**Term Paper**

Open Source and IP in the Digital Society

WS2021

Anh Truc Lam

Anh-Thu Tran

**Test hallo**

In the Open Source Software (OSS) paradigm, developers along with users form a community for an OSS project as they share an interest in using/developing the project. Active community engagement is essen- tial for an OSS project to succeed. OSS communities should strive for greater community participation and engagement through the use of tools, practices, and processes. The primary goal of this paper is to presents a review of studies on community participation and engagement in OSS projects based on sys- tematic mapping study and snowballing technique. This study also provides an understanding about the research topics and gaps in the area, utilized research methods and publication venues. We have analyzed 67 research papers related to the study topic. The findings revealed that most of the studies used a com- bination of survey and questionnaire as a research methodology. We found that community participation and engagement research focuses on 5 main research topics joining process, contribution barriers, moti- vation, retention, and abandonment. The investigated studies provide more evidence on motivation and contribution barriers but less on the joining process and abandonment. The results presented in this paper will be helpful for researchers to understand the latest trends in this area and identifying the cor- responding research gaps.

Inhaltsverzeichnis

[1 Introduction 2](#_Toc61631833)

[2 Theoretical Foundations 3](#_Toc61631834)

[2.1 Definition of Open Source Software (OSS) Projects 3](#_Toc61631835)

[2.2 Open Source Projects as main driver in companies 4](#_Toc61631836)

[2.3 Open Source Projects in IT-Company Facebook 7](#_Toc61631837)

[2.4 Presented Case Study 8](#_Toc61631838)

[2.4.1 Role of Pytorch in Open Source Projects 8](#_Toc61631839)

[2.4.2 Role of React in Open Source Projects 9](#_Toc61631840)

[3 Methodology 12](#_Toc61631841)

[3.1 Research Design 13](#_Toc61631842)

[3.1.1 Defining the Research Question 13](#_Toc61631843)

[3.1.2 Defining the Case Study 14](#_Toc61631844)

[3.2 Literature Search Strategy 15](#_Toc61631845)

[3.3 Analysis 17](#_Toc61631846)

[3.3.1 Literature Analysis 17](#_Toc61631847)

[3.3.2 Case Study Analysis 18](#_Toc61631848)

[3.4 Model of Commercial Objectives 19](#_Toc61631849)

[4 Results 21](#_Toc61631850)

[4.1 Findings 21](#_Toc61631851)

[4.1.1 Overview of Reasons for Companies to Invest into FLOSS Projects 21](#_Toc61631852)

[4.1.2 OSS Project Trends 25](#_Toc61631853)

[4.2 Discussion 26](#_Toc61631854)

[4.3 Conclusion 27](#_Toc61631855)

# **Introduction**

In the age of digitization, open source software (OSS) become increasingly important to companies commercial objectives, which target the question of how to the align OSS projects with commercial practices (Zhou et al., 2016, p. 3). With the increasing number of participations, capabilities and other ressources, open source projects become tremendously complex and interrelatable as it demands more profressionality and management in OSS development (Germonprez et al., 2018, p. 2). Currently, OSS projects face two contradictory values and approaches. On the one hand, OSS is still building on the “[…] traditions and ideals of the free software movement” and egalitarian perspectives (Germonprez et al., 2018, p. 14). But on the other hand, there is a urgent need of professionality of corporate participation, their long-term investment and support, which is strategically important component in companies supply chain (Germonprez et al., 2018, p. 14). This has the reason because companies aim to reduce costs, seek the potential to receive external innovation and raise their profit by participating in open source communities (Zhou et al., 2016, p. 2). The best-known example is the Linux, Apache, Wordpress and its open source project ecosystems, which has an significant role in the evolution of open source projects (Gastil & Davies, 2020, p. 4). However, the current situation reveals many risks as companies are confronting with various challenges to enter, maintain and support OSS projects. One of the risks are the financial challenges to sustain such projects, which raise questions of how to invest into long-term projects as it is critically important to maintain, for example, contributors, who does not belong to a company, and prevent them to leave the OSS project by chance (Germonprez et al., 2018, p. 14). Additionally, the companies participation is fraught with commercial control, which could deprive the communities rights from their code commit privilege (Zhou et al., 2016, p. 2). Several tech-companies like Microsoft, Google and Facebook have already recognized these opportunities and took participation into the open source community in order to develop and provide their own software such as Microsoft’s ASP.NET framework and Facebook’s OSS react JavaScript library (Kochhar et al., 2019, p. 1). To investigate the reasons for companies to invest into OSS projects, we took the company Facebook and its OSS projects in our investigation. This term paper proceed with the following structure. In section 2, we present definition and concepts of OSS projects. In section 3, we describe our methodology and analysis. After that, we present the results in section 4 and discuss our result to come to a conclusion in section 5.

# Theoretical Foundations

This section provides a theoretical foundation of important definition and description of peer-produced goods, OSS and Open Source Projects with regard on IT-companies. Further, general frameworks, specific companies and technologies related to OSS projects will be presented.

## Definition of Free/Libre Open Source Software (FLOSS) Projects

In the beginning, all FLOSS was simply coined as *Free Software* (FS). 1998 the term *Open Source Software* (OSS) was introduced in order to help users clarify, that its software products are not necessarily provided for free, but its source code is freely accessible. (Ballhausen, 2019, p. 82) The terms FS and OSS share the same license terms, which are also referred to as the four freedoms. Thus, in order for software to be licensed as FLOSS it needs to meet four criteria. The first grants the user the freedom to run the program for any purposes the user deems fit. The second freedom allows the user to lookup any source code of the program and adapt its implementation to his needs. Users also should have the freedom to reallocate copies of FLOSS in order to assist others. Lastly, any user wields the freedom to allow access to copies of improved code in order to create benefit to the general public. (Logothetis & Stylianidis, 2016, p. 30) As software itself is protected under copyright law regardless, if a software is licensed as FLOSS, the four licenses outlined above simply define the distribution terms for FLOSS. (Ballhausen, 2019, p. 83) It remains to mention, that it is essential for companies to set up proper FLOSS governance mechanism in order to comply with the licensing requirements and to reduce licensing risk. These help to stay consistent with operational strategy by considering participants’ background and motivation as well as align stakeholder’s interests. (Kemp, 2010, p. 1) The terms FS, OSS and FLOSS can be used interchangeably.

