Similar to Cloud Foundry.

OpenStack. This project is the result of two separate projects: The compute engine ("Nova") was created by NASA, while Rackspace donated the storage component ("Swift"). Today, there are additional projects inside the OpenStack Foundation, all of whom are parts of the OpenStack software releases. Consequently, the unit of analysis is the whole foundation.

3.2.7. Grounded Theory Analysis

In the first step, we assigned labels to specific text fragments (open coding). These codes emerged from the data as we did not have any preexisting labels.

For example, we labeled "Yeah the Apache Foundation was clearly one of the big sources of information and material. No question" and "Mike said that it was becoming more difficult to form a classic non-profit organization, but the Cloud Foundry foundation is going to use the same model as the OpenStack Foundation" as *Learning from existing foundations*.

Open coding happened after each interview, where the emergence of new labels made us revisit the previous interviews to recode them accordingly. By using a constant comparative approach, text fragments and codes were constantly compared against each other.

In axial coding, we combined codes into categories and established relations between them. For example, the open codes *No authority over volunteers* and *Prioritizing project health over vendor dominance* formed the category *Limited foundation power*.

As a result, relations between categories became apparent. For example, *Teaching culture and values* and *Screening for common motivation* in the category *Screening processes* helped establish relations to the categories *Culture* and *Common motivation*.

Finally, we ended up with nearly 600 codes and categories that covered over 1100 codings. Next, we started selective coding to find one or few "core categories". Although the category *Conflict resolution* answered our research questions, we discovered that most of the categories related to *Conflict prevention*. We focused on this category by further developing its sub-categories as well as its relations to other categories. Consequently, we identified the causal

relation to *Bad behavior* and the influencing factor *Foundation type*. Prominent subcategories were *Governance*, *Strategies*, *Culture*, *Screening processes* as well as *Values* and *Common motivation*.

3.2.8. Reaching Closure: Theoretical Saturation

Grounded theory analysis has to be carried out until saturation is reached, that is, considering additional data does not provide new insights (Willig, 2013). Consequently, the process was iterative (Pandit, 1996; Calloway & Knapp, 1992; Lingard, Albert, & Levinson, 2008), since we went back to previous steps in the research process and added more data sources.

3.2.9. Integrating Extant Literature

As mentioned by Fernández et al. (2002) and Charmaz (1996), the emerging theory has to be compared to extant literature in order to become a "substantive theory" (Egan, 2002, p. 280). Both differences and similarities with previous work need to be considered in order to improve internal validity and generalizability (Eisenhardt, 1989; Meyer, 2001).

Since we did not want to "pollute" the emerging theory with existing concepts from literature, we followed the approach of Rodon and Pastor (2007) and delayed the in-depth literature research of the emerging concepts until we had reached saturation.

3.2.10. Writing the Report

Eisenhardt et al. (2007, p. 29) points out that limited space leads to a "trade-off between theory and empirical richness". Consequently, we decided to follow a "theory-building logic" (Yin, 2013, p. 189) where the focus is on the overarching theory rather than on rich narrative. As a result, the reports of individual cases were excluded from this thesis to put more emphasis on the "cross-case analysis" (Yin, 2013, p. 186). Yin (2013) and Eisenhardt et al. (2007) mention several examples where no single cases were presented, including (Gilbert, 2005). Since the aim of this thesis is theory-building instead of highlighting the peculiarities of specific cases, this approach was more suitable. Diagrams and summary tables show the most important evidence, while all sources and more data can be found in the case study data base in Chapter B.

3.3. Elaboration on Theory and Discussion

This section provides more details on the theory and its aspects. Additionally, it relates the findings to existing literature.

3.3.1. Relations Between Values

There are relations between the values of open source projects:

Merit versus equality. Meritocracy means equality of opportunity, not equality of results (Arneson, 2008; Bell, 1971).

Merit requires openness. Merit can only be recognized when contributions are publicly available.

Equality requires neutrality. Members are equal when they have the same rights and when the foundation is neutral in regards to each of them.

Diversity increases independence. A diverse membership base reduces the project's dependence on a single member.

Neutrality requires independence. If the project depends on a single member, it is not likely to be neutral as this member is in a privileged position.

Equality requires independence. Similar to neutrality and independence.

Openness requires transparency. The ability to participate in relevant decisions is only possible if decision processes and communication are transparent.

3.3.2. The Origin of Screening Processes

Screening processes such as incubation can be related to formal membership processes in traditional community-managed open source projects (Sharma, Sugumaran, & Rajagopalan, 2002; O'Mahony & West, 2005), which are also called "joining scripts" (von Krogh, Spaeth, & Lakhani, 2003, p. 1227). These processes assess the motivations and capabilities of candidates.

De Laat (2007, p. 172) highlights the importance of such processes, while also relating them to trust: "Distrust is dealt with at the gates of entry, not inside." He acknowledges that not only technical skills, but also loyality and identity are checked.

3.3.3. How Values Are Protected

The following two sections explain the contents of Tables 2.8 and 2.10.

3.3.3.1. Impact of Governance Structures and Rules

Transparent Affiliations foster transparency and enable the foundation to evaluate the diversity and independence of specific projects and entities. *Distributed Decision-Making* allows the community to participate in decisions openly, thus granting equality, neutrality and independence as the decision power is not with a single privileged entity. *Meritocracy* values merit and provides equality of opportunity. If merit is the sole criteria for gaining influence, openness and neutrality are guaranteed. *Decoupling Funding From Control* means that the foundation treats members equally, is neutral in regards to them and does not depend on specific members. *Separation of Powers* prevents that single entities get too powerful, thus providing independence. *Tiered Membership* reduces resource inequality, thus increasing diversity by allowing more members to participate. *Representation Limits* increase diversity and make entities less dependent on single members. *Independent Entities* ensure that members are treated equally and neutrally.

3.3.3.2. Impact of Preventive Strategies

Monitor Behavior can be used to observe whether members respect all of the values. However, additional sanctioning mechanisms are required to enforce them. *Allow Community Participation* ensures openness and transparency by inviting community members to participate. This also applies to *Enforce Public Communication*. *Project-Specific Strategies* such as sending in committers increase diversity and ensure the independence of particular projects.

3.3.4. The Role of Culture

The different importance of culture can be explained with the origins of the foundations:

Since the ASF was founded by a group of individuals ("Apache Group"[3]), its initial influences came from a "'grass roots' community of user-developers" (West & O'Mahony, 2005, p. 1). Consequently, it is likely to reflect the "cultural beliefs and values" (Elliott & Scacchi, 2002, p. 2) of traditional open source communities. This was also named "hacker ethos" by O'Mahony (2005, p. 393) and described as "hacker culture" by von Hippel (2001, p. 3).

On the contrary, researchers point out that companies are motivated by formal rules and structures known from corporate environments (Gonzalez-Barahona & Robles, 2013; Hunter & Walli, 2013; Weiss, 2011).

