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

Bettina Schwab
MASTER THESIS

The Economics of Open Source User Foundations

Submitted on 31.08.2018

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

Versicherung

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

Erlangen, 31.08.2018

License

Erlangen, 31.08.2018

Abstract

Companies are facing difficult build-or-buy decisions when acquiring enterprise software. Open Source User Foundations provide an alternative way of acquisition that is especially relevant in software marketplaces where a handful of software vendors serve a large number of software buyers.

This thesis presents an exploratory qualitative single-case case study on the software ecosystem that spans around the Open Source User Foundation openKONSEQUENZ (oK). We aimed to build a theory on how software product vendors, consulting firms, and service providers sustainably collaborate in a software ecosystem whose main parameters are being determined by a consortium of software user firms.

The qualitative data analysis of expert interviews yielded an overview of the different players and their economic goals. All companies in the software ecosystem try to achieve similar technological goals. The economic conflict lies in the business model that is closely entangled with technology. The consortium wants to lower market entry barriers and change the business model, while established software vendors want to defend their market-leading position.

The economic conflict seems to be bridgeable from a neutral perspective and we offer possible starting points for conflict resolution.

Keywords

software user foundation, consortial software development, Open Source, community source, openKONSEQUENZ, NetzDatenStrom, SCADA software, distribution system operators

Contents

1 Introduction

This chapter describes the thesis goals that were developed prior to the research.

1.1 Thesis Goals

We aim to build a theory on how software product vendors, consulting firms, and service providers sustainably collaborate in a software ecosystem whose main parameters are being determined by a consortium of software user firms.

To address this question a theory concerning collaboration in software ecosystems that involve a consortium will be build via an exploratory case study.

The unit of analysis of the qualitative research is the ecosystem that spans around openKONSEQUENZ (oK). The software consortium and user foundation openKONSEQUENZ is active in the energy sector and consists of distribution system operators (DSO), but also software vendors and consulting firms. The DSOs are the software user members of the consortium. They collect their requirements and then order software together. The software is directly ordered and also developed in the course of publicly funded common projects (for example NetzDatenStrom). The ecosystem was examined as a whole with the consortium being the embedded unit of analysis.

Furthermore, economic models concerning user foundations (e.g. purchasing pools, special interest groups, etc.) will be explored in a literature review to get an interdisciplinary overview of the matter and to detect parallel problems and solutions under different circumstances.

The result should be a preliminary, not yet validated, theory on how software ecosystems around consortia (user-driven foundations) function and what effect the consortium takes on the ecosystem. The theory should cover the following aspects:

- roles of the different players in the ecosystem and how they interact
- economic goals of the involved players
- conflicts between different economic goals
- suggestions to resolve conflicts on the basis of existing models
- identification of problems that are not resolved so far – if any

1.2 Change to Thesis Goals

At an early stage of the research, the idea emerged to turn the single-case case study into a single-case embedded case study design by adding a survey as an embedded element to receive broader and more formalized information that can be utilized to demarcate the ecosystem. The survey was later omitted due to time constraints and a low benefit expectation.

2 Research Chapter

2.1 Introduction

In doing business and using software to support it, companies would have to take a long way round if they wanted to avoid Open Source Software (OSS). Even risk-averse industries have highly adopted open source (Rooney, 2012) as companies discovered the many advantages thereof – whether they are software users or software producers (Fogel, Vasile, & Finch, 2018). Nevertheless, not all enterprise software needs are addressed by classical open source communities, as Chris Mackie described:

> *"For all that the OSS movement has produced some runaway successes, including projects like Perl, Linux, and Mozilla Firefox, there appear to be certain types of challenges that are difficult for OSS to tackle. Most notably, voluntaristic OSS projects struggle to launch products whose primary customers are institutions rather than individuals: financial or HR systems rather than Web servers or browsers; or uniform, manageable desktop environments rather than programming languages or operating systems*".
>
> *(Mackie, 2008, p. 120)*

Mackie sees the following factors causing the limitation (Mackie, 2008, p. 120):

- the number of developers with special skills might be too small to build a community around an enterprise information system
- enterprise software might be too unglamorous or uninteresting to attract volunteering developers
- the benefits of the software may be too diffuse to encourage beneficiaries to collaborate in the investment and production
- the software may be too complex for its development to be coordinated on a purely voluntary basis
- the software may require the active, committed participation of specific firms or institutions having strong disincentives to participate in OSS

As suggested, a different model to develop software with the help of open source concepts is needed. Such a model could offer an alternative way to acquire enterprise software to companies. Currently, enterprises face build-or-buy decisions and there is no middle way to acquire software in most software markets.

New concepts to obtain enterprise software that satisfies the requirements and needs of commercial users are especially relevant in software marketplaces where a handful of software vendors dominate and serve a large number of software buyers. This is the case for enterprise software. The global software market is dominated by large firms which are mostly located in the US. By revenue, 16 out of the 20 global software leaders are headquartered there. (McCaffrey, 2014)

In contrast, the German economy is built up by small and medium-sized businesses (SMBs)[1]. 99.3% of the German business fall into this category. (Statistisches Bundesamt, 2015) This

1 Small and medium-sized businesses are defined as businesses that employ less than 250 employees and do not have revenues of more than € 50 million annually. (European Commission, 2003)

type of company is in a disadvantageous position when negotiating with larger software firms. Small businesses have less weight when it comes to controlling the direction of development. In many industries requirements cannot be fulfilled by simply buying standard software. Heavy customization and in-house development are necessary which leads to high cost in software procurement and vendor lock-in[2]. Open Source User Foundations could pose a valid option for businesses to collaborate and to jointly acquire software that fits their needs.

Despite the potential, research on Open Source User Foundations is limited. This thesis adds an exploratory study of a yet unobserved case to the research field. The foundation explored is openKONSEQUENZ (oK) which operates in the software ecosystem of the German energy sector, represented users being Distribution System Operators (DSOs). The last in-depth publication on non-technical aspects of oK was published in 2013 as a preliminary study (openKonsequenz, 2013). The preliminary study assessed problems faced by DSOs in the ecosystem and evaluated whether they could be resolved by applying the concept of consortial software development.
The following literature review helps to locate the case at hand in the current understanding of company collaboration to obtain suitable software. A qualitative data analysis of interview data was then used to form an empirically backed theory on the economic interests of the different players in the software ecosystem. Furthermore, early effects of the user foundation are elicited and possible starting points to bridge economic conflict in the ecosystem are suggested.

The research revealed that the technological goals towards software development are quite homogeneous independently of the player's role in the ecosystem (consortium member or non-member). What stands out is that economic conflicts in the ecosystem continue to be evolve mainly around the topic of business model change.

The companies represented in the user foundation aim at business model change. It is the consortium's goal to transform the software market structure and the business model inherent in it towards a more modular and open set-up in which vendor dependency is abolished or at least diminished greatly. It is important to note here that the business model is strongly entangled with technology: the consortium drives software architecture aspects that move away from the so far established monolithic approach towards a modular divide & conquer approach. This technological change is the basis for a different way of doing business in the ecosystem.

Incumbent software vendors (non-members of oK) are not fond of a business model change. In the current predominant model software vendors are general contractors that offer a monolithic software product and services across all stages of the software life cycle. Established software vendors want to stick with this, even though technologically they are moving towards a more modular approach concerning their proprietary software, too.

The impact of the consortium on the software ecosystem is still limited, but the user foundation sent out a clear and loud signal towards dominant vendors that the status quo is unacceptable to them.

The structure of the thesis is as follows: firstly, related literature is introduced in the next section and the case at hand further located in the upcoming research field of Open Source User Foundations. In chapter 2.2 the research question is introduced in detail. Afterwards, the research approach and the formed theory as the main research result are presented in chapters 2.3 to 2.6. The results and their limitations are discussed before concluding the work and explaining opportunities for future research.

2 Vendor lock-in is a service delivery technique that ties customers to a vendor. Customers are dependent on the service and switching costs are extremely high. (Techopedia)

2.2 Related Work

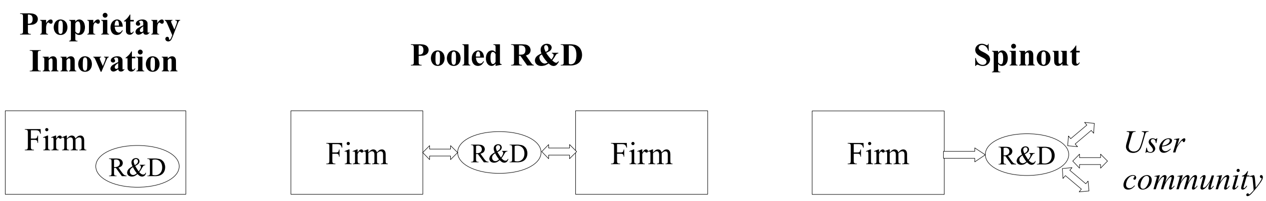
This section provides an overview of relating literature and explains how the research fits into the current state of the field.

2.2.1 Open Source and Company Collaboration

Research and literature on *Open Source Software* (OSS) have addressed various topics, e.g. the economics of OSS, OSS business models as well as the relationship between firms and OSS communities (Schaarschmidt, 2012). The practice and reasons of companies investing and contributing to open source software projects have been discussed by several researchers (Schaarschmidt, 2012, West & Gallagher, 2006, Henkel, Schöberl, & Alexy, 2014). Schaarschmidt collected concepts on how companies can influence and control open source projects. (2012)

Many publications focus on innovation and discuss revealing knowledge and innovation to external rivals. This is called open innovation and is defined as "*systematically encouraging and exploring a wide range of internal and external sources for innovation opportunities, consciously integrating that exploration with firm capabilities and resources, and broadly exploiting those opportunities through multiple channels.*" (West & Gallagher, 2006, p. 320)

Open source software development can be seen as a successful variation of collective invention when it can build on a pro-social intrinsic motivation of a critical mass of participants. (Osterloh & Rota, 2007) Companies recognized the potential to collaborate on the basis of open source and several types of collaboration strategies for research and development (R&D) have emerged (West & Gallagher, 2006):



Graph 1: Knowledge flows in software R&D models following West & Gallagher (2006)

The proprietarty innovation model is non-collaborative: a company is developing proprietary software on its own. When two or more firms collaborate on R&D the authors call this pooled R&D. In a so called spinout a firm is opening up its R&D for collaboration with user communities.

Collaborating via a consortium that consists of software users and software vendors alike with all of the members being firms is not pictured. A software user foundation is a combination of pooled R&D and a spin-out in which the user community consists of firms, too.

