

Adoption of Open Source Software in Software-Intensive Organizations - A Systematic Literature Review

Øyvind Hauge^{*,1}, Claudia Ayala^{1,2}, Reidar Conradi¹

IDI, NTNU, Sem Sælands vei 7-9, NO-7491 Trondheim, Norway

Abstract

Context: Open source software (OSS) is changing the way organizations develop, acquire, use, and commercialize software.

Objective: This paper seeks to identify how organizations adopt OSS, classify the literature according to these ways of adopting OSS, and with a focus on software development evaluate the research on adoption of OSS in organizations.

Method: Based on the systematic literature review method we reviewed publications from 24 journals and seven conference and workshop proceedings, published between 1998 and 2008. From a population of 24289 papers, we identified 112 papers that provide empirical evidence on how organizations actually adopt OSS.

Results: We show that adopting OSS involves more than simply using OSS products. We moreover provide a classification framework consisting of six distinctly different ways in which organizations adopt OSS. This framework is used to illustrate some of the opportunities and challenges organizations meet when approaching OSS, to show that OSS can be adopted successfully in different ways, and to organize and review existing research. We find that existing research on OSS adoption does not sufficiently describe the context of the organizations studied, it is furthermore fragmented, and it fails to benefit fully from related research fields. To aid this situation, we offer directions for future research.

*Corresponding author. Tel: +47 97 71 22 52; Fax: +47 73 59 44 66

Email addresses: oyvind.hauge@idi.ntnu.no (Øyvind Hauge),
cayala@lsi.upc.edu (Claudia Ayala), reidar.conradi@idi.ntnu.no (Reidar Conradi)

¹Norwegian University of Science and Technology

²Technical University of Catalunya

Conclusion: The implications of our findings are twofold. On the one hand, practitioners should embrace the many opportunities OSS offers, but consciously evaluate the consequences of adopting it in their own context. They may use our framework and the success stories provided by the literature in their own evaluations. On the other hand, researchers should align their work, and perform more empirical research on topics that are important to organizations. Our framework may be used to position this research and to describe the context of the organization they are studying.

Key words:

open source software, organizations, software development, systematic literature review

1. Introduction

The open source software (OSS) phenomenon has over the last decade had a significant impact, not only on the software industry, but also on software-intensive organizations in both the public and private sector. The collaborative development model often associated with OSS communities has introduced a new software development model. This model has inspired software companies into evolving their existing development processes [60, 203] and collaborating both internally and across company borders [7]. Next, it is claimed that the existence of freely available software allows faster adoption of technology, increased innovation, and reduced costs and time-to-market [33, 139]. These potential advantages have influenced how organizations acquire software, and have led to a significant adoption of OSS products in several domains [89, 98, 140, 187]. Finally, OSS and its general lack of license fees contribute to shifting the software industry's traditional license-based business models towards service-based models [79]. Hence, OSS is significantly influencing the ways organizations develop, acquire, use, and commercialize software [69].

It is therefore vital to help organizations in meeting the challenges related to OSS, and to align our research efforts with their real needs. The topic for this study is therefore *adoption of OSS in software-intensive organizations, with a particular focus on software development*. To identify what we know about how organizations adopt OSS we have performed a systematic literature review following the guidelines proposed by Kitchenham [115]. With a focus on software development, this systematic literature review seeks to

evaluate, synthesize, and present the empirical research results on OSS within organizations.

The targeted audience for this systematic review is primarily researchers in the OSS, software engineering, and information systems fields, wanting to study settings involving OSS and organizations. However, organizations and practitioners which adopt (or plan to adopt) OSS may also appreciate the review.

This systematic literature review contributes to the literature and ongoing research on OSS within the software engineering and information systems domain in three ways: (1) by reviewing and summarizing what we know about how organizations actually leverage OSS, (2) by providing a classification framework for how organizations adopt OSS, and (3) by offering directions for future research on OSS in organizations.

Researchers and practitioners may use these contributions to more clearly understand the practical challenges when adopting OSS, and properly align their efforts for facing them. Researchers may use this literature review to get an overview of current research, identify new research questions, and position and align their own work. More importantly, they may use our classification framework to describe and discuss the context of the organizations they study. Practitioners may use this framework and the success stories provided here to understand how they may leverage OSS in their own context, and to identify the practical challenges they might face when doing so.

The remainder of this paper is structured as follows: In Section 2 we provide a brief background to OSS and an overview of other reviews of the OSS literature. We develop a classification framework for how organizations adopt OSS. In addition, we relate OSS to relevant research areas and present the objectives of this study. In Section 3 we describe the systematic review process. In Section 4 we characterize the literature on adoption of OSS in organizations and use the classification framework to present key issues from this literature. In Section 5 we answer our research questions and discuss the results with their implications and limitations. Moreover, we provide directions for future research. In Section 6 we conclude the paper.

2. Background

In this section we give a brief background on the OSS phenomenon and summarize other reviews on OSS research. We present a classification framework for organizational adoption of OSS. Then, we relate research on OSS

to relevant areas in software engineering and information systems research. Finally, we present the objectives for this literature review.

2.1. Open Source Software

Eric Raymond describes the development of OSS as a bazaar-like activity driven by volunteers, and claims that OSS is cheaper, has fewer defects, gets improvements faster, and is generally better than “other kinds” of software [154, 155]. Based on the Apache and the Mozilla projects, Mockus et al. [136] describe OSS development as controlled by groups of core developers and supported by large communities of contributors. They hypothesize that the OSS products have lower defect density than commercial software and that OSS development rapidly responds to user requests. Others, e.g. Crowston and Howison [44] and Scacchi [166], support this view and claim that the development in OSS communities is distinctly different from traditional software development.

This view of OSS and OSS development as being something radically different has triggered research on a variety of topics. These include OSS seen as a new innovation model [194], the motivations of OSS developers [102], OSS business models [27] and a wide spectrum of other research topics in computer science, management and organization science, social science, psychology, economics, and law [85].

Software engineering research has for instance studied self-organizing in OSS communities [45, 212], user-to-user support [119], knowledge management [173], and quality assurance [215]. Software engineering researchers have additionally used OSS products to study general software engineering problems like evolution [211], cloning [114], and the use of metrics to identify error prone classes [170].

However, the view that the development of OSS is something radically different from traditional software development is questioned by, for instance Fitzgerald [78] and Fugetta [84]. Østerlie and Jaccheri [146] offer a critique of how OSS development has been described as a homogeneous phenomenon in the software engineering research literature. The literature has not reflected the variety observed in the OSS phenomenon, but rather has focused on large, successful, and community-driven OSS projects. Moreover, Capiluppi et al. [40] provide evidence that the majority of OSS projects struggle to attract contributors, Noll [142] shows that OSS can also be developed inside commercial software development companies without any active communities, and Stamelos et al. [177] show that the quality of OSS software is not

always as good as expected. Finally, Fitzgerald [79] argues that the OSS phenomenon has evolved into a more commercially viable form where volunteers and commercial organizations collaboratively contribute to evolving the phenomenon.

There are thus conflicting views on what the OSS phenomenon actually is and there is not even consensus on which label to use on the phenomenon. We acknowledge that there are (minor) differences between open source, free software, and free (libre) open source software (FOSS/FLOSS). However, this ongoing debate is beyond the scope of this paper. We will instead treat OSS, free software, and FLOSS as synonyms, and focus on software development and the parts of the phenomenon where commercial organizations are involved. We will in particular look at three aspects of it related to organizations: (1) the use of software products licensed with a license approved by the Open Source Initiative [145], (2) the interaction with the communities surrounding many OSS products, and (3) the use of the collaborative software development practices often associated with many of these communities.

2.2. Summary of Previous Reviews

Reviews of the literature on OSS are given by Feller et al. [71], Stol and Babar [180], Scacchi et al. [167], von Krogh and von Hippel [196], and the aforementioned paper by Østerlie and Jaccheri [146]. While all these reviews are on OSS, none of them focus on OSS in organizations. In fact, the adoption of OSS in organizations is hardly mentioned by any of the authors.

Feller et al. [71] aim to identify the kinds of OSS communities that have been studied, the kinds of research questions that have been asked, and the methods researchers have used to answer these questions. The paper mainly focuses on classifying and characterizing a set of 155 publications on OSS. Feller et al. find that the OSS research literature has large gaps, and that commercial organizations are underrepresented as subjects in the research on OSS.

Stol and Babar [180] reviewed 219 publications from the four first International Conferences on Open Source Systems. Like Feller et al. [71], Stol and Babar focus on assessing the quality of the 63 empirical studies and find that the literature needs to be improved. To this end, they offer a set of guidelines for improving the quality of studies on OSS. Moreover, they classify the empirical papers into: research on OSS communities (39.7% of the papers), OSS development and maintenance (20.6%), diffusion and adoption of OSS (28.6%), and characteristics of OSS (11.1%). The 18 papers that

focus on diffusion and adoption of OSS cover the following topics: perceptions of OSS, incentives to adopt OSS, migration to OSS, and usage of OSS. By showing that only 18 (8.2%) of the 219 papers present empirical studies from organizational contexts, Stol and Babar confirm that such studies have attracted limited attention.

In an introduction to a special issue, Scacchi et al. [167] provide an overview of the research on the development processes found in OSS projects. Von Krogh and von Hippel [196] give an overview of some of the research on OSS and organize it into three categories: motivations of contributors, innovation processes, and competitive dynamics.

2.3. A Classification Framework for Organizational Adoption of OSS

To identify the challenges organizations face when approaching OSS and to classify the literature, we developed a classification framework consisting of six ways organizations adopt OSS. By *ways of adopting OSS*, we think of ways in which software-intensive organizations can benefit from OSS products, the communities surrounding many of these products, or the development practices often associated with the collaborative development of many such products. We limit our focus to *software-intensive organizations* which we define as private or public organizations extensively using or developing software. Moreover, we focus on ways of adopting OSS that influence an organization's software infrastructure or software development.

We briefly present this framework in Table 1. By using the empirical evidence identified in this review we will discuss some of the relations between the different categories in Section 5.1.

2.3.1. The Classification Framework

The classification framework in Table 1 contains two main areas in which organizations can benefit from OSS. First, *deploying OSS products* entails the use, and if necessary configuration, of a spectrum of software. These products range from infrastructure software (like operating systems, databases, and application servers), through server-based software applications, to desktop applications. Second, *using OSS in software development* can be broken down into five categories.

Using OSS CASE tools involves using tools like integrated development environments (IDEs), compilers, modeling tools, and so on. The use of OSS CASE tools is indeed an example of OSS deployment. Still, we decided to keep OSS CASE tools as a separate category for three reasons. First, the

Table 1: Organizational adoption of OSS

Way of adopting OSS		Example papers
<i>Deploying</i> OSS products in their operation environment as end users (e.g. deploying OpenOffice.org, Linux, JBoss)		[62, 81, 190]
Software development	Using OSS CASE <i>tools</i> in software development (e.g. using Eclipse, Subversion, GCC)	[13, 131]
	<i>Integrating</i> OSS components into their own software systems (e.g. integrating or extending Hibernate, Google Web Toolkit, Plone)	[8, 43, 186]
	<i>Participating</i> in the development of OSS products controlled by another organization or community (e.g. contributing to Linux, Eclipse, OpenOffice.org)	[28, 101, 158]
	<i>Providing</i> their own OSS products and relating to a community around these products (e.g. providing MySQL, Qt, JBoss)	[7, 17, 202]
	Using software development <i>practices</i> , often associated with OSS communities, within a company or consortium of companies (e.g. using practices like code sharing, peer reviewing, user contributions)	[61, 135, 203]

focus of this review is software development, where CASE tools are extensively used. Second, there are large numbers of OSS CASE tools available³. Third, there is already an established research field focusing on Computer Aided Software Engineering (CASE) tools.

The integration of OSS components involves including OSS components into other software products or systems. This integration may involve modifying, extending, or wrapping the OSS components. Even though both deployment and integration of OSS entail reusing OSS products, it is valuable to separate the two. Organizations that extend and possibly modify an OSS product increase their dependence on the product and face additional challenges related to maintenance. The difference between simply deploying an OSS product and integrating it into one of your software systems is one of degrees. For instance, building applications with tight integration of e.g. architectural frameworks and persistence layers poses significantly different challenges than simply deploying a desktop application.

By participating in the development of OSS, we mean the involvement of organizations in existing OSS communities, although without having decisive control over the OSS product or the community. Providing an OSS product, involves organizations like JBoss, MySQL, and Qt Software, that develop and release OSS products, control the development of these products, and relate to the community around them. The difference between the two categories is again one of degrees. However, the challenges tied to relating to a community

³For instance <http://www.tigris.org/> has more than 500 such tools

of thousands of users around one of your own products are different from those related to contributing a bug-fix to a product controlled by someone else. The division between providing and participating is also noticed by, for instance, Ågerfalk and Fitzgerald [7] and Dahlander and Magnusson [50].

There is no set of development practices that are universal to all OSS projects. Nevertheless, practices such as user participation, short release cycles, and peer code reviews have frequently been associated with OSS projects and are often labeled “OSS practices” [72, 166, 215]. Lately, several organizations have tried to learn from the development practices in successful OSS projects, through applying these practices within their own organization [60, 135, 203].

2.3.2. Related Classifications

The framework extends earlier work in three ways. First, we identify new ways of adopting OSS as compared to Hauge et al. [99] and Ziemer et al. [216]. In [99], we present the four roles of OSS integrator, OSS participant, OSS provider, and inner source software participant as possible ways of approaching OSS. In [216], we present cases from companies that adopt OSS through development with OSS practices and tools, development with OSS products, and development of OSS products. Second, we have a somewhat broader scope than Dahlander and Magnusson [49, 50], who focus on the relationships between organizations and OSS communities. Third, we focus on software development rather than resource allocation as in Grand et al. [94], or business models as in Hecker [100]. However, the levels in the four-level ladder for resource allocation presented by Grand et al. [94] are partly compatible with some of our categories. Table 2 relates our framework to relevant classifications in [94, 99, 216].

Table 2: The framework and its mapping to related classifications

Way of adopting OSS		Mapping to other papers
Deploying OSS products		Level 1 [94]
Software development	Using OSS CASE tools	Level 1 [94] and Development with OSS practices and tools [216]
	Integrating OSS components	Level 2 [94], OSS integrator [99], and Development with OSS products [216]
	Participating in OSS communities	Level 3 [94], OSS participant [99]
	Providing OSS products	Level 3/4 [94], OSS provider [99], and Development of OSS products [216]
	Using OSS development practices	Inner source software participant [99], and Development with OSS practices and tools [216]

2.4. Related Research Fields

In their effort to define OSS, Gacek and Arief [85] consider research fields like computer science, management and organization science, social science, psychology, economics, and finally law as relevant. To put OSS research into context we will relate it to relevant research areas. However, as this literature review focuses on software development, we will consider research areas only within software engineering and information systems. While we draw parallels between OSS and some related research areas in Table 3, it is not an extensive list.

Table 3: OSS research in relation to other research areas

Way of adopting OSS	Related research areas
Deploying OSS products	Introduction, deployment, diffusion, and acceptance of information systems (IS) and information technology [191, 192]
Using OSS CASE tools	Computer Aided Software Engineering (CASE) [83, 207]
Integrating OSS components	Component-Based Software Engineering (CBSE) [35, 122, 134, 208] and software reuse [137, 193]
Participating in OSS communities	No clearly related research area within SE/IS. However, Ågerfalk and Fitzgerald relate their research to offshoring and outsourcing [7]
Providing OSS products	
Using OSS development practices	Software process improvement [1, 65], distributed development [150], global software development [174], and agile development [199]

In this review we will consider three ways of using OSS products. OSS products may first of all be deployed as-is without any changes, used as CASE tools in software development, or integrated into other software systems. All these three ways of using OSS products are related to established research areas within software engineering and information systems research.

The introduction, diffusion, and acceptance of information systems and information technology have already been studied for a long time in the information systems field [191, 192]. As we mentioned above, there is an established research area on CASE tools [83, 207]. Finally, the integration of OSS components is closely related to research on, for instance, Component-Based Software Engineering (CBSE) [35, 122, 134, 208] and software reuse [137, 193].

A large part of the OSS phenomenon is centered on community interaction, either as a provider of an OSS product or as a participant in a community controlled by someone else. Within the software engineering and information systems research fields, we find no clearly related research areas. Nevertheless, Ågerfalk and Fitzgerald [7] relate their research on company intervention in OSS communities with offshoring and outsourcing.

Finally, Scacchi et al. [167] discuss the software development processes and practices used in OSS communities. Such work can be related to, for instance, software process improvement [1, 65]. Other researchers relate the development processes in OSS communities to distributed development [150], global software development [174], and agile methods [199].

2.5. Objectives of this Review

Our overall objective of summarizing what we know about how organizations adopt OSS has been broken down into three more concrete research questions:

- RQ1** In what ways are software-intensive organizations adopting OSS?
- RQ2** What has been the focus of the empirical research on adoption of OSS in organizations?
- RQ3** What are the characteristics and limitations of current, empirical research on organizational adoption of OSS?

