

The Role of Foundations in Open Source Projects

Javier Luis Cánovas Izquierdo
UOC
Barcelona, Spain
jcanovasi@uoc.edu

Jordi Cabot
ICREA
UOC
Barcelona, Spain
jordi.cabot@icrea.cat

ABSTRACT

In the last years, a number of Open-Source Systems (OSS) have created parallel foundations, as legal instruments to better articulate the structure, collaboration and financial model for the project. Some examples are Apache, Linux, Mozilla, Eclipse or Django foundations. Nevertheless, foundations largely differ in the kind of mission they have and the support they provide to their project/s. In this paper we study the role of foundations in open source software development. We analyze the nature of 89 software foundations and then focus on the 18 most relevant ones to study their openness and influence in the development practices taking place in the endorsed projects. Our results reveal the existence of a significant number of foundations with the sole purpose of promoting the importance of the free software movement and/or that limit themselves to core legal aspects but do not play any role in the day-to-day operations of the project (e.g., a few of them are just umbrella organizations for a large variety of projects). Therefore, while useful, foundations do not remove the need for specific projects to develop their own governance, contribution and development policies.

CCS CONCEPTS

• **Social and professional topics** → *Management of computing and information systems*; • **Software and its engineering** → *Software implementation planning*; *Software development techniques*; *Open source model*;

KEYWORDS

Software Foundations; Open-Source Software; Open-Source Software Analysis

ACM Reference Format:

Javier Luis Cánovas Izquierdo and Jordi Cabot. 2018. The Role of Foundations in Open Source Projects. In *ICSE-SEIS'18: 40th International Conference on Software Engineering: Software Track, May 27-June 3, 2018, Gothenburg, Sweden*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3183428.3183438>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICSE-SEIS'18, May 27-June 3, 2018, Gothenburg, Sweden

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5661-9/18/05...\$15.00

<https://doi.org/10.1145/3183428.3183438>

1 INTRODUCTION

The impact of Open Source Systems (OSS) in modern society is undeniable. It has become a relevant part of the software industry and has contributed to the advance of the state of the art in research, education and government [5].

Sustainability of OSS projects relies on active contributions of passionate developers [16, 26]. Therefore, survival of a OSS project largely depends on its ability to retain developers and onboard new ones (i.e., newcomers), as well as to create a community of users who promote its adoption and use [3, 30]. As these projects grow, developers tend to organize and create communities to drive their development [2, 7, 21] but many projects lack of formal models, specially governance models [18], to structure and manage the (potentially large) community around them. Support to deal with all kinds of organizational decisions (including legal and economical aspects) is a huge concern for all projects at this stage¹

In other domains, non-profit initiatives organize themselves around foundations (either public or private) that provide the legal and economical infrastructure for a community. Foundations can also define a number of internal regulations regarding, for instance, the activity, membership and decision-making process for the non-profit and non-governmental organizations. In the last years, we have also seen a number of foundations created around open source projects. Software foundations are non-profit organizations whose mission is to provide the needed grounds for open and collaborative software development. They also provide a legal framework for individual volunteers and enable the donation of resources for the public benefit. But, so far, little is known regarding the number, type, impact of such foundations and the differences among them. For instance, the *Apache Software Foundation* and *Linux Foundation* are maybe two of the most well-known software foundations. However, they follow different strategies for the management of the projects they cover. While the *Apache Software Foundation* proposes a meritocratic system where different committees control and drive the development of several software projects (and the board oversees the whole process); the *Linux Foundation* follows a flexible approach which serves as an umbrella for its projects, which can deploy specific development processes, and therefore concentrates more on the promotion of OSS benefits.

In this paper we study the different flavors of foundations behind open source software and their impact in the development of OSS projects. Our goal is to give a clear picture of the current state of the art of software foundations and to help developers make informed decisions when creating new ones or choosing an existing one to

¹As an example, we are part of an internal reflection taking part in the Decidim community (<https://github.com/decidim/decidim>, a participatory democracy software framework initiated by the Barcelona city council) aimed to decide whether a foundation should be created and, if so, which foundation model it should inspire on.

join. To this aim, we build a dataset of 89 software foundations, which we analyze according to their scope, openness and influence in the development practices taking place in the project itself.

The paper is organized as follows. Section 2 presents the research methodology followed in our study, including the definition of the research questions and the construction of the dataset for our study. Section 3 addresses the research questions and describes the results of our study. Section 4 describes the threats to validity. Section 5 presents the related work. Section 6 finalizes the paper and presents the future work.

2 RESEARCH METHOD

In this section we discuss how our study has been set up. We first present our research questions (Section 2.1) and then describe how we built the dataset for our study (Section 2.2).

2.1 Research Questions

Our objective is to grasp a better understanding of the role played by software foundations in the development of OSS. More specifically, we have identified the following main research questions:

RQ1 How many software foundations are there? As starting point, we want to study how common are these foundations and, more specifically, how many of them are actually aimed at supporting the development of specific open source projects (instead of, for instance, devote themselves to the promotion of the open source philosophy in general).

RQ2 What is their scope? We study the nature of software foundations in terms of their geographical distribution, coverage and mission.

RQ3 Do foundations play a direct role or have, at least, some influence in the way software development is conducted? We are interested in evaluating whether foundations limit themselves to provide a legal coverage or are used by projects as a way to structure their governance and contribution models in detail, acting, in fact, as a kind of supervision and monitoring entity for the development process.

RQ4 How open are software foundations? We evaluate the openness of the software foundations themselves, by examining their structure, how easy is to become a member or how decisions are taken.

2.2 Dataset Construction

To answer the questions above, we built a dataset composed of a number of foundations. To construct the dataset, we initially relied on the list of foundations available in flossfoundations.org², an online community created in 2005 by representatives from the *Python Foundation*, *Apache Software Foundation*, *Perl Foundation* and *Free Software Foundation*, with the aim of sharing their experiences and expertise in the field of free software and foundations. We complemented this list by incorporating additional foundations identified thanks to other sources devoted to promote OSS like opensource.com³ and [oss-watch](http://oss-watch.ac.uk)⁴, as well as our own expertise.

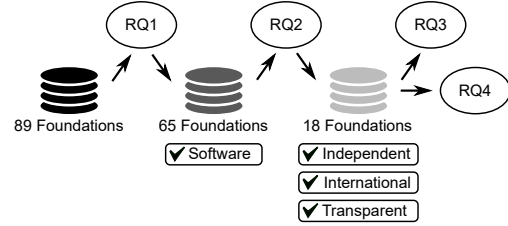


Figure 1: Method followed to answer the research questions.