3.3.5. Literature about Trust

Several researches have acknowledged the importance of trust in both open source and corporate communities:

Lattemann and Stieglitz (2005, p. 9) claim that "social control mechanisms like trust, moral concepts by rituals or ceremonies ... become a crucial part in the governance of open source

communities." Moreover, Sharma et al. (2002, p. 12) name trust as one of the "core assumptions" of culture in open source communities. Jarvenpaa and Leidner (1999) also recognize the role of trust in distributed virtual communities. According to Stewart and Gosain (2006), trust has an impact on the effectiveness of open source contributors. Additionally, the executive director of the Eclipse Foundation describes nonprofit foundations as "trusted agents" (Milinkovich, 2008, p. 38).

In communities allowing corporate members, "network coopetition" (Dagnino & Padula, 2002, p. 14) can be observed. These researchers claim that coopetitive relations between companies are affected by different levels of trust such as "weak trust, semistrong and strong trustworthy behaviors, and even distrust" (Dagnino & Padula, 2002, p. 11). Doz, Olk, and Smith Ring (2000) also point out that trust improves long-term stability in R&D consortia. This point is reinforced by Heikkilä and Kuivaniemi (2012, p. 24) when claiming that business ecosystems are "built on trust".

On the contrary, Gallivan (2001) claims that trust is not as important as control. He identifies explicit mechanisms such as rules and norms, as well as implicit control through individual reputation.

Elaboration Notes

1. (Collier, 2012)
2. OpenStack has a page dedicated to this problem: `https://wiki.openstack.org/wiki/Nova/APIFeatureComparison`
3. MAXQDA document “How the ASF works”.

Appendices

A. Case Study Protocol

This chapter presents the case study protocol that guided this research project. It follows the guidelines proposed by Yin (2013) and Brereton and Kitchenham (2008).

A.1. Overview of the Case Study

A.1.1. Mission and Goals

This case study aims at providing a better understanding of the participation of software companies in non-profit open source foundations. Since we think that there is little contemporary research on this exact topic, we decided for an exploratory case study which tries to build theory from the ground up.

The present case study is part of a master thesis, i.e. it has to be conducted within a timeframe of six months. Moreover, the final results are going to be part of a conference paper. Consequently, its audience consists of a thesis committee at the University of Erlangen-Nuremberg and other researchers interested in open source.

A.1.2. Scope

We examine four open source projects that have been created by a company, and either (a) donated to an existing open source foundation, (b) donated to a newly-founded foundation or (c) are currently in the process of forming a foundation.

A.1.3. Case study questions, hypotheses and propositions

Literature review led to the following three main research questions:

- How do foundations handle conflicting interests of their members?
- How can foundations ensure that their own interests are not impaired?
- How do donors protect their interests after giving up control?

Following the advice of Eisenhardt (1989), we decided not to have any preexisting hypotheses or propositions.

A.1.4. Theoretical framework for the case study

We examine spinout projects, i.e. projects which were created by companies because of commercial interests and then donated to an independent legal entity (foundation).

This scenario contains several stakeholders:

- The foundation
- The company which created the project
- Other companies which participate in the project by either sponsoring or contributing to the project
- Individuals participating in the project
- Government agencies and other non-profits

We assume that the participation of the stakeholders is motivated by different reasons, thus making them pursue their own interests while engaging in the project:

- The foundation acts as a steward for the project.
- The donor still has a commercial interest in its former property.
- Other companies pursue their own commercial interests.
- Volunteer contributors have different motivations than the companies do.

A.1.5. Case study design and selection of cases

This case study follows a holistic multi-case design containing three individual cases. The primary unit of analysis was a nonprofit foundation that hosted an open source project donated by a company. The embedded units were the foundation itself and the participating companies, with a focus on the donor, that is, the former owner of the project.

We defined two criteria for the preliminary selection of case candidates:

- The cases had to be open source projects that were created by a company before being donated to a foundation.
- All projects had to be created within the last decade in order to increase currency of the findings and to ease the access to stakeholders and documents.

Next, we selected four final cases by following a theoretical sampling approach:

- Apache CloudStack was donated to an existing foundation (ASF)

- A dedicated foundation was created to accommodate the OpenStack project
- The CloudFoundry project is currently in the process of moving from a single-vendor open source project to a foundation-owned project
- Eclipse was donated to a new foundation created 10 years ago

A.1.6. Key readings

(Eisenhardt, 1989; Yin, 2011, 2013)

A.2. Data Collection Procedures

A.2.1. Names of contact persons for doing fieldwork

Confidential

A.2.2. Data Sources and data collection plan

We are considering the following data sources:

- Interviews: We will conduct semi-structured with employees of the donors. Due to the limited timeframe, we aim at conducting one interview per case. These interviews will be conducted using a remote chat service such as Skype.
- Documents: We will include documents created by the project such as blog posts, announcements, governance documents, bylaws and interviews of project members. Additionally, we will include external sources such as company announcements and reports from journals and newspapers.

All data is to be stored in a case study database in order to maintain a chain of evidence.

A.2.3. Data analysis

The interviews will be transcribed and then analyzed using grounded theory. Analysis will be carried out using MAXQDA. The resulting model will then be enhanced and strengthened using the documents.

A.2.4. Expected preparation prior to fieldwork

We will contact the relevant companies in order to find employees who (a) have been involved in spinning-out the project and (b) are still involved in the project.

A.3. Data Collection Questions

See interview protocols.

A.4. Guide for the Case Study Report

A.4.1. Audiences for the report and stylistic preferences

The report aims at a thesis committee and fellow researchers. Consequently, it has to maintain an academic style including sufficient references to related works and a clear chain of evidence.

A.4.2. Structure

The resulting report will have the following structure:

A.4.2.1. Introduction

Here we will highlight the relevance of open source foundations in a business context.

A.4.2.2. Research Question and Conceptual Model

An overview of the main aspects of the research question including a description of the underlying conceptual model.

A.4.2.3. Related Literature

A discussion of related works. While other comparative studies of open source cloud projects are mostly irrelevant to our research question, we will examine papers about open source foundations in this section. Moreover, we will explore the different motivations of companies and individual members of such foundations by citing relevant literature.

A.4.2.4. Methodology

An exact description of our research process, including our data sources.

A.4.2.5. Individual Cases: Optional

A short description of each case, including their former owners and relevant dates. The resulting theory is presented individually for each case. Major categories from the theory will form the individual headlines of the sections in this chapter.

Update: We skipped this step due to space restrictions.

A.4.2.6. Cross Case Analysis

Results from the single cases will be interpreted and compared. Extensive use of comparative tables and diagrams, addressing both facts and interpretations.

A.4.2.7. Discussion

Discussion whether and how the resulting theory can be generalized. We will address cloud-specific aspects in this section.

A.4.2.8. Limitations / Future Work

Threads to validity and potential for future work.

B. Case Study Database

Yin (2013) suggests the use of a case study database to increase the reliability of the case study. Consequently, we stored all data in electronic form while maintaining references to its original sources.

There were basically two separate storage systems:

- Factual data about the foundations (e.g. history, corporate involvement, entities) was stored in spreadsheets and text documents. A separate list of sources was maintained and every fact in these spreadsheets contains a reference to its respective source. Consequently, every statement can be traced back to the source, thus building a chain of evidence.

- Qualitative data such as the interview transcripts and blog posts was managed with the software MAXQDA. It was also used for analysis, thus allowing to link specific codes and categories back to the corresponding pieces of text.