Even though we see that organizations approach OSS in different ways there are several publications discussing organizational adoption of OSS without clarifying how or what the involved organizations actually do related to OSS. Some papers discuss “F/OSS usage and adoption” in public administration [148], or companies that “have entered the open source field” [27] and “are active in the OSS domain” [153]. Understanding the practical implications of a specific way of adopting OSS is therefore difficult. So, by RQ1, we want to identify existing ways of leveraging OSS.

When identifying these ways of adopting OSS we will in particular focus on organizations that develop software and approaches to OSS related to software development. This focus is also valid for RQ2. Through RQ2, we seek to identify the focus of research on OSS in organizations, but with a particular focus on software development. Finally, by RQ3 we aim to characterize the research on OSS in organizations, assess its quality, and in particular identify its limitations.

3. Research Method

The evidence-based software engineering (EBSE) paradigm aims to integrate the current best evidence from research with practical experience [117]. Literature reviews, and in particular systematic literature reviews,

have therefore become popular within the software engineering research field as a means of evaluating what we know in a specific area. For instance, researchers have reviewed published evidence on search-based testing [5], knowledge management [21], cost estimation [111], and several other topics [116].

To answer our research questions, we systematically assessed existing evidence related to the adoption of OSS in organizations, using Kitchenham’s guidelines for systematic literature reviews [115]. The review process was split into several stages, each of which was performed individually by two researchers followed by an iterative process to reach consensus before facing a new stage. The following subsections describe the steps of the literature review.

3.1. Search and Classification Process

The search process combined searching digital libraries with manual evaluation and classification of the results. Figure 1 presents an overview of the review process and the number of publications included in each stage.

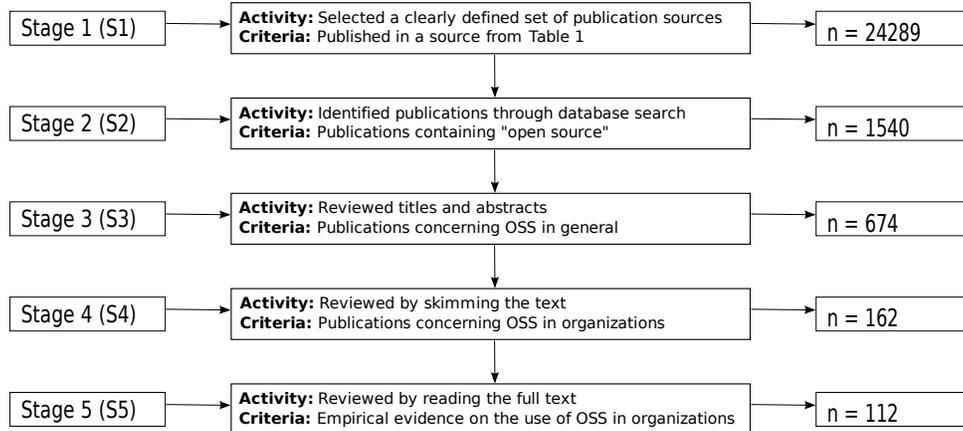


Figure 1: Stages of the study selection process (Adapted from [66])

3.1.1. Stage 1 - Defining a Set of Publications

To have a clearly defined set of publications serving as a basis for this study, we selected a set of publication channels rather than openly searching available digital libraries. Relevant journals and conferences were taken from previous literature reviews on software engineering [66, 92, 113, 116, 171]

and OSS [71, 146, 167, 196]. Table 4 gives an overview of the final sample of publication sources.

Appendices A and B contain more detailed overviews of where we collected the publications, the publication sources’ total number of publications between 1998 and 2008, the number of publications included in each of the five stages (S1-S5), and so on.

Table 4: Publication sources

	Software Engineering	OSS, Information Systems, and Management
Journals/magazines	<ul style="list-style-type: none"> • ACM Transactions on Software Engineering and Methodology (TOSEM) • Communications of the ACM • Empirical Software Engineering (EMSE) • IEE Review • IEE Software Proceedings/IET Software • IEEE Computer • IEEE Software • IEEE Transactions on Software Engineering (TSE) • Information and Software Technology (IST) • Journal of Systems and Software (JSS) • Software Practice and Experience • Software Process: Improvement and Practice 	<ul style="list-style-type: none"> • First Monday • Information Systems Journal (ISJ) • Journal of Database Management • Journal of Industrial Economics • Knowledge Technology and Policy • Long Range Planning • Management Science • MIS Quarterly • MIS Quarterly Executive • MIT Sloan Management Review • Organization Science • Research Policy
Conferences/workshops	<ul style="list-style-type: none"> • IEEE International Symposium on Empirical Software Engineering (ISESE) • IEEE International Symposium on Software Metrics (METRICS) • International Conference on Software Engineering (ICSE) • International Symposium on Empirical Software Engineering and Measurement (ESEM) 	<ul style="list-style-type: none"> • ICSE Workshop on Open Source Software Engineering (WOSSE) • ICSE Workshop on Emerging Trends in FLOSS Research and Development • International Conference on Open Source Systems (OSS)

3.1.2. Stage 2 - Searching with Keywords in Digital Libraries

Publications on OSS from the specific journals and conference proceedings were identified by searching through a variety of digital libraries and manually reviewing several static web pages. Only papers published in English between 1998 and 2008 were considered.

To avoid overlooking relevant publications, we opted for a search strategy with high sensitivity [58]. This means that instead of using keywords like “commercial open source” and “open source in industry”, we conducted all searches using the keywords “open source” (including quotation marks) and searched the digital libraries using all fields, including full text where available. For IEEE Xplore only metadata search was used, even though we

discovered late in the review process that it was also possible to search the database using full text search (see the discussion in Section 5.5).

To verify that using the keywords “open source” included all relevant publications, we also tried searching using the keywords “free software”. However, relevant papers also contained the keywords “open source”.

The searches were conducted from November 2008 and finally revised January 2009. Bibliography for all the publications was stored in the external bibliography system⁴.

3.1.3. Stage 3 - Reviewing Titles and Abstracts - Papers on OSS in General

To identify publications that in fact were about OSS and did not just contain the keywords “open source”, we individually reviewed the 1540 papers from the previous stage based on their titles and abstracts, and if necessary by skimming the full text. Only papers on OSS topics like communities, software development, licensing, business models, adoption, use, and software engineering were included. Papers on other open, collaborative activities (open courseware, wikinomics, open access) were rejected. Additionally, we rejected introductions of panels, conferences, and special issues, book reviews, news flashes, and PhD symposium papers.

First, a total number of 763 publications were included by either of the two first authors. The agreement between the authors was very good (Kappa value of 0.83). A Kappa value between 0.81-1.0 is an almost perfect agreement and between 0.61-0.80 is a substantial agreement [120]. After two joint consensus iterations we discarded 89 of the 763 papers and ended up including 674 publications.

3.1.4. Stage 4 - Skimming the Text - Papers on OSS in Organizations

Next, to identify publications on adoption of OSS in organizations, we individually went through the output of the third stage and evaluated the papers’ topics by reviewing the titles and abstracts, and by skimming the papers. Publications on adoption, use, and development of OSS in organizations were included, while those not related to OSS in software development or actual use of OSS products were rejected. This included approaches to the OSS phenomenon from economical or social sciences and papers on innovation theory, business models, etc. Moreover, papers proposing methods or (classification) frameworks without any empirical validation were rejected.

⁴Aigaion: <http://aigaion.nl/>

Initially 211 publications were included by either one of the two first authors. The agreement was again quite good (Kappa value of 0.68). After another joint iteration we rejected 49 papers and classified 162 publications as being relevant to OSS in organizations.

3.1.5. Stage 5 - Reading the Text - Papers with Empirical Evidence

Then, we classified the publications into three categories inspired by [138]; (R) empirical research papers where the authors present evidence from a research study having an explicit research question, (E) experience reports where the authors report experiences without having defined an explicit research question, and (N) non-empirical papers. The non-empirical category includes opinion papers and theoretical papers without explicit empirical evidence.

Of the 162 included papers, 59 were classified as empirical research papers (R), 53 as experience reports (E), and 50 as non-empirical papers (N). Non-empirical papers may increase the understanding of how organizations adopt OSS, but they are not providing any new evidence of how organizations actually do this. Hence, these papers were not included. Accordingly, the final stage of the review included 112 publications.

3.2. Quality Assessment

As the research field on OSS adoption is still immature, and since there are no other review papers on the same topic, we did not want to exclude papers because of their lack of rigor. The quality assessment was therefore performed only to evaluate the rigor of the presented research in each publication.

We assessed the 59 empirical research papers using the nine quality metrics (QM) presented below. This schema is inspired by other SLRs [66, 178], and the evaluation criteria used in different journals and conferences [138]. The schema was designed to contain only binary values (yes/no). The quality of the experience reports was not assessed as they did not contain any explicit research questions and most often no descriptions of method, findings and so on.

QM1: Does the paper have a description of the research method?

QM2: Does the paper describe an explicit research question/goal/purpose?

QM3: Does the paper describe motivation for the research question(s)?

QM4: Does the paper discuss limitations or validity?

QM5: Does the paper describe the context of the research?

- QM6:** Does the paper describe data collection?
QM7: Does the paper describe data analysis?
QM8: Does the paper describe sampling or selection of the study object(s)?
QM9: Does the paper present any data?

The quality assessment was performed by one of the two first authors and verified by the other. We simply verified whether or not the publications mentioned or discussed issues related to each of the quality metrics. Assessments of the extent to which a paper actually fulfilled each of the quality metrics, and assessments of the papers' relevance to practitioners were not performed. Any differences between the two first authors were solved through discussions until a consensus was established.

3.3. Data Extraction

In the data collection stage we extracted the following from each of the publications: main topic, research question or research goal, research method, study objects, a brief description of empirical evidence relevant to use of OSS in organizations, and the affiliation and home country of the first authors.

Following the recommendations by [36], one researcher extracted the data, while the other confirmed the extracted data. Similar to [67], we frequently used consensus meetings to extract data. These meetings were also used to identify the topics that are discussed in the next sections of this review.

4. Results

In this section we describe the results from the evaluations of the publications included in the final stages of this review. Based on these evaluations we present some characteristics of the literature on OSS in organizations. Then, we use the classification framework from Section 2.3 to give an overview of the topics discussed in the literature, and to identify the most important findings and challenges related to software development.

4.1. Characteristics of the Literature

Here we characterize the literature by presenting the number of publications per year, the contexts in which research has been performed, and the applied research methods. Finally, we present the results from the quality assessment. These overviews focus mainly on the 59 empirical research papers. More detailed data can be found in Appendix B.

4.1.1. OSS Related Activity in the Literature

Both the number of publications related to OSS in general (S3) and to OSS in organizations (S4) has increased significantly the last decade. However, this increase seems to have stabilized in the last years, with about 100 publications per year on OSS and between 20 to 30 publications on topics related to OSS in organizations (see Figure 2). While the results show a significant lack of empirical research papers on OSS in organizations until 2003, the number of such papers has increased since then.

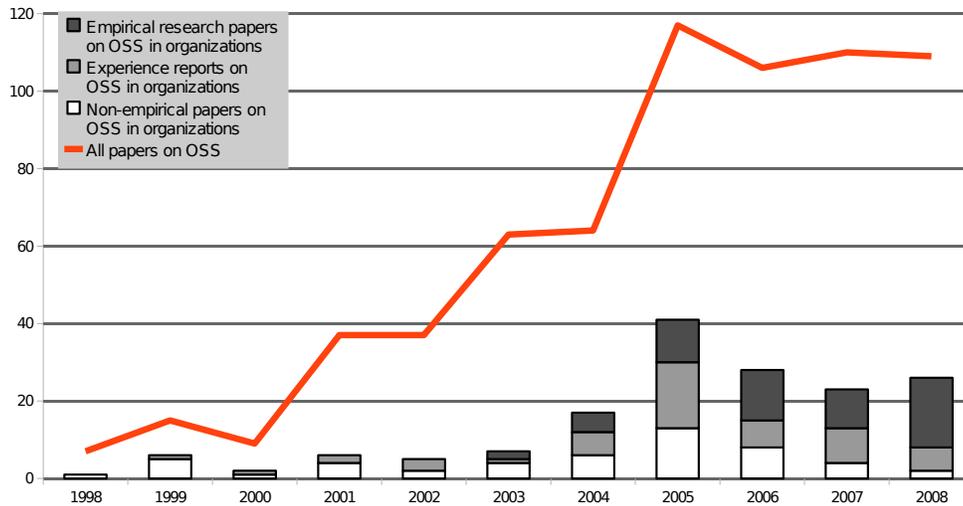


Figure 2: The number of papers on OSS and OSS in organizations

The distribution of the papers with respect to our classification framework is shown in Figure 3. The majority of the 112 empirical papers could be put into one of the six categories. However, 21 papers discussed issues relevant to two or more categories. Another 19 papers were not specific to any particular way of adopting OSS, but rather discussed topics related to adoption of OSS in general. Hence, it was necessary to add another category (OSS adoption in general) to classify all the publications. It is, however, important to note that Table 1 shows how an organization may adopt OSS, while Figure 3 presents a classification of the empirical papers on OSS in organizations.

4.1.2. The Contexts in which Research Is Done

We see that almost all the papers have a first author from Europe (49 empirical research papers (R) and 29 experience reports (E)) or North America

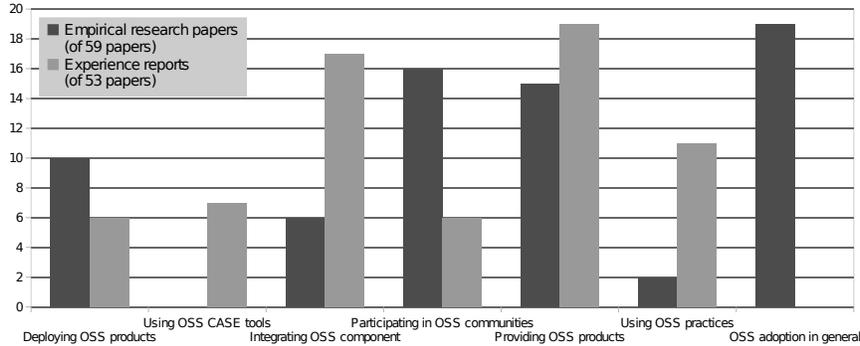


Figure 3: The number of papers in each category

(7 R and 22 E). Publications from the other continents are almost non-existent, as few publications occur from Australia (1 R and 0 E), Asia (2 R and 1 E), and Africa (1 R and 0 E). The USA (6 R and 22 E) and Italy (13 R and 8 E) were the two countries with by far the most publications.

From Figure 4 we see that the contexts where the research is performed come from both the private and public sector. However, the papers often focus on large communities such as GNOME and Debian GNU/Linux, portals like SourceForge [20, 28, 101, 158], large companies as Nokia, Philips Medical, and Hewlett Packard [107, 135, 203], and well known OSS companies like MySQL and JBoss [49, 202]. However, this bias was not as significant as expected.

A few other issues are worth mentioning. First, all of the eight empirical research papers from the public sector focus on deployment of OSS products. Besides [37], which has a mixed sample, no paper focuses on deploying OSS in the private sector. Second, 27 of the 59 empirical research papers report findings from samples of several organizations from the private sector. However, as few as eight papers report findings from one single private organization. Hence, most research papers dedicate relatively little space to describing the individual organizations.

4.1.3. The Research Methods

We classified 53 of the 112 empirical papers identified in this review as experience reports. Hence, the most common method of studying the OSS phenomenon in organizations is through experience reports. These experience reports lack explicit research questions, and most also lack a method description.

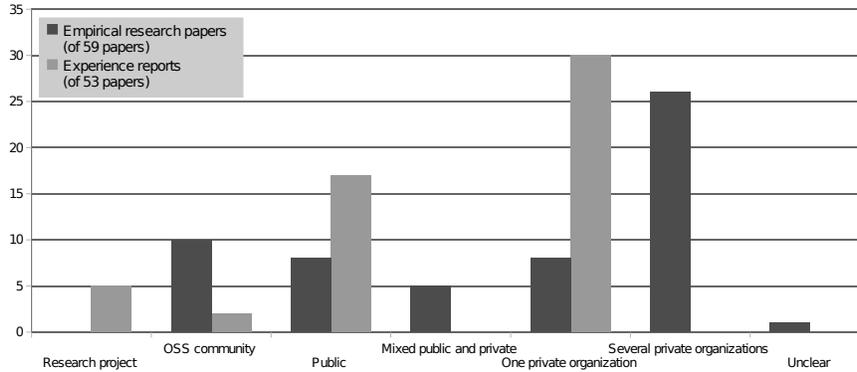


Figure 4: The context of the empirical research papers and the experience reports

To categorize the 59 empirical research papers, with respect to their research methods, we used the classification provided by Glass et al. [92]. Table 5 presents an overview of the research methods reported in these papers, a brief description of these methods, and the number of papers that were classified into each category. We observed a focus on case studies and surveys.