For each foundation, we extracted (a) its URL; (b) the legal organization type (c) the number of projects they cover (if any and publicly available); and (d) a brief description. When needed, additional information is later on retrieved to answer specific aspects of our research questions.

At the end of the process, we constructed a dataset composed of 89 foundations. Table 1 (see first two columns) shows the list of foundations we collected reporting only aspects (a) and (c) due to space limitations.

Our dataset is fully used by RQ1, which identifies those foundations targeting software development. Then, RQ2 analyzes the scope of these software foundations (i.e., RQ1 output). Finally, RQ3 and RQ4 take the independent, international and transparent foundations identified in RQ2 to perform the study. Figure 1 shows the followed method and the size of the dataset involved in each research question.

3 RESULTS

In the following we address our research questions based on the analysis of the foundations considered in our dataset.

3.1 RQ1: Number of Foundations Supporting Open Source Projects

To address this research question, we first check each foundation in our dataset to find out whether its main goal is to support the development of a specific set of software projects. Other goals could include: training, certification or evangelization of open source in general. Table 1, column RQ1, presents the results of this analysis by showing a black square for foundations that fall in the former category (and a cross for the rest).

In total, 24 foundations were discarded at this stage and therefore will not be part of the core dataset for the other research questions (which focus only on studying those foundations whose main goal is to support OSS projects). Discarded foundations include well-known examples such as the *Free Software Foundation* or *Creative Commons*, among others.

The other 65 foundations are currently endorsing and helping a total of 871 software projects. Even though this is a low value in comparison to the number of projects being developed in OSS communities like GitHub⁵, with more than 67 million of repositories⁶; some of the projects covered by these foundations are among

²<http://flossfoundations.org>

³<http://opensource.com>

⁴<http://oss-watch.ac.uk>

⁵<https://github.com/>

⁶Number taken from <https://octoverse.github.com/> though the number of real projects under active development is probably, at least, an order of magnitude smaller than that.

Table 1: Foundations considered in our study and research questions.

NAME	URL	SIZE	RQ1	RQ2			RQ3
				GEO. DIST.	COVERAGE	MISSION	RQ4
Apache Software Foundation	http://www.apache.org/foundation/	312	■	■	■	■	■
Associação SoftwareLivre.org	http://associacao.softwarelivre.org/		×	n/a	n/a	n/a	n/a
Benetech	http://www.benetech.org/		×	n/a	n/a	n/a	n/a
BioBricks Foundation	http://bbf.openwetware.org/		×	n/a	n/a	n/a	n/a
Blender Foundation	http://www.blender.org/blenderorg/blender-foundation/		■	×	■	×	n/a
BSD Fund	http://bsdfund.org/		■	×	×	■	n/a
Creative Commons	http://creativecommons.org/		×	n/a	n/a	n/a	n/a
Django Software Foundation	http://www.djangoproject.com/foundation/	1	■	■	■	■	■
Document Foundation	http://www.documentfoundation.org/	1	■	■	■	■	■
Dojo Foundation	http://dojofoundation.org/		■	×	×	×	n/a
Eclipse Foundation	http://www.eclipse.org/org/foundation/	216	■	■	■	■	■
El Centro de Software Libre	http://www.csol.org/		×	×	×	×	n/a
Electronic Frontier Foundation	http://www.eff.org/		×	n/a	n/a	n/a	n/a
Free Knowledge Institute	http://freeknowledge.eu/		×	n/a	n/a	n/a	n/a
Free Software and Open Source Foundation for Africa	http://www.fossfa.net/		×	n/a	n/a	n/a	n/a
Free Software Foundation	http://www.fsf.org/		×	n/a	n/a	n/a	n/a
Free Software Foundation Europe	https://fsfe.org		■	×	×	■	n/a
Free Software Foundation India	http://www.fsf.org/in/		■	×	×	■	n/a
Free Software Foundation Latin America	http://www.fsfla.org/		■	×	×	■	n/a
FreeBSD Foundation	http://freebsd.foundation.org/		×	n/a	n/a	n/a	n/a
Fundación Via Libre	http://www.vialibre.org.ar/		×	n/a	n/a	n/a	n/a
Gentoo Foundation	https://www.gentoo.org/inside-gentoo/foundation/	1	■	■	■	■	■
GNOME Foundation	http://foundation.gnome.org/	1	■	■	■	■	■
Identity Commons	http://idcommons.net/		■	■	■	×	n/a
Internet Systems Consortium	http://www.isc.org/	9	■	■	■	×	n/a
ITPUG (Italian PostgreSQL Users' Group)	http://www.itpug.org/index.en.html		■	×	■	×	n/a
JS Foundation	https://js.foundation		■	×	×	■	n/a
JPUG (Japanese PostgreSQL Users' Group)	http://postgresql.jp		■	×	■	■	n/a
KDE e.V.	http://ev.kde.org/		■	×	■	■	n/a
Linux Expo of Southern California	http://www.socallinuxexpo.org/		■	×	■	×	n/a
Linux Foundation	http://linuxfoundation.org/	67	■	■	×	■	n/a
Linux Fund	http://linuxfoundation.org/		■	■	×	×	n/a
Linux International	http://www.li.org/		■	■	■	×	n/a
Linux Profesional Institute	https://www.lpi.org/		×	n/a	n/a	n/a	n/a
LogiLogi Foundation	http://foundation.logilogi.org/	1	■	■	■	×	n/a
Mambo Foundation Inc.	http://mambo-foundation.org/		■	■	■	×	n/a
Mozilla Foundation	http://www.mozilla.org/foundation/	9	■	■	■	■	■
NetBSD Foundation	http://www.netbsd.org/donations/	1	■	■	■	■	■
NLnet Foundation	http://www.nlnet.nl/		■	×	×	■	n/a
NLnet Labs Foundation	http://www.nlnetlabs.nl		■	×	×	■	n/a
NumFocus Inc.	https://www.numfocus.org/	43	■	■	×	■	n/a
One Laptop Per Child Association Inc.	http://www.laptop.org/		×	n/a	n/a	n/a	n/a
Open Hardware Foundation	http://www.openhardwarefoundation.org/		×	n/a	n/a	n/a	n/a
Open Health Tools	http://www.openhealthtools.org/		■	■	■	×	n/a
Open Media Now! Foundation	http://www.openmedianow.org/		■	■	■	×	n/a
Open Source Applications Foundation	http://www.osafoundation.org/		■	■	■	×	n/a
Open Source Digital Voting Foundation	http://www.osdv.org/	1	■	■	■	×	n/a
Open Source For America	http://opensourceforamerica.org/about-osfa/our-mission/		×	n/a	n/a	n/a	n/a
Open Source Geospatial Foundation	http://www.osgeo.org/content/foundation/about.html	32	■	■	■	■	■
Open Source Initiative	http://opensource.org/		×	n/a	n/a	n/a	n/a
Open Source Software Institute	http://www.ossinstitute.org/		■	■	■	×	n/a
OpenBSD Foundation	http://openbsd.foundation.org/	7	■	■	■	■	■
Open Education Consortium	http://www.oecconsortium.org/about-oec/		×	n/a	n/a	n/a	n/a
OpenDoc Society	http://www.opendocsociety.org/		×	n/a	n/a	n/a	n/a
OpenID Foundation	http://openid.net/foundation/		×	n/a	n/a	n/a	n/a
OpenSourceMatters	http://www.opensourcematters.org/	1	■	■	■	■	■
Oregon State University Open Source Lab Alliance	http://osuosl.org/		■	■	■	×	n/a
Parrot Foundation	http://www.parrot.org/foundation/	1	■	■	■	■	■
Participatory Culture Foundation	http://pculture.org/		×	n/a	n/a	n/a	n/a
Peer-Directed Projects Center (freenode)	http://freenode.net/pdpc.shtml		×	n/a	n/a	n/a	n/a
Plone Foundation	http://plone.org/foundation	1	■	■	■	■	■
PostgreSQL Brasil	http://postgresql.org.br		■	×	■	×	n/a
PostgreSQL Europe	http://postgresql.eu		■	×	■	■	n/a
PostgreSQL US	http://postgresql.us	1	■	×	■	■	n/a
PostgreSQLFr.org	http://asso.postgresql.fr		■	×	■	■	n/a
Public Software Fund	http://www.pubsoft.org/	43	■	■	■	×	n/a
Python Software Foundation	http://www.python.org/psf/	1	■	■	■	■	■
Shuttleworth Foundation	http://www.shuttleworthfoundation.org/		×	n/a	n/a	n/a	n/a
Software Freedom Conservancy	http://sfconservancy.org/	46	■	■	×	×	n/a
Software Freedom Law Center	http://www.softwarefreedom.org/		×	n/a	n/a	n/a	n/a
Software in the Public Interest	http://www.spi-inc.org/	45	■	■	×	■	n/a
Software Libre Argentina	http://www.solar.org.ar/		■	×	■	■	n/a
Software Libre Chile	http://www.softwarelibre.cl/		■	×	■	■	n/a
Software Livre Brasil	http://www.softwarelivre.org/		■	×	■	×	n/a
Subversion Corporation	http://subversion.org/		■	■	×	■	n/a
Symphony Software Foundation	https://symphony.foundation/	1	■	■	■	■	■
TeX Users Group	http://tug.org/		×	n/a	n/a	n/a	n/a
The Open Planning Project	http://theopenplanningproject.org/		■	■	■	×	n/a
The Perl Foundation	http://perlfoundation.org/	1	■	■	■	■	■
The Software Conservancy	http://www.tsc.org/		■	■	■	×	n/a
Twisted Software Foundation	http://twistedmatrix.com/trac/wiki/TwistedSoftwareFoundation		■	■	×	×	n/a
TYPO3 Association	http://association.typo3.org/		■	■	■	×	n/a
Wikimedia Foundation	http://www.wikimediafoundation.org	1	■	■	■	■	■
Wikiotics Foundation	https://wikiotics.org/en/Wikiotics_Foundation	1	■	■	■	×	n/a
Wordpress Foundation	http://wordpressfoundation.org	1	■	■	■	×	n/a
X.Org Foundation LLC	http://www.x.org/wiki/XorgFoundation	1	■	■	■	■	■
Xiph.org	http://xiph.org/	23	■	■	■	×	n/a
XMPP Standards Foundation	https://xmpp.org/about/xmpp-standards-foundation.html		×	n/a	n/a	n/a	n/a
Zope Foundation	http://foundation.zope.org/	1	■	■	■	×	n/a