Data from both systems was under a version control system (Subversion) with a commit taking place after every major change. Consequently, even the process of analysis can be revisited by browsing the repository logs.

The following sections show excerpts of the documents that are based on factual data. At the time of writing this thesis, some of these resources were no longer available and had to be accessed via `http://archive.org/web/`.

The most important documents are an extended classification of cases based on (Riehle & Berschneider, 2012), an overview of the histories of cases and a list showing company participation in the cases.

Due to the large amount of data, we cannot include the case database documents in this thesis, as even the case classification and the case histories would occupy more than 20 pages. Please refer to the accompanying CD or contact us.

B.1. Case Histories

Figure B.1 shows an excerpt of the case histories of Apache CloudStack and OpenStack. The second column for every case contains a list of sources for the respective statement.

Month	Day	Apache CloudStack		OpenStack	
5	4	**VMOps** renamed to **Cloud.com**, leaves stealth mode after $11m series B, release w. GPL v3	101, 102, 109, 110		
6	8			**White House** suggests cloud solutions to government agencies	47
7	13			**OpenStack Design summit** starts, including **IBM**, **Cloud.com** and **Citrix**	27
	19			**OpenStack launched** by **Rackspace** (CF => Swift) and **NASA** (NebulaNova => Nov	6, 26, 27, 30
	22			**Cloud.com** announces membership in OpenStack initiative	58
9				First beta release of Nebula	33
10	21			First major release: **Austin**	15, 43
2	3			Major release: **Bexar**	23
	9			**Rackspace** acquired **Anso Labs**, increases control over OpenStack	28, 29
3	3			Governance changed by **Rackspace**	1
	> 3			Project Lead/Founder **Rick Clark** leaves after criticizing unilateral governance change	2, 3, 4
4	12				
	15			Major release: **Cactus**	23
5	25			**Citrix** announces a commercial product based on OpenStack: "**Project Olympus**"	10, 54
6	15			**Cloud.com** CTO thinks about **merging** the CloudStack code base with OpenStack	53, 55
7	12	**Cloud.com** acquired by **Citrix**	101, 102, 108, 114		
8					
	17				
	25				
	29	First release after acquisition (2.2.10), **fully open source** (GPL v3)	101, 102, 115, 116		
	30				
9	22			Major release: **Diablo**	23
10	6			**Foundation announced** on OpenStack Conference 2011	6, 9
11	8				
12	15				
2	13	**Citrix** releases **CloudStack 3.0** with **OpenStack Swift** integration	102, 112, 119		
3	20				
4	3	Relicensed under ASL 2, **proposed to ASF**; **Citrix** becomes ASF Platinum Sponsor	101, 102, 106, 107, 117	"**Project Olympus** is dead.": OS too far behind CS in order to integrate both	11, 12, 13, 14, 185
	5			Major release: **Essex; Framework commitment deadline**	23
	11				
	12			Nineteen companies sign Framework Ack. Letter, including **IBM** and **Red Hat**	56
	16	Accepted in **ASF incubator**	101, 102	**Rackspace** announces first commercial product based on OpenStack	49, 50, 51
	30				
5	22			**NASA** announces plans to step back from OpenStack	31, 161
7	18			Governance documents finalized	130
8					
	13			**Red Hat** releases preview of its OpenStack distribution	24
9					
	19			**Foundation** launches with $10M in promised funding	8
	27			Major release: **Folsom**	23
11	6	First release **4.0.0-incubating**	101, 102, 103		

Figure B.1.: Excerpt of case histories from the case study database.

B.2. Corporate Involvement

Figure B.2 shows an excerpt of the company involvement in Apache CloudStack and OpenStack. The second column for every case contains a list of sources for the respective statement.

Company	High. Rank	#Proj.	Apache CloudStack Role	Source	OpenStack Role	Source
HP	1	4	**ASF gold sponsor**	157	**Founding platinum member, HP Cloud Services**	56, 57, 59
IBM	1	4	**ASF gold sponsor, Products (SoftLayer), helped found the ASF**	157, 186	**Founding platinum member**	56, 57
Pivotal	1	3	**ASF silver sponsor**	157		
VMware	1	3	Gold conference sponsor	120	**Gold member**	56, 57
AT&T	1	2			**Founding platinum member**	56, 57
Canonical	1	2			**Founding platinum member, project Solum**	22, 56, 57
Citrix	1	2	**IP owner, ASF platinum sponsor**	120, 157	Initiative Design Summit Attendee	27
Google	1	2	**ASF platinum sponsor**	157		
Rackspace	1	2			**IP owner, founding platinum member**	27, 56, 57
Red Hat	1	2			**Founding platinum member, project Solum**	22, 56, 57
SAP	1	3			Sponsor	57
SUSE	1	2			**Founding platinum member**	56, 57
Yahoo!	1	2	**ASF platinum sponsor**	157	**Founding gold member**	56, 57
Facebook	1	1	**ASF platinum sponsor**	157		
Matt Mullenweg	1	1	**ASF platinum sponsor**	157		
Microsoft	1	1	**ASF platinum sponsor**	157		
Nebula	1	1			**Founding platinum member**	56, 57
Ericsson	2	3			**Gold member**	56, 57
Intel	2	3			**Initiative Design Summit Attendee, gold member**	27, 56, 57
Cisco	2	2			**Founding gold member**	56, 57
Huawei	2	2	**ASF silver sponsor**	157	**Gold member**	56, 57
Juniper Networks	2	2	Initial support, platinum conference sponsor	118	**Gold member**	56, 57
NetApp	2	2	Initial support, gold conference sponsor	118, 120	**Founding gold member**	56, 57
NTT	2	2			Initiative Design Summit Attendee	27

Figure B.2.: Excerpt of company participation from the case study database.

C. Interview Questions

The following questions were used as a guideline for conducting the semi-structured interviews. Due to the exploratory nature of the research, however, we reacted more to what the interviewees said rather than strictly adhering to the interview protocol.

The order of the following sections represents the order in which the interviews actually took place. The only exception is Cloud Foundry since we could not find a suitable interview partner. We refined the questions after each interview in order to incorporate new insights.

See Figures C.1, C.2, C.3 and C.4 for the individual protocols.

1 Introduction

- When did you join?

2 Project interests / Foundation interests

- How do you ensure that everything happens in the **best interest of the project** / foundation?
  How do you protect the foundation from company interests?
- Where is the line? Is there a risk of forks?

3 Competitors

- *Lots of companies, some are **competitors** (IBM, HP, Oracle).*
- How does the foundation handle **potential conflicts of interest** between these companies?
  - Can you remember any **conflicts** or heated debates?
- **Tiered** membership -> are all interests considered equally? Smaller companies?

4 RackSpace / Past

- How did **RackSpace** make sure that its interests are going to be considered in the **future**?
- Did you anticipate the joining of **potential competitors**?
  - How would you deal with this?
- **Trade-off**: project interests vs. RackSpace's interests

5 Comparison Past – Present

- **Ongoing process** vs. initial design
- Lessons learned, expectations met?

6 Optional

- Joining vs. creating a foundation
- Risk of forks?