## Open Source Projects as main driver in companies

In various papers, we find associations of the term open source projects in companies with the term corporate investment, which emphasizes how rapid the changes of open source itself through companies investment was (Germonprez et al., 2018, p. 2). Generally, there exists two types of commercial involvement of a company in OSS (Wagstrom, 2009, p. 8). A firm which is focused in building an active community in order to profit on its related services is called *community-focused*. If a firm is relying on product sales it is called *product-focused*. With these underlying findings, Zhou et al., 2016 (pp. 7–8) developed the model “Dimensions of the community involvement models” to further analyze commercial involvement. As a company firstly needs to verify the common objective of the project and the firm’s strategy as well as assessing the opportunity to create future revenue. Zhou separates this endeavour into the dimension *commercial objective*. Next, in order to work within the project community framework companies need to create the underlying environment for the community to engage. For instance, they need to agree on intellectual property (IP) and legal issues, how to allocate resources, and how to integrate their employees into the community (Zhou et al., 2016, pp. 8–9). This aspect is picked up as community involvement action dimension. (Figure 1)



Figure 1. Dimensions of the community evolvement models (Zhou et al., 2016, p. 8)

The authors identified and assigned serveral concepts to each of the two dimensions, highlighted in yellow by going through academic literature. Zhou then illuminates each concept coloured in green by setting up seven questions, which tries to answer what fuels the companies involvement and how the companies try to reach their goals within the OSS project, coloured in pink (Zhou et al., 2016, pp. 8–9).

Zhou et al. (2016, p. 7) investigated the purpose for a company to invest into OSS project and summarized the five key findings in the dimension *Commercial Objectives*:

|  |  |
| --- | --- |
| **Concepts from Literature** | **Description**  **(Zhou et al., 2016, pp. 7–8)** |
| 1. Software Quality | Companies utilize external knowledge involvement in order to increase quality of OSS projects. |
| 2. Outside Technical Support | Companies present their code base via OSS to raise outside feedback and technical support. They take advantage of creative ideas and capabilites of external contributors. |
| 3. Business Opportunity | Companies pursue a strategic and long-term goal by keeping free access to their open source platform and cultivate business opportunities, such as middle ware software or related services. |
| 4. Business Modell | Companies nurture the open source community by strongly engaging with it. By aligning their business model with the OSS community or supporting community engagement through own employees, companies legitimize themselves to take advantage of newly developed products and services. |
| 5. External Innovation | Companies with fewer resources can benefit by OSS initiatives of larger institutions. They can harness and rely on software quality and continued improvement. |

Table 2‑1 Concepts from Literature - Description

From this model we assessed, that the commercial objectives dimension largely contains answers, which are covering our research question, such that we recycle the concepts for further analysis, namely Software Quality, Outside Technical Support, Business Opporuntity, Business Model and External Innovation (Zhou et al., 2016, p. 8). A description of each concept can be retrieved from Table 1 above. Here the first question is: Is the OSS product critically important for the company’s business? (Q1).

Critically important can be understood in two perspectives, which is “[…] one, gaining profit directly from the OSS product; two, the OSS product is strongly associated with (or greatly helps to gain) profit.” (Zhou et al., 2016, pp. 7–8). We deem the remaining concepts stemming from the second dimension unfit to answer our research question and have excluded this part of Zhou’s model in our paper. It remains to mention, that Zhou’s paper was published within the past five years, during the creation of our paper, such that we can rely on its currentness and ultimately ensure the quality of our conclusions.

## Open Source Projects in IT-Company Facebook

In the past two decades OSS development proved themselves as it has produced a large number of highly reliable projects, such as Linus Mozilla browser, MySQL database and the Hadoop framework (Author test, year, p. ). Thus, the OSS framework remains an attractive model to build and deploy software. An important tool for developers to contribute to OSS projects is Git (Author, year, p. ). Git is closely following different versions of a software and is also described version control system (VCS). Its intrinsic features are the faciliation of distributed development and the capability to handle over thousand developers. Hence, social coding websites such as GitHub or Gitlab offering Git are enabling over 40 million users to collaborate and partake in OSS projects. Further, even large software companies, such as Google and Microsoft are using GIT as their main development platform and started to open source part of their proprietary software as well (Author, year, p. ).

Observing the open source landscape one notices, that Facebook, a social networking service, is involved in OSS projects as well (Author, year, p. ). This company provides a social networking site, also named Facebook, which enables users to connect and share information easily with family and friends. Currently Facebook is the world’s largest social network, with more than 1billion users worldwide (Author, year, p. ). The company engages within numerous projects covering various large areas ranging from Artificial Intelligence, Web Technology and Operating Systems to Security etc. (Author, year, p. ). A member of the company’s open source team shares, that Facebook move towards open source comes naturally as in its main mission the company tries to create a world community (Author, year, p. ). Second to that the firm tries to foster innovation and create better software, as Facebook is facing unique development challenges it has to solve. Here Open source serves the company producing better software and work more transparently.

**Umschreiben:**

Ist Facebook commuity-focused oder product-focused?