the ones with most impact in OSS, like the projects of the *Apache Software Foundation* (e.g., the Apache Web Server), the *Linux Foundation* (e.g., Linux Kernel, one of the top 10 most GitHub forked projects) or the *Symphony Foundation* (e.g., Symphony project, also one of the top 10 projects in GitHub with most code participation of the community).

73% of the analyzed foundations are specifically aimed at supporting software development efforts. Foundations not focused on software are mainly devoted to support and promote the open and free software movement.

directly related to their main tasks like *software*, *development*, *open* or *free*; the term *community* is also often highlighted in the mission descriptions, thus revealing the importance of this dimension. Other relevant terms with high frequency refer to the core promotion and support of OSS (e.g., *promote*, *support*, *defend* or *infrastructure*).

72% of the foundations targeting software development in our dataset have an international vocation and 78% are independent single software foundations. For the 57% of the software foundations with an explicit mission description, the community and defense of OSS are key concepts together with the development support.

3.2 RQ2: Scope of Software Foundations

Once identified the set of foundations targeting software development in our dataset, we study them to get some insights about their scope, according to three orthogonal dimensions, namely: geographical distribution, coverage and mission.

Geographical Distribution. This dimension studies the distribution of the foundations from a geographical point of view (i.e., whether the foundation has an international or local character). We detected 18 software foundations that were mainly focused on the development of local OSS communities. For instance, *El Centro de Software Libre* or *Software Livre Brasil* are two foundations whose activities focus on projects developed in Chile and Brasil, respectively. We are mainly interested in those foundations having an international distribution, as local ones are usually smaller and may adopt some behavior specific to the country they are located. Table 1, column GEO. DIST., marks with a black square the foundations with an international distribution (and a cross for the rest).

Coverage. Foundations can either serve a specific project, a set of projects or provide an umbrella for a number of smaller foundations that use it to simplify its own creation, management and legal processes. We found 14 foundations that mainly act as umbrella for others, like *Linux Foundation*, where inner organizations generally work in a stand-alone way (e.g., *OpenAPI initiative* or *Kubernetes* within the *Linux Foundation*); or as subsidiaries, like *Subversion Corporation*, which is part of the *Apache Software Foundation*. In the next research questions we focus on independent foundations. Table 1, column STRUCTURE, signals our individual foundations with a black square.

