Colliberate

The practices of free and open source software

Reinhard Handler

Faculty of Arts and Social Sciences

Media and Communication Studies

DOCTORAL THESIS | Karlstad University Studies | 2022:15

Colliberate

The practices of free and open source software

Reinhard Handler

DOCTORAL THESIS | Karlstad University Studies | 2022:15

Colliberate - The practices of free and open source software

Reinhard Handler

DOCTORAL THESIS

Karlstad University Studies | 2022:15

urn:nbn:se:kau:diva-89585

ISSN 1403-8099

ISBN 978-91-7867-278-3 (print)

ISBN 978-91-7867-299-8 (pdf)

Distribution:
Karlstad University
Faculty of Arts and Social Sciences
Department of Geography, Media and Communication
SE-651 88 Karlstad, Sweden
+46 54 700 10 00

Print: Universitetstryckeriet, Karlstad 2022

WWW.KAU.SE

A file manifesto: Table of contents

Foreword

Writing this dissertation has been a long journey, longer than anticipated. I expected it to be less rocky and less demanding, on myself and on others. My sincere apologies to those who I disappointed and to those who I neglected. The last few years have not been the easiest. Nevertheless, this project is finished. Many people have contributed to it so that a happy ending was possible. I will not list any names in this foreword so that I do not make the mistake of forgetting one of you, but you know how you are.

I would like to thank everyone who was willing to keep me company along the way. To those who helped me to get through this: I value your input and patience. Shout-out to everyone who reminded me (many times unsuccessfully) that being a doctoral student can also be fun. My respect and admiration to those who shared their knowledge and allowed me to learn. Special thanks to those who became friends. To the people closest to me: True love!

I cannot thank enough the people involved in LibreOffice and / or The Document Foundation. Your helpfulness and attention had no limits. Many of you have invested a lot of time to share your thoughts and give me support and explanations when I needed them.

Peace out!

0. READ.ME – Introduction

The Portuguese man o'war floats on the surface of the Atlantic Ocean. Despite its large habitat, little is known about this animal. This marine hydrozoan resembles a longish bubble or a bellied bottle that lies on the side. It looks like a jellyfish with its bluish shimmer and its raised tail with a pink border. Only under the water surface or when washed ashore it becomes apparent that there is more to the Portuguese man o' war compared to what can be seen above the water. The visible gas bladder, or pneumatophore, is only the sail. It is one part of a colony of smaller units. Each of the units has a specialised function. Under the bladder, which is between 9 and 30 centimetres long, there are tentacles that can be up to 50 meter long. Together with a feeding polyp, the tentacles hunt and feed the organism. A structure consisting of four different zoids, including a jelly polyp with an unclear function, is responsible for reproduction.

These so-called zoids are built exactly like individual organisms that do not live in colonies. However, in a Portuguese man o'war they develop from the same egg. They are connected to each other, and they are interdependent. The individual zoids are inviable, only in combination with the others can the different colonies of organisms perform their function. Depending on the perspective this animal is an individual with zoids performing as organs or it is made up of interconnected individuals which allow the colony to operate as a single animal. (Bardi & Marques, 2007; Munro et al., 2019)

The Portuguese man o'war is the spirit animal of this dissertation about free and open source software (f/oss[1]). Software's habitat is large. Wherever media are, there is software. More and more aspects of our lives are not only surrounded by media but have become increasingly dependent on media. As contemporary life is immersed in media, software has become the de facto infrastructure for everyday life. Yet in

1 There are several possible acronyms for free and open source software: Foss, f/oss, floss, or f/loss. To highlight that the term free software stands for freedom not for free of costs the word libre is sometimes added which results in the designation free and libre open source software. I have chosen to use f/oss. The slash marks the differences between free and open source software. I will discuss these differences in more detail in chapter 1.

contrast to its ubiquity, we still know very little about software. Infrastructure tends to be ignored, it resides in the background (Bowker & Star, 1999a). Media content floats on the surface. Under the surface, software has extended its reach to become the underbelly of contemporary culture. From the recommendation systems that guide our media consumption, to the computer models that calculate our travel routes, the tracing of goods and people in the industrial sector, the distribution systems for deliveries, and the financing systems that guide economic developments, software plays a central role. Software has taken command of media, as Manovich (2013) pointed out. The characteristic qualities of digital media are defined by software. Software has become a meta-medium and the consequence of this development goes beyond the reformatting of media content into digital files.

This dissertation wants to contribute to a richer understanding of software by looking under the hood of LibreOffice, a free and open source software office suite. Many do not know what f/oss is, nor what its characteristics are. Yet, many large web servers run on f/oss, the open source operating system Linux has become widely known and used, Android smartphones are based on free and open source software. F/oss is much more than a gratis alternative to established software products. It is free because it can be used, shared, and modified freely. It is an integral part of software's history and its present state.

Based on the freedom that f/oss offers, it allows people possibilities to collaborate without technical restrictions. It allows continuous adaptions and edits as well as mixing and sharing. It confronts traditional forms of ownership of media, and it offers alternative structures of governance. What seems to be an ordinary office productivity software suite turns out to be a sophisticated interplay of different individuals, groups, ideas, and practices that function together to constitute a f/oss project. The inner workings of this distributed collaborative project are similar the Portuguese man o'war. It relies on the interplay of several teams that are self-organised, yet they are interdependent. They share knowledge and efforts to engage in collaborative practices with which they produce software. Under the surface, I found a rich and complex set of practices for the collaborative production of software.

I have studied the configuration of LibreOffice for two years. I visited their self-organised conferences, followed them to large software conferences, talked to collaborators, conducted 37 interviews, read through protocols and e-mail archives, observed their discussions online, and was present when they argued. The empirical data gathered by this ethnographic approach offers insight into one of the largest and longstanding free and open source software projects.

This study explores software by zooming in on the collaborative practices. It shows how they are negotiated and formed along the diverse attitudes of collaborators towards free and open source software. It explains how a free and open source software project is maintained by people who engage in collaborative practices. It is also about the ideas that underlie the collaborative practices that find expression in making software. At the same time, it is about these practices' materiality and the role that software plays in this process. Thus, this study focuses on the sociotechnicality of collaborative practices, as they link people by linking ideas and materials.

Accordingly, this study's main problem is how collaborative practices emerge, how they are negotiated, and how they are ordered in the context of free and open source software. This concerns the matter of media practices. Associated with these questions are the multiple roles that software has. Software emerges as the result of collaborating whilst it is also collaborated on and collaborated with. This concerns the question of the mediality of software and the sociotechnical character of collaborative practices.

0.1. Free and open source software

The reason for choosing a free and open source software project for this study is straightforward. Collaboration is an essential component of free and open source software. Free and open-source software (or: f/oss) is a term that stands for non-proprietary software that is licensed to be used, modified and shared freely. Open source refers to the source code, the part of the software that is written in a human-readable programming language. An open source allows access to the source code, if it is closed, software cannot be (legally) modified. Software is free when it gives people the freedom to share code, to use it, to change it and possibly to improve other people's work. This is only possible if the

source code is open. People can share, work on and exchange their improvements, they can collaborate freely if the source code is open. However, free and open-source software is not a purely technical phenomenon; if such a thing is possible, it also involves a political and ethical dimension. The close association of technology and political ambitions that characterises free and open-source software becomes clear in the wording of the self-description that LibreOffice collaborators have chosen:

> We believe that users should have the freedom to run, copy, distribute, study, change and improve the software that we distribute. While we do offer no-cost downloads of the LibreOffice suite of programs, Free Software is first and foremost a matter of liberty, not price. We campaign for these freedoms because we believe that everyone deserves them. We seek to eliminate the digital divide and empower all as full citizens, support the preservation of mother tongues, and avoid proprietary software and format lock-in. We work to attain our goals by
>
> providing unfettered access to our office productivity tools at no cost
>
> encouraging the translation, documentation, and maintenance of our software in one's own language
>
> promoting and actively participating in the creation and development of open standards and Free Software via open and transparent peer-review processes. (The Document Foundation, 2021a)

For those not familiar with free and open source-software, the above statement appears to exceed the description of an office suite distinctly. Yet it reflects the language used habitually by speakers and discussants at free software conferences, and is reaffirmed in self-description such as the one above as well as in manifestos, mission statements and statutes of f/oss projects. It points toward issues that have arisen since the early days of the Internet: the concern for the digital divide, access to technology, the complication of property and the commons concerning knowledge, as well as participation, openness, and transparency.

> Though the members of our community hail from many different backgrounds, we all value personal choice and transparency, which translates practically into wider compatibility, more utility, and no end-user lock-in to a single product. We believe that Free Software can provide

> better quality, higher reliability, increased security, and greater flexibility than proprietary alternatives. The community behind LibreOffice is the heart of the project, without which we would not have the resources to continue developing our software. The passion and drive that every individual brings to the community results in collaborative development that often exceeds our own expectations. (The Document Foundation, 2021a)

The Document Foundation is the organisational home of LibreOffice. It is a self-governing, non-profit organisation. Its self-description above underlines how political-ethical values such as openness, transparency, and participation are not just aims to contribute to with a piece of software. These values are also directed inwards, towards the production of the office suite called LibreOffice as well as towards the governing mechanisms that are part of the project. They are articulated as an organisational frame for a community that has formed around the production of a software application.

This study focuses on the collaborative development mentioned in the above self-description. It considers free software not only as a technology but also as a political realm, as a discourse, as a community, and as an institution. Thus, the collaborative production of LibreOffice will not be analysed as a set of technical procedures. Rather, this study examines the practices of a group of people who form a free and open source software project together: how they coordinate, how they express (or hold back) their differences, why they hang out, how they build ordering mechanisms, and what rules and norms they follow. In sum, this study is about how a group of people gather to collaboratively produce a free software office suite.

0.2. Collaboration

Collaboration is at the centre of this study. It is one of the key features of the so-called cognitive(-cultural) capitalism (Moulier Boutang, 2011). The term assumes that capitalism has entered a new phase. The industrial era was characterised by the creation of surplus value through the production and traffic of material goods. A new phase has begun, which rests on value-creation based on immaterial goods. Sectors such as cultural industries, media industries, and high-technology industries have become more important and increasingly created syn-

ergies with each other, surrounded by a booming service sector. Production processes started to diverge from classic industrial forms of organisation by creating more flexible systems. Agile teams and project work – accompanied by freelance work and gigs instead of steady employment – are characteristic for these collaborative production models. My point here is not that self-organised collectives are new phenomena which have come into existence with digital media. The concept of worker cooperatives is well established, agricultural cooperatives are the backbone of family-owned farming, and housing cooperatives allow affordable shared ownership. What is of interest instead are the relations between collaborators in network-based forms of working together. Interest in decentralised organisational structures has developed on a broader in the 1970s when the industrial sector started to look for alternatives to central forms of organisation. Alternative forms of working together began to replace hierarchic regimes of production. In addition, several forms of managing and maintaining common resources have emerged in certain sectors and regions but they have not played a significant economic role considering the impetus on individual ownership that capitalism has thrived on so far.

In this context, collaboration is presented as an alternative to cooperation. Cooperation entails belonging, stability, locality, and collectivity. It is based on solidarity, trust, standardised practices, concerted planning, and common values. Collaboration in contrast is an expression of what Bauman framed as liquid modernity (Bauman, 2000). The sense of cooperation has weakened (Sennet, 2013). Instead of relations that are characterised by solidarity and group thinking, short-term connections emphasise individuality and connectivity. Collaborative connections are of an instantaneous character, they are characterised by weak ties (Granovetter, 1973). They soften the rigid institutions and solid human relations by replacing togetherness with fleeting yet manifold possibilities to connect and interact.

Flat hierarchies, decentralization and immanence have become the defining phenomena for a new spirit of capitalism (Boltanski & Chiapello, 2007). Instead of central authorities and bureaucracy, this organisational model consists of self-organisation, project work and cost-effective control mechanisms. These transversely organised networks are constructed on values like flexibility, mobility, and creativity (Boltanski

& Chiapello, 2007). Collaboration shares many features with a network sociality that Wittel (2001) has described as emphasizing individualisation and embedded in technology, with social relationships as social capital. These new relations are intense, limited in time, and project-based.

Collaboration in media studies has so far been analysed critically as the result of techno-economic relations that mostly focus on these liquid relations in terms of power relations and working conditions. After the expectations that were built up with Web 2.0 on promises of participation, openness, and flat hierarchies, much scholarly analysis has centred – after a phase of excitement – on the breaking of these promises. What was once hoped to become a catalyst for an arena of political and cultural participation has become a playground and a factory (Scholz, 2013). While early examples of the potential of distributed collaboration, such as Wikipedia, show the potential of tapping into the wisdom of the crowds, it has been turned into an economic spiel. Management and marketing rhetoric as well as cyber-libertarian thinking praise collaboration as a form of a free exchange of knowledge and mass participation in knowledge production. Supposedly, *mass collaboration changes everything* (Tapscott & Williams, 2010), that together we can create an *Infotopia* (Sunstein, 2008). The masses have been transformed, the slogan *Here comes everybody* (Shirky, 2009) does not terrify because *the many are smarter than the few* (Surowiecki, 2005) and *the power of the crowd is driving business* (Howe, 2009). A collaborative economy brings forward open processes that allow participation, thrives on interchange between people, and promotes alternative form of consumption. It promotes freedom and flexibility as a cultural value while technology companies' strategies to generate profit rest on user activity. In this 'lean platform economy' (Srnicek, 2017, p. 56) profit is not generated through the ownership of assets, but instead through the coordination of services. Collaboration occurs between users and the platform providers based on asymmetrical power relations. Digital platforms regulate the practices of the users, they control technical developments and make all economic decisions. Tools to analyse the activities of end-users and the resulting data becomes the key assets. The so-called platformisation has also shown how practices are

organised around platforms by governance mechanisms that minimize participation by users.

What can be observed is that many ideas that have become characteristic for the collaborative economy are based on innovation models that have become popular mainly through free and open source software. The general idea of many people sharing the creation, production, distribution and consumption of goods and services is central for f/oss; ordering practices of many collaborators is an indispensable necessity in distributed projects. Yet, free and open source software has also proven that digital commons can emerge out of a collaborative production model by deploying more inclusive ownership models in a socio-economic situation that is generally portrayed as open and participatory, as well as flexible and volatile. The key interest for this study is thus to explore how collaboration is realised in a free and open source software project and how it can be stabilised while being dynamic and open. This interest expands beyond an economic analysis of f/oss. Instead it focuses on the relations between collaborators in a free and open source software project: how their practices are ordered, how the outcome of their collaboration is stabilised, and the linkage between collaborative relations and governance mechanisms.

0.3. Contributions

As mentioned earlier, this an ethnographic study of a f/oss project. By having followed the collaborators and their efforts to maintain LibreOffice, I have gathered empirical data that offer an in-depth discussion at how a f/oss project is maintained. The empirical material offers a contribution to understanding media by highlighting that software is not neutral. Rather, it shows how software is constantly negotiated amongst the collaborators. It highlights the differences between the people involved as well as the discussions and mechanisms that revolve around the practices that are used to produce software.

On the basis of the empirical data, I will make two theoretical contributions. First, I explore the synergy of culture and software by highlighting how a specific cultural milieu forms with software at the same time as cultural forces shape practices of software development. And second, my observations of the collaborators' practices, how they are ordered and negotiated, together with the collaborators' reflections on

their practices through the interviews provide a comprehensive study of their practices. This adds to the discussion of the potential of practices as a concept to explore digital media.

0.3.1. Practices

Along with scholarly analysis of digital media, the notion of media practices has also received renewed attention. Media studies have started to make good use of practice theory from anthropology and STS (science and technology studies) (Bräuchler & Postill, 2010; Couldry, 2004; Gießmann, 2018). The theoretical potential of media practices and the implications for media studies are an ongoing debate. This study adds to this debate by focusing on collaboration in f/oss as a set of practices. At the centre of analysis is: how collaborative practices come into being and how collaborators negotiate and discuss practices. I understand practices as actions that occur with a certain degree of regularity. They are not the expression of spontaneous inspirations, but they are the result of discussions and negotiations about which practices need to be adopted in a specific situation. Hence, practices are not individual choices. They emerge through an interplay between actors. Practices can be followed all by oneself but in essence they are to be performed with others. The activation and development of practices emerges through communication with others. Yet, as I will explain in more detail in the theoretical chapter, human and non-humans act in developing practices. In the context of this study, the software that is produced as well as the software that is used for the production play a major role in defining the practices. Software that is used to work on specific areas of the project allows certain practices and denies others, or it gives paths for specific practices to follow along. Thus, the interplay that results in the activation of a practice involves negotiations between human collaborators as well as between humans and non-humans.

Behind the considerations to focus on practices is a theoretical motivation that directly concerns the concepts and categories that can be used to study media. Software has captured all other media forms, thereby making obsolete the technical or aesthetic differences between media technologies that were traditionally used in media studies to compare the characteristics of media. It has been argued (Gießmann, 2018;

Schüttpelz & Gießmann, 2015) that methodological and conceptual approaches directed towards collaborative practices allow to explore contemporary digital media culture(s), especially if practices are embedded in a web of organisational, technical and social ties. I think that a focus on practices allows to capture this web. It gives agency to humans to use software to evolve and cultivate collaborative practices that have originated in software development (Schmidt, 2008, p. 275). It also allows to give agency to software, as it induces these collaborative formations with a practical-organisational logic. Software culture does not mean that software determines how social ties are structured, how communities are organised, and how people work together. Even though media culture is to a large extent determined by software corporations that control and restrict what practices can be used, software has proven to be open and unpredictable. F/oss shows how software is malleable and is only realised through specific practices. By studying software as a set of practices that are based on computing techniques as well as on the use of digitally networked infrastructures (Gießmann, 2018), we can understand its mediality, that is software's specific capability to impart the interpretation of one self and relations with others.

However, the proposed focus on practices does not mean to disregard the importance of the discourse. The notion of collaboration is embedded in a rhetoric that promises social progress through technological innovation. The first chapter will give insight into the most important discursive formations and tendencies involved in f/oss. These discourses will be prominent throughout the study as they will be endorsed or questioned and scrutinised by collaborators when they reflect on their practices. Struggles within the project also show how discursive formations are not set in stone. Rather they are in flux, offering alternative vantage points and possible breaking lines. Putting practices at the centre of this study does not mean to forget about the importance of the institutions, organisational forms, and governance models as important factors in a cultural formation that "gives birth" to LibreOffice. In that sense, practices and discourse require each other, a nexus that has been highlighted by Foucault's (1969/1972, 1966/1994, 1975/1995) work. Indeed, one way of studying collaboration in f/oss would be to analyse it as an ideology, to deconstruct its

linguistic formations and link it to a set of practices that can be observed. This study however focuses on the practices that collaborators engage in to sustain LibreOffice, a free and open source software project. I will show how collaborators activate practices and how they put them aside. The practices in question do not just concern writing code. Writing documentation, translating, marketing, writing and negotiating legal frameworks, and a lot of invisible work to glue the projects' participants together are cornerstones of LibreOffice. Producing LibreOffice involves technical practices as much as it needs a governance mechanism, and formations to order the practices. Practices need to be coordinated among participants, and these coordinative practices have an important role in a project that is largely realised through distributed remote collaboration.

0.3.2. The sociotechnicality of software

The empirical and theoretical contributions share the same stating point. This study explores how collaborative practices emerge in a software project. From this starting point I want to add to the scholarly understanding of what software is made of when it has become an infrastructure for everyday life. I will not address the collaborative practices solely as a technical effect of software. Rather, I understand collaboration to be influenced by social, economic, and political forces that are interlinked with technology. They are not an expression of software, neither is software the direct result of these force.. The "softwarisation of culture", as David Berry (2015) has called it, needs to be critically assessed by addressing the entanglement of social and technical elements. To further the understanding of this sociotechnicality is this study's contribution. I will show how collaborative practices emerge with the production of software. As software has proven to be the new meta-medium that is replacing the traditional media forms, it is a core interest of media studies to look at the cultural logic that extends around it. It is a move to add to the understanding of software by looking at the collaboratives practices that are activated for its production. The frameworks for collaborative practices are to be analysed and how they are created, maintained, and normalised. Ultimately, it is worth asking: what is under the hood of an office suite besides lines of codes? What cultural concepts go into a free and open source office suite? These questions shall provide a better understanding of software by

highlighting a culture that has media production as its starting point. I do not attempt to deliver explanations of a general software culture. LibreOffice is not even exemplary for f/oss in general, as f/oss culture is too rich and diverse to be described in essential and fixed terms. But the actors of this ethnography can show the potential that f/oss offers and which collaborative practices are possible to be activated in a specific culture that embeds the production of an office suite, a fundamental piece of infrastructure. At a time when software has become part of all important ordering structures it has turned into an infrastructure for our lives. One of the main contributions of this study is that it examines the potential to negotiate this infrastructure and open up possible routes to intervene as citizens, by showing how a collaborative project based on software commons can be sustainable.

0.4. Object of study

Why free and open source software then? The main reason to select f/oss as a field of study for this dissertation was that it reflects the promises and problems that the concept collaboration entails. It offers alternative organisational and governance models that are based on ideals such as equal access, participation, openness and transparency. It is characterised by production processes with low hierarchies, teamwork, and project-based work. It challenges traditional media distribution channels and intellectual property rights as it opposes copyright by using copyleft licenses that allow to use, change, copy, and distribute software. At the same time, f/oss comes with business models that harness the results of a collaboration with communities of volunteers.

F/oss started to receive scholarly attention in the early 2000s. Since then, the ethical foundations (Himanen, 2001) of f/oss have been discussed as well as the opposition that hackers present to the commodification of information (Wark, 2004), while ethnographic work (Coleman, 2013; Kelty, 2008) showed how the production of f/oss relies on political visions as much as on technical expertise. This study arrives late in the sense that the academic hot phase of interest for f/oss may be past. Yet, this off-ness can be turned to advantage by building on the insights of earlier work. In this study, f/oss is also understood as a cultural web of technologies, ethics, political ideals, and economic interests. But putting the emphasis on collaboration as a societal phenome-

non that is marked by ephemeral relations, individuality, and superficiality serves as an antithesis to a consolidated understanding of f/oss provided by earlier studies that emphasised the collectivity and communality of f/oss. This a priori juxtaposition shall highlight intellectual negotiations to offer an insight into models for working together, producing media technology, exchanging knowledge, forms of sociality and the role software plays in all of this.

The object of study for the collaborative practices in the production of free and open source software is *LibreOffice*. Similar to Microsoft's office package it includes programmes for text processing, spreadsheets, slideshows and drawings, as well as a database management system and an editor of mathematical formulae. In contrast to Microsoft's office package, *LibreOffice* is available for free. Not only is it available to download without having to pay for it, it is also free to modulate, change, and distribute. It is also free as it is not bound to one operating system, as it is available for Microsoft Windows, MacOS and Linux distributions while since the announcement of version 5.3 in February 2017 it is also available for the private cloud. And it is free, as its native format is the OpenDocumentFormat (ODF) for all applications. That means that Libre Office supports the file formats of Microsoft office package and most other major office suites.

This study selected LibreOffice based on several criteria. First and foremost, it is one of the largest and longest-serving f/oss projects. It was first released in 2010 and its history dates back to a decade before that. Today it has around 200 million users. The project is an important factor amongst f/oss projects as it shows the possible sustainability of a collaborative project.

Second, LibreOffice was chosen for practical reasons. In contrast to most large groups that engage in free software, it is a project with most of the core members coming from European countries. So far, their annual meetings have all been held in European countries. Their second physical meeting as a group is held at FOSDEM, Europe's largest conference for free and open source software developers, which takes place annually in Brussels. Thus, participant observation, at the LibreOffice conference and at FOSDEM was manageable as the geographical distances are relatively short and the travel costs were low.

Third, LibreOffice is in its nature different to the free software projects that are regularly studied. It cannot be sufficiently explained by focusing solely on the significance of coding. It is not just a group of hackers. As an office suite, a lot of work consists of translating the text of the user interface as well as the documentations. Design is also a fundamental part of the project. Marketing is a central part of LibreOffice: Internal communication to inform the community about what is happening on their own wiki, external marketing to promote the product with governments and interest groups so that they migrate to LibreOffice. The internal translations and flows of communication between these different groups that are needed for producing the software offers an additional layer for a study on collaboration in comparison to other studies on f/oss who generally tend to focus on hackers and writing code.

Fourth, LibreOffice consists of a vibrant community of collaborators. However, it is not an unregulated platform for contributors or a project with a few owners who decide what the next development steps are or decide what contributions to include, as it would be the case for most projects on GitHub, the most popular platform for exchanging software code. The structure and governance model as well as the setup of contributors of LibreOffice is more complicated. Contributors are individual volunteers who are unpaid and who work in their free time. Another major part of contributions come from companies that develop LibreOffice. RedHat, the world's largest open source software company, pays employees (at least partly) to work on LibreOffice. Employees of other companies also contribute to LibreOffice. The third group of people involved in the project consists of a small group of paid employees, some fulltime, some part-time. Such different interests and possible political starting points complement an object of study for the collaborative relations that underlie a free and open source office suite.

0.5. Research questions

Summing up the building blocks: This study is interested in the collaborative production of free and open source software. It asks for the conditions under which collaborative practices in LibreOffice emerge, given that cultural analyses understand collaboration as the expression of a liquid society with an open, unstable character. The main aim is to study collaborative practices by embracing their volatility. This shifts

the focus from seemingly stable media objects toward a dynamic character of media, which are embedded in specific cultural formations. Hence, this exploration shall provide an addition to scholarly understandings of digital media practices. The access point for this cultural analysis are practices to capture the entanglement of humans and non-humans and to highlight the cultural web in which software needs to be embedded to become realised.

1. What are the necessary elements for collaborative practices to emerge?

    1a. What is the nature of the interplay between community, software project, and governing mechanisms?

    2a. How is it possible for collaborators to limit or quit their efforts?

    3a. To what extent are practices an expression of a specific project and can they be transferred to another project?

2. What are the elements of the collective frame of reference which stabilises collaboration and offers options to engage in practices?

    2a. How are practices ordered and who can question the practices?

    2b. How are practices shared and passed on and how are they stabilised?

3. How is collaboration coordinated and governed?

    3a. What ordering mechanisms support the production of LibreOffice?

    3b. What barriers exist for individuals seeking to take part in decision-making about practices?

    3c. Which requirements and/or skills are needed to decide about practices?

    3d. How is the opposition between leadership and self-organisation negotiated?

4. What ideas and beliefs influence collaborative practices and how do they impact a common identity of the project?

    4a. How is a common identity maintained?

4b. To what extent are different beliefs and values allowed?

0.6. Chapter overview

This study is composed of nine chapters. The first is a background chapter that delivers additional context for readers not familiar with free and open source software. A short historical outlook gives insight into the peculiarities of the field of study. It shows that it is centrally anchored in producing software. At the same time, it extends the realm of the technical by offering alternative ways to work together, to share information and to act upon software. Commonalities and differences regarding legal regulations, innovation models, and political ideas within f/oss highlight the diverse setup of the field.

The second chapter offers a review of the literature on collaboration with an emphasis on collaborative and coordinative practices in free and open source software. Research on collaboration and its role in peer production and communities in f/oss will be presented. Studies of f/oss communities show how a collaborative culture has formed around f/oss. This overview frames this dissertation as part of an interdisciplinary research field.

The third chapter presents the theoretical foundation. Here I lay out the proposition to analyse collaboration as a set of practices. By drawing from practice theory, Science and Technology Studies, and specifically from Actor Network Theory, I attempt to provide a framework that understands collaboration as a set of socio-technical formations. I will explain how it is a phenomenon that requires fluidity and heterogeneity, as well as stability. By combining these different approaches, I will elaborate the cultivation of practices within f/oss considering the field's diversity.

Methodology and methods are discussed in chapter four. The problem of combining a social constructivist perspective with the inclusion of technology as an active agent concerns the methodological considerations. The part on methods describes the different stages of the process of writing an ethnography.

Chapter five to eight cover the empirical findings. Chapter five describes the history of LibreOffice. It focuses on the formation of practices amongst a group of people in connection with a discourse and an organisational structure. This formation allows to stabilise a community which ultimately led to the start of LibreOffice.

The next chapter summarises the findings that concern the governance of collaborative practices. It shows the difficulties of providing a stable coherent basis for collaboration, while maintaining heterogeneity, diversity and leeway for individual decision making. It shows how the project's history plays a fundamental role in organising the project and the formation of the collaborative practices.

Chapter seven concerns the coordinative practices within LibreOffice. In the absence of traditional management structures, it presents the effort to maintain a stable line of production while including different levels of expertise and skills. Additional focus is given to the role that status has in these collaborative formations.

The politics of collaboration concern this study's final empirical chapter. The different political ideas within f/oss that were described in the first chapter will show the fault lines within LibreOffice. Conflicting ideas about openness, the role of the community in a f/oss project, and the involvement of business interests highlight the variety and disparity within f/oss as well as the difficulties in maintaining stable practices to produce LibreOffice.

The final notes evaluate the main outcomes of the study. I highlight the composition of the collaborative practices in LibreOffice. I propose a more general model of collaborative practices and its potential advantages for media studies regarding the exploration of digital media.

1. Copyright and licensing information – Free, open source, free and open source

Software comes in many forms with different types of patents, ownership and copyright licenses. Software can be proprietary, non-proprietary, free, or open source. The object of this study, LibreOffice, is defined as free and open source software. The use of this label is an important characteristic for LibreOffice; it is the result of a complex legal struggle, and is fodder for political and philosophical disputes. To provide more context on the implications that has for the notion of collaboration, and for LibreOffice in particular, it is important to unpack of how free and open source software can be defined and what the term stands for. Thereby the implications for LibreOffice self-defining as free and open source software become clearer and the ground for the following chapters is prepared.

1.1. Source code: open or closed

Software code contains all the instructions for the computer to execute tasks. Most programming languages contain two parts: object code and source code. Object code communicates directly with the computer and is usually written in machine code containing only 0 and 1 for the computer's binary system to read. Source code is readable by humans. The written instructions are compiled into object code for the computer to execute. Source code is what programmers work with; it is the foundation for software creation. Users who understand a software language can customise programs, change it to their needs, and develop other programs based on the code written by others. Source code is written in text files which allows comments for other programmers to read to help them understand the code, or other programmers can comment on it. These comments are an important art of software. They can function as a reminder what a specific element of the code stands for or which function it has.

Only if the source code is open it is accessible for users to read and write the code and comment on it, add to it, make adaptions, and share it. Both open source software and free software offer free source code to enable the collaboration between developers. They stand in contrast to proprietary software, which is defined by a closed source code. Users of proprietary, or non-free, software cannot view or change the source

code. Copyright laws (and often patents) are applied to protect the integrity of the code. With a closed source, software is intellectual property with a clearly defined owner and that must not be modified or shared with others. In addition to the legal difference, there is also a contrasting notion that free and open source software has in common: The idea that collaboration between developers should be facilitated by an open source code.

1.2. From open to closed to free and open

Historically[3], the source code of software was open. Software used to be written for a specific machine, to give instructions to a particular set of hardware. In order to adapt it and use it on a different machine it needed to be modified. That had to be done by accessing and editing the source code. As the diversity of hardware filtered down to a few dominating companies, so-called high-level programming languages became independent of the computer's hardware. One programming language could now be used on a variety of hardware architectures. Whereas before, hardware was the only asset of the computer market, software became the distinctive feature of a computer (Wexelblat, 1981).

When computer corporations started to realise the potential market for software in the late 1960s, they began to sell it as closed-source in a compiled-executable state. That meant that users could not study the programmes anymore and adapt it to their needs. Additionally, copyright law was extended to software which meant that sharing the software was illegal, and it was sold under a license. What emerged is now called proprietary software. It is protected by copyright and patents. The license guarantees the publisher or creator to retain the intellectual property rights. Under a proprietary license the use is restricted by specifying the number of machines on which a programme can run. Also, the right to study and share the software, the basis for a free collaboration amongst independent distributed developers, was outlawed with the idea that unregulated sharing stifles innovation and impedes to develop better software products (Hippel & Krogh, 2003).

3 Ceruzzi's A history of modern computing (Ceruzzi, 2003) offers the most compelling historical overview of computing in the Twentieth century, with the caveat that it focuses on developments in the United States.

An alternative viewpoint argues that copyright not only gives the software publishers too much control. Also, by outlawing copying, modifying and changing the software, proprietary software impedes collaboration between developers and therefore suppresses innovation and restrains the development of new ideas and enhanced software. The underlying idea is that giving access to everyone results in a code with fewer mistakes and better security. This position led to the creation of licenses which grant users the rights to share, view, modify, and redistribute their new version of the source code.

These differences between open source and closed source highlight two ideas of how innovation is thought to work best: competition based on propriety opposed to collaboration facilitated by sharing. What both ideas have in common is a techno-legal complex that regulates user practices by creating or impeding the possibility of collaboration. Both free software and open source software refer to non-proprietary computer software that allows users to share knowledge by providing licenses that keep the source code open. They are not the property of a person or an organisation, and they are not protected by a patent or copyright. In contrast, they are publicly available. However, within the group of non-proprietary software are marked differences between free software and open source software that are reflected in LibreOffice.

1.3. The freedom of free software

As a reaction to the increased closedness of software during the 1980s, Richard Stallman, a programmer and researcher at MIT, started to develop a new computer operating system which he decided to call GNU. His aim was not only to build a system that was free to use and to share, but also to create a licensing model that guarantees these rights to its users and prevents the software from being appropriated by for-profit companies. It was specifically designed to protect the freedom of computer users by specifying four essential freedoms:

> - The freedom to run the program as you wish, for any purpose (freedom 0).
>
> - The freedom to study how the program works and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.

> - The freedom to redistribute copies so you can help others (freedom 2).
>
> - The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this. (Free Software Foundation, 2016)

Software is not free because it is free of cost, but because it guarantees the freedom of computer users to study, share, modify, and distribute copies of that software. The freedom of a technology corresponds directly to the freedom to collaborate. Software can only be free if users have the freedom to collaborate. To stabilise this collaborative freedom, Stallman compiled the GNU General Public License (GPL). It is designed to protect the four freedoms and requires that copies and derivative works of a free software programme be offered the same license.

> Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. (Free Software Foundation, 1989)

Software that would follow this general rule is called free software, Stallman decided. The GPL is a so-called “copyleft”, a hack of copyright. It inverts the copyright system by restricting the right to limit sharing. As well as copyright, it protects the integrity of the product while it rests on a different definition of what that integrity is. While proprietary software can be considered as having integrity by not being shared, the idea of free software to obtain integrity is by spreading to allow sharing, using, and collective producing. Free software is not about protecting a code by closing the source or restricting its exchange and use, but is instead about ensuring openness and the ability to collaborate. The license of a particular piece of free software cannot be negotiated, as it is decided by those distributing the software, equal to copyrighted software. Thus, copyleft is a hack in its purest form: an astute observation of a system, followed by using its features for your own needs. The only difference here that it is not computer code but legal code which has been hacked. Thus, free software is a technical and legal expression for

a campaign that supports the freedom of software users. Its fundamental stance is that the social phenomenon of collaboration needs to be accompanied by the materiality of free software.

1.4. Open source software

After distributed collaborative development process of free software started to turn out a success, some developers became concerned that free software's philosophy of opposing software that is non-free would hurt their business interests. A more pragmatic approach should be symbolised by a different label and a new institution. In 1998 the term open source software was coined and the Open Source Initiative (OSI) was founded to emphasises the distinction with free software and with the Free Software Foundation (FSF). The main idea behind the split was to focus on the collaborative production model as the main contribution of open source software by highlighting the importance of the open source code. Discarding the emphasis on political ideas anchored in terms like "copyleft", open source advocates formulated what they understood as a 'pragmatic, business-case grounds' (Open Source Initiative, 2018). Open source was designed to stress the importance of the development and innovation model and disregard the ethical and societal aspects: 'For the Open Source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem and free software is the solution.' (Stallman, 2002, p. 43)

One of the first steps of the Open Source Initiative after its foundation was to develop "The Open Source Definition" (Open Source Initiative, 2007). The list of ten principles includes the four software freedoms of using, sharing, modifying and re-distributing but it avoids the word freedom. Instead, it states that to be considered open source, a license must not discriminate against persons, groups, or fields of endeavour. This intends to signal a more business friendly approach:

> The Open Source Initiative is a marketing program for free software. It's a pitch for "free software" on solid pragmatic grounds rather than ideological tub-thumping. The winning substance has not changed, the losing attitude and symbolism have. (Open Source Initiative, 2002a)

In a strict technical sense, free software and open source software do not deviate from each other. The schism between these two factions did

not start for technical reasons but for different political ideas. The OSI takes the liberal view that politics should not stand in the way of business interests. Therefore, copyleft – which guarantees the software to stay free – should not be applied. Open source can be understood as appropriating the collaborative production model of free software. The goal is to establish a market that makes use of the forces of distributed systems that include private individuals, communities and businesses. Even though the *Free Software Foundation* never advertised not to sell software (Free Software Foundation, 2017), the open source movement positioned itself outside the possibility of being seen as a countercultural movement.

> We have a winning product, but our positioning, in the past, has been awful. [...] Mainstream corporate CEOs and CTOs will never buy "free software." But if we take the very same tradition, the same people, and the same free-software licenses and change the label to "open source" – that, they'll buy. [...] In marketing, appearance is reality. The appearance that we're willing to climb down off the barricades and work with the corporate world counts for as much as the reality of our behavior, our convictions, and our software. (Open Source Initiative, 2002b)

In the two decades that have passed since the start of the *Open Source Initiative*, what was a schism between free software and open source software has mellowed. The direct demarcations against free software have been taken down from the *OSI* website and the differences have been smoothed out to an extent that one large community seems to come together at free and open source software developer conferences, even though occasional remarks and comments about “the other side” or “the others” can be heard. But compared to the strong rejection that characterised the schism between free and open source, the differences are now subordinated under the umbrella that is free and open source software[4].

However, two different belief systems based on software development remain. One the one side there is open source, which stands for a model

4 Free and open source software (foss or f/oss) or free/libre and open source software (floss) are used alternatively. The latter version intends to emphasise that free stands for freedom, not for free of cost by adding the word “libre”. Free can mean both freedom and free of charge, libre unmistakably refers to freedom.

of collaboration in which innovation works by relying on distributed participation to gather the knowledge and efforts of many. Instead of giving access to the source code only to a small group of developers, an open process that allows the participation of everyone would result in superior product as more people equals a better chance to find bugs and solutions. The question of ownership is not necessarily participatory, as in the end the owner of the software decides how to use it. On the other side there is free software that subscribes to the idea that an open process leads to better software but with the added understanding that it is a social act that rests on solidarity - no one shall not share the outcome of a communal effort.

1.5. LibreOffice: Free and open source software

The main reason for choosing the term free and open source software is that LibreOffice labels itself as a free and open source office suite. By using the alternative label free and open source software it avoids the dichotomy within non-proprietary software between free and open source. The political-philosophical differences between free software and open source find expression in licenses: Whereas most open source software is also licensed under copyleft and is therefore free, some open source licenses are permissive but *not* copyleft. In the latter case, the rights to use and share are given but it is also possible to relicense a derivate, even if the chosen license is proprietary.

In the case of LibreOffice, the source code is protected by the Apache License 2.0 which is a permissive license, a legacy of being a successor of OpenOffice.org. Everything built on top of that by LibreOffice is covered by the Mozilla Public License Version 2.0. By releasing the software under the Mozilla Public License Version 2.0 (MPL2.0), LibreOffice has chosen a license that combines aspects of free and open source licenses. The MPL2.0is a hybrid between a permissive license and a copyleft license. On the one side it offers a permissive element by allowing that derivative works that build upon software that is licensed under MPL2.0 can be licensed differently. It still manages to include copyleft by stating that access needs to be given to the original parts that are covered by the MPL2.0. The main feature of this license is compatibility.

Every committer to LibreOffice sends in a license statement to declare the further usage of their contributions under both the Mozilla Public License and the strong copyleft Lesser General Public License:

> All of my past & future contributions to LibreOffice may be licensed under the MPLv2/LGPLv3+ dual license. (The Document Foundation, 2019b)

The license has the advantage for LibreOffice to collect contributions from a big pool of collaborators with different political ideas regarding software. The project claims a 'strong commitment to copy-left licensing' (Lauhakangas, 2019), expressed by using the strong copyleft Lesser General Public License (LGPLv3). On the other hand, it uses a weak copyleft license like the MPLv2, which can be combined with the permissive codebase that LibreOffice is built on. And a strong copyleft license does not provide 'advantages around attracting commercial vendors' (Lauhakangas, 2019). The licenses as an expression of software politics are not used exclusively but they are combined so that both free software and open source elements are covered. The mix of licenses that LibreOffice uses shows how free and open source can deviate or intersect and converge.

1.6. Interim summary and outlook

The differences as well as the parallels between free software and open source software define free and open source software. Starting from the technological foundation of an open source code, a variety of collaborative practices unfolds that is made of legal frameworks, technical expertise, political-philosophical notions, and ideas of social structures. While free software and open source software refer to software with an open source code, there are important differences between these two types of non-proprietary software. Behind the difference code, licenses, and organisations rest two opposing ideas of creating software together. One the one hand, open source software represents a platform to exchange and build code together. A model for innovation rests on the connective nature of open code and the simple idea that more people can build better things together. On the other hand, free software emphasises the collective culture that can result from exchanging code. The idea that code can be an act of freedom is deeply rooted in communitarian beliefs. F/oss and projects such as LibreOffice that declare themselves to belong to this third group are at the junction of these two

tracks of non-proprietary software and a formidable starting point to study the collaborative structure of software and the way that different elements are assembled together.

2. Credits and acknowledgement (Contextualising the study)

There is no unified body of scholarly literature on collaboration in which to embed this study. Contextualising this study thus requires perspectives from different academic disciplines. The first approximation to understand collaboration as a socio-technical phenomenon is offered by research on peer production. This line of research shows how the ideas of network-based collaboration are linked with software. Then I will discuss work on coordination and cooperation where I mainly draw from Computer Supported Cooperative Work (CSCW). After that, I will present research on coordination and peer production in free and open source software, contrasting an often-celebratory rhetoric with more critical assessments. The studies discussed in this part come from media studies, software studies and science and technology studies. Together, these explorations in different fields and disciplines provide a set of initializing ideas for this study of collaborative practices in free and open source software.

2.1. Commons based peer production

The first part of this study is borrowed from the concept of peer production, or commons based peer production. The notions of self-organisation and flat hierarchies in addition are characteristic for the literature on collaboration. The concept of peer production (p2p) starts from the same assumption. The necessary condition is that the cooperation is free, not forced or controlled by a firm or an institution, a necessary distinction to distributed work on digital platforms (Bauwens, 2005). In peer production, it is not an institution that makes a profit upon the shared work of individuals. The concept also adds that the products of the collaboration are shared amongst collaborators. This is one of the differences that distinguish peer production from phenomena such as crowdsourcing. In the absence of a market or a firm, the peer-to-peer project is self-governed; the use-value can be shared and it is freely accessible for everyone. Bauwens (2005) argues that the shift toward peer production is marked by a new mode of production, a new mode of governance, and a new mode of distribution: The production is not centred around use value for its participants instead of exchange value; instead, it is governed by the participants themselves. The produced use-value

is accessible for everyone. Benkler (2016) adds that p2p rests on decentralisation in planning and execution, on the feature to combine diverse motivations, and a governance model that does not rest on property.

The recognition that bottom-up institutions can preserve common goods and that such forms of governance are better suited to achieve sustainable management of resources than centrally managed alternatives, earned Elinor Ostrom the Nobel Prize in Economic Science. In contrast to the classic school of thought in economics, she explained how cooperation and collective action (Ostrom, 1990) can be the driving factor behind sustainable management of common goods. The concept of peer production builds on Ostrom's work: What makes a peer production process commons-based is the access to the resources and the products of the collaboration as well as a communal stewardship. In other words, it 'is based on the open input; a participatory process of coordinating the work; and a commons as output' (Bauwens et al., 2019).

The difference is that information is much easier to share than material goods. The latter need additional effort to transport and distribute them. For information, the opposite is the case. For information to be proprietary, barriers need to be constructed. Therefore, decentralisation received attention with the growing importance of digital media as networks provide a fine infrastructure to share information. It is claimed that these technical changes offer new forms of cooperation. Yochai Benkler (2011) and Clay Shirky (2009) argue that networked digital media manage to trigger pre-existing cooperative networks that have not been activated before. Their argument implies that digital media are an infrastructure for cooperative relations. They are the ground for cooperative networks to be activated. Instead of rivalry and competition these new forms of working together are built on teamwork and cooperation. According to this idea, cooperative networks that grow from the bottom up replace centrally organised systems. They neglect strict hierarchies and they are built on 'trust and long-term cooperation' (Benkler, 2011, p.11). Instead of weak ties and fluid associations which are often used to characterise collaborative relations, peer production offers a model to zoom in on f/oss in regard to a sociality that offers stability. Cooperation is the fundamental ingredient in Benkler's

concept of ‘commons based peer production’ (Benkler, 2006). He argues that structures which foster cooperation require ‘reciprocity and improvement’ where people give contributions freely and expect others to share theirs, with everyone being able to use and elaborate on the combined creations.

This assumption is based on the premise that all participants work in free cooperative relations towards a common goal while they maintain the same status. In the absence of a central authority or a super-structure, peer production rests on the relations among collaborators. The processes of production are governed by a community of peers. The concept of p2p rests on a sociality that emerged in web communities in the early years of the Internet. Some argue that the collaborative production of commons (commoning) is better defined by the social relations amongst collaborators instead of the shared ownership and self-organised governance. It is supposedly ‘a sociality that does not merely exist because it is beneficial for productivity; it is built on care for each other´ (Korczynski & Wittel, 2020, p. 11).

Benkler’s definition of commons-based peer production comes close to Stallman’s arguments in favour of the freedom that free software offers. The requirement for the term “commons-based” is ‘a framework of social relations’ (Benkler, 2006, p. 62) that replaces a proprietary system. Out of such a structure emerges a freedom in which ‘the inputs and outputs of the process are shared, freely or conditionally, in an institutional form that leaves them equally available for all to use as they choose at their individual discretion’ (Benkler, 2006, p. 62). The best-known examples for commons based peer production is free and open source software, and Wikipedia. Yet Benkler does not necessarily understand f/oss as an extraordinary phenomenon that has transformed the software industry. Rather, he sees f/oss as an expression of a broader cultural trend that rests on sharing and cooperating as the key for a new mode of innovation. F/oss has become an important factor beyond the software industry not only because it showed how the inclusion of volunteers reduces production costs. Rather, it is the knowledge of production. Open collaboration has become the dominant model for innovation in some other economic areas. However, it is not only production practices but also ‘values practices, such as loyalty to friends, conviviality, mutual aid, care, and even struggles’ that

are developed in the commons (De Angelis, 2017, p. 12) The production and reproduction in the commons refers to both these practices and even to the outcome of the practices. Reproducibility allows commoning, the process of engaging in the production of commons. Yet, it also refers to the autonomy of the peers as the product and the process are forkable: they can be copied and built on.