Contributing to the attempt of defining different types of open source projects, the Mozilla Foundation and Open Tech Strategies published a piece called *"Open Source Archetypes: A Framework For Purposeful Open Source"*. (Fogel et al., 2018) The authors present ten different types of open source projects. The user foundation approach of oK does not exactly fit one archetype, but mixes aspects of three different types (Fogel et al., 2018, p. 28):

- Business-to-Business (B2B) Open Source
  - the community is oriented towards involving organizations rather than individuals
- Multi-Vendor Infrastructure
  - multiple vendors (+ users) are involved and address a set of shared problems

together
    - governance is ensured by a committee of organizational representatives
    - the community is welcoming but formal; difficult for individuals to join
- Trusted Vendor
    - guiding principle is to avoid vendor lock-in
    - oK involves multiple vendors

We can conclude from this literature overview that user foundations do not fully fit into already established descriptions of open source projects or company collaboration (more details in chapter 3.2).

2.2.2 Definitions of Open Source User Foundations and Community Source

Open Source User Foundations[3] are still a relatively new way for companies to collaborate on software development projects. Research is limited and this gets apparent when one is confronted with the unclear naming of the phenomenon. The term *"community source"* seems to be the most established in the literature. It was either coined by Brad Wheeler (Hanganu, 2008), one of the driving forces behind the Kuali foundation, or the analyst company Gartner (Taft, 2009).

The term *Open Source User Foundation* is used in talks by Riehle (Riehle, 2014, 2016). Other names for the phenomenon we stumbled upon are "*consortial open source software development", "collaborative open source application development"*(Krause, Höfer, Braun, & Seibt, 2008), as well as "*directed open source*". (Mackie, 2008, p. 121)

Apart from the unclear naming, Open Source User Foundations have not been publicly discussed and remain an expert topic. The Wikipedia article on community source for example is very narrow and only covers on the aspect of community source development in higher education (Wikipedia). Other manifestations in different domains are not even mentioned.

Riehle defines an Open Source User Foundation as *"a non-profit organization (foundation, consortium) with the purpose of funding and managing the development of non-differentiating open source software made available to foundation members and the general public."*(Riehle, 2016, p. 6)

The research group around Liu, Hansen, and Zhao call this form of collaboration *community source* and refers back to Raymond's famous analogy of the cathedral and the bazaar (Raymond, 1999) when they describe it as *"shopping mall"* way of developing software. (Liu, Hansen, & Tu, 2014, pp. 90–91) Haganu (2008) calls community source *"the pub between the cathedral and the bazaar"*. The hybrid form between commercial software development and open source software development draws on advantages of both worlds, but is complex to handle because diverse and possibly conflicting requirements of the involved members have to be balanced. (Liu et al., 2014, pp. 90–91) Furthermore, community source requires collaborative investment decisions by the involved institutional partners. (Liu, Hull, & Hung, 2013, p. 4267) as well as the provision of human resources. (Liu, Wang, & Leon Zhao, 2012)

In addition, Feldstein defines community source as *"a particular governance and funding model for open source projects."* (Feldstein, 2014) The central part of the governance model is inter-organizational cooperation and collaboration between firms. This is a well-established

3 Members of an open source user foundation are usually representatives of the companies or institutions involved. This does not necessarily mean that the actual users (meaning: people who operate the software product to perform their work) are represented and directly engaged in the software development process. User involvement can be quite low in consortial software development, because it adds additional coordination efforts to an already complex endeavor involving many stakeholders. Furthermore, not only software user firms are members of the consortium, but usually also other participants of the ecosystem, especially software vendors and service providers.

phenomenon in many industries and can take different forms. A common aspect to be found in all cooperation is that independent companies work together to increase their collective competitiveness in the market. (Weerth, 2018) When expanding the availability of knowledge and resources, participating companies can reduce risk and decrease the cost of product development. (Fjeldstad, Snow, Miles, & Lettl, 2012, p. 734)

Cooperation provides a middle way between buying some product or a service in the free market and completely integrating the production into the own company (hierarchical control or ownership). There is a spectrum between buying in the market and producing internally – a make-or-buy decision – with cooperative forms of production located between the two extremes (Becker, 2007, p. 13). This view is extended by the thought that community source combines the benefits of in-house development and open source (Liu, Zeng, & Zhao, 2008, p. 1). Making use of an open source community for development is essentially like outsourcing to an unknown workforce of developers that can be located all over the world. (Ågerfalk & Fitzgerald, 2008, p. 386) Following this thought, community source is described as a combination of in-house development and outsourcing to an open source community. (Liu et al., 2008, p. 1)
User foundations are a new way of acquiring software and a powerful way to collaborate, but they are also harder to maintain than other forms of collaboration between companies. (Wheeler & Hilton, 2012, p. 80) The effort necessary to found and maintain a user foundation is usually very high. "*User foundations are typically created when the frustration over suppliers outweighs the (expected) hassles of the foundation*." (Riehle, 2016, p. 15)

2.2.3 Kuali – a Prominent Case of Community Source

The Kuali Foundation, founded in 2005 to provide software for higher education institutions, is the most prominently discussed case of community source and the term *community source* is mainly used to describe it. A group of researchers around J. Leon Zhao and Manlu Liu published several papers on Kuali and the community source approach, its antecedents and a framework for investment decisions (Liu et al., 2014, Liu, Wang, & Zhao, 2007, Liu et al., 2007, Liu et al., 2013, Liu et al., 2013, Liu et al., 2008).

In 2014, Kuali commercialized to speed up development of the Kuali software suite. (Hill, 2014, Hanganu, 2008) The step was explained by the need to offer software via the cloud. Initially, there was no plan to collect venture capital, however, in 2018, Kuali received a $10 million investment. (Johnson, 2018) The step towards commercialization hit the community hard (Brooks, 2016) and a blog post even proclaimed the death of the community source model. (Feldstein, 2014)

2.2.4 openKONSEQUENZ and the DSO Software Ecosystem

oK and the software ecosystem for DSOs have not been subject of research yet. Six companies, potential early members of the foundation, published a preliminary feasibility study (openKonsequenz, 2013) in fall 2013. The intended use of this study was to convince others to join the consortium. In addition to this, two publications on software architecture and quality were published from within the consortium (Goering et al., 2016, Goering et al., 2017). oK was also shortly mentioned in a technical publication on the common information model (CIM). (Requardt et al., 2017)

oK is highly interesting due to its special community features: the user foundation offers all development results as open source software to the public. Code development is agile, but not open for anyone to contribute during development. Thus, community source is not a fitting term to describe oK. In community source projects a virtual development unit is built up by developers who are provided by member institutions. The development unit is then managed cooperatively. After an initial closed phase, community source development is opened up and anyone can contribute. (Mackie, 2008, p. 121) Building up such a developer community does not lie within the scope of the oK consortium.

The main purpose of oK is to lower market entry barriers, to change the market structure and to pave a way out of vendor lock-in. In this context, open sourcing the ordered software is used as an instrument to achieve these goals. Thus, the consortium is primarily providing an administrative framework for company collaboration. (Hanganu, 2008)

This difference in governance and goal setting is the reason for adopting the terms *user foundation*, *user consortium* and *consortial open source software development* instead of community source when referring to the case at hand.

The case oK can be categorized as a user foundation that is build up in the spirit of an industry working group (Riehle, 2016). oK is especially interesting when keeping in mind the wider possibilities in the German economic sphere, mainly formed by small and medium-sized enterprises. The access of our research group to oK is good. Prof. Riehle supported oK from the beginning and our research group is a guest member of the consortium.

For oK and its member institutions, this thesis can serve as an external status assessment that could help to identify opportunities for the future.

2.3 Research Questions

The following research questions were worked out to capture the involved players of the software ecosystem and their economic goals in a software user foundation. We try to answer the proposed questions by taking a closer look at the user foundation oK and the surrounding ecosystem with the help of an exploratory single-case case study.

- RQ1: How can a software ecosystem that spans around a consortium be defined and demarcated?
- RQ2: What are the economic goals of the players in the ecosystem and where do they conflict?
- RQ3: How can conflicts be potentially resolved? Are there any unresolved problems?

2.4 Research Approach

The research method followed to answer the research questions is a case study as suggested by Yin (2010). A qualitative exploratory case study is appropriate because the unit of analysis is a complex real-world case that is not fully under the control of the researcher. Additionally, the boundaries between the phenomenon and context are not completely clear.

The unit of analysis is the software ecosystem that spans around the oK consortium. The consortium itself is an embedded unit of analysis. This thesis examines the unit of analysis on the level of organizational theory (Yin, 2010). The question to be answered is how all the companies and institutions involved in the ecosystem can collaborate fruitfully and sustainably.

Yin (2010) is in favor of constructing a theory in the design phase of the case study. The proposed theory should later be used as a template to compare the results of the case study to. To illustrate the rationale behind this, Yin (Yin, 2010) employs the example of Columbus who presented a certain theory to Queen Isabella before he went off to explore the Indies. He ended up in America, but he started off with a completely different, but verbalized goal. We did not perform any previous theory building, but instead followed the goal of theory elicitation during data analysis.

The concrete methodology was adopted from Eisenhardt (1989), who combines case study methodology and grounded theory aspects and offers a guide specifically for theory building concerning organizational theory. Eisenhardt's suggested process is iterative and also displays a tension between divergence (seeing the cases from different angles) and convergence to bring all the insight into one theoretical framework. She emphasizes the strong tie of the resulting theory with empirical evidence. (Eisenhardt, 1989, p. 546)

Software user foundations and their ecosystems are a relatively novel research domain. (Liu et al., 2013) The research questions are quite broad and targeted at establishing very basic knowledge about the domain, so Eisenhardt's approach provided a better fit.

We deviated in several aspects from the suggested method, especially concerning data collection and analysis. The deviations were mainly caused by the circumstances of the research work. The collected data consists of interview transcripts. No quantitative data was used and only one investigator coded the transcripts. The analysis of the collected data was performed after the data collection, so there was no overlap of collection and analysis. Nevertheless, adjustments to the set of interview questions to make use of special knowledge that was collected in previous interviews have been made whenever possible to make use of controlled opportunism. (Eisenhardt, 1989, p. 539) More details on methodological deviations can be found in chapter 3.5.

The within-case analysis of the single case was performed after the data collection phase ended. In case of the identification of severe gaps in data, a second round of data collection would have been conducted. This turned out to be unnecessary.

The research resulted in a theory of Open Source User Foundations on the basis of data analysis which was performed in several steps as illustrated in *Table 1*.