Mission. Beyond the development itself, foundations may aim to help the project on several aspects, like nurturing the community or facilitating the creation of new related projects. These goals are normally stated as part of the foundations' mission as mentioned either in the foundation website or in their bylaws. Surprisingly, a significant number of foundations did not provide any explicit information on their mission. According to our evaluation, 28 foundations did not include any reference to one or more of the following elements: mission, organization, projects or bylaws, being the last one the most difficult to find. Table 1, column MISSION, shows a black square for those foundations whose mission could be analyzed. For those, we collected the mission description and performed a term frequency study. Apart from the high frequency of terms

3.3 RQ3: Role in the Development Process

For the independent, international and transparent (in the sense of having enough information describing its mission and organization available) we study next whether the role of the foundations is limited to providing a legal and organization framework for the software project or if it goes beyond this and it directly dictates and monitors how the software is actually developed. Last column in Table 1 indicates the foundations considered (see the black squares), which are 18 in total.

We focus on four attributes, namely: communication, the process to become a project committer, the governance of the project, and the existence of a technical board. Table 2 summarizes the main results. Next we describe each dimension and report on the findings in more detail.

Communication. In OSS projects, communication plays a key role to disseminate the project and to help onboard new developers, also known as newcomers. For each foundation, we evaluate the communication channels as well as the resources provided to help newcomers to participate such as specific portals, indications to identify where help is required or the existence of a code of conduct.

The most common communication channel is the mailing list, although the use of forums is also widely spread. Besides, the great majority of the foundations provide specific documentation to help onboard newcomers. This lowers the entry barrier to any developer willing to contribute to the software projects covered by these foundations, paving also the way to the future promotion of some of them to core project committers.

Becoming committer. OSS projects capitalize on people that freely contribute to their development. Hence, retaining and capturing developers is crucial for their success. This is specially true for committers who are developers that have write access to the codebase and therefore have the key role to write (and/or review) new code to be committed to the project.

Our results reveal that becoming committer is in principle open to anyone interested in the project providing that has shown enough commitment and interest in it. The actual selection and evaluation process varies from foundation to foundation. Some of the decision-making mechanisms to select committers include: use of vouchers or some mechanism to gather support from other developers (e.g., *Apache Software Foundation* or *Mozilla Foundation*), a mentoring process (e.g., *Gentoo Foundation*) or based on the number

Table 2: Analysis of the development process of the selected foundations.

Foundation	Development									
	Communication		Becoming a committer		Development Governance					Technical Board
	Main	Newcomers	Who	How	Coord.	Impl.	Who	What	How	
Apache Software Foundation	ML	Y	Anyone	Voucher ^a	ML	BZ/Jira	Members	Special ^b	Special ^c	Y
Django Software Foundation	ML	Y	Anyone	By request	ML	Trac	Core devs.	Issues	Special ^d	Y
Document Foundation	ML	Y	-	-	BZ	Gerrit	-	Issues	Voting	Y
Eclipse Foundation	ML, F	Y	Special ^e	Voting ^f	BZ	BZ	Special ^g	Bugs	Voting	Y
Gentoo Foundation	ML, F	Y	Anyone	Mentoring	-	-	-	-	-	-
GNOME Foundation	ML, C	Y	-	-	BZ	BZ	Mentors	Issues	BDFL	-
Mozilla Foundation	ML, F	Y	Anyone	Voucher ^a	BZ	BZ	Special ^h	Issues	BDFL	Y
NetBSD Foundation	ML	Y	-	-	Custom ⁱ	Custom ⁱ	-	-	-	Y
Open Source Geospatial Foundation	ML	Y	-	-	-	-	-	-	-	-
OpenBSD Foundation	ML	N	-	-	Custom ⁱ	Custom ⁱ	-	-	-	-
OpenSourceMatters	ML	Y	-	-	ML, IT	IT	-	Issues	-	-
Parrot Foundation	-	-	Special ^e	By request ^j	ML, C	GH	-	Issues	-	Y
Plone Foundation	Forums	Y	Anyone	By PR	IT	GH	-	Issues	-	-
Python Software Foundation	ML	Y	Anyone	By PR	IT	IT, ML	Committers	Issues	Discussion	Y
Symphony Software Foundation	-	Y	Anyone	By PR	GH	GH	Leader	Issues, PR	BDFL	Y
The Perl Foundation	ML, C	Y	-	-	ML	-	-	-	-	-
Wikimedia Foundation	ML, C, W	Y	Anyone	Special ^k	Phabricator	Gerrit	Committers	Issues	Discussion	-
X.Org Foundation LLC	ML	Y	Special ^e	Sponsoring	ML, Web	ML, BZ	Reviewers	Patches	Discussion	-

Legend: ML = Mailing-list, F = Forums, C = Chat, W = Wiki, BZ = Bugzilla, IT = Issue-Tracker, GH = GitHub, PR = Pull Request, BDFL = Benevolent Dictator For Life.

Empty spaces mean that no information was gathered.

^a The developer has to gather *karma* (i.e., *Apache Software Foundation*) or *vouchers* (i.e., *Mozilla Foundation*) from other developers.

^b (a) code, (b) releases, (c) procedural.

^c (a) code by absolute majority with vetoes, (b) releases require 3 binding votes in favour, (c) procedural by majority rule.

^d Four fifths majority (technical board has veto right).

^e Any developer with a clear track of contributors in the project.

^f Committers vote (at least 3 votes are required) and have veto. The project committee approves/rejects the final decision.

^g Committers and technical board.

^h Bugzilla component owners, module owners, release drivers, ultimate decision-makers.

ⁱ By command line (GNATS system for NetBSD Foundation).

^j Developer requests to be committer (or is nominated), architect approves and metacommitter applies.

^k Committers can apply for it but +2 committers have to be selected after showing merits and passing internal hiring standards.

and success of previous pull-requests (e.g., *Python Foundation* or *Symphony Foundation*).

Governance. Effective and precise prioritization of the development tasks is also a crucial activity in any OSS project. With this purpose, each foundation defines and applies its own set of governance rules to govern the projects under its umbrella. These governance rules describe how to contribute to the project and how decisions regarding the acceptance/rejection of such contributions are going to be made. We study how people coordinate (see COORD. column), the tool/s they use to do so (see IMPL. column) and how they govern the process.

As you can see in the Table, the definition of governance models is maybe the the most ignored issue in the analyzed foundations. They are not clearly specified in one single place. Instead, governance information is scattered across several documents (if available at all). According to our results, most of the foundations rely on issue-trackers to identify development tasks for their projects (see WHAT column), however, it is not clear how these tasks are prioritized, approved or refused (see WHO and HOW columns).