An in-depth analysis of commercial involvement in Gnome and Eclipse [Wagstrom et al. 2010] identified two types of commercial involvement. RedHat was an example of a community-focused company building a vibrant Gnome community and monetizing services. The other type was of product-focused firms that rely on product revenues. We build on this work to define the dimensions of commercial involvement (why and how), as described in Section 3.3. 🡪 (Zhou et al., 2016, p. 3)

# diese Definition hier in 2.3 packen

Warum haben wir das hier genauer bzw anders beschrieben

|  |  |
| --- | --- |
| **Concepts from Literature** | **Description** |
| 1. Software Quality | is based on how often the version of a software comes with improvements that impacts the user and the gap period between the previous version date. We measure the metric “Rate of delivery”. This indicates that the higher rates of software version delivery is the better quality software for customers (AltexSoft, 2017). |
| 2. Outside Technical Support | This means if contributors, who are not employed in the same company as where the software was developed, has contributed to this project. For our purpose we analyzed the top 10 contributors on GitHub. |
| 3. Business Opportunity | described if the software is a product or service from the company and is used to set a strategic and long-term goal as a business opportunity. We follow the question: “*Does the open source software save costs or create revenue?”* |
| 4. Business Modell | This means if a software was developed by promoting proactive engagement with open source communities to develop this software. We follow the question: “Was this software created to solve a business model problem? For example expanding ressources. |
| 5. External Innovation | describes if the software has impacted smaller companies which benefit from the innovation of the software |

**Umschreiben oder löschen:**

The three projects use SVN for version control. We ob- tained the commit history from SVN repositories. The commit history contained a set of tuples consisting of the time, the developer (indicated by the login), the revision number, the comment, and the list of modified files, as shown in Table III. 🡪 (Zhou et al., 2016, pp. 10–11)

**Identifying External Developers.** To investigate the inflow and retention of volun- teers, we need to identify whether a developer was a volunteer when he or she started contributing code. We used email domains to identify developers of the corresponding commercial companies (internal developers). We consider the remaining developers to be external. 🡪 (Zhou et al., 2016, p. 11)

**Identifying Roles of External Developers.** We take a further step to understand vol- unteer inflow: we looked into different developer groups (roles). In particular, we sep- arated them into application developers who write code that runs on the application server and infrastructure developers who write the application server itself, because the commercial involvement may have different effects on different types of external developers. JBossAS has the largest of the three communities with a sufficient number of developers in each role for us to be able to observe the differences. Getting involved in an OSS project often starts from the mailing list [von Krogh et al. 2003]. The mail archives may, therefore, suggest the senders’ background (e.g., the domain name in the email address or the signature). Personal details, such as past jobs, could be found on social network websites (e.g., LinkedIn, Facebook). We used these rich information sources to identify the roles of the external developers, as described in Online Appendix B. 🡪 (Zhou et al., 2016, pp. 11–12)

## Presented Case Study

To understand the research question with greater clarity, we want to illuminate reasons why Facebook is engaging in OSS projects. Here we conduct a case study on two open source projects provided by Facebook. Naturally a case study includes an interview to arrive at qualitative answers to our problem (Author, year, p. ). However, due to limited resources, we have to rely on other reliable sources, such as literature review and Github repository data. We put our main focus on the Facebook’s open source projects: Pytorch and React (Author, year, p. ).

### Role of Pytorch in Open Source Projects

The Pytorch project released in 2016 is contributing in the field of Artificial Intelligence (AI) by providing a Python library for deep learning (Author, year, p. ). It gained popularity in the research community as its performance is similar to other deep learning libraries, although by making use of the widely adopted computer language Python familiar to many Data scientists due to its simplicity (Author, year, p. ). The objective is to put researchers first by aiding them in reducing difficulty in creating new models, handling and wrenching data (Author, year, p. ).

**Umschreiben:**

* Blogbeitrag: Announcing PyTorch 1.0 for both research and production

<https://developers.facebook.com/blog/post/2018/05/02/announcing-pytorch-1.0-for-research-production/>

The path for taking AI development from research to production has historically involved multiple steps and tools, making it time-intensive and complicated to test new approaches, deploy them, and iterate to improve accuracy and performance. To help accelerate and optimize this process, we're introducing PyTorch 1.0, the next version of our open source AI framework. PyTorch 1.0 takes the modular, production-oriented capabilities from Caffe2 and ONNX and combines them with PyTorch's existing flexible, research-focused design to provide a fast, seamless path from research prototyping to production deployment for a broad range of AI projects. With PyTorch 1.0, AI developers can both experiment rapidly and optimize performance through a hybrid front end that seamlessly transitions between imperative and declarative execution modes. The technology in PyTorch 1.0 has already powered many Facebook products and services at scale, including performing 6 billion text translations per day. PyTorch 1.0 will be available in beta within the next few months, and will include a family of tools, libraries, pre-trained models, and datasets for each stage of development, enabling the community to quickly create and deploy new AI innovations at scale. 🡪 (Jia, 2018, p. 1)

PyTorch's imperative front end allows for more rapid prototyping and experimentation through its flexible and productive programming model. The first version of PyTorch launched a little over a year ago, and its speed, productivity, and ability to support cutting-edge AI models such as dynamic graphs quickly made it a popular and important development tool for AI researchers. It has more than 1.1 million downloads and is the second-most cited deep learning framework on arxiv over the last month. For example, UC Berkeley computer scientists put PyTorch's dynamic graph capabilities to use for their noteworthy CycleGAN image-to-image transform work. 🡪 (Jia, 2018, p. 1)

* Tutorials are provided

<https://developers.facebook.com/blog/post/2020/10/07/pytorch-tutorials-refresh-behind-the-scenes/>

* Pytorch interagiert auch mit anderen Partnern und Unternehmen wie Amazon

<https://developers.facebook.com/blog/?filters%5B0%5D=artificial_intelligence&filters%5B1%5D=pytorch>

* Auch mit Tesla

https://en.wikipedia.org/wiki/PyTorch

### Role of React in Open Source Projects

The second technology, React, is another open source library of Facebook, which utilizes a JavaScript framework to help reduce complexity in developing user interfaces in the web development field (Author, year, p. ). This framework introduces an innovative take in viewing interchangable content, such as advertisements, on websites, which were initially not fully welcomed by the community as it went against the best practises of JavaScript itself (Author, year, p. ). Facebook faced technological issues with the general workflows and best practises in web development at that time (Author, year, p. ). Although the problems were not unique to Facebook, it was them who took the charge and provided a solution, which not only benefitted themselves but others as well. React was thus created for a single reason of solving how a website is dealing with displaying data (Author, year, p. ).