Coding Step	Activity & Goal
Skimming of interview transcripts and interview notes	Getting a broad overview of the discussed topics.
Free/open coding	Viewing all text passages and labeling them.
Axial coding	Finding connections and condensing the pool of labels to get a better idea which topics were mostly discussed.
Thematic networks (Attride-Stirling, 2001)	Illustrating connections between identified stakeholders of the ecosystem and the most prominent topics.
Sorting through the codes	Condensing codes and excluding topics that came up in only one interview.
Further condensation	Condensing and ordering codes. Six main topics were identified, forming the basis of the resulting theory.

Table 1: Analytical process

2.5 Used Data Sources

The case of openKONSEQUENZ was mainly selected because of the good access to interview partners through Prof. Riehle's network.

From December 2017 to March 2018, six semi-structured interviews with representatives of players in the ecosystem were conducted. One interview was performed in person, the others via phone or teleconferencing. All interviews were performed in German. The interviews focused on the motivation and goals of the represented companies or institutions to participate or not to participate in the consortium. Furthermore the noticed effects of oK on the software ecosystem were discussed. The interviews were transcribed and analyzed with the help of the qualitative data analysis tool MaxQDA.

The selection of the interviewees is presented in Table 2. It was led by the goal to gain a broad insight into the software ecosystem. We interviewed at least one representative of each member type of oK (driver members, service provider members, guest members) and one representative of a software supplier that is not part of the consortium. Our interview partners fulfill different roles and we were therefore able to collect a variety of interesting industry perspectives. Nevertheless, not the full range of players that can be identified in the software ecosystem were covered as can be seen in *Graph 2* in the following chapter.

Interview partner	**Role in company**	**OK Membership**	**Role software ecosystem**
Interview partner1	User representative/ domain expert	Yes (Driver Member)	Customer/ user
Interview partner 2	Project lead	Yes (Service Provider Member)	Software supplier
Interview partner 3	Development lead	No	Software supplier
Interview partner 4	Consultant/domain expert	Yes (Guest Member)	Service provider/ consultant
Interview partner 5	User representative/ domain expert	Yes (Driver Member)	Customer/ user
Interview partner 6	Business Development	Yes (Service Provider Member)	Software supplier

Table 2: Interview matrix

All the interview partners wished to stay anonymous. To fully respect this wish the characterization of the interviewees and the companies they represent is kept coarse-grained in Table 2 to avoid the possibility of deriving identities from characteristics. We believe that the value of insights gathered is not diminished by the anonymization. Finer characterizations were used in the selection process. Each person who was interviewed received a copy of this thesis before publication to collect approval.

The general interview guidelines were tailored before each interview to gather appropriate information. Nevertheless, the interviews were semi-structured and rather followed the natural flow of the conversation that unfolded. This flexible way of interviewing is suggested by

Eisenhardt (Eisenhardt, 1989) as it allows topics to emerge that were not previously anticipated.

2.6 Research Results

In short, the theory that will be presented in this chapter provides the following insights:

RQ1: How can a software ecosystem that spans around a consortium be defined and demarcated?

The ecosystem is fluent and cannot be clearly defined. A less rigid way of defining the ecosystem by locating the impact strength of the consortium is suggested to handle the moving target (see chapter 3.4 for details).

RQ2: What are the economic goals of the players and where do they conflict?

The economic goals of the players differ as expected from role to role. There is a high homogeneity of goals within the consortium. The main economic conflict in the ecosystem as a whole revolves around business model change. Further challenges are caused by a shortage of developers and internal stakeholder issues (see chapter 3.5 and 3.6).

RQ3: How can conflicts be potentially resolved? Are there any unresolved problems?

The business model conflict will most likely be solved through market mechanisms or through a better understanding among user companies and software suppliers. Company internal conflicts could be solved by employing best practices from business theory to gain management support and communicate well with stakeholders.

The detailed theory is structured according to the research questions. The theoretical basis that forms and underlies the theory are six thematic categories that were identified during data analysis.

These topics are:

- Business Model
- openKonsequenz Community
- Market Structure
- Software & Complexity
- openKonsequenz Stakeholders
- Open Source

A more detailed overview of the categories and labels that emerged during data analysis can be found in Appendix B.

The consortium has not reached a stage yet in which it can determine the main parameters of the software ecosystem. The consortium is heard and reached some effects in the ecosystem, but the influence on the ecosystem as a whole was perceived lower than originally assumed when designing the thesis topic.

2.6.1 The Software Ecosystem around oK

The range of software used by Distribution System Operators (DSO) is quite broad and it is hard to draw a strict line to demarcate the software ecosystem around oK. The information collected in the interviews did not allow for a clear-cut in this regard. For example, one discussion evolved around enterprise resources planning software (ERP). Each DSO uses ERP software, so ERP system providers are part of the software ecosystem. Still, ERP software by itself is not part of the current scope of oK.

The consortium’s scope is quite clear: activities are focused on software systems that are used in grid operation management and are closely connected to the core control unit, also called the supervisory control and data acquisition (SCADA) system. The SCADA system itself is not part of the scope.

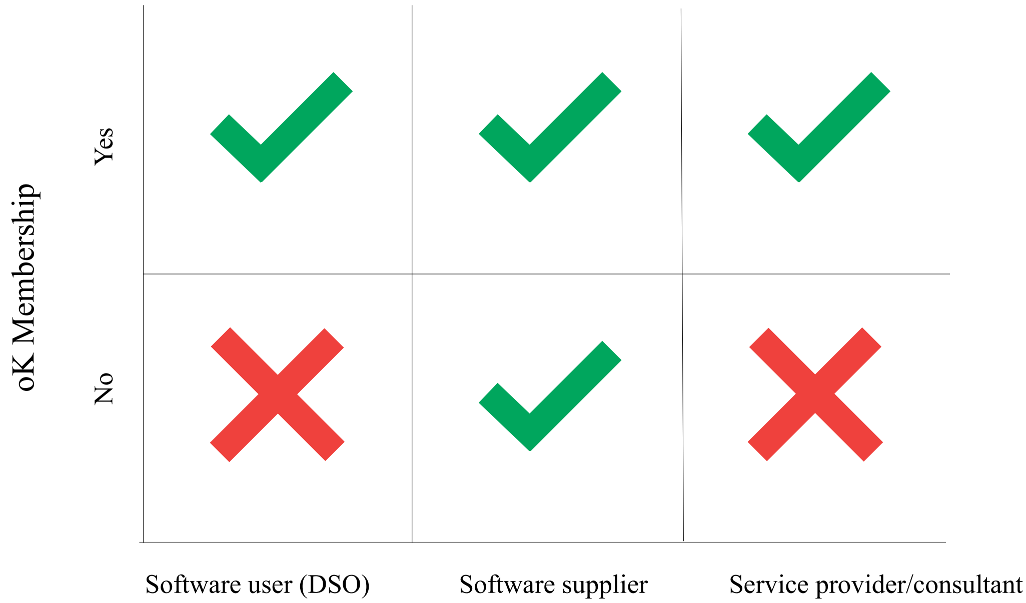
This focus occurred, because DSO employees working close to the SCADA system are well acquainted with each other as they are part of a user group organized by one software vendor. Even before oK, there were attempts to join forces and thereby give common demands more weight. Grid operation management is a very technical and independently working department and grid operations are handled very similar by all DSOs, even if requirements and software scope differ (i.e. some DSO manage gas and water grids, others do not).

The German energy transition introduced many new regulations and forced the grid operation departments close to the SCADA software systems to be open-minded and forward thinking. The law for the digitalization of the energy transition issued in 2016 (Gesetz zur Digitalisierung der Energiewende, 2016) pushes DSOs even more to gain new insights into possible technological innovations.

We, therefore, identified the biggest impact of the consortium on the software ecosystem on those software vendors that offer products closely connected to the SCADA system. Software vendors that are not impacted now, could become part of the direct ecosystem in case consortium focus shifts in the future. The boundary is fluent and a clear-cut distinction not possible. A suggestion to describe the software ecosystem graphically to create a basis for discussion is discussed in chapter 3.4 in the elaboration chapter.

2.6.2 Economic Goals of the Players in the Ecosystem

The software ecosystem around the oK consortium is a multi-faceted unit of analysis: The different players can be split up further into their organizational units and even down below to the individual actor representing her company. When describing the economic goals of the players in the ecosystem we will focus on the two-level organizational perspective: the companies and institutions active in the ecosystem and their internal organizational structure. The perspective of individuals is excluded because of its subjective nature.



Graph 2: Interview coverage of different players

The empirical data collection process did not cover all the players that can be identified in the software ecosystem. The graph above shows gaps in the interview coverage. No software user that is not part of oK was interviewed. Furthermore, we did not talk to a representative of a service provider or a consultancy that is not actively engaged in oK. Other interview partners sometimes included information about companies that fit these two categories. These statements were collected and taken into account, but nonetheless handled with care, because they could be biased or based on pure assumptions. In the following, such information is flagged.

In the following the above matrix is traversed from the top left field to the bottom right field. The last subchapter (2.6.2.7) discusses stakeholders within companies of the software ecosystem.

2.6.2.1 Software User, oK Member

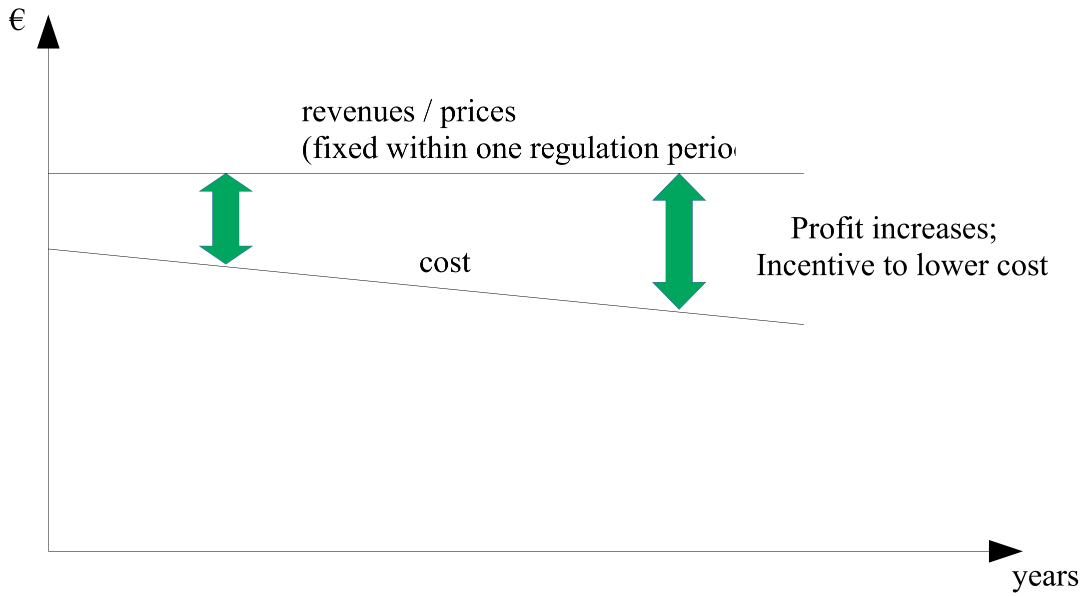
Software suppliers are for-profit companies and their first and foremost goal is to ensure their further existence by covering at least their expenses and optimally earn more to make profits.