A remarkable exception is the *Apache Software Foundation* that clearly describes its development process where committees have the decision-making authority regarding the content and direction of the Apache projects. In turn, these committees are overseen by the foundation's board. Decision making processes normally follow a lazy consensus approach: a few positive votes with no negative

vote is enough for approval. Negative votes always include some explanation for the reason to reject and may include an alternative proposal to be discussed.

Mozilla Foundation also describes how its development process is governed. The governance of Mozilla modules is a meritocracy, where developers need to demonstrate their abilities and find someone in the community who has adequate authority and will vouch for his/her competence. Module development is governed by the module owners, while release management is decided by the release drivers. The full governance process is overseen by super-reviewers, who review code for its effects on the overall state of the development branch and its adherence to Mozilla coding guidelines; and ultimate decision-makers, who are trusted members of the community who have the final say in the case of disputes.

In some cases, the foundation explicitly states that it takes no part in the day-to-day activities of the project, like the statement done by the *Open Source Geospatial Foundation*, which reads *the foundation is not interested in controlling foundation projects*, thus clearly separating the foundation structure and the development process of its projects. However, even in those cases, the foundation may give some recommendations for the governance of its projects, like the rejection of the *benevolent dictator for life* governance role or the enforcement of code management systems.

Technical Board. Some software projects may employ boards composed of recognized developers to drive the main technical decisions that may raise during the development process. Becoming part of these boards is usually hard and requires demonstrating clear compromise and commitment to the project.

Around half of the analyzed foundations make use of technical boards. Typically, these boards focus on technical aspects with an advisory role and are not used as a “formal” bridge between the project and the foundation internal organization. An exception is Apache and Eclipse, where the project committee (i.e., the Project Management Committees, PMCs) periodically report to the board about the status of the project.

Most of the foundations provide communication means and useful information for newcomers, but have limited implication and influence in the software project day-to-day work and decision process.

3.4 RQ4: Openness

As a final aspect of our study, we wanted to see whether the foundations themselves (and not the projects they support) follow as well an open philosophy. Or, if in an apparent contradiction to their mission, they are closed organizations. To answer this question, we study the organization according to three main attributes, including the board, the membership and the meetings. It turns out that the internal organization of the analyzed foundations follows always a similar pattern, with only a few exceptions. We report on this in the following.

Board. In an organization, the board of directors is usually a recognized group of people who is responsible for the management and oversight the organization. The directors’ powers, duties and responsibilities are generally specified in the bylaws of the foundation, where it is also common to specify the size of the board, who can be elected and how they are chosen or removed. In our analysis, we study essential information about the board including its size, the term, who can be part of the board, how they make decisions and how they are elected/removed from their seats.

Table 3 summarizes this information. In general, the board of directors has a term of one year and is elected by the members of the organization; besides, board actions are taken by majority with a majority quorum. A variety of different mechanisms are put in place for electing and removing directors. Also, while the election of directors usually involves the members of the organization, their removal sometimes does not count on them (e.g., *Django Foundation*, *Symphony Foundation* and *X.Org Foundation*), thus restricting the members’ freedom.

The *Eclipse Foundation* is maybe the most peculiar one in our analysis. It classifies the organization membership into four main groups, namely: (a) strategic developers, (b) strategic consumers, (c) add-in providers and (d) committers. Each group has the right to elect/remove a number of corresponding directors for the board. The election for the first two groups is unanimously while for the last two groups is by single transferable vote. The removal, however, is usually by majority for the four groups. Finally, the decision-making mechanism depends on the action being taken

(see table footnotes), but simple majority is applied by default. This board organization shows a customized structure for the roles of the community, which may promote a better management of the organization.

Other issues worth noting are how some foundations select a special group of people to manage the board and the definition of causes for removing board members. Although they are called differently (e.g., *trustees*, *active members* or *committee members*) they share a common objective and can be considered the *elite* in the organization, as they have the power to modify the board. On the other hand, some of the foundations clearly establish the procedures to follow to remove directors from the board with/without a cause, thus proposing different decision-making mechanisms for each one.

Membership. Most organizations admit people to become members, thus allowing them to participate in the decision-making processes like the election of the board or other affairs of the corporation. We are interested in analyzing who can be part of the foundation membership as well as how they are elected/removed.

Table 4 shows the results of the analysis of the membership for the selected foundations. Decisions on new members usually relies on the current members, who participate in the election and removal processes. The decision-making mechanism applied differs across the foundations. Although some of them apply the well-known majority procedure, most of them rely on specific procedures, for instance, nomination mechanisms (e.g., *Open Source Geospatial Foundation* and *Parrot Foundation*), or simple agreement (e.g., *NetBSD Foundation* or *Python Software Foundation* for basic member level).

Meetings. Most foundations hold an annual meeting, where members and the board can manage the business and affairs of the corporation. These meetings are also used to renew the board and elect other possible committees of the foundation. We study how frequent these meetings are and who can take part in them (Table 5). In the vast majority of foundations, any member can participate in the meeting and meeting decisions are based on a majority procedure with a quorum. However, most bylaws do not specify whether decisions made in these meetings can involve aspects that directly affect the development processes of their OSS projects. Only *Apache Software Foundation* defines the purpose of the meetings, which includes a revision of the status of every project, the election of developers for the committees or the approval of releases, among others.

The analyzed foundations show a high level of openness with most decision procedures based on member voting and democratic practices.

3.5 Additional Discussion Points

Beyond the main conclusions reported so far we would like to highlight some additional contributions derived from the results.

Utility of umbrella foundations for new projects. As we have seen, some foundations serve as umbrella for other smaller foundations or sets of related projects. For instance, the *Linux Foundation* clearly states that it *provides unparalleled support for open source*

Table 3: Analysis of the board for the selected foundations.