The best-known work on peer production is characterised by a positive tone. It highlights flat hierarchies, the absence of bureaucracies or other control systems, and intellectual property rights that opposed copyright. But some of this early work highlights the problematic nature of an idealised version of peer production. In the absence of central authorities, Shirky (2009) stresses the importance of negotiation and hints at some form of struggle that defines collaboration.

These struggles are further analysed by more recent work that focuses on the problems of equality in peer to peer projects and the development of governance models that include hierarchical structures (Shaw & Hill, 2014). Some studies come to the conclusion that hierarchies are the reason for successful peer to peer projects (Healy & Schussman, 2003). Network-based form of organisations are increasingly understood as "heterarchies" (Cumming, 2016) whereby bottom-up, top-down and peer to peer models complement each other. Another line of inquiry has asked for the political substance of p2p projects (see Kreiss et al., 2011; Tkacz, 2015). From this perspective, peer production is analysed for the ideologies which are at work to produce the imagination of sharing equally or openness.

The early work on commons based peer production gives fruitful topics for the analysis of collaboration in f/oss. In addition, more recent research shows possible lines of critical inquiry into collaborative projects and the importance of scrutinising it regarding its complex realities. These complex realities are the mental horizon for this study. By emphasising the term communing, the focus shifts from the product to the processes that are deployed and need to be reproduced; the bonds between collaborators, the collective governance, the politics in play. Thus, peer production emerges as a set of practices that remain under construction.

2.2. Computer supported cooperative work and work place studies

The second pillar for this study stems from computer supported cooperative work (CSCW). The focus on teamwork characterised this field from the beginning (see Greif, 1988). Bannon and Schmidt (1989, p. 362) point out that 'the term "cooperative work" is the general and neutral designation of multiple persons working together to produce a product or service'. Even though teamwork or cooperation as the common denominator of this field is not further scrutinised, CSCW research delivers insights that are useful for this study.

The celebratory tone that characterised much of the early literature on commons based peer production is largely absent in CSCW. The negotiation between individual autonomy and the interventions by orderings structures, mainly organisational processes and computational practices, are central to CSCW. Conflict amongst group members, competition and rivalries, the rejection to cooperate in technologically induced settings are at the centre of interest. CSCW makes the possibility of conflict and negotiations part of the analysis of collaboration in f/oss (see Easterbrook, 1992).

CSCW is a broad interdisciplinary field that has assembled a vast array of different concepts from various disciplines. As such it is best understood along concepts that have become foundational for CSCW (for a proper overview see Schmidt, 2008). Of interest for this study are those that focus on question such as: How is collaboration in light of different interests established and maintained? What are the collaborative practices that are activated? What role do non-human actors have in terms of organising a collaborative project? How do material settings influence collaborative structures?

2.2.1. Articulation work

One of the concepts that has been fundamental for CSCW and for this study as well is called articulation work. Developed by Anselm Strauss, it highlights the efforts to order and structure teamwork for any cooperation to emerge. Strauss defines this articulation process as 'the overall process that brings together as many as possible of the interlocking and sequential elements of the total work, at every level of organisation – and keeps the flow of work going. (Strauss, 1988, p. 175). Articulation

work is an inevitable part of cooperative processes. Strauss' model does not start from the premise that work is tightly organised. Rather, his conceptualisation of work focuses on dynamics and flexibility. The division of tasks can only succeed if they are negotiated through such articulation work. Putting together sequences, tasks, and lists help to order collaborative projects.

2.2.2. Situated actions

Lucy Suchman's concept of situated actions stands symbolic for the CSCW approach as it is concerned with problem solving and the negotiation of a collaborative order. Suchman's work offers elements to understand collaboration as a set of sociotechnical practices even though she uses the term 'actions'. Actions, she points out, need to be situated. They need to be adapted as they are situated in a specific context. With the process of adaption, Suchman's concept targets the interplay between machines and humans at the workplace. She argues that the problem in using intelligent systems is not based on technical errors. Instead, the problem is a lack of experience how to deal with intelligent systems. The solution is not to tweak or to optimise the system: 'However improved the machine interface or instruction set might be, this would never eliminate the need for active sense-making on the part of prospective users.' (Suchman, 1987, p. 9) Thus, actions, she argues, do not follow standard procedures or plans that have been made in advance. Rather they need to be situated. By establishing routines through articulation work, she argues, humans create the necessary conditions for technical processes. She proposes to study actions on the microlevel (Suchman, 1996). The focus on micro practices also points the way to ethnography as an appropriate method. Empirical observations are pivotal for her analysis of plans, situated actions and the necessary negotiations.

2.2.3. Boundary object

The bond between human and non-human actors, and the role which non-humans play in collaborative projects is further highlighted by Susan Leigh Star's work. She introduced the concept of boundary objects to highlight the web of connections which is made up of humans and non-humans. Non-humans take part in sense-making not in a determining entity but as objects that can be interpreted in different ways.

Just as humans, non-human objects are inherently ambiguous without losing their identity: 'Boundary . . . are not just temporary solutions to disagreements about anomalies. Rather, they are durable arrangements among communities of practice.' (Bowker & Star, 1999, p. 307)

I will come back to these three concepts in chapter 3, where I will present the theoretical framework for this study. However, they also served as a starting point for this study as they have served as vectors for research on coordinating and organising collaborative projects.

2.3. Collaboration and coordination in f/oss

Further context is given to this study by studies of collaboration and coordination in free and open source software. Some studies show how it is possible to present f/oss as a success story without neglecting a critical analysis. Many of them draw from concepts that are widely used in CSCW and workplace studies. The connection to commons-based peer production is given, as f/oss served as one of the prime research objects.

Anthropologists have widened the view on f/oss by embedding the production of software in its cultural and political context. Christopher Kelty (2008) has shown the cultural significance of f/oss. His study has a general positive tone, yet it critically asks for the socio-political ideas behind free software. He concludes that rather than ideologies, it is the practices that define f/oss: 'Free Software and Open Source share practices first, and ideologies second' (Kelty, 2008, p. 113). As such, he dissociates f/oss from social movements and focuses on them as "communities of practice" (Lave & Wenger, 1991; Wenger, 2008; Wenger et al., 2002). Collaboration in f/oss becomes a question of situated learning. Referring to Suchman's work on situated actions, collaborative practices in f/oss are situated practices that need to be learned and ordered. This shall not mean to disregard the role of politics in f/oss. Coleman's ethnography "Coding for freedom" (Coleman, 2013) highlights that hackers understand their practices of sharing code as a political expression. In this specific environment, collaboration itself becomes a political expression. Coleman and Golub (2008) assess libertarian ideals as a constitutive element of hacker practices, yet in contrast to Himanen (2001) they do not summarise these ideals under the umbrella of a unifying hacker ethic. Rather they point towards recurring liberal ideas

that are constantly re-negotiated. Debating freedom is a decisive element for a moral discourse amongst participants in free and open source software. These two major contributions to studying f/oss offer an analysis of the connections between practices and politics in LibreOffice.

2.3.1. Hierarchies

The analysis of a f/oss culture is enhanced by a critical assessment in comparison to the promises of commons-based peer production. Mathieu O'Neil (2009) has analysed the practices of ordering in Debian. Debian is probably the most researched project not only because it is large but also because it has developed a social contract, the Debian Social Contract (Debian, 2004). This text contains that Debian can only include free software, that user requests have priority, and that bugs are managed in an openly accessible system. Yet, O'Neil's work shows that Debian is not shy to implement hierarchies to order the production of software. A project lead is elected every year, and the admission of new participants follows a designated plan of mentoring. The idealised notion of equality that is part of the concept of commons-based peer production is also contrasted by Crowston and Howison (2005). They crystallised an organisational model out of case studies on f/oss projects. They conclude that f/oss projects share a similar structure that has a group of programmers at the centre who do the major work on the code. Around them co-developers help in the maintenance and repair of the code as well in the implementation of new features. Even bigger is the group of active users that do not code themselves. They test the software and report errors. Finally, passive users do use the software but do not report back to the project. Several studies (Berdou, 2011; Crowston & Howison, 2006; Louridas et al., 2008) have highlighted the existing hierarchies and power laws in f/oss projects. This is a portray of a rather centralised organisational formation which stands in contrast with the idea of flat hierarchies that has defined earlier research on f/oss and commons based peer production in general. In addition, the question then is if f/oss projects are run by an organisational elite that is in steady contact with an inner circle of other projects that condition the f/oss scene en bloc. From this perspective collaboration in f/oss emerges as a phenomenon of ordering within a project and as a way of ordering a whole scene.

2.3.2. Organising and equality

By understanding the cultural peculiarities of f/oss, another line of research highlights the difficulties to balance important values such as the individual autonomy for collaborators with the governance structures that are needed to organise a project. Spehr (2003) proposes for a free collaboration that individual autonomy is essential. Individuals are free to collaborate means that they are free to accept (or to not accept) and question the distribution of cooperation. They can also subject their collaborative efforts to one or more condition. The importance of autonomy is of special relevance for free and open source software projects. F/oss is embedded in a hacker tradition that implicitly subscribed to a set of ethical guidelines such as sharing, openness, decentralisation, free access to computers, and world improvement (Levy, 2010). Even though the importance of ethics for hacking is disputed (cf. Vadén & Stallman, 2002), individual autonomy and a mistrust in authority is still central for free and open source software projects (metac0m, 2003). Those who have position of authority in f/oss projects are often met with mistrust or suspicion (Coleman, 2013). O'Neil (2009) has shown how infighting in distributed collaborative projects can start because of this mistrust in authority and the importance of autonomy. In the Debian project, one of the oldest and largest free software projects, a conflict between a contributor (SL) and team leaders escalated on questions of who was right about a bug. After a heated exchange on IRC, the team decided to remove the contributor's access rights to upload files directly to the repository. Instead, the contributor had to send their work to other in order to integrate them. SL refused to do so but the decision stood even after a long discussion that spread beyond the project. In the end, SL got expelled. It harmed the project but it was 'a question of honour' and autonomy as O'Neil (2009, p. 143) called it. It was also a balancing act between these ideals and a governance structure that gives some people in a collaborative project additional duties and decision-making rights.

Conflicts that emerge in accordance with the task of organising f/oss projects are a popular theme in research on f/oss. Studies contrast the ideal of equality of peer production, and values such as autonomy and openness in f/oss, with the necessity of organising a distributed project

that gives people certain duties such as mentoring newcomers or leading the process of decision-making. Therefore, the coordination of a f/oss project needs to be constantly negotiated as the opposition between hierarchies and collaboration in f/oss is dynamic and sways between organising and resistance, as Karatzogianni and Michaelides (2009) point out.

Rozas' (2018) also highlights this organisational dynamic between the urge for decentralisation and the need for formalisation and the subsequent constant negotiations between contributors. His research on the Drupal community shows the emergence of polycentric governance, which is characterised by a number of centres for decision-making. The introduction of a distributed authority is intended to find a balance between hierarchy and the equality of contributors.

Thus, tensions are not necessarily the effect of the failure to organise a project properly. Rather tensions between contributors are given with the structure of distributed collaborative projects. In these debates, the quality of the contribution in question plays a dominant role. In the open source scene, quality is the benchmark for right and wrong. Thus, a culture of debate, often creative and witty, evolves around technical discussions (cf. Coleman, 2013). The implementation of a structure that embraces these "bestiary" of organisational forms (Nunes, 2014) is needed to allow for the conflicts to be resolved. And, as O'Neil's example above shows, the developers need to be willing to compromise otherwise the projects suffers.

2.3.3. Community and practices

The practices typical for f/oss also create tensions between contributors as a lot of them require reviewing other people's work and judge it. The debate culture and the governance structures that are needed to facilitate these practices are embedded in an ethic built on intrinsic motivations as well as on a desire to learn. Alami, Cohn and Wasowski (Alami et al., 2019) argue that in f/oss projects, an ethic of caring can be found. Not only do participants want to learn more and better their skills, they also want help others to learn: they care about the software and about the others who contribute to the same project.

This ethic of caring can also relativise the importance of the technical quality as a benchmark for accepting a commit into a project or not. F/oss communities depend on ongoing contributions. Thus, the review process is not only relevant from a technical aspect but also from a social perspective. Alami, Cohn and Waiswowski (Alami et al., 2020) interviewed contributors of five f/oss communities. They found that the review process can have lenient tendencies so that new contributors are not discouraged by harsh criticism and rejection. For f/oss projects, it is fundamental to organise the review process with that in mind. Mentoring plays a significant part here as they (Alami et al., 2020) explain - not only to establish qualitative norms but also to create a welcoming environment.

Of interest is also the political economy of free and open source software and how it influences the practices and strategies of organising. F/oss has served as a prime example that self-organised production can work and that it can compete with software produced by large firms. However, the notion of commonly accessible use value as the product of commons-based peer production and the absence of firms has been scrutinised in recent years. The sustainability of distributed collaborative projects is regularly tested - in different forms compared to a workplace (Söderberg, 2012). In recent years, the direct and indirect involvement of for-profit companies in f/oss has become the norm rather than the exception. Free and open source software in particular shows how the computer industry has adopted f/oss and its practices and developed new business strategies that include peer-to-peer development structures, as well as cooperation with and investments in f/oss (Young, 1999). Corporate involvement in f/oss has become the norm (Birkinbine, 2020). This further muddies the clear separation between companies and communities that has been made by earlier literature. In that context, Berdou (2011) showed how paid developers are more likely to maintain critical part of a projects' code base. Butler et al. (2021) have taken a closer look into the engagement in f/oss projects by developers paid by companies. They conclude that they engage in practices which are congruent with their capabilities, and which satisfy primarily their own needs, or their companies' strategies. However, they point out that the engagement of companies differs in many ways. Apart from code contribution, companies also take part in community

events, they financially support projects, and they engage in governance structures to advance their strategic interests.

The evolving dynamics between companies and f/oss projects have also led to analyses of the practices that f/oss projects can activate to maintain their sustainability amid possible domination by companies. Forking as a practice to move the community to a different project while keeping the code has been at the centre of attention. Originally a technical practice to modify code (Nyman & Mikkonen, 2011), it has received a social dimension in f/oss projects, as it is used to keep the community intact if a company threatens its sustainability. Forks have become a frequent strategy (Robles & González-Barahona, 2012). Some even argue that the possibility to fork the code keeps f/oss 'communities vibrant and companies honest' (Nyman & Lindman, 2013). In direct relation to LibreOffice, Gamalielsson nd Lundell (2014) have shown how it was successfully forked the OpenOffice.org project. They argue that the community is the decisive factor by including long-term members and the most active collaborators. At the centre of this fork was the 'successful transfer and evolution of know-how and work practices . . . beyond individual Open Source software projects' (Gamalielsson & Lundell, 2014, p. 144).

2.4. Summary

In this chapter I have presented the context for a study on the collaboration in free and open source software. The different phases of research on commons based peer production show the need to balance a celebratory rhetoric about the cultural potential of peer production with a more critical assessment. Studies show that free and open source software is better understood as a set of complex realities. This suggests taking a closer look at how central concepts in the discourse surrounding digital media such as equality or openness influence the practices of free and open source software.

After that I have pointed towards CSCW and workplace studies to deliver an understanding of free and open source software as a set of micro practices. Hereby, collaboration in f/oss is understood in terms of practices. I have presented concepts that I will come back to again in

more detail in the theory chapter. Not only do they show how collaboration is organised and coordinated, they also point towards the connection between technology, people, practices and discourses.

Finally, I have referred to research on free and open source software which understands software not as a technical object that is produced along technical guidelines that are strictly followed. Rather, practices are embedded in a complex interplay of ethics and politics, discourses, people, technology, organisational mechanisms and governance schemes. This chapter highlights the changing nature of software and points towards considering 'software as a timely object' (Cohn, 2019, p. 423). However, this does not concern the software only. Closely linked to the technological changes are negotiations about how to organise the project, about its politics and the lives of the people involved. The next chapter will present a theoretical framework that can sustain such an assessment of the collaborative practices in LibreOffice.

In the following chapter I will outline a theoretical approach that allows to search for the idea of a peer based decentralised production process at LibreOffice. A framework is needed that offers enough stability to support the production of commons while being flexible and dynamic enough to account for the negotiations and discussions amongst collaborators that have been highlighted by CSCW. Instead of romanticising the teamwork and community spirit, the theory shall provide a basis to focus on practices by acknowledging hierarchies and possible power laws that have been pointed out by research on other f/oss communities. Enough flexibility has to be built into this theoretical framework so that it allows to combine the emphasis on technical excellence, individual autonomy and a mistrust in authority that is typical for hackers according to the presented literature with care and attention towards creating an atmosphere that accommodates people with different backgrounds whether that is knowledge and skills, language, or specialisiation.

3. Installation instruction (Theoretical framework)

Following the literature review in the last chapter, I propose to study the collaboration in LibreOffice as a formation of practices. These practices cannot fully be explained as social formations, or as the result of organisational structures, as the actions that developed out of discourses, or as directly linked to their political economy. Nor are they a purely technical effect or the necessary outcome of technical arrangements. Instead, all these elements come together to inform the collaborative practices that constitute a free and open source software suite such as LibreOffice. Rather than untangling this specific sociotechnicality, separating it in discrete spheres of technology and sociality where one uses the other and exercises power over the other, this chapter will outline how to understand it as an interplay of these forces that inform collaborative practices.

Additionally, the recent literature on (commons based) peer production, free and open source software in particular, and the research that I have pooled under the headers CSCW and workplace studies, all suggest changes in practices and in struggles, negotiations and conflicts as characteristics for collaborative projects. Thus, collaborative practices will be treated as contested. They are in-the-making, and need to be constantly negotiated and activated so that they can be performed. This chapter is thus concerned with theories that can assist in building a theoretical framework which supports a conceptualisation of collaborative practices which are of sociotechnical character. They combine stability and dynamism so that they can be shared while they are also open for negotiations and change.

3.1. Sociotechnical practices

The first approach for theorising sociotechnical practices comes from a renewed interest in practices as a paradigm for media studies. Couldry (2004) has proposed an understanding of media as practices to ask what 'people are doing in relation to media across a whole range of situations and contexts' (Couldry, 2004, p. 199). He characterises practices as regular actions; they are not individual idiosyncrasies but social constructions; and they are directed towards human needs such as coordination, interaction, community, trust and freedom.

Couldry's approach is difficult to include with more depth in this study. He draws from Wittgenstein's philosophy of language, which is hardly compatible with the sociotechnical character of collaborative practices. Yet, it is worth taking note of Couldry's point of departure to develop a perspective to understand media as practices. His practice-based approach does not understand media as objects, tools, or texts. Instead, they are defined by the practices that people engage in - in relation to media. This way of thinking about media connects with the observation that the technical and aesthetic differences between different forms of media have become obsolete as software has combined them (Rieder & Schäfer, 2008; Schüttpelz & Gießmann, 2015). Additionally, the ubiquity and pervasiveness of software complicates differentiations between humans and non-humans in the networks they form together (Rieder & Schäfer, 2008). Thus, to study the collaborative practices in software production can deliver an insight into their sociotechnical character and thereby contribute to an understanding of software as medium.

A shift towards practices helps to understand media as the result of collaborative practices and as the infrastructure for collaborative practices. A practice perspective allows to zoom in on the access to producing information that is provided for everyone by digital media. Software facilitates participation in the design and making of media (Löwgren & Reimer, 2013a). A remix culture (Lessig, 2008) has emerged that allows to act with digital media by sharing and (re-)producing media content at the same time as it allows acting on them. Media have become open to modification in an infrastructural sense. Having been 'backgrounded and taken for granted', they are now open for design (Löwgren & Reimer, 2013b). Yet the openness of digital media is never fully neutral (Kranzberg, 1986). As an infrastructure for collaboration, it is not without influence on the culture that is built with it. Rather it comes with a rucksack full of in-built technical structures, as well as political and ethical ideas that condition collaboration. Studying the underlying processes of creation are of importance then because 'the way we create technical artefacts – and software most importantly – heavily influences the cultural role they will play' (Rieder & Schäfer, 2008). What I mean by the materiality of software thus is not only code, or rare earth materials. Instead, the interplay of discourse, ideas and

technical objects, how they arrange and how it informs the practices of producing LibreOffice is of interest.

3.1.1. The materiality of software

Software studies have provided insightful contributions into the materiality of software. It is not a contradiction to use Kittler's (1993) essay *There is no software* as a starting point, even though it seems to deny software as an object of study. Kittler's dictum stems from the history of the computer, in particular from the history of processor chips. During the early era of personal computers and Intel's x86 architecture every person with appropriate know how could directly tweak the processor chip. In this so-called real mode, a programmer could inform the microprocessor in the CPU's language by developing a set of instructions in binary code, 0 and 1, or in hexadecimal form, 16 symbols (0 to 9 and A to F). The control unit of the processor also uses binary code. Thereby it was possible to give the processor direct instructions in their own language, or machine language as it is often referred to. As chip architecture became more complex, Intel restricted access to the processor. In protected mode, as it is called, it is automatically regulated how much processor power is allocated to a program. Additionally, it gives read and write privileges to programs. Certain possibilities to directly address the hardware were closed for the user. In other words, the user is impeded to optimise the processor their personal purposes. For Kittler (1993, pp. 209-211) that means that users cannot control the machines anymore; instead, machines control users. He emphasises how materiality runs in the background. Visible or not, closed or open, it limits its use, suggest or demands how to use it, offers some possibilities and denies others.

The same problematisation that Kittler applies to hardware is applied in software studies. Lev Manovich (2002) mentioned the need for software studies in his book *The language of new media*. Manovich calls for a critical assessment of software in order to analyse the societal formations that are built into software and hereby made durable. At the time Manovich wrote this, media studies were at large engaged in studying the digital as the virtual, as an alternative to the real, celebrating its immateriality and its exceptional features. Kirschenbaum (2003)

called this the new media studies' romance with the virtual and proposed to address the digital as something very real and material. He argued that digital objects are interpenetrated by material patterns and circumstances - a reformulation of Katherine Hayles' (1999, pp. 13-14) conceptualisation of the 'virtual as a cultural perception that material objects are interpenetrated by information patterns'. Hence, Kirschenbaum takes a materialist perspective in order to look for the ideologies of the virtual. He argues that media studies have neglected to appreciate the importance of materiality. For him, 'software studies is what media theory becomes after the bubble bursts' (Kirschenbaum, 2003, para. 14).

What such an approach could look like is further outlined by Kathrine Hayles (2005) in her book *My Mother Was a Computer*. By comparing computer code to speech and writing, she concludes that the boundaries between humans and machines have become blurred. She claims the universe is not created by speech acts or writing and difference anymore, but 'by computational processes running on a vast computational mechanism underlying all of physical reality' (Hayles, 2005, p. 3). She seconds Kirschenbaum's call for software studies as an ideological critique of code. Computer code should not only be a matter of practical applications, but one of critical analysis of what Hayles (2005, p. 61) understands as an 'intermediation of human thought and machine intelligence'. An entanglement between humans and computers becomes of interest that is concerned with the patterns of the computer that conditions collaboration as much as with the ideas and values that are built into software. Studying materiality of software thus means to look under the hood at its logics and politics. Such an approach includes the political economy of software, its cultural significance, technical processes and practices, as well as the politics and ethics involved. Fuller (2008a, p. 2) makes a similar argument. His call is to not overfetishise software but to lay bare the structures that software is built in and built on top of. A like-minded Federica Frabetti (2015, p. xii) calls for a 'radical demystification' of new technologies through a demystification of software'. This approach follows her key argument that technology and culture cannot be analysed separately.

In the last few years, the research on software as an assemblage of code and society has picked. Rob Kitchin and Martin Dodge (2011), human

geographers by trade, have made examined how software produces space and how space becomes dependent on code. Benjamin Bratton (2015) draws from political philosophy and architectural theory to argue that computational methods and applications form a new megastructure that functions as a governing apparatus. Noah Wardrip-Fruin (2009) looks underneath digital media to find artificial intelligence techniques in computational processes that generate the desired "expressions" on the frontend. Jussi Parikka offers a genealogy of digital culture that combines theories from various fields, including a media archaeology that is based on Kittler's work (Parikka, 2007, 2015; Parikka & Sampson, 2009). He does so by opting for breaking media down to their most integral materiality instead of approaches such as political economy, discourse analysis, or others more commonly applied in the humanities. Similar to Hayles, Parikka offers a concern for how things are made and what they are made of, and how media cross the line between humans and non-humans.

What software studies thus provides for this study is a lens with a focus on materiality. The different approaches have in common to understand software as a constitutive element of culture. For this study, the material lens thus shall provide a focus on the underlying structures of collaborative production of LibreOffice. This means to find out how collaboration is structured, which practices are activated and for what reason, and which political and ethical ideas of f/oss influence a software product. In addition, an entanglement between humans and non-humans is emphasised that allows to apprehend collaboration as an assemblage of materiality and sociality whereby the two spheres need to be understood as one. When 'we peel back the deepest layer of materiality, we find people and practices underneath' (Coleman & Brunton, 2014). Vice versa, if we peel away the layer of the social, we find materiality.

The computer, or software for that matter, appears to be a protean technology (Mahoney, 2011). It is an open technology that is realised through practices. Given this openness, aesthetic or technical characteristics do not have sufficient analytical substance to understand media and empirical and theoretical approaches that are directed at the practices are needed (Gießmann, 2018). Technology does not determine practices, but it offers a range of practices that can be realised and

activated. Culture is not an object produced by technology but a project – and humans become projects instead of subjects (Flusser, 1998). Technologies are not portrayals of reality, but they are possibilities to realise ideas and values. As such, their materiality as computations, which is their history and their social entanglements, needs to be addressed at the same time as their flexibility and possibilities in order to develop practices. Based on the above arguments, software is infrastructure for our culture as much it produces meaning.

This study is thus concerned with how practices form and how they become stable, stay dynamic or become obsolete. Practices can be individual actions, but they are produced and reproduced collaboratively. If they are not practised anymore, they cannot be recalled anymore. F/oss thus becomes a set of sociotechnical practices that is realised through collaboration. Then it becomes stable and concrete to serve as an infrastructure for collaboration. Only then can it produce and reproduce conditions of possibility (Kornberger et al., 2019) for collaboration.

3.1.2. Actor-network theory

The sociotechnical character of practices is emphasised by actor-network theory (ANT). ANT explores the interconnections between different actions including the technology that is used. Actions can be found in any structure or organisation, and while they are multiple networks of heterogeneous actors in more or less stable associations (Callon, 1991), they share interactions, and they also share interactions for sharing. In ANT, the terms actions or interaction are frequently used to describe the actors in a network. I will rely more on the characteristics that are used in ANT to describe the linkage between human and non-human actors than on the concrete terminology. What actions and interactions within ANT describe are relations between actors that are dynamic and changing, yet they became stable associations. The processes through which associations became stable are called translations. Callon (1991, p. 145) notes that a 'successful process of translation thus generates a shared space, equivalence and commensurability'. If the translation works, actors can communicate with each other. If the translation fails, the actors 'disalign', 'they reconfigure themselves in separate spaces with no common measure' (Callon, 1991, p.

145). Despite the fact that ANT does not use the term practices, what this line of thought offers is a certain dynamic movement between actors, that looks for stability but offers exit routes that allow change and a realignment in a specific space.

ANT's framework in combination with the infrastructural approach is especially beneficial for studying the practices of free software where attention should be carefully directed at how practices emerge, or how they are accepted, how they are declined, negotiated or modified. In addition to objects, tools, and devices, there is also a need to grasp the institutional surroundings like governance and policy, and the bundle of interrelated practices in between the different groups that are in the LibreOffice project.

In order to achieve this, the following route map is suggested: Firstly, the researcher should adopt the symmetrical tenet of ANT and not decide in advance what is related and important, whether an interaction is micro or macro, big or small, social or technical. Scale is the actor's own achievement (Latour, 2007, p. 185). ANT is all boundary without an inside and outside, the only important question is whether or not a connection is established between two elements (Latour, 1996, p. 6). The task is then to carefully follow the practices by looking for regularities in connections that can be observed. If connections emerge, clusters of connections can be identified. Latour (2007, p. 130) has made clear that if an element in a network makes no difference, it is not an actor. This means interlinking collaborative practices to their material and social context: 'Society, organizations, agents and machines are all effects generated in patterned networks of diverse (not simply human) materials' (Law, 1991, p. 380). Digital media practices thus, emerge as associations that highlight the interlacing between technology and sociality. Practices materialise in software in the form of a double bind: On the one hand, software can be conceptualised as something that is constantly in the making (Berlant, 2016; Velkova, 2017). On the other hand, software is a form of infrastructure that is needed to develop and share the very same practices.

This means that software is required as an infrastructure for collaborative practices, but it is also produced through and maintained collaboratively. Collaborative practices are the result of an interplay between a social, institutional and technical realm which give cultural meaning

to media by formulating, activating and framing a technology that is structurally open. Digital media need to be collaboratively practised in order to be become concrete, to produce and reproduce conditions of possibility (Kornberger et al., 2019). Given this openness, aesthetic or technical characteristics do not have sufficient analytical substance to understand media and empirical and theoretical approaches that are directed at the practices are needed (Gießmann, 2018). However, this means not studying software in terms of user practices as human-computer-interaction or design theory would suggest. Rather, the practices that are oriented towards software and the role of a software in ordering the collaborative production practices in the social world become the focus of such an approach. Software is negotiated as a set of micro-practices that surround and support collaborative practices and which exists of the same practices.

3.1.2. a) Software as a hybrid

From the perspective of media studies, which has historically oscillated between sociodeterministic and technodeterministic approaches, Actor Network Theory (ANT) delivers a fruitful attempt to bridge this debate between society and technology. One of ANT's main assumptions is that the entanglement of social and technical elements in a network is so deep that attempts to dissect the conglomerate and isolate the two sides are artificial: 'There exists no relation between "the material" and the "the social world" because it is this very division which is complete artefact.' (Latour, 2007, pp. 75-76) Thus, humans are not in an autonomous position who can manipulate passive objects. Software, and f/oss in particular, is social and technological at the same time. The social and technological are locked in an interplay that needs to be described as the process of a formation of a collaborative culture. For ANT it is not important if an actor is human or non-human, rather the action is highlighted. Actors are 'entities that do things' (Latour, 1992, p. 241):

> The distinction between humans and non-humans, embodied or disembodied skills, impersonation or 'machination', are less interesting than the complete chain along which competences and actions are distributed. (Latour, 1992a, p.243)

What is of interest for ANT is not whether an actor is social or technological. The action itself is of the highest importance. The difference that is crucial for ANT is, who or what causes an action. An actor has

specific competences and 'acts or shifts actions' (Latour & Akrich, 1992, p. 259). Latour proposes a symmetry between human and non-human actors in regard to actions. To highlight that analytical symmetry, in ANT the term actant is preferred over actor. All actants are afforded same value and potential for action. However, giving agency to non-humans does not mean to attribute technology with the power to act independently or to determine social constellations.

By including non-human elements as active components in the formation of collectives, the social is reassembled (Latour, 2007). Not only Latour but also other advocates of Actor-Network-Theory, or representatives of the "Social construction of technology", along with assemblage theory thinkers such as DeLanda or Deleuze as well as thinkers of new materialism such as Donna Haraway or Kathrin Hayles have received criticism for supposed post- or anti-humanist thinking (Chagani, 2014; Chandler, 2013). Chris Gregory believes that the inclusion of non-humans results in 'a theory of value that attributes agency to things' (Gregory, 2014, p. 45) and that 'Latour is a theological thinker who has devoted his life to attacking humanist thought' (Gregory, 2014, p. 49). Such criticism however is due to having overlooked that ANT uses a flat ontology only to assign the potential to make associations to everything that contributes to collectivity. The aim is not necessarily to leave humanism behind but 'to move beyond deterministic models that trace organizational phenomena back to powerful individuals, social structures, hegemonic discourses or technological effects' (Whittle & Spicer, 2008, p. 616). ANT is an invitation to focus on the associations between the actors that emerge through connections in a network rather than on structures that impose power relations or hierarchies. Human and non-human actors associate with each other in a "hybrid collectif" (Callon & Law, 1997) and ANT emphasises the role that non-humans play in the scenario that is studied. The conception of agency in ANT is not linked to intentionality or free-will. Instead, the conception 'is minimal because it catches every entity that makes or promotes a difference in another entity or in a network' (Sayes, 2014, p. 141). If an actor has agency can be tested with two questions. If the answer to both is yes, then it does not matter if the actor is human and non-human: 'Does it make a difference in the course of some other agent's action or not? Is there some trial that allows someone to detect this difference?' (Latour, 2007, p. 71).

For the purpose of this study, collaborative symmetry does not mean that software has agency that is similar to those of humans. Rather a symmetrical view allows to analyse the human agency in collaborations better by aligning it with a pool of characteristics that is often reserved for non-human elements. Thus, for this study, humans are confronted with questions such as: What role do they play in the collaborative production of software? How do they mediate collaboration? The aim is then to describe the associations between human and non-humans in an assemblage. Vice versa, in collaborative connections, ANT's perspective allows to analyse the role of software in collaborative practices, which things it does, and how actions are taken to make, keep, and maintain it. As established before, the difference between doings (or actions) and practices is the regularity in which they are activated and performed.

The connection between focusing on practices and ANT will show how practices are (per)formed, how they are established, and how they are negotiated and how they cease to be activated. I will point towards several concepts that are characteristic for ANT (translation, association, intermediaries) that offer to bring forward the negotiations, moments of flexibility as well as the necessary elements for stability. This lense shall add to a refined understanding of collaborative practices that I deem to be characteristic for digital media. One major offering by ANT to understand digital media is to include non-human elements as actors in the formation of practices. Instead of a dialectic relation between technology and society that would underline their binary adversarial relationship, Latour proposes to understand objects and society as being entangled in a co-production: 'Is not society built literally – not metaphorically – of gods, machines, sciences, arts and styles?' (Latour, 1993, p. 54). Rather than lines that connect humans with technology to produce culture, human and non-human conflate into hybrids. He argues we should not emphasise too much on dialectics, as dialectics foreground the existing dichotomies; instead, he proposes to focus on quasi-objects.

> Dialectics literally beats around the bush. Quasi-objects are in between and below the two poles (...). Quasi-objects are much more social, much more fabricated, much more collective than the 'hard' parts of nature, but they are in no way the arbitrary receptacles of a full-fledged society. On the other hand they are much more real, nonhuman and

> objective than those shapeless screens on which society - for unknown reasons - needed to be 'projected'. (Latour, 1993, p. 55)

With quasi-objects, ANT provides a concept for collectivity that includes technology as an active part which conditions and frames the possible actions. Through this symmetry, technology however loses its status as an object that is manipulated by a subject. It is not a stable entity anymore but malleable. For ANT, things result out of connections. Through the entanglements between actants both are informed and keep their status and keep their character. As soon as the connection breaks up, it can be changed through a new connection. Through this perspective software can be looked at as temporary sociotechnical collective, as something that needs connections to come into being. It is not necessarily stable though, and can get into different modes of existence.

3.1.2. b) Heterogeneous stability

Stability of a network 'does not come from concentration, purity and unity, but from dissemination, heterogeneity and the careful plaiting of weak ties' (Latour, 1996, p. 3). A network needs to be heterogeneous to able to start and make connections Latour argues. However, some stability needs to be reached. Callon introduced two terms for stability. One is convergence: 'Convergence measures the extent to which the process of translation and its circulation of intermediaries leads to agreement.' (Callon, 1991, p. 144) Practices can offer that kind of stability. For collaborative practices the translation of a governance, the technological elements, and the social formation have to be accepted to such an extent that it 'becomes self-evident, a matter on which everyone can agree' (Callon, 1991, p. 145). Such as successful convergence 'generates a shared space, equivalence and commensurability' (Callon, 1991, p. 145).

The heterogeneity must be given to start connections and to maintain them, negotiations, discussions, and adaptions are needed. Callon (1986) calls these acts translations. Only what is translated can be become part of the collective. Thus, in its violence translations are very similar to power relations. They force categories, conditions and structures over another agent. When a network is stabilised it creates conventions or '*c*o-ordination or translation regimes' for these translations

(Callon, 1991, p. 147). I would call these regimes of practices. These practices ensure integrity in a collective in order to build a platform for exchange. Collaborative practices need to be shared to be maintained and new members need to be taught (or translated) to keep the network working. Learning these practices is similar to learning other skills, and the transfer of skills is particularly important in a techno-economic network. 'No description of skills is possible unless the networks of humans, texts and machines within they are expressed and put to work are constituted.' (Callon, 1991, p. 138)

Given all the strategies to find stability in a network, it is also important to understand how thing change. Dynamic in a network is guaranteed because established translations are not immutable: 'All translations, however apparently secure, are in principle reversible', Callon (1991, p. 150) explains. If a practice is established for years, or a person has a high status because of their skill or merit for continuously assisting the project, or a software program cannot be used because its license is proprietary because the generally accepted value system in LibreOffice is against: everything is reversible. Beliefs can change, ethical standards get renewed or not, technical standards are updated.

As described, networks consist of actors, actants and their associations. Without an actor, there is no network, and vice versa, without network an actor cannot act. What is unclear is how a network starts, or in the words of this study: How does collaboration emerge? ANT gives various reasons for a network to emerge. It could be that scallops in a bay in the Normandie lose their coral (Callon, 1986a), or a new invention such as a bulk shipping container for liquids (Callon & Law, 1992), or a person transferring knowledge to others (Callon, 1991). Also, small changes in a network can lead to change as well as no changes can result in translations going stale and the network losing its dynamic. At the beginning of a network always stands an intermediary who (or which) brings other actors and actants together. As illustrated by the examples above an intermediary can be human or non-human, they 'describe their networks in the literary sense of the term. And they compose them by giving them form' (Callon, 1991. p. 135). To look for an intermediary that stands for the start of a network is thus a point of departure for an analysis.

Instead of starting from an apriori which either assumes that technology has an effect on society, or that society moulds technology, the emphasis is put on the reciprocity between human actors and non-human actants. Practices are formed and stabilised through this reciprocity, and are material and social in nature. Practices are formed and stabilised through this reciprocity, with material and social aspects. F/oss development – which is considered as a media practice – is thus not a material practice but instead a set of socio-material, or socio-technical practices. Practices are formed with qualities, categories, and concepts from both sides. These characteristics become only active parts through the involvement of both sides. Technical categories become active through social adaption while societal relations form on the same material basis. The result is a techno-economic, Callon (1991) calls it. Its characteristic as "a coordinated set of heterogeneous actors which interact more or less successfully to develop, produce, distribute, and diffuse methods for generating goods and services" (Callon, 1991, p. 133) is a suitable concept to study f/oss.

3.1.3. Infrastructuring

I have tried to show how ANT proposes a 'method to describe the deployment of associations' (Latour, 1996, p. 9). I think of it as a useful approach to understand the dynamics of practices in a free and open source software development which can be understood as a network of associations among different individuals and groups: developers who write code, translators who localise the software product, lawyers who help in wording manifestos or write legal texts, the software that runs the code, a free and open source software that can be copied and shared, a bug tracking software that orders the so-called tickets, etc. Any outcome, any result is based on a process of translation which transforms inputs into outputs. Software does not necessarily care if it is written properly yet it requires a specific process to function properly. Either way, it translates an input into an output.

Callon based translations on 'three principles, those of agnosticism (impartiality between actors engaged in controversy), generalised symmetry (the commitment to explain conflicting viewpoints in the same terms) and free association (the abandonment of all a priori distinctions between the natural and the social) (Callon, 1986, p. 196). Latour

eschews any recourse to a priori structures: 'Instead of predicting how an actor should behave and which associations are allowed a priori, ANT makes no assumption at all and, in order to remain uncommitted, it needs to set its instrument by insisting on infinite pliability and absolute freedom' (Latour, 1996, p. 374).

This conceptualisation makes it possible to start this investigation at a certain point of time without the need to restructure every association that happened before. I can concentrate on the practices that are activated at LibreOffice. Indeed, f/oss has a history (see chapter 1: Copyright and licensing information). Borrowing from ANT does not mean to disregard this history. But the specific interests and beliefs, technical standards, social norms or organisational procedures do not form an a priori that explains the practices which I studied. This does not mean to disregard politics or social norms. Instead of presupposing them as an apriori, they are the result of translations. They come to the surface through the practices and through the reflections of the collaborators on their practices. The character of the elements involved in terms of their political and economic value, and their potential to change the mode of existence is an important factor of this study. Power relations and politics are not excluded from an ANT perspective. Indeed, they are studied as being expressed by technology as well, similar to Langdon Winner's (1980) argument that artefacts exert power:

> The issues that divide or unite people in society are settled not only in the institutions and practices of politics proper, but also, and less obviously, in tangible arrangements of steel and concrete, wires and transistors, nuts and bolts. (Winner, 1980, p. 128)

Winner uses the example of low bridges in New York designed, he argues, for the purpose of excluding public buses from parkways that led to beach areas, because the buses typically carried less wealthy people and racial minorities. Based on this, Winner suggests that artefacts do have politics, that artefacts exert power. This example led to severe criticism. Joerges (1999) argues that Winner did not fully investigate the facts, stating that bridge height was typically kept low by regulation, and buses, trucks and commercial vehicles weren't allowed on parkways at that time anyway. He concludes that 'the power represented in

built and other technical devices is not to be found in the formal attributes of these things themselves' (Joerges, 1999, p. 19). His argument is that the form of an artefact does not have political power; artefacts are subservient to the systems in place.

Winner points out however that certain artefacts do require certain political and social conditions for them to emerge. He observes that, 'the things we call 'technologies' are ways of building order in our world' (Winner, 1980, p. 127). Consciously or not, deliberately or inadvertently, societies choose structures for technologies that influence how people are going to work, communicate, travel, consume, and so forth over a very long time. In the processes by which structuring decisions are made, different people are differently situated and possess unequal degrees of power as well as unequal levels of awareness.

Important to conclude is that both Winner and Joerges agree that materiality takes part in communication – not as a neutral platform that allows certain forms of communication and complicates or impedes others but as a formative agent. Joerges does not deny that artefacts have politics but he criticises the deterministic assumptions that Winner argues for. He denies a causal relation between technology and social practices. Technologies get their authority from the outside, he argues. Important for Joerges is authorisation, the legitimate representation. This authorisation gives shape to the effects that may come. Authorisation is in the discourse, people (social forms), and through an institution. It exists within the acceptation of power of the artefact, following a presentation through people whereas in Winner's theory, social order and disorder are presented as planned. A social process is given a definite form by a technology.

To avoid this dichotomy represented by Winner and Joerges, the concepts of infrastructures and infrastructuring present a solution to move forward and to capture the dynamic between humans and non-human elements in collaboration. I have established that this study is concerned with the recursiveness of f/oss. Not only does software facilitate collaborations; they are also produced by collaboration. The concept of infrastructure opens another entry point to study the collaborative production of software. I propose to look at software as infrastructure. It shares the quality that Bowker and Star (1999) highlighted about infra-

structure: It is embedded by being sunk into the metamedium computer as it structures social arrangements, and thereby it dimensions and conditions collaboration. More often than not, software is an invisible layer (Chun, 2005; Galloway, 2006), only when it is not working anymore, in the moment of malfunction, its existence becomes apparent, it becomes visible upon breakdown (Bowker & Star, 1999a). Infrastructures might be considered neutral but they matter just because they are matter as Herzogenrath (2015) puts it, echoing Kittler's (1986) famous dictum 'media determine our situation'.

Besides embeddedness and invisibility, Star (1999, pp. 381-382) highlights other characteristics that make understandable that infrastructures are part of a socio-technical assemblage. Conventions of practice stabilise infrastructure, past conventions are often built into infrastructure without updating them. Thus, an a priori is built into them.

Infrastructure has become a genuine socio-technical key concept that extrapolates the conditions for collaboration but also the collaborative materiality of digital media (Johnson, 2013). Thus, free software is also infrastructure which allows and mediates infrastructuring. Moving from the noun infrastructure to the verb, sets infrastructures in motion. Materiality is then a 'historically given micro-network of technologies and techniques' (Siegert, 2013, p. 58). These "cultural techniques" as Siegert calls them generate media through practices and processes, and as such they allow a reflection on the different formats and diverse materials that are integrated into the morphology of media. It extends the thinking about infrastructure beyond stability and immobilised material towards infrastructures in the making. The task then is to find and make comprehensible the invisible negotiations that are producing the infrastructure (Sandvig, 2013, p. 89).

3.1.4. Boundary object

The "boundary object" (Star & Griesemer, 1989) is particularly helpful to understand infrastructure as disputed and negotiated, as durable and flexible at the same time. The boundary object (Star & Griesemer, 1989) is one of most widely used concepts from Science and Technology Studies. Boundary objects are those that are specified in a local application and used purposively, but at the same time are available in a wider circulation, without losing their identity.

> Boundary objects are objects which are both plastic enough to adapt to local needs and the constraints of the several parties employing them, yet robust enough to maintain a common identity across site. (Star & Griesemer, 1989, p. 393)

Boundary objects point towards practices and allow negotiations and differences to be part of collaborative practices. Star (2010) underlines that this concept is best used when exploring a group that actively works together, that is formally organised, and that is connected through a shared infrastructure. The boundary objects 'are the stuff of action' (Star, 2010, p. 603), they allow collaboration while being flexible. From such a perspective, the conceptualisation of collaboration as people working towards a shared goal (Shirky, 2009) can be broadened so that it becomes an 'ambivalent process constituted by a set of paradoxical relationships between co-producers who affect each other' (Schneider, 2006). Such a perspective helps to capture collaboration as an ambivalent process and to hold a debate about it 'in terms of differences, negotiations and connections' (Lovink, 2011. p.12).

From this sociotechnical perspective, collaboration must not only be studied in moments of stabilisation and consensus. Infrastructuring shows that collaboration without consensus (cf. Star, 1993) in combination with boundary objects holds organisations and institutions running and in some cases even originates them - only to be endangered by the claim for consensus. Hence, it requires collaboration and communication. I propose that this additional layer provides a model to understand software as an infrastructure that can be studied along the communication amongst the contributors. As such it highlights software as the material infrastructure for its production as well as a media object to communicate upon.