Table 5: Research methods used

Research method	Number of publications
Case study The combination of several data collection techniques like interviews, workshops, and analysis of documents to study one or a few contexts (cases).	29
Survey The use questionnaires or interviews to survey opinions, practices, and so on from a (large) population.	18
Data analysis The analysis of data from code repositories, web-portals, and other historical data sources. Mainly statistical or trend analysis.	6
Experiment The testing of a hypothesis under (relatively) controlled conditions.	2
Case study and survey Two phased studies, combining initial case studies with broader surveys.	2
Field study The direct observation of work, including participation in meetings and other workplace activities.	1
Grounded theory The generation of a theory grounded in empirically collected data.	1

Almost all of the studies were retrospective and gathered information about past events. This was often done through the use of interviews and questionnaires. Moreover, most of the studies collected information at one point in time. Of the 15 publications we classified as longitudinal, seven presented research in the form of mining and analysis of historical data.

4.1.4. Quality Assessment of the Research on OSS in Organizations

As described in Section 3.2, we developed nine quality assessment metrics. Each of the empirical research papers were evaluated to either cover or not to cover these metrics. This was indicated by assigning each of the nine metrics a binary value of either 1 (cover) or 0 (do not cover). Most of the 59 empirical research papers got a relatively high score in this quality assessment (see Table 6). The median score of this assessment was 8, the mode 9, and the mean value was 7.0.

Table 6: Quality assessment: Distribution of research papers

Quality assessment score	0	1	2	3	4	5	6	7	8	9	Total
Number of papers	0	0	2	5	1	3	8	8	15	17	59

The binary metrics unfortunately do not reflect the precise extent to which the papers deal with the issues covered by each of the quality metrics (see further discussion in Section 5.5). In fact, most papers do little more than mention such issues as research questions, limitations, data analysis, and so on. We furthermore found that many of the publications lack details about the research methods and findings. As a consequence, several papers have limitations when it comes to how they describe these issues. Moreover, many of the research papers are explorative and they are therefore lacking a precise focus and clear contributions.

4.2. Research on OSS in Organizations

In this section we present the focus of the research literature on OSS in organizations. This overview is presented according to the classification framework presented in Table 1. Given the review’s focus on software development, we will emphasize the findings and challenges relevant to software development.

In the overview we will focus on the 59 empirical research papers, and will give a brief overview of all the included empirical research papers in Tables 7 to 12. However, we will in some cases include some of the experience reports as well. References to all the experience reports may be found in Appendix B.

4.2.1. General Topics on Adoption of OSS

As many as 19 research papers cover topics that are general to OSS adoption, rather than directly related to the individual ways in which organizations adopt OSS. These papers discuss mainly (1) the perceived benefits

and drawbacks of OSS or the motivations for adopting it [33, 139], (2) the success factors for adoption of OSS [93, 172], and (3) the extent to which OSS is actually adopted [98, 141]. See Table 7 for an overview of the empirical research papers.

Table 7: Focus and findings from the empirical research papers on adoption in general

ID	Topic
[6]	Discusses different strengths/opportunities and threats/weaknesses to OSS in the secondary software industry. Concludes that the industry is aware of both the pros and cons.
[27]	Companies select hybrid (OSS and proprietary software) business strategies when approaching OSS.
[29]	Of 769 companies only 19 provide OSS (based) solutions alone and 236 provide both proprietary and OSS (based) ones.
[32, 33, 162]	Companies emphasize (pragmatic) economic and technological motivations for approaching OSS. However, intrinsic motivations may play a role as well.
[91]	Motivations for (reduced costs, freedom from vendors) and extent of adoption of OSS products (used by all respondents).
[93]	OSS adoption is complex. Develops a framework for adoption of OSS products consisting of an external environment, individual factors, and organizational and technological contexts.
[94]	Develops a four level model for resource allocation to OSS, showing varying dedication to OSS.
[99]	Illustrates four different ways of adopting OSS.
[98]	Shows that about 50% of the Norwegian software industry integrates OSS into their products, that about 16% participate in the development of OSS products controlled by someone else, and that about 5% provides their own OSS products.
[127]	OSS ideas may be adopted in different ways. Uptake of OSS is high. The interest in OSS is beyond simply using the LAMP stack.
[139]	Discusses several benefits and drawbacks of OSS.
[141]	About 50% of the software companies in the Finnish survey use OSS products in their business.
[153]	Organizations are only willing to collaborate if they get financial gains. Discusses motivations for approaching OSS.
[172]	Analyzes different factors that may influence the adoption of OSS products in software development.
[188]	Motivations for using OSS server software: lower cost, higher reliability, availability of external support. Source code only interesting for integrators.
[189]	Organizations are pragmatic in the adoption of OSS. The influence of ideology could matter in smaller organizations.
[204]	Discusses hybrid (open and proprietary) business strategies in large companies.

4.2.2. Deploying OSS Products

In total, 10 research papers and six experience reports focus on the deployment of OSS products. The majority of these are short papers (six pages or less) discussing the experience from one context with one or a few OSS products. Most of these papers illustrate successful use of one or more OSS products (see Table 8 for an overview of the research papers).

Table 8: Focus and findings from the empirical research papers on deploying OSS

ID	Topic
[91]	Respondents have not performed cost analysis.
[37]	Identifies seven critical success factors for migrating to an OSS desktop: financial motivation, management support, user awareness, planning planning, analysis and testing, training, pilots, and support.
[62]	Overcoming barriers for migration of the desktop environment. Migration of the desktop could be hard work. Stakeholders’ attitude may vary and change over time.
[148]	Identifies several enablers and inhibitors of deployment of OSS products
[161]	Monitors the introduction of OpenOffice.org.
[159]	Monitors the use of OpenOffice.org. Illustrates significant adoption.
[160]	Illustrate the public sectors’ strong commitment to proprietary document formats, despite using OpenOffice.org
[164]	Monitors the introduction of OpenOffice.org. User’s lack of experience is a barrier.
[187]	OpenOffice.org migration is possible, but not problem free. Issues often specific to environment. Training is important. No formal cost analysis was performed.
[190]	Benefits of OSS depend on context. Comparing “OSS” directly with “proprietary software” in general is futile. Respondents did not perform cost analysis. This should be done.

While the deployment of OSS can give cost savings, migrating from one technological platform to another is hard work which often includes customization, adaptation, integration, and testing [62, 187].

The general costs related to such a migration are unclear [62, 187], and there are very few studies showing complete calculations of the true costs and savings of (1) introducing OSS products into organizations, and (2) keeping the OSS products operational over a longer period of time. One paper reports cost savings from an OSS migration project at Beaumont Hospital [81], but it is published just after the initial stage of the project is finished.

Despite this lack of clarity, many organizations seem to be blinded by the perceived advantages of OSS and have therefore adopted it without performing any cost-benefit evaluations in their own context [91, 187, 190]. The adoption of OSS is furthermore frequently bottom-up, in the sense that it is introduced by engineers rather than strategic top-level decisions [188].

4.2.3. Using OSS CASE Tools

Despite having a close relation to the CASE research field, only seven experience reports discuss the use of OSS CASE tools in the context of organizations. Given the number of OSS CASE tools available, it is surprising that the use of such tools has not been studied in any empirical research papers.

4.2.4. Integrating OSS Software into Software Systems

Only six empirical research papers discuss issues related to integration of OSS components. There are, however, as many as 16 experience reports that briefly describe the integration of some OSS components.

Table 9: Focus and findings from the empirical research papers on integrating OSS

ID	Topic
[8]	Software organizations can increase productivity and quality through integrating systematic reuse of OSS components. OSS reuse does not require any special skills and experience other than software reuse in general.
[43]	Major cost with OSS is learning and understanding new components. Local knowledge and compliance to requirements were the most decisive factors in choosing components. A high number of components needed fixes or modifications.
[106]	There is a need for human computer interaction experts in the OSS context. Reuse of OSS enables them to spend more time on user interface. It is challenging to decide what to contribute because of licenses and patents, and as user interface code can be considered a competitive advantage.
[123]	There are many similarities in using OSS and COTS. Source code is seldom used. Hard to assess providers' (community) reputation.
[130, 186]	Identifies challenges related to modifying OSS components, and strategies (contributing, snapshot, forking, and initiating a new OSS project) for dealing with these modifications.

The integration of OSS components is one of the most popular ways of adopting OSS, in particular in the software industry. From a sample of 146 OSS firms, 69.5% reported that they had adapted OSS to customer needs [27]. In another sample of 769 companies 33% “provide solutions which are based on OSS” [29]. Moreover, 48% of 62 software companies use OSS in their business [141], and in a sample of 569 software companies, 46.8% integrate OSS in their software systems [98]. These software systems represent a great variety of application areas from all major vertical sectors [98]. Finally, Nokia claims that as much as 75% of the software architecture for its Internet tablet consists of OSS [107].

Ajila et al. [8] and Li et al. [123] claim that the similarities between reuse of OSS and COTS (commercial-off-the-shelf) are significant. No particular skills or experience is needed to integrate OSS components into another product [8].

Li et al. [123] claim that the source code of OSS components is seldom modified. Still, Chen et al. [43] report that it is necessary to modify or adapt many of the OSS products. Mannaert and Ven [130, 186] and Iivari et al. [106] discuss some of the challenges related to modifying and extending an OSS product. These challenges are somewhat special for OSS since any developer has access to the product’s source code and since it is possible to return the

modifications to the provider of the product. Finally, Ven and Mannaert [130, 186] present four strategies for dealing with these modifications: contributing to the OSS community, relying on a snapshot of the code base, forking the OSS product, or releasing modifications to the OSS product as a new OSS product.

Some of the most significant challenges of OSS-based projects are, according to Chen et al. and Li et al. the cost related to learning and understanding the OSS components, and estimating the time it takes to integrate them [43, 123]. This could be the reason why local expertise is so important in the decision making when selecting an OSS component [43].

4.2.5. Participating in the Development of OSS

As stated previously, software companies and software intensive organizations have started to base their existence on the use of OSS components developed by OSS communities [98]. As many as 16 empirical research papers and six experience reports provide evidence on organizations' participation in OSS communities. These organizations are thereby playing an increasingly important role in many OSS communities.

Most organizations seem to have rather limited contributions to the OSS communities [33, 43, 91, 98]. The most common way of participation is being an active user that reports occasional bugs to the community [43, 98, 99]. Only one of 32 respondents from a sample of tertiary education institutions had participated actively by writing code, while 14 had contributed to an OSS community through reporting bugs [91]. Furthermore, Bonaccorsi and Rossi [32] found that 46.2% of companies using OSS components have not joined any OSS projects, whilst 38.5% have joined five or fewer.

While the majority of organizations contribute on a rather limited level, the total body of all organizations contributes substantially to various OSS products [28, 158]. First, Bonaccorsi et al. [28] report that organizations participated in 97 of the 300 most active projects on SourceForge. Second, in a sample of community controlled OSS projects, paid developers contributed almost 50% of the code [200]. Moreover, the number of organizations contributing to OSS seems to be increasing, for instance the number of companies participating in the Debian GNU/Linux community has increased from 200 (in 1998) to 1500 (in 2005) [158]. Organizations within the Linux Kernel community are central to the community and do in fact have significant influence on it [2]. Organizations may have positioned themselves there because they believe they have to be involved to influence the community [51].

Table 10: Focus and findings from the empirical research papers on participating in the development of OSS

ID	Topic
[2]	Influence in the Linux Kernel community is centered to a small group. Organizations have a significant influence on this group.
[20]	Shows different types of employee participation. OSS projects are not uniform and homogeneous. Could be challenging to get contributions accepted.
[31]	Many companies participate in OSS communities. Companies mainly adapt the OSS product to their needs.
[28]	Companies coordinate, develop code for, or provide libraries to one third of the 300 most active Sourceforge projects. Projects where companies are involved are larger.
[50]	Organizations make use of OSS communities through accessing, aligning, and assimilating the communities.
[51]	Companies strategically sponsor individuals to influence OSS communities. Firms believe they need someone on the inside to influence the communities.
[49]	Organizations may have a symbiotic, commensalistic, or parasitic relationship to OSS communities.
[70]	Shows how a network of companies around an OSS product can collaborate to deliver a “whole product”.
[101]	Different companies have different reasons for contributing. Code sharing is very heterogeneous. Companies share more than they have to.
[106]	There is a need for human computer interaction experts in the OSS context. Reuse of OSS enables them to spend more time on user interface. It is challenging to decide what to contribute because of licenses and patents, and as user interface code can be considered a competitive advantage.
[129]	Software developers who participate in OSS communities spend close to 50% of this time through their work.
[130, 186]	Identifies challenges related to modifying OSS components, and strategies (contributing, snapshot, forking, and initiating a new OSS project) for dealing with these modifications.
[132]	Time based releases seem to boost development in community-company collaborations.
[158]	A large number of companies give significant code contributions to Debian.
[200]	OSS providers must write most (+90%) of the code themselves. About 50% of the code in community controlled OSS projects was written by paid developers.

Nevertheless, companies are clearly becoming a very important part of the OSS community.

To make sure that an OSS product prospers, and to make sure that modifications to the product's code are maintained, an organization may decide to contribute to the product's community. However, Ven and Mannaert [186] mention several barriers for contributing to an OSS product. The organization must first of all spend resources on getting to know the community. Then they may have modifications that are very specific to their own organization or which influence several OSS projects. If the patch is accepted into an OSS product, the organization may have to spend resources maintaining it as part of the OSS project. Deciding not to contribute can also be risky as one may be forced to maintain a parallel copy of the product. In addition, Iivari et al. [106] illustrate how the fear of losing the competitive edge, and difficulties with licenses and patents, can prevent an organization from contributing.

Dahlander and Magnusson [49, 50] discuss company-community relationships and how organizations can benefit from communities. In [49] they identify three types of organization-community relationships:

- Symbiotic: Both the community and the organization benefit from the relationship.
- Commensalistic: The organization benefits from the relationship but the community is not affected.
- Parasitic: The organization benefits from the relationship but at the same time it damages the community.

Later Dahlander and Magnusson [50] show how organizations use three strategies to benefit from OSS communities. First, organizations access existing communities or start their own communities. Second, they align their strategies with the community. Third, they assimilate the community through dedicating resources to evaluating contributions from the community or by contributing non-strategic code to the community.

4.2.6. Providing OSS Products

Many organizations have over the last years released their software as OSS. In total, 15 empirical research papers and 19 experience reports show how these organizations have developed and provided their own OSS products.

In a sample of 368 software companies, 5% said they provided their own OSS products [98]. In a sample of 134 software products, developed by 70

Table 11: Focus and findings from the empirical research papers on providing OSS products

ID	Topic
[7]	Shows a shift from OSS as individual to OSS as a community of organizations. Explores the opportunities organizations have related to community collaboration.
[17]	Presents three cases with robot (hardware) vendors that also provide OSS software for their robots.
[49]	Organizations may have symbiotic, commensalistic, or parasitic relationship to OSS communities.
[50]	Organizations make use of OSS communities through accessing, aligning, and assimilating the communities.
[52]	Analyses the transfer of OSS developed in academia to commercial products.
[82]	Companies with many patents will more likely release OSS. Companies with many software trademarks are less likely to release OSS. Companies with many hardware trademarks are more likely to release OSS.
[110]	Organizations must balance leadership vs. too strict control.
[126]	Of 134 products from 70 Italian companies, 27 are released as OSS. These products are considered at least as innovative as the proprietary ones.
[132]	Time based releases seem to boost development in community-company collaborations.
[144]	Provides recommendation for succeeding with an OSS product: Improve product and documentation, listen to the community, make it easy to download and install it.
[200]	OSS providers must write most (+90%) of the code themselves. About 50% of the code in community controlled OSS projects was written by paid developers.
[202]	Illustrates the evolution of JBoss' business model.
[201]	Discusses "second generation" OSS business models and presents some data from four OSS providers.
[204]	Discusses hybrid (open and proprietary) business strategies in large companies.
[206]	Explores some challenges related to legitimizing the use of an OSS business model.

Italian companies, 27 (20%) products were released as OSS [126]. Furthermore, 36 (12%) of the 300 most active projects on SourceForge were founded by companies [28]. Moreover, we observe that Microsoft [133], research institutions [52], and other companies in several domains [19, 34, 104, 202] regularly release new OSS products.

While some provide OSS products to attract and benefit from a community, others merely release OSS to attract attention or to disseminate their software or research results. OSS providers perceive benefits like simpler dissemination of their products, reduced marketing costs, simpler recruitment of new employees, and community contributions in form of bug reports, bug fixes, feature requests, and added functionality [7, 14, 95, 99, 100, 197, 200, 201, 204]. Most of these advantages are related to actually having a community around the product.

Providing an OSS product is described as a global sourcing strategy where "commercial companies and open source communities collaborate on development of software" [7, page 385]. Furthermore, Ågerfalk and Fitzgerald [7] relate their research with offshoring and outsourcing of software develop-

ment. This suggests that there is some division of labor involved in providing an OSS product. However, based on the observation that the OSS providers do most of the work and write most of the code themselves, Wasserman and Capra [200] claim that OSS is primarily a distribution model and not a development model for organizations.