The trigger for DSOs, which form the group of software users, to organize themselves in a consortium were not primarily economic goals, but software delivery and quality issues that could not be avoided due to vendor lock-in.

Insufficient software quality and ergonomics and a long delivery time for new functionality made the vendor lock in hard to bear. This was mostly felt during maintenance and support phases of the software life cycle. Sometimes, new maintenance and support contracts were not closed because software vendors tried to dictate prices and no agreement could be reached.

The main economic goal of the software users that are part of oK is to break the vendor lock-in and thus, to provide themselves with more flexibility in software procurement. The software users want to be able to choose the offer that fits their needs best. The pricing of the software plays a role, but quality and functionality are even more important factors to DSOs.

DSOs are regulated by the German Federal Network Agency (Bundesnetzagentur), however, besides their focus on quality and functionality, they still operate profit-oriented. DSOs do not actually compete with each other because their activities are regionally exclusive (natural monopoly). The Federal Network Agency set up an incentive regulation that incentivizes DSOs to lower operational cost.



Graph 3: Basic principle of incentive regulation (Rogat, 2015)

One year serves as a base year which is used to calculate costs. On this basis, the regulated pricing of the following five years is set. DSO can increase their profits by decreasing their operational cost.Therefore, DSOs look for ways to lower their costs. They want to achieve this in a three-folded way concerning software:

- lowering costs of development through collaborative investment
- lowering costs in software procurement
- lowering costs in grid operations through better software tools

Investments in oK are done collaboratively. A DSO who wants to develop a module takes over a leading role (driver role) in the resulting project and usually covers 70% of the necessary financial investment and human resources (one or two employees who fill the product owner role during the agile development of the module). The rest of the DSO members in oK will cover the remaining 30% of the cost. This principle ensures that development costs are distributed among the DSOs which is economically beneficial compared to in-house development financed by a single DSO. Trust is a very important factor in adopting foundation developments later on. (Parry, 2009)

DSO representatives reported in the interviews that they want to choose the best software tools for their needs to operate their grid more efficiently and thereby lower operational costs. Choosing the best fitting modules is only possible, when the used software follows a modular approach and offers standardized interfaces. If the consortium can reach its goal to offer a vendor-independent open source platform, DSOs are more flexible in software selection and can combine modules from different vendors to reach optimal operational efficiency.

A side effect that could occur is that prices to procure software go down[4] because DSO have

4 Several interview partners stated that software licenses for grid management software are not priced high compared to business software (i.e. ERP system licenses) anyway. Margins for software vendors are low already, so the effect might not occur.

more freedom to choose which puts them into a better negotiation position.

A secondary and non-economic goal pursued by DSOs goes hand in hand with the open source approach: the range of possible service providers should be enlarged to reach even more flexibility and an even better negotiation position when procuring software.

2.6.2.2 Software user, No oK Member

We did not collect any direct empirical data by interviewing software users that are not members of oK.

All network providers in Germany are regulated. We can safely assume that all DSOs pursue the goal to lower their costs and thereby increasing profits.

Non-member DSOs do not collaboratively invest in oK, so if they made use of the resulting open source software, they could benefit from the produced software without contributing. This behavior is called free riding and is a threat to the consortium because the critical mass of contributors might not be reached. So far, no cases of free riding were known to the interview partners.

In general, the consortium would like to motivate smaller DSOs (municipality networks) to join the consortium. Even free riding could be positive to establish the introduced modules and technologies and to increase the impact of the consortium.

2.6.2.3 Software Supplier, oK Membership

The term *software supplier* subsumes companies with very different characteristics:

- **Product and service portfolio:** some software vendors sell SCADA systems and other modules and services along the software lifecycle. Other software suppliers are IT service providers that develop modules on-demand and support and maintain them afterwards.
- **Form of organization:** a number of software suppliers are incorporated; others are companies with limited liability. Some are owner-operated.

All software suppliers stated that their current investment of time, expertise and membership fees to oK is higher than the short-term profits that are made via the consortium.

One software supplier which mainly provides on-demand development noted that the economic short-term goal is to win future tendering processes or to take over other tasks (e.g. quality assurance) to earn some money through the consortium. All software suppliers, independent of their oK membership, compete for these orders.

In contrast, a quite established supplier member stated that the volume of the oK orders is not very high and has not reached a level that would enable short-term profits in the course of a year for any of the service provider members.

Generally, the interviewed software suppliers engage in oK because they strive for long-term goals. Software suppliers want to be a thriving power in reshaping the structure of grid management software and therefore, the market structure of the software ecosystem. They want to reach modularity and standardization of interfaces in the software to enable a market change that moves from very general software providers (general contractors) to more specialized software providers. Software suppliers actively engage in improving consortium processes and knowledge sharing as well as promoting consortium ideas in public and research.

In the long-run, oK software suppliers are aiming to have an advantage over non-oK suppliers because they can actively add their views to technical discussions in the consortium. This helps them to ensure that their own products will be compatible and competitive in the future.

Being part of the consortium is a signal to buyers that the software supplier is a trusted vendor. (Fogel et al., 2018, p. 22)

Part of this strategy is accomplished by donating modules to the consortium. Such donations need to be adapted and fitted to consortium quality standards and guidelines. A donation is attractive for oK software provider members because they can spread their technologies and strengthen their future market position. Developers are motivated when their work is open sourced and when they know it will be put to a good use.

2.6.2.4 Software Supplier, No oK Member

Software suppliers that are not part of oK are just as diverse in their characteristics as member companies.

Software suppliers that are not part of oK are just as diverse in their characteristics.

We interviewed one software vendor which is in a market leading position and not part of the consortium. Some remarks about the motivations of other software suppliers were added by the neutral interview partner (guest member) and included here.

As stated above, software suppliers are companies that want to ensure their existence and make profits. The interviewed software supplier additionally aims at keeping the market leader position in Germany, as well as growing market share in international markets.

Other software suppliers would like to maintain the old market structure and business model to ensure predictable revenues. One player wants to promote the development towards cloud technology more quickly.

2.6.2.5 Service Provider/Consultant, oK Member

The interviewed guest member is categorized as a service provider and consultant because this reflects best the offers made to the DSOs.

One economic interest of this stakeholder group is to sell consulting projects. However, their more important goal is not directly economical: a close industry collaboration and access to knowledge and technical systems are invaluable in doing meaningful research.

2.6.2.6 Service Provider/Consultant, No oK Member

No representative of a player that falls into this category was interviewed and we do not know anything about their economic goals first hand.Remembering the conditions for founding a user foundation, we can assume that these companies expect the hassle of joining oK to be bigger than the expected positive outcome.

It is presumably interesting for oK to win more of these members for consortium collaboration to ensure access to domain experts and research concerning future developments.

2.6.2.7 Company-internal Stakeholders

All interviews were conducted with company representatives who work in departments that are closely connected to oK. We did not collect any empirical data on the opinion towards oK from other departments, but company-internal stakeholders and their actions and presumed goals were discussed during the interviews. Due to the lack of empirical data on the actual goals of these other internal stakeholders, we refrain from making assumptions and only address two points that came up repeatedly in several interviews.

Firstly, the goals and agenda of management have a crucial impact on the resource allocation and success probability of oK. Management goals differ strongly within all companies of the software ecosystem.

econdly, other departments and their employees are also internal stakeholders with a diverse set of goals. Especially the internal IT departments of DSOs were perceived to be another important player. The agendas of the IT departments vary widely, and this can lead to several different stances towards oK. All departments that are affected by changes caused by oK are stakeholders of the consortium. For example, the new way of procuring software even affects business units, i.e. the accounting department. Therefore company-internal stakeholders should not be neglected when thinking about the software ecosystem.

2.6.3 Conflicting Economic Goals

Keeping the different players and their goals presented in the previous chapter in mind, we can now turn to the question of conflicting economic goals.

The interviews revealed some economic target conflicts in the software ecosystem, but they appear to be less intense than expected. Homogeneity of goals within the oK consortium is high. Even though the different member classes have different economic goals, these play a secondary role and the common goal is emphasized by all members. The service provider members collaborate in the consortium, but they are aware that they are still competing companies. The economic target conflict is, therefore, to be located between consortium members and institutions of the ecosystem which are not part of oK.

2.6.3.1 Business Model Change

The main economic conflict occurs between those who want to change the technological approach (monolithic vs. modular) and therefore the business model of software vendors and the market structure of the ecosystem and those who would like to stick with the current predominant business model. The entanglement of technical and economic goals is very high and both aspects cannot be fully separated. The conflict was willingly provoked by founding oK and still remains to be the same until now.

A modular approach to the software architecture is the basis for the business model which is then again the basis for economic conflict. Enabling a different business model could lead to a profound change of the market structure. Vendor lock-in was and is possible because software vendors and service providers still have more power in the market than the buyers (DSOs).

The consortium pursues business model change and attempts to achieve this by altering the technical groundwork that led to the vendor lock-in situation in the first place. The plan is to change the underlying business model by changing the software technology from a monolithic approach to a divide & conquer approach, which is based on a vendor-independent platform that enables software modularity.

Market leading software vendors see a threat to their current way of earning money and their management is not fond of the development. On the other hand, no actions to fight the consortium have been taken, yet. The interview partner described the attitude towards oK as *curious* and mentioned that the impact of the consortium was expected to be much higher and that this impact is not clearly felt within the established software provider.

2.6.3.2 Old and New Business Model

Following the old and still predominant business model sofware providers are general contractor. All the services and products included in the software lifecycle were bundled together and provided by a single software vendor. The only other player usually involved were application operators (data center providers).