# Dieser Abschnitt könnte eig in 3.1.2 Defining the Case rein

We have picked these two projects as they at first seemingly share the common objective to help developers improving on their capability to operate efficiently in their respective fields. The nature of each project is generally to minimize the complexity in development. This already provides social motivation to developers to engage in these OSS projects to gain reputation and help the community learn. However, the projects mentioned are living within the ecosystem of Facebook and are not out of scope of their business acumen, thus hinting to further relations this company has with these OSS project, which lastly leads to our research question we try to answer in this term paper.

Umschreiben:

* Release jahr: React wurde in 2013 entwickelt (Verlinkung auf Wiki!)

<https://en.wikipedia.org/wiki/React_(web_framework)>

O P E N source software systems have seen significant growth over the past few years. Open source soft- ware (OSS) development follows a somewhat different way of building and deploying software systems ranging from small to very large scale, with contributors spread out in different parts of the world. Over the past two decades or so, OSS development has produced several high- quality projects such as the Linux operating system, Mozilla browser, MySQL database system, and Hadoop framework. Social coding websites such as GitHub, which has over 26 million users and more than 74 million repositories, provides a new way for developers to collaborate. GitHub’s user-friendly interface and its wide adoption by millions of users has attracted even large software organizations to adopt it as their development platform. Recently, large software companies like Microsoft, Google, and Facebook have joined the open source commu- nity and have open sourced some of their proprietary soft- ware such as Microsoft’s ASP.NET framework, Facebook’s react JavaScript library [1], and Google’s Android platform. 🡪 (Kochhar et al., 2019, p. 1)

**Beispiel von Zhou (2016)**

The three projects we investigated, JBossAS,3 Apache Geronimo,4 and JOnAS,5 are open-source application servers conforming to the standard specifications of JavaEE. JBossAS and JOnAS started in October 1999, while Geronimo started in August 2003. At the time of study (September 2010), JBossAS was hosted by RedHat and was the most popular open-source application server. Geronimo was heavily sup- ported by IBM. JOnAS was nursed in the OW2 Consortium, which was established by the collaboration of several organizations. Both JBossAS and JOnAS use the LGPL as their open-source license, while Geronimo uses the Apache License.

The same JavaEE specification and similar development periods control for im- portant confounding factors and make it more likely that the observed variation in outcomes would be a result of the particular style of commercial involvement.

We chose to control for three main factors: technology, environment, and project context. To control for technology and environment, we selected three products imple- menting the same specification over a similar time period. This ensured some level of control over external factors such as application domain and the changes in the technology landscape and world economy.

To control for project context, we compared several epochs of the same project with each epoch representing its unique mixture of company policies and actions. This allowed us to observe the effect of changes in such policies and actions within a single project context. We discovered nine distinct epochs of relatively stable community strategies for the three projects. This setup allowed a comparison of different epochs within a project and a comparison of similar epochs among the projects. 🡪 (Zhou et al., 2016, p. 5)

**GitHub: Pros & Cons**

All the projects we investigated in this study were moved and are now hosted on GitHub. GitHub hosts millions of projects and provides various features such as forking a repository, a push-pull mechanism and an issue tracking system. From the developer interviews, we found that there are some GitHub characteristics which developers like, whereas some developers preferred using the internal systems. We discuss these aspects below. Table 5(e) shows the hypotheses related to GitHub pros and cons. Some of these were mentioned in the previous studies [18], [27]. GitHub is known to and accessible by a large community of developers with expertise in different areas. Such an environment increases the likelihood that a hosted project may be successful. GitHub also provides social transparency which promotes increased awareness [24] and makes it easier for teams to find developers with relevant experience as well as help developers in finding projects they would like to contribute to. Teams can get faster feedback which can help them tackle situations such as solving bugs faster. One developer mentioned, “GitHub gives a tighter feedback loop with technically minded individuals in the community” (D2) and was agreed by 80% of the respondents (S33). GitHub has its own integrated issue tracking system. It provides features such as labels for issues, prioritized issues, voting, and closing issues from commit messages. The projects we studied were using TFS to manage bugs be- fore open-sourcing. Developers reported that GitHub issue tracking is not as powerful as the one in TFS – especially when there is a large number of bugs – and prioritization is easier in TFS. A developer mentioned: “Bug management is not as good as we used to. What GitHub provides issue tracking/bug management is different and so far doesn’t seem to be as powerful. It’s good for everyday. . . . But when you have to do triage, you have to sort through hundreds of bugs and figure out what action needs to be taken and who will address them and when and how ... the tools are not as powerful” (D1). 50% of the respondents agreed that TFS item tracking is better (S34) and similar views were expressed by professional developers using GitHub for commercial projects [12]. GitHub’s pull request mechanism lets external develop- ers contribute to the project by pushing changes to a repos- itory. The project team can review the changes submitted, give suggestions and push code to the repository. 🡪 (Kochhar et al., 2019, p. 10)

# **Methodology**

In this section, we provide an overview about our steps of investigation in the research design. We explain our research question and our case study choice. Additionally, we describe our literature search strategy, which included a selection process. Based on that, we explain how we analyze the data and how we develop our model.

## Research Design

Our main goal is to provide reasons why companies invest into FLOSS projects, which leads to several challenges on investigations in this term paper. First, there is a broad range of intensive knowledge with regard on software projects, which makes it difficult to focus on a specific scope of technologies and OSS projects (Author, year, p. ). Further, as we decided to focus on two OSS projects as described in chapter 2.4, there are no possibilities to validate our theories with interviews based on experts opinion of the company Facebook due to a lack of time and human ressources.

We decided on a data-driven approach, which included scientific literatures and OSS projects on Facebook’s GitHub. Our investigation is summarized in the following steps:

1. We defined the main research questions (Q1) and divided it in two sub-questions indicated in section 3.1.1, which gave us a clearer focus on our research direction.

2. We chosed OSS projects of the IT company Facebook as our case study subject. We made our first screen on their current projects and selected two projects (3.1.2).

3. Then, we conducted a systematic literature review to build a theoretical foundation of current investigations of related fields as explained in section 3.2.

4. Based on theories found, we derived an overview of reasons why companies invest into FLOSS projects based on the model of Zhou et al. (2016, p. 8) provided in 4.1.1.