Other papers report challenges related to similar difficulties such as attracting contributors, involving people at the right time, establishing a common infrastructure and so on [23, 26, 87, 109, 156]. Despite reporting different problems, few focus on identifying and solving these challenges. Another study presents experiences related to establishing a community and how limited continuity and a too strong focus on one stakeholder made it difficult to build a vivid community [109]. The provider must also consider which code to release as OSS and which to retain under proprietary licenses. Providing OSS is in other words no free lunch, and simply making the code available is not enough [100].

One of the exceptions that tries to solve some of these challenges identifies the obligations and expectations an OSS provider has to its community and vice versa [7]. Ågerfalk and Fitzgerald [7] moreover show that the provider and his community have different perceptions of the extent to which they fulfill these obligations. Being an OSS provider is not a static business model. OSS providers have to develop and adapt their business model according to the needs of their customers and communities [144, 202].

4.2.7. Using OSS Development Practices

Table 12: Focus and findings from the empirical research papers on using OSS development practices

ID	Topic
[125]	Illustrates the use of a company internal Sourceforge-like portal.
[135]	Adoption of OSS practices may be a way to standardize development processes. Transparency has both advantages and disadvantages. The OSS phenomenon is adapted to the organizations when it is adopted.

While there is quite a lot of research on specific development processes in OSS communities e.g. [167], there is little research on using these processes and practices inside organizations. We identified two empirical research papers and 11 experience reports that discuss the use of such practices.

Several companies have adopted “OSS practices” internally or within a consortium of partners. Common to the cases reported in the literature is

that they are large companies with technologies that are reused in a large-scale, distributed development environment. While the use of “OSS practices” within these large companies shares commonalities, they are labeled somewhat differently e.g. “Progressive Open Source” [60, 61, 135], “inner source” [125, 203], or “Corporate Source” [96, 97].

These attempts of adopting “OSS practices” may have slightly different purposes. First, one may want to improve the collaboration between the people responsible for a core platform and the people reusing it throughout the company [96, 97, 203]. Second, it is possible to increase the visibility of reusable software components by providing them through a common platform, much like an internal SourceForge [60, 61, 125]. Third, to increase transparency and standardize diverse development practices, an organization may create a common development platform as a vehicle for collaboration [61].

The introduction of “OSS practices” is however more a social, rather than a technical change [61]. While the changes could provide benefits for the organization, it may also be painful for the individual and the company to change existing work practices [135]. For instance the experience (or lack of) with OSS could influence the adoption of new “OSS practices” [179], and when adopting such practices in a commercial setting they are shaped to the organization [135].

5. Discussion

In this section we will discuss each of the research questions in the light of the findings from the literature review. Then, we discuss opportunities for future research and finally, possible limitations of this study.

5.1. RQ1: Adoption of OSS in Organizations

In Section 2.5, we asked the following research question: *In what ways are software-intensive organizations adopting OSS?*

The short answer to this question is that there exist several ways of adopting OSS in software-intensive organizations. This literature review provides evidence that such organizations approach OSS in different ways. With a focus on software development, we identified six ways in which organizations adopt OSS (see Section 2.3).

Each of these ways of adopting OSS offers different benefits and challenges. For instance, while access to source code is not that important to

organizations that only deploy OSS products [81], it could be a significant advantage for OSS integrators [139]. An organization providing an OSS product may get feedback and code contributions from their community, but at the same time they have to deal with a (potentially) large number of stakeholders [109, 200]. However, relating to a community is not relevant in the same way for an organization that simply integrates OSS components into their own products.

We furthermore see variations in the motivations that organizations have for adopting OSS, the context these organizations are in, the resources they have, and so on. While OSS seems to be an option in almost all kinds of settings, we agree with Glynn et al. [93] and Melian and Mähring [135] in that OSS needs to be understood in the organization-specific situation it is adopted. This is particularly important as OSS adoption may also involve organizational changes [81, 135]. According to Glynn et al., adoption of OSS is a complex situation consisting of an internal and an external environment, individual factors, and a technological context [93].

5.1.1. The Classification Framework: Inter-dependencies and Internal Differences

It must be emphasized that there are interdependencies between the different ways of adopting OSS, and that an organization may approach OSS in several ways at the same time. These inter-dependencies are further complicated as organizations may evolve their approach to OSS over time [39]. Hence, the categories in our framework are not mutually exclusive.

There are in particular a few categories that are closely related. First, organizations that participate in the development of an OSS product are most likely integrating this product into one of their own systems (see e.g. [107, 130]). Second, organizations adopting OSS development practices are also frequently using OSS CASE tools to facilitate the adoption of these practices, e.g. [125, 203]. Tools like revision control systems, mailing lists, wikis, build-environments, and documentation systems shape the development process and enable the introduction of practices like code sharing, increased transparency, having a core team controlling the core of the product, peer review of code, and so on [87, 128, 131].

Next, the difference between a few of the categories is, as mentioned in Section 2.3, one of degrees rather than orthogonality. Grand et al. [94] have similar observations, in that the dedication to OSS is a matter of degree, or in other words a matter of allocating resources. There are overlapping

areas between deploying OSS products and integrating OSS products into a system [3, 4], and between participating in the development of an OSS and providing an OSS product [7, 48]. For example, different organizations may use the same OSS product quite differently. In one case, a sample of organizations simply deployed Linux on their servers [188], while others extended it, integrated it into their own products, and participated in the development of it [101].

There are also internal differences within each of the ways of adopting OSS. Even though different organizations provide OSS products, they have different motivations, resources, and success. Where Bleek et al. [23] describe a public project that is struggling, Watson et al. [202] report the (successful) story of JBoss. Li et al. [124] show that the reuse of OSS components is adapted to an organization's current development process. There are also differences in how organizations participate in the development of OSS products. While some organizations donate large amounts of source code or actively participate in the development of the product [28, 101], others provide more modest contributions like occasional bug reports or forums posts [43, 98].

We observe situations very much alike these in other cases as well. Despite approaching OSS in similar manners, there are clear differences between the various organizations. The classification framework must therefore be understood and used as a tool for identifying and discussing the opportunities and challenges different organizations may find when adopting OSS. It is not an attempt to completely identify all minor variances in how organizations adopt OSS.

Nevertheless, the categories are still valuable when discussing an organization's adoption of OSS. The benefits and challenges related to a specific organization's approach to OSS should be discussed as a combination of the different ways of adopting OSS in the framework. Hence, we recommend placing an organization in all the categories it would fit into, and investigating the specific challenges related to each of the categories in the framework.

5.1.2. *Implications*

Practitioners need to be aware that there is not just one correct way of adopting OSS. OSS rather offers several opportunities that each have their unique benefits and drawbacks. Each organization should therefore evaluate the implications of approaching OSS in their own context [189]. This is increasingly important when we see that many organizations adopt OSS

without knowing whether they will benefit from it and without having a clear strategy behind this adoption [91, 187, 188, 190].

Practitioners should also be aware that there are most likely others that have adopted OSS in a similar context. Here, we show that OSS is a viable option for many organizations and we provide several references to a variety of successful cases of OSS adoption. These references could be used as a starting point for an organization’s adoption of OSS.

Researchers ought to avoid treating OSS and the adoption of OSS as one homogeneous phenomenon. We should acknowledge the individual contexts in which OSS is adopted, precisely describe how the organizations we study approach OSS, and carefully consider how this adoption influences our findings.

5.2. RQ2: Focus of the Research Literature

The 59 empirical research papers and the 53 experience reports included in this review cover a very large span of topics but have not had any particular focus. The research that is specific to how organizations adopt OSS covers topics from OpenOffice.org migration projects, through code contributions to existing OSS communities, to using “OSS development practices”. Both within and between these topics there are several unexplored gaps.

Apart from the studies that investigate the motivations for adopting OSS products and the migration to OpenOffice.org, there are few overlapping studies discussing closely related topics. Even though there are a few exceptions like [27, 31, 32, 33], [81, 93], [130, 186], and [60, 61, 135], the majority of the papers come from rather fragmented studies. Few studies have been reported in more than one publication and almost no publications report follow-ups or continuations of earlier research.

This lack of focus may be caused by the relatively recent birth of OSS in organizations as a research area. The many variations in how organizations adopt OSS may also contribute to diversify the research. Although similar diversity is seen in research on both software engineering [92] and information systems [192], *OSS researchers* could benefit from intensifying their efforts on a few common problems, rather than exploring an increasingly larger number of issues.

5.3. RQ3: Limitations of Existing Empirical Research

There are relatively few empirical publications on OSS in organizations, and the quality of published work is not good enough. Much of the published

research lacks relevance and a clear focus, and does not draw enough support from related literature. These observations are not particular to research on OSS. For instance, Kitchenham et al. [117], Vessey et al. [192], and Zelkowitz and Wallace [214] observe a lack of relevant empirical research of high quality within both the software engineering and information systems fields. Finally, we would also like to see more research from outside Europe and the USA.

5.3.1. *Little Empirical Research*

Even though OSS is changing how software is developed, acquired, used, and commercialized, relatively few empirical research papers on OSS in organizations are being published. The number of such papers is particularly low in quality software engineering journals (see Appendix A).

Only 86 (12.8%) of the 674 publications relevant to OSS (S2) and five (4.5%) of the 112 papers included in the fifth stage (S5) are published in quality software engineering journals like IST, JSS, EMSE, Software Practice and Experience, Software Process: Improvement and Practice, TOSEM, or TSE. On the other hand, 380 (56.47%) of the papers related to OSS and 75 (67.0%) of the 112 publications included in the Stage 5 are published in the IEEE Software magazine, or through the International Conference on Open Source Systems and the ICSE Workshop on Open Source Software Engineering.

5.3.2. *Limited Quality*

Many of the empirical research papers achieved a decent score in the quality assessment. Despite this, almost half of the publications we identified were experience reports and many of the empirical research papers were troubled by missing information, low rigor, limited validity, and unclear contributions. These observations are in line with previous work e.g. [66, 71, 92, 180, 213, 214], in the sense that much of the published research has limitations when it comes to planning, execution, and reporting.

5.3.3. *Limited Relevance*

As many as 19 of the 59 empirical research papers study topics that are not directly related to how organizations adopt OSS. While these papers are valuable for understanding the OSS phenomenon, they are not particularly relevant to the specific problems practitioners face every day, and provide little concrete advice to practitioners.

5.3.4. Lack of Complex and Longitudinal Studies

The research on OSS in organizations is as mentioned a fairly young field. The immaturity may explain why there are few longitudinal studies and few studies looking at complex issues beyond the deployment or integration of a single OSS product. However, Höfer and Tichy [103] made similar observations concerning longitudinal studies of software engineering.

5.3.5. Does not Reflect Context

Parts of the literature neither describe nor reflect on the actual context in which OSS is adopted. This is seen when papers fail to describe how the studied organizations adopt OSS. It is also seen when researchers study samples of organizations without distinguishing between these organizations' different approaches to OSS and try to make the generalization that to all organizations "adopting OSS". This lack of context is problematic since, according to Basili et al. [18], every software development/maintenance environment is different.

5.3.6. Does not Benefit Fully from Related Research Fields

OSS and OSS development has been described as revolutionary and something totally different from software engineering [30, 154, 167]. The history of OSS has furthermore been characterized by contrasts like OSS vs. proprietary or closed source software [149], OSS vs. free software [175], the cathedral vs. the bazaar [154], copyleft vs. copyright [55], OSS development vs. software engineering [59], and so on. These contrasts and the perception that OSS is something different, have contributed to creating a gap between OSS and other research areas.

We agree with Fitzgerald [78] and Fuggetta [84] in questioning whether these differences are significant. Moreover, we find evidence that much of the research on OSS in organizations has in fact profound similarities with other research areas. The review revealed that organizations deploying OSS faced the same challenges as with adoption of any other technology [37, 81, 148, 187], and that these issues were often organizational rather than technical [135, 139]. The adoption of OSS seems to be more depending on the organization adopting it and the situation in which it is adopted, than on the technology being released as OSS. Next, we saw that integrating OSS components was very much the same as integrating COTS components [8, 123]. Finally, many of the advantages of OSS (like reduced development effort, increased quality, and so on) are really advantages of software re-use

[137, 193]. OSS products can clearly give these benefits, but this is not necessarily because these products are licensed as OSS.

Despite these similarities, parts of the literature have still treated OSS as something new and quite different from other information technologies and methods for software development. Treating a novel research area as something totally different from existing areas is not a new phenomenon as “[t]here is a tendency for IS researchers [as well] to treat new technologies as virgin or greenfield, thereby acting on the belief that prior theories or models are not appropriate” Vessey et al. [192, page 169].

Some researchers are looking to related research and theories to explain OSS phenomena e.g. adoption of IT innovations [93], psychological contract theory [7], and business models and business networks [70]. However, many researchers present an introverted view of OSS, and when treating OSS as something totally unique, they fail to draw valuable support from related literature.

This literature review focuses on empirical studies from the OSS and software engineering literature. By widening its scope it would have encompassed publications that extended literature from a broader spectrum of areas. Nevertheless, we find that many researchers should draw more support from related research fields both when identifying research questions and when discussing their findings.

5.4. Directions for Future Research

Even though the research so far has some limitations there are several opportunities for further work. Based on the literature review and the answers to RQ 2 and 3, we give some recommendations for this work.

5.4.1. General Recommendations

Maturing the research field on OSS in organizations and dealing with some of its limitations may be done through four main steps:

1. Focus research on topics that are relevant to how organizations approach OSS
2. Strive to increase the rigor of the empirical studies
3. Conduct more longitudinal, in-depth studies
4. Align our research with related research fields

There are several unexplored issues in relation to the “adoption of OSS” in general. However, we agree with for instance Charters et al. [42] in that

researchers need to pay more attention to issues that are interesting to practitioners. Hence, we recommend focusing on topics related to the ways in which organizations actually approach OSS, and issues that could benefit practitioners, rather than general “adoption issues”. Researchers and practitioners should increasingly collaborate to define a common research agenda and study research questions that matter to practitioners. These research questions should be answered through several related studies from different contexts.

The overall rigor of the studies performed on OSS, both within organizations and in general, is furthermore not good enough. Consequently, we should strive to do better work and to present this work in more detail [180]. In particular, we agree with Kitchenham et al. [118] in that the context of the organizations being studied should be given much more attention.

We observed that few of the studies were longitudinal and that few publications focused on providing in-depth details from one or a few organizations. To really understand the profound consequences of approaching OSS, we believe there is a need for both more longitudinal and in-depth case studies.

Finally, we found evidence that OSS is not that different from other information technologies. *OSS researchers* should therefore increasingly rely on research and theories from related fields (see Section 2.4). *Software engineering and information systems researchers* should see OSS as an opportunity to investigate general software engineering and information systems research challenges.

5.4.2. Topics for Future Research

In this section we suggest topics for further research for each of the approaches to OSS. The topics discussed below may work as an initial starting point for discussing and staking out the direction of the research on OSS in organizations. We would in particular recommend investigating two issues: (1) topics related to integration of OSS components and (2) topics concerning participation in organization-community or inter-organizational OSS collaborations. We find these issues important because integration of OSS components concerns most software-intensive organizations [98] and because participation in collaborative software development is increasing [7, 185]. The research could focus on identifying the characteristics of successful approaches to OSS, the challenges these organizations met, and how they solved them.

Deploying OSS: Many claim that reducing costs is one of the advantages

of deploying OSS server software, infrastructure, and applications. However, a recent study by Fitzgerald [80] is one of few studies with a longitudinal view on deployed OSS products. This highlights a need for more studies on:

- What are long-term costs and consequences of deploying and keeping OSS products operational?

Using OSS CASE tools: The research on OSS CASE tools has been very limited. However, Wicks and Dewar propose a new agenda for research on tool integration, requesting a more business-oriented approach to future research [207]. The use and development of OSS CASE tools and research on such tools could easily fit into this new agenda. Robbins provides an extensive overview of OSS tools for development, and claims that CASE research has a lot to learn from OSS [157]. OSS should furthermore be particularly interesting to academia since they have access to professional state-of-the-art tools and the tools' source code. This enables them to extend existing tools and test new ideas in collaboration with each other.

Increased participation in OSS projects, increased collaboration between organizations, and increased use of OSS practices will most likely require improved collaborative development tools. Hence, there is a potential for research on:

- What kinds of tools are needed for collaborative software development across organizational and community borders?
- How do organizations collaborate using such software development tools?

Integrating OSS components:

Navigating through the amounts of OSS components and related information available across the Internet is a significant challenge [143]. However, the information offered over the Internet through OSS communities, web forums, and so on, constitutes at the same time a valuable resource. Due to the easy access to reusable software components, we see that software systems are constantly growing. Software developers are integrating an increasingly larger number of OSS and commercial components into their products. In doing so they have to relate, adapt, and possibly contribute to a large number of providers. Therefore, we believe research could focus on the following questions:

- How may organizations most efficiently navigate through available information and select OSS components?