This old business model delivered operational security to the software vendor, but also to the DSOs. The pricing and maintenance were fixed for a certain period of time (in most cases 5 years or more). Bundling information goods can be beneficial to a vendor because it is easier to predict the customer's valuation for the bundle than for an individual information good. (Bakos & Brynjolfsson, 1999)

In contrast, the business model that is favored by oK and based upon modular technology is aspiring the unbundling of goods and services. The same aspiration was also described by Brad Wheeler of the Kuali Foundation. (Wheeler, 2004)

2.6.3.3 Changing the Market

The consortium wants to change the rules of the game in the ecosystem and move from buying software with little influence (conventional procurement) to specifically ordering software and having very strong influence and authority over its development. Established software vendors are not in favor of this because they prefer to stick with the known model and be certain to maintain influence, and even more importantly, authority.
The consortium wants to change the rules of the game in the ecosystem and move from buying software with little influence (conventional procurement) to specifically ordering software and having very strong influence and authority over its development. Established software vendors are not in favor of this because they prefer to stick with the known model and be certain to maintain influence, and even more importantly, authority.

It is quite interesting that the form of organization and the internal hierarchical set-up of companies seems to have only a secondary impact on the attitude of the institution towards oK. No rule of thumb like "Any AG will not like the business model changes suggested by oK" or "Owner-operated service providers are more likely to recognize opportunities created by oK" can be identified.
According to the interviewees, the attitude towards the consortium is mainly dependent on the perspective of decision-makers on the software ecosystem.
This leads to further challenges discussed in chapter 3.5.

2.6.4 Conflict Resolution

We can think of three scenarios that would resolve the business model conflict. Evaluating the three scenarios from our neutral point of view, two of them could be assessed as being a success for oK and its mission, while one scenario amounts to consortium failure.

1) **Conflict resolution – oK success:**

- **Convincing arguments/change of opinion:** the established software vendors, which are not part of the consortium yet, understand the consortium's goals better and are convinced by its arguments. They change their opinion and recognize the opportunities that come up when their business model gets more flexible.
- **Pressure:** oK is very successful in developing the platform and providing useful software modules. Pressure on established service providers is so high, it becomes the best economic option to cooperate with the consortium or even to become part of it.

2) **Conflict resolution – oK failure:** oK does not reach its goals or cannot maintain a critical mass of members. Consortium failure would probably lead to a situation in which the old market structure sustains.

2.6.4.1 Resolving Business Model Conflict

The main economic conflict "*changing the business model vs. keeping the established business model*" is one that will be solved by standard market dynamics. All three scenarios sketched above are possible and it is not clear yet which business model will dominate in the long-term. Can the consortium change the rules of the market? Or will their efforts not be effective enough? There is no conflict resolution mechanism towards business model changes except for standard market mechanics. It is possible that the two business models will exist next to each other. One business model might get stronger over a longer period of time or a disruptive change occurs.

The interviews showed that new innovations from smaller companies and start-ups are coming up. One interview partner uttered concerns that software suppliers are not sufficiently aware that change could happen very quickly. Especially in the area of low voltage metering and control, cloud technologies and intelligent grid and IoT platforms innovations are being developed. These are not interesting for the incumbents because only niches are served, still the potential for a classic Innovator's Dilemma as described by Christensen (Christensen, 1997) can be detected.

As soon as standards such as the Common Information Model (CIM) are widely in use, the market for DSO software gets more accessible and new players can establish more easily. A modular approach to grid operation software and the open source philosophy lower market entry barriers further.

Astonishingly, during the interview analysis, we, from our neutral, external viewpoint, received the impression that the business model conflict is bridgeable. Statements of interview partners lead to the conclusion that there are some blind spots when it comes to shared characteristics. An important factor of the economic conflict concerning the business model is the intense entanglement with the technological approach to building up the software used for grid operations.

The following technological goals were verbalized by all interview partners:

- both the consortium and the established software vendor try to reach modularity and tackle complexity with a divide & conquer strategy.
- standardized interfaces are favored by all. The consortium wants to open source them, while the software vendor did not make a statement in favor of open or proprietary interfaces. The advantage of a common data format and standardized interfaces was recognized by all interview partners.

Even though the technological goals are quite similar, software suppliers that are not part of the consortium are not fond of a business model change that is enabled by modular software architectures. Certainly, the fear of losing business is given and for a vendor in a market leading position, this is justified. Additionally, some practical issues have been raised by the interview partner:

- legal issues
- warranty issues concerning reliability and availability of the software
- development structure incompatible with the oK approach
- philosophy of not using third party software – understanding of the whole code basis is crucial for adequate support

The established software vendor that is not a member of oK communicated numerous activities that could make a cooperation more likely in our opinion:

Partnering with implementation partners: the software supplier is building up a partner network to internationalize their product distribution. This is a step away from the German business model of being a general contractor that occupies the whole software lifecycle, towards a more product-oriented business model in which others take over services and support in later phases of the software lifecycle.

The main value of work is not seen in the code: the main value of the work is located in the collaborative development of concepts together with DSO clients to solve very specific engineering problems and provide solutions to specific requirements.

Outsourcing of coding activities: some implementation activities are being outsourced. The delivered code needs revision and runs through a quality assurance process. The same effort could be invested in oK modules to ensure their quality – the "no third party software" philosophy problem could be resolved as soon as this is understood well enough by top management. rThe understanding needs to build up that open source code is basically nothing else than code that was produced via outsourcing – just for free. This factor offers great potential to save resources to software suppliers, especially SCADA system providers as they can focus on their core capabilities.

Interesting technological approach: the software architecture and concepts that were developed in the oK consortium are admired by the established vendor and software engineers are keen on using the new technologies the consortium employs.

When oK was founded, the established software vendor perceived the signal very strongly and expected to lose some big DSO clients. The expectation on the software vendor's side was that oK will directly attack their core business. However, this is not the case and even leading oK driver members recently bought upgrades of the proprietary system from the software vendor. The interviewed oK members all stated that the core module of the grid control software, the SCADA system, is surely not a short-term and also not a mid-term project appropriate for consortial software development.

One DSO interview partner called for more pressure and was pessimistic about convincing the software vendor that is currently dominating the market to join if no business pressure can be built up by the consortium. The interviewee on the software suppliers side that is not part of oK stated that the opinion towards oK is to observe it curiously. Joint research projects are conducted together and can help to raise understanding for each other's position and to bridge the economic conflict.

A new business model could help software suppliers handle their projects more easily, specialize and be confronted with less development resource problems. A collaborative conception of the market conditions and the business model could also help established software suppliers to invent their own business anew. Collaborating in the consortium means to engage and actively create a business model and market change. Participating software vendors can be sure to be well prepared for market developments if they do so. In the long-term a modular instead of a monolithic approach to the grid operation management software systems is necessary to handle complexity as grid management systems have to be real-time efficient and data is produced very quickly. Additionally, innovations in smart grid, IoT that are coming up ask for more speed and flexibility – concerning technology, as well as business models.

Many other industries unbundled years ago and follow a more modular production concept, for example the automotive industry. This allows focus and specialization and suppliers can then provide superior, specialized services in their niche compared to a general contractor.

When technological change occurs, business model change will follow. One suggestion to relax the situation between the different players and to avoid an "us and them" kind of thinking was brought up in one of the interviews: it could be helpful not to talk about the current coarse-grained roles of the players (DSO, software suppliers, consultants…), but instead to introduce finer granularities by using tasks in the software lifecycle and the ecosystem as distinction characteristics. This opens up a more flexible playing ground to come to new agreements and divisions of work among all the companies in the software ecosystem.

2.6.4.2 Unresolved Conflicts

The business model conflict is an unresolved conflict. Nevertheless, we decided not to bring it up in this chapter because business model changes are usually causing conflicts which are inherently solved through standard market dynamics. Mostly the business model changes occur through innovation or a business model change was triggered by a single company. In the case of oK several companies formed a community to change the business model willingly.

2.7 Discussion, Limitations and Future Work

The findings of this research concerning economic conflicts in the software ecosystem are of a rather general nature. The main conflict that was identified is evolving around business model change to break vendor lock in. This is not a new insight because this conflict was essentially willingly established by the user consortium oK in the first place. The main reason for persistence of this conflict is the low maturity of the user foundation: most interview partners mentioned that oK is still in an early phase of its development and will need more time to impact the ecosystem.

The empirical evidence collected here points towards something else: the slow advancement of the consortium might allow to open doors that seemed to be shut. Technologically, many goals are congruent even with non-member institutions. At least the interviewed representative of one incumbent suggested that even though they are not in favor of business model change, the technological advancements of the consortium towards the standardization of interfaces is very interesting.

Consortium members expect that it will take another 5-10 years until oK will have a real effect on the software ecosystem around DSOs. We see a chance of maintaining the consortium goal to develop a vendor independent platform that allows a flexible and modular use of technology to control smart grids, but at the same time to unite the whole software ecosystem and lead a standardization effort in the industry.

This study followed a methodology suggested by Eisenhardt. (Eisenhardt, 1989) And while building theory on cases comes with the strength of close tie with empirical data and a high likelihood of generating novel theory, there are also some downsides concerning the choice of methods. The results are often rather modest and no grand theory emerges. This is the case here. Nevertheless, a case study research was the best choice to examine this complex real life case.

The theory developed in this thesis turned out to be more specifically explaining the case of oK than originally intended. A single-case case study is not an optimal base for theory building. Nevertheless, the findings add to the research field by discussing a so far unobserved case and its current status.

The gain of general knowledge about user foundations and consortial software development is limited. Other cases consortias need to be studied to create a more general theory on user foundations and to find similarities and differences via a comparative study.

2.8 Conclusion

This thesis presented an exploratory qualitative single-case case study on the software ecosystem that spans around the Open Source User Foundation openKONSEQUENZ. After taking a look at the current status of research and clearing up different definitions, oK was identified as a special case that does not thoroughly fit in any of the established categories of open source projects in the literature.

The qualitative data analysis of the interviews yielded an overview of the different players in the ecosystem and their goals. All companies in the software ecosystem try to achieve similar technological goals. Conflict lies in the business model that is closely entangled with technology. The consortium was found to change the rules of the game in the ecosystem by offering an alternative platform and standardized interfaces. This technological modular divide & conquer approach is the basis for aspired business model change. To break up vendor lock-in the consortium tries to lower market barriers. Software vendors get the chance to specialize. The established software vendors are not in favor of the business model change, but we suggested possible starting points for conflict resolution.

We hope that the insights gained through this research help all companies, which operate in the software ecosystem for DSOs, to better understand the positions other players take and to see similarities that help to shape the future of the software ecosystem.