3.1.5. Interim summary

This module has provided some important insights. Not only is this study concerned with stability and the strategies to let collaboration emerge and stabilise it; it also focuses on conflicts, negotiations, rupture, and things breaking down. Research question 1 in particular is concerned with this problem. Chapter 5 will show how a community can use specific sociotechnical practices when in conflict. Chapter 6 and 7 focus on the negotiations and strategies to keep the project running.

In addition, ANT and STS deliver a set of concepts that allow to include the materiality of software into the analysis of collaboration. Concepts borrowed from ANT and STS offer ways to analyse software as an assemblage of code and people by considering the negotiations and fluidity of infrastructure, which is usually considered to be stable. Chapter 1, 2 and 3 show how this assemblage is running in different phases of the project.

By combining these two approaches, collaborative practices take shape between moments of consensus and conflict, between stabilisation and rupture, from emergence to break, and in between social formations and technical structures which are mutually dependent upon each other. Research question 2.2 is concerned with the dynamic of practices. This dynamism can be contrasted with a long durée of infrastructure, which in the case of f/oss is belief system that is also in between as it is negotiates free software and open source software and their respective ideas on innovation and commerce. To use the jargon of ANT: there is a translation between discourse and practices in f/oss. How discourse and practices are interlocked is asked by research question 4, chapter 8 will focus on the politics involved in the project. Chapter 1 has sketched the discourse that f/oss is embedded in, and has provided insights into how to conceptualise collaboration as practices.

3.2. Ordering practices

Apart from the sociotechnical approach to understand collaborative practices, I have shown in chapter that a line of research is concerned with organising and coordinating in collaborative projects, and studies that have focused on f/oss projects in particular. Coming back to that line of thought, the rest of this chapter is concerned with theories that help to understand how the practices in a distributed collaborative project such as Libreoffice are organised and coordinated. While ANT gives several useful insights into how practices can be established between heterogeneous actors, the next part offers possible guidelines, and rules that can be used to understand organising and coordinating practices in commons based project such as LibreOffice.

3.2.1. Governance

I have referred to Ostrom's work above. Her studies of the management of common goods starts from the assumption that individuals need others to share knowledge structures: 'I presume that individuals have very similar limited capabilities to reason and figure out the structure of complex environments.' (Ostrom, 1990, p. 25) These structures are needed to make sense of the world, and through this shared understanding (a shared vocabulary or some other common ground) of the world, cooperation can succeed. Ostrom then explains that external mechanism are needed to transform these shard structures into society: Institutions must create these structures and order the environment (Crawford & Ostrom, 1995). She underlines that in order to create sustainable economic systems, governance is needed. Thus, institutions must be judged on their capacity to channel potentially self-interested motivations in ways that generate mutually beneficial outcomes (Ostrom, 1990). Cooperation needs to be governed but instead of a centralised regulation, Ostrom suggests that cooperative institutions that are organised and governed by the resource users themselves can solve the tragedy of the commons.

Ostrom (2009) delineates ten design principles that affect the likelihood of people to govern a common resource by self-organisation. These include:

1. Moderate size of a system (too big means high costs, too small means not enough products);

2. The productivity of system (self-organisation is less likely if a resource is either over abundant or already exhausted);

3. Predictability of system dynamics (users need to estimate what would happen if they were to establish particular harvesting rules or no-entry territories);

4. Resource unit mobility (mobility implies high costs because it requires observing);

5. The number of users (larger groups can mobilise more resources but require higher transaction costs);

6. Leadership (local leaders with high reputation and skills facilitate self-organisation);

7. Shared norms, moral and ethical standards;

8. Knowledge of the system (self-organisation rests on users sharing knowledge);

9. Importance of resource to users (self-organisation more likely if users' livelihood depends on resource)

10. Collective choice rules (full autonomy to craft and enforce rules).

The size of the community to sustain it is an interesting topic for f/oss. On the one hand, the premise of open source software is to attract as many contributors as possible according to the assumption that 'given enough eyeballs, all bugs are shallow' (Raymond, 1999, p. 30). On the other hand, this raises questions how the coordination of technical practices and community management is affected by a growing number of participants.

Ostrom's understanding of governing the commons is an example of a social contract that builds on the human capacity to cooperate. Not only has her work shown that commons can be the result of the representation of a collective will. More than economic relations, common resources become an active set of cooperative practices that result in an ongoing active process of 'communing' (Linebaugh, 2008). The freedom is to be found among people who collaborate to build and maintain common goods.

3.2.2. Freedom

To relate these ideas to digital culture and collaboration, the concept of free cooperation (Spehr, 2007) provides a model for collaboration as an active process. He understands free cooperation as opposed to forced cooperation, which has three defining features.

First, their hierarchies, rules, and protocols are neither negotiable nor flexible. Given that a company is a cooperation of team members to solve business problems and produce profits, an employee must first go their boss if they have problems with a co-worker. Then their boss either deals with the matter or tells the next person "up the ladder" to become involved.

The second defining feature of a forced cooperation is that they never stop even if a contributor encounters a problem, solved or unsolved. While the employee of the example above tries to have their problem resolved, the activities in the company do not stop. Even if the system suffers interruptions, it keeps running.

And third, a worker faces serious repercussions if they break the rules or leave the cooperation. Their employment may be terminated, they may be pushed to quit, or they may be excluded from social activities. Without a chance of avoiding these consequences, workers need to keep doing their job. A forced cooperation cannot be left without considerable consequences for the abandoner.

Free cooperation, in contrast, offers an alternative triad of features. First, all rules can be questioned by everybody. The rules for how to handle a problem of a team member would be negotiable. Such rules would have been designed with the equal input of all members of the cooperation. The power hierarchies would be very different from a forced cooperation. They would give power to every single member to renegotiate the rules.

Secondly, all participants can quit, limit or condition their collaborative effort, and the price for leaving a project is equal and bearable for all parties. If a team member waits for the solution of a problem, they can decide not to work. Nobody but the individual themselves decides when they contribute.

The third feature of a free cooperation is that co-operators can leave the group without facing major consequences themselves nor for the group. Abandoning a project will not cause any more or any less disruption than if any other team member left. Nobody is too important to leave and, conversely, no one person is considered useless to the group.

The difference between forced and free cooperation is important because we are all constantly engaged in cooperation, argues Spehr. To speak of cooperation as exceptional is misleading. Rather, the type of cooperation is the decisive factor. Spehr's concept of free cooperation offers an understanding for new forms of collaborative practices that emerge with digital media as it gives ethical and political benchmarks.

On a structural level that can be used for an analysis of f/oss, the theories and concepts on organising collaboration show that it does not emerge automatically – if at all – when a group of people decides to produce and govern a common good. Specific conditions must be met and a set of rules established to build the right environment that allows a group to cooperate successfully. While some problems that common goods usually face are obsolete in the production of information in a distributed network such as the transaction costs, the scarcity of the good, or mobility, many other points that Ostrom raised can be used for studying LibreOffice.

3.2.3. Interim summary

This module has provided a problematisation of the ordering structure for collaborative practices. One theme which has been raised is the size of the project. Research questions 1, 2, 3, and 4 are partly concerned with this problem. The second empirical chapter (chapter 5) discusses the problem of the community's size and boundaries concerning governance. The third empirical chapter (chapter 7) includes the problem regarding the coordination of the project.

Mobilisation for f/oss refers to how a group coordinates to solve a problem (see research question 3). The question of leadership involves how people gain reputation and the role those technical skills play (see research questions 2 and 3). Leadership is problematised with research question 4. Chapter 6 deals with decentralisation, self-organisation and openness and a possible opposition to governance and decision-making by an authority. Leadership in production will be discussed in the third empirical chapter (chapter 5), in combination with the question about the knowledge of the system.

Shared norms and ethical standards are an interesting topic for f/oss (see research question 4). As described in chapter 1 *Copyright and licensing information – Free, open source, free and open source*, free and open source software is the result of two different ideas about how innovation and collaboration should be structured. Political ideas and how they influence the project are discussed in the fourth empirical chapter (chapter 8).

The importance of resources to users is reflected in the setup of LibreOffice. Volunteers, employees and the contributions of companies, together with the different political ideas and opposing ideas of value that are assembled within the project offer a pluralistic intellectual process that renders a special touch to this problem. Research questions 1-4 are motivated by it. This theme will be present throughout the empirical part.

3.3. Conclusion

As a result of the discussion so far, practices can be defined along five characteristics: First, practices emerge when actions are regular (Couldry, 2012, p. 33). To become collective actions, they need to be copied and shared. This implies that newcomers need to be taught and introduced to the practices in order to fit in.

Second, they are social in the ANT sense. They need to be recognised and evaluated by another actant (human or non-human) in order to establish associations. Only if an association is built can they become practices.

Third, practices are sociotechnical. Traditionally, practices can only be enacted by humans while being closely linked with non-human elements (Swidler, 2001). From a symmetric perspective which is offered by ANT, non-human elements also embody practices: Free software is the result of this sociotechnicality as it is 'composed of, surrounded by, and immersed in or consuming physical matter and non-human dimensions' (Küpers, 2016, p. 2).

Fourth, practices are ordered. A structure, as Crawford and Ostrom (1995) underline, creates the necessary order on which practices can develop. While a shared structure allows translations to happen, an ordering system is not pre-determined. Rather it emerges out of practices, adapting to the needs of collaborators to associate. The composition of such an ordering system is thus open for negotiation.

On the basis of these characteristics, software practices develop as a combination of doings, objects, and ordering. Collaborative practices emerge at the crossroads of these elements. These three elements are not related to each other in a hierarchical order, nor in a harmonious

symbiosis. Rather, they allow tensions between them, they allow to enable each other as much as they can cancel each other out. Looking at collaborative practices like this the structure becomes circular rather than dialectic: Practices are constantly re-made as doings with objects that need to be put in order. From an anthropological point of view such an approach of practice theory 'seeks to explain the genesis, reproduction, and change of form and meaning of a given social/cultural whole' (Ortner, 2006, p. 149) in a micro-network of technologies and techniques.

4. Operations manual (Methodology & methods)

This chapter explains the methodological underpinnings for an exploration of LibreOffice's collaborative practices. It will also give an insight into the methods that I have deployed to focus on the collaborative practices and their role in this specific f/oss project. The theoretical chapter has already given some insights into this study's methodological premises. Collaboration has been theorised as a set of practices that can be activated through an assemblage of rules and structures together with a specific know-how in connection with a technological infrastructure. This chapter explains how to study practices as sociotechnical formations. Special consideration is given to the character of collaborative practices. They are not locked in a stable state but they are constantly shaped and changed. Therefore, a social constructivist perspective seems to be an appropriate ontological perspective for this study, as it undergirds the emphasis on the dynamics that bring software into being. However, a social constructivist perspective creates tension with the sociotechnicality that ANT proposes. I will attempt to resolve these tensions in this chapter before moving on to a discussion on the methods that I have chosen to deploy for this ethnography.

While there is a wide range of constructivist thinking depending on disciplines and nuances, the general idea is to underline that knowledge is the result of a cultural process: '[M]eanings are constructed by human beings as they engage with the world they are interpreting.' (Crotty, 1998, p. 43) It is an alternative to an understanding of the world as an immutable object that can be sorted and mapped out along neutral and unwavering categories. Instead, the concepts and frameworks that describe reality can be fluid, emergent and disappearing. Reality becomes the result of a social process. As a consequence, the emphasis of research shifts from the aim to discover reality and more accurate ways to study the world with an interest in the social dynamics of the construction of reality.

A social constructivist approach offers solid links to explore the emergence and development of practices that are performed in the chosen f/oss project to allow collaboration in a pluralistic setting. The social in social constructivism refers to the idea that reality is not just out there, ready-made for the scientist to analyse it. But it also means that the

construction is social, not technical, or natural, or individual. This emphasis on the social character of the construction complicates the inclusion of software as an actively contributing agent into the analysis. And even though references to the technical affordances and human creative usage are made, it is impossible to separate the material from the mental in software production which is the result of the linkage of mental and technical categories. The inclusion of technical categories has ontological and epistemological consequences.

4.1. Ontology

Crotty's (1998) argument is that ontology and epistemology are interdependent. Thus, 'to talk about the construction of meaning is to talk of the construction of a meaningful reality' (p. 10). What we can know of the world (epistemology) is created by a construction and therefore all there is in the world is a construction. Instead of an ontology and epistemology, Crotty proposes social constructivism as a worldview or what others have called a paradigm (Guba & Lincoln, 2018). It guides the entirety of the research process, starting from ontological and epistemological premises down to the choice and application of methods.

This perspective entails that reality depends on human thought. Latour (1999) points out that it relies on the idea that a constructor acts on a neutral material to manifest their ideas. He argues that it ignores the involvement of non-human actants in this process. Media further problematise the problem of the subject-object relation. Software underlines how media are visible tools while at the same time they act as obfuscated infrastructure. The non-human actants are ontological hybrids taking part in the construction of reality as an idea of the world as much as they are involved in the process of exploring and maintaining it. The discussion of whether technology has agency or not can sometimes be pedantic. I lean towards the definition that Latour has brought forward. This is a way to move away from an anthropological argument towards a view that emphasises reality as in the making, as a process of actants. He (Latour, 1996, p. 370) defines actants a 'something that acts or to which activity is granted by others. It implies no special motivation of human individual actors, nor of humans in general. An actant can be anything, provided it is granted to be the source of an action'.

This proposal means that there is no gap between a subject and the world that needs to be bridged with language and categories. The construction itself is real and it is the result of connections between humans and technologies. To separate materiality and ideas is a 'simplification of ontology that has led to the enormous complication of epistemology' as Viveiros de Castro (2012, p. 152) concludes. Objects are not silent and pacified so that subjects can fill the construction with ideas. He argues that 'at the heart of the matter, there is no stuff; only form, only relation' (Viveiros de Castro, 2004, p. 484).

The world that is the result of relation suffers from a lack of distinction. There are no neutral objects to study scientifically. The relatedness of objects leads to the loss of their aura as Latour points out. By pointing towards the relatedness of objects:

> (. . .) we always appear to weaken them, not to strengthen their claim to reality. (. . .) Why can we never discover the same stubbornness, the same solid realism by bringing out the obviously webby, "thingy" qualities of matters of concern? (Latour, 2007, p. 236).

4.2. Epistemology

How is this ontological position connected with an epistemology? Cresswell (2007) points out that social constructivism rests on the assumption that the world can be interpreted by individuals. They construct their own meaning within a certain paradigm. I would argue that his worldview is more akin to a radical constructivism. A social constructivism always starts from a social process. Meaning is constructed together in a web of institutions, practices, and discourses. Humans habitualise and typify the world through processes, habits and categories as Berger and Luckmann (1990) point out. Based on these exchanges, institutions are built that determine our ideas and actions.

The same argument made for ontology is also used here for the epistemological considerations. Technologies are not neutral tools that are used to construct meaning. They take part in the process of creating knowledge of the world. The togetherness includes non-human actants. Actants can be both human and non-human, and it would then seem

strange to claim that the latter do not exist or can be reduced to constructions. Moreover, reality is not neutral to operations on it. It resists but it is fluid enough to allow construction work that continues all the time.

This does not mean that there is construction and reality, similar to a back end and front end where an untouchable reality lingers in the background and humans only deal with a chimera that is portrayed at the front. What we can know of the world is thus how it is constructed. The leading questions are then: How is it constructed? What elements and ideas constitute the planning process? What is the history of the construction? These question rest on premises that come from social constructivism and ANT. The constructions are as much the result of an interdependent relation of subjects and objects and we can only know of them through this relatedness. Media became an important factor in this cultural web. They offer categories, points of view, patterns for practices. As ontological hybrids they are part of the world as much as they take part in knowing the world.

4.3. Research process design

Having discussed the methodological problems of a study that rests on the relatedness of technology and humans to form practices and the resulting tension with a constructivist worldview, the question then is how to study these practices? One suggestion would be to borrow from action research or participatory design approaches. Objects of study become co-researchers and the researcher himself becomes immersed into the group that is studied. Given the multifaceted structure of LibreOffice of various governance levels, organisational groups, local communities in all continents and the setup of the production, based on individual teams that specialise on coding, translating, or design, the study objects scale made it impossible to be everywhere or to take part in all processes.

Instead, I have chosen to write an ethnography consisting of observations and interviews as the main methods. This has allowed me to participate in the studied practices to a certain extent but it still allows to separate (at least artificially) to separate the studied practices from the research process regarding my own position as a researcher. Getting into the action was still part of a research process that allowed me go

into the field. Following the actors (Latour, 2007, p. 68) does not mean to follow people but to follow associations (Latour et al., 2012). Associations are those relations between actants that hold together assemblages. I followed these associations by following practices. The research process was driven by the theoretical notion of collaboration as a set of practices. The ontological and epistemological conflation of humans and non-humans implicates not to follow just people but a web of relations.

The research process was designed from a constructivist perspective, but with adaptions to make it useful for a study of practices. Cresswell (2014, p. 36) defines constructivist research as defined by four elements: Understanding of people, multiple participant meanings, social and historical construction, and theory generation. The research process here is defined by understanding of practices, multiple participant meanings, socio-technical and historical construction, and theory generation.
Understanding practices meant to understand the role that software plays for the formation of the socio-technical practices. The interplay between software and people and how it lets these practices emerge or leaves some practices forgotten was in the focus. The affordances of software as much as their realisation through practices was the guiding interest of this research process.

Multiple participant meanings refers to how the actors understand and interpret their practices. The interviews asked for the participants ideas about the practices, how they engage in them – or why not. The socio-technical aspect for this part is given by the conceptualisation of software as a boundary object (Star & Griesemer, 1989). Technology becomes realised through practices, yet the technology does not give fixed paths for action but it is flexible enough to give room for interpretation. The interviews thus reflect this process of adaption and fabricating practices.
The social-technical and historical construction concerns the matters that have been discussed as the ontological and epistemological premises for this study. Understanding practices presumes to understand them as a socio-technical construction. But I will also focus on LibreOffice as a historical construction in the sense that Callon (1991) argued that every network emerges out of an existing network. Hence, in

addition to interviews and observations, I have also used archives to read press reports, email conversations, and other documents. This has resulted in chapter 1 which gives the reader a short insight into the history of f/oss as well in the basis for chapter 4 which provides an overview of the history of LibreOffice. The aim with this approach is to understand to what extent practices are built on tradition or on situatedness in this specific assemblage that LibreOffice represents.

The aim of this research process is also constructivist in the sense that it is driven by theory generation. It is not change-oriented in terms of critical realism that it intends to show underlying mechanism that were hidden to the people who are part of the practices. It is also not real-world practice-oriented as an approach akin to pragmatism would be, even though many of the people I have engaged with during this process asked me to present my research to them in the form of guidelines and suggestions that could help to advance the project. I had to politely decline and explain that my research cannot offer them such insights.

4.4. Methods

Observations and interviews were chosen as the main methods along with work in the archive. An understanding of the collaborative practices shall be reached via a description of the practices as well as the reflections and sense-making of the participants via the interviews. With the aim to describe a culture, the research approach was akin to writing an ethnography. I spent extensive time in the field, amongst an unfamiliar culture that I want to portray. With the focus on the practices of specialists who build a piece of software together, the approach is similar to science studies where on strives for 'a detailed study of the daily activities of scientists in their natural habitat' (Latour & Woolgar, 1986, p. 274).

The three methods complemented each other and resulted in a research process that can be partitioned in three phases[5]. Phase 1 was characterised by familiarising myself with f/oss. I had no prior in-depth knowledge about hacker culture, nor could I rely on preestablished contacts in the f/oss scene. To start off, I travelled to FOSDEM in Brussels, the largest European free and open source software conference. I

5 For a more detailed overview please consult table 1 "Overview data collection" in the appendix.

conducted my first interviews there, with a focus on the reflections that people had on the f/oss scene and f/oss in general. I observed people at the conference and listened to talks and discussions. After my interest in f/oss as a research object had manifested, a second trip to FOSDEM in Brussels in 2017 was used to establish rapport with representatives of LibreOffice. In addition, I was able to speak to many people informally and to interview more people who had experience in the f/oss scene to get their view on it. Between the two trips I started to read extensively on free and open source, especially from a historic perspective to understand the underlayer of the current situation that presented itself. My fieldnotes show that I had detected the first clusters during these two fieldwork trips to FOSDEM: the question of openness, the entanglement between companies and community, and the role of politics were topics that have come up regularly during interviews and informal talks.

The second phase started with fieldwork at the LibreOffice conference in Rome in October 2017 and lasted until the end of spring 2018. This was a phase of intense fieldwork at the conference, FOSDEM in Brussels in February 2018 including the LibreOffice hackfest that was held there and the LibreOffice hackfest in Hamburg in April 2018. At these three occasions I observed the practices and interviewed participants. This was accompanied by many informal talks which pointed me towards other people to interview or material to read to learn about the history of LibreOffice. The search of and inspection of archived web documents concerning LibreOffice through the Wayback Machine was also part of this stage. In addition, I started to learn coding, a bit of C++ and some Python, to get a better grasp of the technical layer of the culture I studied. During this phase the cluster about coordination of institutional, social and technical practices was added to the research plan while clusters that were detected earlier solidified. The more the fieldwork advanced, the more my fieldnotes focused on the four topics presented in this study: coordination of practices, governance and openness, the entanglement between companies and community in f/oss, and the role of political ideals and values.

The last phase consisted of a final round of interviews in spring and summer 2019. After a phase of personal reflection of the data collected

so far and the categories for the final writing established, I chose to interview seven more TDF members to get deeper insights and reflections for the categories that I felt were missing from the data collected so far.

4.4.1. Interviews

In total 37 interviews were conducted over a timespan of more than three years. between 2016 and 2019. [6] In addition, my data collection benefited from numerous informal talks that sometimes stretched over hours. Information from these talks have resulted in field notes and memos. Also, they helped to create an atmospheric picture of the LibreOffice community and reach an understanding of the project and individuals that would not have been able through an interview.

Professional audio recording equipment has saved time and also influenced my fieldwork. The very first interview was taken in the middle of a hall full of 300 people at a conference. The second took place in a university cafeteria that was filled with the talk and laughter of around 100 people and the sound of two large Italian coffee machines; the next one was in a noisy university hallway, while some others took place on the stairs of the capitol hill in Rome, where the LibreOffice conference was held in 2017. The recording devices provided by my university, the powerful Zoom H4n, not only made transcribing the interviews so much easier because of the excellent sound quality it delivers. It also allowed me to take some interviews on the spot at conferences and hackfests. With excellent equipment, I found that arranging a meeting in a quiet place was unnecessary. Around half of the interviews were taken at conferences or community meetings. These were characterised by a more relaxed attitude by the respondents. They did not think that long or hard before answering a question. The other half of the interviews were conducted online via video calls on Jitsi, and two had to be moved to the telephone after not being able to set up a stable connection. The online interviews were more difficult to handle as the respondents hesitated to answer questions, fearing to put the LibreOffice project and people involved into a bad light, especially concerning the political significance of free and open source software and the history

6 See table 2 “Overview interviews”.

of the project, which involved conflicts between software corporations and the community. The physical distance and the lack of a conference or hackfests where people relaxe led to answers that were often very diplomatic. The interviewees took the occasion more seriously and their answers reflected that they represented TDF and wanted to avoid statements that could be interpreted by others as confrontational. Three respondents wanted to get transcripts to make sure they did not say anything that could land others in difficulties or to review that they did not say anything they perceived as compromising. In the end they only asked for minor clarifications that did not change the content of the statements made in the interview.

The interviews were generally unstructured. Coding was more complex and time consuming because of that choice. However, as I was trying to capture a phenomenon that was not fully defined, the unstructured interviews allowed me to learn about an aspect and then move on to another aspect after a set of interviews. The research process was therefore more flexible. I could interview people on specific topics, could dig deeper and ask follow-up questions, or just let the interviewees dictate the topics of the interview. Morse (2018, p. 1374) points out that the advantage of such a strategy with unstructured interviews is that interviewees can be used to verify information from other interviewed, thus add validity, and 'are encouraged to speak from their own experience as well as from others'.

These advantages were vital for the research process. In the first few interviews I wanted to find out general ideas and notions about the free and open source software community. It corrected a presupposition that I had about the free and open source software. The academic literature on free software focuses on hackers, their technical expertise and the cultural and political significance. The crucial involvement of tech companies that sell proprietary software to fund many f/oss projects became clear during the first interviews. Other tropes, such as the impetus on sharing and learning, the importance of different motivations and ideas of what f/oss is, and how diverse the setup is of what participants call the community, became important themes for the research. After the first round I moved on to choosing a project to investigate that would reflect the themes of the first interviews. LibreOffice emerged as the ideal project based on the findings so far. The community involved

in the project has proven to be sustainable amidst the influence of firms, it showed a diverse setup amongst the collaborators bringing together developers with designers, translators, and its history showed that the project has managed to combine different ideas and interests. Starting from the second round, all other interviews were conducted with members of the LibreOffice community. In connection with the observational data the next themes I learned about was the need for coordination and the open structure of the project and I followed these up in interviews. Participating in the LibreOffice conference allowed me to learn the importance of meeting up to discuss different opinions or possible conflicts as well as the significant role that hanging out plays. It also helped with gaining acceptance from the people and explaining my project and the reason for me intruding into their community. Establishing rapport with some collaborators at the conference helped me find key informants. The LibreOffice conference proved to be an appropriate setting to get to know the people involved. I shared coffee breaks and meals with them, and went for walks and quick sightseeing tours with them.

The specific setup of the project which relies on the involvement of companies as much as on volunteers became evident there as well and subsequent interviews investigated the reasoning behind it. Archived email lists gave me the opportunity to engage in historical work, reading on the discussions that led up to the split from OpenOffice.org and the start of LibreOffice and The Document Foundation in 2011. On the background of the interviews about the history of the project, different interpretations and reflections were made about the current status. The resulting data helped to understand observations made in the field and resulted in more interviews about the politics that are involved in the project. Interviews gave a first lead, then observed some political themes in the field, which I asked about in subsequent interviews. This strategy allowed me to uncover a discussion that most members did not want to speak about in an interview at first. Thus, unstructured interviews allowed me to move from one aspect to another. In the end the most important themes formed clusters.

One challenging aspect of this approach is the impetus on the interviewer to learn about one topic after another. Due to my position as a listener motivated by the spirit of learning about this subculture, I put

myself unconsciously at the centre of the process. The temporal structure of the research project with distributed phases of "offline" fieldwork and interviews worked to my advantage. Pauses that were filled with coding the interviews and ordering the fieldnotes were also time for reflection, going back to literature to read up on specific topics. Consequently, the focus for the last two rounds of interviews shifted from a focus of learning about the project to personal reflections of collaborators. This data was ultimately vital in gaining a deeper understanding of the specific form of diversity that characterises the project.

Sampling

LibreOffice is a software product which results out of the efforts of several groups and individuals. It is a collaborative effort as it combines groups with different agendas, individuals which are affiliated with other software communities as well, and teams specialised on specific components of the product or areas of the project. To represent this eclectic mix representatives from all areas of the project have been interviewed. I talked to developers, people who contribute to quality assurance, or to documentation, translation, infrastructure, or marketing. I interviewed some who are in engaged in the governance of the project, volunteers, company owners and employees as well as some who are employed by TDF directly. Thus, the sampling was not random but purposive (Borg & Gall, 1989).

In addition I interviewed experienced community members and some who have started to contribute to the project more recently. Interviewees from different countries were selected to get a representative sample because the local communities in LibreOffice have their own history, while they share a more broader common development. Then I have interviewed people who consider themselves to be short-term contributors to balance the data collected from the enthusiastic technologists and TDF advocates. The inclusion of those at the margins of the community provides an insight into potentially more complex and less organised structures on the periphery of the network (Handler, 2018). All together these strategies provide a nuanced and balanced view on the study object.

A few notes on (pseudo-)anonymisation and confidentiality

All interviewees were informed before the start of the respective interview that its purpose was this dissertation. At the same time, it was

highlighted that the resulting data could also be used for other, future publications. It was pointed out to all interviewees that they have the right to anonymity if they wished. In addition, they were presented with the offer of confidentiality and anonymity.

Anonymity emerged as an issue during the fieldwork. The first interviews were held in public spaces, visible and audible to others. These interviews were conducted without the interviewees bringing up confidentiality or anonymity. The online interviews were more difficult in that regard. It happened several times that informants highlighted during an interview that without being guaranteed anonymity they would not say what they were about to say. For that reason, I have chosen to anonymise all interviews with members of TDF or contributors to LibreOffice.

Anonymity can range from fully anonymous to very nearly identifiable (Scott, 2005). On that spectrum, two competing priorities needed to be balanced: 'maximis[ing] protection of participants' identities and maintaining the value and integrity of the data' (Saunders et al., 2015, p. 616). I have prioritised the anonymity that was offered to all interviewees. Many of them have given me insights into their private life and individual beliefs and opinions.

Sometimes confidentiality was required and offered to interviewees in addition. Merriam and Tisdell (2015) highlight the problem of anonymisation at a local level. Insiders might recognise who is the source of an anonymised quote. By pointing towards certain topics or talking about their private life, they become identifiable for insiders. I have therefore erased markers that could identify them whenever it was possible to keep the integrity and value of the data. I do not specify which position the people who talk have in the project. I also left out some quotes for that reason as well. The interviews are numbered in chronological order. Yet, full anonymisation cannot be guaranteed. Attentive readers who are familiar with the people who engage in the development of LibreOffice will most probably be able to connect a few interview passages with the corresponding person, either on the basis of the wording or for the message that is made in the statements. Therefore,

it might be more fitting to describe this process as pseudoanonymisation. Nevertheless, I have done my best to respect the interviewees wished for confidentiality and anonymity.

4.4.2. Observations

Observations were another cornerstone for the data collection. Six rounds of observation were completed for this study in total[7]: At FOSDEM in Brussel in 2016, 2017, at FOSDEM and the LibreOffice hackfest and team meetings in2018, at the LibreOffice conference in 2017, at the LibreOffice hackfest in 2018, as well as an observation of the main LibreOffice Telegram channel and further online team meetings between April 2017 and December 2018.

As discussed before, the first rounds were characterised by familiarising myself with the f/oss scene and then starting in 2017 to get to know the LibreOffice project better. Typical for this phase was a distant approach regarding the observations. I kept in the background, not revealing proactively that I was a researcher observing the field. At a conference as large as FOSDEM it was easy to not get called out and I blended into the crowd. As a consequence, the first fieldnotes contained a lot of descriptive information to capture the physical environment, how people moved around the conference, the procedures of talks and discussions.

A different approach was then applied starting with the LibreOffice conference in Rome. I announced myself beforehand that I would come to the conference as a researcher and I registered for the conference. Establishing rapport was easy, as the people I started to talk to informally pointed me to other people to talk to and, in that way, I was introduced to the structure and those who were considered core members of the community. At the LibreOffice conference the observation was turned into participant observation. At the first day when I stepped into the first session, I was immediately offered to participate. The session, which lasted a full day, was about interoperability. Tests were run to check whether LibreOffice's applications can inter-operate with other office suites or operating systems. I was given a short introduction and then I started to check if paragraphs were in the same place if

7 See table "Overview data collection" in the appendix.

a document that was saved in Microsoft Word would then be opened in the LibreOffice writer.

Starting with this conference in Rome, I established rapport to most of the people that I later interviewed. This fieldwork trip was also characterised by many informal talks while taking walks together or drinking coffee. The observations included community dinners and lunches. To write fieldnotes of these informal talks and observations was a much harder task. The importance of hanging out is a great advantage for an ethnographer. Moments to present yourself, to get to know each other, to chat and ask questions or, during later stages, to catch up were to be found easily. Taking handwritten notes during these moments or opening the laptop would have been irritating for everyone involved. To avoid losing a lot of details, most of the time I abstained as much as possible from the free alcoholic drinks that were offered at LibreOffice events – even though the craft beers during a long late summer night in Rome were really tempting. Following the advice that fieldnotes should be written as soon as possible (Hammersley & Atkinson, 2007), metro or bus rides were used to start writing notes and finished in my hotel rooms sometimes until late at night. It highlighted Van Maanen's (1988) suggestion that an ethnography is about writing. Fieldwork certainly consists of continuous writing and to combine observations and taking notes can be an intricate task.

Writing the fieldnotes of observations directly, sitting in public transport or my hotel room was certainly more difficult than sitting at a desk when it come to the online observations. The online observations served two purposes. The Telegram chats offered an insight into the general tone and structure of conversations within the community. It showed their humour (often quirky), as well as how communication was an integral part of the project. The flow of messages never stopped and sometimes it was exhausting to catch up after two weeks of not having had the time to read the messages. It was not unusual to have 200 messages within 24 hours. The second important insight that these messages gave was about coordination. Bugs were discussed a lot on the messages. As all the Telegram groups were open for anyone to join, community outsiders often asked question about technical problems or hinted at what they considered a bug. In that respect, it was interesting to observe how messages from outsiders who did not find the right tone

were treated by community members and also which people responded. Apart from the Telegram group, I also observed team coordination meetings and a board meeting. For these meetings, I asked for permission to join even though they were open for everyone. These meetings delivered data on organisational coordination, the relation between companies and community, as well as into decision-making processes.

4.4.3. Coding

Traditionally, the data that is available for ethnographers is produced by themselves through fieldnotes, a field journal, interviews or logs. In the case of this study additional data was available due to the effort of coordinating a distributed project. Collaboration comes with the need of constant communication which results in a large quantities of blogs, wikis, meeting notes, reports, and videos. To deal with the vast amount and variety of data was to switch from hard-copy coding to electronic coding with *Dedoose*.

The coding process started with open coding which offered flexibility and openness to develop an understanding of the free and open source software community. Then I moved into conductive coding, which was appropriate as I had little knowledge about the research subject to develop a codebook beforehand, and did not want I to impose a definition of collaboration that might affect my fieldwork. For the coding categorisation, I wrote short memos which reflected on the research process. Memos are 'the narrated records of a theorist's analytical conversations with him/herself about the research data' (Lempert, 2007, p. 247). The analytical distance gained with the memos not only helped to form patterns and clusters of the codes, but also forced me to rethink the project as a whole, my position as an ethnographer and the analytical and theoretical angle I was using. In the end the memos had a strong influence in how to understand and reflect on the research process, to question my preliminary assumptions about free and open source software and the adversities that are involved in the project.

Memos have been instrumental for the research process as they reminded me of suggestions to connect different interpretations, or how certain codes are related. These connections also brought forward new questions and issues to investigate in the next round of fieldwork or to

look out for in the daily messages, updates on the wiki or emails. The memos helped with ambiguity (Saldaña, 2013), and the themes that developed through the combined process of coding and writing memos formed a picture of a community that has its moments and practices of (re-)formation, while also reflecting the differences and practices to negotiate these differences. Different takes on the same story became a theme that helped to create a narrative when it came to writing the ethnography.

4.4.4. Online, offline, virtual, digital

The data that was collected online is not treated differently from the other data. They are both understood as an expression of the same site. As mentioned above, online data was sometimes different from data collected offline, especially concerning interviews. However, I used these differences productively and tried to find ways to let the offline data inform the online data and vice versa. Due to short stays at events and the constant availability of online communication, the importance of concise, continuous communication became apparent. On the other hand, personal meetings at events showed how hanging out and socialising in person is a much-needed part of the project.

This mindset is very similar to the one that characterises digital methods. The point of departure for digital methods is to embrace the web and its artefacts as a cultural expression that allow us to draw conclusions about culture and society in general, not just about a virtual culture or online communities. Understanding the web as societal data means to end the dichotomy between the virtual and the real. Rogers calls this 'online groundedness' (Rogers, 2013, p. 24). The online data does not need to be grounded offline, and does not require proof. Online data has its own legitimacy and I even used it to ground offline data.
Therefore, this ethnography is not a virtual ethnography (Hine, 2008). Whereas virtual methods aim at studying the dynamics of online cultures by perceiving them as different from the rest of the society, digital methods attempt to capture larger society by analysing computer-mediated communication.

4.5. A problem of scale

Returning again and again to the site has special importance for a study in what Denzin and Lincoln (2018, p. 1143) call a 'digitally complicated context[s]'. This was not only true for the fieldwork at conferences and hackfests but even more so for the online activities that constitute the everyday coordinative work of LibreOffice and The Document Foundation. Even though the importance of meeting face to face and to hang out together was underlined by many community members, not only is the project built online but team meetings are held on a regular basis by using video calls, continuous discussions and coordination work is done on IRC chats or in Telegram groups, and the project attempts to put as much information as possible about all the different parts of the projects on the wiki page such as interviews with contributors, the coordination of community events, blogs of community member, information about organisational matters. It also provides information and access to all teams such as design, localisation, marketing. The project is described in all possible details, including programming guidelines, howtos and faqs and access is given to all parts of it.

The vast amount of possible data to collected resulted in a problem of scale. This was one of the, if not the, biggest problem of going through with this ethnographic study. It is a problem that holds true for ethnographies of technical systems in particular, and it is well described by Susan Leigh Star (1999, p. 383) in her reflections on *The Ethnography of infrastructure*:

> It is possible (sort of) to maintain a traditional ethnographic research project when the setting involves one group of people and a small number of computer terminals. However, many settings involving computer design and use no longer fit this model. Groups are distributed geographically and temporally, and may involve hundreds of people and terminals. There have always been inherent scale limits on ethnography, by definition. The labor-intensive and analysis-intensive craft of qualitative research, combined with a historical emphasis on single investigator studies, has never lent itself to ethnography of thousands.

The Document Foundation provides access to all proceedings and communication through the wiki page. At my disposal were notes of all team meetings and board meetings during the three years I studied the project. I could have possibly participated in all video chats, and all

emails were accessible to read including an email archive dating back to the days of OpenOffice. IRC chats were open to join without the need to sign up as well as a large number of chat groups on the messaging app Telegram: channels for several languages and national communities and for each team as well as the largest channel for all general things. Especially the amount of data available through the chat groups presented a serious problem. I spent many evenings trying to catch up with the endless flow of communication that was going all day and night. Having to concentrate on other tasks, or going on a vacation, led to a congestion of unread messages that could mount up to around 100 over a regular weekend. In addition, the projects offers access to all transaction logs that concern the technical development of the software. It was tempting to find ways to use all the data. I started to pull data from Telegram with a tool called Telegram-Export but after a few weeks I had to admit that the vast amount of data, in conjunction with the technical logs were just too much to manage or condense it to a level that would make it analytically valuable.

Star (1999) recommends three alternative ways to study infrastructure: (1) Identify the master narrative; (2) Surface invisible work; and (3) the paradoxes of infrastructure. Transaction logs for software development include, amongst other elements, a time stamps, a user name, the importance of the task and a description that should allow others to know exactly what has been done. There is no alternative for the production method and contributors who miss a detail are consequently reprimanded and asked to correct incomplete or imprecise information. Getting things done corresponds with the master narrative which demands order, rationality and intentionality. This corresponds with online communication, which is purposeful and to the point. Reflecting the motivations and reasoning about the cultural significance of the project is excluded from the chats. Thus, I searched for moments when the master narrative was broken and discussions about meaning and identity were brought to the surface.

4.6. Writing an ethnography

Van Maanen (1988) points out that writing an ethnography is office work, not fieldwork. In the end doing an ethnography is an act of storytelling. After a period of observations and talks and writing field-

notes, the impressions and interpretations turn into a text. Texts impose an order on others - the research objects and subjects as well as the readers. Yet they also have an influence on the author. Through the act of writing an author gets a different perspective on their own thoughts. The mediation of thoughts allows to relate oneself to their own and the practices of writing. Conducting an ethnography is not a form of debating or representing reality. It is instead a form of getting to know others as well by mediating a particular part of the world that is created by the chosen method. Writing as one of the methods creates the research object as much as it creates myself as an author.

An ethnography can be written from several different positions. Van Maanen (1988) identifies three types of ethnographic storytelling: realist, impressionist, and confessional tales. 'Realist tales are not multivocal texts'; they rather offer 'one reading and culls its facts carefully to support that reading' (Van Maanen, 1988, p. 53). Aristotle (2013, p. 18) defined narration as the representation of actors and activities through one voice. The narrator adopts the voices of other and filters them. The practices of others are observed and framed through a master narrative. The author does not appear in the text to give the reader the impression to be presented a reality. In a documentary style, quotes and details of common activities are recalled to give an impression of realism.

The difficulty of such a position is the imagined objectivity of the produced text. An ethnography is always an interpretation, and these are hardly neutral, they rather are the product of a set of predispositions. Against the backdrop of writing about "the other" an ethnography is always a result of differences. As an author of a realist tale I (re-)present a hierarchy of meaning that creates a particular category to position the other(s). Skeggs (2001, p. 431) highlights the suggestion made by Marcus that 'realist ethnographers believe in coherence, community, historical determination and structure'. Defining the other is a core part of writing an ethnography. A realist perspective tends to de-individualise groups to give an interpretation and description of a culture, and tends to overlook differences within. Generalising terms that intend to capture a community help the readers to construct a mental picture because few of them have a prior knowledge that is highly differentiated. The category of the other is not a realist view of reality, but

it is a fantasy, a substitute that is used to differentiate the self from the other to construct the former. This debate was the cornerstone of a controversy about ethnography in the last three to four decades which started with Clifford and Marcus's (1986/2008) book *Writing Culture*. It highlights the construction of reality and offers an alternative reading of an ethnography as a dialogue between the author and the research subjects. On that note, Sherry Ortner (2006, p. 42) describes ethnography as 'the attempt to understand another life world using the self – or as much of it as possible – as the instrument of knowing'.

I have attempted to incorporate the major concerns of this postmodernist debate in this study. My position as a male researcher in a predominantly male environment helped me to get access that a person of another gender might not have had, especially at the start of the fieldwork when I attended events and approached people of interest unannounced. Personally, I did not witness any account of hostility towards women but some informants commented on the topic and its problematic nature in the wider tech community that is traditionally dominated by men. However, I was excluded once for identifying as a man. That was the Libre Ladies meeting at the conference in Rome when all female contributors met to discuss the position of women in the project and in the Free and Open Source scene in general. The worry I had at the start to deal with an almost exclusively white and male community did not come true. The majority of the contributors are white men but with Marina Lantini, the Document Foundation had a female chair at the time I conducted my fieldwork, and it has other active female community members that play an important role in the project. Second, LibreOffice has become a truly international project. While being centred around a few European countries and Brazil from the start, the appeal of having a free office suite available in their native language drew many people to the project from around the world. More importantly, I have tried to trace the translocal in this study: a community centred around a technology which is a digital commons and which can be translated for anyone to use without needing to understand a foreign language.

To problematise the concept of culture, I have refrained from labelling the contributors to LibreOffice as geeks or as part of a geek culture because the term is too vague and overloaded with different meanings -

even though the term is used for self-description by some members of the community as a form of positive reinforcement. For similar reasons I avoid the term hacker; what was used to describe technical excellence and the ability to find clever solutions is in the community not used that much anymore – when I have heard it was mainly used to describe long term community members as "original hackers". Further, in the broader public the term hacker either has a negative connotation for people committing "cyber crimes" or it is used excessively to describe different kinds of changes to manufactured products (think Ikea hacks) or for finding some sort of solutions for a mundane problem called "life hacks". I have included the informants into writing their culture by asking them about how to represent them. One of the most convincing answers was to avoid the differentiation between volunteers and staff members, because volunteers contribute to the community in their free time in the evenings and on weekends, and this is as important to the project as the work of staff members. Also, staff members keep on working on LibreOffice I their free time. This area gets even greyer in what is typical for free and open source software: Some volunteers get paid by their employers to contribute to a community. I have attempted to avoid the differentiation between staff members and volunteers as much as possible. In some passages, the difference is important though. This concerns the coordination within the project, the role that staff members play in this practices and the introduction of new volunteers to the community and the project. Generally, I think of the contributors to LibreOffice as members of a "recursive public". Kelty (2008, p. 3) introduced the term to describe people active in the free and open source software scene as being 'concerned with the material and practical maintenance and modification of the technical, legal, practical, and conceptual means of its own existence as a public'. Some people in the LibreOffice community hack code, some translate, others negotiate with politicians, while others write legal texts. The term "recursive public" reflects this diversity of the LibreOffice community.

By discussing ethnography and the problem of presentation and positioning, I have adopted the second type of ethnography which Van Maanen (1988) calls "confessional tales". I have switched to the first person to emerge from the narrative shadow and become the centre of

the story which is the story of the research process, the obstacles, struggles, and solutions. It combines 'a partial description of the culture alongside an equally partial description of the fieldwork experience itself' (Van Maanen, 1988. p. 91). This position allows me to mention personal biases such as the difficulty of maintaining a distance to informants, whom I perceived as highly educated with clearly-defined ideas about the project and themselves, or the above-mentioned problem of scale and the impossible attempt to analyse all available data. Thus I inevitably missed some data, as it was physically impossible to be present in two rooms at the same time or to analyse all communication data in various chatrooms over two years. I could have looked in more detail into more national communities to get a more international picture of the project. For time restrictions I chose not to. The presentation of LibreOffice as a socio-technical assemblage is incomplete. It could have taken the technical side of the project much more into account by analysing the software or the software tools that are used to produce the software in detail. I have described the strategies as an ethnographer to work around that problem. This method chapter is for the most part a description of ethnographic techniques, or a confession as Van Maanen puts it. It demystifies fieldwork by highlighting the research design, the consequent strategic choices and the resulting problems 'to demonstrate that an ethnographic report is more than a personal document; that it is something disciplined by proper fieldwork habits, including the attention an ethnographer pays to the epistemological problems characteristic of social science' (Van Maanen, 1988, p. 74). In the empirical chapters, I will make some more confessions. I will do so by using both ways that Clifford describes: As a newcomer to the free and open source software community who tried to study it in as detailed a manner as possible, and as a translator of the field, I have studied to give an understanding (a constructed one, a postmodern ethnographer might add) to readers.

Finally, I will also use what Van Maanen calls "impressionist tales" as an attempt to capture the readers' attention for some events that I witness and which I found unique or representative for the discusses practices that LibreOffice is made of. This form of narrative aims at a dramatic effect. It does not offer a direct interpretation but gives clues to what is important to the LibreOffice project. As 'narrative rationality is

of more concern than an argumentative kind', 'the standards are largely those of interest (does it attract?), coherence (does it hang together?), and fidelity (does it seem true?) (Van Maanen, 1988, p. 105). What seems to be inappropriate for a scientific study is encouraged by Van Maanen as it can offer a more effective representation than realist accounts.

I have used it to paint a better picture of certain practices by highlighting what rarely happens. Considering that practices are routine actions and behaviours, the extraordinary reflects on the ordinary and vice versa. Thus, the impressionist hints at the realist as well as at the confessionalist. The different tales stand beside each other rather than replace it.