Figure C.1.: OpenStack interview protocol.

1 Introduction

- When did you join?

2 Questions for Citrix

2.1 Motivation of Citrix

- Why did Citrix donate its software to the ASF?
  - **Common enemy** (VMware) as a motivation?
- Why did it not create a **new foundation**?

2.2 Joining the ASF

- How did Citrix deal with the ASF and the Apache Way?
- What was the learning process like?
- Was there the possibility to be rejected by the ASF?
  - Are there companies that should not donate their projects to the ASF?

2.3 Interests

- How did **Citrix** make sure that its interests are going to be considered in the **future**?
- Did you anticipate the joining of **potential competitors**?
  - Who would you deal with this?
- Compatibility as a potential source of conflicts?
  - **VMWare hypervisors**, (AWS APIs)
- Did you have to make any major trade-offs? Which ones?

2.4 Comparison to OpenStack

- **Best kept secret** -> Apache Way makes marketing and sponsorship difficult
- Other disadvantages of being user driven rather than company driven?
  - What are the Advantages?
  - Opinion on OpenStack foundation?
- Is this a **concern** for the **ASF**, the CloudStack PMC or does it only affect Citrix?

3 Project-level Questions

3.1 Project interests / Foundation interests

- How do you ensure that everything happens in the **best interest** of the project / foundation?
- How do you protect the foundation from company interests? "Individual rule" enough?
- What happens if the interests of Citrix and the ASF / PMC were different?

3.2 Common goals

- How do you make sure that all committers / PMC members share a **common goal**?
- How do you make sure that the interests of participants are aligned with those of the project?

3.3 Affiliations & Takeover

- ***Foundation of individuals***, *based on meritocracy and votes, people wear multiple hats, independent of their corporate affiliation*
- ***Corporate interests***: paid contributors (mailing list), corporate CLAs, "independent committers"
- Do you see this as a source of potential conflicts?
- *High percentage of Citrix employees; graphs by sebgoa*
  - Potential **takeover**?
  - How **dependent** is the project on Citrix?

3.4 Controversies & Conflicts

- How do you handle conflicts or controversies between committers / PMC members?
- Do you remember any conflicts? Issues with heated debates / lots of controversy?
- Is trademark usage a problem?

3.5 Communication

- What is the primary conflict resolution medium => mailing lists?
- How important are F2F meetings?

4 Expectations / summary

4.1 Project

- Is the system working?
- Has the governance changed over time?
- Has the PMC / committer base changed, i.e. lots of rotation?

4.2 Citrix

- Expectations met? Lessons learned?

Figure C.2.: Apache CloudStack interview protocol.

1 Introduction

- *IBM / Eclipse*:
- When did you join?
- How did you get involved with Eclipse?

2 Questions about IBM

2.1 Motivation of IBM

- Why did IBM want to **create** a new foundation?
  - Motivated by competition?
- Why did IBM **not join** an existing one (**ASF**)?
- Did IBM look at the structure and governance of **existing foundations**?
- What were the **expectations** of IBM?

2.2 Interests

- How did **IBM** make sure that its interests were going to be considered in the **future**?
- Did IBM anticipate the joining of **potential competitors**, e.g. HP?
  - Were there any problems related to this?
  - How did IBM handle this scenario?

3 Foundation-specific Questions

3.1 Project interests / Foundation interests

- How does the foundation make sure that everything happens in the **best interest** of the project / foundation?
- How does the foundation **protect** itself from company interests?

3.2 Governance & Bodies

- *Board, PMCs*
- **Why** did IBM / the foundation go with this approach?
- Is there a **separation of powers** regarding technical and business issues?

3.3 Tiered Structures

- *Tiers: silver, gold and platinum sponsors; several member classes*
- Why are there currently **no platinum sponsors**?
- How does the foundation ensure an **equal representation of companies**, especially of smaller ones?

4 Members

4.1 Effects of being company-driven

- *ASF = user driven, Eclipse and OpenStack = company-driven*
- Why is the foundation not "individual-driven" like the ASF?
- Is being **company-driven** better? Why?
- *OpenStack = "shark tank"*
- Did IBM think about potential **downsides** of this approach?

4.2 Selection of Members / Common Goal

- How does the foundation make sure that all (board) members share a **common goal**?

- How does the foundation ensure that the interests of **members** are **aligned** with those of the **project**?

4.3 Affiliation & Takeover

- ***Bylaws**: Affiliation is considered for board representation of committer members*
- Is **affiliation** generally a topic inside Eclipse?
- Are there other rules / considerations in regards to affiliation?
- Is this enough to prevent a potential **takeover** by a dominant company?
- How **dependent** is the project on **IBM** or any other corporate member?

5 Conflict Resolution

5.1 Controversies & Conflicts

- How does the foundation handle **conflicts** or **controversies** between members?
- Do you remember any **controversial decisions / issues**?
  - Is **trademark usage** a problem?

5.2 Communication

- What is the primary conflict **resolution** medium - mailing lists?
- How important are **F2F** meetings?

6 Comparison Past – Present

6.1 Engagement of IBM

- *From gold to silver sponsor (09/2012), share of IBM committers declining*
- What is the **future of IBM** in the Eclipse foundation?
- Why has IBM's **engagement decreased** over time?

6.2 Changes

- What has **changed** over time?
  - Governance, members, rules
- **Lessons learned, expectations met?**

Figure C.3.: Eclipse interview protocol.

1 Introduction

- Pivotal / Cloud Foundry: When did you join? How did you get involved with Cloud Foundry?

2 Questions for Pivotal

2.1 Motivation of Pivotal

- Why **create** a foundation?
  - **Common enemy** (Proprietary players) as a motivation?
- Why **not join** an existing one?
- Did you look at **existing foundations**?
  - Eclipse-inspired model

2.2 Interests

- How does **Pivotal** make sure that its interests are going to be considered in the **future**?
- Do you anticipate the joining of **potential competitors**?
- How will you handle this scenario?
- Do you expect any major **trade-offs**?

3 Foundation-specific Questions

3.1 Project interests / Foundation interests

- How will you ensure that everything happens in the **best interest** of the project / foundation?
- How will you **protect** the foundation from company interests?
- What will happen if the interests of Pivotal and the foundation will be different?

3.2 Governance & Bodies

- *Board, Technical Committee, CAB + PMCs*
- **Why** did you go with this approach?
- Why **separation** of powers?
- Future role of the **CAB**?
- ***Tiered membership**: platinum and gold members*
- **Equal representation**, even of smaller companies?

4 Members

4.1 Effects of being company-driven

- *ASF = user driven, OpenStack = company-driven*
- Why is being **company-driven** better?
- *OpenStack = shark tank*
- Did you think about potential **downsides**?

4.2 Selection of Members / Common Goal

- How will you make sure that all committers / board members share a **common goal**?
- How will you make sure that the interests of participants are aligned with those of the project?
- Will there be a **screening process** for new members?

4.3 Affiliation & Takeover

- Are there specific rules for dealing with **affiliation**?
  - OpenStack has limits, CloudStack focuses on individuals
- Do you track **affiliations** / influence of specific companies?
- Potential **takeover**?
- How **dependent** is the project on **Pivotal** or any other single vendor?