2.9 Acknowledgments

First and foremost I want to thank all the interview partners who made this thesis possible. I appreciate that all of you took out the time to talk to me and to share your views and experiences.

I am very grateful for Professor Dr. Dirk Riehle's supervision of this master thesis.

Additionally I want to thank Professor Dr. Joachim Henkel (TU Munich) for his pointers in the beginning of my research process and Ann Barcomb and Nikolay Harutyunyan for spending some time on discussing and commenting aspects of my work.

3 Elaboration Chapter

The elaboration chapter is a collection of different aspects of my work that did not make the cut for the research chapter, but are related to the topics discussed there. References in the research chapter point to the information in the following subsections.

3.1 Open Source Software as Commons and the Aspect of User-led Innovation

In many domains, intellectual property rights are perceived to support innovation by preventing a *Tragedy of the Commons* situation in which rational behavior of a single player leads to collective disadvantage.

A *commons* is defined as a resource shared by a group of people. (Hess & Ostrom, 2011, p. 4) Commons are usually accompanied by problems in use, governance and sustainability. Materialistic commons are of a substractive nature so that the use of these resources reduces the available benefit for others. They suffer from behavior such as over-harvesting and free riding. In contrast, knowledge resources are non-substractive and even profit from sharing. (Hess & Ostrom, 2011, p. 6) Knowledge commons are threatened by commodification or enclosure, pollution and degradation and non-sustainability in the long-term. (Hess & Ostrom, 2011, p. 5) Software can be treated as such an immaterial knowledge resource and even shows some additional special characteristics:

Source code can be easily compiled into an application, but it is nearly impossible to infer back from the application to the original source code. In a sense, source code is very similar to recipes. (Osterloh & Luethi, 2008) Applications, in parallel, are like cake: you can use them and they make you happy, but if you want to recreate or tweak them it is very hard, because you do not know the original recipe and therefore cannot use it as a basis for your work. If you do not have access to it, you might never find the secret sauce you need. You are also unable to improve the cake or develop a variation that fits your needs. That is the underlying problem of proprietary software and the root cause for the establishment of the open source movement.

Open source software is a public good and non-rival (Hippel, 2001, p. 975) in the sense of a knowledge common. In an open source developer community the whole cycle of innovating, developing and consuming a software product is performed by a community that consists of users. (Hippel, 2001)

One crucial advantage of such user-led innovation and development in communities is rooted in the stickiness of knowledge about needs towards products: users' needs change frequently and it is hard to make them explicit and to reproduce the conditions under which they come up. Manufacturers cannot know in all detail what users want, but users themselves do and no costly transfer of information is necessary.(Hippel, 2001)

Another point made by Hippel (Hippel, 2001) is that manufacturers do not have an incentive to match user needs exactly, but rather to aim at generalizing in order to reach a broad group of customers.

Three different conditions have to be met to make user innovation communities probable:

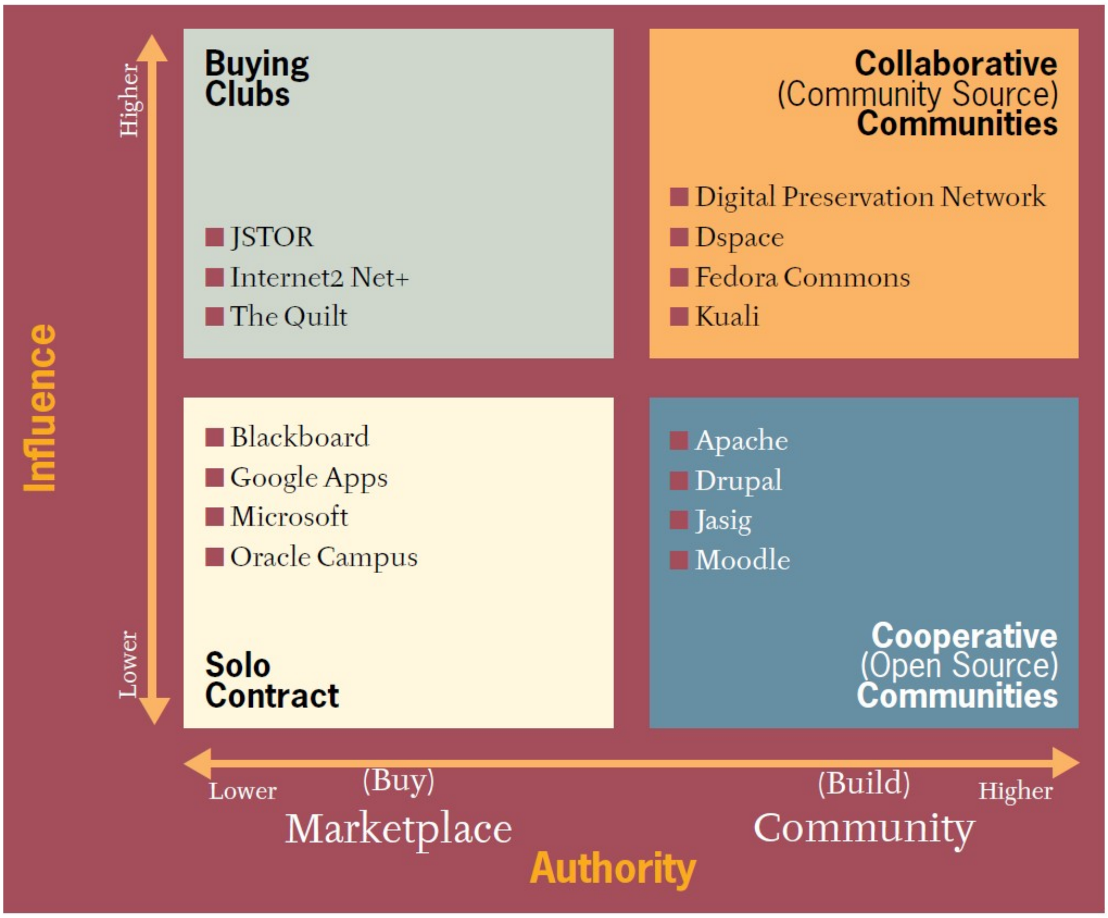
- users have an incentive to innovate – expected benefits exceed their costs. Mostly lead-users see this incentive.
- users have an incentive to reveal their innovations freely
- the diffusion of the innovation can compete with commercial product diffusion

These conditions are met for developer communities and similar prerequisites seem to be necessary to enable Open Source User Foundations, too. The main difference is that the collaboration happens between companies, not individuals.

3.2 Company Collaboration and Build-or-Buy Decisions

Cooperation provides a middle way between buying some product or a service in the market and completely integrating the production into the own company (hierarchical control vs. ownership). This leads to a spectrum between buying in the market and producing internally – a make-or-buy decision – with cooperative forms of production located between the two extremes (Becker, 2007, p. 13). This view is extended by the thought that community source combines the benefits of in-house development and open source (Liu et al., 2008, p. 1). Making use of an open source community for development is essentially like outsourcing to an unknown workforce of developers that can be located all over the world. (Ågerfalk & Fitzgerald, 2008, p. 386) Following this thought, community source is described as a combination of in-house development and outsourcing. (Liu et al., 2008, p. 1)

A software user foundation is in a certain sense a purchasing pool for software that goes further: the common goal is not only to reach a better negotiation position and therefore a better pricing, but to gain influence over the software development. This has been called “borrow path”. (Hanganu, 2008) Additionally, the opportunity to engage is open up for any institution because the developed software is open sourced. Nevertheless, it is usually another motivating factor to reach a financial benefit over single contract software purchasing. Companies that do not engage in the Software User Foundation could make use of the software for free and gain an even bigger financial benefit, but lack influence and decision power in such a governance model.
Brad Wheeler, former leader of the Kuali foundation suggests to introduce influence as a factor and to build up a matrix to map the different types of building and purchasing software. (Wheeler & Hilton, 2012)



Graph 4: Marketecture matrix by Wheeler & Hilton (2012)

Authority, depicted on the horizontal axis, describes means of resolving conflicts. This can be done via a market of buyers and sellers or the ownership authority. This axis therefore describes the same circumstance as expressed in our previously described spectrum.

The vertical axis describes institutional influence. A classical and well-known form of open source projects is described as cooperative community. The community owns the authority to control the direction of software development; single players can influence the development only to a limited extent.

3.3 More on Methodology and Data Sources

As described in the research chapter our actual research process deviated in some aspects from the roadmap suggested by Eisenhardt. (Eisenhardt, 1989) Table XXX gives an overview over the suggested steps and the deviations.

Research Step	Activity	Deviation
Getting Started	Definition of research question, Possibly a priori constructs	No a priori constructs
Selecting Cases	Neither theory nor hypotheses, Specified population, Theoretical, not random, sampling	
Crafting Instruments and Protocols	Multiple data collection methods, Qualitative and quantitative data combined, Multiple investigators	Only one data collection method (interviews), only qualitative data and one investigator
Entering the Field	Overlap data collection and analysis, including field notes, Flexible and opportunistic data collection methods	No overlap of data collection and analysis. Due to outside conditions interviews were completed before analysis started
Analyzing Data	Within-case analysis, Cross-case pattern search using divergent techniques	Single-case case study – no cross-case patterns possible
Shaping Hypotheses	Iterative tabulation of evidence for each construct, Replication, not sampling, logic across cases, Search evidence for “why” behind relationships	Single-case case study – no cross-case activities
Enfolding Literature	Comparison with conflicting literature, Comparison with similar literature	
Reaching Closure	Theoretical saturation when possible	

Table 3: Research roadmap and deviations

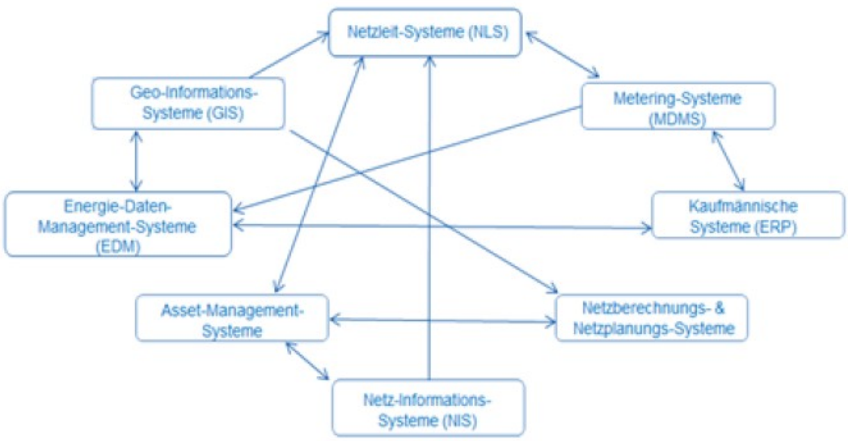
Coding and analysis was only performed by one investigator. Eisenhardt suggests to have multiple investigators as this usually enables better and more creative insights. If several investigators converge, the resulting theory is supported better and the findings can be promoted with more confidence. (Eisenhardt, 1989, p. 538) This research was performed in the course of a master thesis and therefore had to be completed independently.