# Vorgehensmodell updaten



Figure 1 - Research Design Steps

### Defining the Research Question

At the beginning of the course lecture, we defined our main research question:

*“Which reasons does the IT-company have to contribute to FLOSS projects?” (Q1)*.

Due to a lack of resources we need to limit our scope of research on this. Therefore, we specify our investigation and divide Q1 into two sub-questions in the following: *“Is the OSS product criticially important for the company’s business?” (Q2)* and *“Which FLOSS-Trends can be identified in Facebook?” (Q3)*.

We provide an overview about the reasons to invest into FLOSS in companies perspectives based on our literature review in 4.1.1 as well as presenting the latest FLOSS-trends of OSS projects in section 4.1.2.

### Defining the Case Study

To get deeper insights into the reasons why companies invests into FLOSS projects we choose to conduct a case study based on Facebook’s OSS projects. We focus on two main IT technologies: AI und web technologies. However, these two technologies have a broad research field, thus we put the focus on two specific cases. By limiting to two OSS projects of Facebook, we choose an AI project *PyTorch* and *React* in the field of frontend web technologies (Facebook, 2020) as described in section 2.5. In this case study, we collected data of project repositories, programming language and contribution histories on GitHub in this section. We developed a model based on scientific literature and applied this case study on our theories found to validate the model as described in section 3.4.

Often, a case study includes a qualitative research method, for e.g. interviews (Autor), however, due to our limited time frame we choose to apply this case study on our developed model based on Zhou et. al (p. 8, 2016) the model “Dimensions of the community involvement models” as we considered it as a good construct to answer our research question. We report the results in section 4.2.

In the Table 3‑1, we collected data on OSS project React as well as Project PyTorch in Table 3‑2. Occupation of contributors were not listed in the table of GitHub statistics. But we identified the user name as well as the formal name of the top 10 contributors automatically listed on the statistics extracted from the OSS projects on GitHub insights (Quelle). React’s top 10 contributors make up half of the commitments (6893 commitments, 50%). Even though there are 5.259.696 users and 1.527 contributors, only a few of people contributes a significant amount of commits on the React OSS project (Quelle GitHub).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Start Year** | **Languages** | **%** | **Users** | **Contributors** | **Commits on Master Branch** | **Top 10 Contributor and Commits** |
| 2013 | JavaScript | 95,20% | 5.250.696 | 1527 | 13.775 | 6893  (50%) |
| HTML | 2,00% |
| CSS | 1,20% |
| C++ | 0,80% |
| TypeScript | 0,30% |
| CoffeeScript | 0,30% |
| Other | 0,20% |

Table 3‑1 - React OSS Project on GitHub (2020)

In comparison to React, the OSS project Pytorch has 5,1 million less users, but Pytorch has more than the double of commits and similar but less count of contributors. Pytorch has approximately hundred contributor more than React, even though the Pytorch project started 4 years later than React. Ein Diagramm Vergleich hier wäre nice!

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Start Year** | **Languages** | **%** | **Users** | **Contributors** | **Commits on Master Branch** | **Top 10 Contributor and**  **Commits** |
| 2016 | C++ | 52,80% | 53.373 | 1.699 | 32.371 | 7748  (24%) |
| Python | 33,70% |
| Cuda | 6,00% |
| C | 3,90% |
| Cmake | 1,30% |
| Objective-C++ | 0,60% |
| Other | 1,70% |

Table 3‑2 - Pytorch OSS Project on GitHub (2020)

## Literature Search Strategy

We conducted the systematic literature based on the guideline of …(Autor). We summarized our steps in Figure 3. First, we identified digital libraries ACM Digital Library, IEEE Digital Library and Springer Link and evaluate them as situable for our literature review because they possess a broad repository for computer science research (Kaur et al., 2020, p. 4). We took ScienceDirect as an additional digital library. We build the key word search string based on the first literature screen: *(“FLOSS” OR “Open Source Software”) AND ("company" OR "enterprise") AND "facebook"*. We applied this search string in ScienceDirect, ACM Digital Library and IEEE Digital Library. For the digital database SpingerLink, we made a change on the symbols in the search string due to their websites capability to read the search string properly, but the key words stayed similar: *Facebook AND FLOSS AND "Open Source Software" AND (Company OR Enterprise).* By varying different key words, for example by adding “firm”, “industry” and “project” the results often includes non-relevant papers in our research field. The previous presented search strings turned out to be suitable because of the scope of literature and its content we collected. We found 1002 papers. Second, we screened the paper by applying predefined exclusion and inclusion criteria. We restricted the search for papers to peer-reviewed, open access and free access, journals, English language and from the year 2015 to 2020. According to Kaur et al. (2020, p. 4), it is important to filter paper with the focus on the field of Computer Science to gain a better result. Lastly, by reading the title and abstract we checked for the egibility of those papers, which left us 15 papers. After we search for other paper manually, we included one additional paper to our collection. Finally, we found 15 papers in total.



Figure 3 - Selection Process

## Analysis

### Literature Analysis

- Einleitung (1 Page)

- Beschreibung der Analysevorgehen

- Zusammenfassung und Analyse der Literaturen (Konzepte und einordnen ins Modell)

- Erkenntnisse aus diesen Daten kommen in 4.1.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Literature Review**  **Key word search string:** (“FLOSS” OR “Open Source Software”) AND ("company" OR "enterprise") AND "facebook" | | | | |
| **1. Identification** | **ScienceDirect**  n=605 | **ACM**  n=331 | **IEEE**  n = 22 | **Springer**  n = 44  'Facebook AND FLOSS AND "Open Source Software" AND (Company OR Enterprise)' |
| **2. Screening** |  |  |  |  |
| 2.1 Peer-Reviewed | n=25 | Only journals  n=35 | / | / |
| 2.2 Year:  2015-2020 | n=21 | n=28 | n = 11 | n = 28 |
| 2.3 Language:  English | / | / | / | n = 26 |
| 2.4 Full text or Open Access | n=7 | Media format: PDF  n=28 | / | N=20 |
| 2.5 Subject Area: Computer Science | n=4 | / | / | N=11 |
| **3. Eligibility** | | | | |
| 3.1 Titel and abstract reviewed | n=2 | n=3 | n=4 | n=5 |
| 3.2 Inclusion criteria | / | / | / | / |
| **4. Additional Papers Included** | n=0 | n=0 | n=0 | n=0 |
| **Papers in total** | n=2 | n=3 | n=4 | n=5 |

Table 3‑3 - Literature Review Overview

### Case Study Analysis

2.) Beschreibung wie wir mit den case study Zahlen aus GitHub umgehen, zB statische Berechnung oder iwas einfacheres?