- How may organizations benefit from OSS communities and the resources available over the Internet?
- How can organizations maintain and secure the sustainability of software systems consisting of components from a variety of providers?

While there are a few studies outside the scope of this review focusing on software selection [46, 56, 105, 184] and knowledge sharing within OSS communities [119, 173, 195], none of these are directed towards studying actual practice in organizations. A few studies have started to look at some of the challenges in the borderlands between integrating an OSS component and contributing to the development of it [106, 130, 186], but further research is needed to solve the maintenance challenges facing developers who integrate a large number of components into their products.

Participating in OSS communities: To enable organizations to reap benefits from their participation in OSS communities, the research community should dedicate much more attention to questions concerning this [48, 165]. While there are a few examples [50, 101, 176, 186], more work is needed to aid organizations in participating in communities and collaborating with other organizations through collaboratively working on OSS products [7] and to solve questions like:

- When, how, and with what should organizations participate in the development of OSS products controlled by others (including inter-organizational collaborations)?
- How can companies (effectively) allow products to be partly OSS and partly commercial products?

Providing OSS products: Succeeding at providing an OSS product is not necessarily easy as there are challenges related to collaborating with a community, like attracting and relating to contributors, requirements engineering from a community, balancing focus on community and paying customers, and so on [109, 200]. We hope to see more research on these topics like e.g. [7, 205] on the following topics:

- How are OSS providers able to attract and sustain a community?
- What are the success criteria for incorporating contributions (requirements, code, bug reports/fixes, etc.) from a community?

Using OSS practices: It is more and more difficult to talk about “OSS practices” as the practices used in OSS communities are heterogeneous, and as organizations are increasingly getting involved in, and influenced by, the development of OSS. Nevertheless, there are opportunities for further research on the use of development practices for distributed software development. OSS development in large communities and in and between organizations, are areas where researchers could have an impact on practice. OSS research has so far focused mainly on processes in communities of volunteers [167], but some of this research could turn its focus towards the application of their findings within organizations and questions like:

- How can development practices from OSS communities be adopted within organizations?
- How may organizations successfully collaborate through community- or consortium-based software development?

5.5. Limitations of this Study

Even though this systematic literature review has been supported by a pre-defined study protocol, explorative pilot testing of each of its stages, and continuous interaction between the authors, it has some limitations.

5.5.1. Completeness of the Selected Set of Publications

By basing the review on a clearly defined set of publications (see Section 3.1), we excluded certain types of publications, work published through other channels or outside the defined time frame. We can therefore not claim to have included all relevant publications. However, we based the review on an extensive set of publications from core software engineering and OSS publication channels. The most relevant publications should therefore be included.

5.5.2. Different Search and Data Extraction Facilities

The search and data extraction facilities provided by the various digital publication databases are different and not necessarily developed with systematic literature reviews in mind [36, 68]. This posed two challenges.

First, the data provided by digital libraries is not always reliable. For instance, we found several publications that were to be printed in 2009, registered as published in 2008. This means that we may have included papers due for publication in 2009 as published in 2008.

Second, we wanted to cover as many publications as possible and decided to use full text search, wherever possible. However, the standard search in IEEE Xplore provides only a meta-data search. Full text search is possible, but we did not discover this until late in the review process. The use of full text search in IEEE Xplore would have implied that the number of database hits (Stage 2) in TSE, ICSE (1998-2005, 2007), ISESE (2002-2005), METRICS, ESEM (2007), and particularly IEEE Software and IEEE Computer would have been significantly higher.

Even though it was possible to do full text search in IEEE Xplore we decided against it because of several reasons. First, using a full text search reduces the precision of the search quite dramatically [58]. Our experience with the other databases confirms this, and a large number of the publications included in Stage 2 were not within the scope of this literature review. The 112 papers from Stage 5 constitute only 7.3% of the papers included in Stage 2. Therefore, Dieste and Padua [58] recommend searching through only abstracts and titles as a good searching strategy. Finally, we randomly reviewed a significant amount of the papers not included in the initial meta-data search. None of these publications would have been included in Stage 4 or 5 of this literature review.

5.5.3. Classification of Papers and Missing Information

In Stage 2 we classified the publications primarily based on their titles and abstracts. This is sometimes hard as many abstracts often omit relevant information [38]. As a consequence, Brereton et al. [36] recommend reviewing also the conclusions of the papers in addition to the titles and abstracts. However, acquiring the whole text for more than 1500 publications and reviewing them was not a viable option. Conducting a systematic literature review is already a very resource demanding exercise.

Reading the papers more closely (Stages 3-5) clearly increased the precision of the classification, but it could still be quite difficult to classify a paper due to the paper's lack of detail and because some papers discuss several topics. This was particularly relevant since much of the OSS research has had a focus on qualitative data. Kitchenham's guidelines for systematic literature reviews recommend contacting the authors of such papers to get the necessary details [115]. Even though this was not done, we achieved a good agreement in the classification process and solved any disagreement by reaching a consensus.

5.5.4. *Quality Assessment*

To ensure scientific rigor, we performed a quality assessment by assessing whether or not the publications covered the nine quality metrics defined in Section 3.2. This binary (yes/no) scale was in retrospect not sensitive enough and the quality assessment scores are therefore a bit inflated. Many of the publications barely mention issues related to some of the quality metrics, but were still evaluated to cover them.

Using a three-level scale like Kitchenham et al. [116] could have been a better option. However, this would have required a larger classification effort and even more subjective judgment. Given the amount of literature reviews concluding that many publications lack rigor e.g. [66, 71, 92, 180, 213, 214], spending even more time on detailed quality assessments would hardly have provided any new insight.

5.5.5. *Data extraction and author bias*

The most challenging part of the literature review is perhaps extracting relevant data or findings from the publications and writing a synopsis like this paper based on such data. Author bias in this process is a potential problem as the extracted data has primarily a qualitative nature and as we must prioritize what to include in the synopsis. To reduce this problem we used a pre-defined study protocol, piloted the various stages of the review, performed most stages individually, and had continuous discussions about the review process.

The systematic literature review is one of evidence based software engineering's key tools for integrating research with practice. One of the objectives of a systematic literature review is therefore creating guidelines for practitioners. This was difficult due to the many limitations of the OSS literature, and is something future research should try to achieve.

6. Conclusion

Unlike existing literature reviews on OSS, this review is a systematic literature review that focuses on OSS in organizations. We provide an extensive overview of the OSS literature, and together with Stol and Babar [180] we introduce systematic literature reviews to the OSS research arena.

Our results show that *organizations adopt OSS in distinctly different ways*. To better understand these ways of adopting OSS, we have provided a classification framework that shows that an organization may (1) deploy OSS

infrastructure and applications, (2) use OSS CASE tools, (3) integrate OSS components, (4) participate in external OSS communities and contribute to the development of OSS products that are controlled by someone else, (5) provide their own OSS products, and (6) use OSS development practices.

Known *success stories* from other organizations is an important factor that increases an organization’s confidence in OSS, according to Ågerfalk et al. [6] and Glynn et al. [93]. Here, we show that OSS provides organizations with several opportunities, that OSS is widely adopted, and that organizations make up a significant part of the OSS phenomenon. We furthermore provide references to a large number of success stories that could educate practitioners and make them feel more confident about adopting OSS.

We see that while the research community’s *attention to OSS in organizations has been limited*, it seems to be increasing. In addition, this literature review reveals that the research literature is rather fragmented and lacks a clear focus, rigor, and longitudinal studies. OSS researchers have furthermore not reflected well enough over how organizations adopt OSS and they have not benefited fully from related research fields.

The research on OSS in organizations has some identified limitations, but there are several *opportunities for future research*. We provide directions for this work, which apply not only to OSS researchers, but also to software engineering and information systems researchers, who want to study contexts in which OSS is developed and used.

From these results there are several *implications*. Researchers could use these contributions to find new research challenges and align their work with the work of others. They may also use the classification framework to position their work and to describe the context of the organizations they are studying. We furthermore advise researchers to put emphasis on how the studied organizations actually use OSS, and on problems that really matter to practitioners. Practitioners should be open to OSS and see that it offers several opportunities. However, they must first evaluate the implications of adopting OSS in their own context.

Acknowledgements

Claudia Ayala’s work was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme. We acknowledge Thomas Østerlie and Torgerir Dingsøy for their comments during the execution and writing phase of this work, and Sari Cunningham for her assistance with language and proofreading.

References

- [1] I. Aaen, J. Arent, L. Mathiassen, O. Ngwenyama, A Conceptual Map of Software Process Improvement, *Scandinavian Journal of Information Systems* 13 (2001) 123–146.
- [2] T. Aaltonen, J. Jokinen, Influence in the Linux Kernel Community, in: Feller et al. [77], pp. 203–208. doi:[10.1007/978-0-387-72486-7_16](https://doi.org/10.1007/978-0-387-72486-7_16).
- [3] P. Adams, C. Boldyreff, D. Nutter, S. Rank, Adaptive Reuse of Libre Software Systems for Supporting On-line Collaboration, in: Feller et al. [75], pp. 1–4. doi:[10.1145/1082983.1083259](https://doi.org/10.1145/1082983.1083259).
- [4] P. Adams, D. Nutter, S. Rank, C. Boldyreff, Using Open Source Tools to Support Collaboration within CALIBRE, in: Scotto and Succi [168], pp. 61–65.
- [5] W. Afzal, R. Torkar, R. Feldt, A systematic review of search-based testing for non-functional system properties, *Information and Software Technology* 51 (6) (2009) 957–976. doi:[10.1016/j.infsof.2008.12.005](https://doi.org/10.1016/j.infsof.2008.12.005).
- [6] P. J. Ågerfalk, A. Deverell, B. Fitzgerald, L. Morgan, Assessing the Role of Open Source Software in the European Secondary Software Sector: A Voice from Industry, in: Scotto and Succi [168], pp. 82–87.
- [7] P. J. Ågerfalk, B. Fitzgerald, Outsourcing to an Unknown Workforce: Exploring Outsourcing As a Global Sourcing Strategy, *MIS Quarterly* 32 (2) (2008) 385–409.
- [8] S. A. Ajila, D. Wu, Empirical study of the effects of open source adoption on software development economics, *Journal of Systems and Software* 80 (9) (2007) 1517–1529. doi:[10.1016/j.jss.2007.01.011](https://doi.org/10.1016/j.jss.2007.01.011).
- [9] J. Akkanen, H. Demeter, T. Eppel, Z. Ivánfi, J. Nurminen, P. Stenman, Reusing an open source application – practical experiences with a mobile CRM pilot, in: Feller et al. [77], pp. 217–222. doi:[10.1007/978-0-387-72486-7_18](https://doi.org/10.1007/978-0-387-72486-7_18).
- [10] C. A. Ardagna, E. Damiani, F. Frati, M. Montel, Using Open Source Middleware for Securing, e-Gov Applications, in: Scotto and Succi [168], pp. 172–178.
- [11] F. Attilio, P. Di Nunzio, F. Di Gregorio, A. Meo, A graphical installation system for the GNU/Linux Debian distribution, in: Damiani et al. [53], pp. 337–338. doi:[10.1007/0-387-34226-5_35](https://doi.org/10.1007/0-387-34226-5_35).
- [12] L. Augustin, D. Bressler, G. Smith, Accelerating Software Development through Collaboration, in: Tracz et al. [183], pp. 559–563.
- [13] N. Ayewah, D. Hovemeyer, J. D. Morgenthaler, J. Penix, W. Pugh, Using Static Analysis to Find Bugs, *IEEE Software* 25 (5) (2008) 22–29. doi:[10.1109/ms.2008.130](https://doi.org/10.1109/ms.2008.130).

- [14] J. Ayre, F. Gasperoni, A Successful Business Model for Free Software, in: Scotto and Succi [168], pp. 135–139.
- [15] C. Bac, O. Berger, V. Desbordes, B. Hamet, Why and how-to contribute to libre software when you integrate them into an in-house application?, in: Scotto and Succi [168], pp. 113–118.
- [16] M. Banzi, G. Bruno, G. Caire, To What Extent Does It Pay to Approach Open Source Software for a Big Telco Player? , in: Russo et al. [163], pp. 307–315. [doi:10.1007/978-0-387-09684-1_27](https://doi.org/10.1007/978-0-387-09684-1_27).
- [17] P. Barrera, G. Robles, J. M. Cañas, F. Martín, V. Matellán, Impact of Libre Software Tools and Methods in the Robotics Field, in: Feller et al. [75], pp. 1–6. [doi:10.1145/1083258.1083261](https://doi.org/10.1145/1083258.1083261).
- [18] V. R. Basili, R. W. Selby, D. H. Hutchens, Experimentation in Software Engineering, IEEE Transactions on Software Engineering 12 (7) (1986) 733–743.
- [19] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE: A software framework for developing multi-agent applications. Lessons learned, Information and Software Technology 50 (1-2) (2008) 10 – 21. [doi:10.1016/j.infsof.2007.10.008](https://doi.org/10.1016/j.infsof.2007.10.008).
- [20] E. Berdou, Insiders and outsiders: paid contributors and the dynamics of cooperation in community led F/OS projects, in: Damiani et al. [53], pp. 201–208. [doi:10.1007/0-387-34226-5_20](https://doi.org/10.1007/0-387-34226-5_20).
- [21] F. O. Bjørnson, T. Dingsøy, Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used, Information and Software Technology 50 (11) (2008) 1055–1068. [doi:10.1016/j.infsof.2008.03.006](https://doi.org/10.1016/j.infsof.2008.03.006).
- [22] W.-G. Bleek, M. Finck, Migrating a Development Project to Open Source Software Development, in: Feller et al. [74], pp. 9–13.
- [23] W.-G. Bleek, M. Finck, B. Pape, Towards an Open Source Development Process? Evaluating the Migration to an Open Source Project by Means of the Capability Maturity Model, in: Scotto and Succi [168], pp. 37–43.
- [24] P. Boccacci, V. Carrega, G. Doderò, Open source technologies for visually impaired people, in: Feller et al. [77], pp. 241–246. [doi:10.1007/978-0-387-72486-7_22](https://doi.org/10.1007/978-0-387-72486-7_22).
- [25] C. Boldyreff, K. Crowston, B. Lundell, A. I. Wasserman (Eds.), Proceedings of the 5th IFIP Working Group 2.13 International Conference on Open Source Systems (OSS2009) - Open Source Ecosystems: Diverse Communities, June 3-6, Skövde, Sweden, Vol. 299/2009 of IFIP International Federation for Information Processing, Springer, 2009. [doi:10.1007/978-3-642-02032-2](https://doi.org/10.1007/978-3-642-02032-2).

- [26] C. Boldyreff, D. Nutter, S. Rank, Communication and Conflict Issues in Coollaborative Software Research Projects, in: Feller et al. [74], pp. 14–17.
- [27] A. Bonaccorsi, S. Giannangeli, C. Rossi, Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry, *Management Science* 52 (7) (2006) 1085–1098. doi:10.1287/mnsc.1060.0547.
- [28] A. Bonaccorsi, D. Lorenzi, M. Merito, C. Rossi, Business Firms’ Engagement in Community Projects. Empirical Evidence and Further Developments of the Research, in: Capiluppi and Robles [41], pp. 1–5. doi:10.1109/floss.2007.3.
- [29] A. Bonaccorsi, L. Piscitello, M. Merito, C. Rossi, How is it possible to profit from innovation in the absence of any appropriability?, in: Damiani et al. [53], pp. 333–334. doi:10.1007/0-387-34226-5_33.
- [30] A. Bonaccorsi, C. Rossi, Why Open Source Software Can Succeed, *Research Policy* 32 (7) (2003) 1243–1258.
- [31] A. Bonaccorsi, C. Rossi, Contributing to OS Projects. A Comparison between Individual and Firms, in: Feller et al. [74], pp. 18–22.
- [32] A. Bonaccorsi, C. Rossi, Intrinsic motivations and profit-oriented firms. Do firms practise what they preach?, in: Scotto and Succi [168], pp. 241–245.
- [33] A. Bonaccorsi, C. Rossi, Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business, *Knowledge, Technology, and Policy* 18 (4) (2006) 40–64. doi:10.1007/s12130-006-1003-9.
- [34] G. Bortis, Experiences with Mirth: an open source health care integration engine, in: W. Schäfe, M. B. Dwyer, V. Gruhn (Eds.), *Proceedings of the 30th International Conference on Software Engineering (ICSE 2008)*, May 10th-18th, Leipzig, Germany, ACM, 2008, pp. 649–652. doi:10.1145/1368088.1368179.
- [35] P. Brereton, D. Budgen, Component-Based Systems: a Classification of Issues, *Computer* 33 (11) (2000) 54–62. doi:10.1109/2.881695.
- [36] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, *Journal of Systems and Software* 80 (4) (2007) 571–583. doi:10.1016/j.jss.2006.07.009.
- [37] D. Brink, L. Roos, J. Weller, J.-P. Van Belle, Critical Success Factors for Migrating to OSS-on-the-Desktop: Common Themes across Three South African Case, in: Damiani et al. [53], pp. 287–293. doi:10.1007/0-387-34226-5_29.