A further challenge that we want to mention is the leveling problem when collecting empirical data to analyze an organizational or institutional topic. Interviewees are representative of the institutional players participating in the ecosystem. Naturally, all individuals hold their own views that might differ from company policies and goals. Consequently, there is a leveling problem, because the research focuses not the personal level (e.g. what are the personal motivational factors to contribute to the consortium), but on the institutional level. (Liu et al., 2013) During interviews, the right balance between posing questions targeted at the individual level that give deeper and more precise insights and the ones targeted at the institutional level had to be found. This factor was also kept in mind during analysis to ensure correct interpretation of the data.

The only type of data source used was interview data. Other possible data sources such as published documents and information on the website was examined and read to gain insight into the officially communicated information before conducting and analyzing the interviews. After revising these data sources we decided not to include these sources into the coding phase. The reasoning behind this is to avoid weakening empirical insights from interview data by including outdated information. The actual coding therefore only included one type of data source: interview transcriptions. The goal to triangulate evidence (Eisenhardt, 1989) to achieve a stronger base for the theory Jick (1979) was only met via the variety of interview partners and not via employing different types of data sources.

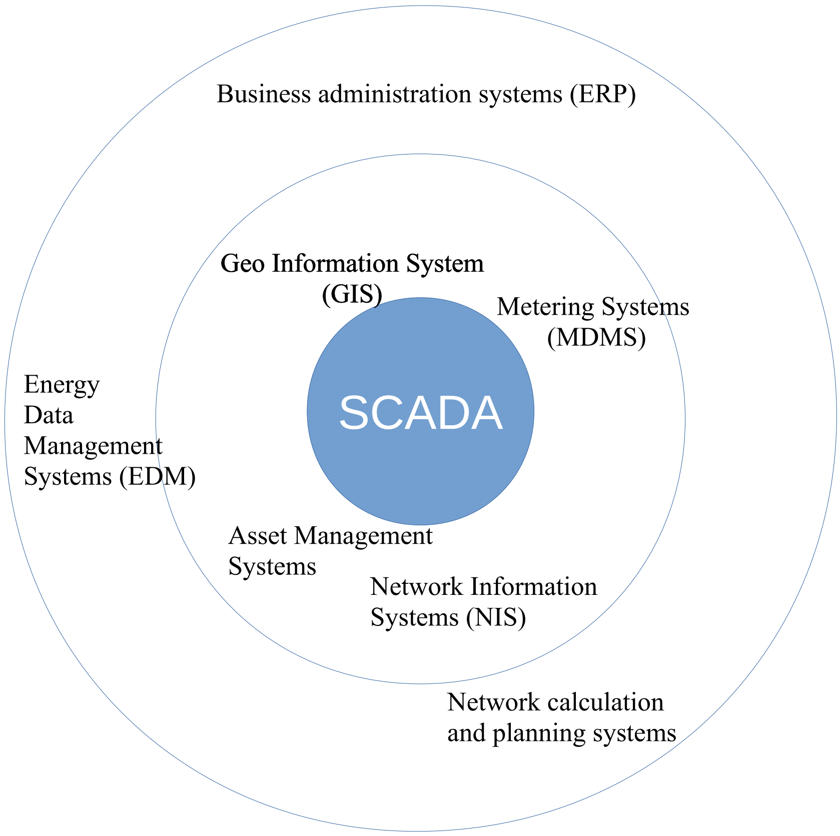
3.4 Suggestion for a Software Ecosystem Representation

To further define the proposition “close to the SCADA system”, we used an IT system landscape diagram (cf. graph 2) provided by Westnetz (Westnetz, 2015, p.2) to locate the epicenter of oK activity in the software ecosystem.



Graph 5: IT landscape illustration by Westnetz (2015)

The SCADA system builds the epicenter of oK activity. Nevertheless, all interviewed oK members emphasized that the SCADA software itself is not part of the short- or mid-term scope of the consortium. SCADA systems grew historically and were developed over a long period of time. These systems are critical and their reliability and availability requirements are extremely high. With the help of a simple network analysis, another illustration of the systems (see graph 6) was created.



Graph 6: Suggested graphical representation of the software ecosystem

It is to note that the basis for this illustration was not discussed with a domain expert. In the case of ERP, the illustration seems to serve its intended purpose well: ERP systems are further away from the epicenter than other systems. This aspect is illustrated correctly, yet we are not confident that this is the case for the other represented systems, too. The whole graph is very coarse grained and might not reflect reality in any way. The sole purpose of this illustration is to suggest a possibility to describe the software ecosystem around oK graphically to enhance future discussions. To arrive at a truly useful illustration of the ecosystem, detailed and specific input from domain experts is essential.

3.5 Further Challenges in the Software Ecosystem

3.5.1 Shortage of Development Resources

An additional cause for conflict is a shortage of human resources, especially developers. DSOs are willing to pay for software, but they want high quality software and good service. Software suppliers want to serve a lot of customers, but often lack the development resources to deliver appropriately.

Two interview partners addressed the issue a shortage of qualified software developers. It was mentioned that employee movement between the different players in the software ecosystem occurs.

A shortage in human resources causes problems in reaching economic goals.

3.5.2 Company Internal Challenges

Next to ecosystem challenges, companies face internal challenges.

The two most pressing internal challenges oK members have to overcome to successfully engage in the consortium are:

- clearly defining responsibilities and collaboration with internal IT departments
- ensuring management support

Several interview partners mentioned that decision making of top management is dependent on the manager's previous knowledge and perspective on business. Leaders who have gained knowledge in the open source community before or those who take a rather technical stance are reported to be more fond of the oK consortium. Strictly business-oriented managers tend to dismiss the value of engaging in the consortium.

3.6 Addressing Further Challenges

3.6.1 Addressing Developer Shortage

The shortage of development resources is a problem for software suppliers and the software ecosystem as a whole.

One service provider explained that their software planning process was reworked completely as a reaction to the consortium's criticism concerning feature delivery times. This ensures a better overview and helps to serve the DSOs better. This can be seen as a positive effect of oK.

Apart from planning optimization, the shortage of resources is a long-term problem that cannot be resolved quickly.

On the one hand, the software supplier, which is not part of oK, stated that outsourcing coding work is a way they use to handle their internal developer shortage. On the other hand, this service provider was very strict about not integrating third party modules. Main arguments against doing so where the warranties that are usually given in the service level agreements. Integrating third party modules would lead to not knowing the code so well as if it was developed in-house.

This argumentation seems to forget that open source software is basically just as accessible as code that was developed by an outsourcing company. Code that was developed somewhere else needs to be well understood and the quality ensured. This is done with outsourced code development and could be done with open source modules developed by a different service provider and paid by the consortium just as well. And the service provider would not have to finance the development of such a module itself.

Software modularity and open source modules can deliver relaxation to the developer shortage of the software ecosystem.

3.6.2 Addressing Internal Challenges

Categorizing and evaluating company internal challenges is a task that goes far beyond the scope of this thesis. Each company has very specific characteristics concerning company culture, organizational hierarchy and history. Moreover, the insights gained through empirical data collected do not go far enough and are limited to one interview partner per company. The following two points are shortly discussed because they came up repeatedly and potentially offer interesting links for further research.

Both interviewed DSO representatives named the internal IT department as an important player and stakeholder concerning oK. Additionally, other cases of IT departments having a positive or negative attitude towards oK were discussed.

One DSO struggles with internal responsibility claims of the IT department. The IT department operates as a cost center and does not have the financial means, nor the man power to engage fruitfully in oK. Nevertheless, there are claims and aspirations to take over the software questions from grid management to regain sovereignty over the IT systems of the whole corporation. A similar situation has happened before and led to a DSO to leave oK.

In another case, the grid management employees who are engaged in oK, would like to receive more technical support and expertise from the internal IT department, but resources are scarce. There is even a case in which the IT department wants to be part of oK, but management support is low.

All three internal challenges with IT can be settled more easily if responsibilities are clearly communicated and enforced by management. Eventually, the engagement in oK is a project for a company and the decision-making power lies in the hands of management. As in all IT projects, top management support is a crucial success factor and top managers have to recognize the fact that they personally have the most influence on project success or failure. (Young & Poon, 2013, p. 955)

Concluding the points discussed, the internal decision-making process is dependent on several factors:

- Is there someone with very high personal commitment who promotes and fosters the oK idea internally?
- What is the top management perspective on oK?
- What is the standing of the IT department towards oK? Do they support it?
- Are responsibilities towards the IT systems clearly communicated and enforced?

Company representatives who actively engage in oK should regularly check, if the project is still supported by top management and if responsibilities are clear. If detected early on, internal conflicts can be addressed more easily.

Appendix A List of Abbreviations

CIM	Common Information Model
DSO	Distribution System Operator
ERP	Enterprise Resource Planning
IoT	Internet of Things
KPI	Key Performance Indicator
OK	openKONSEQUENZ
R&D	Research & Development
SMB	Small and Medium sized Businesses
SCADA	Supervisory Control and Data Acquisition

Appendix B Coding Results

The coding steps described in chapter 2.4 yielded the following results which were used as basis for theory building. The six main topics were:

- Business Model
- oK Community
- Market Structure
- Software
- oK Stakeholders
- Open Source

Details on which codes were subsumed under these topics can be found below.