4.7. Validity

Fetterman (2010, p. 11) suggest to assess the validity of an ethnography from an emic and from an etic perspective: 'The ethnographer's task is not only to collect information from the emic, or insider's, perspective but also to make sense of all the data from an etic, or external social scientific, perspective.' That means 'to gather sufficient and sufficiently accurate data to feel confident about research findings and to convince others of their accuracy' (Fetterman, 2010, p. 9).

To feel confident about my emic perspective I travelled to a LibreOffice conference, two hackfests, and to FOSDEM three times. Every time I spent several days with people who contribute to LibreOffice. Apart from observing and interviewing, I spent many hours of hanging out with them. While reviewing my field notes, I found many descriptions of social activities: drinking coffee or beer, going for lunch or dinner, standing in the rain in the line of a food truck, or sneaking out to have a cigarette. I was invited to take part in community dinners and enjoyed the pasta dishes prepared by members of the Italian community. That let me realise how hanging out and socialising is an integral part of a culture that is commonly associated with anti-social loners. But hanging out in conference breaks and sharing experiences at conferences helped to establish rapport. The connections deepened and I was accepted being there, which led to invitations to community dinners and parties. This acceptance was particularly helpful to feel confident about my perspective. LeCompte and Schensul (2010) underline that it takes

time and practice to be a good listener and emic researcher. The long research process also helped in that respect. I was able to step back and reflect on it from an etic perspective. At two points during the research process, I reached a stage of confidence about my insights. At the end of the second phase in 2018 the patterns that I had detected repeated themselves in many interviews. After a time of reflection, I wanted to interview some more people, many of them experienced community members to get their reflection. This last round of interviews consisted of only eight interviews, but they gave me a feeling of confidence that I had understood the project. I have provided an overview of the data collection including time, place, the methods used, and a short description of the purpose for each phase in the appendix.

The long research process was also needed to gain the trust of the actors. Before starting the fieldwork, I only had contacted one person of the project. Thus, from the start I often arrived at locations unannounced. To avoid making people suspicious of my presence, I talked to as many people as possible in breaks, before meetings, and conferences. As mentioned, I tried to hang out with key actors as much as possible. To establish these field relations is the key for successful participant observation and interviews as Hammersley and Atkinson (2007) point out. I did not manage to interview everyone I planned to. Three people did not respond to my requests for an interview. With these three I did not communicate a lot beforehand, and I had not established a strong rapport with them.

The issue of trust also influences the observations. To obtain reliable data from the observations, Argyris (2010) argues that researchers need to be trusted so that the actors do not behave abnormally. Regarding this possible problem, it helped to start the fieldwork at a large conference. Amongst hundreds of participants, I was not noticeable as 'the researcher'. Therefore, I could gradually build a network of trust with contributors to the LibreOffice project by informing them about my research and the future steps this would involve. This strategy helped me to be known and get accepted in smaller settings such as hackfests or community dinners.

Listening to the audio files and reading the interview transcripts multiple times is another strategy that creates closeness. Fiddler (2021)

suggests that listening deeply not only creates closeness with the interview material, it is also a valid interpretative process. In order not to take stories out of context, it is important to remember the whole of the conversation. Listening multiple times to the interviews I had assembled over a timespan of three years also helped to bring back the memories and the specific practices the interviewees talked about. The close listening had the effect that I had memorised the context and content of the interviews to avoid possible misinterpretations of the data.

Apart from the deep engagement in the field, validity is also given by triangulation of data sources, methods, and investigators to establish credibility (Denzin, 2017). Methodological triangulation is provided by a mixed-methods approach that includes observations, interviews, and documents. Topics of interests and pattern that were found via one method were followed up with another method. Actions that I observed (both online or offline) were followed up in interviews to get the actors' perspectives. The triangulation of data sources is given through the number of interviews as well as it is a result of the methodological triangulation. Through this process, patterns detected in observations could be clarified by interviewees, or comments made in interviews pointed me towards areas to look closer into when observing or searching for documents. Investigator triangulation was not given in this study.

4.8. Research ethics

Ethnographers 'often pry into people's innermost secrets, sacred rites, achievements, and failures' (Fetterman, 2010, p. 133). It is a very personal research method that requires a set of ethical considerations to avoid that the actors are harmed. I have already described the need to anonymise interviews and avoid any markers that could identify a person. I offered all interviewees to get a copy of the transcript to give their consent, as well as the possibility to delete certain passages. Fetterman points out the need for permission. I have asked a representative of the LibreOffice board for permission before the start of a round of observation. In addition, I have also asked individuals for permission if it would be accepted to join a meeting. Many times, this request has been unformal by asking people if they would be willing share their impressions of a specific practice. I always made the intent of my research clear and was also willing to give a more detailed description

if it was requested. As an act of reciprocity, I am in regular contact with some participants of the LibreOffice project to inform them about the progress of my research and I will make this dissertation available to them when finished.

The most delicate part of this study in terms of research ethics concerned the data that I obtained concerning politics which is presented in chapter eight. I observed a very intense discussion amongst board members about the future development of the project. After that I asked others about this dispute and how they see it. And even though the responses were made in an appeasing manner, it was clear that a bigger conflict is brewing in the project. In addition to that, an interview with one of the participants of the observed discussion delivered many harsh and direct criticism of the project and its current status. After listening to the interview a few times, I decided to ask in other interviews for more opinions on the topic. I was not sure if the publication of the material would break the trust of the actors. In the end I have withheld some passages. I judge that they do not add to the analysis while they might have been uttered in a moment of hot-headedness, or that this person expresses themselves in a manner that others could interpret as too adversarial.

5. History - From Star to Open to Libre

How does software start? What are the origins of the software called LibreOffice? In this chapter, the history of a technological object is at the centre. But instead of the technical features as a product, LibreOffice is examined as a project. A humanistic perspective draws from the perspectives and reflections of the involved people to ask questions such as: How has LibreOffice managed to gather people around it to get it started? When and how did the first attempts at forging connections or alliances happen? When have the paths of people and software crossed? These are the questions that will concern this chapter. These questions align with an ANT perspective.

At best some of them would be answered by investigating the origins of the community to find the first exchange of code, a software that is ready to be performed collaboratively, the first discussions about creating LibreOffice and the standardisation of practices that have come to define the software. This proposition partly required an approach similar to the historical sciences. It meant to read mail archives and archived discussion lists to learn about the positions, motivations, and ideas of the people involved. It also meant to search for archived websites to find press releases and conference schedules which deliver further traces of this software project's roots. Some of the sources were difficult to find and a lot of this chapter's material would not have been discovered without the help of some LibreOffice community members[8]. The digital archive offered by the Wayback Machine was particularly helpful to see and analyse some vanished web content. The Mail Archive allowed me to browse through the e-mails of the OpenOffice.org-community. In combination with interviews taken with people who have experienced the beginnings of LibreOffice some historic lines can be drawn, resulting in a sketch that provides a grounding for the additional chapters.

5.1. Starting points

Finding a starting point for LibreOffice is difficult. The 25th of January, 2011 could be used. That is the date of LibreOffice's initial release, the

8 Special thanks to Björn Michaelsen for pointing me towards the archived OpenOffice.org-mailing list.

software was born. The first version of LibreOffice was officially available for download from that day on. But it is also possible to set the birth of LibreOffice on the 28th of September 2010, the day on which the project LibreOffice was announced and the first beta version was released. Or does one need to look further back to understand when and why some people decided to start a new office suite and call it LibreOffice? The actor-network-perspective proposed in the theory chapter urges to explain that. A network always emerges out of another network, as Callon (1991) has pointed out. Thus, to deliver a better understanding of the collaborative practices that are important for LibreOffice it is useful to describe some of the networks that have preceded LibreOffice. This history does not deny the importance of dates; as a matter of fact it contains many of them but more importantly it relies on what is called symmetry (Latour & Woolgar, 1986) in ANT, both humans and non-humans are treated equally. Therefore, this chapter includes technical developments, organisational structures and how they link with people and the formation of practices.

5.2. Star Office

LibreOffice emerged out of OpenOffice which itself followed StarOffice. StarOffice was developed in a garage – as the story (Schäfer, 1997) goes – by a group around Marco Börries, a 16 year old German high school student. They founded the company Star Division and what was at first StarWriter, a text editor, expanded gradually into StarOffice by integrating other individual programs that are part of an office suite such as a spreadsheet application, a formula editor and a drawing software. At the start it was marketed as a cheaper alternative to Microsoft Office (Deignan, 2001). Even while it was given away for free, StarOffice was not free software as the source code was not open. What some commentators called a 'calculated suicide' (Dworschak, 1998) contained a key aspect of what would later become called the economics of open source (Bitzer & Schröder, 2006). Software as a product was not at the centre of the business model but services that support the software's users. StarOffice generated money by offering services that included the installation and setup of the program, by training users, tailor-made customisations and maintenance.

In terms of freedom and openness, StarOffice made a step towards being more collaborative in regard to using and exchanging files. While

Microsoft Office and Apple Works were only operable on the respective operating systems, StarOffice version 3.0 supported for the first time all operating systems including Windows, Apple, IBM OS/2 and also Linux i386. StarOffice became the world's first cross-platform Office suite (*What Is the Difference?*, n.d.) by producing binary files[9]. While it was still given away for free but as proprietary software, it made a first step toward a more open software landscape by not being exclusively produced for one platform. However, by being proprietary, the software did not allow collaborators, associations between the technology and collaborators could not be formed. The software was not open enough to be inspected or worked on. Translations could not happen and a network did not start.

5.3. OpenOffice.org

Sun Microsystems acquired Star Office in 1999 and quickly made it available as a free download. At the same time, the source was released under the GNU General Public License. The change was connected to a new technical feature. While StarOffice's compatibility was based on binary digits, Star Division already worked on a change towards XML and Sun deployed that change (OpenDocument XML.org, 2006). A few years later this file format became one of the core technical components of the Web 2.0. Its advantage is that it provides a lightweight programming model that allows loose coupling between different systems. XML is designed for remixability and hackability as it is human-readable and can be easily copied (O'Reilly, 2005). Regarding the Office suite landscape, XML's advantage was that it can be transformed quickly into other text formats. It allowed easier portability from one operating system to another and better interoperability with other applications that support XML (Deignan, 2001). As a result, StarOffice documents could now be opened in a web browser.

---

9 Binary files consist of binary digits. They allow compatibility a program to run on different operating systems. If a file is produced in one operating system it can be read with a different operating system as the digits stay the same.

5.3.1. A community starts

XML was also at the centre of another development. Based on the XML file format, Sun Microsystems (2000) started to develop the Open Document Format (ODF). To boost the process of working on the specifications of that open file format Sun decided to found the open-source project OpenOffice.org. One TDF member who participated in the community that formed around OpenOffice.org explained that 'open source was actually always a strategic project for Sun and never had to earn the money that it cost' (Interviewee 36). Sun's strategy was partly motivated 'to look good in comparison to Microsoft, to show that they are different, to show that they care (. . .)' (Interviewee 20). Sun's strategy of founding a community might not have been financially viable in the short-term, but the idea of supporting a community was aligned with a more general notion of belonging to a worldwide f/oss community. Showing that they care had other positive effects.

> [E]specially young people who join this community are much sought after because all of us are constantly looking for more people. We constantly want to have more developers in our communities because we want to achieve a lot, that is, we are more interested in binding people and taking them with us and showing them why what we do is exciting, interesting and fulfilling. (Interviewee 1)

The spirit of building something that could be used by anyone, independent of an operating system or a specific application drew people to the community. OpenOffice.org served as the website where employee and volunteers together worked on office suite that would run on the most used platforms based on XML An open document standard should be defined through a community effort (OpenDocument XML.org, 2006). The software became open for collaboration for strategic business reasons but also – at least to a certain degree – to connect with a growing worldwide community that was centred around sharing software. As soon as the software provided the ability to be collaborative, people were able to make the necessary alignments needed for collaboration. Here is the start of a f/oss community, or a network of humans and non-humans as ANT would call it.

Indeed, many people were interested to contribute to the project, some becoming employees and getting paid for what they used to do in their free time.

> I started to use open source software when I was at university and I also contributed to open source. Then I started at Sun Microsystems in 2011 in their OpenOffice team. And why did I do that? Because they developed open source which I found outright amazing. It was [also] brilliant to develop open source as a day job and to get paid for it. (Interviewee 36)

Free and open source software allowed people to earn money with their production skills but they also felt part of a technological movement. The free software movement delivered an ideological foundation for their personal endeavour by explaining that code should be free in order to be shared so that better technology contributes to a better world. Contributing to open source software was a high motivation for many who participated. Getting paid was not the only reason for people to become interested in OpenOffice.org. Participants' motivations were different: while some considered it a technological challenge, for others it was political postulation while some considered the licensing of free software a legal master stroke. The associations that happened here were not only between people but also between humans and non-humans. In ANT, the process through which associations became stable are called translations. Callon (1991, p. 145) notes that a 'successful process of translation thus generates a shared space, equivalence and commensurability'.

> For me, copyleft was intellectually appealing. The idea to deploy copyright, a legal instrument, and to use it to impose conditions on software so that it stays free. I found it particularly interesting that it rendered national borders meaningless. Furthermore, I was a Microsoft user and I could not read my old documents anymore. So, I have come across free software with which I was able to read these documents. (Interviewee 8)

Some started to learn one or two computer languages as a teenager or university student but soon realised that they needed an exchange with like-minded individuals to have support and camaraderie on their journey of writing beautiful code, of finding a backdoor to a system, or of a clever and neat solution to a problem that was thought to be more complicated to solve. Thus, the community combined technological interest with socialising around a piece of software.

> I have used StarOffice before. I have joined OpenOffice.org because a friend of mine liked the project and so I joined. I have certainly played around a bit with software when I was younger. As a student, I programmed a bit and I developed a game to try to impress a woman. (…) At OpenOffice.org I stayed because I digged the people. And my personal connections grew stronger and stronger the longer I stayed. (Interviewee 18)

The interview excerpts above have shown how the OpenOffice.org community started around associations with people and software. People made associations with the openly accessible software at the same time as they made associations with other people. The network started to grow as 'material practices are brought to bear' (Law, 2009). The associations happened between people and software, a network started that can be characterised as being made of heterogeneity. Not only through the heterogeneous materiality of humans and non-humans but also through heterogeneous strategies and resources (Law, 1991, p. 13).

5.3.2. Community of practice

One of the tasks of OpenOffice.org was to develop a standard file format which was open and usable on all digital infrastructures. Yet, standardisation was also important for the inner-workings of the community. It is often suggested in the literature that standardised practices in free and open source software improve collaboration (Bonvoisin et al., 2020; Gallivan, 2001; Gamalielsson & Lundell, 2017). They make the associations between collaborators reproducible. Through the standardisation of organisational practices solutions to problems are ready-made without the need for discussions about the right approach or the best solution for a problem. In the later chapters, the most important practices for LibreOffice are analysed in detail. The following short insights into the most defining practices are to show how collaborative the character of f/oss is. The development generally relies on peer-review, and the process is similar to the editing of a paper submitted to a scientific journal. In software development there is also a discussion between the authors and reviewers. After a contributor (the author) has programmed code it usually is considered to be work in progress. It gets uploaded to a review tool where it gets reviewed by another developer. The reviewer can have comments for the author, or the piece of code cannot be merged with the rest of the code. In these cases, the author

needs to make changes. If the reviewer cannot see anything wrong with the contribution, it gets tested and, if successful, it is merged into the code.

> If young people come to the community, they get taught how to it is done. It is a way of learning by doing. They are taken under the wing so to speak. There is a specific way [of doing it], and if the people are willing to learn, they can learn from the best. It is free of tuition, so to speak. (Interviewee 4)

It shows that f/oss can be explained as a *community of practice* (Lave & Wenger, 1991; Wenger, 2008) that builds around a craft or a profession where people share knowledge and experiences with each other. Yet, f/oss as a craft is not only characterised by the exchange of information to standardise and stabilise practices. There are general moral values and a philosophy that are constitutive for a hacker culture such as 'sharing, openness, decentralisation, free access to computers, world improvement (foremost, upholding democracy and the fundamental laws we all live by, as a society)' (Levy, 2010, p ix). These ideas are anchored in communities of practices and vice versa.

> You share certain values, you have a common language, you have common goals that you are trying to achieve, you may even try to achieve something with each other in some form and you have the feeling that you simply have a connection with these people because of the similarities. I mean people work quite simply in many ways and we need common ground in order to feel we belong to each other and in this community, it is just the case that we have a certain affinity for technology, that is one thing, but at the same time it is not just technology that counts, but it's also about almost a little bit of respect or appreciation for the work that goes into it. The understanding that many people have of software is that it would be an industrial thing, as if an assembly line worker could produce it, and they are all completely interchangeable and that's just nonsense. You can write software like this, for example there is Java, but the result is rarely really good, the really good software stories are written by teams, where it is more of a craft, a craft where the style of the individual master is also recognizable as if it were somehow a machine-reproducible story. In many respects, software is more comparable to high-quality craftsmanship than to industrial mass production. (Interviewee 1)

At that point it seems that a common language, common values, and standardised practices together are the foundation of a f/oss culture. What hackers managed to do is to spread their belief in the importance of making technology accessible for everyone as long as they are willing to learn. The key components of a hacker ethic that started to develop in the 1960s appear to still hold true. Sharing and creating access to information transcend the realm of being mere practices into becoming the idea of a productive freedom (Coleman, 2013) that places the exchange of knowledge above intellectual property restrictions. In doing so, a collaborative ethic formed that allowed a diverse set of practices to anchor. Collaborating on OpenOffice.org became a common point of reference (Chrisman, 1999) for the members of the community. The actor-network model that Callon and Latour (1981) proposed for research, if transferred to f/oss development, underlines the importance of practices and sharing those practices: Just as research is understood as an ongoing effort, f/oss is always in the making. It is a series of actions that lead to the formation of a network which in turn gives options for actions and resources for contributors. This was also the case for the OpenOffice.org community which was more diverse than earlier hacker communities were. A hacker ethic was transferred to projects such as OpenOffice.org that did not consist of just hackers but was rather characterised by a mix of software developers, people who write documentation, translators and designers. Access to knowledge and sharing are not restricted to coding but a project like OpenOffice.org require a pool of several different practices. Standardised practices are central for such a network and these practices not only concerned coding. Standardisation also concerns documentation to give one example.

> [Documentation] definitely is important for several reasons. One is that you need to develop a culture around the software so that people understand the tool that they are using. It is also important to get a reference of all the software that you have. Often, [young developers] don't like to write documentation. (. . .) And the consequences are that people don't know how to use, people don't know that it exists, and the developer doesn't get the credit for having a feature that people use. Documentation is important not only for improving the quality of the software but also to improve the ecosystem where people exchange, use, interact. (Interviewee 33)

To make it more approachable to others who did not work on the original code base, documentation is added to the code. Free and open source software is not only a result of coding but also needs documentation as much as it needs design and translation. By standardising practices, a culture emerges around a piece of technology. People that join a f/oss community need to learn about these practices to take part. Yet, the community can also be understood in a wider sense, in a worldwide community of people who share common practices.

> One thing leads to another and when I had learnt about how things work at Sun (. . .) I moved on to SuSe and I kept on working on OpenOffice. What is also great about being an open source developer is that you can change employers and you work on the same code. You can't usually do that with proprietary software. No matter if you are an employee or a freelance, that is totally brilliant, because what you do is not gone, but still there. You can use it in the next job, in the next gig. And the knowledge, that is the value and one of the values that you develop, that remains. And there are no employees who say your experience will keep you here or bad luck. (Interviewee 36)

Standardised practices are a strength of open source development, as they foster the maturity of the f/oss community in a wider sense. There is an understanding between those involved to be part of a wider community that is based on common practices, as many interviewees and people who I talked to informally confirmed. The software alone can provide technical openness, yet this openness is much better characterised as open collaboration and communication. Through standardised practices, free and open source software becomes collaborative, it allows interaction between itself and collaborators. And the standardisation of practices allows to form and foster new associations. Software emerges as a boundary object as defined by Star and Griesemer (1989) as an artefact if they are created and understood by a community.

5.3.3. A coalition of interest

Standardised practices played an important role in the formation of a culture at OpenOffice.org. Even though people were connected through email-lists and online production tools, different local communities

had their own identity, their own history. The heterogeneity of the network can also be about strategies and resources as Law (Law, 1991, p. 13) argued.

While a large part of the German speaking community centred around coding, the Italian community was started by localising OpenOffice:

> With OpenOffice there was a group of volunteers that was dealing with localisation and QA. There were also some developers, but the majority of the contributors are mainly in these two fields I mentioned. (. . .) As a local community we decided to create a new local association because the association in Italy is called PLIO, "Projecto Linguistico Italiano OpenOffice". So, it was directly connected to the localisation of OpenOffice. (Interviewee 34)

Localising, that is translating OpenOffice, also played a major part in other countries such as Brazil, France, and many Asian countries. The local communities came with their own history, whether for a community of a certain practice or a local community that has its own history. What combines these to local communities is the aim of providing access to a free-to-get free software office suite. In some communities this was largely a problem of software development, whereas in others digital inclusion was a question of translation from English.

OpenOffice required translation to be used worldwide. While hackers can resort to code as a language to talk to each other, an office suite is more than code for the users. A software application such as a word processor or a spreadsheet program needs to be built for regular users, and as much as English has become a lingua franca, there are large parts of the world where people do not speak English or where they feel much more comfortable completing their daily tasks within a software environment that is translated into their mother tongue or local language. Think of all the different tabs, menus, buttons and commands that are part of your word processor; it is no surprise that a strong arm of the OpenOffice community evolved around translation. Translations that played a central role in making the Office.org office suite globally successful.

The local singularities also depended on the political willingness to embrace f/oss as an alternative to closed source software. An IT boom in Brazil was largely due to the government investing in hacker spaces

where people could collaborate, teach, and train. Brazil also hosts the biggest open source conference in Latin America. (Oram & Safari, 2016)

> In the early 2000s, the government was supporting open source, we had a lot of support and opportunities of implementing open source in the public administration. That generated quite a bit of business. (. . .) The Brazilians did quite a lot and were reference to all countries in Latin America. (. . .) You need to develop a business around it locally and we had a situation where public administration invested heavily in open source software, (. . .) also to cut the cost of Microsoft licenses. (Interviewee 33)

In addition to the national differences between the communities, the interest of the individuals differed. There was not one single motivation for people to join the community. Contributing to OpenOffice.org was fun for some, while others saw a business opportunity.

> It was interesting to work und understand open source software. And I saw that [it] was a professional opportunity and it was easily accessible. The entry barrier for open source is usually very low. You download the code, do some tweaks and then you start to understand how it works. An office suite at that time, 20 year ago, was something very interesting in terms of the marketplace. And the only competition was Microsoft Office. It was something that was teasing me. Not in terms of business but in terms of having fun with an international community. (Interviewee 33)

What is portrayed as "having fun" is a social interest but also a political motivation. It is an interest in contributing to challenge a global corporation that, at that time, dominated the office suite sector. Others, as shown in the previous section, combined their personal interest, their geekiness, with getting paid as a developer. Some joined because they were fascinated by a product that was available for free and wanted to make it better known.

> In 2001 I read about OpenOffice and I thought let me try [it]. OpenOffice 1.0 was full of bugs but there was someone thinking about the software, it was not a clone of Microsoft word. They understand that bugs were there and then they cleaned it. And then I started to look into the community because how does it happen that I get that for free. I started following the email-list. Then I sent an email to John McCreesh [who was the marketing director for

> OpenOffice.org at that time]. The email was: John, you have a fantastic product but sorry your marketing sucks. (. . .) John answered me and I explained and showed him my CV and said to him that I think I can help you. So we met and (. . .) I started to do classic PR for the Italian community. I issued a press release every two weeks for the first six months. Downloads of the Italian version went up from 100.000 per week to 1.000.000 a week. (Interviewee 9)

The assemblage of different interests was characteristic for the OpenOffice.org-community. The diverse skills together played into a collaborative ethic that was further materialised with the release of OpenOffice 2.0. It was the first version that introduced ODF as the new standard file format for all modules. At that time, ODF has been already standardised. It could be used in all applications as a universal method of storing and processing information. This was an effort of Sun employees and volunteers who worked together to release software.

One could perceive OpenOffice.org at that point as an example of a boundary object (Star & Griesemer, 1989, p. 393) that was 'plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites', with 'different meanings in different social worlds but their structure is common enough to more than one world to make them recognizable', and coherence across social worlds that is maintained by the creation and management of the boundary object. A boundary object allows cooperation without consensus (Star, 1993). While on one level, the OpenOffice community confirms the simple definition of collaboration as a community of the like-minded who all work towards a shared goal of producing a better shareable version of OpenOffice, it is also possible to understand it as a non-consensual setup of people from different communities of practice with different standpoints, opinions, motivations, goals and values.

5.4. Divisions & frictions

While OpenOffice became popular due to local communities who translated the product, connecting and being mentored by more experienced contributors, the fabric of the community OpenOffice.org was tested throughout its existence. The boundaries that were most stressed throughout the project were those between the business interests of Sun and the motivations of the volunteers.

5.4.1 Licensing

In chapter 1, I explained the importance of licenses in f/oss. The licenses that were discussed there concerned the recipients of a piece of software. Licenses are also used internally in projects to determine the property rights every contributor has. Everyone who wanted to work on OpenOffice had to sign up as a member of OpenOffice.org. In order to contribute to the software, a Contributer Licensing Agreement (CLA). Signing a CLA is a common practice in f/oss: '[F]air licensing of all contributions adds a strong sense of confidence to the security of the community' (Bacon, 2012). Yet, a CLA is also viewed critically by legal experts specialised in f/oss. It 'gives rise to an asymmetry of power among a project's participants' as the project organiser 'gets special rights' (Fontana, 2019). In the case of OpenOffice.org, there was a clear asymmetry that was legally anchored with the CLA. Every contributor had joint ownership over their work, sharing the copyright with Sun. But the volunteers were legally not entitled to the contributions made by others. That right was exclusively reserved for Sun who had joint ownership of all single contributions. With that legal construction, Sun could control the software project as a whole (Gamalielsson & Lundell, 2014). Licenses are often depicted as a bureaucratic hurdle that drives away potential contributors. 'When the community begins to see more bureaucracy and repetition than useful and enjoyable contributions, something is wrong', Bacon (2012) determines. In the case of OpenOffice.org it was the asymmetry in power that frustrated contributors not the bureaucracy they had to engage with by signing the CLA but still most of the contributors were 'still kind of ok with it' (Interviewee 37).

5.4.2. Work arounds

The repetition that stood in the way of the fun that could be had by freely and creatively contributing to the project, was also not the most pressing concern within the OpenOffice.org-community. In contrast, it was a lack of standardised practice that connected the community's efforts with the development of the software.

> [The coordination] was just like in any other software company. The lion's share of development came from Sun. There was a project plan, a product management and a release plan. People managed their tickets for releases and features and worked through that. (. . .) There was no

> assignment in the sense of: community do that! It was more like: We accept patches when someone has a good idea or a feature, and we will benevolently examine and either accept or discard or ignore it. It was relatively difficult to get things into the Sun code because you always had to go through the code owner with the code review and depending on whether you were on good terms with the person or not, or whether it did fit into the product strategy [or not], it could be that you couldn't get it in or that people ignored it for years. (Interviewee 36)

'[I]t was hard to have some influence on the project as a volunteer' another interviewee (Interviewee 31) concluded. The perceived lack of influence was felt by community members in two ways. On the one hand the contributions made by the community were not taken seriously in comparison with the in-house developers at Sun. On the other hand, there was a perceived lack of communication and discussions regarding the relation between Sun and the OpenOffice.org-community.

> At OpenOffice it was possible to contribute as a volunteer but that level of freedom was not exactly… not comparable at all [to LibreOffice]. That one was a project with a company behind it that decided everything for the community. Like a normal company that is deciding what to do with the employees. You can't handle a community of volunteers like employees! (Interviewee 34)

For the problem of getting code into OpenOffice, community members found work-arounds.

The notion of a work-around explains well how users of a computer system found ad-hoc methods to solve problems either in way not intended by the system by not using the technology and finding other ways (Gasser, 1986). Confronted with the lack of standardised practices and the difficulties to get a patch reviewed or accepted, community members had to find a work-around.

> [Because of the difficulties to get patches accepted] all Linux distros[10] did not use the actual OpenOffice code, but a project called OOO Build, which made it much easier for the distros to package it. And a lot of patches have just accumulated over the years. That means it was a relatively easy way to get changes, at least in the Linux distros by simply adding a patch. (. . .) First put the patch in there and

10 Distro = short for distribution. Here it refers to a Linux operating system.

> then deliver it to Sun and then it took a year until it landed in OpenOffice and then you could delete the patch in the OOO build. (Interviewee 36)

The work-around described here shows how contributors had to use skill and knowledge of the wider f/oss field because of a system that was designed by a company without considering the needs of the community. From an ANT perspective, this can be marked as a moment of unsuccessful translation. Callon (1986) would call these work-arounds a failed "enrolment". Enrolment is the stage in which a set of strategies is established 'to define and interrelate the various roles' of others. At this point of the history of OpenOffice collaborators did not accept the strategies that were at place. Those who wanted to contribute to OpenOffice in a less bureaucratic, more efficient and self-fulfilling manner established their own practices. Resistance against an institutional system was expressed by a technological tweak in order to circumvent the company's mechanisms and to subvert the lack of institutionally established practices for collaboration with the community.

5.4.3. Control

Commons-based peer production means that the 'inputs and outputs of the process are shared, freely or conditionally, in an institutional form that leaves them equally available for all to use as they choose at their individual discretion' (Benkler, 2006, p. 73). The previous subchapter showed how the software as an assembled product of individual contributions was only owned by Sun Microsystems and thereby contravening the principle that the inputs and outputs are available to everyone. Benkler's definition also points towards the institutional format of a commons-based peer community and its importance in regulating and organising such a process. The way that Sun controlled the project was not limited to the development of the software. The company also maintained close control over OpenOffice.org, the community it founded to boost the development of OpenOffice and an open document standard.

> There have always been ambitions [to start an independent foundation], there have always been efforts, since 2002/2003 or so, from day one actually. The mention of an independent foundation was even in the initial press release in 2002, when Sun open-sourced OpenOffice. And that just never happened, on the contrary, the control mechanisms,

> that was really a multilayer strategy in all areas, somehow having checks and balances, with belt and suspenders, so that Sun could keep control of the project. So, there has been an effort all along, there have been people saying, what about the independent foundation, dear people at Sun? And things [were] not great, but ok for most. There were a few people who said we had to be independent, but most people came to terms with it. (Interviewee 36)

Sun published a press release quickly after the acquisition of StarOffice. Accordingly, the OpenOffice.org community was supposed to be under the roof of the OpenOffice.org Foundations. Community members should have been in control of governance. Here is the press release:

> The OpenOffice.org Foundation
>
> The OpenOffice.org project will establish the OpenOffice.org Foundation, a non-profit organization that will oversee the operations, technology strategy, incorporation of technology contributions, and establishment of standards in conjunction with other standards bodies and open source projects as appropriate. The intention is that this foundation will be modeled after the Apache Software Foundation. A Steering Committee (or board) will be established with members from the open development community and SISSL licensees. Sun Microsystems will hold a minority representation in this governance structure. (Sun Microsystems, 2000b)

The independent foundation as an entity to govern OpenOffice.org was never established by Sun. Community bodies that were introduced were introduced but 'they deliberately excluded anyone from any authority [and] anyway there was no way for volunteers to get into governance unless they were Sun-backed' (Interviewee 25).

> One was a project launched by the Sun company where all the threads came together at representatives of the company. There were pro forma community bodies such as a community council, but it was carefully composed so that Sun always had a majority. But no earth-shattering decisions were made anyway. (Interviewee 36)

The lack of collaboration not only became apparent on a technical level, but there were also no effective concessions made by Sun regarding the organisation and governance of the project. The rough consensus that people shared was tested on many levels. Taken all together, there was a lack of transparency and respect for the community members as the

project was run based on the company's interests. The lack of openness in the practices at OpenOffice.org is also exemplified by the announcement (Sun Microsystems, 2007) of the collaboration between IBM and Sun. The idea was to join the work that IBM has done on their word processor software, Lotus Notes, and the efforts made at OpenOffice.org. Only one week after the collaboration was announced, IBM released a beta version of a new office suite that assembled a spreadsheet application, presentation programme and a document editor. It became obvious that a week of sharing code and knowledge could not have been enough to expand Lotus Notes into Lotus Symphony. Thus, IBM had been working on this release secretly for some time with the consent of Sun while the community was not informed about this arrangement. Another detail enraged the community. Sun released the OpenOffice.org code to IBM under a proprietary contract license instead of an open-source license. Sun had the legal control over OpenOffice.org via the CLA that all contributors had signed. (Schoonmaker, 2018) While this asymmetry was already a thorn in the community's side, to use a proprietary license was 'an insult to the community' (Interviewee 27). While f/oss communities do accept to collaborate with corporate partners, such an obvious disregard of the value of the community was considered an affront to the OpenOffice community. They had already accepted a licensing agreement with Sun that made the company the legal owner of the product. The secret deal with IBM was another move that disregarded the community, which did not even have the same rights as Sun for the contributions or for the product of their collaboration.

5.5. The fork

West and O'Mahony (2008) clarify that in case an organisation found their own open source communities, attempts to keep up control will hurt the growth and ultimately the existence of the community. A lack of transparency and accessibility will lead to the community to either find work arounds or to a fork. I show how the community found workarounds to contribute to OpenOffice because Sun' lack of openness did not offer community members a standardised access to participate in the production process. The deal with IBM underlines how transpar-

ency was not deemed to be important for Sun, by giving away community members' contributions to IBM and by relying on a proprietary license. At some point work-arounds were not satisfying anymore for the community and it led to a fork.

A fork is a common practice in software development. Basically, it is best to think of it as a fork in the road. When a user copies a software repository and makes changes, the two versions become independent and changes in one do not affect the other. Without changing the original base, developers can try new things and later decide what to merge back in the main trunk without losing control over the myriad of branches that are under development. This practice is central to software development. If someone wants to contribute to a f/oss project, they can make a full copy of a certain version. If someone wants to use a project's version as a starting point for their own, they are also allowed to do that. If someone wants to stop working on the main line of development, and to be able to explore several options, a fork or multiple forks are created. After that, it is possible to merge the two versions so that the resulting line contains the changes. I will explain in a bit more detail the practices that are involved in a fork to provide a background for the last part of this chapter. It shows how a fork involved the split of the OpenOffice.org community. It is of special interest how a common technical practice in software development can also be used to finalise a schism in a f/oss community.

5.5.1. Version control

Another form of control is important for what should be the start of LibreOffice. In software production, so-called version control is crucial. It documents project development and facilitates the coordination of collaboration. Every step needs to be documented and the code needs to be backed up and brought into agreement with other collaborators. A version control system provides a complete log of changes of every file; every change made by anyone needs to be recorded, introduction or deletion of files included. In a flexible and sometimes remote process such as software development, every step needs to be documented, and the code needs to be backed up and brought into agreement with other collaborators. Version control lets developers track the changes that are made to the code as well as it allows easy backtracking of the

changes. In case of a mistake, contributors can compare several versions of the code, which helps them in fixing the mistake more quickly and with greater confidence. (G. F. Franklin et al., 2002; Sink, 2011)

Two practices are central here: Branching and merging. In rough terms, branching means that code is copied and two teams work parallel on the same code on different branches independent from each other. Merging reconciles different branches or integrates a branch into the main line of development. Multiple teams can work in parallel without immediately affecting each other. In order to control these cross-functional processes, controlling the versions is key. If several development teams work on the same codebase, the risk of intersecting or overlapping needs to be minimised. The aim is to have 'a clean, releasable version at the end of each iteration' (Kniberg, 2008, p. 2). A branch is a separate line of development that is independent, yet it shares a common history with another line at same point in the past. If the resulting codes do not conflict, branches can be merged or a branch can be merged back to the original code (the tree). In order to work on an extra feature for a software project, also a branch is needed to avoid disturbing the development of the main line or other branches. Branches are also important to release one candidate and continue the development on another and a team with multiple projects divides the work into teams and each of them works in their own branch. Connected to this control systems are also rules and policies. If one team merges their branch back into the main line, the other team must merge these changes into their branch to avoid a conflict in code. To avoid extra work, branching has a core policy that asks for an estimate of the work that has be put into a branch. A branch should only be started if the cost of branching and merging is lower than continuing to work on the main line. (Handler, 2018b)

5.5.2. Boiling point

The technical practices of controlling software production are required to write good code, but in the context of free software they can also be used to break free from a project while copying the source code. In such circumstances, the fork is a practice of leaving the party, of disconnecting with the project. The connection between freedom and control becomes apparent here. Free collaboration that allows to leave a project

by copying the source code and starting a new project is only possible because of control in software development. The tracking of all activities, the meticulous documentation, and the storage of data allow coders to break free in an environment such as free software that provides this possibility.

Because of Sun's unwillingness to engage with the OpenOffice.org community in an open collaboration that would entail equality of access to the product as well as to user contributions, there were discussions about the possibility of a fork throughout the years the project was running. The community was considering if the costs of a fork would be higher or lower than to continue working with Sun.

> We had discussions here and there over the years but nothing serious. We never took a decision. But the discussions flamed up here and again. Not necessarily on the mailing lists but at the conferences or community meetings. Every time Sun took a decision without the community, the discussion started again in smaller circles. (Interviewee 18)

The boiling point, or the point when the community agreed that the costs of forking were lower than staying with main line, came when Oracle acquired Sun Microsystems in 2009. Oracle was one of the world's largest software companies at that time and purchased Sun for US$7.4 million. The future of the OpenOffice project under Oracle's ownership started intense discussion on the OpenOffice mailing lists but in general tone of the discussion at the beginning was still to appease and observe what Oracle was about to do, even though it was unclear whether Oracle would continue OpenOffice as free software or close it.

> I think that right now nobody knows what is going to happen. (. . .) Of course I would also like to get information if Oracle has concrete plans but I think we might need to wait for a more days to know more. At the moment I am fairly relaxed about the situation. OpenOffice.org is free software, that means that even in the worst case not a lot can happen to us. We should just wait and see. (. . .) For me it is pretty clear that the OOc community is strong enough to "digest" every decision by Oracle. Let us not forget: WE are the community. Not Oracle. And you cannot buy a community. (Effenberg, 2009)

Soon however it became clear that the conditions for the collaboration between Oracle and OpenOffice.org worsened for the community members.

> When Oracle went in, things quickly changed for the worse - in many areas: communication, openness, availability regarding when things changed and when changes in the code were made. (. . .) (P]eople, even independently of one another, quickly said: It can't go on like this! (. . .) This has led some people to think out loud, what if? What if we did something of our own? And what if we don't do anything of our own and Oracle turns the tap off in two years? (Interviewee 36)

Forking also guarantees a remix culture (Lessig, 2008) by making the combination of projects available. OpenOffice has been forked several times before (see figure 1) but up to this point it was never accompanied by a community schism. Throughout, the OpenOffice community had stayed intact. But starting with the Oracle's takeover of Sun, which meant that Oracle was also the new steward of the community, it seemed that a point had been reached when the community considered whether the costs for staying have become too high. The tone on the mailing lists also changed. While hackers have the tendency to communicate in a direct and sometimes harsh way, the existing fault lines between employees (formerly of Sun, now Oracle) and community members became apparent when the community wanted to talk about how to deal with the worsened situation since Oracle took over.

> Christian,
>
> Christian Lippka wrote on 2010-10-08 16.58: >For me, this "discussion" is over for now (. . .). This is an OOo list, let us talk about OOo and let us work on it. There are other lists for other things. >
>
> We talk about OOo in case you still have not understood this. If it is not convenient for you what the community does [and] about what it talks, then you [the Oracle staff] have to unsubscribe us [the community] and show hereby that you [the Oracle staff] have not understood what a community is. We are OOo, we talk ab out OOo. (Effenberg, 2010)

Oracle's take over underlined the existing problems of a community that was, in essence, run by a for-profit company. Not respecting the needs of a community shows that the process of developing f/oss in a

peer production scheme requires not only participation from communities, but also the ongoing development of those communities themselves. Attempts to control or shut down a community sparked action by the OpenOffice.org community to redirect the project.

5.5.3. Fork is a five letter word

Forking code has often been portrayed as detrimental for free software. It was viewed as a bad thing because 'forks tend to be accompanied by a great deal of strife and acrimony between the successor groups over issues of legitimacy, succession, and design direction' (Raymond, 2000) In this context forking is understood as splitting a community into rivalling factions and it is considered 'the cardinal sin of OSS' (Ågerfalk & Fitzgerald, 2008). Further, some scholars have claimed that the most common outcome of a fork is its death (Wheeler, 2015). However, there are arguments that this negative view on forking is based on an outdated definition of the term and that it actually 'represents the single greatest tool available for guaranteeing sustainability in open source software' (Nyman & Mikkonen, 2011).

Forking OpenOffice went ahead in a silent way. In August 2010, the OpenOffice.org community gathered at an OpenOffice.org conference in Budapest. The conference itself went ahead as planned with key talks, discussion rounds (OpenOffice.org, 2010). A community event was also planned in a boat trip on the Danube. A core group of community members had agreed beforehand to abstain from the social event. Instead, they wanted to discuss their interest in creating a foundation that would ensure a more stable, community-based structure for the project and support its development as a free software office suite.

> In communities around the world, two or three groups independently of one another came to the result that the time has come, that the project, which many always had in the back of their minds, must now be realised. This was when these condensation nuclei formed and they came together at the last OpenOffice conference in Budapest where these people also met in person, and that was more or less when it was decided. The critical mass was there and then we said: Oh, let's just do it! (. . .) [The boat trip] (. . .) was in Budapest. (. . .) So, this event was very practical. Everyone who wanted to meet [to talk about the fork] said: Oh, I ruined my stomach, the goulash from the previous evening. About 15

> people said, oh, I'm not fine at all and didn't come to the boat trip and met conspiratorially in Budapest. (Interviewee 36)

None of the Oracle staff members who were part of the community and present at the conference were invited to that meeting (Interviewee 10). There was also no discussion on the OpenOffice.org mailing lists about the upcoming fork. On September 28, 2010, the committed community members from the OpenOffice.org community announced that they were breaking away to form a new organization. This group included members of the Community Council and several project leads. They formed a Steering Committee of developers and national language project managers to create *The Document Foundation* (TDF). TDF's mission was to build the OpenOffice.org suite into a free software office suite that was more widely accessible to users and developers. They gave this new office suite the provisional name of LibreOffice. Unlike OpenOffice.org, LibreOffice would not rely upon one firm's commercial interests. By contrast, it would be structured through an independent foundation, as envisioned in OpenOffice.org's original charter. TDF would thus provide a new ecosystem for individuals, corporations, governments, and other interested users to contribute to the software's development.

However, still nobody wanted to talk about forking because it was something of a taboo in the free software scene. Nobody wanted to say out loud that LibreOffice in fact was a fork. The press release (The Document Foundation, 2010a) to announce LibreOffice and The Document Foundation as an independent foundation which hosts LibreOffice carefully avoids mentioning the word fork, nor does it make any reference to a fork. Instead references to freedom and independence mark the tone of the text, as the first two paragraphs show:

> The community of volunteers who develop and promote OpenOffice.org, the leading free office software, announce a major change in the project's structure. After ten years' successful growth with Sun Microsystems as founding and principal sponsor, the project launches an independent foundation called "The Document Foundation", to fulfil the promise of independence written in the original charter. The Foundation will be the cornerstone of a new ecosystem where individuals and organisations can contribute to and benefit from the availability of a truly free office suite. It will generate increased competition and choice for the benefit of

> customers and drive innovation in the office suite market. From now on, the OpenOffice.org community will be known as "The Document Foundation".

The community's new website www.documentfoundation.org started with a faq-section where it was explicitly denied that TDF is a breakaway project:

> Q: So, is this a breakaway project?
>
> A: Not at all. The Document Foundation will continue to be focused on developing, supporting, and promoting the same software, and it's very much business as usual. We are simply moving to a new and more appropriate organisational model for the next decade - a logical development from Sun's inspirational launch a decade ago. (The Document Foundation, 2010b)

During the subsequent days in early October 2010 the OpenOffice.org mailing list were the place to exchange views, opinions, and explanations. Those who were part of the newly founded TDF also took part in the discussion. Again and again, they wanted to make clear that LibreOffice was not a fork.

> The situation is special as LibreOffice explicitly does *not* want to be a fork but it wants to move the OpenOffice.org project forward together with *all* partners. It is unfortunate that Oracle own all of the technical infrastructure and does not shy away to use that in order to enforce their will unilaterally. Therefore, I am quite happy to have secured a server whose content is not deleted suddenly tomorrow. It is a pure safety measure. (Behrens, 2010)

It was especially some the Oracle staff that felt aggrieved to be confronted with a fait accompli. They had been left out of the discussions to form TDF and start LibreOffice and now they thought that the reluctance to call LibreOffice a fork was a chimera.

> That is just a sham! LO builds its own infrastructure, its own mailing lists, bug tracker und its own independent code repository which is different to that of OOo from the start, it contained various patches and moves further away from it. (. . .) Where is that not a fork, however you want to call the [new project] and whatever the noble intentions are. (Rathke, 2010)

The new Document foundation's goal was to expand the range of contributors to encourage greater innovation and involvement. Since the foundation would be independent from a single corporate vendor, this would provide incentives for a range of companies to become involved, stimulating competition and eventually increasing consumer choice (The Document Foundation, 2010). This also included to make an offer to Oracle to take part in TDF. The reason to avoid mentioning a fork was to use it as a last resort, as some people involved in the process told me in person.

> It was always a possibility. We wanted to save our bacon so to speak. We did not trust Oracle at OpenOffice.org because they continued to keep the reins tight. But we honestly offered them to collaborate with us. (Interviewee 27)

But Oracle declined to collaborate with the TDF and contribute to LibreOffice (Heise online, 2010). While Oracle's decision had legal consequences for TDF that among other concerned the trademark rights, TDF has shown how a technical practice such as forking can be used as a negotiation tool beyond the technical realm. However, the negotiation tactic would only be successful if the community moves to LibreOffice – in other words, if the LibreOffice would be able to make translations to allow associations between collaborators. Going back again to Callon's conceptualisation of translation, the moment of threatening with engaging in forking could only be successful if it offers "mobilisation". If 'various relevant collectivities were properly able to represent those collectivities' (Callon, 1986, p. 196) without being betrayed. Yet, the unwillingness to clearly describe the status of the preparations for LibreOffice was not well received by some community members. They could not be sure to that they would be represented. They felt that they were left out of the preparations and discussions to form TDF and they asked publicly for an invitation (Lippka, 2010).