- [38] D. Budgen, B. A. Kitchenham, S. Charters, M. Turner, P. Brereton, S. Linkman, Presenting software engineering results using structured abstracts: a randomised experiment, *Empirical Software Engineering* 13 (4) (2008) 435–468. doi:[10.1007/s10664-008-9075-7](https://doi.org/10.1007/s10664-008-9075-7).
- [39] M. Campbell-Kelly, D. D. Garcia-Swartz, Pragmatism, not ideology: Historical perspectives on ibm’s adoption of open-source software, *Information Economics and Policy* 21 (3) (2009) 229 – 244. doi:[10.1016/j.infoecopol.2009.03.006](https://doi.org/10.1016/j.infoecopol.2009.03.006).
- [40] A. Capiluppi, P. Lago, M. Morisio, Evidences in the evolution of os projects through changelog analyses, in: Feller et al. [73].
- [41] A. Capiluppi, G. Robles (Eds.), *Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS 2007)*, May 21, Minneapolis, USA, IEEE Computer Society, Washington, USA, 2007.
- [42] S. Charters, D. Budgen, M. Turner, B. A. Kitchenham, P. Brereton, S. Linkman, Objectivity in Research: Challenges from the Evidence-Based Paradigm, in: C. Fidge (Ed.), *Proceedings of the Australian Software Engineering Conference (ASWEC’09)*, April 14th-17th, Gold Coast, Australia, IEEE Computer Society, 2009, pp. 73–80. doi:[10.1109/aswec.2009.25](https://doi.org/10.1109/aswec.2009.25).
- [43] W. Chen, J. Li, J. Ma, R. Conradi, J. Ji, C. Liu, An Empirical Study on Software Development with Open Source Components in the Chinese Software Industry, *Software Process: Improvement and Practice* 13 (1) (2008) 89–100. doi:[10.1002/spip.v13:1](https://doi.org/10.1002/spip.v13:1).
- [44] K. Crowston, J. Howison, The social structure of free and open source software development, *First Monday* 10 (2).
- [45] K. Crowston, Q. Li, K. Wei, U. Y. Eseryel, J. Howison, Self-organization of teams for free/libre open source software development, *Information and Software Technology* 49 (6) (2007) 564 – 575, *qualitative Software Engineering Research*. doi:[10.1016/j.infsof.2007.02.004](https://doi.org/10.1016/j.infsof.2007.02.004).
- [46] D. Cruz, T. Wieland, A. Ziegler, Evaluation Criteria for Free/Open Source Software Products Based on Project Analysis, *Software Process: Improvement and Practice* 11 (2) (2006) 107–122. doi:[10.1002/spip.257](https://doi.org/10.1002/spip.257).
- [47] P. Currión, C. de Silva, B. V. de Walle, Open source software for disaster management, *Communications of the ACM* 50 (3) (2007) 61–65. doi:[10.1145/1226736.1226768](https://doi.org/10.1145/1226736.1226768).
- [48] L. Dahlander, L. Frederiksen, F. Rullani, Online Communities and Open Innovation: Governance and Symbolic Value Creation , *Industry & Innovation* 15 (2) (2008) 115–123. doi:[10.1080/13662710801970076](https://doi.org/10.1080/13662710801970076).

- [49] L. Dahlander, M. G. Magnusson, Relationships between Open Source Software Companies and Communities: Observations from Nordic Firms, *Research Policy* 34 (4) (2005) 481–493. doi:[10.1016/j.respol.2005.02.003](https://doi.org/10.1016/j.respol.2005.02.003).
- [50] L. Dahlander, M. G. Magnusson, How do Firms Make Use of Open Source Communities?, *Long Range Planning* 41 (6) (2008) 629 – 649. doi:[10.1016/j.lrp.2008.09.003](https://doi.org/10.1016/j.lrp.2008.09.003).
- [51] L. Dahlander, M. W. Wallin, A Man on the Inside: Unlocking Communities as Complimentary Assets, *Research Policy* 35 (8) (2006) 1243–1259. doi:[10.1016/j.respol.2006.09.011](https://doi.org/10.1016/j.respol.2006.09.011).
- [52] J.-M. Dalle, G. Rousseau, Toward Collaborative Open-Source Technology Transfer, in: Feller et al. [74], pp. 34–42.
- [53] E. Damiani, B. Fitzgerald, W. Scacchi, M. Scotto (Eds.), Proceedings of the 2nd IFIP Working Group 2.13 International Conference on Open Source Software (OSS2006) - Open Source Systems, June 8-10, Como, Italy, Vol. 203/2006 of IFIP International Federation for Information Processing, Springer, 2006. doi:[10.1007/0-387-34226-5](https://doi.org/10.1007/0-387-34226-5).
- [54] E. Davini, E. Faggioni, G. Granatella, D. Tartari, M. Scotto, Open Source in Public Administration, a real example: OSS for e-government Observatories, in: Scotto and Succi [168], pp. 119–124.
- [55] P. B. de Laat, Copyright or copyleft?: An analysis of property regimes for software development, *Research Policy* 34 (10) (2005) 1511 – 1532. doi:[10.1016/j.respol.2005.07.003](https://doi.org/10.1016/j.respol.2005.07.003).
- [56] J.-C. Deprez, S. Alexandre, Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS, in: A. Jedlitschka, O. Salo (Eds.), Product-Focused Software Process Improvement Proceedings of the 9th International Conference on Product-Focused Software Process Improvement (PROFES 2008), June 23rd-25th, Monte Porzio Catone, Italy, Vol. 5089/2008 of Lecture Notes in Computer Science, Springer, 2008, pp. 189–203. doi:[10.1007/978-3-540-69566-0_17](https://doi.org/10.1007/978-3-540-69566-0_17).
- [57] M. Di Giacomo, MySQL: Lessons Learned on a Digital Library, *IEEE Software* 22 (3) (2005) 10–13. doi:[10.1109/ms.2005.71](https://doi.org/10.1109/ms.2005.71).
- [58] O. Dieste, A. G. Padua, Developing Search Strategies for Detecting Relevant Experiments for Systematic Reviews, in: Juristo et al. [112], pp. 215–224. doi:[10.1109/esem.2007.19](https://doi.org/10.1109/esem.2007.19).
- [59] T. T. Dinh-Trong, J. M. Bieman, The FreeBSD Project: A Replication Case Study of Open Source Development, *IEEE Transactions on Software Engineering* 31 (6) (2005) 481–494. doi:[10.1109/tse.2005.73](https://doi.org/10.1109/tse.2005.73).

- [60] J. Dinkelacker, P. K. Garg, Corporate Source: Applying Open Source Concepts to a Corporate Environment, in: J. Feller, B. Fitzgerald, A. van der Hoek (Eds.), Making Sense of the Bazaar: 1st Workshop on Open Source Software Engineering, 15 May, Toronto, Canada, IEE Software Proceedings, 2001, pp. 1–9.
- [61] J. Dinkelacker, P. K. Garg, R. Miller, D. Nelson, Progressive Open Source, in: Tracz et al. [183], pp. 177–184. doi:10.1145/581339.581363.
- [62] L. Dobusch, Migration Discourse Structures: Escaping Microsofts Desktop Path, in: Russo et al. [163], pp. 223–235. doi:10.1007/978-0-387-09684-1_18.
- [63] G. Doderò, K. Lupi, E. Piffero, Comparing macro development for personal productivity tools: an experience in validating accessibility of Talking Books, in: Damiani et al. [53], pp. 247–252. doi:10.1007/0-387-34226-5_24.
- [64] B. Donnellan, B. Fitzgerald, B. Lake, J. Sturdy, Implementing an Open Source Knowledge Base, IEEE Software 22 (6) (2005) 92–95. doi:10.1109/ms.2005.155.
- [65] T. Dybå, An Empirical Investigation of the Key Factors for Success in Software Process Improvement, IEEE Transactions on Software Engineering 31 (5) (2005) 410–424. doi:10.1109/tse.2005.53.
- [66] T. Dybå, T. Dingsøy, Empirical studies of agile software development: A systematic review, Information and Software Technology 50 (9-10) (2008) 833–859. doi:10.1016/j.infsof.2008.01.006.
- [67] T. Dybå, T. Dingsøy, Strength of Evidence in Systematic Reviews in Software Engineering, in: D. Rombach, S. Elbaum, J. Münch (Eds.), Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '08), October 9th-10th, Kaiserslautern, Germany, ACM, 2008, pp. 178–187. doi:10.1145/1414004.1414034.
- [68] T. Dybå, T. Dingsøy, G. K. Hanssen, Applying Systematic Reviews to Diverse Study Types: An Experience Report, in: Juristo et al. [112], pp. 225–234. doi:10.1109/ESEM.2007.58.
- [69] C. Ebert, Guest Editor’s Introduction: How Open Source Tools Can Benefit Industry, IEEE Software 26 (2) (2009) 50–51. doi:10.1109/ms.2009.38.
- [70] J. Feller, P. Finnegan, J. Hayes, Delivering the ‘Whole Product’: Business Model Impacts and Agility Challenges in a Network of Open Source Firms, Journal of Database Management 19 (2) (2008) 95–108.
- [71] J. Feller, P. Finnegan, D. Kelly, M. MacNamara, Developing Open Source Software: A Community-Based Analysis of Research, in: E. M. Trauth, D. Howcroft, T. Butler, B. Fitzgerald, J. I. DeGross (Eds.), Social Inclusion: Societal and Organizational Implications for Information Systems FIP TC8 WG 8.2 International Working Conference, 12th-15th July, Limerick, Ireland, Vol. 208 of IFIP

- International Federation for Information Processing, Springer, 2006, pp. 261–278. doi:10.1007/0-387-34588-4_18.
- [72] J. Feller, B. Fitzgerald, Understanding Open Source Software Development, Addison Wesley, 2002, ISBN :0-201-73496-6.
- [73] J. Feller, B. Fitzgerald, S. A. Hissam, K. R. Lakhani (Eds.), Taking Stock of the Bazaar: 3rd Workshop on Open Source Software Engineering (WOSSE 2003), 2003.
- [74] J. Feller, B. Fitzgerald, S. A. Hissam, K. R. Lakhani (Eds.), Collaboration, Conflict and Control Proceedings of the 4th Workshop on Open Source Software Engineering (WOSSE 2004), 2004.
- [75] J. Feller, B. Fitzgerald, S. A. Hissam, K. R. Lakhani, W. Scacchi (Eds.), Open Source Application Spaces: Proceedings of the Fifth Workshop on Open Source Software Engineering (WOSSE 2005), ACM, 2005.
- [76] J. Feller, B. Fitzgerald, K. R. Lakhani, S. A. Hissam (Eds.), Perspectives on Free and Open Source Software, The MIT Press, Cambridge, Massachusetts, 2005.
- [77] J. Feller, B. Fitzgerald, W. Scacchi, A. Sillitti (Eds.), Proceedings of the 3rd IFIP Working Group 2.13 International Conference on Open Source Software (OSS2007) - Open Source Development, Adoption and Innovation, June 11th-14th, Limerick, Ireland, Vol. 234/2007 of IFIP International Federation for Information Processing, Springer, 2007. doi:10.1007/978-0-387-72486-7.
- [78] B. Fitzgerald, Has Open Source Software a Future?, in: Feller et al. [76], pp. 93–106.
- [79] B. Fitzgerald, The Transformation of Open Source Software, MIS Quarterly 30 (3) (2006) 587–598.
- [80] B. Fitzgerald, Open Source Software Adoption: Anatomy of Success and Failure, International Journal of Open Source Software & Processes 1 (1) (2009) 1–23.
- [81] B. Fitzgerald, T. Kenny, Developing an Information Systems Infrastructure with Open Source Software, IEEE Software 21 (1) (2004) 50–55. doi:10.1109/ms.2004.1259216.
- [82] A. Fosfuri, M. S. Giarratana, A. Luzzi, The Penguin Has Entered the Building: The Commercialization of Open Source Software Products, Organization Science 19 (2) (2008) 292–305. doi:10.1287/orsc.1070.0321.
- [83] A. Fuggetta, A Classification of CASE Technology, Computer 26 (12) (1993) 25–38. doi:10.1109/2.247645.
- [84] A. Fuggetta, Open source software—an evaluation, Journal of Systems and Software 66 (1) (2003) 77 – 90. doi:10.1016/s0164-1212(02)00065-1.

- [85] C. Gacek, B. Arief, The Many Meanings of Open Source, *IEEE Software* 21 (1) (2004) 34–40. doi:10.1109/ms.2004.1259206.
- [86] A. Galatescu, V. Florian, L. Costea, D. Conescu, Issues in Implementing an Open Source-based XML Repository Manager for Application Maintenance and Adaptation, in: Feller et al. [73], pp. 57–62.
- [87] K. Gary, L. Ibanez, S. Aylward, D. Gobbi, M. B. Blake, K. Cleary, IGSTK: an open source software toolkit for image-guided surgery, *Computer* 39 (4) (2006) 46–53. doi:10.1109/mc.2006.130.
- [88] D. M. German, The evolution of the GNOME Project, in: Meeting challenges and surviving success: the 2nd Workshop on Open Source Software Engineering (WOSSE 2002), 25 May, Orlando, USA, 2002, pp. 1–4.
- [89] R. A. Ghosh, *Study on the Economic Impact of Open Source Software on Innovation and the Competitiveness of the Information and Communication Technologies (ICT) Sector in the EU*, Tech. rep., UNU-MERIT (2006).
URL <http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf>
- [90] P. Giacalone, OSS implementation solutions for Public Administration applications, in: Scotto and Succi [168], pp. 259–262.
- [91] D. G. Glance, J. Kerr, A. Reid, Factors affecting the use of open source software in tertiary education institutions, *First Monday* 9 (2).
- [92] R. L. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, *Information and Software Technology* 44 (8) (2002) 491–506. doi:10.1016/s0950-5849(02)00049-6.
- [93] E. Glynn, B. Fitzgerald, C. Exton, Commercial Adoption of Open Source Software: An Empirical Study, in: J. Verner, G. H. Travassos (Eds.), *Proceedings of International Symposium on Empirical Software Engineering (ISESE 2005)*, November 17th-18th, Noosa Heads, Australia, IEEE Computer Society, 2005, pp. 225–234. doi:10.1109/ISESE.2005.1541831.
- [94] S. Grand, G. von Krogh, D. Leonard, W. Swap, Resource allocation beyond firm boundaries: A multi-level model for Open Source innovation, *Long Range Planning* 37 (6) (2004) 591–610. doi:10.1016/j.lrp.2004.09.006.
- [95] M. Gschwind, D. Erb, S. Manning, M. Nutter, An Open Source Environment for Cell Broadband Engine System Software, *Computer* 40 (6) (2007) 37–47. doi:10.1109/MC.2007.192.
- [96] V. K. Gurbani, A. Garvert, J. D. Herbsleb, A Case Study of Open Source Tools and Practices in a Commercial Setting, in: Feller et al. [75], pp. 1–6. doi:10.1145/1083258.1083264.

- [97] V. K. Gurbani, A. Garvert, J. D. Herbsleb, A Case Study of a Corporate Open Source Development Model, in: Osterweil et al. [147], pp. 472–481.
- [98] Ø. Hauge, C.-F. Sørensen, R. Conradi, Adoption of Open Source in the Software Industry, in: Russo et al. [163], pp. 211–222. doi:10.1007/978-0-387-09684-1_17.
- [99] Ø. Hauge, C.-F. Sørensen, A. Røsdal, Surveying Industrial Roles in Open Source Software Development, in: Feller et al. [77], pp. 259–264. doi:10.1007/978-0-387-72486-7_25.
- [100] F. Hecker, Setting Up Shop: The Business of Open-Source Software, IEEE Software 16 (1) (1999) 45–51. doi:10.1109/52.744568.
- [101] J. Henkel, Selective revealing in open innovation processes: The case of embedded Linux, Research Policy 35 (7) (2006) 953 – 969. doi:10.1016/j.respol.2006.04.010.
- [102] G. Hertel, S. Niedner, S. Herrmann, Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel, Research Policy 32 (7) (2003) 1159 – 1177, open Source Software Development. doi:10.1016/s0048-7333(03)00047-7.
- [103] A. Höfer, W. Tichy, Status of Empirical Research in Software Engineering, in: V. R. Basili, D. Rombach, K. Schneider, B. A. Kitchenham, D. Pfahl, R. W. Selby (Eds.), Proceedings of the International Workshop on Empirical Software Engineering Issues. Critical Assessment and Future Directions, Dagstuhl, June 26th-30th, Castle, Germany, Vol. 4336/2007 of Lecture Notes in Computer Science, Springer, 2007, pp. 10–19. doi:10.1007/978-3-540-71301-2_3.
- [104] T. Hoffman, SchoolTool: Defining Our Niche in the Open Source Architecture of Schools, in: Scotto and Succi [168], pp. 334–337.
- [105] O. Hummel, W. Janjic, C. Atkinson, Code conjurer: Pulling reusable software out of thin air, IEEE Software 25 (5) (2008) 45–52. doi:10.1109/ms.2008.110.
- [106] N. Iivari, H. Hedberg, T. Kirves, Usability in Company Open Source Software Context - Initial Findings from an Empirical Case Study, in: Russo et al. [163], pp. 359–365. doi:10.1007/978-0-387-09684-1_33.
- [107] A. Jaaksi, Experiences on Product Development with Open Source Software, in: Feller et al. [77], pp. 85–96. doi:10.1007/978-0-387-72486-7_7.
- [108] J. Jacobson, M. Lewis, Game engine virtual reality with CaveUT, Computer 38 (4) (2005) 79–82. doi:10.1109/mc.2005.126.