# Irgendwo sollte hier eine leere GitHub Such-Übersichtstabelle auftauchen, zB welche Daten wir uns angeschaut haben zB Commitments, Jahre, Mitgliederzahl, Häufigkeit.

# Sollen wir hier statische Regression der Projekt Beteiligung zeigen?

(4) We collected project repository data and analyzed the affiliation history of project participants, as described in Section 3.5.

(5) We used generalized linear models and survival analysis to quantify how differ- ent types of commercial involvement impact contributor inflow and retention and report the results in Section 4.2. 🡪 (Zhou et al., 2016, p. 4)

We use project repository data to quantify two key variables that may be affected by a company’s involvement: the inflow of new contributors and the retention of existing developers. In Section 3.5.1, we retrieve and process historical data from VCSs. In Section 3.5.2, we identify internal and external developers. In Section 3.5.3, we identify different developer roles. 🡪 (Zhou et al., 2016, p. 10)

🡪 so könnte man ungefähr die Daten in GitHub analysieren, aber wir wollen ja schauen wie wir die Fragen beantworten können, zB schauen ob Angestellte mehr Commitments machen als Freelancer

## Model of Commercial Objectives

The model “Dimensions of the community involvement models” developed by Zhou et al., 2016 (pp. 7–8) as described in chapter 2, lays our foundation to analysis and understand our literature and case study more in detail. We limit the model to the dimension “Commercial Objectives” and we derived two table in the following in which we classify our results into five concepts of commercial objectives. We created seven columns in which we categorize our papers and case study in: paper title, author, year and five concepts from literature. According to the approach of Zhou et al. (2016, p. 10), we marked records in following two tables if the paper and Facebook OSS projects include the information related commercial objectives. We have different source to retrieve additional information, for example, on Facebook Developer blogposts (Jia, 2018, p. 1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Literature** | | **Commercial Objectives** | | | | |
| **Concepts from Literature** | | | | |
| **Paper titel** | **Author (Year)** | **C1** | **C2** | **C3** | **C4** | **C5** |
| [1] xxx | Autor X, 2016 | x |  |  | x |  |
|  |  |  | x |  |  | x |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# **Results**

## Findings

In this section, we provide our findings of the literature review and the case study, which will be presented as an overview of reasons for companies to invest into OSS projects and recent OSS projects trends.

### Overview of Reasons for Companies to Invest into FLOSS Projects

We found 9 scientific papers, which investigate on open source projects in companies, which we listed in Table 4‑1 and classified into five concepts: Software Quality, Outstanding Technical Support, Business Opportunity, Business Model and External Innovation. The following Table 4‑1 provides an overview of our findings:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Literature Review** | | **Commercial Objectives** | | | | |
| **Concepts from Literature** | | | | |
| **Paper titel** | **Author (Year)** | **C1** | **C2** | **C3** | **C4** | **C5** |
| [1] Understanding community participation and engagement in open source software Projects: A systematic mapping study | Rajdeep Kaur, Kuljit Kaur Chahal, Munish Saini, 2020 | x |  |  |  |  |
| [2] Eight Observations and 24 Research Questions About Open Source Projects: Illuminating New Realities | Matt Germonprez, Georg J.P. Link, Kevin Lumbard, Sean Goggins (2018) | x | x | x | x |  |
| [3] Digital Democracy: Episode IV - A New Hope\*: How a Corporation for Public Software Could Transform Digital Engagement for Government and Civil Society | John Gastil, Todd Davies (2020) |  |  | x | x |  |
| [4] |  |  |  |  |  |  |
| [5] |  |  |  |  |  |  |
| [6] |  |  |  |  |  |  |
| [7] |  |  |  |  |  |  |
| [8] |  |  |  |  |  |  |
| [9] |  |  |  |  |  |  |
| **Legend**  C1: Software Quality  C2: Outstanding Technical Support  C3: Business Opportunity  C4: Business Model  C5: External Innovation | | | | | | |

Table 4‑1 - Overview of Reasons for Companies to Invest into OSS Projects

**C1: Software Quality:** According to Kaur et al. (2020, p. 12), the assurance of software quality and productivity depends on the active participation of the contributors. Thus, it is important to support and motivate the developers, especially, the beginners in open source communities to get involved and to avoid flucation. Such actions to maintain the contributors in the future by providing fast and polite responses to inexperienced developers, providing an experienced mentor and also a friendly environment (Kaur et al., 2020, p. 12). However, they do not investigate OSS in companies specifically but they conducted a literature review on 67 scientific papers, which deals participation and engagement in OSS projects generally (Kaur et al., 2020, pp. 4–5). [1]

In another paper, Germonprez et al., (2018, p. 13) described that the health of open source projects is an indicator for software quality. A healthy open source projects environment involves a stable and successful environment, in which contributors feels valued and acknowledged. To measure the health of open source projects companies can use analytics tools. But since there were not any consensus about which metrics should be measured to acknowledge the health of open source projects, the Linux Foundation’s Community Health Analytics Open Source Software (CHAOSS) standardized the procedure of how to measure metrics with regard on the health of open source metrics (Germonprez et al., 2018, p. 13). [2]

**C2: Outstanding Technical Support:** Germonprez et al. (2018, p. 2) investigated on open source project ecosystem of the Linux Foundation as they describe that foundation support play a major role in the evolution of open source projects (Germonprez et al., 2018, p. 2). [2]

**C3: Business Opportunity:** In our definition a business opportunity defines companies, which set a strategic and long-term goalto benefit from open source by providing open access to its platform and gaining new business opportunities as described in chapter 2.2. For example, Wikipedia set in the early era of social media platform a strategic goal to keep their platform open and provide access in order to increase their company size mentioned by Gastil & Davies (2020, p. 5). [3]