> It was a 'clear strategy not to offend anyone. So, you have the irony that by not communicating clearly some people got offended' (Interviewee 25).

Many messages on the mailing lists were emotional, the schism of the community between Oracle staff and other community members becoming apparent. Only some occasional mistakes lightened the mood and triggered some humour.

> >>A fork is not something indecent. So, why should we pretend that it is not a fork?
>
> >Because nobody liked to do what had to be done. That is why it is hard to spell it out.
>
> Fork, fork, fork, fork, fork, fork, fork, fork, fork! You see how easy that was? It is an f-word but it has five letters :-) (Bauer, 2010)

"How many letters?", was the first response in the discussion. The mistake alleviated the tension in the discussion. The direction was clear. The initiative for the fork came from a team of core members that have been around from the start. After going through the process of compromises and years of waiting for the independent foundation they had the wish to have an independent entity to collaborate again freely with the software and with each other. To convince the rest of the community apparently was an easy task.

> Almost the entire community who were not employed by Sun or Oracle moved to LibreOffice. With very few exceptions, maybe ... no idea how many that were, 300 to 400 people and maybe 20, 30 must ... who stayed loyal to the project. And most of them had some personal differences with one of the LibreOffice project organizers. So almost all community people who didn't have to think too much now migrated. (Interviewee 36)

One of the advantages the community had to organise this process was their local character. In every country core members of the community tried to explain the motivations and reasons for the fork and managed to convince most members.

> [I moved to LibreOffice] [b]ecause the community moved. That was a community decision. It was not a company decision. I would say 95% of the community was with LibreOffice within one week, two weeks probably. All of OpenOffice. Of course, it was a decision of a group that was representing the community. I was with the Italians, Florian and Thorsten were the Germans, Sophie with the French, Olivier with the Brazilians. We were the frontmen, but we were speaking with the community. It was clear at the time that Oracle and IBM were with OpenOffice because their objective was to control the community. (Interviewee 30)
>
> During the fork (. . .) we tried to share with the rest of the [local] community the reasons for the fork and why it was

> from our point of view a good idea to move to the new project. Unfortunately, we still left a few people on the other side of the moon. (Interviewee 34)

In the end, most community members moved with the fork to TDF to work on LibreOffice. It is thus difficult to qualify the fork as a schism of the community. “On the other side of the moon” were mainly the Oracle employees. For them the cost of leaving was higher than the cost of staying with OpenOffice. There concerns were motivated by a lack of information and the insecurity that the new project brought with it.

> You want to nix the organisational structure, the licensing model, the QA process, the release plan and the entirety of the development plan but you have no plan how to manage it. Thus: We think it was dumb so far but we don’t know how to make it better but we start with it nevertheless. That might be enough for a free time contributor who translates a few comments in the source code (. . .). (Michaelsen, 2010)

In the end, LibreOffice succeeded not only to exist but is still running as the most successful f/oss Office application. Several of the Oracle employees who collaborated on OpenOffice.org with the community joined LibreOffice after all when Oracle announced to stop the production of OpenOffice and donated the code to the Apache Foundation. Negotiating with a fork in petto turned out to be a successful way to redeploy a technical practice.

> What was a challenge at that time was the decision of forking the software. That was a very hefty decision. The implications were strong. We were absolutely not confident to be successful by doing that because we were fighting against some very big companies such as Oracle and IBM. And all the strategy that we took at that time was how to survive against these giants. But in the end we succeeded. They left the open-source-business for office applications and we stayed. (Interviewee 33)

5.6. Summary

This chapter has shown how LibreOffice has emerged out of other projects that have existed before. The role of technology in fostering associations has been central to this history. From an early start of a heterogeneous network the openness of software was a decisive factor. As soon as it allowed to make associations, a community of collaborators

formed. Heterogeneity was one factor in this development. Heterogeneous actants (software and people) with heterogeneous interests and strategies formed a network. The standardisation of collaborative practices served as an anchor for this community. Standards need to be complied with to allow a reliable coordination of the practices. They are not only important to control the production process, but they are also needed to facilitate low entry costs for collaboration. New associations can be made with the software for people outside the network. Thus, collaborative practices do not automatically emerge as an effect of a technology. While software itself offers certain practices organisational structures frame those very practices. In these formats they are offered as a resource for possible collaborators.

Problems in the project started when equal access to the practices was not given. The software could not be collaborated on by everyone in the community. If the practices are not equally accessible the network does not offer resources for actions for everyone involved. At this point in OpenOffice, work-arounds (Suchman, 1987) were developed to keep collaborating with the software. These workarounds highlighted a malfunctioning in organising the collaboration. At first, the workarounds could satisfy contributors who were not given equal access. But the fork of OpenOffice showed that a boundary object can break despite its flexible character. As standardised practices across the different sections of the project were lacking, it was impossible anymore to maintain a common identity in the community. While some associations disalign (Callon, 1991), other were reinforced. Yet, the boundary object that was OpenOffice.org was not malleable enough to make the necessary associations.

At the centre of this was the reinterpretation and shaping of technical practices to the needs of the community. Practices that are used as a common practice in software development can be reshaped and used as resources for ending the collaboration. Control is a significant part of digitality as a cultural logic (Franklin, 2015) but the practices can be redirected. Version control is central for software development, yet in the context of f/oss it is a practice that offers a rupture of a development process. Forking is a helpful practice to test innovations, but LibreOffice has shown how to use it as a practice to negotiate a conflict. Thus, a technical practice can also become a practice for negotiation.

The history of LibreOffice has shown that the history of a network is important for the formation of practices. A technology does not determine practices, neither can an organisational format frame the practices completely to their will. Practices do not suddenly emerge because a technology has started. Rather their formation is the result of a string of associations including a discourse that adds ideal concepts and visions to practices. Such is the fabric of a community of collaborative practices. Software itself has be collaborative as much as the ordering structure that is built around it has to allow collaborations. The fork has shown how such a community can be more sustainable than a project that does not allow free collaboration.

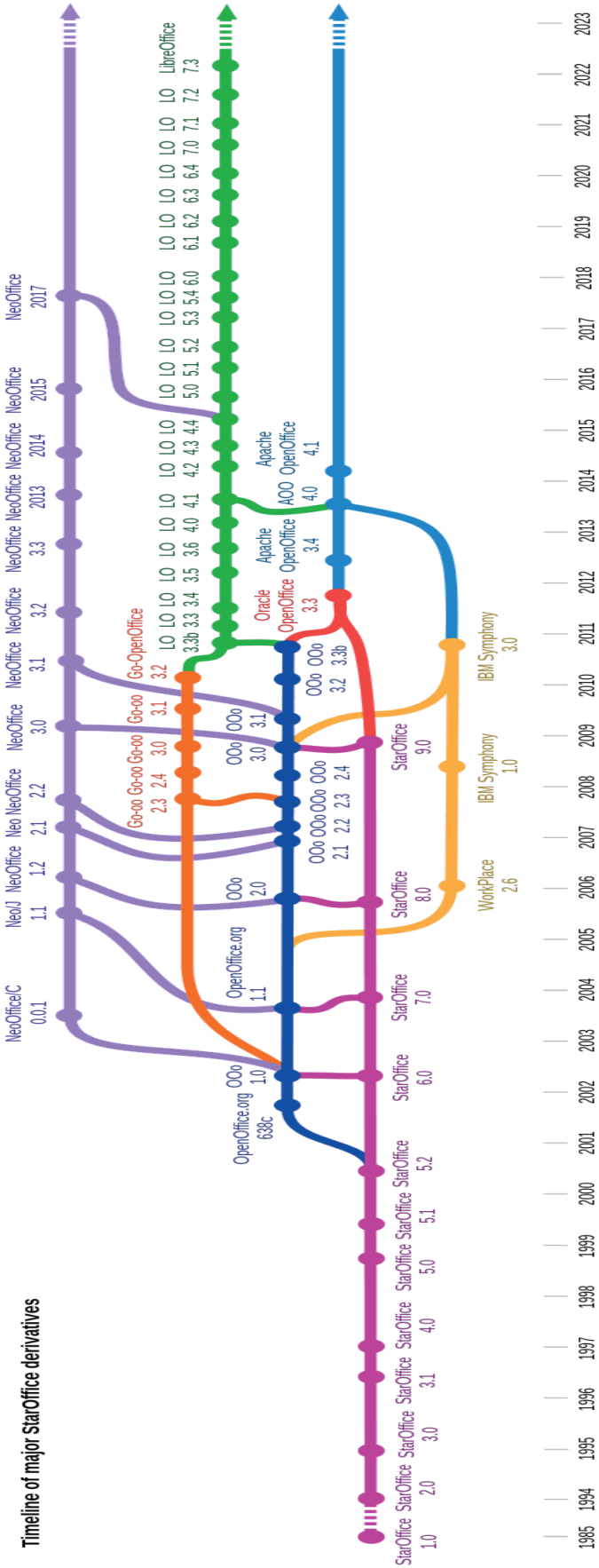

Figure 1: A timeline of major derivatives of StarOffice and OpenOffice.org with LibreOffice in green

6. Governing collaboration

After the fork from OpenOffice.org, the OpenOffice.org community announced that they were breaking away to form a new organisation. The Document Foundation was formed as a parent organisation. The code was the same and almost everyone from the community joined the new project, as it promised to allow collaboration without the disruption caused by limitations to free collaboration. This group included members of the Community Council and several project leads. They formed a Steering Committee of developers and national language project managers to create *The Document Foundation* (TDF). TDF's mission was to build the OpenOffice.org suite into a free and open source software office suite that was more widely accessible to users and developers. They gave this new office suite the provisional name of LibreOffice. Unlike OpenOffice.org, LibreOffice would not rely on any one firm's commercial interests. In contrast, it would be structured through an independent foundation, as envisioned in OpenOffice.org's original charter. TDF would thus provide a new network for individuals, corporations, governments, and other interested users to contribute to the software's development. By expanding the range of contributors, TDF advocates hoped to encourage greater software innovations and more involvement from the collaborators. Since the foundation would be independent from a single corporate vendor, this would provide incentives for a range of companies to become involved, stimulating competition and eventually increasing consumer choice (The Document Foundation, 2010).

The foundation was also developed along legal and ethical lines to acknowledge shared norms that put the community at the centre to avoid company-like structures.

> [W]hat companies do not understand is that a community is what I call a liquid organisation. If you give them space, they fill up all the spaces - but in a completely unstructured way, just like a liquid. If a manager says that a corner should be empty, the liquid fills it anyway. And companies start to be less hierarchical than in the past, but it is still too hierarchical compared with a community. (. . .) Their solution is that the community needs to adapt to the company, and this will never work. Companies must adapt to the community. If someone does something for free you

> can't tell a volunteer what to do. You can't tell a person what to do in their free time. These companies have to respect it and understand. (Interviewee 9)

This chapter analyses the transformation of the ideals of free collaboration into an organising structure that supports the practices needed for the production of LibreOffice. It shows how a legal structure and a governance body, in conjunction with software, provide a stable basis for the production of a f/oss office suite, confronted with distributed collaboration and defined by heterogeneous strategies. It also shows the challenges of governing and organising different forms of heterogeneity.

6.1. Statutes

Free and open source software is not automatically organised and governed in a specific form. What ultimately led to the start of LibreOffice was a company that could not offer enough moments and elements for translations. The collaborators could not freely 'forge alliances, work with one another, and circulate' (Callon, 1986b, p. 26). By reserving exclusive rights not only for the software but also for the governance of the community, the company could not speak for everyone involved. For LibreOffice, the aim was to create an organisation that can avoid the dominance of any single company. Thus, the history of LibreOffice is important to understand its governance structure. At OpenOffice.org, the rules and statues of the community were never legally binding and there was no assurance for the community that the collaboration would not be terminated by Oracle. In such a case, the costs would have been immense for the community members, as the company held all the rights to the code and the trademark.

> What we have done is to consider how we have worked together so far and how it has worked so far. What were the unwritten rules, which we possibly had, how was our togetherness, our codices, whatever you want to call it. And then we said, well, we would really like to put this into the form of a charter, into something that is legally binding. (. . .) We all know under which parameters we contribute and that is now legally binding. That is special about us, that for the time we thought aloud – in an extralegal space – how this should look like. The second step was to think how to put it into a legal format and how can we put this into a charter or statutes. The special thing about this foundation is, that is

> has started exactly like this. (. . .) What we built is exactly the right fit for us. (Interviewee 20)

The common practice, or unwritten rules mentioned by the interviewee, of collaboration were the basis to form a legal entity for the community. To a certain extent, these values and norms are shared with the wider free and open source software scene: open source code, sharing, collaborating are the key aspects of this realm (see chapter 1). Yet, The Document Foundation is also the result of misaligned associations that culminated in a fork. The collaborators in the Openoffice.org project were confronted with a company that did not share these norms and values. After the fork, the dominance by a single actor was out of the picture, the code was available, and the community could start to find a legal expression of their collaborative mechanisms. To repeat the words by Crawford and Ostrom (1995): They transformed belief structures into society by creating an ordering structure. It is important to point back to chapter 1 to emphasize that f/oss does not necessarily share a sophisticated belief structure, neither are the ordering structures only of one type.

> Free software can exist in different legal forms. KDE, for example, is a registered society which has members and they can change their statutes, if those members have new ideas. A foundation does not allow that. With a non-profit foundation constituted under civil law, such as the Document Foundation, it is not possible to change the core of the statutes. That is exactly the reason why we chose to form a foundation so that companies cannot foreground their interests and they close the source code. Insofar, we have set in stone our structures. Two are better than one, so to speak. With a change of people, the project can develop in very different direction. So, we have implemented a few instruments that should prevent that – such as that the general aim of the foundation cannot be changed. (Interviewee 8)

The practices, ideals, and values were the same as before, but now they could be written down and realised. They were translated into a stable platform that Callon (Callon, 1991, p. 145) deemed vital to generate 'a shared space, equivalence and commensurability'. On the basis of this stability a heterogeneous community could start to realise their translations by starting to speak for all collaborators. The second paragraph of the statutes explains the goals of TDF that are unchangeable.

> The foundation promotes and supports a sustainable, independent and meritocratic community for the development of user-centered free/libre open source software (FLOSS) based on open format standards (e.g. OpenDocument format). FLOSS can be used for any purpose, researched, altered for one's own purposes, shared and improved. Standards are considered open, if they:
>
> - are subject to public evaluation and use without hindrance in a broadly accessible manner;
>
> - do not contain any components or extensions that depend on formats or protocols, which, for their part, do not correspond to the definition of an open standard;
> - are free from legal or technical conditions that limit their use;
>
> - are developed independently of any single supplier in a process open to equal participation by all who are interested;
>
> - are available in various complete implementations to different suppliers or are equally available to all involved as a complete implementation. (The Document Foundation, 2012)

The foundation's charter was a clever solution to find stability with a diverse community in a fluid field-like software with a diverse community.

> The fact that the foundation's object is immutable is only possible under German law. Others such as Mozilla or KDE can work profit-oriented if they want to, something that we cannot do. The Gutenberg Foundation has survived 500 years and all kinds of wars. It was a conscious decision to build such a tower of strength. Especially in a fast-moving area such as software, we wanted to show that we are here to stay. (Interviewee 8)

These statutes show intent to formalise an ethic for collaborative practices. The project is supposed to be independent and meritocratic. The inclusion of independence is owed to the history of the project; TDF wants to avoid the dominance by a single actor as it had happened with OpenOffice.org. In alignment with the ideals of open source the statutes also mention meritocracy. Open source software advocates often refer to meritocracy as a fundamental characteristic. The central idea is that technical excellence is the only decisive factor in the open source

community. Everyone is welcome and respected, given they show their willingness and ability to collaborate. How the TDF's aspiration to establish meritocratic methods in their ordering can conflict with other methods and practices will be discussed in more detail in chapters 7 and 8.

More importantly now is to consider the statutes as a programme for translations. They offer a language that represents all collaborators. They act as an intermediary in the sense that Callon points out that 'an intermediary is anything passing between actors which defines the relationship between them' (Callon, 1991, p. 134). Statutes are not just the result of a process of wording, but also the result of negotiations between different interests. They are part of the formation of a network which is not a unified and streamlined production line. Latour's (2007) notion of networks makes it clear that they consist of these moments of negotiations, translations, and struggles.

6.2. Licenses

The legal aspect plays an important role in another way as well. Licenses determine the collaborative potential of a software product by granting or reserving rights to copy, use, share code. Licenses are an integral part of governing a f/oss project. They concern what Ostrom calls the predictability of system dynamics. Licenses guarantee predictability. They make clear how the software can be shared and under which circumstances. The problem with the CLA licenses during the OpenOffice era has shown how important they are. Despite all that, they tend to be neglected.

> One of the more recent tension is between the traditional free software activists which are very careful about licensing and about the rights of the software we are publishing and being able to only use software that is properly licensed and that you know you have the rights for. And there is, let's call it a new generation of hackers where that theme, that necessity of copyright is much less. [It was called] the post-open-source-society in the sense that the feeling is "fuck copyright, let's put it on github!". A generation of developers

> feels I'm gonna do what I do best because the licenses and legal affairs are complicated and troublesome, I can't be bothered with that. I just put it on Github and someone uses it and they don't have a license they are not violating copyright. And that might be ok if you are doing your small thing, and no one really cares about that. Of course, it will be a problem if you are a company and you base your business on code which others don't really have the right to use. (Interviewee 3)

Writing a software license or subscribing to software licenses is one of the practices that are accessible to provide information about the status of free and open source software. LibreOffice chose a free software license, but this particular license is considered to be a weak copyleft license. The Mozilla Public License Version 2.0 is a hybrid between a permissive license and a copyleft license. As shortly introduced before, copyleft implies that if you modify and share software that is under a copyleft license, it must be distributed under the same license as the original software. Permissive licenses permit to use a different license and can even allow to use proprietary licenses that close the source code. One important feature of licenses is that they can be combined. In the case of LibreOffice, the codebase is under the Apache License 2.0 which is a permissive license, a legacy of being a successor of OpenOffice.org. Everything built by LibreOffice on top of that is covered by the Mozilla Public License Version 2.0. This license has the capacity to include a range of many other licenses, both copyleft and permissive. As the future of the license was not clear, every committer to LibreOffice sent in a license statement to declare the further usage of their contributions under both the Mozilla Public License and the strong copyleft Lesser General Public License:

> All of my past & future contributions to LibreOffice may be licensed under the MPLv2/LGPLv3+ dual license. (The Document Foundation, 2019b)

The license has the advantage for LibreOffice to collect contributions from a big pool of collaborators with different political ideas regarding software. The project declares to have a 'strong commitment to copyleft licensing' (Lauhakangas, 2019) expressed by using the strong copyleft Lesser General Public License (LGPLv3). On the other hand, it uses a weak copyleft license like the MPLv2, which can be combined with the permissive codebase on which LibreOffice is built. And a

strong copyleft license does not provide 'advantages around attracting commercial vendors' (Lauhakangas, 2019). The specific form of licenses that LibreOffice allows reflect the collaboration without consensus that is symptomatic of the project. The licenses as an expression of software politics are not used exclusively but they are combined so that both free software and open source elements are covered. As such, the license for LibreOffice is a boundary object itself as much as it helps to form the whole project as a boundary object. The license is open and flexible enough to provide meaning for heterogeneous actors. It combines elements from free software with others from open source software. Such a construction brings the two groups together even though they subscribe to different ideas, it is a bridge model (Star, 2010). Licenses as boundary object offers ANT's translations between different actants and allow the standardisation of practices. However, they can only have this function if they are flexible enough to be interpreted in different ways. Besides this mutability that allows collaboration without consensus, Star points out that these objects are not random. Rather they are the result of information needs and local work arrangements; 'what is important for boundary objects is how practices structure, and language emerge, for doing things together' (Star, 2010, p. 602). Thus, flexibility or mutability can characterise boundary objects, but they also offer phases of stable practices that allow stability. Yet, if the mutability is not given as it was the case with OpenOffice.org, the boundary object does not offer any translations and practices cannot develop.

6.3. Manifesto

The foundation was also developed along legal and ethical lines to acknowledge shared norms instead of on the interests of a company. Collaboration is not only created on a technical basis, but it is anchored in social norms and governance. The Document foundation has codified its key principles in two related documents: the Statutes of "The Document Foundation" and the manifesto. The statutes relate to legal issues in terms of staying open as a foundation and not being taken over by a company as was the case with OpenOffice.org. They also describe the formal procedures of governance. The manifesto (The Document Foundation, 2011) refers to the values of the foundation. It gives a clear ethical interpretation of how collaboration is understood and how it

should be provided. References are made to access to technology, access in terms of language, openness through an open document format that is interoperable, open peer-review:

> Our values
>
> We commit ourselves
>
> To eliminate the digital divide in society by giving everyone access to office productivity tools free of charge to enable them to participate as full citizens in the 21st century.
>
> To support the preservation of mother tongues by encouraging people to translate, document, support, and promote our office productivity tools in their mother tongue.
>
> To allow users of office productivity software to retain the intellectual property in the documents they create by use of open document formats and open standards.
>
> To an open and transparent peer-reviewed software development process where technical excellence is valued.
>
> We reject
>
> The ownership of office productivity tools by monopoly suppliers which imposes a de-facto tax on global electronic free speech and penalises the economically disadvantaged.
>
> The creeping domination of computer desktops by a single language, forcing all people to learn a foreign language before they can express themselves electronically.
>
> The ownership of file formats by proprietary software companies - documents belong to their creators, not software vendors.
>
> A closed software development process where errors can lie hidden and poor quality is accepted.
>
> These values are connected to a specific mode of production, which is described in the rest of the manifesto.
>
> Our way of working
>
> Our core values lead us to believe in the following way of working:
>
> - the home for our activities should be an independent self-governing democratic foundation

> - membership of the foundation will be open to any individual who agrees with our core values and contributes to our activities
>
> - we encourage corporate participation, e.g. by sponsoring individuals to work as equals alongside other contributors in the community
>
> The Foundation will be realised in a not for profit organisation which will own assets and conduct financial and legal transactions on behalf of the Community.

This manifesto is a strong statement of intent concerning The Document Foundation's role, commitments, and goals, declared notably beyond the LibreOffice project to the users of this Office suite. It elevates the virtues of openness, transparency and accountability, and seeks to foster a commonwealth that upholds the production of free software and the pragmatic needs of users. The charter affirms the participation of corporations, but with the caveat of a well-defined moral commitment to the values of the community of users. It also rejects the use of closed-source software and actively takes an opposing position towards Microsoft and IBM, by relating to the ownership of file formats by proprietary software companies.

> What the foundation does is that it sets the criteria the development of free software has to fulfil. We are talking free software according to the definition of OSI. Certain measures how this has to be implemented are written down in the statutes. Together with the preamble that states that it has to be an office software with the aim to provide the usual application such as a word editor and a spreadsheet app. But more importantly, it must be for everyone – and that includes companies and administration. Thus, we communicate with them in that way, and we wish to get into a conversation with them, and that they communicate via the tools that we provide. Also, everyone should be able to take part in the digital society. Insofar, what we provide is a tool that allows to exchange and collaborate. (Interviewee 8)

The manifesto is yet another attempt to lay down in writing norms and values. It serves as framework for all the activities within the community. Here, the TDF carefully avoids any asymmetries to establish a ground for communication with everyone: all languages should be included, companies as well as volunteers. Again, the linkage of ethics

and values to the specific qualities of f/oss can be found. Also, the mention of technical excellence represents another hint towards meritocracy without mentioning it explicitly. The gained value of the manifesto is unclear. As one interviewee said: “All projects have a manifesto. So, we thought we might as well have one. I mean, it is about being nice. Just be nice and produce software.” (Interviewee 20) However, it is easy to overlook the effect that such a manifesto can have. It can help to create a sense of togetherness for a network, it can be an anchor for diverse interests and strategies. In that sense, the manifesto as well as the statutes and license function as ordering elements in the project. Ultimately, these ordering systems need the software to become realised:

> I can tell people about values as much as I want, then they all say yes, they all think it's good - well I've never seen anyone who says I think that's stupid, I don't want that, I've never experienced that - then they say “and now please give me” and then you sometimes have problems because we have to deliver the technologies that people can use to live the values in the digital world. (Interviewee 1)

6.4. Governance structures

Ostrom’s work on the commons has underlined that governance structures are needed to achieve sustainable systems. To paraphrase her: Instead of a central authority, collaborative groups need to be governed by the users of the common resource themselves. The governance need to be judged on their capacity to channel potentially self-interested motivations in ways that generate mutually beneficial outcomes (Ostrom, 1990).

> A crucial task of the board is to make sure that a group does not get too strong. The board has to work on balancing the interests that influence the project. If LibreOffice would develop in a direction that it could only be used by governing authorities then the board would have failed. The same is true for companies or private individuals. (Interviewee 8)

TDF’s governance structure (see figure 2) consists of three entities and an advisory board, which does not have the status of an entity. The three entities are the board and a board of trustees; these are individuals, members, similar to an association, and the membership committee which ultimately decides who can become a member. You need to

be a member to become part of the board or get elected into the committee. This setup guarantees that the selection happens in a meritocratic way, in which one must contribute for six months to the project to be able to become a member. Thus, the ideal of meritocracy that is often emphasised in open source software and which is included in TDF's statutes is boiled down to a commitment to stay active in the project for six months. The linkage of meritocracy to the excellence in writing code is superseded by the aspiration for stability.

Becoming a member is the first level from which one can then become part of the board or the membership committee. Then there is the board of trustees. Companies can become members of that board, those who have substantially sponsored the project financially – as it happens in most cases. The possible self-oriented motivation of companies and their representatives that have seats in the governance entities is limited.

> What we have done, with the ulterior motive to avoid being too dependent, is to stipulate that only a natural person, not a corporate entity, can become a trustee or a member. A person can work for a company, but the seat is not bound by contract to a company. In addition, only a third of the seats in one entity can be held by employees of one company. (Interviewee 20)

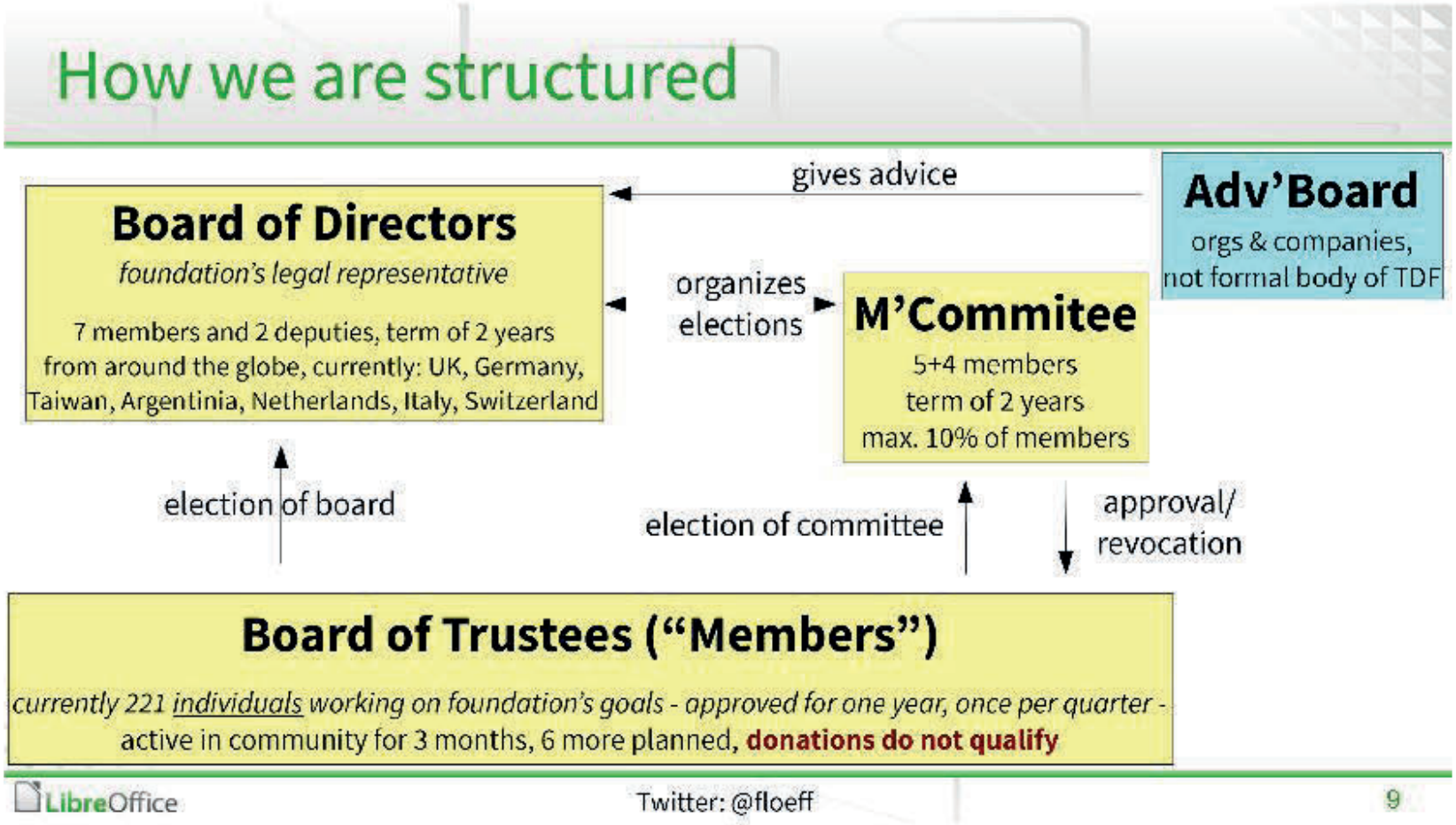

Figure 2: Governance structure TDF

The domination of Sun and later Oracle at OpenOffice has clearly influenced the strategies for setting up the foundation and its governance structures. The history of the preceding network serves as the foundation for the new one.

> What is special is our history. (. . .) But it can be explained by our history. We wanted to avoid at all costs an unbalanced situation. Generally, what we do and how we do it - many others do it as well. That we have a few special rules with the foundation because of our history is special. (. . .) What we have done is to introduce elements that are typical for associations - such as the intention of the members – into a foundation. The main difference in Germany is that associations can change respective of the intentions of the members, the associations, the association's objective can change if the members decide to do so. The foundation in contrast is defined by the intention of the founder. (. . .) We have a hybrid form. We do not have only one founder in the non-material sense, and we have added the intentions of the members as a central element. (Interviewee 20)

The immutability of the foundation's aims to be independent and to produce free and open source software is balanced with the more dynamic element of the intentions of the members. Ideally, this hybrid form gives room for negotiation to configure and mould the project, so long as it does not violate the aims of the Foundation. Various interests are balanced under the roof of the foundation, represented in the governance structures. But what for some is a balancing of interest in a heterogeneous community, is stagnation for others:

> [With Sun it] was a vendor-dominated cock-up. This is a non-vendor dominated cock-up. But it still pretty much is. But the good news is that at least the things we decided early in the project - apart from having it in Germany, having it as a foundation and whatever - seem to have continued and run reasonably well: much of the code, licensing. All the decision made early on seem to stand the test of time and work reasonably well. But even small changes to direction are very difficult to achieve. (Interviewee 25)

While Ostrom (1990) underlines the importance of governance structures to stabilise a project that manages commons, Callon (1991) and Latour (1996) point towards heterogeneous practices to provide stability but they require agreement on the governance structure. The gov-

ernance structures force condition over some in the project. To negotiate, discuss and adapt them is a vital part of the process as Spehr (2007) pointed out. The heterogeneous practices in governance result in stagnation for those who thrive for more flexibility and negotiations. For others the governance structures provide an important safety net so that history does not repeat itself. According to this view, is possible to make new connections but the dynamic nature of a heterogeneous network is decelerated.

> It has to be said that an organisation like a foundation is there to build a stable basis. And it shouldn't make it easy to change everything - on the contrary. Otherwise, someone could come and throw everything overboard and destroy it. So, a certain time and indolence is wanted and built into it. But I wouldn't say that's stasis or that no change is happening. (Interviewee 36)

A stasis is certainly not wanted by anyone within the project. But exactly this could happen if some members decide they don't want the board to make decisions for the project. The precondition for Spehr's second rule for a free collaboration is given in Libre Office: All participants can quit, limit or condition their collaborative effort.

> If the members are displeased with the actions of the board, then a certain number of members of the membership committee can bring forward a motion. Until a decision is made about that, the board cannot make any decisions. This is also a strong influence by members that is not typical for a foundation. The possibility to have a say is here even stronger than in an association. In the most extreme case, 30 % of the committee members are needed to initiate impeachment proceedings. (Interviewee 8)

What the presentation of the governance structures show is how the preceding networks can influence the setup of a network. While ANT, especially Latour, emphasises connections and associations as spontaneous actions, the governance structures of TDF show a direct influence of the project's history. The project is anchored in a governance structure that is almost impossible to negotiate for a minority. The intention was to provide stability. Yet this stability can impede the possibility of others to form new associations which in ANT have the key role to ensure that the network keeps running. This stability however can

cause frictions with different interests that want to concentrate on producing software and see it disconnected from the community, and especially from the ordering mechanism that the governance structure is:

> [A]ctually no one makes things happen because TDF just doesn't do anything. It doesn't change, it doesn't contribute much to anything. When you think about LibreOffice you really have to disconnect the relatively successful software project which is doing quite well – despite the odds. (Interviewee 25)

A break line within the project becomes visible around the question of how to order the project. One the one side there is a governance structure that is supposed to deliver stability. On the other side there is a demand for change and more dynamism. From a technical point of view, code needs to be maintained, to be curated and change is in its nature. From a governance perspective, stability is key to provide a basis for possible association. Stability of the network, as Latour (1996) noted, does not come from purity and unity. However, as Callon (1991) points out, convergence is necessary. This is reached through agreement to allow ongoing processes of translations.

6.5. Ecosystem

There was certainly no stasis when the fork happened. At that moment, the majority of the OpenOffice.org-community decided to quit their collaborative effort. The community stayed intact regarding the people involved. That also meant that the diversity of the group was maintained.

> Those [who started TDF] were the people from the OpenOffice community. That includes people who work on OpenOffice or open source software in a day job and those who do it as a day job - hence the label community, people who somehow contributed, who were active in the project. They were people, some engineers who really did it privately, people who did translations, some housewives to people who had a consulting business (. . .) to people (. . .) who worked (. . .) as software developers. (Interviewee 36)

The foundation provides a proper balance to make sure that all the interests are represented and advocated. That was the main aim that supported the fork. Without the start of an independent foundation the success of the fork was doubtful because it depended so much on the

robustness of the community to migrate to The Document Foundation. To balance interests also meant to be independent from a company. Instead, many companies take part in the production of LibreOffice. The result is what in f/oss is commonly described as an ecosystem that has emerged around LibreOffice. The term ecosystem is broadly accepted in the software industry to describe the collaboration between various projects, or to describe the involvement of different external groups in a project.

According to Lungu (2009) a software ecosystem is an environment in which several software projects develop and co-evolve. The LibreOffice ecosystem is large, consisting of volunteers and companies which develop LibreOffice. The company sell added value on top of LibreOffice. They provide support, consult, train users, or develop custom-made versions of LibreOffice. The companies that develop LibreOffice can sell LibreOffice if they want to but they must contribute back either by letting the code they build flow back into the project or by letting employees contribute in the project, or by doing both. The differentiation between a product and a project is decisive here. The companies work on a product but the project reaches beyond the software product. It includes participation in a community, or even being a member of TDF. In fact, at conferences and hackfests, many representatives of the companies in the ecosystem are present. Positions in the governance structure are also regularly held by company employees or business owners. They are part of the community; they socialise and hang out just like volunteers. Company employees were also part of forking OpenOffice. The connections between certain companies and TDF have a long history and they are strong. This is not a project-based development plan that rests on weak ties.

> I think [the relation between companies and community is] a symbiosis. (. . .) [M]y day job would not exist without LibreOffice. On the other hand, everything we do here: every line of code, every minute of work that I do here, directly or indirectly benefits the project. In my opinion, this is a symbiosis. Nothing should be taken for granted. One should not assume that those who do this in their spare time can be expected or demanded to do so. Nor should one indulge in the illusion that this is a project of the size that could be kept alive by investing one hour after the end of the workday. (Interviewee 36)

The idea of stability and balance can also be found regarding the ecosystem. An ecosystem cannot serve everyone if there is one dominant company. To collaborate on LibreOffice in an ecosystem does not require consensus but it requires diversity. To make heterogeneous connections outside of the ecosystem is vital. Therefore, the diversity needs to be governed.

> But then I suppose you try to interact with the community, try as much as possible and try to create an ecosystem that is fair for all. There is not one company and you need to find the proper balance so that not one company is getting all the money, alle the costumers. So, its complex. (Interviewee 34)

Diversity is a matter of importance in TDF. There is a clear marketing strategy that underlines the diversity of the project. It emphasises the achievements of the volunteers in the community to balance the influence of the companies.

> There are companies in the ecosystem that write a lot of code for LibreOffice. There are developers out there being paid full time, working for many companies, certified developers that write a lot of code. But so many of our translations for instance are native language projects that are done purely by volunteers. And this then gives people Office suites all around the world in their own language that Microsoft may not want to provide - or other proprietary software vendors - because they don't see a financial benefit to it. But, with our volunteers we have LibreOffice translated in rather exotic languages, languages that are not spoken by so many people. But in those countries, it gives them a free office suite. Contributors do a lot in design as well. Testing bug reports: Some great work by volunteers there because checking bug reports is not a very cool, sexy job, you know. It's a lot of grunt work, behind-the-scenes work. So, massive shout outs to those people. As with lots of open-source projects, it's a mixture of commercial interests contributing but without the community we would be far, far less of a project today. (Interviewee 21)

TDF has declared in its manifesto that ‘we encourage corporate participation’ (The Document Foundation, 2011) but the marketing strategy is designed to highlight the community. Some companies in the ecosystem do not agree with this balancing act.

> So, they are eager to present other people’s work as if it comes from the document foundation, very eager to. If you

> look at the pages people actually visit, none of the pages actually say that anyone but TDF or volunteers coordinated by TDF produce the code or produce the project. (. . .). This is just self-defeating. The reality is that companies contribute the vast majority of work and mentor the rest of it. So, why would you not want to grow these companies. Why do you not want to make the ecosystem more diverse? Why would you have such a short-termed view? (Interviewee 25)

The question of how to balance the model for the ecosystem was continuously discussed during my fieldwork. The tendency to highlight the community is understandable. It connects with the spirit of f/oss that a group of volunteers can build software together and rival corporate products. It also reflects the history of the fork when the community was the driving factor for the start of LibreOffice and TDF. After ten years of existence and growth, LibreOffice is one of the largest f/oss projects worldwide. The business model for the collaboration has not changed during that time. For that reason, there is tension between TDF and the companies who contribute the majority of the development work. As one interviewee described:

> One of the frustrations is that TDF is effectively a product company masquerading as an open source thing. It generates donations which it used to fund itself, but it doesn't actually do any of the development work. So, this creates a tension between the people who do do it, we need to fund that work, and the TDF. (...) We set the thing up and we staffed it with people doing admin. And now we spend a vast amount of our budget doing admin and marketing and nothing on improving the software in any way. So, you have the irony where TDF spends around three quarters of the budget on admin. And raises money on the false premises that people think that it improves software, but it is not. It is going to pay for marketing, admin and so on. Which is unfortunate really. So, TDF to me is bit of a disappointment. (Interviewee 25)

The Document Foundation does not contribute to LibreOffice by hiring engineers. However, the tension rests to a certain extent on different philosophies of what a f/oss foundation is. Some of the above quotes showed a clear alignment with the open source idea that is centred around innovation and distributed development. The foundation in contrast has incorporated a closer alignment to free software. The value of the community, the impetus on sharing and exchanging knowledge

are anchored in the statutes, choice of licenses, as well as in governance structures.

> We are not active on the market ourselves; we do not sell anything. But it is important that companies are in the same boat with us. The foundation does not provide developers. We have some people that are skilled, but we do not employ any developers. What we do is that we introduce developers [to the software], we create a footing, we share knowledge. (Interviewee 20)

Companies want their contributions to become more visible so that they can attract more customers. In turn, they contribute the development work back into the project. The argument however is not based on commercial revenue but on a central notion of f/oss: meritocracy. Meritocracy is part of also included at TDF which describes itself as a meritocratic community (The Document Foundation, 2012).

> It needs to credit the people who do the work, it needs to encourage enterprises to buy support and services from someone whoever- someone who can put funds back into the ecosystem. Back into ecosystem so that it grows and doesn't decline. It creates a virtuous circle of things being fixed and more people putting more money in. That's what we want. (Interviewee 20)

This tension between the value and importance of the community and those of professional developers is characteristic for LibreOffice. It shows the success of f/oss in general and the need for well-developed office suite. With volunteers alone, this work cannot be done.

> What is really important is to find the proper balance. On the one side, we have a giant worldwide community of volunteers that are contributing, who do a great job, a really great job. But it's not enough. Because we can't only rely on volunteers, it doesn't scale, we can't grow constantly. The commitment of the companies is important because they are paying developers, they are paying employees, they are working daily on the project. (Interviewee 34)

This tension of diversity is a result of the f/oss business model. It can be balanced in governing structures, through manifestos, but in the end commercial interests are also vastly important in organising and maintaining this collaborative project.

> We want a virtuous circle of economic growth where companies can invest and get a return for their investment which they will then re-invest and so on and so on. [. . .] There has to be that. The idea that volunteers will attract more volunteers who will do the work of professional engineers who understand the code and can sit down for a week and fix a bug is just Lalaland, it is just fairy stories. It doesn't make any sense. (Interviewee 20)

6.5. Summary

This chapter has focused on the ordering structure of TDF and how it facilitates the collaborative practices in the project. I have shown how an interplay of humans and non-humans (licenses, manifestos, statutes) link to build such an ordering system: Statutes, licenses, a manifesto, and governance mechanism are part of the project to sustain collaborations. Together with these elements, collaborative practices can be established as some of these elements serve as a boundary object as they facilitate interactions between heterogeneous interests. They offer robustness 'to maintain a common identity' (Star & Griesemer, 1989, p. 393). However, it has become visible that a common identity does not equal unity. This, from an ANT perspective, is not to the project's detriment. Heterogeneity is an asset as it allows to make new connections for practices. While there are frictions and differences to be found in the project, the convergence that Callon (1991) refers to is still reached.

The attempt to build such a stable legal basis for f/oss as TDF has attempted to do, is only kept functioning if dynamism and movement is possible. The stability is only given, so long as collaborative practices can be performed. These objects and the ordering system that they are part of need to allow collaborations. Otherwise the associations will not be aligned anymore. Hence, collaborative practices as much as the elements needed to facilitate them always shift between stability and change.

The second characteristic about the governance structure that can be distilled from the observations made in this chapter is that it is not neutral. The values inscribed in this structure are in direct connection to ideals that are traditionally linked to free software and open source software (see chapter 1 for more details). ANT does not focus on the

history of networks, yet I argue that the history of LibreOffice as a successor of OpenOffice.org is an important factor in the project. The statutes, licenses, manifestos and governance mechanisms are all informed by the experiences that the community has made before TDF. These non-human elements order practices. Practices thus become an expression of values and ideas, and these values become concrete through software.

I have shown how these values represent the two legacies of f/oss; the importance of sharing to increase knowledge from free software and the importance of sharing to build better software from open source ideals. While both sets of values can be found in TDF, not all aspects are fully activated. While TDF refers to meritocracy in its statutes, they are almost no practices to be found that are influenced by this idealised imagination to run a project. The reality of collaborative practices and the governing mechanisms around them are much messier.

7. Coordinating collaboration

Between July 2018 and July 2019, a total of 15, 137 code commits have been made to Libre Office (Nouws, 2019). Commits are changes in the source code. They are reviewed by another developer and if accepted as a useful contribution they are merged with a branch of the software. From 23 November 2016 to 21 May 2017 a total of 3,664 bugs have been reported. Bugs are programming errors. If spotted, a collaborator ideally reports them. Another person responsible for Quality Assurance (QA) reviews the bug report and, if possible, fixes the bug. Additionally, many other types of contributions are made to LibreOffice such as translations or writing documentation. They all have in common that they need to be reviewed by another person by using a software programme. Thus, sociotechnical practices emerge in order to coordinate contributions to LibreOffice. Yet, the group of contributors is diverse in different aspects in terms of expertise, regarding their status within the project, if they work for a company, if they are employed by TDF or if they are volunteers. This chapter shows how, in the absence of coordinating structures based on hierarchies, coordinating practices emerge. It explains how different levels of expertise are managed, and how newcomers are introduced to the project. It asks for the relevance of status in the project and how status can be achieved. It explores how volunteers and company employees coordinate, and it looks at how different areas of the project (code, design, documentation, translation, …) coordinate their work with a focus on bug reports and quality assurance.

7.1. Easy introduction and mentors

Different levels of expertise are normal in the project. Some people know more about coding than others, some are more familiar with the LibreOffice code base, or they know more about software design than others. Differences that play an important role in the project often concern knowledge of practices used in LibreOffice. It is a challenge to coordinate a f/oss project, as it needs to ensure that the software is improved without forgetting the need to keep the project open enough for newcomers who might not have the necessary skills or knowledge to make high-end contributions. In the personal communication this has been highlighted by many people I have talked to. A substantial amount

of time and manpower is invested into welcoming new people to the project.

> One of the biggest obstacles is having people around to mentor newcomers. This is a problem also again for many free software projects, I think. If somebody joins one of our mailing lists and wants to help, we need to reply quickly and need to point them into the right direction. It's great to have an influx in new people and after certain events we see a bunch of people jump onto our mailing lists and say "I want to help out". OK, that's great but can you tell us first what you can do or what is your ability and why do you want to help out as well. One of the biggest obstacles is mentoring and managing a big influx of newcomers. (Interviewee 21)

When it comes to developers, easy hacks are a measure used to welcome newcomers and to introduce them to the project. So-called easy hacks are bugs that can be fixed by correcting the issue. On their wiki, the Document Foundation provides *LibreOffice Easy Hacks* together with a description on how to get started. Detailed descriptions are given on how to run the software, including video tutorials, as well as some tips and warnings directed toward people who do this for the first time. This form of introduction is a first coordinative device that has several functions. It assists people who come to the project for the first time and makes them feel welcome, as the people working on LibreOffice welcome newcomers. It also makes the LibreOffice codebase – which is known to be large – better approachable for participants.