Board										
Foundation	Size	Term	Actions		Election			Removal		
			How	Quorum	Who	How	Quorum	Who	How	Quorum
Apache Software Foundation	9	1 year	M	M	Members	M	-	Members	M	-
Django Software Foundation	3 – 15	1 year	M	M	Members	M	-	Directors	M	-
Document Foundation	7	2 years	SM	Half	Trustees ^a	STV	-	Trustees	STV	-
Eclipse Foundation	> 1	Vary ^b	SM ^c	SM	Vary ^d	Vary ^e	-	Directors	Vary ^f	SM ^g
Gentoo Foundation	5	1 year	SM	M	Members	Condorcet	-	Members	M	-
GNOME Foundation	Not fixed	1 year	M	M	Members	STV	-	Board	-	-
Mozilla Foundation	5 – 15	1 year	M	M	Board	M	M	Board	M	Full
NetBSD Foundation	3 – 9	2 years	SM	M	Committee ^h	SM	25%	Active Members	85%	25%
Open Source Geospatial Foundation	5 – 9	1 year	M	M	Members	M	M	Vary ⁱ	Vary ^j	-
OpenBSD Foundation	> 3	1 year	M	-	Members	M	M	Members	M	M
OpenSourceMatters	Special ^k	1/2 year	M	M	Team leaders	M	-	Members	2/3	-
Parrot Foundation	> 2	1 year	M	M	Members	-	-	Members	M	-
Plone Foundation	–	1 year	M	M	Members	M	-	Members	M	-
Python Software Foundation	3 – 11	Vary ^l	M	M	Members	Vote ordering ^m	-	Board	2/3	-
Symphony Software Foundation	21 – 29	Vary ⁿ	M	M	Board ^o	M	-	Board	-	-
The Perl Foundation	> 1	2 years	M	M	Board	-	-	Board	M	M
Wikimedia Foundation	>= 9	3 years	M	M	Board	M	-	Board	M	-
X.Org Foundation LLC	< 8	1-2 years	M	M	Members	Vote ordering ^p	-	Board	2/3	-

Legend: M = Majority. SM = Simple Majority. STV = Simple majority with single transferable vote + Meek Method.

Empty spaces mean that no information was gathered.

^a Trustees are developers who have verifiably contributed time and intellectual work.

^b (a) Strategic developer and consumer directors until their removal (b) Add-In directors and committer directors annually.

^c (a) Simple majority as default, (b) unanimous consent for license-related issues, (c) super-majority consent for foundation-related issues, (d) unanimously when no meeting, (e) membership approval for some actions.

^d (a) Strategic developer directors by strategic developers, (b) strategic consumer directors by strategic consumers, (c) add-in provider directors by add-in provider members and (d) committer directors by committer members.

^e (a) Strategic developer directors unanimously, (b) strategic consumer directors unanimously, (c) add-in provider directors and committer directors by single transferable vote.

^f (a) With cause by majority, (b) with no cause by two-thirds of the directors.

^g Good Standing directors (or one-half when even number required).

^h 10 voting volunteers from active members.

ⁱ (a) Without cause: member entitled to vote, (b) with cause: board.

^j (a) Without cause: two thirds, (b) with cause: majority of directors (except the one being removed).

^k Number of departments.

^l Cohort A directors have three (3) years beginning. Cohort B directors have two (2) years. Cohort C directors have for one (1) year.

^m Candidates receiving the highest number of votes shall be designated cohort A directors, the four receiving the next highest number of votes shall be designated cohort B directors, and the three receiving the third highest number of votes shall be designated cohort C directors.

ⁿ Depends on director's type.

^o Within their types.

^p The four (4) members receiving the highest vote totals will be considered an elected director and each such director will hold office for a term of two (2) years. members receiving the next highest vote totals required to bring the total number of directors to eight (8) will be considered an elected director and each such director will hold office for a term of one (1) year.

communities through financial and intellectual resources, infrastructure, services, events, and training. We believe that this kind of foundations are especially useful for young OSS projects as they provide the required scaffolding and liberate them from organizational issues while freeing them also from the burden of setting up their own foundation. Also, they create an ecosystem where projects and developers can easily collaborate together, which may help them to boost their development.

Weak alignment between the foundation and the project's concrete development practices. Our study has revealed that, while the organization itself is well defined, this organization does not generally extends to the software projects that depend on it. As we have commented before, as the projects grow, there is the need to put in place adequate structures to manage the community around the project and optimize its contributions. Since it has been reported before that projects do not typically take care of this issue themselves, we were hoping that, foundations could play that role.

Unfortunately, this does not seem to be the case. We believe this is a clear aspect where foundations could help. A tighter integration between the foundations and the project could help projects benefit from the organizational knowledge available at the foundation to decide the best governance and contribution model for the project. This does not necessarily mean that the project should strictly follow the internal organization of the foundation itself but that at least the project could use some of the know-how available there.

Lack of precise documentation. One of the main issues we found in our study was the lack of precise documentation about how the foundation works. This may scare away some potential contributors that want to first clearly understand how their effort (e.g., in terms of time invested in coding patches for reported bugs) will be evaluated. Conversely, sometimes the information was abundant but unclear and scattered (e.g., the *Wikimedia Foundation*), thus causing over-documentation issues and confusion, specially since it is easy that at some point this redundant documentation becomes partially

Table 4: Analysis of the membership for the selected foundations.

Foundation	Members						Special
	Election			Removal			
	Who	How	Quorum	Who	How	Quorum	
Apache Software Foundation	Members	M	-	Members	2/3	-	-
Django Software Foundation	Board	M	-	Board	M	-	-
Document Foundation	Trustees ^a	STV	-	Trustees ^a	STV	-	2 years term
Eclipse Foundation	None	Paid	-	Board	2/3	-	-
Gentoo Foundation	Trustees ^a	Special ^b	-	Trustees ^a	M	-	-
GNOME Foundation	Board	M	-	Board	M	-	2 years term
Mozilla Foundation	None ^c	-	-	-	-	-	-
NetBSD Foundation	None	Agreement ^d	-	-	-	-	-
Open Source Geospatial Foundation	Members	Nomination ^e	-	Members	2/3	-	-
OpenBSD Foundation	Board	M	-	-	-	-	-
OpenSourceMatters	Members	2/3	-	Members	Unanimously	-	-
Parrot Foundation	Members	Special ^f	-	Committee	Time-based ^g	-	-
Plone Foundation	Board	M	-	Board	2/3	-	1 year
Python Software Foundation	Board	Special ^h	Board	Members	2/3	-	-
Symphony Software Foundation ...	None	Paid	-	Board	SupM	-	-
The Perl Foundation	None ^c	-	-	-	-	-	-
Wikimedia Foundation	None ^c	-	-	-	-	-	-
X.Org Foundation LLC	None	Agreement ^d	-	Board	3/4	-	-

Legend: M = Majority. SM = Simple Majority. SupM = Super Majority. STV = Simple majority with single transferable vote + Meek Method.

Empty spaces mean that no information was gathered.

^a Trustees are developers who have verifiably contributed time and intellectual work.

^b Any active developer can become member unless there is an absolute majority of trustees that opposes it.

^c There shall no be members.

^d Signing a developer agreement.