4.4 Potential Conflicts

- **Competitors**
  - Different competitors: **IBM** and **HP** – is this going to be a problem?
- **RackSpace**
  - *mirantis.com/blog/openstack-havanas-stern-warning-open-source-or-die/*
  - ***RackSpace** is a sponsor, but OpenStack moves up in the stack (Solum)*
  - Do you see a conflict here?
- **Compatibility**
  - *Discussion about hypervisors, AWS API compatibility*
  - **Compatibility** as a potential source of conflicts?
- **Trademarks**
  - Is **trademark usage** a problem?

5 Conflict Resolution

5.1 Controversies & Conflicts

- How will you handle **conflicts** or controversies between committers / PMC / CAB members?
- Do you remember any **conflicts** from the CAB? Issues with heated debates / lots of controversy?

5.2 Communication

- What is the primary conflict **resolution** medium => mailing lists?
- How important are **F2F** meetings?

6 Data from Blogs / CAB

- **Platinum sponsors**
  - EMC, IBM, HP, Pivotal, **Rackspace**, SAP and VMware
  - 1.5M over three years
- **Gold sponsors**
  - ActiveState and CenturyLink
  - 250k annually

Figure C.4.: Cloud Foundry interview protocol.

D. Biography

As stated in the "Allgemeine Prüfungsordnung für die Bachelor- und Masterstudiengänge an der Technischen Fakultät der Friedrich-Alexander-Universität Erlangen-Nürnberg (ABMPO/TechFak)", §32 (6), a short biography is required:

Florian Weikert is currently pursuing his Master's Degree in Computer Science at the Friedrich-Alexander-University of Erlangen-Nuremberg. He has been working as a software developer since his teenage years, including more than four years as a student employee at the Fraunhofer Institute for Integrated Circuits (IIS) and the prestigious Extreme Blue Internship at IBM Germany.

His research interests focus on commercial open source software. Together with Dirk Riehle he has authored a conference paper called "A Model of Commercial Open Source Software Product Features" that was presented at the 4th International Conference on Software Business (ICSOB 2013) in Potsdam, Germany.

Florian holds a B.Sc. in Computer Science from the Friedrich-Alexander-University of Erlangen-Nuremberg.

References

Allan, G. (2003). A critique of using grounded theory as a research method. *Electronic Journal of Business Research Methods*, *2*, 1–77. Retrieved from http://www.ejbrm.com/issue/download.html?idIssue=16

Arneson, R. (2008). *Equality of Opportunity* (Fall 2008 ed.). Edward N. Zalta. Retrieved from http://plato.stanford.edu/archives/fall2008/entries/equal-opportunity/

Arshad, Y., & Ahlan, A. (2013). Combining Grounded Theory and Case Study Methods in IT Outsourcing Study. *Journal of Information Systems Research and Innovation*. Retrieved from http://eprints2.utem.edu.my/11331/1/Pub10_GroundedTheoryCaseStudy.pdf

Bell, D. (1971, November). On Equality: I. Meritocracy and Equality. *Public Interest*. Retrieved from http://eric.ed.gov/?id=EJ065658

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, *11*(3), 369–386. Retrieved from http://www.jstor.org/stable/248684 doi: 10.2307/248684

Black Duck Software. (2014). *2014 Future of Open Source Survey Results.* Retrieved 19.07.2014, from http://www.slideshare.net/blackducksoftware/2014-future-of-open-source-survey-results

Brandenburger, A. M., & Nalebuff, B. J. (1996). *Co-Opetition: A Revolution Mindset That Combines Competition and Cooperation: The Game Theory Strategy That's Changing the Game of Business*. Doubleday/Currency.

Brereton, P., & Kitchenham, B. (2008). Using a protocol template for case study planning. *Proceedings of the 12th International Conference on Evaluation & Assessment in Software Engineering (EASE)*(2006), 8. Retrieved from http://www.researchgate.net/publication/228573079_Using_a_protocol_template_for_case_study_planning/file/9fcfd50b722f5e2dcb.pdf

Calloway, L. J., & Knapp, C. A. (1992). Using Grounded Theory to Interpret Interviews. *Information Systems Journal*, *2008*(212), 1–2. Retrieved from http://csis.pace.edu/~knapp/AIS95.htm

Capiluppi, A., Stol, K.-J., & Boldyreff, C. (2012). Exploring the Role of Commercial Stakeholders in Open Source Software Evolution. In *Ifip advances in information and communication technology 378 (oss 2012)* (Vol. 378, pp. 178–200). IFIP AICT.

Charmaz, K. (1996). The Search for Meanings- Grounded Theory. In J. Smith, R. Harré, & L. van Langenhove (Eds.), *Rethinking methods in psychology* (pp. 27–49). doi: 10.1016/B978-0-08-044894-7.01581-5

Charmaz, K. (2008). Grounded Theory as an Emergent Method. In P. Leavy & S. Nagy Hasse-Biber (Eds.), *Handbook of emergent methods* (pp. 81–110). Gilford Press. Retrieved from `http://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2008-03814-007&site=ehost-live`

Collier, M. (2012). *OpenStack Foundation Update.* Retrieved 31.07.2014, from `http://www.openstack.org/blog/2012/04/openstack-foundation-update/`

Corbin, J. M., & Strauss, A. L. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, *13*(1), 3–21.

Creswell, J. W. (1998). *Qualitative inquiry and research design: Choosing among five approaches* (1st ed.). Thousand Oaks, CA: Sage Publications.

Crowe, S., Cresswell, K., Robertson, A., Huby, G., Avery, A., & Sheikh, A. (2011). The case study approach. *BMC medical research methodology*, *11*, 100.

Dagnino, G. B., & Padula, G. (2002). Coopetition strategy. Towards a new kind of interfirm dynamics? In *Euram. the european academy of management. second annual conference -"innovative research in management"* (p. 32).

Dahlander, L., & Magnusson, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy*, *34*(4), 481–493. doi: 10.1016/j.respol.2005.02.003

De Laat, P. B. (2007). Governance of open source software: State of the art. *Journal of Management & Governance*, *11*(2), 165–177. doi: 10.1007/s10997-007-9022-9

Deshpande, A., & Riehle, D. (2008). The Total Growth of Open Source. In B. Russo, E. Damiani, S. Hissam, B. Lundell, & G. Succi (Eds.), *Proceedings of the fourth conference on open source systems* (Vol. 275, pp. 197–209). Springer Boston. Retrieved from `http://dx.doi.org/10.1007/978-0-387-09684-1_16` doi: 10.1007/978-0-387-09684-1_16

Doz, Y. L., Olk, P. M., & Smith Ring, P. (2000). Formation Processes of R&D Consortia: Which Path to Take? Where Does it Lead? *Strategic Management Journal*, *21*(3), 239–266. doi: 10.1002/(SICI)1097-0266(200003)21:3<239::AID-SMJ97>3.0.CO;2-K

Dunne, C. (2011). The place of the literature review in grounded theory research. *International Journal of Social Research Methodology*, *14*(2), 111–124. doi: 10.1080/13645579.2010.494930