> LibreOffice has been talked about as being impossible to grasp. Well, the source code of LibreOffice is huge. It is indeed a very large code base and for many reasons we couldn't get in to fix the source code. We desperately needed a clean-up. It was harder to read and now it is better structured. Now it is harder to get wrong and people can contribute easier. Avid developers as well as beginners whose patches can be integrated much easier. Easy hacks point towards the area where we need solutions and they can help out. (Interviewee 17)

The code base of LibreOffice can have a deterrent effect. The sheer size of the code makes it hard to grasp the software as a whole. The software thus does not provide low-entry costs to facilitate collaborative practices. The easy hacks were introduced after LibreOffice was forked. The

organisational changes allowed to produce new practices that were directed towards new connections with people. Several members commented that it was the fork in combination with the easy hacks that reflected on an open sociality at LibreOffice.

> They made it really easy to join the project. I found them very welcoming, I found it easy to get involved. They were really pushing hard to make it easy for new people. Back in the time, I looked for other open source projects but this one was the easiest one in order to get involved. (Interviewee 19)

By providing step-by-step workflow processes, newcomers are also introduced to the established practices when it comes to managing bugs. Apart from describing how to get started, users are also told how to coordinate their involvement with the community. People should not take more than one part of the task at a time, should provide updates regularly, and if they do not manage to complete a task they should rest it in Bugzilla, the software used to manage bugs. Coordination is not limited to the technical side; it also includes thoughts about the maintenance of the community and to keep it open enough.

> Even if you are deeply skilled, please consider doing one little easy hack, to get used to the process. After that, you are invited to move on up to the more difficult tasks, leaving some of the easy tasks to others so they can get involved and achieve change themselves. (The Document Foundation, 2018)

Weighing up technical expertise and a lively community is the common struggle of *LibreOffice* and symptomatic of most free and open source software projects that seek an openly structured community. The EasyHacks together with the detailed description for newcomers are simple tools for coordination for the technical development as well as for a heterogeneous community. Triggering the interests of advanced collaborators by giving them space to follow their interests needs to be balanced with keeping a community open. People with beginner-level coding skills are eased into the technical challenge while they are provided with support from the community. They are assured that 'someone will review your commit and push it to master' as well as that they 'will receive a notification via email' after they submitted their patch, the fixing of a bug (The Document Foundation, 2018). Sometimes the slower and lesser effective approach needs to be favoured against the

speed and dynamism that would make the software better in a faster way. Hence, the appeasement plea to keep learning:

> The quicker you move up the pile, the more quickly you can be making large scale, user-visible changes and improvements to LibreOffice - of which these easy hacks are just the tip of a very interesting iceberg. (The Document Foundation, 2018)

The coordination of such a collaborative of newcomers with varying technical skills, the willingness to create a diverse and open community requires plasticity and coherence to become robust[11] (Star, 1993). Collaborative robustness means to be plastic enough to integrate contributors with varying needs, motivations, and skills while being coherent enough to coordinate their contributions into the same project. Research on f/oss communities (Birkinbine, 2020; Gamalielsson & Lundell, 2014; O'Neil, 2009) has shown that a set of shared practices provide stability and robustness, while practices to address different skills and knowledge levels need to be in place. Technical expertise is important to produce a good piece of software, but the collaborative robustness that is needed to achieve it involves more than selecting the most elegant piece of code.

Mentors are used in the project to guide newcomers. But this is not a job title that is given to someone necessarily. There are different paths to become a mentor, and this reflects the manifold structure of the project.

> We have two forms of mentorship. First, we employ people, we as a foundation look for people. That is the official mentor. And then there are many mentors, disseminators, in the project – people that are in charge of an area. They are contact persons. How do I become a mentor? A mentor is not a formal title. If someone is active, someone is contributing, someone is visible more than the others then this person automatically becomes a contact person. We have people, who are not employed by the foundation nor by a company, that have started to do something in an area and they contributed a lot. And these people are recognised as

11 This is an adaption of the categories that Susan Leigh Star introduced as defining qualities of scientific robustness: 'Plasticity here means the ability of the theory to adapt to adapt to different local circumstances, to meet the heterogeneity of the local requirements of the system. Coherence means the capacity of the theory to incorporate many local circumstances and still retain a recognizable identity.' (Star, 1993, p. 97)

> someone to ask, as a contact person. This happens automatically. (Interviewee 20)

To become employed by TDF it is not important to be a long-time contributor to LibreOffice. Some of the people who manage a part of a project have not contributed to LibreOffice before or had contributed very little. But concerning the unofficial mentor roles, being active and visible is the decisive factor. People who have contributed regularly become visible within the project. By reading the Telegram channels and looking at the logs I experienced myself how I can find out who to ask for an insight into an area of the project or about a certain a practice. However not everyone who has contributed a lot is deemed to be the right person to become a mentor. At a meeting held openly while a hackfest was going on, a discussion was held about a person to become a mentor. Other participants agreed that the person in question had the necessary technical skills, but some were unsure if the person that they were talking about had the social skills for that role. Thus, mastering the software is not enough to become a mentor since they are deemed to be important to grow the project, to attract newcomers.

> Of course, we try to get more members - in all areas. At the beginning [newcomers] get babysat, they get more attention. It happens rarely that something comes out of it. Very few keep on contributing and make progress – also in the way they deal with ideas. In most cases the ideas are very weird. Sometimes the ideas are bad and messy. If someone brings forward the same things and in the same way, it is increasingly difficult for me and others to react in a positive way. I see users writing in our bugtracker. That is how you start. If a newcomer starts to chip in regarding other tickets and expresses their opinion, it will be included. Everyone who comes and expresses their opinion in a comprehensible way, it has the same relevance as an idea from someone who is part of the project for 100 years. Sometimes it happens that people have good ideas. From this perspective it is easy to participate. The competence to say, for example, that this icon has to be visible, a few steps have had to happen to get to this position, you have to earn it. And these are things that present a hurdle for beginners. (Interviewee 18)

This quote stresses the openness of the project in terms of the general possibility for large-scale participation that was discussed as a cornerstone of commons-based peer production by Benkler (2016) and Bauwens (2005; Bauwens et al., 2019). In ANT, the ability to make new

associations refers to the same problem. Practices play a key role in this aspect. Collaborative robustness is achieved through standardised practices. The mentors' job is to introduce newcomers to the project by giving an understanding of these standardised practices. This concerns technical knowledge, as well as the social competence to learn how to present an idea or a problem.

7.2. Learning and trust

Coordinating an open community is an intricate task that involves technical expertise, a sociality that embeds this technical thinking and institutional openness. This assemblage provides practices which are open enough to allow new people to contribute to the project. To facilitate these distributed practices, people have to adapt to each other. For newcomers this means to learn how hackers approach learning.

> The open source community is like a union. People are supportive. But is a certain way of helping. You get pointed towards solutions but you have to make the effort to learn it yourself. (Interviewee 5)

Contributors get plenty of material to read. There are manuals for technical practices concerning LibreOffice itself, each software that is used to help produce LibreOffice has its own, there are wikis and FAQs. These documents and the other software programs necessary to coordinate the project are auxiliary media (Schüttpelz & Gießmann, 2015). Auxiliary media come in many forms in LibreOffice. There are documents such as manuals for a programme such as Bugzilla which is used to report and review bugs or for Gerrit where patches are submitted, reviewed and released. The wikis and FAQs serve as manuals as they show a step-by-step procedure for how to report bugs (Lauhakangas, 2020), or on how to debug (Stahl, 2021), and other common practices. For every area of the project there is plenty of material on The Document Foundation's wiki webpage (The Document Foundation, 2021b).

The plethora of material was also part of the problem of scale (Star, 1999) that was discussed in chapter 3.6. During my ethnographic research, I was constantly pointed toward manuals, wikis, and other communications that I should read to get a grasp of the project. Those pointers were not presented to me in a dismissive tone; people were helpful and did their outmost to facilitate my research. However, the

idea to learn as much as possible by yourself is important for hackers. Gabriella Coleman (2013) also points towards this principle of self-sufficiency which is embedded in a community that thrives on learning and sharing knowledge. As the interview excerpt above shows, mentors can grow tired and have a hard time staying positive when people ignore the standardised practices as it is often perceived as a lack of commitment to finding out on your own how things work. However, during the time I followed the Telegram discussion groups and IRC chats, rarely did I observe such instances. The famous rebuff "Read the Fucking Manual" that Coleman uses as an example to show that hackers can use an elitist tone when others do not respect the general rule of showing the effort to learn by yourself, rarely happened in LibreOffice. Most of the times, participants are helpful even if some people post questions in the wrong discussion channel.

> <individual-it> I'm locking into the UI tests of libreoffice and would need a bit of help. Who could help me and give me some ideas e.g. how to communicate with the search bar
>
> <thorsten> individual-it: hmm, that's actually a development question - best asked on #libreoffice-dev, but wait a few hours for the hackers to wake up ;)
>
> <individual-it> tried it there last week, will ask again. what time zone do most contributors of LO live in?
>
> <Zdeněk> I think search bar is not supported yet. See https://wiki.documentfoundation.org/Development/UITests#Unsupported_ui_items
>
> <individual-it> should have read the doc more carefully (Telegram channel #libreoffice-qa, 11 June 2019)

The helpfulness shown in this example is a form of sociality that reflects the needs of free software contributors. This is further shown during Q&A sessions. They have a coordinative function as they are used to exchange knowledge, to help each other to get more individual knowledge, which is turned into the advancement of free software. Sometimes the friendliness can change to a more abrasive tone. If a user does not show willingness to become active but just keeps reporting bugs and complains about the long waiting time for a reaction, they can be reminded that 'open source is not a spectator sport' (Bjoern Michaelsen, Telegram channel #libreoffice-de, 3 August 2017). Tensions emerge – as in this case – not because people have not read the

manual or because they do not bring the necessary technical knowledge to the project, but because they are not contributing: 'Just complaining is rather not a contribution!' (Interviewee 10) There are procedures to report problems with the software. The descriptions of problems by someone who constantly ignores these standardised practices are not seen as helpful but as a complaint.

While people need to learn to frame questions in a detailed manner in the right channel while showing eagerness to contribute and willingness to learn, those in a position to help others need to have the capacity to listen and to show expertise.

To coordinate certain parts of the project, the Document Foundation employs several people as heads of the respective groups. Their function is to oversee the workings of a specific part, to keep the deadlines that are needed for a new version of LibreOffice, or to keep the infrastructure running, to facilitate regular communication amongst people who are interested contributors with weekly or monthly calls that coordinate activities. The heads have power positions, as they are backed in their decisions, which are coordinated with the board of the foundation. But they do not restrict access for people if they want to contribute:

> You need to walk the fine line between not destroying the community, keeping it welcoming but also not accept every request. I don't want to feel this people to be rejected because they don't [get] what they want and I think that's a common struggle in free software communities or more generally in communities with volunteers. (Interviewee 26)

Providing a system of learning together by being helpful while keeping the system open is of utter importance to keep the community alive. Trust is then built quickly if volunteers show commitment, expertise, and willingness to learn.

> We give them freedom to what they want. We never tell them, you have to do this, or you can't do that. We always try to look after what they are doing. If they are doing something wrong, we tell them what they did wrong and how they should do it. Especially at the beginning, we kind of teach them how to do it. Sometimes you have to control a bit – especially newcomers. But I don't control every change they do, just a normal overlook. But if I see a new volunteer

> contribute for a month, I just trust this person. (Interviewee 19)

The degree of coordination is different, depending on the users' expertise. Even newcomers to the project, who would normally be looked after if they need help or if they have strong technical skills, are sometimes trusted immediately.

> Actually, I do not coordinate at all. I am trusted to do my stuff. I am working on something that no one else works on. But I also just upload my contributions to Gerrit where it can be checked and modified. But normally the things I upload find their way into LibreOffice as they are. I upload it on Gerrit and then those you have some expertise – they are part of the developer group – they sit down and look at it. (. . .) Then we discuss [it] if necessary, sometimes they are changes to make and suggestions are made. After that I can sit down and make the discussed changes. That is how the collaboration works. (Interviewee 32)

If a contributor brings specialist knowledge to the project, he is less likely to be mentored. Especially, like in this case, if the person follows the standardised practices to share his contribution for review and is willing to edit his contribution accordingly. This contributor works autonomously to a certain extent, but coordination is still given by the review system. Reviewing is a central practice in software production. In contrast to management review in some other company environments, the peer element is decisive here. It is a collaborative practice to improve and assure the quality of the software. But if people show they are capable of working autonomously by following all standards, that characteristic peer practice becomes obsolete.

> I mean in Gerrit you upload your patch and then it is released. To release it there are two ways. Either you release it yourself [or you need someone else to review and release it]. Therefore, you need the rights to release it and you only get this after a while. I asked after a while if it would not be simpler if I myself could release my patches and there was no discussion. I see it that way. At the start it is nice if contributions get reviewed so that you know that you do not mess up something. You are glad if someone reviews your work because it can only get better. It is the case with LibreOffice that the code base is relatively big and I don't know everything. It is difficult to say, well, I make my contributions and I don't want to bother with the everything

> else. So, after a while, if people now your work and trust you, you can upload directly. (Interviewee 32)

Sometimes the individual approach of coding stands in contrast with a coordinated collaboration. Yet coordination practices are flexible enough to be adapted if needed. However, only technical expertise, together with having proven to be in control of the standardised practices allow contributors to get the trust to disregard the norms, including the social norms.

> I rather stay away from community events. Not enough happens there. At another project I went to a community event. You begin to talk and can meet face2face and people were very open and very family like. If you wanted to become part of a specific group who focused on a specific part of the project you did not need to pledge or fulfil some tasks but from the early start on you were a full participating member with the same rights as everyone else. If you contribute something you are a team member, that is it, full stop. I also liked another meeting that lasted a full week and the only purpose was to work on the software. And then there are other events where it is about talking, and networking first and foremost. And that's not for me. I am happy to contribute and discuss contributions, but I do not need to hang out. Well, I prefer to work together, to sit and talk code. (Interviewee 14)

The role that trust plays in the formation of collaborative practices refers to the importance of the relation between collaborators emphasised by the concept of peer production. The relations at place in LibreOffice are generally marked by a care for each other. A sociality is developed that embeds practices and stabilises them. Yet, as I have shown there is enough flexibility given so that some collaborators do not strictly have to follow the social practices. That is especially true for those who have proven to have enough skills to be trusted. Software plays an important part in these processes. Because of the size of the code base, few collaborators have a complete overview. Thus, the code base requires that collaborators help and engage with each other, even though some might prefer to stay away from social events. Instead, software and the work on software becomes the intermediary that offers a language to make associations. Software translates by becoming an intermediary (Callon, 1991).

7.3. Invisible work

While hackers appreciate clever code as much as UX designers admire elegant menu systems, and sharing them among peers bring recognition and status, coordinating a heterogeneous community does not get the same recognition. However, community mangers have become more visible in the last few years. At many free and open source conferences, organisers offer development rooms to groups that either work on the same project or want to discuss a specific topic. In the last few years, the rooms that discuss community management issues have grown to be amongst the most visited.

A broad project with many areas such as LibreOffice needs to coordinate the different communities of practice (Wenger, 2008): translators, developers, documentation work, design, infrastructure, QA, marketing. The different sections need to have an overview of what is happening in the project, which is too large for a person to be involved in all sections.

> That's something I am trying to improve but one of the issues with a large software project is that we have many many different tools and communication channels. So, lots of things can be found on: We have people chatting on IRC, people chatting on telegram, people chatting on mailing list, there's work on the wiki, people meet in person, some people chat on Jitsi. So, there's so much going on in these different tools, we have Redmine, an issue tracking tool as well. So, all of these things are happening and linking them together all these disparate systems is quite a challenge. (Interviewee 21)

This is also articulation work in the sense that it 'names the continuous efforts required in order to bring together discontinuous elements - of organizations, of professional practices, of technologies - into working configurations' (Suchman, 1996, p. 407). Articulation work is often invisible work in free and open source software studies. Community management that is not directly related to technical developments is an important part of the project.

> We don't have a community manager. We had one in the past but that was the most hated person in the community. A community manager should not be a manager but a facilitator. You can't tell a community what to do but it is great if you can tell a community what it did and what can be done. (Interviewee 9)

Instead of community management, LibreOffice calls this area marketing. Public communication is used to raise awareness of the software, making people aware of the project, and what happens in the project. It is also used to make contributors feel rewarded and welcome in the community. These are many tasks that show that free software does not happen automatically, but that a lot of work is involved before and after the production process.

LibreOffice needs more than people who write code, translations for instance and native language projects that are done purely by volunteers. Highlighting the work of these volunteers is a vital part of LibreOffice.

> [Y]ou can do something quite small and it makes a big difference. You don't have to spend six months getting familiar with the LibreOffice source code. You can update an icon theme[12] and that has a major impact on end users. You can update one piece of documentation for a commonly used feature for, let's say mail merge[13], makes a big difference for end users. Translating user interfaces, as well. But especially in terms of design, there are lots of ways that people can just jump into the project for even just a few days or weeks to make a big difference and then millions of people around the world benefit from it and I think that's important. We have seen this with the notebook bar[14], the updated user interface design in LibreOffice 6.2: it's largely in the work of a handful of people in their spare time. But it's been enormous, people love it, everybody loves it. It has generated interest in the software. So that's one of the things I say to people about how you can get involved: You can make a big difference, even in a short space of time, and even in a very large project. (Interviewee 21)

Marketing, while often less visible, can articulate these achievements and generate interest in the software.

> An office suite is not as exciting as other free software projects. Projects around code, Linux for example, are far more dynamic in terms of developments and features.

12 Icon theme = a set of icons with a shared design or look. An icon is a graphical representation of a function, file or program.

13 Mail merge = a feature that allows mass mailing. Variables in a text document enable users to send the same document to multiple recipients by importing specific data costumised for each recipient.

14 An new form of organising commands in LibreOffice, alternative to the classic rows of cascading menus and toolbars.

> LibreOffice in contrast is a more serious piece of software with a large codebase that is not changed that much. (Interviewee 11)

Marketing work is done to highlight the achievements of the community, as well as to the outside to show what forms of contributions are possible. Presence at conferences, handing out flyers and stickers is a common strategy in the free software world.

> Anybody who contributes in any way gets a sticker. Ok, that's not an amazing thing but we don't have the resources to send anybody laptops or big bulky items but it's a way of expressing our thanks. Other free software projects do similar things as well. (Interviewee 21)

Stickers seem to be a banal thing, but people take pride in their stickers. Whether at FOSDEM or at hackfests, people in the f/oss scene like to put a lot of stickers on the lid of their laptops. They show which projects they like, or to which projects they have contributed. It contributes to a feeling of a broader community that is united in spirit and a way to express belonging are these popular stickers that every project offers at conferences. The people involved in LibreOffice marketing also tell people about the different parts of the project and how newcomers can get involved by giving them flyers with links that show them how to join. Every year, the marketing team writes an annual report that summarizes activities across the whole project, so that everyone involved gets an overview of what is happening.

The nature of being a distributed and broad project with diverse areas is also affecting this part of the project. Another challenge for LibreOffice to coordinate its community comes with one of the reasons why it is successful worldwide. Local communities with a language that is not spoken by many people in the project tend to not get recognition. I have been told that the community in Japan is doing great work that is reflected in download numbers and contributions in Japan. But they do not communicate back to the centre of the project so that they understand what is going on. Sometimes the project is too decentralised to coordinate it centrally.

> We are trying to increase, to expand the contact with the local communities because we discover that in general, we have big communities and because of language issues they are not in contact with the international project. We are

> trying to grow more the local communities. We are trying to share the message that the level of English is not important, that the important thing is sharing with the others. Without sharing the information, as a project we can't cover the needs for the local communities. The attempt is to connect the local communities better to increase [their] interaction and to grow properly our project. At the same time, we try to push the experiences across the communities and the aim is to grow better the local communities. (Interviewee 34)

Coordination as internal marketing is a tricky task due to the scale of communication. There are weekly meetings for different sub-projects in LibreOffice: the design community, the marketing community, QA. And others have their weekly meetings and they send out an email to the mailing list. The marketing publishes a monthly recap blog post but it is hard to keep track of all developments. Another issue which also affected my research is that there are many different tools and communication channels used. Some people favour a specific way of communication and do not want to change that. This articulation work is not directly working on the software, but is still an important aspect of coordination for this free and open source software project, as it contributes to a feeling of belonging to a community, eases the problem of a lack of direct contact between local communities, and creates understanding of the work that is done in different areas of the project. F/oss projects can sometimes forget the importance of articulation work. Writing code is often seen as the most important area of a project. The notion of meritocracy in f/oss is linked to the ability to write code that others deem to be of high quality. Commits can be easily counted and can be attributed to a specific person. Thus, technical contributions can be counted. Yet, these "invisible" practices are not counted. In comparison with the detailed graphs and charts that zoom in on the development process, efforts to keep a community intact are not counted in the same way. They are not attributed directly to a person such as a bug fix would be. They are explained and portrayed in the annual report or on the wiki. But they cannot be broken down in number, as in software development. Yet, without the practices of holding meetings or mentoring newcomers the practices of developing software cannot become collaborative. This invisible work offers a moment for translations to happen and for associations to be made.

7.4. Coordinating with software

Many easy hacks involve small bugs. Fixing bugs is an integral part of free and open source software. The appeal of the innovation model of open source software lies in the promise that the more people participate, the better the product: 'Given enough eyeballs, all bugs are shallow'. (Raymond, 1999, p. 30) A functioning bug fixing process is vital for LibreOffice:

> I was a user of OpenOffice.org and I reported some bugs but... let's say it was not the best experience of my life. I never received a response and the issues I was working on were never fixed. When LibreOffice started I did not hesitate to move over from OpenOffice.org. There was no comparison to Openoffice.org. My patches were looked at and discussed. I received feedback and my patches were accepted quickly. That was the start and I became a regular contributor since then. (Interviewee 12)

The management of bug fixing relies on standardised practices that are necessary to coordinate this part of the project. To coordinate the fixing bugs LibreOffice uses a programme called Bugzilla. Bugzilla was developed by Mozilla and has been released as free software under the Mozilla Public License. It is used by a range of free and open source projects, with LibreOffice one of them. Bugs need to be reported in Bugzilla so that others know that there is a bug and what its nature is. Ideally a report contains a detailed description, under which circumstances the bug has shown up and what the problem seems to be. The good practice is to provide a short but descriptive summary of the problem. It is hard for others to help out if the description is "I cannot print a document". Bugs that are not described well enough tend to get ignored. A good description has a better chance of being checked by others: 'In the Description, give a detailed list of steps to reproduce the problem you encountered. Try to limit these steps to a minimum set required to reproduce the problem.' (Bugzilla, n.d.) Not only can others understand the problem better, they can also prevent to try possible solutions that have already been attempted, and they can exclude possible problems: 'This will make the life of developers easier, and the probability that a bug is considered in a reasonable period of time will be much higher.' (Bugzilla, n.d.)

Software developers value clever technical solutions and appreciate clearly written documentation that shows appreciation for their skills, knowledge and the time they invest, even if they do it for fun and passion.

> [Documentation] [d]efinitely is important for several reasons. One is that you need to develop a culture around the software so that people understand the tool that they are using. It is also important to get a reference of all the software that you have. In the open source business we don't exactly follow all the rules applied to commercial software. We have a development that is driven by chaos. I mean you don't have plans, you don't have specifications, you don't have all the paperwork that precedes the development of the software. Most of the developers are temporary developers, they come and go. Often, they don't like to write documentation. So, we have a lot of features that are not documented. And the consequences are that people don't know how to use, people don't know that it exists, and the developer doesn't get the credit for having a feature that people use. Documentation is important not only for improving the quality of the software but also to improve the ecosystem where people exchange, use, interact. (Interviewee 33)

LibreOffice's Bugzilla is open to access. Anyone can open an account and the possibility to file a report is unrestricted. Collaborators have the equal right report a bug. Coordinated is this open system by a good practice how to write a bug report as well as by a computational logic of the system. Well-written bug reports do not contain any misspellings. This is not a rule to please a marked preference that programmers would have for orthography – even though the love for detail could also be reflected in other areas than coding, and especially the translators and those who write documentation sure do – but a necessity for coordination to function properly. If a bug's documentation contains misspellings, others may be unable to find it if they search for a specific word. Programmes like Bugzilla help to construct a more reliable development process. They discipline the users by following a specific structure. These are not automated processes that are expected to run smoothly in the background without getting noticed. FAQs, and manuals need to be written and workflows as well as information must be entered manually into the system to make it function. The practices of coordination are reciprocal acts between humans as wells as between

humans and the computer. The system affords that some specific information is filled in and norms should be followed so that other contributors understand the information. The coordinative performance of the software does not happen automatically. It relies on a translation between humans and non-humans through a specific grammar of the software. Similar to Agre's (1994) "grammars of actions" activities are divided into smaller units, which then can be ordered by a software system. This requires communication and manuals to explain the grammar of programmes such as Bugzilla. To report a bug, information has to be put into the system such as a name, organisation, the product and the product version, the part of LibreOffice that seems to have a defect, accompanied by an explanation as described. If the information is filled in correctly the software files it in the database and creates a time stamp and a ticket number, so that the system can order and sort the entries to make accessible for users. These practices are also typified by the fact that if a bug is resolved, its status needs to be changed manually. The software creates a translation by offering a standard language for all collaborators. It orders collaboration at the same time as it needs collaborators to stay relevant. If nobody follows the standard anymore, there is no translation left on which associations can be made. Software acts as an ordering mechanism. That is how collaborative practices are stabilised. What is also of interest in regard to stability, is the constant input that is needed to uphold collaborative practices. Latour highlights the need for permanent activity: 'It's the work, and the movement, and the flow, and the changes that should be stressed.' (Latour, 2004, p. 63) Bug fixing is not a process that is determined by the programme that is used to coordinate it. Bugzilla provides a grammar to successfully report or review a bug: a log in, categories to fill in, buttons to click. After the grammar is followed, the work of fixing it begins. The programme does not determine how important a bug is, this part needs to be decided by a team of people who contribute to QA (Quality Assurance).

> In [QA] we (. . .) analyse user reports. We do a pre-analysis of bugs to see which are important, which need to get fixed earlier. We make life easier for developers, so they don't have to spend time with testing. (. . .) If we have a new release and in a few days, we get many reports about the same problem we can say it is a problem that affects many users, it is a severe bug. Once we analyse the problem, we can say its's a

> regression[15] which was recently introduced in the code, we normally increase the severity to avoid having new regressions in the software. Developers work on new features, so it's kind of expected that they introduce new regression. If we find regression are recent, we want to fix it as quickly as possible. Some bugs affect all systems, then we consider it more important because it affects more users. (Interviewee 19)

Coordinated management of bugs shows an entanglement of technology and sociality that replaces direct interaction to some extent. It shows coordination is facilitated by a software such as Bugzilla that affords a certain grammar, but there is work to be done around it.

The resulting database can then be used to manage bugs and it is also used to evaluate the activities of users. By following the grammars, all the data can be collected statistically and analysed. Collecting and coordinating are inherently intertwined. The performance calculations show the number of bugs in a certain period can be compared, which areas of the project the bugs belong to, the number of bugs between the different versions can be compared, how many bugs are resolved and in which timeframe, and subsequently the activity of users can be measured. As every contributor logs into Bugzilla the system attaches their activities to tasks. Who has reported a bug, who has solved a bug, when was the user active in the system is recorded, stored and ready to be looked at. With a few clicks, statistics can be produced that show the most active users and most active periods. It is the same computational logic of grammars that social network platforms use to sell user data to advertisers, but in the context of free and open source software these activities are embedded in a model of governance that reflects accountability, responsibility, and transparent structures and processes. The collected data is not sold to other companies, nor is it used for other reasons than to facilitate quality management. Possible data practices that we see in other areas that rely on the extraction of economic value by transforming user activities into labour are not activated in the so-called LibreOffice ecosystem.

15 A software regression is a software bug that makes a feature stop functioning as intended after a certain event such as a new version, or the change of the year.

The statistics are shown regularly to the community, at least once a year when the QA team presents last year's activities. As such they receive a performative character as they are related back to the community. The statistics also give clues about the development of the community: how many people are active, are their significant changes, etc. As it has been explained to me, the statistics give an overview of and an insight into the community for others who work on different parts of the project, instead of being used to monetise data practices.

7.5. Coordinating interests

Statistics are used for all areas of the project. All contributions require a login in the respective system whether that is translation, design, or QA. Therefore, it is automatically registered who makes the contribution. What these statistics show is that an overwhelming part of the contributions come from people who work for one of the companies in the LibreOffice ecosystem.

Keeping the project stable but permeable is a difficult task. Berdou's (2011) study on GNOME and KDE shows that in the case of these two free software projects, paid developers are more likely to contribute and to maintain critical parts of the code base and volunteers do the peripheral work. In the case of LibreOffice, it is also true that paid developers do the work on the most critical parts of the code. This hierarchy is not institutionalised, but is instead an effect of the time that the employees of the companies in the LibreOffice ecosystem spend developing the code. The codebase is so large and complicated that even experienced members who know about development do not make any changes because they do not have the necessary experience.

> Sometimes I do some development, just simple patches but sometimes I get scared of how big the code, the project is, I get afraid of breaking things, that's why I don't do development too much. I don't have enough knowledge of the code or C++ to do this. (Interviewee 19)

What the QA statistics also show is the activity of individual users and which organisation they belong to. The community of LibreOffice does not only consist of volunteers, but companies are part of the ecosystem too which has become typical for open source projects. When it comes to QA, the statistics highlight the big impact that these companies have. Around 60 percent of the development is accomplished by employees

of the companies. The people employed by companies who take part in the production of LibreOffice play a vital role in the coordination of the project. Because their job is to work on LibreOffice full time, they develop an expertise and deep knowledge of the software. These paid developers take part in the community like volunteers. They attend conferences and hackfests and they also function as mentors.

> Without the mentorship of the companies, it would be more difficult, of course. They work on the codebase all the time. Of course, they know more about this. But then without companies, the development process would be much slower. We have a new release every half a year with a lot of features. And most of the features are made by companies. That's the benefit of the environment we have. (Interviewee 23)

What bug gets fixed first is a question of negotiation. Those who report a bug can assess its importance, critical as the highest priority. The priority rating however does not determine how quickly the bug gets fixed. This is still a negotiation process where different interests come together.

> Sometimes we [at the QA team] think it's a big bug but developers don't think so. Then we have a give-and-take discussion and we have to find an agreement then. And the companies involved, they have their own interest as well. From the QA perspective a bug might be important, but from the companies it is not that important, they have other priorities. It's about finding a balance. Let's say we find a regression that was introduced by an employee, and we push hard to get this regression fixed. They don't have time to work on this regression because they are working on their features. There are different priorities on each side. (. . .) Companies have their own interest. Sometimes those interests conflict with the community's interest or with other companies' interests. That's the problem but that's kind of expected. (Interviewee 19)

A process to coordinate the repair and maintenance of the software is not only a technical process. It also shows how the business model in f/oss gets reflected in the coordination of the technical development. The interest of companies is to develop LibreOffice in a certain direction. In negotiation moments like these a company that has the resources decides according to their needs and not necessarily what the person who reported the bug wanted:

> [I]n the end it's up to the developers to fix the bugs or not. Even if we push hard to get them fixed, if they are not interested there is nothing we can do. But here it is different, we have paid developers but we also have volunteers. You can't force a volunteer to fix an issue. The last word is with the developers if they fix it, if not, there's nothing we can do. (Interviewee 19)

The coordination of bug fixing becomes more than a technical necessity but influenced by different interest. To mark a break within the project between companies and volunteers would be improper. The distinction between volunteers and employees was already a difficult moment during the fork of OpenOffice.org. A former Oracle employee who worked together with the community told me how he felt left behind, uninformed, and betrayed as they were confronted with the fork. Most employees felt part of the community even though they were no volunteers. They read messages on the mailing lists after business hours, tried to help out on the weekends, took part in community meetings, and travelled to conferences. One argument presented to me was that the distinction between volunteers and employees is insignificant due to the passion and commitment shown by contributors.

> [Volunteers and employees], that's not really separable. That's not a helpful distinction in my experience. Most of the people I know who [work on open source software] as a day job, they do it out of conviction and do it well beyond that. A day job is about six to eight hours, but they do so much that you could say: actually, that is a volunteer. They do so much after work which isn't necessary for the job they're doing there. Or they are even sponsored and promoted by the companies where they work, that's a grey area. I shy away from making this separation voluntarily - involuntarily. (Interviewee 36)

A clear line exists between companies that are part of the what is considered an ecosystem and those who are not. The companies who contribute to LibreOffice do not only negotiate with other contributors in the project. The coordination of the development also draws interest from other companies who do not contribute to LibreOffice.

> [These companies] just want to claim the product and blame the free software project when it goes wrong, file the bugs upstream and then create pressure to have them fixed. There's a lot twisting one's arm via emotional nonsense to get bugs fixed in LibreOffice. And you get this from large

> companies, filing bugs in Bugzilla with anonymous gmail-accounts. And then asking why this isn't fixed. And they are shipping this as a product to paying customers and they expect free fixes, free support but do not contribute back. (. . .) This is horribly anti-social. (Interviewee 25)

Negotiation between contributors is an accepted part of the collaboration. However, negotiations by those who do not contribute is not an accepted strategy. Even more so when companies want to abuse the f/oss model by selling the software without offering a support structure and without access to the community. The companies within the LibreOffice ecosystem can negotiate as they built up their reputation by being active and visible. Employees are active in IRC channels that discuss LibreOffice and they attend hackfests and conferences. Coordination between companies and community runs on a technical level, a social level, and an institutional level.

Credibility also needs to be built in term of knowledge of the software. If the community needs help, the companies' employees should be there to assist. Ideally, volunteers get help and feedback from the employees directly. While this form of coordination is sometimes spontaneous, it can be planned and organised by bringing people from different parts of the ecosystem together. At the hackfest, I witnessed several occasions when employees or one of the heads of the companies helped volunteers with technical problems. The volunteers were grateful for the help, as they would not have the same knowledge as an employee whose job is to use and develop the software daily.

Negotiations are not necessarily based on business interests. Different opinions are also the case between the TDF staff and volunteers.

> What we see quite often is that if users have bugs that affect them, they want to get them fixes as quickly as possible. Sometimes these users are angry. In average we get 150 reports each week. Some of them are critical or severe but others are lying around for some time. Some kind of users, after nothing has been going on, the users comes again [and ask about the bug they reported]. (Interviewee 19)

I have shown above how a quick response to bug reports by newcomers was important for them to make further contributions to LibreOffice. Similar to the negotiations with companies the fixing of bugs is also a

negotiation process between mentors and volunteers. The QA team decides together which bugs need to get fixed first and then they have to convince the developers. The important caveat is that everyone who wants to can join the QA team. In the negotiation with volunteers, the QA team cannot just follow their interest however. The goal is to keep volunteers, and therefore they are not given petty, irrelevant tasks that might lead them to lose interest. It is not the volunteers but often the employees paid by The Document Foundation who take care of the minor tasks because nobody else wants to do them.

> I do the boring things, delegate the things that people want to do. I think it's always like this in teams when you have volunteers: they get to pick what they want to, and I do the boring things. (Interviewee 26)

The heads of the different sections employed by TDF have a significant role in this open distributed system. They became coordinators and facilitators for others. What they do qualifies as articulation work (Strauss, 1985) as they articulate what activities need to be done, they teach newcomers the necessary practices how they have to be done. This 'supra-type of work' (Strauss, 1985, p. 8) is a vital component in the distributed collaborative project LibreOffice. Volunteers are semi-autonomous in their work. They pick and choose what they do but they also get engaged in the community to exchange and they need to report back so that others can syntonise with them. This articulation of their work is partly facilitated by the community and to another part by the heads of sections. Through articulation, permeability can be achieved to keep the project robust. This results in a combination of not restricting the contributions that volunteers want to make with the offer to help and assist as quickly as possible.

7.6. Coordinating teams

LibreOffice is a diverse project. As we have seen it combines many different interest that play a vital role in the coordination of the project. What also needs to be coordinated is the work of the different teams to produce a complete version of LibreOffice that can get released.

> We are not a developer library. We have one but LibreOffice is not a developer library or a sever component that is directed towards a very specific target group. Our package is directed towards the simple user as well as towards huge

> companies. The same applies to the areas one can contribute to. In a developer library you won't find certain things whether that is development, marketing, QA, localisation. We have a broad base of users and contributors that we can call on. It is a main advantage that we have such a broad scope. However, that also means that we have to work on many different topics. (Interviewee 20)

Easy introductions are given to newcomers in regard to the practices that constitute the project. These introductions are given for a specific area, and thus it is hard for contributors to get an overview of the whole project that consists of several teams.

> When I joined [the LibreOffice project] it was kind of overwhelming because I didn't know the interaction between people and everything was new to me. I didn't know how the developers work or how the translators work. It has worked out quite well so far and has become easier. (...) [N]ow I feel like I am part of the community. (Interviewee 26)

To work on different topics means that it is hard for people to get meaningful insight into areas where they do not contribute. To get a holistic perspective in turn helps to create a community feeling. This community is not a homogenous collective. The coordination between the different teams shows the difficult character of togetherness.

> [T]he foundation decided to be quick (. . .) with two releases per year. This is good for the software but not that good for the documentation team. The speed of updating is very fast and the process that we had requires a high level of quality and revision, a long cycle of updating the documents. And we miss all the targets. So, the documentation is too old and we are rushing to update for the latest release. With the team we have at the moment it is not feasible. So we have to devise a new process. The idea is to follow the software and release whatever we have at the deadline - even we don't have a perfect text. But it is better to have a sub-par text than no text at all. (Interviewee 33)

TDF has decided on a release plan that works well for developers but complicates work for others. The software product changes its characteristic, hence the whole network changes. Callon (1991, p. 135) noted that 'intermediaries describe their networks in the literary sense of the term. And they compose them by giving them form'. If the software release plan is changed in the interests of one group, the practices need

to be adapted. These instances show the mutability of collaborative practices. When the software is pushed, core practices become questioned, changed or ignored in order to follow the release plan.

> [I]f we need to deliver the documentation fast, we need to shorten the cycle of the revision. It may be necessary even to have a set of high-skilled documenters who we can trust to write and we don't need to review it. We trust [all our volunteers] but we only can do that with people who we trust in terms of seniority and commitment. Somebody who just wants to have fun on a rainy Sunday, that is not what we want to have. (Interviewee 33)

Many in the project who do not work in the development team complained about the priority that is given to writing code and the neglect of other areas.

> There is a conflict insofar as it is all about writing code and if no one implements the code, nothing gets changed. Everybody can say that it is a great idea but [if the code is not written] nobody does anything. The developers rank first. For me, this is a principal problem of open source [software]. Developers rank first and they decide about what gets done and how it gets done. Normal users have to gain the competence or they have to have very good reasons so that things are changed. (Interviewee 18)

Writing code naturally plays a big part in a software project. Communication is needed with other areas so they can coordinate. The rhythm of work is different in the areas. Documentation needs to wait for the code to be ready to document it, designers need back and forth communication as they are used to project pitches, brainstorming and discussions at different stages. Even though people are in the same community, they come from different fields of work that is based on a specific way of doing things. Understanding the others can be a difficult task.

> There is a bit of friction with developers sometimes because they are not technical in the same way. I mean they are focusing on code and not on infrastructure and the scalability of infrastructure in that sense. So sometimes (. . .), I mean it's one of the natural things that you disagree best with people you actually can relate to. I mean if someone you have no understanding whatsoever, there is nothing to disagree about. But if you actually understand you know… There has been a bit of friction but no fights or anything. It's

> just that sometimes that they are a bit vocal about what they want. Sometimes there's disagreement. And the other community with which there had been problems where the translators or documentation: people who don't have much technical background at least not in terms of infrastructure or development. It's a bit harder to understand what they want. (Interviewee 26)

Tensions between teams arise due to the priority given to the developers. But then the community can be divided into two other groups: those who are skilled in coding and those who work on areas that are not code-related. So, people wish that coordination between teams would lead to a deeper understanding and a more shared collaborative effort where everyone, developer or not, can take part in the decision-making.

> It would be great if we had more tasks where members of different areas work together, for example translators and designers, or developers. Sometimes one team produces something and the other needs to implement it and then complications can come up. So, one team does not know what the other really means. If we would have more occasions when two or more teams work together directly, then this would not happen that often, I guess. (Interviewee 32)

7.7. Summary

Coordinating LibreOffice has shown to be a set of practices to facilitate collaborations. These coordinative practices consist of organisational, technical and social layers while these layers are not fully separable from each other. From a technical point of view, the software that is used to coordinate quality assurance shapes the practice of bug reporting. It asks for a certain grammar that needs to be followed and this grammar is then expected by those who read the reports. If this grammar is not followed, collaboration cannot emerge. LibreOffice's codebase also acts in coordinating collaborations. Its large size can be frightening for possible collaborators. They may not know where to start, as it is hard to find the right entry point. Therefore, the practice of easy hacks has been introduced. Easy hacks work around the large size of the LibreOffice code base by providing insights into smaller sized technical problems. They allow to make new associations as ANT would argue. Yet, they also generate trust amongst collaborators. The

importance of the interrelations between collaborators that is emphasised by peer production becomes apparent here. Other practices such as mentoring also connect with the social aspect. It shows the care for the software as much the care about the other who contribute to the same project. Alami, Cohn and Wasowski (Alami et al., 2019) have provided a similar interpretation of f/oss projects.

From a perspective that is more concerned with the sociality of the project, the importance of trust is shown as well. The continuous discussions and interactions during online meetings, in chatrooms, and at conferences are central. Mostly, they concern technical problems but they also have a important function as they create convergence (Callon, 1991) around the software. People discuss software and they learn from each other. The collaboration between the those who are engaged in these processes are not one-off moments that could be captured in a simulated trade-off between individuals. Countless are the connections that accrue from the various approaches that the people involved bring to the project. The constant communication also has a social aspect. It forges associations and allows negotiations.

The organisational aspect is often forgotten. I have characterised it as invisible practices, referring to Susan Leigh Star's (1999) concept of invisible work. Internal communication to keep the community alive has presented itself as an important part of coordinating the project even though the impact of these practices is hard to measure. Yet it creates an overview over the large project, it connects people with each other. The coordination that is needed goes beyond the technical realm and rests on invisible work, creating a sense of community, as much as on negotiations and technical expertise.

Altogether, standardised practices have shown to be a decisive element for coordinating LibreOffice. However, the trust that can be established amongst collaborators through knowledge and outreach to other collaborators allows for circumventing standards. It becomes visible how flexible the application of practices can be, while standardised practices are needed to ensure that collaboration is possible. If a collaborator is trusted to have the significant knowledge, their work is sometimes not reviewed. These coordinating practices are thus standardised but allow a certain flexibility. The robustness of the project is based on a set of diverse sociotechnical practices. Coordinating a collaborative project

offers alternatives to hierarchies or a bureaucratic system. Tasks are not assigned to people but they are negotiated through the introduction of various practices. Such a form of coordination considers differences between collaborators. It requires coherence as well as plasticity to become robust (Star, 1993).

8. The politics of collaboration

This last empirical chapter addresses the politics of LibreOffice. In Chapter 1 I have shown how f/oss combines two different positions on politics. While free software offers an explicit position on the political potential of free software through sharing knowledge, in open source software any political attitude is generally denied. This division between advocates of free software and those of open source software has softened over the years: 'We should not underestimate that is there to some extent but most of us try to bridge that gap and try to move on' as one long-time member of the f/oss community (Interviewee 3) told me. Even though the traditional dichotomy between free and open source does not carry great weight any longer, there is a division within f/oss regarding politics. One faction denies that f/oss has a political program; Coleman (2004) calls this the political agnosticism of f/oss. The other recognises the political significance of f/oss for the distribution of resources and makes an explicit political stance.

A significant amount of studies (Coleman, 2013; Coleman & Golub, 2008; Himanen, 2001; Juris et al., 2013) on f/oss refers to the second faction and explores the importance of an ongoing debate of ethics, morality and its political significance within free and open source software projects and in the wider scene. Kelty (2008) however argues that the f/oss scene understands itself as a community that is foremost characterised by its shared practices rather than politics. However, he underlines that it is without a doubt that free and open source software has political significance. As a set of practices, it expresses a technological vision to change the world. It is, as Kelty (2008, p. 144) put it, a materialised 'vision how economy and society should be ordered collectively'. According to this analysis, f/oss is inherently political but it is expressed in forms that are not considered to be political acts, nor are these practices necessarily acknowledged as political by those who deploy them.

To ask for the politics of the collaborative practices in LibreOffice, I refer to a broad definition of politics. Heywood's (2019, p. 2) conception of politics as the 'activity through which people make, preserve and amend the general rules under which they live' provides a suitable plat-

form to explore the several notions and expressions of politics in LibreOffice. Specifically, I am interested in the impact of political positions of collaborators on the practices. I do not attempt to evaluate practices and politics against each other. Rather, this chapter asks for different political ideas in the project and how they are expressed in practices.

At first, different ideas of openness, access and participation are characteristic for the f/oss scene and for a wider digital scene that includes message boards and email lists. Openness, access and participation are all central elements of collaborative practices in f/oss. Openness stands not only for access to technology and technological production but also for the possibilities to participate in previously closed processes of cultural production, politics, and the construction of knowledge. The question of openness is fundamental for free software. It does not only concern access to an open source code it reaches beyond the technical sphere and relates to the possibilities to participate in a project by addressing its ordering systems.

Secondly, different political ideas within LibreOffice are discussed. This part will focus on the existing tensions between advocates of free software and open source software[16], the question of political agnosticism within f/oss, and the role that the idea of a community has given the growing importance of business interests within the project.