^e Nomination by other member and affirmative vote of the members.

^f Two nominations and affirmative vote of the board of directors.

^g The Membership Committee may terminate any membership, after a period of 12 months without contribution or voting activity from the member.

^h Depends of the member type, for basic level it's immediate. There are five levels, namely: (1) basic members, (2) supporting members, (3) sponsor members, (4) managing members, (5) contributing members and fellows.

outdated and even contradictory. To maximize the efficiency and impact of foundations, we believe a clear and concise information about all foundation aspects is a must.

No historical data publicly available. Most of the foundations analyzed in our study (with very few exceptions, like the *Apache Software Foundation*) do not provide easy means to access the assets tracking the foundation activity (e.g., minutes of general meetings, past composition of committees, historical list of projects supported, meetings hold by the committees, etc.). This makes really difficult to evaluate the real impact the organization has on open source since we can only have the current snapshot. No longitudinal studies can be done at this point. Therefore, we encourage foundations to make the effort to open more of its data for further analysis.

4 THREATS TO VALIDITY

Our work is subjected to a number of threats to validity, namely: (1) internal validity, which is related to the inferences we made; and (2) external validity, which discusses the generalization of our findings.

Regarding the internal validity, the dataset construction process mainly relies on the set of foundations identified by `flossfoundations.org`. To minimize the risk of missing foundations not listed there, we explored other online sources to extend the initial list of foundations.

Another threat to validity is our subjectivity in selecting and classifying the foundations of our dataset. A wrong perception or misunderstanding on our side of a given foundation may have resulted in a missclassification of the foundation regarding the reported research questions. To minimize the chances of this to happen, the study and classification of the foundations was validated by all authors, discussing any misalignment until reaching a full agreement.

As for the external validity, note that our study is based on a sample of software foundations we constructed and therefore our results should not be generalized to other kinds of foundations linked to (non-software) collaborative projects, like book writing efforts or similar initiatives also commonly used GitHub or other code hosting platforms.

5 RELATED WORK

Research on OSS development has been widely addressed in the scientific literature specially given the growing number of organizations adopting open source practices [27]. Crowston et al. [8] review the empirical research on Free/Libre and Open-Source Software (FLOSS) development and assess the state of the literature. Likewise, Cosentino et al. [6] perform a systematic mapping study of all research works studying software development aspects in the context of the GitHub code hosting platform. As an example, there are a number of works studying development models [23], diversity of developer teams [29], community structures [4], documentation procedures [1] or collaboration practices [10], among many others.

Table 5: Analysis of the meetings hold by the selected foundations.

Foundation	Meetings		Actions		
	Annual	Special			
			Who	How	Quorum
Apache Software Foundation	Y	Y	Members	M	1/3
Django Software Foundation	Y	Y	Members	M	M
Document Foundation	-	-	-	-	-
Eclipse Foundation	Y	Y	Members	SM ^a	SM ^a
Gentoo Foundation	Y	Y	Members	M	1/3
GNOME Foundation	Y	Y	Members	M ^b	-
Mozilla Foundation	Y	Y	Board	M	M
NetBSD Foundation	Y	Y	Active Members	M	1/4
Open Source Geospatial Foundation	Y	Y	Members	M	M
OpenBSD Foundation	Y	Y	Members	M	M
OpenSourceMatters	Y	-	Members	M	M
Parrot Foundation	Y	Y	Members	-	1/10
Plone Foundation	Y	Y	Members	M	M
Python Software Foundation	Y	Y	Members ^c	M	1/3
Symphony Software Foundation	Y	Y	Platinum Members ^d	M	51%
The Perl Foundation	Y	Y	Board	M	M
Wikimedia Foundation	Y	Y	Board	M	M
X.Org Foundation LLC	Y	Y	Members	M	1/4

Legend: M = Majority. SM = Simple Majority. SupM = Super Majority. STV = Simple majority with single transferable vote + Meek Method.

Empty spaces mean that no information was gathered.

^a Each member has 1 vote (except committers of the same organization, who will have only 1).

^b The vote is usually cast by voice (or by ballot when required by the chairman). Fractional votes are not allowed

^c Basic members are not included.

^d There are four member levels, according to the fee paid, namely: Platinum, Gold, Silver, At-Large.

Nevertheless, the role of open source foundations to organize and coordinate open source projects has been largely ignored. Previous works on the topic of OSS organization have mainly focused on the (self-)organization of developer communities, for instance, by analyzing the social ties [2] or the blossoming of spontaneous collaboration networks [4, 7, 13] as the project grows and evolves. This evolution over time has also been covered by Panichella et al. [22] and Joblin et al. [14]. Additionally, a number of works have studied different coordination mechanisms that may be put in place to organize the work [9, 12, 15]. The formalization of such collaboration is typically done by means of defining the governance model of the project [19, 21]. Nevertheless, the explicit definition of governance models is still mostly missing in OSS projects, as analyzed by Cánovas Izquierdo et al. [18].

Among the few exceptions, we would like to mention the work of O'Mahony [20] and Riehle [24] (that discusses the role of software foundations in OSS from a more economic/business perspective), the studies of Lindman and Hammouda [17] (performing a qualitative analysis of how foundations support OSS projects) and Riehle and Berschneider [25] (evaluating six foundations to build a comparative model that should help project leaders understanding the differences between existing foundations). These studies (especially the latest two) are useful to complement, from a more qualitative point of view, our RQ3 and RQ4 research questions even if our work covers a larger number of foundations and dimensions.

Some of the dimensions studied in this paper have been individually addressed by other works but at the project level. For instance, Tourani et al. [28] study the importance of codes of conduct in OSS, where instructions on how to be part of the project community or to become committer are usually included. Guzzi et al. [11] study

the role of communication channels in the development process of OSS, in particular, the use of mailing lists.

6 CONCLUSION

We have studied 89 software foundations to better understand the role they play in the development of open source software projects. Our results show that very few foundations are devoted to provide complete support (i.e. including guidelines on the governance and contribution rules they should follow) to the projects they endorse. Their missions is more directed towards providing legal support and leading evangelizing actions.

As further work, we would like to compare the role foundations play in OSS with the role they have in other kinds of non-governmental organizations (and non-profit organizations in general) where the distinction between the foundation and the organization itself (the software project in our case) is much blurred. Since foundations in OSS are still rather young, we hope that by learning from more established foundations in other fields, as well as best practices of the most senior ones in OSS, we can bring back some useful recommendations for OSS projects interested in setting up its own foundation (or looking for a established one they could join). A more qualitative study, comprising interviews with actual users and contributors of open source projects will also help to explore their opinions and views on the needs and expectations they have from software foundations.