Egan, T. M. (2002). Grounded Theory Research and Theory Building. *Advances in Developing Human Resources*, *4*(3), 277–295. doi: 10.1177/15222302004003004

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Academy of Management Review*, *14*(4), 532–550. doi: 10.5465/AMR.1989.4308385

Eisenhardt, K. M., Graebner, M. E., Huberman, A. M., & Miles, M. B. (2007). Theory Building

from Cases: Opportunities and Challenges. *Academy of Management Journal*, *50*(1), 25–32. Retrieved from http://aom.pace.edu/AMJ/editorials/Eisenhart.Graebner.2007.pdf doi: 10.5465/AMJ.2007.24160888

Elliott, M. S. (2003). The Virtual Organizational Culture of a Free Software Development Community. *Group*, *25*, 45–49. Retrieved from http://opensource.ucc.ie/icse2003/

Elliott, M. S., & Scacchi, W. (2002). Communicating and mitigating conflict in Open Source software development projects. *Projects & Profits*. Retrieved from http://www.ics.uci.edu/~melliott/commossd.pdf

Elliott, M. S., & Scacchi, W. (2003). Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration. In *Group '03 proceedings of the 2003 international acm siggroup conference on supporting group work* (pp. 21–30). Retrieved from http://dl.acm.org/citation.cfm?id=958164 doi: 10.1145/958160.958164

Elliott, M. S., & Scacchi, W. (2004). Free Software Development: Cooperation and Conflict in a Virtual Organizational Culture. In S. Koch (Ed.), *Free/open source software development* (pp. 152–173). Idea Group Publishing. Retrieved from http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.3952 doi: 10.4018/978-1-59140-369-2.ch007

Esteves, J., Ramos, I., & Carvalho, J. (2002). Use of grounded theory in information systems area: An exploratory analysis. In *European conference on research methods (ecrm), reading (uk)*. Retrieved from http://repositorium.sdum.uminho.pt/handle/1822/346

Fernández, W. D. (2004a). The grounded theory method and case study data in IS research : issues and design. *Information Systems Foundations: Constructing and Criticising Workshop at The Australian National University*(July 16-17), 43–59.

Fernández, W. D. (2004b). Using the Glaserian approach in grounded studies of emerging business practices. *Electronic Journal of Business Research Methods*, *2*, 83–94.

Fernández, W. D., Lehmann, H., & Underwood, A. (2002). Rigour and relevance in studies of IS innovation: A grounded theory methodology approach. *European Conference on Information Systems*, 110–119.

Finger, R. (2014). *The Trend To "Open Source" Software And What It Means For Businesses And Consumers.* Retrieved 16.07.2014, from http://www.forbes.com/sites/richardfinger/2014/02/04/the-trend-to-open-source-software-and-what-it-means-for-businesses-and-consumers/

Fitzgerald, B. (2006). The Transformation of Open Source Software. *MIS Quarterly*, *30*(3), 587–598. Retrieved from http://ezproxy.library.capella.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=21940322&site=ehost-live&scope=site doi: 10.1.1.107.2529

Freeman, S., & Siltala, J. (2004). *Freedom and Profit: how suits and hackers are working it out on the desktop.*

Gallivan, M. J. (2001). Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies. *Information Systems Journal*, *11*(4), 277–304. doi: 10.1046/j.1365-2575.2001.00108.x

Garg, S., & Eisenhardt, K. M. (2012). *Building Theory From Multiple Case Studies.* Retrieved from `http://www.iacmr.org/conferences/conf2012/PDW/SamGarg_Eisenhardt_PDW_IACMR_12.pdf`

Gilbert, C. G. (2005). Unbundling the Structure of Inertia: Resource Versus Routine Rigidity. *Academy of Management Journal*, *48*(5), 741–763. doi: 10.5465/AMJ.2005.18803920

Glaser, B. G., & Strauss, A. L. (2009). *The Discovery of Grounded Theory: Strategies for Qualitative Research.* Transaction Publishers.

Gonzalez-Barahona, J., Izquierdo-Cortazar, D., Maffulli, S., & Robles, G. (2013, September). Understanding How Companies Interact with Free Software Communities. *IEEE Software*, *30*(5), 38–45. Retrieved from `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6560081` doi: 10.1109/MS.2013.95

Gonzalez-Barahona, J., & Robles, G. (2013). Trends in Free, Libre, Open Source Software Communities: From Volunteers to Companies/Aktuelle Trends in Free-, Libre-, und Open-Source-Software-. *it-Information Technology*.

Gonzalez-Barahona, J., Robles, G., & Herraiz, I. (2007). Impact of the Creation of the Mozilla Foundation in the Activity of Developers. *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)*. doi: 10.1109/MSR.2007.15

Halaweh, M., Fidler, C., & McRobb, S. (2008). Integrating the Grounded Theory Method and Case Study Research Methodology Within IS Research: A Possible Road Map. In *International conference on information systems* (pp. 165–178).

Hallberg, L. (2010). Some thoughts about the literature review in grounded theory studies. *International Journal of Qualitative Studies on Health and Well-being*. Retrieved from `http://journals.co-action.net/index.php/qhw/article/viewArticle/5387/5828`

Heath, H., & Cowley, S. (2004). Developing a grounded theory approach: A comparison of Glaser and Strauss. *International Journal of Nursing Studies*, *41*(2), 141–150.

Hecker, F. (1999). Setting Up Shop: The Business of Open-Source Software. *IEEE Software*, *16*(1), 45–51. Retrieved from `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=744568` doi: 10.1109/52.744568

Heikkilä, M., & Kuivaniemi, L. (2012). Ecosystem Under Construction: An Action Research Study on Entrepreneurship in a Business Ecosystem. *Technology Innovation Management Review*. Retrieved from `http://www.timreview.ca/article/564`

Höst, M., & Runeson, P. (2007). Checklists for Software Engineering Case Study Research. In *Empirical software engineering and measurement, 2007. esem 2007. first international symposium on empirical software engineering and measurement* (pp. 479–481). doi: 10

.1109/ESEM.2007.46

Hunter, P., & Walli, S. (2013). The Rise and Evolution of the Open Source Software Foundation. *International Free and Open Source Software Law Review*, *5*(1), 31–42. Retrieved from `http://www.ifosslr.org/ifosslr/article/view/64`

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and Trust in Global Virtual Teams. *Organization Science*, *10*(6), 791–815. doi: 10.1287/orsc.10.6.791

Jensen, C., & Scacchi, W. (2005). Collaboration, Leadership, Control, and Conflict Negotiation and the Netbeans.org Open Source Software Development Community. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. doi: 10.1109/HICSS.2005.147

Krishnamurthy, S. (2003). An Analysis of Open Source Business Models. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Making sense of the bazaar: Perspectives on open source and free software* (Vol. 54, pp. 267–278). The MIT Press. Retrieved from `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=650001`

Lattemann, C., & Stieglitz, S. (2005). Framework for Governance in Open Source Communities. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. doi: 10.1109/HICSS.2005.278

Lauckner, H., Paterson, M., & Krupa, T. (2012). Using Constructivist Case Study Methodology to Understand Community Development Processes : Proposed Methodological Questions to Guide the Research Process. *The Qualitative Report*, *17*, 1–22.