8.1. Ordering openness: The tale of a mascot

Differing ideas on openness, access and participation within the f/oss scene have manifested themselves when LibreOffice started the search for a mascot. Most free software projects have a mascot: Java has the jumping and cartwheeling cartoon character Duke, the penguin Tux charms the users of Linux, the open source browser Mozilla chose a fearsome red Tyrannosaurus Rex, the chameleon Geeko shall reflect the flexibility and choice offered by SUSE Linux, and a wildebeest represents the free software project GNU. Maybe the playfulness that hackers show in coding and everyday life (Coleman, 2015b) is also expressed through the medium of a mascot. LibreOffice decided in 2017 that they wanted to have a mascot too. They announced the start of a

16 See chapter 1 for details.

competition to find a mascot on their blog. Before that they had asked on the mailing lists about input to build some categories of concepts associated with LibreOffice: the categories were freedom / openness, speed / improvement, intelligence, cuteness / seven lives. Some examples were also given for each category but those were all mere suggestions. Those willing to participate could enter the contest, the competition was not restricted to members. Everyone could send in a design with a name and a short description that explains the idea. The process was designed to be fully open with very little ordering structure. The only restrictions were for technical and for institutional reasons: The mascot could not violate LibreOffice branding guidelines (The Document Foundation, 2019a). Amongst other things these rules simply state that "LibreOffice" is the product name and not any other, that the Document Foundation is the name of the foundation, and it specifies a certain visual order and appearance for the logo. And for legal reason, it was made clear that the design should be completely new, even if based on some initial artwork, as it is customary in open source. In this case it must be licensed CC0, a creative commons license that can be used to waive copyright and database rights. It was announced that the results will be pre-selected by a commission named by the Board of Directors, and amongst those finalists the community will vote for a winner with the caveat that it will be confirmed by the Board of Directors. This procedure was chosen to avoid problems with crowdsourcing processes in the past in which the wisdom of the crowd was realised in form of trickery and prankful playfulness[17].

The Design Team monitored the discussion amongst users, gave feedback to the people who sent in their designs, made clear that the rules of the competition need to be kept in line with for more than two months; almost 300 images were sent in. After the deadline complaints about the structure of the process set in. The Document Foundation was criticised for the combination of design-by-committee, crowdsourcing and majority voting. The first suggestion started on the blog about the problems of the decision making process and its lack of openness. The Design Team apologised to the community in public. They admitted that the Board of Directors has chosen a structure for

17 Users of the popular website 4chan rallied behind voting for a school for hearing impaired children as the winner of a contest to host a concert by pop singer Taylor Swift.

the procedure that was not optimal in terms of openness and transparency. The participants had no possibility to influence the procedures and the pre-selection by the board was not transparent. However, the alternative presented by the Document Foundation to only let a few people decide which design to choose. A clear but very diplomatic answer followed to remind the Document Foundation about the openness required in Open Source:

> I strongly disagree that these are the only two options - I think that as the OSD community we should work together to improve the community design process. I don't mean this post to be overly critical of you or LO, I think you do fantastic work - but in terms of open source design, we all have a long way to go in developing and applying some kind of best practice for community design. It's a matter of finding the right process which includes and values the contributions of a community, and produces a high-quality end result which serves the desired purpose, and that the community is proud of. I simply think that the particular execution of the LO mascot design was not the ideal process, and I would love to work with you and others here to work out what can be done better in the future. So it would be great to discuss this at the summit! (Sam Muirhead, 2017)

This comment also shows that the ideals of openness stretch beyond hackers. Many different branches and people with different backgrounds are affected by the ethic of open collaboration and are willing to defend those publicly. It plays its part that people like open source developers or contributors who are used to open collaboration and meritocratic organisations, easily express the anxiety that power could potentially corrupt those who enjoy privileges and block conditions for public self-development as well as institute a rigid form of vertical authority emerges from time to time. In the 1990s it was a running joke on the discussion system Usenet to express your discomfort over the potential for corruption by meritocratic leaders played out, usually stated as a denial: 'There is no cabal.' (Pfaffenberger, 1995) The discussions about the search for the mascot spread over many websites related to open source or free software and also reached the image board websites reddit, 4Chan and other related websites:

> It's looking more and more to me like our votes never mattered in the first place. They were always going to pick the winner themselves. (Cuprite_Cane, 2017)

Starting roughly around 2005, Anonymous was a name used almost exclusively by some 4chan users to troll, harass, humiliate, prank, and sometimes ruin the reputations of chosen targets. The idea of openness is different from those represented in the open source ecosystem. It is not clearly defined in its dynamic tension that builds 'between cool and hot, openness and secrecy, pranks and seriousness, and predictability and unpredictability' (E. G. Coleman, 2012, p. 14). A decision making process that involved many top-down elements such as the one chosen by the Document Foundation, and the increased importance that it was such a well-respected project in the open source community, put it directly into the focus of Anons.

> In case you were too fucking busy, here's what happened last Thursday:
> >libreoffice opens contest>many shitty nominees
> >libbie amongst them all
> >everyone votes for her
> >vote closes
> >reopens as booru
> >no anime girl
> >no green spurdo
> >no Libbie
> >anon finds out the penguin mascot is a traced stock art
> >another anon finds the owl is also a ripoff
> >everyone votes shitty penguin as protest
> >libreoffice tries to make the octopus win but fails
> >wipes imageboard
> >tries to pull pr stunt
> meanwhile
> >tyson upset over libbie getting outed for bullshit reasons
> >anons and drawfags make lots of OC for tyson and for /tech/
> >sends them to Tyson
> >tyson appreciates and also tells anon that other artists furries got upset about him releasing free art (pic related 3)
> >another anon says that other people didn't want him to enter the contest because he'd make them look bad
> HOW DEEP DOES THE RABBIT HOLE GO?
> WELCOME TO THE GAYEST, MOST POINTLESS CONSPIRACY IN THE /TECH/ HISTORY STARRING:
> >Tyson Tan
> >Libreoffice Design Team
> >Pajeetguin
> >Libbie the Cyber Oryx
> FEATURING:
> >Drawfags

>OC creators
A DOCUMENT FOUNDATION PRODUCTION
(Anonymous, 2017)

Spam, flame wars on message boards and via emails, as well as denial of service attacks followed; practices of protest that are not in the repertoire of The Document Foundation. The openness of its governance structures reflects the stakeholders involved. Openness as a de jure standard format requires to stability. The practices of openness are very much geared towards keeping the production running while nurturing a community towards this goal: mentoring, bug fixing, easy hacks, community meetings, conferences, and so on. Playfulness and pranks are not necessarily in the focus and when these practices that arise from a momentous, radical understanding of openness are confronted with a large project like LibreOffice that bases its openness on stability and decision making processes that reflect its governance model and the involvement of companies, compromises to the idea of pure openness are necessarily made. In the end, the Design Team apologised for misunderstandings and mistakes that have been made – not without complaining about the tactics used by some groups and individuals – and shut down the process (Vignoli, 2017).

> [The mascot situation], we handled it badly. We admitted that. Lots of free software projects have mascots: GNU has the wildething, Gnome has the foot… So, we thought, let's have a mascot. We have a very serious document logo. That's not super exciting and that's great for businesses and end users but we can have a mascot the community can use, we can put it on t-shirts on events, make it fun. I think the idea was fair enough but then the implementation… (Interviewee 21)

All actors that I asked about their reflections unanimously agreed that LibreOffice had made mistakes in the process of the search for a mascot. They lamented unclear communication of the structures of the communication and a switch of strategy during the process that started the hijacking of the competition by 4chan users.

> [Regarding the mascot], we had a communication problem. The board (. . .) made changes to the procedures [that had been planned] and it counteracted everything. It obstructed all transparency and dismissed many contributions. Rules have been changed retrospectively, all the mistakes one

> could make have been made. Transparency is very important; it doesn't work without it. Irrespectively it could have worked, would it not have been for trolls. They became aware of what happened, and they destroyed it. (Interviewee 18)

The people involved in organising the competition were aware of the lack of transparency that is required in free and open source software. This notion does not lose its importance beyond producing software. It stretches from practices of doing collaboration to those of ordering collaboration. It becomes apparent how the political significance of free software is a view on society as a whole rather than on software.

The reflections on the search of the mascot not only underline the importance of openness and transparency as important ethical elements. They also show a fault line within the f/oss scene that become apparent during the search for a mascot.

> Somebody has to make decisions in the end, and we were trying to do this as open and democratic as possible, open to all submissions, massive votings [sic], anyone could come and upvote, downvote and comment. I completely support of openness and transparency, but you do need some people to guide them and make a decisions. We can see this in other free software projects: If nobody makes a final decision, things can just linger. People say that one of the reasons the Linux kernel has been so successful and hasn't been forked in a millions of projects and is still going well today and used on everything, is because Linus Torvalds is just the benevolent dictator. It's a free and open source project, and meritocracy, anyone can submit patches but you got a guy at the end who is well respected and can take very authoritative decisions, saying "no, we are just doing it like that". Is that democratic? I don't know but it works. (Interviewee 17)

These interview excerpts highlight different production practices in free and open source software. This interviewee refers to the central decision-making practices in the Linux Kernel project. Linus Torvalds makes all technical decisions in this project. He does so in a transparent manner but there is no possibility to negotiate the rules for the collaboration in the Linux Kernel, a prerequisite for the model of free cooperation that Spehr (2007) has conceptualised.

> We opened it up enormously to pretty much everyone to submit ideas. Which is a good, free, open, democratic thing to do but we were swamped with ideas, with very low-quality

> ideas, we were swamped with porn, and 4chan got all over it. (. . .) [T]he 4chan technology board (. . .), they were just trying to attack it from every angle: denial of service attacks, uploading pornography, stuff like that. So, we said, ok, we are just having to remove a huge number of entries here, we had hundreds of submissions and many of them were terrible or just completely inappropriate, so we started to call them, remove massive amount of them. Then people got pissed off, saying: Oh, well, you're not democratic and this is a dictatorship, you are just removing the ones you don't like. We said: No, we just have to remove the ones that are just not appropriate. (Interviewee 21)

The need for ordering practices collided with the request for radical openness. LibreOffice did not provide that radical openness. First, for legal reasons concerning branding guidelines and later for implementing a stage of preselection that was not transparent. After that LibreOffice was confronted with a form of collective action that probably attacked LibreOffice and boycotted the project for the lulz (for a in depth explanation of this phenomenon see Coleman, 2015a). They started to troll the project as a reaction of not being provided with an idealised version of radical openness. Pranks of various kinds were used to make the point that LibreOffice had broken with the ideals of openness.

> [W]e kept trying to narrow it [the number of submissions] down but then were being accused of being non-democratic, even though we wanted to put it to a vote. I think we were being naive in how it would work and underestimated just how much time 4chaners have and... yeah... If we ever do it again, we do it very differently. Still make it democratic but maybe have only members of TDF vote for it, for instance. (Interviewee 21)

The people of LibreOffice that were involved in the process apparently do not share the enthusiasm for sabotaging a process they have addressed in a serious manner and in which they have invested time and effort. This points towards diverging ethical considerations related to openness and transparency. It also shows different ethical ideas on which methods to use and which approach to take for protest in a digital milieu that brings together software developers with other cultural phenomena such as pranksters and trolls.

In addition, the search for the mascot also highlighted distinct ideas on how to implement transparency and openness in practices within LibreOffice.

> Could we have done it in a better way? Yeah, sure. (. . .) We could have done it in a very democratic way. This means [to] take all the contributions and (. . .) give feedback to each one. And if someone copies a picture from the Internet and the community still wants to have it, then I could still say no, sorry, that picture is stolen. The board did not want to get into a situation like that and they screened and filtered. (Interviewee 18)

8.2. Opening documents

The setup of the LibreOffice community shows that the idea of openness and sharing is not only an idea that is rooted in a hacker ethic that can be traced back to the Twentieth century. Technical openness in terms of interoperability played an important role in attracting people to LibreOffice. Community members told me that they have become users of LibreOffice because it offers interoperability. They could not open and read old .doc format files with a newer version of Microsoft Word and started to look for ways to solve the problem.

LibreOffice is interoperable with all version of the .doc format and can open certain formats that cannot be read by Word. Just this technical feature of being open and making practices of interoperability possible is enough to attract users of the technology. Openness and interoperability became a main feature for collaborative practices. The benefits of interoperability are that users save time and money because adopting the software to your needs is possible as openness is provided; it fosters innovation as it can be shared; and it furthers public policy goals instead of a corporation's interest.

> It is easy to convince people that open standards and the open document format are important. But if you need to install software to read a document that you have just been sent via email that is a different proposition. But if you have a browser and you just need to follow a link, and edit the document in your browser, you don't need to install LibreOffice to make use of it. That means we can bring interchangeable open standards to everybody in their browser. (Interviewee 25)

The openness of the document format that is used by LibreOffice contains practical considerations and an ethical idea. LibreOffice is not just a playground for software developers, or a community to hang out, or a ground to express political ideas. It is all these things combined. The above reflection shows that actors in LibreOffice understand people outside of the f/oss realm can be best convinced about the importance of free software by providing them with tools that are easy to use. It shows the belief that practices that are easy to learn can convince users about the political significance of free software.

8.3. The politics of software

Even though the project wants to convince users to use free and open source software not everybody involved in the project is a user of free and open source software. There are many who use proprietary software are software with proprietary elements such as the messaging service Telegram was the main communication channel for LibreOffice. Fast and constant communication is so important for LibreOffice that a ready-made platform that can host several channels of large sizes and that allows instant encrypted communication has a central function – even if some of its elements are licensed as proprietary.

> But TDF has this position where as an entity it makes more compromises than me as a person but the good thing about that is that it's possible to reach out to more people. And reaching out to more people, I see that as a compromise to reach out to all people and that's one way to bring them in. (Interviewee 10)

The sentiment to lower the entry barriers to introduce more people to the project has been expressed several times during my fieldwork's conversations and observations. Elitist thinking and the feeling of moral superiority based on the knowledge and expertise to use and produce f/oss was subordinated to the realisation that an elitist project cannot become a mass project. However, some collaborators do not want to soften their personal approach to proprietary software.

> I guess I am one of the most extreme at TDF when it comes to free software. (. . .) But since more than 15 years now I don't use proprietary software and I am very strict about that and I want proprietary software to die. When something like a bank is trying to force something on me, I try to find an

> alternative. And if there is none, I try to find another bank or system. I guess I am quite extreme. (Interviewee 14)

Most of the daily communication in the project runs on Telegram but not all agree with it. There is no division within The Document Foundation regarding this topic but for some collaborators using proprietary software is against their beliefs. In such cases, work arounds are needed to accommodate those who do not want to use proprietary software. In order to connect all collaborators a so-called bridge was built that connects the Telegram channels with corresponding IRC channels so that those who do not want to use Telegram can be included in the project's communication.
What the interviewee expressed is a true belief in free and open source software. It surpasses the frictions and schisms within f/oss by locating the political struggle in the contrast between f/oss and proprietary software. This belief is supported in the project and practices need to be adapted and redesigned.

> Within TDF I try to implement that in a way with like what we have talked before. Many communities, sub-communities like translators, documentation, and so on, they used to use Google Talk. Some still use Telegram I guess but I see that as my responsibility to show them an alternative. Now I think people don't use Google Talk anymore, they switched to Jitsi and I think that's a good thing. That's not something I want to force but show them that this is a fully free software stack, that's available on TDF's infrastructure, we are not giving away your data to Google, you don't need a Google account, whatever. I try to show without pushing even though my agenda is that I would like people to come in but I first show and then let people decide. (Interviewee 14)

There is a division with the project concerning the politics of software that can separate those who do technical work (coding or infrastructure) and others who contribute to different parts of the project. It directly concerns the practices of LibreOffice. If some would refuse to take part in specific practices because it would contradict their political ideas concerning software, this would be major issue for the project. Yet there is no friction because of that issue. Rather, there is a diplomatic approach to bring everyone into the realm of f/oss. In a project like LibreOffice that is not a project only for software developers this is a vital approach. Instead of confronting others directly with a political

stance, a different and more consensus oriented way to convey the message of free software is chosen. However it shows the politics that is inherent in the practices' materiality. Practices are not deployed on a neutral basis but their material foundations are eagerly contested political fields.

However, the political arguments presented so far in this chapter are all directly related to software. This focus on producing technology often gives reasoning to a self-imposed political celibacy in regard to all other political expressions beyond the software itself in LibreOffice.

> There's a lot of injustice in the world, fair enough, and fight it on your own terms if you want but let's not drag our free software projects into these big worldwide political battles. It's very, very risky. A lot of people join free software projects to get away from this. They are sick and tired of the politics and the infighting and you could say identity politics as well and they just want to hang out in a community of geeks and make software. (Interviewee 24)

This focus on technology is a form of supressing politics. Instead of understanding the relation of hackers to politics as political agnosticism (Coleman, 2004), statements such as the one above show that the collaborators are well aware that software has politics – both as being a political tool and as a canvas for politics. Yet, they attempt to channel the ongoing discussions about the norms and values of software into an idealised purified version of technological practices.

> A lot of people come here to have deep technical conversations with other people. Others come here because it's their more political belief that we need to liberate society by using free and open source software. And both of those are equally valid. Both are founded in a belief that technology is important for the future development of society. It is just focusing on it in slightly different ways. (Interviewee 3)

Technical conversations are thus also political albeit they are not necessarily considered to be political even by those who engage in these conversations. What has become clear was that within LibreOffice there is a consensus on focusing on producing a free and open source software office suite even though not everybody that I have spoken to consider this to be a political act.

> What would put me off [LibreOffice] is too much politics that do not belong to the domain of the technology itself. (Interviewee 21)

What is supressed here is an understanding of politics that is not directly tied to software. Here, politics is characterised as something that intervenes the process of producing software. Politics come from the outside while technology is depicted as a neutral ground, free from political struggles. This luxury problem can be traced to the beginnings of free software. Hackers used to be mainly men from a similar social class[18] with similar resources, skin colours and education. The members of these homogenous circles could afford to concentrate their political efforts on writing clever code to change the world for the better.

Even if there is a visible change in the last few years, the free and open source software scene is a clear reflection that gender barriers exist: Contributors are mainly white men. As of 2016, only 8% of the Document Foundation's members were female. Around 10% of the governance position are taken by women. Despite the tendency to ignore politics, the people of the Document Foundation are aware of that - judging by the decisions that have been taken in the last few years. With Marina Lantini, The Document Foundation, is one of the few Free and open source software projects that has a chairwoman. The group LibreLadies has been started for women to share their experiences and to increase the visibility with the hope to encourage more women to join the project. But even if women join a free and open source project, there are societal forces that influence which part of projects they contribute to. At conferences, the most female speakers are generally to be found concerning questions of community and community management. In the LibreOffice project, there are significantly less women involved in writing code compared to writing documentation or translations. Even if there is awareness that LibreOffice is influenced by sociopolitical circumstances, there is the implicit guideline not to let politics drown relations. Rather, politics are only imagined as being directly linked with the production of f/oss.

---

18 Levy's book on hackers had a big influence on the perception of the history of hackers (Levy, 2010). He describes the first hacker movement to have had started at MIT in the United States of America in the 1950s and 1960s. There are alternative hacker histories though. The history of hacking in other countries than the United States is mostly underdeveloped.

It has become common practice for free and open source software projects to write a code of conduct to make a clear statement that reflects the changes within a community that is still male dominated but also getting more diverse.

> In order to keep TDF and its projects a fun, interesting and positive experience for everybody, we expect participants to follow the guidelines below.
>
> TDF aims to be a free, open and cooperative community. This means we expect collaboration from all the participants to TDF's projects, and for everybody to behave respectfully towards all others, including those that are different or think differently from yourself.
>
> Please be helpful, considerate, friendly and respectful towards all other participants. We don't condone harassment or offensive behaviour in our projects and or events. We consider it against our values as human beings. We're voicing our strong, unequivocal support of exemplary behaviour by all participants. (The Document Foundation, n.d.-a)

The Document Foundation's code of conduct is exemplary for many free and open source software projects. While it addresses the diversity and how participants in the project should deal with it, it is not a clear political statement but it is rather a guide for an undisturbed collaboration that should focus on producing software. It shows a recognition of the imbalance amongst collaborators, yet it attempts to keep the collaborative practices free from the influence which these imbalances might have. TDF oscillates between the need to ease the traditional imbalance between collaborators in f/oss and the wish to not become an arena for identity politics. It attempts to be apolitical by not engaging in practices that are not directed towards the software itself. The intention is to keep the collaborative practices free from other factors beyond making the software better.

> You know that a lot of people in the free and open source software world come from more alternative communities you could say. There's a lot of diversity which is great, it is a very rich mixture of people, a lot of people can bring their own personal politics into projects. I have seen some projects going down this line, getting way more political and it's quite scary. Other projects resisted. … [T]here are political battles going on, sometimes it looks a bit worrying

> if you see a nice free software projects and a mailing list full of people arguing about gender pronouns and removing source code because it has the word master or slave[19] even it has nothing to do with slavery. But people get very very passionate about this stuff and all these massive flames and arguments are up on these project mailing list and look at it you are thinking: . . . People, why don't you just focus on your project. But that's open source, it's public and open. . . . It may not be something specific to free software communities it's just that but everybody can see it in our cases. But that is a concern to me, the politisation of fee software. (Interviewee 21)

The statements above reflect how an idealised version of software production as a clean process that can distance itself from other forms of politics beyond the software itself function as a marker of identity. Software here acts as a symbolic template for the value system of the community. Its rationality and strict logic serves as a template for the whole project. While a certain messiness is allowed to negotiate the practices that are geared towards sustaining the project, political expressions that concern other areas apart from software are not characteristic for TDF.

8.4. What is a merit?

A political term that is widely accepted in f/oss even though it is not an automatic expression of software is meritocracy. A notion of meritocracy is a stable category within f/oss. It is often used to govern projects by deploying it as the decisive category for decision making. The appeal of merit is to base decisions on quantifiable criteria. Whoever writes the most code or designs the most icons can make the final decision should there be a conflict of opinions. Meritocracy and the idea of a neutral technology beyond politics link well together and unsurprisingly meritocratic ideals united the first hackers at US universities (Levy, 2010). A similar critique to the perceived neutrality of software can be made in regard to meritocracy. What is on the surface a meritocratic openness where everyone can join a project and their status evolves only on the basis of the value of their individual contributions

19 Master/slave describes the relation asymmetric relation of two devices or processes wherein one device (the master) controls the other device (slave).

is insofar political as it implicitly ignores the factors that enable people to gain the decisive merit.

Meritocracy is also part of the TDF's statutes. In their first sentence The Document Foundation states that it 'is an independent self-governing meritocratic community' (The Document Foundation, n.d.-b). Meritocracy supposedly values only individual skills, and any bias on the basis of other criteria such as social class, gender, age, skin colour is denied. Identity politics are supposed to be superimposed by technological practices that should be kept pure and undiluted by other factors. Apart from skill, all factors apart from skill are deemed irrelevant. I will discuss and critically assess how practices in LibreOffice are informed by these meritocratic notions.

I have shown how the idea of meritocracy is enacted in the governing practices of LibreOffice. While everyone is welcome to join LibreOffice activities the voting procedures are only open for members. To become a member of the foundation it is required to be continuously active for at least six months in the project. Donations do not qualify to become a member, only contributions to the project count. The model provides stability for a contribution model that is otherwise characterised by agility, flexibility and openness. The board of the foundation is elected on a yearly basis by its members. All members can take part in these elections. A membership committee, also elected by all members, approves new members and revokes membership of existing members (see chapter 6 for more details). If asked how meritocracy is enacted in the LibreOffice project or the Document Foundation, the people involved referred to these institutional practices.

The role of meritocracy in the Document Foundation is unclear. Meritocracy is anchored in the belief that status and importance are based on skills and performance. Yet it is unclear how meritocracy is enacted and what practices are considered to be central to the project. To measure the performance of community members, LibreOffice has implemented a so-called karma system. On the webpage ask.libreoffice.org users can ask questions to all things concerning LibreOffice. In a gamified environment users get karma points for answers, if their answers are upvoted by others, if they start tags for questions. In combination with badges contributors can get, the resulting numbers are calculated, and the users are ranked. The quantification of helping, the social side

of collaborating, could lead to a system that combines self-measuring, neoliberal maximisation strategies, and disciplinary action. However, it seems that these karma points do not play an important role within the Document Foundation. Status as the product of helping and discussing seems to be of tacit character. None of the actors referred to the karma points, the system was never presented publicly, and I did not receive informing answers about the system from the interviewees.

The question of merit and status becomes a bit clearer by looking at the involvement of companies in the so-called ecosystem of LibreOffice. Ecosystem connotes a peaceful, almost naturally occurring coexistence between companies and a community. Yet the question of merit shows the tension between the business interest and the foundations' goal to foster a community. Merit is also measured in the Document Foundation regarding the contributions to the product. The technical contributions to the product are counted. That includes bug fixing, design, documentation, the organisation of events, infrastructure, marketing, translations, and writing code. As shown in chapter 7, activities are distributed in a format of tickets. If a participant assigns themselves to a ticket, they promise to fulfil this task within the given timeframe. In doing so, all contributions can be measured and they could be used to feed into an extended meritocratic organisation in which only the contributors with the most merit could take part in the decision making. This however would violate the legal constitution of the foundation. For a while there was an ongoing debate between Collabora, the company with most contributions in LibreOffice, and the Foundation. As contributions are counted and the statistics are published, it has become clear that the companies in the ecosystem contribute about two thirds of code. Based on their merit, Collabora has asked for an increased visibility. LibreOffice online, a cloud-based version of LibreOffice, has been mainly developed by Collabora. The Document Foundation promotes LibreOffice Online under its own brand and Collabora finds it hard to get revenues out of building more than 95% of the code base. The clear numbers that are possible by measuring led to a disturbance amongst the collaborators as some of those who deliver the most contributions do not get the merit they expect. This debate highlights the underlying differences within the project. One of these differences is

the self-understanding of developers as the driving force of a software project who want to supress politics.

> There used to be many more developers on the board and developers are relatively easy to work and communicate with. They have concrete goals and work together on the code. And increasingly we have fewer developers on the board and more politically appointee type of people and that is unfortunate. I think that is one of the trends. (. . .) [O]ver time the politics drove away the people that got things done rather than talk about getting things done and fight for petty political advantage. And today . . . It is just an organisational phase that you go through: you produce all of this stuff and then you get subpar people involved. And it just goes… I don't have the recipe but I notice that it is a pattern in open source. The founder who did all the work get pushed out and then they are succeed by a series of… other people. I mention no names and the thing is that these people are all well-meaning. They are not evil, they are not bad, it is just an organisational disfunction of epic proportions. (Interviewee 25)

The frustration expressed in this excerpt has two roots. One is the belief that engineers get things done as they concentrate on the things that have to be done. This shows an understanding of technical practices as free of politics. Politics is what surrounds the technical practices. All the other practices that are needed to sustain the technology are believed to complicate technical advancement.

> It needs to credit the people who do the work, it needs to encourage enterprises to buy support and services from someone (. . .) who can put funds back into the ecosystem so that it grows and doesn't decline. It creates a virtuous circle of things being fixed and more people putting more money in. That's what we want. We want a virtuous circle of economic growth where companies can invest and get a return for their investment which they will then re-invest and so on and so on. (Interviewee 25)

The ecosystem here is only understood as a platform for companies to collaborate. The notion of a community was not included in the reflections of this interviewee. Another interviewee also reflects upon the politics done by TDF and the software that is worked on. Yet, here the community is included in the reflections.

> The volunteers are doing good things, there is a good will and good feeling, the code is improving significantly in many

> ways - against the sort of dead hand of the horrendous TDF governance thing that sits and squats by it and demotivates everyone who is involved with it. It is all good in theory but in reality, it is just pointless politics. If you want to get something done you have to invest in endless interpersonal nonsense rather than getting it done. The success of TDF is to separate the coding from the board, the engineering piece is totally separate. We just get on with it and we rather stay friendly with each other. It is reasonably well led, and it works. (Interviewee 25)

The politics of the TDF board are increasingly considered to prevent professional engineers from being more successful according to some interviewees that are representative of companies that are part of the LibreOffice project. These fault lines also go along the spilt between free software and open source software. The board is criticised for not assisting the advancement of the technology.

> Sure, volunteers do fantastic work IN Libre Office, they do really good work. And they do things which you couldn't justify doing, they have no short-term economic benefit but they have huge long term economic benefits. But still, things couldn't be done without professional engineers that mentor and support and guide and architect and review what is going on. We need a virtuous circle somewhere to get more volunteers and do more work. It seems obvious to me. I don't think many people disagree with it. But what they say is things like: We should give the software away for free unrestricted to everyone, so we get more users, and then we get more developers, more volunteers. That is the typical, that is what TDF's approach has been for a very long time. It doesn't work. The areas we see growth and investment are the areas where that is not the case. (Interviewee 10)

A more business minded approach that leans towards technical expertise and professional product marketing is confronted with an approach by the board to balance the interest of the different factions with diverging interests.

> We have definitely downplayed the importance of companies in TDF in the early years. Instead, we have celebrated the volunteers for their contributions because, well, the others earned money. They did not need another reward. On Blogs and on our website, we portrayed the community as the driving force of the project. But yes, we need to be more explicit about the investments of companies, if there is a new feature which is sponsored by a

> company. I consider such a claim to be legitimate. On the other hand, every company can do their own marketing. The community does not have to provide that. (Interviewee 36)

8.5. Summary

This chapter has attempted to search for the political ideas and belief systems that are in interplay with the collaborative practices. The controversy over the search of the mascot has highlighted the importance of openness, transparency and participation in f/oss. It has also shown that there is no unified interpretation of these topics within the f/oss scene. A radical interpretation of openness and transparency requires an effort in coordination. While TDF offers a wide range of transparent coordination within the project, it has failed to use the same principles in the search for the mascot.

In regard to the collaborative practices it has shown that software has politics and it informs the practices in the project. However, while LibreOffice acts as a boundary object by allowing different beliefs and values, they are not openly expressed if they do not concern the production of software. Politics should only be made in a technical sense, through the development of open standards and interoperability to guarantee practices of open sharing. This results in a tendency to supress politics in LibreOffice beyond the expression of making software. Producing free and open source software is considered to be the only valid political expression. Identity politics in contrast is seen by many participants as a disturbance. The informants on this topic were sensitive to questions of diversity but they agreed to prefer a community that concerns itself with producing a f/oss office suite.

The final part concerned the tension between the technical side and the ordering side of the project. They can be seen as two opposing political ideas on f/oss: one that focuses on sharing and building a learning community, the other which sees better code that is faster produced as the main goal of LibreOffice.

Together these topical clusters have addressed the politics of collaborative practices in the LibreOffice project. They point at the embeddedness of practices into a political realm in which they are presented,

shared, discussed, negotiated. They show the importance of the discourse in which practices are linked with and the variety of discourses within f/oss. The resulting tension are not necessarily solved by reaching a consensus. The specific form and combination of licenses that LibreOffice offers, proved the legal basis for a possible collaboration without consensus (Star, 1993) that is symptomatic of the project. The frictions between the different political ideas are to a certain extent a requirement for heterogeneous relations that are important for STS to keep a network running. Collaborative practices are produced and reproduced despite the manifold differences in politics. They are constant negotiations and moments of struggle about the practices on the basis of differences in ideas.

9. Final notes

This dissertation has looked under the hood of a free and open source software project to understand what makes it sustainable. I want to emphasise again that it is not common for f/oss projects to reach the size and the longevity of LibreOffice. The production of free and open source software is generally characterised by weak ties that allow collaborators to work on several projects as well as to end their collaborative efforts after a short amount of time. LibreOffice has shown to have found a repertoire to balance the dynamism that is needed to produce software with a stable community that sustains the project. It manages to stabilise collaboration while allowing enough fluidity and dynamism to negotiate and change practices. Collaborations, I wrote in the introduction, are short-termed relations, they emphasise individuality and connectivity. Cooperation in contrast is stable and marked by collectivity which is based on trust and offers belonging. The case of LibreOffice shows that a free and open source software project can offer elements of both sides. It includes negotiations and frictions while being sustainable. The production of a free and open source software project offers belonging for collaborators. Yet, the sustainability of LibreOffice is tested regularly by different values, conflicting interests, and strategies.

I have shown in this study how collaborative practices can be a fruitful point of view to explore how a f/oss project can be sustained. I have shown how practices can be performed individually, they are always formed collaboratively, through learning together, sharing, showing, pointing out, discussing. I have shown how these practices are not purely human. Materiality is involved whether that is through ordering systems in a bug tracker, or through the need for constant communication online, or at hackfests, conferences and other meetings. By referring the idea of practices back to the idea of actor-networks, I would point out that as long as practices are active, as long as they are practised, collaboration is ongoing. I think that a generalisation that reaches beyond this study is possible here: Not only Libre Office, or f/oss, but digital media in general can be explored as a set of sociotechnical practices that are realised through collaboration. Hence, digital media can be understood as collaborative media (Löwgren & Reimer, 2013a). This starting point for analysis is not as stable as the material

solidity of media as objects but understanding practices as a negotiated, disputed allows to capture the dynamic and fluid landscape which digital media provide.

I have proposed to engage with Actor-Network-Theory to capture this dynamism. The advantage of this approach is that it allows to loosen the focus on commonality that practice theory often comes equipped with. Instead it offers to zoom in on the differences in a collaborative constellation. LibreOffice shows to be linked by practices. Differences in political standpoints, the disputes between community and company interests, the differences in knowledge and skills, and the different interest spheres are bridged by practices. The (early) literature on f/oss often suggests that collaborators are bound by a common teleology. In LibreOffice there is no common teleology apart from keeping the project running. The reasons for this common goal are manifold. They can roughly be split in two groups as I have shown. Some collaborators believe that software should be free and open to exchange knowledge and to avoid being at the mercy of a privately owned company. Others think that free and open source software is better software because of the possibility of collaboration outside of a company. The stabilising elements in LibreOffice are practices rather than belief or value systems. They provide a dynamic character and also a stabilising effect. They are open enough for negotiations and change. They reach out to other potential collaborators while they also serve as platofroms for collaboration. Latour (1996) has pointed out that the stability of a network does not come from unity or purity but from heterogeneity and dissemination. A practice perspective agrees with that sentiment as they are open to share, open to negotiate and function as anchor points for a multifaceted group of collaborators. While a practice perspective induced by ANT included dynamism and movement, replacing practices with ANT's actions is an important factor for this study. Practices involve a collaborative element. I have mentioned in the theoretical chapter that a practice is something that is performed with regularity. A practice moves from an action to becoming a practice by being performed and shared by actors. By collaborating they become activated, and they develop through reciprocity. Collaborative practices are learned, they are shown to each other and learned together.

Software acts in manifold ways in this project. It asks for a certain speed and efficiency to get improved. It constantly asks for maintenance and attention: bugs need to be resolved, commits are for review, new version need to be released. The question of speed and efficient methods to improve the software collide with the slower pace of governing a project which ensures the sustainability of the community in which the software is embedded in. Software asks for coordinated efforts by collaborators. It needs standardisation to be produced as much as it offers the possibility for standardisation amongst collaborators. I have shown that software is not neutral. Not only does it act in accordance with Actor-Network-Theory as an actant in a network made of humans and non-humans, it also has politics – and I might add, politics increasingly have software too. Software serves as a bottleneck for the expression of politics but it can also give room for expressing political beliefs. It is values made durable as it is informed and stabilised through them. The values that inform LibreOffice are to a certain extent borrowed from a wider belief system in which free and open source software is embedded. Hereby, the collaborative network to which software belongs also has a history. LibreOffice has shown how its legacies have influenced the project. A governance structure was developed based on that specific history, the coordination of the production process has been formed along the lessons learned from the preceding project that failed to be sustainable. Thus, software does not only act as a rational structuring system that needs to be fed with code. Its nature is malleable. It can be flexible enough to allow the transmission of ideas and values, yet it can also become rigid when needed, for example when it serves as an infrastructure for coordinating or when it is forked by a community that has lost its sustainability through limited access. Software has shown to be malleable enough to bridge negotiations between collaborators whether that concerns interests, different levels of expertise, or contrasting strategies. It is also sturdy as it serves as an infrastructure for collaboration and it can contribute to forming allegiance, it provides access to collaborative resources, and it offers the arrangement of collaborative associations. Software is open for contention. Negotiations make it robust if they successfully lead to collaborations. These characteristics deviate from an understanding of media as neutral conveyers of communication – a point of view that might seem outdated from a media studies perspective, yet constant reminders about

the cultural logic that digital media bring with them are necessary due to the ongoing perception of platforms and algorithms as producers of blank canvases or objective standards. I have shown how software comes with a cultural logic through which it offers possibilities for interventions. To be able to grasp this potential the moments of creation, maintenance and activation through values provide entry points for studying software. To understand this potential of software, I have proposed that practices are a valid starting point. By focusing on practices, the possible negotiations around and with software become visible. This perspective underlines that software is the result of sociotechnical practices. Software and collaborators together interpret and formate practices. Software provides certain practices and does not accept others. ANT's symmetric perspective helps to understand the material aspect of practices. An addition with ANT elements such as symmetry and heterogeneity to the conceptualisation of media practices offers a fitting approach to study software. Traditionally in practice theory, practices can only be enacted by humans while being closely linked with non-human elements (Schatzki et al., 2001, p. 63). From a symmetric perspective, non-human elements also embody practices. LibreOffice is a socio-technical constellation that is, repeating Küpers (2016, p.2) 'composed of, surrounded by, and immersed in or consuming physical matter and non-human dimensions'.

The elements of collaborative practices

Collaborative practices have shown to emerge when a combination of doing, materiality and ordering is in place. Under doing I subsume all actions, regular or irregular that emerge. They can be intentional or not, but they need to be directed towards the aim to establish collaborative relations. These actions can be performed by humans and non-humans alike. Materiality refers to all objects involved in practices whether that is a software program itself or a conference hall, an email archive or a wiki. Ordering can be understood as arranging doings and things by putting them together in a certain order or infusing them with a specific logic.

Ordering

Ordering directly influences collaboration because it orders practices, it gives them a form so that they can be shared. Collaboration as doing

things together would be impossible without coordination. Ordering needs to be accountable. They need to be understandable and transparent enough to be explicable. Informing and explaining others how the ordering practices are structured and how to engage in them is central for them to get shared. In these moments of transfer, they are also susceptible to change and adaption. These moments can be triggered if new collaborators are introduced to the specific constellations. I have explained the importance of mentoring and the practices that are bundled with including its material aspect such as easy hacks. Other moments of negotiating ordering practices are the regular meetings that collaborators in LibreOffice have. Here, even, longstanding traditions such as highlighting the community efforts in the LibreOffice project are questioned and negotiated and alternative ordering practices that would align better with the interests of companies are put forward. What LibreOffice has shown is that hierarchies are part of the ordering mechanisms, but the practices do not derive directly from this mechanism as a form of top-down ordering. Rather ordering practices are open for negotiations and change. They become practices by being agreed upon.

However, ordering practices cannot just be imprinted on any object nor can they order all doing. There are always exit points from ordering, possibilities to avoid the practices in favour of doing. As shown exceptions can be made from mentoring, or traditional and f/oss culture-defining practices such as reviewing can be circumvented to the benefit of quicker development. Making connections (see Latour and Callon) seems sometimes more important than following practices. Nor does f/oss have one automatic ordering mechanism that can be copied and used. The Linux kernel for example has a very different ordering mechanism following a development model that is marked by a benevolent dictatorship.

Materiality

Materiality is the ground for practices. It is not a neutral ground but it comes with preconditions. Some doings might not be accepted by the material ground. Materiality must not necessarily be software, also licenses or meeting rooms and conference halls are part of the material basis for collaboration. While materiality is concrete it is susceptible to

change – a material object such as software is easier to change than a university for example of course - because it is the concrete outcome of negotiations. Doings can change materiality. For that to happen an ordering system needs to be established. Ordering is not a stable form of ruling over doings hence the use of the verb ordering instead of order or governance.

Also, it is not possible to establish practices that can be swapped to other objects. An object, as shown is not a blank canvas. It comes with a history and it is embedded in a discourse, it brings its own register of practices that can be activated or not. Software offers to make practices countable in order to make them accountable. Yet as LibreOffice shows not all practices can be counted, mentoring, discussing, negotiating cannot be counted in its full sense: The average duration of a mentoring process can be counted if the involved people log in their time but that does not tell the full story of the social impact of the practice. To a certain extent they are invisible work [quote] that function like a glue for the rest of the project. They provide social ordering. And even if learning practices are logged, counted and displayed such as the karma system, they are not used as a metric for status which a strictly meritocratic system would do.

Doing

Doing is another form of action next to ordering. I would argue that is is best understood as actions with a certain aim. Doing software is akin to the concept of doing gender. Gender is generally understood as being performed through shared actions. If these actions get socially accepted they get assessed. This line of thinking is useful to understand doing digital media. All forms of actions are possible. But for a doing to become socially (or culturally) accepted it needs sharing and approvement. From a socio-technical perspective both objects and humans perform and are performed. They are in doing. The practices that are activated are based on accepted conceptions within a specific culture. Collaboration activates practices which then can get stabilised by being used for collaboration in linkage with ordering mechanisms and objects.

Together these three elements are constitutive for collaborative practices. They are not aligned in some sort of hierarchy, nor is there a mandatory sequence in which they are performed. They all depend on each other. Also, they can all interfere in each other. Their relation is circular rather than hierarchic as the fieldwork has shown. LibreOffice does not have an ordering structure that dominated all doings and puts a strict order on software. Rather the three elements are influencing each other. In some instances, ordering is a dominant element while other push for doing software aiming at bypassing ordering as much as possible. In some cases I have shown how software has a certain power over ordering and doing as it imposes structures that need to be followed so that it keeps accessible for others. A certain sociotechnical circularity is given here.

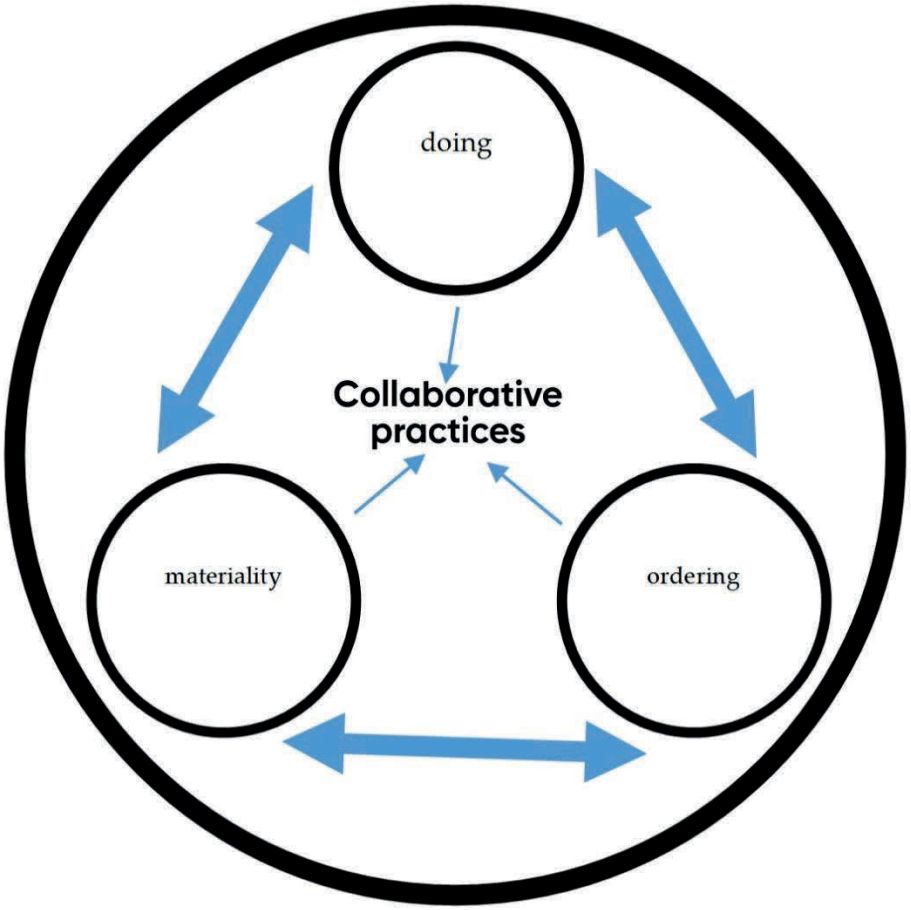

Figure 3: The elements of collaborative practices

This dynamic interplay provides a perspective from which we can begin to ask for the setup of network-based form of projects: How do collaborative practices emerge (see research question 1) that forms the basis for these projects? There is no universal answer to this question. The case of LibreOffice has shown how doings are informed by a material base of code, licenses, governance mechanisms as well as by actions of ordering in order to become practices. There is a sociotechnical interplay at hand that differs from project to project. The fact that there is software does not automatically lead to the emergence of collaborative practices.
Research question 2 asked: What are the elements of the collective frame of reference which stabilises collaboration and offers options to engage in practices? It points towards the negotiations as well as for the stabilisation of practices.

I have shown how central the possibility for negotiations are in the LibreOffice project. Collaborators can question the practices that are activated as long as they accept that there is ordering involved and they are willing to engage in doing. These negotiations do not weaken the collaborative practices. Rather they make the more solid as negotiations keep the project running.

Research question 3 relates to the problem of governance and a possible conflict with the concept of self-organisation. Along this question, LibreOffice highlights how too much stability in governance can become the cause for a possible conflict. Self-organisation relies on speed and efficiency and possible tensions with a stable, yet slow-moving governing mechanism, can arise. Licenses and the specific setup of the governance in place play an import role, LibreOffice has shown. And, research question 4, refers to the belief system of collaborative projects. The politics and ideals of collaborators is a decisive factor in LibreOffice as I have shown. LibreOffice has shown how its values and belief system are the result of an interplay between the software and its history and diverse discourses around f/oss. The belief system itself orders the project as it allows or impedes certain practices. The software as well gives materiality for collaborative practices, yet it hinders other forms as it is informed by its history and specific values that differ from project to project. This belief system is part of the making of the constellation. It is best described as a discourse in which collaborative

practices can be activated. Discourse is a way of constituting knowledge, a way of regulating what knowledge is as Foucault points out. Discourse is always together with practices. Together they can produce knowledge and create associations between people. Discourse is not part of the model but it has been mentioned several times in this study. Thus, it is clearly important. The construction of the discourse would require its separate circle that is in interplay with the collaborative practice circle. Practices and discourse together produce culture. Software can be addressed form this perspective, asking which practices are used, and which are forbidden or supressed. What forms of ordering, materiality and doings are possible to become an element for the formation of practices? What can be negotiated and to which extent?