ACKNOWLEDGMENTS

This work has been supported by the Spanish government (TIN2016-75944-R project).

REFERENCES

- [1] Karan Aggarwal, Abram Hindle, and Eleni Stroulia. 2014. Co-evolution of Project Documentation and Popularity within GitHub. In *Int. Conf. on Mining Software Repositories*. 360–363.
- [2] Mohammad Y. Allaho and Wang-Chien Lee. 2013. Analyzing the Social Ties and Structure of Contributors in Open Source Software Community. In *Int. Conf. on Advances in Social Networks Analysis and Mining Analyzing*. 56–60.
- [3] Christian Bird, Alex Gourley, Prem Devanbu, Anand Swaminathan, and Greta Hsu. 2007. Open Borders? Immigration in Open Source Projects. In *Int. Conf. on Mining Software Repositories*. 6.
- [4] Christian Bird, David Pattison, Raissa D Souza, Vladimir Filkov, and Premkumar Devanbu. 2008. Latent Social Structure in Open Source Projects Categories and Subject Descriptors. In *Int. Symp. on Foundations of Software Engineering*. 24–35.
- [5] European Commission. 2015. The Economic and Social Impact of Software & Services on Competitiveness and Innovation. (2015).
- [6] Valerio Cosentino, Javier Luis Cánovas Izquierdo, and Jordi Cabot. 2017. A Systematic Mapping Study of Software Development With GitHub. *IEEE Access* 5 (2017), 7173–7192.
- [7] Kevin Crowston, Qing Li, Kangning Wei, U. Yeliz Eseryel, and James Howison. 2007. Self-organization of Teams for Free/Libre Open Source Software Development. *Information and Software Technology* 49, 6 (2007), 564–575.
- [8] Kevin Crowston, Kangning Wei, James Howison, and Andrea Wiggins. 2012. Free/Libre open-source software development: What We Know and What We Do Not Know. *Comput. Surveys* 44, 2 (2012), 1–35.
- [9] Kevin Crowston, Kangning Wei, Qing Li, U. Yeliz Eseryel, and James Howison. 2005. Coordination of Free/Libre Open Source Software Development. In *Int. Conf. on Information Systems*.
- [10] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. In *Conf. on Computer Supported Cooperative Work*. 1277–1286.
- [11] Anja Guzzi, Alberto Bacchelli, Michele Lanza, Martin Pinzger, and Arie van Deursen. 2013. Communication in Open Source Software Development Mailing Lists. In *Int. Conf. on Mining Software Repositories*. 277–286.
- [12] James D. Herbsleb and Rebecca E. Grinter. 1999. Splitting the Organization and Integrating the Code: Conway's Law Revisited. In *Int. Conf. on Software Engineering*. 85–95.
- [13] Qiaona Hong, Sunghun Kim, S. C. Cheung, and Christian Bird. 2011. Understanding a Developer Social Network and its Evolution. In *Int. Conf. on Software Maintenance*. 323–332.
- [14] Mitchell Joblin, Sven Apel, and Wolfgang Mauerer. 2017. Evolutionary Trends of Developer Coordination: A Network Approach. *Empirical Software Engineering* 22, 4 (2017), 2050–2094.
- [15] Robert E. Kraut and Lynn A. Streeter. 1995. Coordination in Software Development. *Comm. ACM* 28, 3 (1995), 69–81.
- [16] Sang-Yong Tom Lee, Hee-Woong Kim, and Sumeet Gupta. 2009. Measuring Open Source Software Success. *Omega* 37, 2 (2009), 426–438.
- [17] Juho Lindman and Imed Hammouda. 2017. Investigating Relationships Between FLOSS Foundations and FLOSS Projects. In *Int. Conf. on Open Source Systems*. 14–22.
- [18] Cánovas Izquierdo Javier Luis and Jordi Cabot. 2015. Enabling the Definition and Enforcement of Governance Rules in Open Source Systems. In *Int. Conf. on Software Engineering, Software Engineering in Society track*. 505–514.
- [19] M. Lynne Markus. 2007. The Governance of Free/Open Source Software Projects: Monolithic, Multidimensional, or Configurational? *Journal of Management & Governance* 11, 2 (2007), 151–163.
- [20] Siobhán O'Mahony. 2007. *Non-Profit Foundations and their Role in Community-Firm Software Collaboration*. MIT Press, 393–414.
- [21] Siobhán O'Mahony and Fabrizio Ferraro. 2007. The Emergence of Governance in an Open Source Community. *Academy of Management Journal* 50, 5 (2007), 1079–1106.
- [22] Sebastiano Panichella, Gerardo Canfora, Massimiliano Di Penta, and Rocco Oliveto. 2014. How the Evolution of Emerging Collaborations Relates to Code Changes: An Empirical Study. In *Int. Conf. on Program Comprehension*. 177–188.
- [23] Martin Pinzger and Arie Van Deursen. 2014. An Exploratory Study of the Pull-based Software Development Model. In *Int. Conf. on Software Engineering*. 345–355.
- [24] Dirk Riehle. 2010. The Economic Case for Open Source Foundations. *IEEE Computer* 43, 1 (2010), 86–90.
- [25] Dirk Riehle and Sebastian Berschneider. 2012. A Model of Open Source Developer Foundations. In *Int. Conf. on Open Source Systems*. 15–28.
- [26] Ravi Sen, Siddhartha S. Singh, and Sharad Borle. 2012. Open Source Software Success: Measures and Analysis. *Decision Support Systems* 52, 2 (2012), 364–372.
- [27] Diomidis Spinellis and Vaggelis Giannikas. 2012. Organizational adoption of open source software. *Journal of Systems and Software* 85, 3 (2012), 666–682.
- [28] Parastou Tourani, Bram Adams, and Alexander Serebrenik. 2017. Code of Conduct in Open Source Projects. In *Int. Conf. on Software Analysis, Evolution and Reengineering*. 24–33.
- [29] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark G. J. van den Brand, Alexander Serebrenik, Premkumar T. Devanbu, and Vladimir Filkov. 2015. Gender and Tenure Diversity in GitHub Teams. In *Conf. on Human Factors in Computing Systems*. 3789–3798.
- [30] Minghui Zhou and Audris Mockus. 2015. Who Will Stay in the FLOSS Community? Modeling Participant's Initial Behavior. *Software Engineering, IEEE Transactions on* 41, 1 (2015), 82–99.