According to Germonprez et al., 2018 (pp. 14–15), open source projects need to have a secure long-term investment in order to maintain software quality. There is a urgent need in innovation of market mechanism to coordinate the development of open source projects for example through crowd marketing or funding campaign (Germonprez et al., 2018, pp. 14–15). These authors clearly indicates that companies needs to align their use of OSS and internal innovation process, which means that companies have to perform internal actions to deal with open source project health risks (Germonprez et al., 2018, p. 18). We assign this paper to C3 because the authors describes that companies have to set a strategic and long-term goal, in this case a long-term investment to deal with future risks. [2]

**C4: Business Model:** In our definition a business model aims to promote proactive engagement with open source communities to develop new products and services.Open source projects become increasingly commercialized by large organizations that have the intention to make profit from these projects and should serve their supply chain instead of keeping the open source projects as the “social constructions of freedom to the forefront” as it was traditionally (Germonprez et al., 2018, p. 11). [2]

For example, Gastil & Davies (2020, p. 5) mentioned that Facebook is a succesful business model because it stands out comparing to other companies like Friendster or MySpace in the early era of social media platform. The most striking point is that Facebook won loyal users and the most skilled programmers by putting the relationship aspects of people’s real life in the foreground, roll ads slowlier at the beginning, used venture capital to build a huge community base and showed the community the information they are interested in (Gastil & Davies, 2020, p. 5). Facebook used its resources efficiently, aligned its srategy with the communities requirements and attracted users, which at the same time, are the ground and success of Facebook. [3]

**C5: External Innovation:** New contributors involvement changes the OSS projects dynamics by gaining more ideas and skills while solving technical issues and thus, enhance innovation (Kaur et al., 2020, p. 12). However, we did not cross “C5: External Innovation” for this paper because we search for more specific information on how smaller companies benefit from innovation of greater companies, which was missing here. [1]

### Case Study: Reasons for Companies to Invest into FLOSS Projects

Additonally, we categorize our findings by analyzing the Facebook’s OSS project React and PyTorch as our case study into the five concepts as described in the previous section as well. The overview is provided in the following Table 4‑2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case Study** | | **Commercial Objectives** | | | | |
| **Concepts from Literature** | | | | |
| **Open Source Project** | **Company (Year)** | **C1** | **C2** | **C3** | **C4** | **C5** |
| [1] React | Facebook ? Jahr |  |  |  |  |  |
| [2] PyTorch | Facebook and Pytorch ? Jahr | x | / | x |  |  |

Table 4‑2- Case Study Overview of Commercial Objectives

To evaluate our case studies with Facebook OSS projects PyTorch and react, we need another definition of the five concepts C1-C5 in order to suit the OSS project requirements and our expectations. These five concepts are described in chapter 3.4

**C1 Software Quality:** According to PyTorch (2020), the latest version 1.7.0 release was on the 27th October 2020. In comparison the previous version 1.6.0 was announced on the 28th July 2020. The span between the release dates is only 4 months, so we can say that the quality of the software is often worked on and thus follows a professional quality standard. [2]

**C2: Outside technical support:** We found out that 8 of the top 10 contributors where employed in Facebook while one contributor was employed at Google but was a former employee at Facebook. The second ranked contributor in Table 5‑1 was not found on social media platforms. However, we assume that all of the top 10 contributor has a relation to the company Facebook whether in the past or present time.

**C3: Business Opportunity:** According to Zhou et al. (2016, p. 8), an OSS product is critically important for the company’s business when it fulfills to cases. First, is gaining direct profit from the OSS product. Second, is to have a strong association with profit or receiving help to gain profit (Zhou et al., 2016, p. 8). There is no evidence found that Pytorch is critically important for the company’s business by gaining a direct profit from the OSS product. However, Pytorch is partnered with large enterprises like Tesla, Amazon and Uber etc. This is an indication that Pytorch is associated with great profit (Wikipedia, 2020). [2] 🡪 React profit genauso begründen!

**C4: Business Modell:** …

**C5: External Innovation:** …

|  |  |
| --- | --- |
| 4. Business Modell | This means if a software was developed by promoting proactive engagement with open source communities to develop this software. We follow the question: “Was this software created to solve a business model problem? For example expanding ressources. |
| 5. External Innovation | describes if the software has impacted smaller companies which benefit from the innovation of the software |

### OSS Project Trends

# Diagramm hier wäre nice!

# Kann man hier einen Trend erkennen oder sogar vorhersagen? Predictive Analytics als Methode denkbar? Ansonsten nur eine kleine Tabelle machen und kurz erklären.

# Trend von was genau? Projekt-Trend in welche nun mehr investiert werden? Und in welche nicht? Wäre cool hier zwei drei Linien im diagramm zu haben, wo gezeigt wird, dass zB weniger in web technologies investiert wird als in AI, oder vllt auch gleich viel?

**Trend wird hier genau beschrieben:**

The new realities of open source projects are shifting who benefits from the wealth generated in these projects. We observed many new realities (e.g., changes of membership from volunteers to paid employees to a lack of effective market mechanisms). These new realities are interrelated and cannot be solved by a single open source project or organization, but provide new forms of engagement in the design and distribution of software. New mechanisms of practice and research can be developed to incentivize the development of healthy projects that produce secure software, incentivize the engagement of volunteers who are not affiliated with an organization, and continue to leverage the traditional and successful open source development model. 🡪 (Germonprez et al., 2018, p. 18).

**Trend in innovation:**

As open source project engagement becomes increasingly domesticated and stable — it becomes more closely aligned with corporate innovation. While there may be free agents within open source projects, they exist on a considerably more limited scale when open source project engagement is a means for getting corporate work done. This does not mean that these engagements are not open source project engagements, but it does mean that the membership of these open source projects is changing. In short, the licensing of the project remains open, but the member practices around the work on these projects are materially distinct from earlier characterizations of open source project research. 🡪 (Germonprez et al., 2018, p. 12)

## Discussion

(2 Pages)

* Das besondere wäre in unserem Term Paper, dass unser Modell angewendet werden kann, um Gründe zu finden warum ein IT Unternehmen in FLOSS porjects intervestiert, ohne dabei auf die leute zu zugehen und Interviews durchzuführen.