Business Model

Complete solution

Conventional procurement

Fear towards open source

Fearing market changes

New business model

Old business model

Donation of modules to oK

oK Community

Benefit from oK

Collaboration and exchange of ideas

Collaborative investment

Common goals oK

Donation of modules to oK

Engagement for oK

Homogeneity of goals in oK

Cooperation obstacles

Critical mass

Expertise in oK

Investing expertise in oK

Joint reseearch projects

Motivation NOT to participate in oK

Motivation to participate in oK

No short-term profits via oK

Future development of oK

Project volume

Project tendering

Scope of oK
Software architecture
Personal motivation to contribute to oK
Possibility of oK failure
Time invest in oK
Visibility of oK
Working atmosphere in oK
Promotion of oK idea and processes internally & externally

Market Structure
Competition among service providers
Developer shortage
Effects of consortium on ecosystem
Employee movement
High demands from DSOs
Income structure service providers
Innovation in the ecosystem
Long-term profits wanted
Market leadership
Market structure
Networks between DSOs
No specialization possible
Opportunism service providers
Price dictation
Price Fight
Service provider
Software lifecycle
No short-term profits via oK
Donation of modules to oK
Fear towards open soruce
Fearing market changes

Software
Complexity
Ensuring reliability
Ensuring availability
Failed modules
Integral/marginal software
Maintenance & support
Modularity (divide & conquer)

Monolithic approach
No understanding of code
Changing requirements
SCADA
Security requirements
Software quality
Standardization of interface
Value of code
Vendor independent platform
Software lifecycle

oK stakeholders
Decision maker dependency
Internal argumentation (own company)
Internal IT
Planning process
Regulation effects
Top management support
Top management view

Open Source
Establishment of open source needs time
Getting to know open source
Opinion towards open source

References

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

Attride-Stirling, J. (2001). Thematic networks: An analytic tool for qualitative research. *Qualitative Research*, *1*(3), 385–405. https://doi.org/10.1177/146879410100100307

Bakos, Y., & Brynjolfsson, E. (1999). Bundling Information Goods: Pricing, Profits, and Efficiency. *Management science*, *45*(12), 1613–1630. https://doi.org/10.1287/mnsc.45.12.1613

Becker, T. (Ed.). (2007). *Netzwerkmanagement: Mit Kooperation zum Unternehmenserfolg* (2., überarbeitete und erweiterte Auflage). Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-71891-8

Brooks, K. (2016). The numbers don't lie.

Energiewirtschaftsgesetz, Bundesministerium für Wirtschaft und Energie 13.07.2005.

Gesetz zur Digitalisierung der EnergiewendeBundesgesetzblatt Jahrgang 2016 Teil I Nr. 43 2034 (Bundestag 29. August 2016).

Christensen, C. M. (1997). *The innovator's dilemma: When new technologies cause great firms to fail*. *The management of innovation and change series*.

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *The Academy of Management Review*, *14*(4), 532. https://doi.org/10.2307/258557

European Commission. (2003). What is an SME? Retrieved from http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition/

Feldstein, M. (2014). Community Source Is Dead. Retrieved from https://mfeldstein.com/community-source-dead/

Fjeldstad, Ø. D., Snow, C. C., Miles, R. E., & Lettl, C. (2012). The architecture of collaboration. *Strategic Management Journal*, *33*(6), 734–750. https://doi.org/10.1002/smj.1968

Fogel, K., Vasile, J., & Finch, P. (2018). Open Source Archetypes: A Framework For Purposeful Open Source. Retrieved from https://blog.mozilla.org/wp-content/uploads/2018/05/MZOTS_OS_Archetypes_report_ext_scr.pdf

Glaser, B. G., & Strauss, A. L. (1967). The discovery of grounded theory: Strategies for qualitative theory. *New Brunswick: Aldine Transaction*.

Goering, A., Meister, J., Lehnhoff, S., Herdt, P., Jung, M., & Rohr, M. (2017). Reference architecture for open, maintainable and secure software for the operation of energy networks. *CIRED - Open Access Proceedings Journal*, *2017*(1), 1410–1413. https://doi.org/10.1049/oap-cired.2017.0965

Goering, A., Meister, J., Lehnhoff, S., Jung, M., Rohr, M., & Herdt, P. (2016). Architecture and Quality Standards for the Joint Development of Modular Open Source Software for Power Grid Distribution Management Systems, 36–39.

Hanganu, G. (2008). The Community Source Development Model. Retrieved from http://oss-watch.ac.uk/resources/communitysource

Henkel, J., Schöberl, S., & Alexy, O. (2014). The emergence of openness: How and why firms adopt selective revealing in open innovation. *Research Policy*, *43*(5), 879–890.

Hess, C., & Ostrom, E. (2011). *Understanding knowledge as a commons: From theory to practice* (1st MIT Press pbk. ed.). Cambridge, Mass.: MIT Press.

Hill, P. (2014). Kuali For-Profit: Change is an indicator of bigger issues. Retrieved from https://mfeldstein.com/kuali-profit-change-indicator-bigger-issues/

Hippel, E. von. (2001). Innovation by user communities: Learning from open-source software. *MIT Sloan management review*, *42*(4), 82. Retrieved from https://sloanreview.mit.edu/article/innovation-by-user-communities-learning-from-opensource-software/

Johnson, S. (2018). After Almost 4 Years as For-Profit, Kualis Raises $10M from Owl Ventures. Retrieved from https://www.edsurge.com/news/2018-02-13-almost-four-years-after-creating-a-for-profit-kuali-raises-10m-from-owl-ventures

Krause, J., Höfer, H., Braun, M., & Seibt, R. (2008). Collaborative Open Source Application Development (COSAD) Whitepaper. Retrieved from https://openbit.eu/wp-content/uploads/2016/03/COSAD-Whitepaper-DE-2008-06-30.pdf

Liu, M., Hansen, S., & Tu, Q. (2014). The community source approach to software development and the Kuali experience. *Communications of the ACM*, *57*(5), 88–96. https://doi.org/10.1145/2593687

Liu, M., Hull, C. E., & Hung, Y.-T. C. (2013). Antecedents of Community Source Network Formation: The Case of Kuali. In R. H. Sprague (Ed.), *2013 46th Hawaii International Conference on System Sciences (HICSS 2013): Wailea, [Maui], Hawaii, USA, 7 - 10 January 2013 ; [proceedings]* (pp. 4267–4276). Piscataway, NJ: IEEE. https://doi.org/10.1109/HICSS.2013.95

Liu, M., Wang, H. J., & Leon Zhao, J. (2012). Technology flexibility as enabler of robust application development in community source: The case of Kuali and Sakai. *Journal of Systems and Software*, *85*(12), 2921–2928. https://doi.org/10.1016/j.jss.2012.06.026

Liu, M., Wang, H., & Zhao, L. (2007). Achieving flexibility via service-centric community source: The case of Kuali. *AMCIS 2007 Proceedings*, 103.

Liu, M., Zeng, D. D., & Zhao, J. L. (2008). A Cooporative Analysis Framework for Investment Decisions in Community Source Partnerships. *AMCIS 2008 Proceedings*, 278.

Mackie, C. J. (2008). Open source in open education: Promises and challenges. *Opening up education: The collective advancement of education through open technology, open content, and open knowledge*, 119–131.

McCaffrey, M. (2014). Global 100 Software Leaders by revenue. Retrieved from https://www.pwc.com/gx/en/industries/technology/publications/global-100-software-leaders/explore-the-data.html

OpenKonsequenz. (2013). *Machbarkeit Konsortiale Softwareentwicklung auf Basis von Open Source Softwarevon.*

Osterloh, M., & Luethi, R. (2008). Commons without tragedy: das Beispiel Open Source Software.

Osterloh, M., & Rota, S. (2007). Open source software development—Just another case of collective invention? *Research Policy*, *36*(2), 157–171. https://doi.org/10.1016/j.respol.2006.10.004

Parry, M. (2009). Business Software, Built by Colleges for Colleges, Challenges Commercial Giants: Though many are wary, the Kuali Foundation's free program for finance has been adopted by a few institutions. Retrieved from https://www.chronicle.com/article/Business-Software-Built-by/49147

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23–49. https://doi.org/10.1007/s12130-999-1026-0

Requardt, B., Wende - von Berg, Sebastian, Wagner, T., Töbermann, J.-C., & Braun, M. Modular system architecture for processing of CIM.

Riehle, D. (2014, February). *Open Source User Foundations*. Retrieved from https://de.slideshare.net/driehle/open-source-user-foundations

Riehle, D. (2016, July). *Open Source User Foundations.* BITKOM. BITKOM Forum Open Source, Berlin.

Rogat, S. (Sommersemester 2015). *Kapitel 3: Einführung in die Regulierung der Netze*. Wirtschaftlichkeit des Netzbetriebs, Biberach an der Riß.

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

Schaarschmidt, M. (2012). *Firms in open source software development: Managing innovation beyond firm boundaries*. @Koblenz-Landau, Univ., Diss., [2012]. *SpringerLink Bücher*. Wiesbaden: Gabler Verlag. Retrieved from http://dx.doi.org/10.1007/978-3-8349-4143-5

Simon, H. (2018) Verkäufermarkt. Retrieved from https://wirtschaftslexikon.gabler.de/definition/verkaeufermarkt-46956/version-270229

Statistisches Bundesamt. (2015). Anteile kleiner und mittlerer Unternehmen an ausgewählten Merkmalen 2015. Retrieved from https://www.destatis.de/DE/ZahlenFakten/GesamtwirtschaftUmwelt/UnternehmenHandwerk/KleineMittlereUnternehmenMittelstand/Tabellen/Insgesamt.html

Taft, D. K. (2009). Community-Source Development Appeals in Tough Times: A combination of traditional and open-source development models, community source can save companies money and reduce vendor lock-in. Retrieved from http://www.eweek.com/development/community-source-development-appeals-in-tough-times

Techopedia. Vendor Lock-In. Retrieved from https://www.techopedia.com/definition/26802/vendor-lock-in

Weerth, C. (2018). Kooperation. Retrieved from https://wirtschaftslexikon.gabler.de/definition/kooperation-39490/version-262897

West, J., & Gallagher, S. (2006). Challenges of open innovation: The paradox of firm investment in open-source software. *R and D Management*, *36*(3), 319–331. https://doi.org/10.1111/j.1467-9310.2006.00436.x

Westnetz. (2015). *CIM Workshop Herausforderungen im Alltag des Netzbetreibers in Übertragung und Verteilung: Was müsste sich ändern?* .

Wheeler, B. (2004). The Inevitable Unbundling of Software and Support. Retrieved from https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/CMPSTECH/S040301W.pdf

Wheeler, B., & Hilton, J. L. (2012). The marketecture of community.

Wikipedia. Community Source. Retrieved from https://en.wikipedia.org/wiki/Community_source

Yin, R. K. (2010). *Case study research: Design and methods* (4. ed., [Nachdr.]). *Applied social research methods series: Vol. 5*. Los Angeles: SAGE.

Young, R., & Poon, S. (2013). Top management support—almost always necessary and sometimes sufficient for success: Findings from a fuzzy set analysis. *International journal of project management*, *31*(7), 943–957. https://doi.org/10.1016/j.ijproman.2012.11.013