- [109] J. Järvensivu, T. Mikkonen, Forging a community not: Experiences on establishing an open source project, in: Russo et al. [163], pp. 15–27. doi:[10.1007/978-0-387-09684-1_2](https://doi.org/10.1007/978-0-387-09684-1_2).
- [110] C. Jensen, W. Scacchi, Collaboration, Leadership, Control, and Conflict Negotiation in the Netbeans.org Community, in: Feller et al. [74], pp. 48–52.
- [111] M. Jørgensen, M. Shepperd, A systematic review of software development cost estimation studies, IEEE Transactions on Software Engineering 33 (1) (2007) 33–53. doi:[10.1109/tse.2007.3](https://doi.org/10.1109/tse.2007.3).
- [112] N. Juristo, C. B. Seaman, S. Vegas (Eds.), First International Symposium on Empirical Software Engineering and Measurement (ESEM'2007), Madrid, Spain 20-21 September, IEEE Computer Society, 2007.
- [113] V. B. Kampenes, T. Dybå, J. E. Hannay, D. I. K. Sjøberg, A systematic review of quasi-experiments in software engineering, Information and Software Technology 51 (1) (2009) 71–82. doi:[10.1016/j.infsof.2008.04.006](https://doi.org/10.1016/j.infsof.2008.04.006).
- [114] C. Kapser, M. W. Godfrey, Cloning considered harmful considered harmful: patterns of cloning in software, Empirical Software Engineering 13 (6) (2008) 645–692. doi:[10.1007/s10664-008-9076-6](https://doi.org/10.1007/s10664-008-9076-6).
- [115] B. A. Kitchenham, Guidelines for performing Systematic Literature Reviews in Software Engineering, Tech. rep., Software Engineering Group, School of Computer Science and Mathematics, Keele University, and Department of Computer Science, University of Durham, eBSE Technical Report, EBSE-2007-01 (2007).
- [116] B. A. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering - A systematic literature review, Information and Software Technology 51 (1) (2009) 7–15. doi:[10.1016/j.infsof.2008.09.009](https://doi.org/10.1016/j.infsof.2008.09.009).
- [117] B. A. Kitchenham, T. Dybå, M. Jørgensen, Evidence-based software engineering, in: A. Finkelstein, J. Estublier, D. Rosenblum (Eds.), Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), May 23th-28th, Edinburgh, Scotland, IEEE Computer Society, 2004, pp. 273–281.
- [118] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, J. Rosenberg, Preliminary Guidelines for Empirical Research in Software Engineering, IEEE Transactions on Software Engineering 28 (8) (2002) 721–734.
- [119] K. R. Lakhani, E. von Hippel, How open source software works: 'free' user-to-user assistance, Research Policy 32 (6) (2003) 923 – 943. doi:[10.1016/s0048-7333\(02\)00095-1](https://doi.org/10.1016/s0048-7333(02)00095-1).

- [120] J. R. Landis, G. G. Koch, The Measurement of Observer Agreement for Categorical Data, *Biometrics* 33 (1) (1977) 159–174.
- [121] N. Lesiecki, Applying AspectJ to J2EE application development, *IEEE Software* 23 (1) (2006) 24–32. doi:10.1109/ms.2006.1.
- [122] J. Li, R. Conradi, C. Bunse, M. Torchiano, O. P. N. Slyngstad, M. Morisio, Development with off-the-shelf components: 10 facts, *IEEE Software* 26 (2) (2009) 80–87. doi:10.1109/MS.2009.33.
- [123] J. Li, R. Conradi, O. P. N. Slyngstad, C. Bunse, M. Torchiano, M. Morisio, An Empirical Study on Decision Making in Off-The-Shelf Component-Based Development, in: Osterweil et al. [147], pp. 897–900. doi:10.1145/1134285.1134446.
- [124] J. Li, M. Torchiano, R. Conradi, O. P. N. Slyngstad, C. Bunse, A State-of-the-Practice Survey of Off-the-Shelf Component-Based Development Processes, in: M. Morisio (Ed.), *Proceedings of the 9th International Conference on Software Reuse (ICSR'06)*, June, 12th-15th, Torino, Italy, Vol. Volume 4039/2006 of LNCS, Springer, 2006, pp. 16–28. doi:10.1007/11763864_2.
- [125] J. Lindman, M. Rossi, P. Marttiin, Applying Open Source Development Practices Inside a Company, in: Russo et al. [163], pp. 381–387. doi:10.1007/978-0-387-09684-1_36.
- [126] D. Lorenzi, C. Rossi, Assessing Innovation in the Software Sector: Proprietary vs. FOSS Production Mode. Preliminary Evidence from the Italian Case, in: Russo et al. [163], pp. 325–331. doi:10.1007/978-0-387-09684-1_29.
- [127] B. Lundell, B. Lings, E. Lindqvist, Perceptions and Uptake of Open Source in Swedish Organisations, in: Damiani et al. [53], pp. 155–163. doi:10.1007/0-387-34226-5.
- [128] S. Lussier, New Tricks: How Open Source Changed the Way My Team Works, *IEEE Software* 21 (1) (2004) 68–72. doi:10.1109/MS.2004.1259222.
- [129] B. Luthiger, C. Jungwirth, Pervasive fun, *First Monday* 12 (1).
- [130] H. Mannaert, K. Ven, The Use of Open Source Software Platforms by Independent Software Vendors: Issues and Opportunities, in: Feller et al. [75], pp. 35–38. doi:10.1145/1083258.1083266.
- [131] K. Martin, B. Hoffman, An Open Source Approach to Developing Software in a Small Organization, *IEEE Software* 24 (1) (2007) 46–53. doi:10.1109/MS.2007.5.
- [132] J. Martinez-Romo, G. Robles, J. M. González-Barahona, M. Ortuño-Perez, Using Social Network Analysis Techniques to Study Collaboration between a FLOSS Community and a Company, in: Russo et al. [163], pp. 171–186. doi:10.1007/978-0-387-09684-1_14.

- [133] J. Matusow, S. McGibbon, D. Rowe, Shared Source and Open Solutions: e-Government Perspective, in: Scotto and Succi [168], pp. 263–266.
- [134] D. McIlroy, Mass Produced Software Components, Scientific Affairs Division NATO, Brussels, Belgium, 1969, pp. 138–151.
- [135] C. Melian, M. Mähring, Lost and Gained in Translation: Adoption of Open Source Software Development at Hewlett-Packard, in: Russo et al. [163], pp. 93–104. doi: [10.1007/978-0-387-09684-1_8](https://doi.org/10.1007/978-0-387-09684-1_8).
- [136] A. Mockus, R. T. Fielding, J. D. Herbsleb, Two case studies of open source software development: Apache and Mozilla, ACM Transactions on Software Engineering and Methodology 11 (3) (2002) 309–346. doi: [10.1145/567793.567795](https://doi.org/10.1145/567793.567795).
- [137] P. Mohagheghi, R. Conradi, Quality, productivity and economic benefits of software reuse: a review of industrial studies, Empirical Software Engineering 12 (5) (2007) 471–516. doi: [10.1007/s10664-007-9040-x](https://doi.org/10.1007/s10664-007-9040-x).
- [138] M. Montesi, P. Lago, Software engineering article types: An analysis of the literature, Journal of Systems and Software 81 (10) (2008) 1694 – 1714. doi: [10.1016/j.jss.2007.11.723](https://doi.org/10.1016/j.jss.2007.11.723).
- [139] L. Morgan, P. Finnegan, Benefits and Drawbacks of Open Source Software: An Exploratory Study of Secondary Software Firms, in: Feller et al. [77], pp. 307–312. doi: [10.1007/978-0-387-72486-7_33](https://doi.org/10.1007/978-0-387-72486-7_33).
- [140] L. Morgan, P. Finnegan, How Perceptions of Open Source Software Influence Adoption: An Exploratory Study, in: H. Österle, J. Schelp, R. Winter (Eds.), Proceedings of the Fifteenth European Conference on Information Systems (ECIS 2007), June 7-9 2007, St. Gallen, Switzerland, University of St. Gallen, 2007, pp. 973–984.
- [141] U. Nikula, S. Jantunen, Quantifying the Interest in Open Source System: Case South-East Finland, in: Scotto and Succi [168], pp. 192–195.
- [142] J. Noll, What Constitutes Open Source? A Study of the Vista Electronic Medical Record Software, in: Boldyreff et al. [25], pp. 310–319. doi: [10.1007/978-3-642-02032-2_27](https://doi.org/10.1007/978-3-642-02032-2_27).
- [143] J. S. Norris, Mission-critical Development with Open Source Software: Lessons Learned, IEEE Software 21 (1) (2004) 42–49. doi: [10.1109/MS.2004.1259211](https://doi.org/10.1109/MS.2004.1259211).
- [144] A. Onetti, F. Capobianco, Open Source and Business Model Innovation. The Funambol Case, in: Scotto and Succi [168], pp. 224–227.
- [145] The Open Source Initiative - Open Source Licenses (2009). URL <http://opensource.org/licenses>

- [146] T. Østerlie, L. Jaccheri, A Critical Review of Software Engineering Research on Open Source Software Development, in: W. Stanislaw (Ed.), Proceedings of the 2nd AIS SIGSAND European Symposium on Systems Analysis and Design, June 5th, Gdansk, Poland, Gdansk University Press, 2007, pp. 12–20.
- [147] L. J. Osterweil, D. Rombach, M. L. Soffa (Eds.), Proceedings of the 28th International Conference on Software Engineering (ICSE 2006), May 20th-28th, Shanghai, China, ACM Press, 2006.
- [148] B. Özel, U. Jovanovic, B. Oba, M. van Leeuwen, Perceptions on F/OSS Adoption, in: Feller et al. [77], pp. 319–324. doi:[10.1007/978-0-387-72486-7_35](https://doi.org/10.1007/978-0-387-72486-7_35).
- [149] J. W. Paulson, G. Succi, A. Eberlein, An Empirical Study of Open-Source and Closed-Source Software Products, IEEE Transactions on Software Engineering 30 (4) (2004) 246–256. doi:[10.1109/TSE.2004.1274044](https://doi.org/10.1109/TSE.2004.1274044).
- [150] A. Persson, B. Lings, B. Lundell, A. Mattsson, U. Ärlig, Communication, Coordination and Control in Distributed Development: an OSS Case Study, in: Scotto and Succi [168], pp. 88–92.
- [151] J. Ploski, W. Hasselbring, J. Rehwinkel, S. Schwierz, Introducing Version Control to Database-Centric Applications in a Small Enterprise, IEEE Software 24 (1) (2007) 38–44. doi:[10.1109/ms.2007.17](https://doi.org/10.1109/ms.2007.17).
- [152] C. Puschmann, P. Reimer, DiPP and eLanguage: Two cooperative models for open access, First Monday 12 (10).
- [153] P. Ravesteyn, G. Silvius, Willingness to Cooperate Within the Open Source Software Domain, in: Russo et al. [163], pp. 367–373. doi:[10.1007/978-0-387-09684-1_34](https://doi.org/10.1007/978-0-387-09684-1_34).
- [154] E. S. Raymond, The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O’Reilly, Sebastapol, CA, 2001.
- [155] E. S. Raymond, Up from Alchemy, IEEE Software 21 (1) (2004) 88, 90. doi:[10.1109/ms.2004.1259228](https://doi.org/10.1109/ms.2004.1259228).
- [156] P. C. Rigby, D. Cubranic, S. Thompson, D. M. German, M.-A. Storey, The challenges of creating open source educational software: the Gild experience, in: Scotto and Succi [168], pp. 338–340.
- [157] J. E. Robbins, Adopting Open Source Software Engineering (OSSE) Practices by Adopting OSSE Tools, in: Feller et al. [76], pp. 245–264.
- [158] G. Robles, S. Dueñas, J. M. González-Barahona, Corporate involvement of libre software: Study of presence in debian code over time, in: Feller et al. [77], pp. 121–132. doi:[10.1007/978-0-387-72486-7_10](https://doi.org/10.1007/978-0-387-72486-7_10).

- [159] B. Rossi, B. Russo, G. Succi, A study on the introduction of Open Source Software in the Public Administration, in: Damiani et al. [53], pp. 165–171. doi:10.1007/0-387-34226-5_16.
- [160] B. Rossi, B. Russo, G. Succi, Open Source Software and Open Data Standards as a form of Technology Adoption: a Case Study, in: Feller et al. [77], pp. 325–330. doi:10.1007/978-0-387-72486-7_36.
- [161] B. Rossi, M. Scotto, A. Sillitti, G. Succi, Criteria for the non invasive transition to OpenOffice, in: Scotto and Succi [168], pp. 250–253.
- [162] C. Rossi, A. Bonaccorsi, Why Profit-Oriented Companies Enter the OS Field?: Intrinsic vs. Extrinsic Incentives, in: Feller et al. [75], pp. 47–51. doi:10.1145/1083258.1083269.
- [163] B. Russo, E. Damiani, S. A. Hissam, B. Lundell, G. Succi (Eds.), Proceedings of the 4th IFIP Working Group 2.13 International Conferences on Open Source Software (OSS2008) - Open Source Development Communities and Quality, September 7th–10th, Milano, Italy, Vol. 275/2008 of IFIP International Federation for Information Processing, Springer, 2008. doi:10.1007/978-0-387-09684-1.
- [164] B. Russo, P. Zuliani, G. Succi, Toward an Empirical Assessment of the Benefits of Open Source Software, in: Feller et al. [73], pp. 117–120.
- [165] C. Santos Jr., Understanding Partnerships between Corporations and the Open Source Community: A Research Gap, IEEE Software 25 (6) (2008) 96–97. doi:10.1109/ms.2008.167.
- [166] W. Scacchi, Free and Open Source Development Practices in the Game Community, IEEE Software 21 (1) (2004) 59–66. doi:10.1109/ms.2004.1259221.
- [167] W. Scacchi, J. Feller, B. Fitzgerald, S. A. Hissam, K. R. Lakhani, Understanding Free/Open Source Software Development Processes, Software Process: Improvement and Practice 11 (2) (2006) 95–105. doi:10.1002/spip.255.
- [168] M. Scotto, G. Succi (Eds.), Proceedings of The First International Conference on Open Source Systems (OSS2005), July 11th–15th, Genova, Italy, 2005.
- [169] N. Serrano, S. Calzada, J. M. Sarriegui, I. Ciordia, From Proprietary to Open Source Tools in Information Systems Development, IEEE Software 21 (1) (2004) 56– 58. doi:10.1109/MS.2004.1259219.
- [170] R. Shatnawi, W. Li, The effectiveness of software metrics in identifying error-prone classes in post-release software evolution process, Journal of Systems and Software 81 (11) (2008) 1868 – 1882. doi:10.1016/j.jss.2007.12.794.