While analyzing the two Facebook OSS projects, we try to answer the question “Q2: Is the OSS product critically important for the company’s business?”.

Warum haben wir Pytorch und react gewählt?

Although we do not have a complete knowledge of the companies’ intentions, we could learn about company actions, visions, goals, market share, and values based on the information gathered from company websites, news articles, personal blogs, commit comments, and other documents. Therefore, we conducted an extensive Internet search for the materials relevant to this study: (Zhou et al., 2016, p. 9)

In dem paper wird nur gesagt, dass OSS durch den US staat gefördert werden sollte indem die bevölkerung ihren teil dazu beiträgt und vom government gefunded und gemanged wird. Es gibt keine direkten bezug von OSS projects im unternehmen, aber allgemein werden diese unternehmen wie facebook immer wieder erwähnt und sozusagen als vorreiter bzw. vorbild genommen wie man OSS projects erfolgreich durchzieht

The reasons to build public software in the United States are thus both negative and reactive to recent events on one hand, as well as positive and of long standing on the other. We wish to stop the threats that commercial- driven social media pose to our democratic culture (a negative reason), but also to build a healthier civic and community life than we have ever had. Positive motivations like those have spurred various efforts to innovate over the past 20 years, encouraged by the surprising success of free and open source software (e.g., Linux, Apache, Firefox, Wordpress, and Drupal) and the very democratically operated Wikipedia, powered by the MediaWiki software, which is licensed under the GNU General Public License v2+. (Gastil & Davies, 2020, p. 4)

## Conclusion

(1 Page) (+ Research Gap)

**Aim & Short summary:** …

**Method**: We identified two OSS projects invested by Facebook in the field of AI and web technologies. We collected scientific literature, repository data and histories based on GitHub Facebook projects participation. To extract relevant information we used generalized linear models (?) to measure project participation and built a two dimensional-model to provide a guideline how to identify companies reason to invesrt in FLOSS projects generally.

**Result:** …

# Appendix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rank** | **GitHub User ID** | **Facebook employee** | **Non Facebook employee** | **Commits** |
| **PyTorch** | | | | |
| 1 | ezyang | x |  | 1.456 |
| 2 | gchanan | / | / | 1.078 |
| 3 | apaszke |  | x @Google | 870 |
| 4 | jerryzh168 | x |  | 734 |
| 5 | soumith | x |  | 734 |
| 6 | zdevito | / | / | 623 |
| 7 | Yangqing | x |  | 601 |
| 8 | colesbury | x |  | 582 |
| 9 | smessmer | x |  | 552 |
| 10 | zou3519 | x |  | 544 |
| **React** | | | | |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| Pytorch: <https://github.com/pytorch/pytorch/graphs/contributors>  React: | | | | |

Table 5‑1 - Top 10 Contributors of PyTorch and React

**Literature**

AltexSoft. (2017). *What Software Quality (Really) Is and the Metrics You Can Use to Measure It*. https://www.altexsoft.com/blog/engineering/what-software-quality-really-is-and-the-metrics-you-can-use-to-measure-it/

Ballhausen, M. (2019). Free and Open Source Software Licenses Explained. *Computer*, *52*(6), 82–86. https://doi.org/10.1109/MC.2019.2907766

Facebook. (2020). Open Source projects by Facebook. *Facebook Open Source*. https://opensource.facebook.com/projects

Gastil, J., & Davies, T. (2020). Digital Democracy: Episode IV—A New Hope\*: How a Corporation for Public Software Could Transform Digital Engagement for Government and Civil Society. *Digital Government: Research and Practice*, *1*(1), 1–15. https://doi.org/10.1145/3342194

Germonprez, M., Link, G. J. P., Lumbard, K., & Goggins, S. (2018). Eight Observations and 24 Research Questions About Open Source Projects: Illuminating New Realities. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), 1–22. https://doi.org/10.1145/3274326

Jia, B. (2018). Announcing PyTorch 1.0 for both research and production. *Facebook for Developers*. https://developers.facebook.com/blog/post/2018/05/02/announcing-pytorch-1.0-for-research-production/

Kaur, R., Kaur Chahal, K., & Saini, M. (2020). Understanding community participation and engagement in open source software Projects: A systematic mapping study. *Journal of King Saud University - Computer and Information Sciences*, S1319157820305139. https://doi.org/10.1016/j.jksuci.2020.10.020

Kemp, R. (2010). Open source software (OSS) governance in the organisation. *Computer Law & Security Review*, *26*(3), 309–316. https://doi.org/10.1016/j.clsr.2010.01.008

Kochhar, P. S., Kalliamvakou, E., Nagappan, N., Zimmermann, T., & Bird, C. (2019). Moving from Closed to Open Source: Observations from Six Transitioned Projects to GitHub. *2019*, 1–22. https://doi.org/10.1109/TSE.2019.2937025

Logothetis, S., & Stylianidis, E. (2016). BIM Open Source Software (OSS) for the documentation of cultural heritage. *Virtual Archaeology Review*, *7*(15), 28. https://doi.org/10.4995/var.2016.5864

PyTorch. (2020). *PyTorch 1.6 released w/ Native AMP Support, Microsoft joins as maintainers for Windows*. https://pytorch.org/blog/pytorch-1.6-released/

Wagstrom, P. A. (2009). *Vertical Interaction in Open Software Engineering Communities*. https://patrick.wagstrom.net/thesis/wagstrom-phd-thesis-single.pdf

Wikipedia. (2020). PyTorch. In *PyTorch*. https://en.wikipedia.org/wiki/PyTorch

Zhou, M., Mockus, A., Ma, X., Zhang, L., & Mei, H. (2016). Inflow and Retention in OSS Communities with Commercial Involvement: A Case Study of Three Hybrid Projects. *ACM Transactions on Software Engineering and Methodology*, *25*(2), 1–29. https://doi.org/10.1145/2876443