Letellier, F. (2008). Open source software: the role of nonprofits in federating business and innovation ecosystems. In *Association for financial markets in europe (2008)*.

Lingard, L., Albert, M., & Levinson, W. (2008). Grounded theory, mixed methods, and action research. *British Medical Journal*, *337*, 459–461. Retrieved from `http://www.health.heacademy.ac.uk/doc/resources/groundedtheory-aug2008.pdf/at_download/file.pdf`

Loebecke, C., Van Fenema, P. C., & Powell, P. (1999). Co-opetition and knowledge transfer. *ACM SIGMIS Database*, *30*(2), 14–25. doi: 10.1145/383371.383373

Lomas, N. (2014). *Android Still Growing Market Share By Winning First Time Smartphone Users.* Retrieved 08.08.2014, from `http://techcrunch.com/2014/05/06/android-still-growing-market-share-by-winning-first-time-smartphone-users/`

Lynch, D. (2014). *Google Chrome Surpasses Internet Explorer In Combined Browser Market Shares For First Time.* Retrieved 08.08.2014, from `http://www.ibtimes.com/google-chrome-surpasses-internet-explorer-combined-browser-market-shares-first-time-1595187`

McGhee, G., Marland, G. R., & Atkinson, J. (2007). Grounded theory research: Literature reviewing and reflexivity. *Journal of Advanced Nursing*, *60*(3), 334–342.

Meyer, C. B. (2001). A Case in Case Study Methodology. *Field Methods*, *13*(4), 329–352.

doi: 10.1177/1525822X0101300402

Milinkovich, M. (2008). *A Practioners Guide to Ecosystem Development.* Retrieved 15.08.2014, from http://www.eclipse.org/community/training/webinars/081015_Ecosystems_Webinar.pdf

Mills, J., Bonner, A., & Francis, K. (2006). The Development of Constructivist Grounded Theory. *International Journal of Qualitative Methods*, *5*(1), 25–35. doi: 10.2307/588533

O'Mahony, S. (2003). Guarding the commons: how community managed software projects protect their work. *Research Policy*, *32*(7), 1179–1198. doi: 10.1016/S0048-7333(03)00048-9

O'Mahony, S. (2005). Nonprofit Foundations and Their Role in Community-Firm Software Collaboration. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 393–413). Cambridge, Massachusetts: The MIT Press.

O'Mahony, S., & Bechky, B. A. (2008). Boundary Organizations: Enabling Collaboration among Unexpected Allies. *Administrative Science Quarterly*, *53*(3), 422–459.

O'Mahony, S., & Ferraro, F. (2004). Managing the Boundary of an 'Open' Project. *Market Emergence and Transformation*(03), 1–50. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=474782 doi: 10.2139/ssrn.474782

O'Mahony, S., & West, J. (2005). What makes a project open source? Migrating from organic to synthetic communities. *Academy of Management conference, Technology and Innovation Management division, Honolulu, August 2005*, 39.

Pandit, N. R. (1996). The Creation of Theory: A Recent Application of the Grounded Theory Method. *The Qualitative Report*, *2*(4), http://www.nova.edu/ssss/QR/QR2–4/pandit.html. Retrieved from http://www.nova.edu/ssss/QR/QR2-4/pandit.html/pandit.html doi: 10.1186/gb-2006-7-9-r80

Pearce, J. (2014). *9.9 million lines of code and still moving fast - Facebook open source in 2014.* Retrieved 12.08.2014, from https://code.facebook.com/posts/292625127566143/9-9-million-lines-of-code-and-still-moving-fast-facebook-open-source-in-2014/

Pingdom. (2009). *The 8 most successful open source products ever.* Retrieved 08.08.2014, from http://royal.pingdom.com/2009/05/29/the-8-most-successful-open-source-products-ever/

Prattico, L. (2012). Governance of Open Source Software Foundations: Who Holds the Power? *Technology Innovation Management Review*. Retrieved from http://www.timreview.ca/article/636

Raymond, E. S. (1998). Homesteading the Noosphere. *First Monday*, *3*(10), 1–35. Retrieved from http://www.tuxedo.org/~esr/wri

Riehle, D. (2007). The Economic Motivation of Open Source Software: Stakeholder Perspec-

tives. *Computer*, *40*(4), 25–32. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4160218 doi: 10.1109/MC.2007.147

Riehle, D. (2009). The Commercial Open Source Business Model. In *Proceedings of the fifteenth americas conference on information systems* (Vol. AMCIS 2009, pp. 1–10). Retrieved from http://aisel.aisnet.org/amcis2009/104

Riehle, D. (2010). The Economic Case for Open Source Foundations. *Computer*, *43*(1), 86–90. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5398790 doi: 10.1007/s10257-010-0149-x

Riehle, D. (2011). Controlling and Steering Open Source Projects. *IEEE Computer Society*(July).

Riehle, D. (2012). The single-vendor commercial open course business model. *Information Systems and e-Business Management*, *10*(1), 5–17.

Riehle, D., & Berschneider, S. (2012). A model of open source developer foundations. In *The 8th international conference on open source systems (oss '12)* (pp. 7–28). Springer. doi: 10.1007/978-3-642-33442-9_2

Robles, G., Dueas, S., & Gonzalez-Barahona, J. (2010). Corporate Involvement of Libre Software: Study of Presence in Debian Code over Time. *IFIP International Federation for Information Processing*. Retrieved from http://dl.ifip.org/index.php/ifip/article/view/11326

Rodon, J., & Pastor, J. A. (2007). Applying Grounded Theory to Study the Implementation of an Inter-Organizational Information System. *Journal of Business Research*, *5*(2), 71 – 82. Retrieved from http://ezproxy.library.capella.edu/login?url=http://search.ebscohost.com.library.capella.edu/login.aspx?direct=true&db=bth&AN=31566737&site=ehost-live&scope=site

Rooney, P. (2012). *Five out of six developers now using or deploying open source.* Retrieved 16.07.2014, from http://www.zdnet.com/five-out-of-six-developers-now-using-or-deploying-open-source-7000008499/

Runeson, P., & Höst, M. (2008). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164. Retrieved from http://www.springerlink.com/index/10.1007/s10664-008-9102-8 doi: 10.1007/s10664-008-9102-8

Shakir, M. (2002). The selection of case studies: strategies and their applications to IS implementation case studies. *Research Letters in Information and Mathematical Science*, *3*, 191–198. Retrieved from http://muir.massey.ac.nz/handle/10179/4373

Sharma, S., Sugumaran, V., & Rajagopalan, B. (2002). A framework for creating hybrid-open source software communities. *Information Systems Journal*, *12*(1), 7–25. doi: 10.1046/j.1365-2575.2002.00116.x

Shimel, A. (2012). *10 most successful open source projects of 2012.* Retrieved

08.08.2014, from http://www.itworld.com/software/331041/10-most-successful-open-source-projects-2012

Siltala, J., Freeman, S., & Miettinen, R. (2007). *Exploring the tensions between volunteers and firms in hybrid projects.* Retrieved from http://www.edu.helsinki.fi/activity/publications/files/343/Exploring.pdf

Skerrett, I. (2009). Collaborative Software Development in the Enterprise. *Open Source Business Resource.* Retrieved from http://www.timreview.ca/article/219

Skerrett, I. (2011). Best Practices in Multi-Vendor Open Source Communities. *Open Source Business Resource.* Retrieved from http://www.timreview.ca/article/409

Steenhuis, H., & Bruijn, E. (2006). Building theories from case study research: the progressive case study. POMS: Production and Operations Management Society. Retrieved from http://doc.utwente.nl/73615/1/building.pdf

Stewart, K. J., & Gosain, S. (2006). The impact of ideology on effectiveness in open source software development teams. *Mis Quarterly*, *30*(2), 291–314.