However, I have shown in this study that the importance of discourse differs from classic practice theory. Schatzki who proposes that what he calls practice-arrangement complexes '[…] embrace interrelated actions, conscious, deliberate, and purposeful cooperation, the pursuit of common goals, common rules, and enjoined teleologies.' (Schatzki, 2017, pp 73-74). Schatzki embeds practices in a discursive formation that have to be commonly accepted as much as those practices. Not only have I shown that exit points and work arounds for practices exist in form of a boundary object, I have also shown that collaborative constellations can work without a common teleology. The commonality that is reached in LibreOffice is more akin to what Callon (1991, p. 145) calls a 'successful process of translation' which 'generates a shared space, equivalence and commensurability'. Rather than a discourse that regulates practices, I have shown that there is a dynamic movement between actors at place. A movement that looks for stability but offers exit routes that allow change and a realignment in a specific space. I lean more toward the John Law's definition to describe LibreOffice: 'Society, organizations, agents and machines are all effects generated in patterned networks of diverse (not simply human) materials' (Law, 1991, p. 380). Digital media practices thus, emerge as associations that highlight the interlacing between technology and sociality. ANT proposes the social and technological to be locked in an interplay that needs to be described as the process of a formation of a collaborative culture. What the fieldwork has shown however, is the importance

of discourse. In that I favour ANT as an approach to underline the relational dynamics of practices, yet it has become clear how important a discourse is for the project and the practices that are activated.

The practice framework also allows to ask for the everyday practices in which humans engage in relation to media. We could ask what people are doing with software as well as what software is doing to us. We can also ask what doings are accepted and which doings are dismissed or ignored, and additionally why they are not accepted. Maybe the code is programmed badly, or it does not fit the ordering structure. In such a case the possibilities for negotiations become of interest and how they are ordered. The ordering structure itself can be questioned: By which interests is it influenced? How can it be changed? What is the entry level for negotiating the ordering structure?

Practices can be critically examined further with this approach: Why can I not engage in collaborative practices with others? Does the materiality not allow sharing? Is there an ordering structure that restricts actions so that they cannot become practices? The setup of the materiality can be critically assessed. What is its history? What belief and value systems do inform the materiality?

In the introduction and in the theory chapter I have emphasised the important contribution that software studies have made to media studies by focusing on the materiality of software. With this focus I have explored the entanglement of the collaborative practices that are in place at LibreOffice. I have talked with collaborators about at the political and ethical ideas in which the practices are embedded, I have explored the ordering mechanism that activate and channel practices, I have seen how practices emerge and how others are not used anymore. Not only has the focus on practices shown how humans and non-humans are both actants in ANT's sense as materiality and sociality both take part in making software. It has also shown how ideas and values are built into software. Subsequently this means that software needs to be understood within the specific cultural context it is produced and reproduced in. Such an approach does not only offer to study a free and open source software project. By exploring collaborative practices, it is possible to get closer to the nexus of material and social, to understand the complex cultural constellations that media are.

References

A. Figures

Figure 1: A timeline of major derivatives of StarOffice and OpenOffice.org with NeoOffice in light purple [Graphic], by David Gerard, Донор, Shunesburg69 (2021). Available from: https://en.wikipedia.org/wiki/NeoOffice#/media/File:StarOffice_major_derivatives.svg

Figure 2: How we are structured (Graphic), by Florian Effenberg (2020). Available from: https://blog.effenberger.org/wp-content/uloads/2020/10/libocon_tdf.pdf

B. Texts

Ågerfalk, P., & Fitzgerald, B. (2008). Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. *MIS Quarterly*, *6*(1), 385–409.

Agre, P. E. (1994). Surveillance and capture: Two models of privacy. *The Information Society*, *10*(2), 101–127. https://doi.org/10.1080/01972243.1994.9960162

Alami, A., Cohn, M. L., & Wąisowski, A. (2020). How Do FOSS Communities Decide to Accept Pull Requests? *Proceedings of the Evaluation and Assessment in Software Engineering*, 220–229. https://doi.org/10.1145/3383219.3383242

Alami, A., Leavitt Cohn, M., & Wasowski, A. (2019). Why Does Code Review Work for Open Source Software Communities? *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 1073–1083. https://doi.org/10.1109/ICSE.2019.00111

Anonymous. (2017, November 27). Libreoffice Mascot Thread 6: Komm, Süsser Todd. *8chan*. https://8ch.net/tech/res/828482.html

Argyris, C. (2010). Diagnosing Defenses Against the Outsider. *Journal of Social Issues*, *8*(3), 24–34. https://doi.org/10.1111/j.1540-4560.1952.tb01615.x

Aristotle. (2013). *Poetics* (A. Kenny, Trans.). Oxford University Press.

Bacon, J. (2012). *The art of community*. O'Reilly.

Bannon, L. J., & Schmidt, K. (1989). CSCW - Four Characters in Search of a Context. *DAIMI Report Series*, *18*(289). https://doi.org/10.7146/dpb.v18i289.6667

Bardi, J., & Marques, A. C. (2007). Taxonomic redescription of the Portuguese man-of-war, Physalia physalis (Cnidaria, Hydrozoa, Siphonophorae, Cystonectae) from Brazil. *Iheringia. Série Zoologia*, *97*(4), 425–433. https://doi.org/10.1590/S0073-47212007000400011

Bauer, M. (2010, October 8). *Re: [De-dev] Oracle macht bei LibreOffice nicht mit*. https://www.mail-archive.com/dev@de.openoffice.org/msg30504.html

Bauman, Z. (2000). *Liquid modernity*. Polity Press ; Blackwell.

Bauwens, M. (2005). The political economy of peer production. *CTheory*. www.ctheory.net/articles.aspx?id=499

Bauwens, M., Kostakis, V., & Pazaitis, A. (2019). *Peer to peer: The commons manifesto*. University of Westminster Press. https://www.doabooks.org/doab?func=fulltext&uiLanguage=en&rid=32892

Behrens, T. (2010, October 8). *Re: [De-dev] Oracle macht bei LibreOffice nicht mit*. https://www.mail-archive.com/dev@de.openoffice.org/msg30486.html

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. Yale University Press.

Benkler, Y. (2011). *The penguin and the Leviathan: The triumph of cooperation over self-interest*. Crown Business.

Benkler, Y. (2016). Peer production and cooperation. Forthcoming. In M. Latzer & J. Bauer (Eds.), *Handbook on the Economics of the Internet*. http://www.benkler.org/Peer production and cooperation 09.pdf

Berdou, E. (2011). *Organization in open source communities: At the crossroads of the gift and market economy*. Routledge. http://public.eblib.com/choice/publicfullrecord.aspx?p=592964

Berger, P. L., & Luckmann, T. (1990). *The social construction of reality: A treatise in the sociology of knowledge*. Anchor Books.

Berlant, L. (2016). The commons: Infrastructures for troubling times*. *Environment and Planning D: Society and Space, 34*(3), 393–419. https://doi.org/10.1177/0263775816645989

Berry, D. M. (2015). *Critical theory and the digital*. Bloomsbury.

Birkinbine, B. J. (2020). *Incorporating the Digital Commons: Corporate Involvement in Free and Open Source Software*. University of Westminster Press. https://doi.org/10.16997/book39

Bitzer, J., & Schröder, P. J. H. (Eds.). (2006). *The economics of Open Source Software development*. Elsevier.

Boltanski, L., & Chiapello, È. (2007). *The new spirit of capitalism*. Verso.

Bonvoisin, J., Molloy, J., Häuer, M., & Wenzel, T. (2020). Standardisation of Practices in Open Source Hardware. *Journal of Open Hardware, 4*(1), 2. https://doi.org/10.5334/joh.22

Bowker, G. C., & Star, S. L. (1999a). *Sorting things out: Classification and its consequences*. MIT Press.

Bowker, G. C., & Star, S. L. (1999b). *Sorting things out: Classification and its consequences*. MIT Press.

Bräuchler, B., & Postill, J. (Eds.). (2010). *Theorising media and practice*. Berghahn Books.

Bugzilla. (n.d.). Filing a bug. *Bugzilla 5.0.4 Documentation*. Retrieved 11 June 2019, from https://bugs.documentfoundation.org/docs/en/html/using/filing.html#reporting-a-new-bug

Butler, S., Gamalielsson, J., Lundell, B., Brax, C., Sjoberg, J., Mattsson, A., Gustavsson, T., Feist, J., & Lonroth, E. (2021). On Company Contributions to Community Open Source Software Projects. *IEEE Transactions on Software Engineering*, *47*(7), 1381–1401. https://doi.org/10.1109/TSE.2019.2919305

Callon, M. (1986a). Some elements of a sociology of translation: Domestication of the scallops and the fishermen of Saint Brieuc Bay. In J. Law (Ed.), *Power, action, and belief: A new sociology of knowledge?* (pp. 196–233). Routledge & Kegan Paul.

Callon, M. (1986b). The Sociology of an Actor-Network: The Case of the Electric Vehicle. In M. Callon, J. Law, & A. Rip (Eds.), *Mapping the Dynamics of Science and Technology* (pp. 19–34). Palgrave Macmillan UK. https://doi.org/10.1007/978-1-349-07408-2_2

Callon, M. (1991). Techno-economic networks and irreversibility. In J. Law (Ed.), *A sociology of monsters: Essays on power, technology, and domination* (pp. 132–161). Routledge.

Callon, M., & Law, J. (1992). The Life and Death of an Aircraft: A Network Analysis of Technical Change. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 21–52). MIT Press.

Callon, M., & Law, J. (1997). Agency and the Hybrid Collectif. In B. H. Smith & A. Plotnitsky (Eds.), *Mathematics, Science, and Postclassical Theory* (pp. 95–117). Duke University Press. https://doi.org/10.1215/9780822382720-006

Ceruzzi, P. E. (2003). *A history of modern computing*. MIT Press.

Chagani, F. (2014). Critical political ecology and the seductions of posthumanism. *Journal of Political Ecology*, *21*(1), 424. https://doi.org/10.2458/v21i1.21144

Chandler, D. (2013). The World of Attachment? The Post-humanist Challenge to Freedom and Necessity. *Millennium: Journal of International Studies*, *41*(3), 516–534. https://doi.org/10.1177/0305829813481840

Chun, W. H. K. (2005). On Software, or the Persistence of Visual Knowledge. *Grey Room*, *18*, 26–51. https://doi.org/10.1162/1526381043320741

Clifford, J., & Marcus, G. E. (Eds.). (2008). *Writing culture: The poetics and politics of ethnography*. Univ. of California Press. (Original work published 1986)

Cohn, M. L. (2019). In J. Vertesi & D. Ribes (Eds.), *DigitalSTS: a field guide for science & technology studies* (pp. 423–446). Princeton University Press.

Coleman, G. (2004). The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast. *Anthropological Quarterly*, *77*(3), 507–519. https://doi.org/10.1353/anq.2004.0035

Coleman, G. (2012). Anonymous. In C. Wiedemann & S. Zehle (Eds.), *Depletion design: A glossary of network ecologies* (pp. 11–17). Institute of Network Cultures.

Coleman, G. (2013). *Coding freedom: The ethics and aesthetics of hacking*. Princeton University Press.

Coleman, G. (2015a). *Hacker, hoaxer, whistleblower, spy: The many faces of Anonymous*. Verso.

Coleman, G. (2015b). The anthropological trickster. *HAU: Journal of Ethnographic Theory*, *5*(2), 399–407. https://doi.org/10.14318/hau5.2.024

Coleman, G., & Brunton, F. (2014). Closer to the metal. In T. Gillespie, P. J. Boczkowski, & K. A. Foot (Eds.), *Media technologies. Essays on communication, materiality, and society* (pp. 77–98). The MIT Press.

Coleman, G., & Golub, A. (2008). Hacker practice: Moral genres and the cultural articulation of liberalism. *Anthropological Theory*, *8*(3), 255–277. https://doi.org/10.1177/1463499608093814

Couldry, N. (2004). Theorising media as practice. *Social Semiotics*, *14*(2), 115–132. https://doi.org/10.1080/1035033042000238295

Couldry, N. (2012). *Media, society, world: Social theory and digital media practice*. Polity.

Crawford, S. E. S., & Ostrom, E. (1995). A Grammar of Institutions. *American Political Science Review*, *89*(03), 582–600. https://doi.org/10.2307/2082975

Creswell, J. W. (2007). *Qualitative inquiry & research design: Choosing among five approaches*. Sage Publications.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches*. SAGE.

Crotty, M. (1998). *The foundations of social research: Meaning and perspective in the research process*. Sage Publications.

Crowston, K., & Howison, J. (2005). The social structure of free and open source software development. *First Monday*. https://doi.org/10.5210/fm.v10i2.1207

Crowston, K., & Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, *18*(4), 65–85. https://doi.org/10.1007/s12130-006-1004-8

Cumming, G. S. (2016). Heterarchies: Reconciling Networks and Hierarchies. *Trends in Ecology & Evolution*, *31*(8), 622–632. https://doi.org/10.1016/j.tree.2016.04.009

Cuprite_Cane. (2017). *Reddit r/Linux*. https://www.reddit.com/r/linux/comments/7ddbk3/libreoffice_mascot_survey_the_progress_so_far/dpxqac8?utm_source=share&utm_medium=web2x

De Angelis, M. (2017). *Omnia Sunt Communia: On the commons and the transformation to postcapitalism*. Zed Books.

Debian. (2004, April 26). *Debian Social Contract*. Debian. https://www.debian.org/social_contract

Deignan, M. P. (2001, October 11). *StarOffice shaping up as true Office alternative*. ZDNet. https://www.zdnet.com/article/staroffice-shaping-up-as-true-office-alternative-5000296788/

Denzin, N. K. (2017). *Sociological methods: A sourcebook*.

Denzin, N. K., & Lincoln, Y. S. (Eds.). (2018). *The SAGE handbook of qualitative research*. Sage.

Dworschak, M. (1998). Kalkulierter Selbstmord. *Der Spiegel*, *48*(1998). http://www.spiegel.de/spiegel/print/d-8035571.html

Easterbrook, S. M. (Ed.). (1992). *CSCW: Cooperation or conflict?* Springer-Verlag.

Effenberg, F. (2009, April 22). *Re: [De-dev] Persönliches Statement zur Übernahme durch Oracle*. https://www.mail-archive.com/dev@de.openoffice.org/msg25748.html

Effenberg, F. (2010, October 8). *Re: [De-dev] Oracle macht bei LibreOffice nicht mit*. https://www.mail-archive.com/dev@de.openoffice.org/msg30496.html

Fetterman, D. M. (2010). *Ethnography: Step-by-step*. SAGE.

Fiddler, J. L. (2021). Listening Deeply: Indexing Research Conversations in a Narrative Inquiry. In C. Vanover, P. Mihas, & J. Saldaña (Eds.), *Analyzing and interpreting qualitative research: After the interview* (pp. 279–294). SAGE Publications, Inc.

Flusser, V. (1998). *Vom Subjekt zum Projekt: Menschwerdung*. Fischer.

Fontana, R. (2019, February 28). *Why CLAs aren't good for open source*. Opensource.Com. https://opensource.com/article/19/2/cla-problems

Foucault, M. (1972). *The Archeology of Knowledge and the Discourse on Language*. Pantheon. (Original work published 1969)

Foucault, M. (1994). *The order of things: An archaeology of the human sciences* (Vintage books edition). Vintage Books. (Original work published 1966)

Foucault, M. (1995). *Discipline and punish: The birth of the prison*. Vintage Books. (Original work published 1975)

Frabetti, F. (2015). *Software theory: A cultural and philosophical study*. Rowman & Littlefield International.

Franklin, G. F., Powell, J. D., & Workman, M. L. (2002). *Digital control of dynamic systems*. Addison-Wesley.

Franklin, S. (2015). *Control: Digitality as cultural logic*. The MIT Press.

Free Software Foundation. (1989). GNU General Public License, version 1. *GNU Operating System*. https://www.gnu.org/licenses/gpl-1.0.html

Free Software Foundation. (2016, December 27). What is free software? *GNU Operating System*. https://www.gnu.org/philosophy/free-sw.html

Fuller, M. (2008). Introduction, the stuff of software. In M. Fuller (Ed.), *Software studies: A lexicon* (pp. 1–14). MIT Press.

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. https://doi.org/10.1046/j.1365-2575.2001.00108.x

Galloway, A. R. (2006). Language Wants To Be Overlooked: On Software and Ideology. *Journal of Visual Culture*, *5*(3), 315–331. https://doi.org/10.1177/1470412906070519

Gamalielsson, J., & Lundell, B. (2014). Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved? *Journal of Systems and Software*, *89*, 128–145. https://doi.org/10.1016/j.jss.2013.11.1077

Gamalielsson, J., & Lundell, B. (2017). *On the potential for improved standardisation through use of open source work practices in different standardisation organisations: How can open source-projects contribute to development of IT-standards?* (K. Blind & K. Jakobs, Eds.; pp. T137–T155). Wissenschaftsverlag Mainz.

Gasser, L. (1986). The integration of computing and routine work. *ACM Transactions on Information Systems*, *4*(3), 205–225. https://doi.org/10.1145/214427.214429

Gießmann, S. (2018). Elemente einer Praxistheorie der Medien. *Zfm (Zeitschrift Für Medienwissenschaft)*, *19*(2), 95–109. http://dx.doi.org/10.25969/mediarep/1228

Granovetter, M. S. (1973). The Strength of weak ties. *American Journal of Sociology*, *78*(6), 1360–1380. https://doi.org/10.1086/225469

Gregory, C. (2014). On religiosity and commercial life: Toward a critique of cultural economy and posthumanist value theory. *HAU: Journal of Ethnographic Theory*, *4*(3), 45–68. https://doi.org/10.14318/hau4.3.005

Greif, I. (Ed.). (1988). *Computer-supported cooperative work: A book of readings*. Morgan Kaufmann.

Guba, E. G., & Lincoln, Y. S. (2018). Paradigmatic controversies, contradictions, and emerging influences. In N. K. Denzin & Y. S. Lincoln (Eds.), *The SAGE handbook of qualitative research* (pp. 191–215). Sage.

Hammersley, M., & Atkinson, P. (2007). *Ethnography: Principles in practice*. Routledge.

Handler, R. A. (2018a). Centre and periphery: How to understand a network. In L. Peja, N. Carpentier, F. Colombo, M. F. Murru, S. Tosoni, & R. W. Kilborn (Eds.), *Current perspectives on communication and media research* (pp. 247–256). edition lumière.

Handler, R. A. (2018b). Protocols of Control: Collaboration in Free and Open Source Software. In P. Bilić, J. Primorac, & B. Valtýsson (Eds.), *Technologies of Labour and the Politics of Contradiction* (pp. 175–192). Springer International Publishing. https://doi.org/10.1007/978-3-319-76279-1_10

Hayles, K. (1999). *How we became posthuman: Virtual bodies in cybernetics, literature, and informatics*. University of Chicago Press.

Hayles, K. (2005). *My mother was a computer: Digital subjects and literary texts*. University of Chicago Press.

Healy, K., & Schussman, A. (2003). *The ecology of open-source software development*. [Technical report]. University of Arizona.

Heise online. (2010, October 5). *Oracle macht bei LibreOffice nicht mit*. Heise Online. https://www.heise.de/newsticker/meldung/Oracle-macht-bei-LibreOffice-nicht-mit-1102282.html

Herzogenrath, B. (Ed.). (2015). *Media matter: The materiality of media, matter as medium*. Bloomsbury Academic.

Heywood, A. (2019). *Politics* (Fifth edition). Macmillan International Higher Education/Red Globe Press.

Himanen, P. (2001). *The hacker ethic, and the spirit of the information age*. Random House.

Hine, C. (2008). Virtual ethnography: Modes, varieties, affordances. In N. Fielding, R. M. Lee, & G. Blank (Eds.), *The SAGE handbook of online research methods* (pp. 257–270). SAGE.

Hippel, E. von, & Krogh, G. von. (2003). Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science*, *14*(2), 209–223. https://doi.org/10.1287/orsc.14.2.209.14992

Howe, J. (2009). *Crowdsourcing: Why the power of the crowd is driving the future of business*. Three Rivers Press.

Joerges, B. (1999). Do Politics Have Artefacts? *Social Studies of Science*, *29*(3), 411–431. https://doi.org/10.1177/030631299029003004

Johnson, D. (2013). *Media franchising: Creative license and collaboration in the culture industries*. New York University Press.

Juris, J. S., Caruso, G., Couture, S., & Mosca, L. (2013). The cultural politics of free software and technology within the Social Forum process. In J. S. Juris & A. Khasnabish (Eds.), *Insurgent encounters: Transnational activism, ethnography, and the political* (pp. 342–365). Duke University Press.

Karatzogianni, A., & Michaelides, G. (2009). Cyberconflict at the edge of chaos: Cryptohierarchies and self-organisation in the open-source movement. *Capital & Class*, *33*(1), 143–157. https://doi.org/10.1177/030981680909700108

Kelty, C. M. (2008). *Two bits: The cultural significance of free software*. Duke University Press.

Kirschenbaum, M. G. (2003, August 29). Virtuality and VRML: Software studies after Manovich. *Electronic Book Review*. http://www.electronicbookreview.com/thread/technocapitalism/morememory

Kitchin, R., & Dodge, M. (2011). *Code/space: Software and everyday life*. MIT Press.

Kittler, F. (1986). *Grammophon, Film, Typewriter*. Brinkmann & Bose.

Kittler, F. (1993). Es gibt keine Software. In *Draculas Vermächtnis: Technische Schriften* (pp. 225–242). Reclam.

Kniberg, H. (2008, March 31). Version control for multiple agile teams. *InfoQ*. https://www.infoq.com/articles/agile-version-control

Korczynski, M., & Wittel, A. (2020). The Workplace Commons: Towards Understanding Commoning within Work Relations. *Sociology*, *54*(4), 711–726. https://doi.org/10.1177/0038038520904711

Kornberger, M., Bowker, G. C., Elyachar, J., Mennicken, A., Miller, P., Nucho, J. R., & Pollock, N. (Eds.). (2019). *Thinking Infrastructures*. Emerald Publishing Limited.

Kranzberg, M. (1986). Technology and History: 'Kranzberg's Laws'. *Technology and Culture*, *27*(3), 544. https://doi.org/10.2307/3105385

Kreiss, D., Finn, M., & Turner, F. (2011). The limits of peer production: Some reminders from Max Weber for the network society. *New Media & Society*, *13*(2), 243–259. https://doi.org/10.1177/1461444810370951

Küpers, W. M. (2016). Embodied, relational practices of human and non-human in a material, social, and cultural nexus of organizations. *On_Culture: The Open Journal for the Study of Culture*, *2*. http://geb.uni-giessen.de/geb/volltexte/2016/12352

Latour, B. (1992). Where are the missing masses? Sociology of a few mundane artefacts. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 225–259). MIT Press.

Latour, B. (1993). *We have never been modern*. Harvard University Press.

Latour, B. (1996). On actor-network theory. A few clarifications plus more than a few complications. *Soziale Welt*, *47*, 369–381.

Latour, B. (1999). *Pandora's hope: Essays on the reality of science studies*. Harvard University Press.

Latour, B. (2004). *Politics of nature: How to bring the sciences into democracy*. Harvard University Press.

Latour, B. (2007). *Reassembling the social: An introduction to Actor-Network-Theory*. Oxford University Press.

Latour, B., & Akrich, M. (1992). A Summary of a Convenient Vocabulary for the Semiotics of Human and Nonhuman Assemblies. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society: Studies in sociotechnical change* (pp. 259–264). MIT Press.

Latour, B., & Callon, M. (1981). Unscrewing the Big Leviathan: How Actors Macrostructure Reality, and How Sociologists Help Them To Do So. In K. Knorr-Cetina & A. V. Cicourel (Eds.), *Advances in social theory and methodology: Toward an integration of micro- and macro-sociologies* (pp. 277–303). Routledge.

Latour, B., Jensen, P., Venturini, T., Grauwin, S., & Boullier, D. (2012). 'The whole is always smaller than its parts' - a digital test of Gabriel Tardes' monads: 'The whole is always smaller than its parts'. *The British Journal of Sociology*, *63*(4), 590–615. https://doi.org/10.1111/j.1468-4446.2012.01428.x

Latour, B., & Woolgar, S. (1986). *Laboratory life: The construction of scientific facts*. Princeton University Press.

Lauhakangas, I. (2019, April 23). Development/Re-Basing. *The Document Foundation's Wiki*. https://wiki.documentfoundation.org/Development/Re-Basing#Why_go_with_the_MPLv2_.3F

Lauhakangas, I. (2020, August 4). *How to Report Bugs in LibreOffice*. The Document Foundation. https://wiki.documentfoundation.org/QA/BugReport

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press.

Law, J. (Ed.). (1991). *A sociology of monsters: Essays on power, technology, and domination*. Routledge.

Law, J. (2009). Actor Network Theory and Material Semiotics. In B. S. Turner (Ed.), *The New Blackwell Companion to Social Theory* (pp. 141–158). Wiley-Blackwell. https://doi.org/10.1002/9781444304992.ch7

LeCompte, M. D., & Schensul, J. J. (2010). *Designing & conducting ethnographic research: An introduction*. AltaMira Press.

Lempert, L. B. (2007). Asking Questions of the Data: Memo Writing in the Grounded Theory Tradition. In A. Bryant & K. Charmaz, *The SAGE Handbook of Grounded Theory* (pp. 245–264). SAGE Publications Ltd. https://doi.org/10.4135/9781848607941.n12

Lessig, L. (2008). *Remix: Making art and commerce thrive in the hybrid economy*. Penguin Press.

Levy, S. (2010). *Hackers. Heroes of the computer revolution*. O'Reilly Media.

Linebaugh, P. (2008). *The Magna Carta manifesto: Liberties and commons for all*. University of California Press.

Lippka, C. (2010, October 20). *Re: [De-dev] 3 Wege und ihre Folgen für die Community*. https://www.mail-archive.com/dev@de.openoffice.org/msg30730.html

Louridas, P., Spinellis, D., & Vlachos, V. (2008). Power laws in software. *ACM Transactions on Software Engineering and Methodology*, *18*(1), 1–26. https://doi.org/10.1145/1391984.1391986

Lovink, G. (2011). *Networks without a cause: A critique of social media*. Polity.

Löwgren, J., & Reimer, B. (2013a). *Collaborative media: Production, consumption, and design interventions*. The MIT Press.

Löwgren, J., & Reimer, B. (2013b). The Computer is a Medium, Not a Tool: Collaborative Media Challenging Interaction Design. *Challenges*, *4*(1), 86–102. https://doi.org/10.3390/challe4010086

Lungu, M. F. (2009). *Reverse Engineering Software Ecosystems* [Doctoral Dissertation, University of Lugano]. http://scg.unibe.ch/archive/papers/Lung09b.pdf

Manovich, L. (2002). *The language of new media*. MIT Press.

Manovich, L. (2013). *Software takes command: Extending the language of new media*. Bloomsbury.

Merriam, S. B., & Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation*. John Wiley & Sons.

metacom. (2003, December). *What is Hacktivism 2.0*. The Hacktivist. http://eprints.soton.ac.uk/8762/2/whatishacktivism.pdf

Michaelsen, B. (2010, October 10). *[de-dev] Re: Oracle macht bei LibreOffice nicht mit*. https://www.mail-archive.com/dev@de.openoffice.org/msg30537.html

Morse, J. M. (2018). Reframing Rigor in Qualitative Inquiry. In N. K. Denzin & Y. S. Lincoln (Eds.), *The SAGE handbook of qualitative research* (pp. 796–817). Sage.

Moulier Boutang, Y. (2011). *Cognitive capitalism*. Polity Press.

Munro, C., Vue, Z., Behringer, R. R., & Dunn, C. W. (2019). Morphology and development of the Portuguese man of war, Physalia physalis. *Scientific Reports*, *9*(1), 15522. https://doi.org/10.1038/s41598-019-51842-1

Nouws, C. (2019, September 3). Updated LibreOffice growth infographic (2019). *Collabora Office Community News*. https://www.collaboraoffice.com/community-news/updated-libreoffice-growth-infographic-2019/

Nunes, R. (2014). *Organisation of the Organisationless: Collective Action After Networks*. Post Media Lab. https://doi.org/10.25969/mediarep/14006

Nyman, L., & Lindman, J. (2013). Code forking, governance, and sustainability in open source software. *Technology Innovation Management Review*, *3*(1).

Nyman, L., & Mikkonen, T. (2011). To fork or not to fork: Fork motivations in sourceforge projects. In S. A. Hissam, B. Russo, M. G. de Mendonça Neto, & F. Kon (Eds.), *Open Source Systems: Grounding Research* (Vol. 365, pp. 259–268). Springer. http://link.springer.com/10.1007/978-3-642-24418-6

O'Neil, M. (2009). *Cyberchiefs: Autonomy and authority in online tribes*. Pluto Press.

Open Source Initiative. (2002a). *Frequently Asked Questions*. Open Source Initiative. https://web.archive.org/web/20021204155057/http://www.opensource.org/advocacy/faq.php

Open Source Initiative. (2002b). *The Open Source Case for Hackers*. Open Source Initiative. https://web.archive.org/web/20021204155022/http://www.opensource.org/advocacy/case_for_hackers.php#marketing

Open Source Initiative. (2007). *The Open Source definition*. Open Source Initiative. https://opensource.org/osd

Open Source Initiative. (2018). *History of the OSI*. Open Source Initiative. https://opensource.org/history

OpenDocument XML.org. (2006, August 28). *History of OpenDocument*. OpenDocument XML.Org. http://opendocument.xml.org/milestones

OpenOffice.org. (2010, August 15). *Schedule*. OpenOffice.Org. http://web.archive.org/web/20100815081350/https://www.ooocon.org/index.php/ooocon/2010/schedConf/schedule

Oram, A., & Safari, an O. M. C. (2016). *Open Source in Brazil*. https://www.safaribooksonline.com/library/view//9781492050827/?ar

O'Reilly, T. (2005, September 30). *What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software*. https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=1

Ortner, S. B. (2006a). *Anthropology and social theory: Culture, power, and the acting subject*. Duke University Press.

Ortner, S. B. (2006b). *Anthropology and social theory: Culture, power, and the acting subject*. Duke University Press.

Ostrom, E. (1990). *Governing the commons. The evolution of institutions for collective action*. Cambridge University Press.

Ostrom, E. (2009). A General Framework for Analyzing Sustainability of Social-Ecological Systems. *Science*, *325*(5939), 419–422. https://doi.org/10.1126/science.1172133

Parikka, J. (2007). *Digital contagions: A media archaeology of computer viruses*. Peter Lang.

Parikka, J. (2015). *A geology of media*. University of Minnesota Press.

Parikka, J., & Sampson, T. D. (Eds.). (2009). *The spam book: On viruses, porn, and other anomalies from the dark side of digital culture*. Hampton Press.

Pfaffenberger, B. (1995). *The USENET book: Finding, using, and surviving newsgroups on the Internet*. Addison-Wesley.

Rathke, E. (2010, October 8). *Re: [De-dev] Oracle macht bei LibreOffice nicht mit*. https://www.mail-archive.com/dev@de.openoffice.org/msg30493.html

Raymond, E. S. (1999). *The cathedral & the bazaar: Musings on Linux and open source by an accidental revolutionary*. O'Reilly.

Raymond, E. S. (2000). *The Jargon File (version 4.4.7)*. http://www.catb.org/jargon/html/index.html

Rieder, B., & Schäfer, M. T. (2008). Beyond Engineering: Software Design as Bridge over the Culture/Technology Dichotomy. In *Philosophy and Design* (pp. 159–171). Springer Netherlands.

Robles, G., & González-Barahona, J. M. (2012). A Comprehensive Study of Software Forks: Dates, Reasons and Outcomes. In I. Hammouda, B. Lundell, T. Mikkonen, & W. Scacchi (Eds.), *Open Source Systems: Long-Term Sustainability* (pp. 1–14). Springer Berlin Heidelberg.

Rogers, R. (2013). *Digital methods*. The MIT Press.

Rozas, D. (2018). *Self-organisation in commons-based peer production: Drupal: 'the drop is always moving'*. (p. 401). University of Surrey.

Saldaña, J. (2013). *The coding manual for qualitative researchers*. SAGE.

Sam Muirhead. (2017, October 2). Open Source Design. https://discourse.opensourcedesign.net/t/flawed-design-process-for-libreoffice-mascot/300/3

Sandvig, C. (2013). The Internet as infrastructure. In W. H. Dutton (Ed.), *The Oxford handbook of Internet studies* (pp. 86–108). Oxford University Press.

Saunders, B., Kitzinger, J., & Kitzinger, C. (2015). Anonymising interview data: Challenges and compromise in practice. *Qualitative Research*, *15*(5), 616–632. https://doi.org/10.1177/1468794114550439

Sayes, E. (2014). Actor–Network Theory and methodology: Just what does it mean to say that nonhumans have agency? *Social Studies of Science*, *44*(1), 134–149. https://doi.org/10.1177/0306312713511867

Schäfer, U. (1997, January 12). Die neue Gründerzeit. *Der Spiegel*, *1977*(3). https://www.spiegel.de/politik/die-neue-gruenderzeit-a-8682ade8-0002-0001-0000-000008649493?context=issue

Schatzki, T. R., Knorr-Cetina, K., & Savigny, E. von (Eds.). (2001). *The practice turn in contemporary theory*. Routledge.

Schmidt, K. (2008). *Cooperative work and coordinative practices*. Springer.

Schneider, F. (2006). The dark side of the multitude. Theory kit. *Sarai Reader*, *06*, 572–576.

Scholz, T. (Ed.). (2013). *Digital labor: The Internet as playground and factory*. Routledge.

Schoonmaker, S. (2018). *Free software, the internet, and global communities of resistance*. http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1692971

Schüttpelz, E., & Gießmann, S. (2015). Medien der Kooperation—Überlegungen zum Forschungsstand. *Navigationen: Zeitschrift Für Medien- Und Kulturwissenschaften*, *15*(1), 7–55.

Scott, C. R. (2005). Anonymity in Applied Communication Research: Tensions Between IRBs, Researchers, and Human Subjects. *Journal of Applied Communication Research*, *33*(3), 242–257. https://doi.org/10.1080/00909880500149445

Sennett, R. (2013). *Together. The rituals, pleasures and politics of co-operation*. Penguin Books.

Shaw, A., & Hill, B. M. (2014). Laboratories of Oligarchy? How the Iron Law Extends to Peer Production: Laboratories of Oligarchy. *Journal of Communication*, *64*(2), 215–238. https://doi.org/10.1111/jcom.12082

Shirky, C. (2009). *Here comes everybody: Thepower of organizing without organizations*. Penguin Books.

Siegert, B. (2013). Cultural techniques: Or the end of the intellectual postwar era in German media theory. *Theory, Culture & Society*, *30*(6), 48–65. https://doi.org/10.1177/0263276413488963

Sink, E. (2011). *Version control by example* (1st ed). Pyrenean Gold Press.

Skeggs, B. (2001). Feminist Ethnography. In P. Atkinson (Ed.), *Handbook of Ethnography* (pp. 426–442). SAGE Publications Ltd. https://doi.org/10.4135/9781848608337.n29

Söderberg, J. (2012). *Hacking capitalism: The free and open source software (foss) movement*. Routledge.

Spehr, C. (2003). Gleicher als andere. Eine Grundlegung der freien Kooperation. In C. Spehr (Ed.), *Gleicher als andere: Eine Grundlegung der freien Kooperation* (pp. 19–116). Dietz.

Spehr, C. (2007). Free cooperation. In G. Lovink & T. Scholz (Eds.), *The art of free cooperation* (pp. 65–180). Autonomedia.

Srnicek, N. (2017). *Platform capitalism*. Polity.

Stahl, M. (2021, May 19). *How to debug*. The Document Foundation. https://wiki.documentfoundation.org/Development/How_to_debug

Stallman, R. M. (2002). *Free software, free society: Selected essays* (1st. ed). Free Software Foundation.

Star, S. L. (1993). Cooperation without consensus in scientific problem solving: Dynamics of closure in open systems. In S. Easterbrook (Ed.), *CSCW: Cooperation or conflict?* (pp. 93–106). Springer.

Star, S. L. (1999). The Ethnography of Infrastructure. *American Behavioral Scientist*, *43*(3), 377–391. https://doi.org/10.1177/00027649921955326

Star, S. L. (2010). This is not a boundary object: Reflections on the origin of a concept. *Science, Technology, & Human Values*, *35*(5), 601–617.

Star, S. L., & Griesemer, J. R. (1989). Institutional ecology, 'translations' and boundary objects: Amateurs and professionals in Berkeley's museum of vertebrate zoology, 1907-39. *Social Studies of Science*, *19*(3), 387–420. https://doi.org/10.1177/030631289019003001

Strauss, A. (1985). Work and the Division of Labor. *The Sociological Quarterly*, *26*(1), 1–19. https://doi.org/10.1111/j.1533-8525.1985.tb00212.x

Strauss, A. (1988). The Articulation of Project Work: An Organizational Process. *The Sociological Quarterly*, *29*(2), 163–178. https://doi.org/10.1111/j.1533-8525.1988.tb01249.x

Suchman, L. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge University Press.

Suchman, L. (1996). Supporting Articulation Work. In R. Kling (Ed.), *Computerization and controversy: Value conflicts and social choices* (pp. 407–425). Academic Press.

Sun Microsystems. (2000a). *Star Office TM XML File Format. Working Draft. Technical Reference Manual*. http://xml.coverpages.org/openoffice-xml_specification_draft200012.pdf

Sun Microsystems. (2000b, December 8). *The OpenOffice.org Project*. OpenOffice.Org. http://web.archive.org/web/20001208110200/http://www.openoffice.org/white_papers/OOo_project/openofficefoundation.html

Sun Microsystems. (2007, September 12). *IBM joins OpenOffice.org community*. OpenOffice.Org. http://web.archive.org/web/20070912215543/https://www.openoffice.org/press/ibm_press_release.html

Sunstein, C. R. (2008). *Infotopia: How many minds produce knowledge*. Oxford University Press.

Surowiecki, J. (2005). *The wisdom of crowds*. Anchor Books.

Swidler, A. (2001). What anchors cultural practices. In T. R. Schatzki, K. Knorr-Cetina, & E. von Savigny (Eds.), *The practice turn in contemporary theory* (pp. 74–92). Routledge.

Tapscott, D., & Williams, A. D. (2010). *Wikinomics: How mass collaboration changes everything*. Portfolio/Penguin.

The Document Foundation. (n.d.-a). Code of Conduct. *The Document Foundation's Wiki*. Retrieved 1 July 2019, from https://www.documentfoundation.org/foundation/code-of-conduct/

The Document Foundation. (n.d.-b). *Home*. The Document Foundation. Retrieved 11 January 2016, from https://www.documentfoundation.org/

The Document Foundation. (2010a, September 28). *OpenOffice.org community members launch the Document Foundation*. https://lwn.net/Articles/407383/

The Document Foundation. (2010b, October 1). *Frequently Asked Questions—FAQ*. The Document Foundation. http://www.documentfoundation.org/faq/

The Document Foundation. (2011). The Next Decade Manifesto. *The Document Foundation's Wiki*. https://wiki.documentfoundation.org/TDF/Next_Decade_Manifesto

The Document Foundation. (2012, February 7). *Statutes of 'The Document Foundation'*. LIbreOffice. The Document Foundation. https://www.documentfoundation.org/foundation/statutes/

The Document Foundation. (2018, May 8). LibreOffice Easy Hacks. *The Document Foundation's Wiki*. https://wiki.documentfoundation.org/Development/EasyHacks

The Document Foundation. (2019a, March 1). LibreOffice Branding Guidelines. *The Document Foundation's Wiki*. https://wiki.documentfoundation.org/Marketing/Branding

The Document Foundation. (2019b, September 19). Developers and Contributors list. *The Document Foundation's Wiki*. https://wiki.documentfoundation.org/Development/Developer

The Document Foundation. (2021a, April 24). *Who are we?* LibreOffice. https://www.libreoffice.org/about-us/who-are-we/

The Document Foundation. (2021b, May 19). *Main page*. He Document Foundation's Wiki. https://wiki.documentfoundation.org/Main_Page

Tkacz, N. (2015). *Wikipedia and the politics of openness*. University of Chicago Press.

Vadén, T., & Stallman, R. (2002). *The Hacker Community and Ethics*. GNU Operating System. https://www.gnu.org/philosophy/rms-hack.html#pub

Van Maanen, J. (1988). *Tales of the field: On writing ethnography*. University of Chicago Press.

Velkova, J. (2017). *Media Technologies in the Making User-driven Software and Infrastructures for Computer Graphics Production*. Södertörns högskola.

Vignoli, I. (2017, December 22). LIBREOFFICE MASCOT – LESSON LEARNED AND AN APOLOGY. *LIBREOFFICE DESIGN TEAM*. https://design.blog.documentfoundation.org/2017/12/22/libreoffice-mascot-lesson-learned-apology/

Viveiros de Castro, E. (2004). EXCHANGING PERSPECTIVES. The Transformation of Objects into Subjects in Amerindian Ontologies. *Common Knowledge*, *10*(3), 463–484.

Viveiros de Castro, E. (2012). *Cosmological Perspectivism in Amazonia and Elsewhere. Four Lectures given in the Department of Social Anthropology, University of Cambridge, February–March 1998*. HAU books. https://haubooks.org/cosmological-perspectivism-in-amazonia/

Wardrip-Fruin, N. (2009). *Expressive processing: Digital fictions, computer games, and software studies*. The MIT Press.

Wark, M. (2004). *A hacker manifesto*. Harvard University Press.

Wenger, E. (2008). *Communities of practice: Learning, meaning, and identity*. Cambridge Univ. Press.

Wenger, E., McDermott, R. A., & Snyder, W. (2002). *Cultivating communities of practice: A guide to managing knowledge*. Harvard Business School Press.

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

Wexelblat, R. L. (Ed.). (1981). *History of programming languages*. Academic Press.

What is the difference between openoffice and libreoffice and why could openoffice disappear? (n.d.). Retrieved 10 April 2021, from https://en.uma-computers.com/cual-es-la-diferencia-entre-openoffice-y-libreoffice-y-porqu-openoffice-podr-desaparecer

Wheeler, D. A. (2015, July 18). Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers! *David A. Wheeler's Blog*. https://www.dwheeler.com/oss_fs_why.html

Whittle, A., & Spicer, A. (2008). Is Actor Network Theory Critique? *Organization Studies*, *29*(4), 611–629. https://doi.org/10.1177/0170840607082223

Winner, L. (1980). Do Artifacts have Politics? *Daedalos*, *109*(1).

Wittel, A. (2001). Toward a network sociality. *Theory, Culture & Society*, *18*(6), 51–76. https://doi.org/10.1177/026327601018006003

Young, R. (1999). Giving it Away: How Red Hat Software Stumbled Across a New Economic Model and Helped Improve an Industry. *The Journal of Electronic Publishing*, *4*(3). https://doi.org/10.3998/3336451.0004.304

Appendix

Overview data collection

date	place	collection techniques	phase / description
30-31 january 2016	FOSDEM Brussels	observation, interviews	establish rapport, familiarising with f/oss scene
4-5 february 2017	FOSDEM Brussels	observation, interviews	establish rapport, familiarising with f/oss scene & with LibreOffice
april 2017 - december 2018	Online: Telegram groups, email archives	observation	direct observations; studying group dynamics; detecting first patterns
10-13 october 2017	Libre Office conference Rome	observation, interviews, informal talks	participant observation; detecting patterns; establish rapport
3-4 february 2018	FOSDEM Brussels	observation, interviews, informal talks	focus on organisation & governance; recognition of patterns; establish rapport
5-6 february 2018	LibreOffice hackfest and team meetings Brussels	observation, informal talks	focus on governance & hacking; recognition of patterns, establish rapport
6-7 April 2018	LibreOffice hackfest Hamburg	observation, interviews, informal talks	focus on hacking; recognition of patterns; establish rapport
spring 2018	online	documents and records	analyses of archived records: mailing lists, news reports, press releases
spring 2018	Vide conference: Jitsi	interviews	First phase with TDF members

spring and summer 2019	Video conference: Jitsi	interviews	Second phase with TDF members

Overview interviews

#	date	place	length (min.)
01	30 Jan 16	FOSDEM Brussels	29:02
02	30 Jan 16	FOSDEM Brussels	27:04
03	31 Jan 16	FOSDEM Brussels	40:45
04	31 Jan 16	FOSDEM Brussels	22.23
05	31 Jan 16	FOSDEM Brussels	28:18
06	5 Feb 17	FOSDEM Brussels	24:44
07	5 Feb 17	FOSDEM Brussels	46:07
08	11 Oct 17	LO conference Rome	28:14
09	12 Oct 17	LO conference Rome	42:10
10	12 Oct 17	LO conference Rome	17:50
11	12 Oct 17	LO conference Rome	19:22
12	13 Oct 17	LO conference Rome	13:57
13	13 Oct 17	LO conference Rome	15:31
14	13 Oct 17	LO conference Rome	10:46
15	4 Feb 18	FOSDEM Brussels	13:20
16	4 Feb 18	FOSDEM Brussels	24:33
17	4 Feb 18	FOSDEM Brussels	08:29
18	20 Mar 18	Video conference: Jitsi	54:01
19	25 Mar 18	Video conference: Jitsi	41:53
20	25 Mar 18	Telephone interview	50:18
21	4 Apr 18	Video conference: Jitsi	53:36
22	7 Apr 18	LO hackfest Hamburg	24:54
23	7 Apr 18	LO hackfest Hamburg	30:11
24	7 Apr 18	LO Hackfest Hamburg	28:24
25	10 Apr 18	Video conference: Jitsi	44:50
26	11 Apr 18	Gothenburg	47:14
27	22 Apr 18	Video conference: Jitsi	56:28
28	3 May 18	Video conference: Jitsi	13:12
29	4 May 18	Video conference: Jitsi	22:53

30	30 Oct 18	Video conference: Jitsi	50:34
31	27 Mar 19	Video conference: Jitsi	26:29
32	13 Jun 19	Telephone interview	40:12
33	2 Jul 19	Video conference: Jitsi	38:34
34	3 Jul 19	Video conference: Jitsi	39:29
35	3 Sep 19	Video conference: Jitsi	12:51
36	4 Sep 19	Video conference: Jitsi	59:32
37	5 Sep 19	Video conference: Jitsi	17:11

Colliberate

Many people use an office suite every day. All those documents, spreadsheets and presentations are boring, right? This dissertation begs to differ. It stresses that we need to know more about software, a technology that has become the de facto infrastructure for everyday life. Colliberate is a study of LibreOffice, an office suite that you can download, use, share, modify and redistribute freely. It highlights the diversity and richness of ideas and intentions amongst the people who build and maintain it. It shows how volunteers and entrepreneurs collaborate in a new form of teamwork. It embraces the ample forms of negotiations and discussions that go into the development of an office suite. By focusing on the practices that are deployed to keep the project sustainable, this dissertation zooms in on a complex interplay of different ethical ideas, technical skills, and governance mechanisms that together form a software project.

ISBN 978-91-7867-278-3 (print)

ISBN 978-91-7867-299-8 (pdf)

ISSN 1403-8099

DOCTORAL THESIS | Karlstad University Studies | 2022:15