- [171] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, A. C. Rekdal, A survey of controlled experiments in software engineering, *IEEE Transactions on Software Engineering* 31 (9) (2005) 733–753. doi:[10.1109/tse.2005.97](https://doi.org/10.1109/tse.2005.97).
- [172] S. Y. Sohn, M. S. Mok, A strategic analysis for successful open source software utilization based on a structural equation model, *Journal of Systems and Software* 81 (6) (2008) 1014–1024. doi:[10.1016/j.jss.2007.08.034](https://doi.org/10.1016/j.jss.2007.08.034).
- [173] S. K. Sowe, I. Stamelos, L. Angelis, Understanding knowledge sharing activities in free/open source software projects: An empirical study, *Journal of Systems and Software* 81 (3) (2008) 431–446. doi:[10.1016/j.jss.2007.03.086](https://doi.org/10.1016/j.jss.2007.03.086).
- [174] D. Spinellis, Global software development in the FreeBSD project, in: P. Kruchten, D. Moitra, W. Strigel, C. Ebert (Eds.), *Proceedings of the 2006 International Workshop on Global Software Development for the Practitioner*, ACM Press, 2006, pp. 73–79. doi:[10.1145/1138506.1138524](https://doi.org/10.1145/1138506.1138524).
- [175] R. M. Stallman, L. Lessig, *Free Software Free Society: selected essays of Richard M. Stallman*, Free Software Foundation, 2002.
- [176] W. Stam, When does community participation enhance the performance of open source software companies?, *Research Policy* 38 (8) (2009) 1288–1299. doi:[10.1016/j.respol.2009.06.004](https://doi.org/10.1016/j.respol.2009.06.004).
- [177] I. Stamelos, L. Angelis, A. Oikonomou, G. L. Bleris, Code quality analysis in open source software development, *Information Systems Journal* 12 (1) (2002) 43–60.
- [178] M. Staples, M. Niazi, Experiences using systematic review guidelines, *Journal of Systems and Software* 80 (9) (2007) 1425–1437. doi:[10.1016/j.jss.2006.09.046](https://doi.org/10.1016/j.jss.2006.09.046).
- [179] K. Staring, O. Titlestad, Networks of Open Source Health Care Action, in: Damiani et al. [53], pp. 135–141. doi:[10.1007/0-387-34226-5_13](https://doi.org/10.1007/0-387-34226-5_13).
- [180] K.-J. Stol, M. A. Babar, Reporting Empirical Research in Open Source Software: The State of Practice, in: Boldyreff et al. [25], pp. 156–169. doi:[10.1007/978-3-642-02032-2_15](https://doi.org/10.1007/978-3-642-02032-2_15).
- [181] F. Tiangco, A. Stockwell, J. Sapsford, A. Rainer, Open-source software in an occupational health application: the case of Heales Medical Ltd., in: Scotto and Succi [168], pp. 130–134.
- [182] K. Toth, Experiences with Open Source Software Engineering Tools, *IEEE Software* 23 (6) (2006) 44–52. doi:[10.1109/ms.2006.158](https://doi.org/10.1109/ms.2006.158).
- [183] W. Tracz, J. Magee, M. Young (Eds.), *Proceedings of the 24th International Conference on Software Engineering (ICSE 2002)*, May 19th-25th, Orlando, Florida, ACM, 2002.

- [184] M. Umarji, S. E. Sim, C. Lopes, Archetypal Internet-Scale Source Code Searching, in: Russo et al. [163], pp. 257–263. doi:10.1007/978-0-387-09684-1_21.
- [185] F. van der Linden, B. Lundell, P. Marttiin, Commodification of Industrial Software: A Case for Open Source, IEEE Software 26 (4) (2009) 77–83. doi:10.1109/ms.2009.88.
- [186] K. Ven, H. Mannaert, Challenges and strategies in the use of Open Source Software by Independent Software Vendors, Information and Software Technology 50 (9-10) (2008) 991–1002. doi:10.1016/j.infsof.2007.09.001.
- [187] K. Ven, D. Van Nuffel, J. Verelst, The Introduction of OpenOffice.org in the Brussels Public Administration, in: Damiani et al. [53], pp. 123–134. doi:10.1007/0-387-34226-5_12.
- [188] K. Ven, J. Verelst, The Organizational Adoption of Open Source Server Software by Belgian Organizations, in: Damiani et al. [53], pp. 111–122. doi:10.1007/0-387-34226-5_11.
- [189] K. Ven, J. Verelst, The Impact of Ideology on the Organizational Adoption of Open Source Software, Journal of Database Management 19 (2) (2008) 58–72.
- [190] K. Ven, J. Verelst, H. Mannaert, Should You Adopt Open Source Software?, IEEE Software 25 (3) (2008) 54–59. doi:10.1109/ms.2008.73.
- [191] V. Venkatesh, M. G. Morris, G. B. Davis, F. D. Davis, User Acceptance of Information Technology: Toward a Unified View, Mis Quarterly 27 (3) (2003) 425–478.
- [192] I. Vessey, V. Ramesh, R. L. Glass, Research in Information Systems: An Empirical Study of Diversity in the Discipline and Its Journals, Journal of Management Information Systems 19 (2) (2002) 129–174.
- [193] P. Vitharana, F. M. Zahedi, H. Jain, Design, Retrieval, and Assembly in Component-based Software Development, Communications of the ACM 46 (11) (2003) 97–102. doi:10.1145/948383.948387.
- [194] E. von Hippel, G. von Krogh, Open source software and the "private-collective" innovation model: Issues for organization science, Organization Science 14 (2) (2003) 209.
- [195] G. von Krogh, S. Spaeth, S. Haefliger, Knowledge Reuse in Open Source Software: An Exploratory Study of 15 Open Source Projects, in: R. H. Sprague, Jr. (Ed.), Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005), January 3rd-6th, Big Island, Hawaii, IEEE Computer Society, 2005, p. 198.2. doi:10.1109/hicss.2005.378.
- [196] G. von Krogh, E. von Hippel, The Promise of Research on Open Source Software, Management Science 52 (7) (2006) 975–983. doi:10.1287/mnsc.1060.0560.

- [197] J. Waldo, Alive and well: Jini technology today, *Computer* 33 (6) (2000) 107–109. doi:[10.1109/2.846324](https://doi.org/10.1109/2.846324).
- [198] D. A. E. Wall, Using Open Source for a Profitable Startup, *Computer* 34 (12) (2001) 158–160. doi:[10.1109/2.970592](https://doi.org/10.1109/2.970592).
- [199] J. Warsta, P. Abrahamsson, Is open source software development essentially an agile method?, in: Feller et al. [73].
- [200] A. I. Wasserman, E. Capra, Evaluating Software Engineering Processes in Commercial and Community Open Source Projects, in: Capiluppi and Robles [41], pp. 1–5. doi:[10.1109/floss.2007.6](https://doi.org/10.1109/floss.2007.6).
- [201] R. T. Watson, M.-C. Boudreau, P. T. York, M. E. Greiner, J. Donald Wynn, The Business of Open Source, *Communications of the ACM* 51 (4) (2008) 41–46. doi:[10.1145/1330311.1330321](https://doi.org/10.1145/1330311.1330321).
- [202] R. T. Watson, D. Wynn, M.-C. Boudreau, JBoss: The Evolution of Professional Open Source Software, *MIS Quarterly Executive* 4 (3) (2005) 329–341.
- [203] J. Wesselius, The Bazaar inside the Cathedral: Business Models for Internal Markets, *IEEE Software* 25 (3) (2008) 60–66. doi:[10.1109/ms.2008.79](https://doi.org/10.1109/ms.2008.79).
- [204] J. West, How open is open enough?: Melding proprietary and open source platform strategies, *Research Policy* 32 (7) (2003) 1259 – 1285. doi:[10.1016/S0048-7333\(03\)00052-0](https://doi.org/10.1016/S0048-7333(03)00052-0).
- [205] J. West, S. O’Mahony, The Role of Participation Architecture in Growing Sponsored Open Source Communities , *Industry & Innovation* 15 (2) (2008) 145–168. doi:[10.1080/13662710801970142](https://doi.org/10.1080/13662710801970142).
- [206] A. Westenholz, Institutional Entrepreneurs and the Bricolage of Intellectual Property Discourses, in: Damiani et al. [53], pp. 183–193. doi:[10.1007/0-387-34226-5_18](https://doi.org/10.1007/0-387-34226-5_18).
- [207] M. N. Wicks, R. G. Dewar, A new research agenda for tool integration, *Journal of Systems and Software* 80 (9) (2007) 1569–1585. doi:[10.1016/j.jss.2007.03.089](https://doi.org/10.1016/j.jss.2007.03.089).
- [208] Y. Yang, J. Bhuta, B. W. Boehm, D. N. Port, Value-Based Processes for COTS-Based Applications, *IEEE Software* 22 (4) (2005) 54–62. doi:[10.1109/MS.2005.112](https://doi.org/10.1109/MS.2005.112).
- [209] Z. Yang, M. Jiang, Using Eclipse as a Tool-Integration Platform for Software Development, *IEEE Software* 24 (2) (2007) 87–89. doi:[10.1109/ms.2007.58](https://doi.org/10.1109/ms.2007.58).
- [210] T. S. Yoo, M. J. Ackerman, Open Source Software for Medical Image Processing and Visualization, *Communications of the ACM* 48 (2) (2005) 55–59. doi:[10.1145/1042091.1042120](https://doi.org/10.1145/1042091.1042120).

- [211] L. Yu, Understanding component co-evolution with a study on Linux, *Empirical Software Engineering* 12 (2) (2007) 123–141. doi:[10.1007/s10664-006-9000-x](https://doi.org/10.1007/s10664-006-9000-x).
- [212] L. Yu, Self-organization process in open-source software: An empirical study, *Information and Software Technology* 50 (5) (2008) 361 – 374. doi:[10.1016/j.infsof.2007.02.018](https://doi.org/10.1016/j.infsof.2007.02.018).
- [213] M. V. Zelkowitz, An update to experimental models for validating computer technology, *Journal of Systems and Software* 82 (3) (2009) 373–376. doi:[10.1016/j.jss.2008.06.040](https://doi.org/10.1016/j.jss.2008.06.040).
- [214] M. V. Zelkowitz, D. R. Wallace, Experimental Models for Validating Technologies, *IEEE Computer* 31 (5) (1998) 23–31.
- [215] L. Zhao, S. Elbaum, Quality assurance under the open source development model, *Journal of Systems and Software* 66 (1) (2003) 65 – 75. doi:[10.1016/s0164-1212\(02\)00064-x](https://doi.org/10.1016/s0164-1212(02)00064-x).
- [216] S. Ziemer, Ø. Hauge, T. Østerlie, J. Lindman, Understanding Open Source in an Industrial Context, in: A. Dipanda, R. Chbeir, K. Yetongnon (Eds.), *Proceedings of the 4th IEEE International Conference on Signal-Image Technology & Internet-Based Systems (SITIS'2008)*, November 30th-December 3rd, Bali, Indonesia, IEEE Computer Society, 2008, pp. 539–546. doi:[10.1109/SITIS.2008.99](https://doi.org/10.1109/SITIS.2008.99).

APPENDICES

A. Included Publication Sources

Table 13 contains an overview of the journals/magazines and the workshop and conference proceedings included in this review, the databases from which the publications were identified, and the number of publications included in each stage (S1-5).

Source	Database	Number of publications																		
		S1	S2	S3	S4	S5			S5											
						R	E	N												
ACM Transactions on Software Engineering and Methodology (TOSEM)	ACM Digital Library	207	22	2	0	0	0	0	0	0										
Communications of the ACM	ACM Digital Library	3252	217	25	9	1	2	6												
Empirical Software Engineering (EMSE)	Springerlink	272	37	11	0															
First Monday	Firstmonday.org	965	247	76	13	2	1	9												
ICSE Workshop on Open Source Software Engineering (WOSSE)	opensource.ucc.ie (2001-2004), ACM Digital Library (2005)	95	95	90	20	7	7	6												
ICSE Workshop on Emerging Trends in FLOSS Research and Development (FLOSS)	IEEE Xplore	15	15	12	3	2	0	1												
IEE Review	ietdl.org	999	5	3	2	0	0	2												
IEE/IET Software Proceedings	ietdl.org	323	12	6	0	0	0	0												
IEEE Computer	IEEE Xplore	2638	30	23	8	0	5	3												
IEEE International Symposium on Empirical Software Engineering (ISESE)	IEEE Xplore (2002-2005), ACM Digital Library (2006)	177	8	5	1	1	0	0												
IEEE International Symposium on Software Metrics (METRICS)	IEEE Xplore	249	7	6	0	0	0	0												
IEEE Software	IEEE Xplore	1439	76	50	21	1	14	6												
IEEE Transactions on Software Engineering (TSE)	IEEE Xplore	900	16	16	0	0	0	0												
Information and Software Technology (IST)	Sciencedirect	1089	81	14	2	1	1	0												
Information Systems Journal	Wiley InterScience	279	9	7	1	0	0	1												
International Conference on Open Source Systems (OSS)	oss2005.case.unibz.it (2005), Springerlink (2006-2008)	210	209	175	55	27	19	9												
International Conference on Software Engineering (ICSE)	ACM Digital Library (2006 and 2008), IEEE Xplore (1998-2005, 2007)	1324	105	32	5	1	4	0												
International Symposium on Empirical Software Engineering and Measurement (ESEM)	IEEE Xplore (2007), ACM Digital Library (2008)	151	18	12	0	0	0	0												
Journal of Database Management	ProQuest	294	18	9	3	2	0	1												
Journal of Industrial Economics	Wiley InterScience	304	2	2	0	0	0	0												
Journal of Systems and Software (JSS)	Sciencedirect	1567	119	23	2	2	0	0												
Knowledge Technology and Policy	Springerlink	364	28	13	7	1	0	6												
Long Range Planning	Sciencedirect	1017	19	2	2	2	0	0												
Management Science	ProQuest	1787	17	12	1	1	0	0												
MIS Quarterly	ProQuest	406	7	3	1	1	0	0												
MIS Quarterly Executive	Misce.org	146	1	1	1	1	0	0												
MIT Sloan Management Review	ProQuest	836	27	3	0	0	0	0												
Organization Science	ProQuest	674	10	3	1	1	0	0												
Research Policy	Sciencedirect	1232	52	18	4	4	0	0												
Software Practice and Experience	Wiley InterScience	763	10	8	0	0	0	0												
Software Process: Improvement and Practice	Wiley InterScience	315	21	12	1	1	0	0												
Total		24 289	1540	674	162	112	59	53	50	na										

Table 13: Included journals and conferences

B. Characteristics of the Sample of Publications

The tables below contain overviews of the number of publications per year (Table 14, the contexts described in these publications (Table 15 and 18), the research methods used in the empirical research papers (Table 16), and the quality assessment (Table 17).

We use the following abbreviations: Stages 1 to 5 (S1-5), empirical research paper (R), experience report (E), non-empirical paper (N). The categories from the classification of the papers are abbreviated as follows: adoption of OSS in general (A), deploying OSS (D), using OSS CASE tools (T), integrating OSS (I), participating in OSS communities (PA), providing OSS products (PRO), and using “OSS development practices” (PRA).

Year	S2	S3	S4	S5		
				R	E	N
2008	302	109	26	18	6	2
2007	254	110	23	10	9	4
2006	262	106	28	13	7	8
2005	226	117	41	11	17	13
2004	156	64	17	5	6	6
2003	119	63	7	2	1	4
2002	86	37	5	0	3	2
2001	74	37	6	0	2	4
2000	26	9	2	0	1	1
1999	24	15	6	0	1	5
1998	11	7	1	0	0	1
Total	1540	674	162	59	53	50

Table 14: Number of publications in each stage distributed per year

	A	D	T	I	PA	PRO	PRA
Several private organizations	[6, 27, 32, 33, 94, 98, 99, 127, 139, 141, 153, 162, 172, 188, 189, 204]			[8, 43, 123]	[31, 49, 50]	[17, 49, 50, 82, 204]	
OSS community					[2, 20, 28, 51, 70, 101, 129, 132, 158]	[110, 126, 132]	
One private organization				[106, 130, 186]	[106, 130, 186]	[144, 201, 202, 206]	[125, 135]
Public	[91]	[62, 91, 148, 159, 160, 161, 164, 187]					
Mixed public and private	[29, 93]	[37]			[200]	[7, 52, 200]	
Unclear		[190]					

Table 15: The type of contexts described in the empirical research papers

	A	D	T	I	PA	PRO	PRA
Case study	[94, 99, 139, 188, 189]	[37, 62, 159, 160, 164, 187, 190]		[106, 130, 186]	[20, 49, 50, 70, 101, 106, 130, 186]	[17, 49, 50, 52, 110, 144, 201, 202, 206]	[125, 135]
Survey	[27, 29, 32, 33, 91, 98, 127, 141, 153, 162, 172]	[91, 148]		[8, 43, 123]	[31, 129, 200]	[200]	
Data analysis					[2, 28, 51, 132, 158]	[82, 132]	
Experiment		[161]				[126]	
Field study	[204]					[204]	
Grounded theory	[6]						
Case study and survey	[93]					[7]	

Table 16: Research methods used

QA score	Papers
2	[29, 202]
3	[52, 110, 164, 201, 206]
4	[17]
5	[91, 200, 204]
6	[32, 37, 106, 127, 130, 148, 159, 160]
7	[20, 62, 93, 94, 132, 135, 141, 162]
8	[2, 6, 28, 31, 33, 49, 99, 101, 123, 139, 144, 153, 158, 172, 187]
9	[7, 8, 27, 43, 50, 51, 70, 82, 98, 125, 126, 129, 161, 186, 188, 189, 190]

Table 17: Quality assessment: Distribution of research papers

	D	T	I	PA	PRO	PRA
Private		[131, 143, 151, 198, 209]	[9, 87, 90, 107, 121, 143, 169, 181, 198]	[9, 107, 128]	[14, 16, 19, 34, 87, 95, 100, 131, 133, 197, 210]	[12, 60, 61, 96, 97, 128, 131, 179, 197, 203]
Public	[24, 57, 63, 81]	[13, 182]	[10, 15, 54, 152, 182]	[11, 15]	[22, 23, 47, 104, 108, 156]	[182]
Research project	[3, 4]		[3, 64, 86]		[26]	
Community				[88]	[109]	

Table 18: The type of contexts described in the experience reports