Stoecker, R. (1991). Evaluating and rethinking the case study. *The Sociological Review*, *39*(1), 88–112. Retrieved from http://onlinelibrary.wiley.com/doi/10.1111/j.1467-954X.1991.tb02970.x/abstract doi: 10.1111/j.1467-954X.1991.tb02970.x

Strauss, A. L., & Corbin, J. M. (2008). *Basics of Qualitative Research* (3rd ed., Vol. 3). SAGE Publications, Inc. doi: 10.4135/9781452230153

Strauss, D. A. (1992). The Illusory Distinction between Equality of Opportunity and Equality of Result, The. *Wm. & Mary L. Rev.*, *34*(1), 171–188. Retrieved from http://scholarship.law.wm.edu/wmlr/vol34/iss1/10/

Strübing, J. (2007). Glaser vs. Strauss? Zur methodologischen und methodischen Substanz einer Unterscheidung zweier Varianten von Grounded Theory. *Historical Social Research*, *19*, 157–174. Retrieved from http://www.jstor.org/stable/40981075

Urquhart, C. (2000). An encounter with grounded theory: tackling the practical and philosophical issues. In E. Trauth (Ed.), *Qualitative research in is: Issues and trends* (pp. 104–140). Hershey, PA, USA: Idea Group Publishing.

Urquhart, C., Lehmann, H., & Myers, M. D. (2009). Putting the 'theory' back into grounded theory: guidelines for grounded theory studies in information systems. *Information Systems Journal*, *20*(4), 357–381. Retrieved from http://blackwell-synergy.com/doi/abs/10.1111/j.1365-2575.2009.00328.x doi: 10.1111/j.1365-2575.2009.00328.x

US IRS. (2014a). *Business Leagues.* Retrieved 16.07.2014, from http://www.irs.gov/Charities-&-Non-Profits/Other-Non-Profits/Business-Leagues

US IRS. (2014b). *Exemption Requirements - 501(c)(3) Organizations.* Retrieved 16.07.2014, from http://www.irs.gov/Charities-&-Non-Profits/Charitable-Organizations/Exemption-Requirements-Section-501(c)(3)-Organizations

Van Wendel de Joode, R. (2004). Managing Conflicts in Open Source Communities. *Electronic Markets*, 104–113. Retrieved from http://dx.doi.org/10.1080/10196780410001675059 doi: 10.1080/10196780410001675059

von Hippel, E. (2001). Innovation by User Communities: Learning from Open-Source Software. *MIT Sloan Management Review*, *42*(4), 82–86. Retrieved from http://search.ebscohost.com.ezp.essec.fr/login.aspx?direct=true&AuthType=ip,url&db=bth&AN=4834450&site=ehost-live doi: Article

von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, *32*(7), 1217–1241. doi: 10.1016/S0048-7333(03)00050-7

W3Techs. (2014). *Usage statistics and market share of Apache for websites.* Retrieved 08.08.2014, from http://w3techs.com/technologies/details/ws-apache/all/all

Wagstrom, P., Herbsleb, J., Kraut, R., & Mockus, A. (2010). The impact of commercial organizations on volunteer participation in an online community. In *2010 academy of management meeting.* Montreal, Canada. Retrieved from http://herbsleb.org/web-pubs/pdfs/wagstrom-impact-2010.pdf

Wasserman, A. I. (2013). Community and Commercial Strategies in Open Source Software. *it - Information Technology*. Retrieved from http://www.degruyter.com/view/j/itit.2013.55.issue-5/itit-2013-1003/itit-2013-1003.xml

Watson, R. T., Boudreau, M.-C., York, P. T., Greiner, M. E., & Wynn, D. (2008). The business of open source. *Communications of the ACM*, *51*(4), 41–46. Retrieved from http://portal.acm.org/citation.cfm?doid=1330311.1330321 doi: 10.1145/1330311.1330321

Weiss, M. (2011). Control and diversity in company-led open source projects. *Open Source Business Resource*. Retrieved from http://www.timreview.ca/article/436

West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. *Research Policy*, *32*(7), 1259–1285. doi: 10.1016/S0048-7333(03)00052-0

West, J., & Gallagher, S. (2004). *Key Challenges of Open Innovation: Lessons from Open Source Software.* Retrieved from http://www.cob.sjsu.edu/west_j/Papers/WestGallagher2004.pdf

West, J., & Gallagher, S. (2006a). Challenges of open innovation: the paradox of firm investment in open-source software. *R&D Management*, *36*(3), 319–331. Retrieved from http://doi.wiley.com/10.1111/j.1467-9310.2006.00436.x doi: 10.1111/j.1467-9310.2006.00436.x

West, J., & Gallagher, S. (2006b). *Patterns of Open Innovation in Open Source Software* (Vol. pp; H. Chesbrough, W. Vanhaverbeke, & J. West, Eds.) (No. October 2005). Oxford University Press. Retrieved from http://openinnovation.berkeley.edu/ranp_chapters/05.pdf

West, J., & O'Mahony, S. (2005). Contrasting Community Building in Sponsored and Community Founded Open Source Projects. In *Proceedings of the 38th annual hawaii international conference on system sciences.* doi: 10.1109/HICSS.2005.166

West, J., & O'Mahony, S. (2008). The Role of Participation Architecture in Growing Sponsored Open Source Communities. *Industry & Innovation*, *15*(2), 145–168. doi: 10.1080/13662710801970142

Whelan, P. (2014). *Cloud Foundry Advisory Board Meeting - 2014 February.* Retrieved 31.07.2014, from `http://www.activestate.com/blog/2014/02/cloud-foundry-advisory-board-meeting-2014-february`

Willig, C. (2013). *Introducing qualitative research in psychology Adventures in theory and method* (3rd ed., Vol. 2nd; C. Willig, Ed.) (No. 7). Open University Press. doi: 10.1177/1468794106058877

Xie, Z. (2008). Open Source Software Foundations. *Open Source Business Resource*. Retrieved from `http://timreview.ca/article/194`

Yin, R. K. (2011). *Applications of case study research* (3rd ed.). Thousand Oaks, California: SAGE Publications.

Yin, R. K. (2013). *Case Study Research: Design and Methods* (5th ed.). Thousand Oaks, California: SAGE Publications.

Zainal, Z. (2007). Case study as a research method. *Jurnal Kemanusiaan*, *9*(9), 1–6. Retrieved from `http://www.gslis.utexas.edu/~ssoy/usesusers/l391d1b.htm` doi: 10.1016/S0890-4065(96)90002-X