DRAFT V. 0.7 – WSL2012

Free Software trajectories: from organized publics to formal social enterprises?

Morgan Currie[1], Christopher Kelty[2], Luis Felipe Rosado Murillo[3]

[1]Department of Information Studies, University of California, Los Angeles

[2]Center for Society and Genetics, University of California, Los Angeles

[3]Department of Anthropology, University of California, Los Angeles
{msmorgancurrie,ckelty,luisfelipe}@ucla.edu

Abstract. *By looking at the history of long-lasting and successful Free and Open Source Software (FOSS) projects, one can observe a common trajectory: they tend to start with a few core developers, then increase in code base size, complexity, and number of contributors and users, finally creating the necessity of a formal organization to help coordinate the development efforts, secure funding, manage donations, seek partnerships, manage hosting infrastructure, and struggle to protect its members from patent and copyright disputes. The question we discuss in this paper is "what are the characteristics of participation in those projects that do not describe the common trajectory – which is to start from a small community to subsequently constitute formal social enterprises (non-profit organizations or companies)?"*

1. Introduction

By looking at the history of long-lasting and successful Free and Open Source Software (FOSS) projects, one can observe a common trajectory: they tend to start with a few core developers, then increase in code base size, complexity, and number of contributors and users, finally creating the necessity of a formal organization to help coordinate the development efforts, maintain hosting infrastructure, secure funding, manage donations, seek partnerships, and struggle to protect its members from patent and copyright disputes. The question we discuss in this paper is "what are the characteristics of participation in those projects that do not describe the common trajectory – which is to start from a small community only to subsequently constitute formal social enterprises (non-profit organizations or companies)?"

In order to respond to this question, we will compare projects with different trajectories: both those that were initially sponsored by a company and then created a community around it, and those that never constituted (or refuse to constitute) a formal social enterprise. By addressing this question, we will highlight fundamental differences and similarities between projects: what makes them grow stronger or fail to attract, foster collaboration, and further forms of public participation. In order to establish parameters for comparison, five dimensions of FOSS projects will be compared and discussed: 1) *project geneology*; 2) *tasks* (how are they defined, described, and distributed?); 3) *alliances* (who are the partners? Are they from the public sector, private sector, or both?); 4) *governance* (is there a formal procedure for decision-making? If not, how are decisions made?) 5) *availability* (which licenses are used? What is the rationale behind the decision of using a particular license?). We will explore the following projects in order to respond to the questions above: Dyne.org, Debian, Mozilla, and Xara Extreme Linux.

This paper is based on research data from the project “Birds of the Internet”, sponsored by National Science Foundation (NSF), and hosted at the Center for Society and Genetics (UCLA). The project examines the nature of participation in FOSS and Internet-based projects and has as its primary goal to understand social, political, and technical conditions for the promotion, sustainability, and expansion of public participation.

2. Free and Open Source Software Trajectories

For three decades now Free and Open Source Software (FOSS) has generated an intense and intricate dispersion of technical objects and practices based on international collective efforts. Recent anthropological and sociological accounts of Free Software as a political, technical and cultural practice further investigate the current transformation of the modernist notion of individual property over intangible goods and the opposition created by FOSS to the advancement of the transnational intellectual property regime (Kelty 2008; Coleman 2005; Leach 2009). FOSS offers a counterpoint, made possible by the virtue of its licensing schemes: the constant rebuilding effort over a set of public software licenses that permit (re)distribution, free use and adaptation of software code. The resulting phenomena are situated in between, at least, two main registers: the general reciprocity oriented towards the free circulation of software code as public good, and the market economy in which computer technicians offer their FOSS expertise.

From an anthropological standpoint, FOSS is curiously made up by boundary practices in a multitude of social ties and sociotechnical arrangements, bringing together people, associations, and technical objects: it is a craft that is hard to analyze without problematizing the boundaries of our established categories and their taken-for-granted oppositions between individual/society, private/public, gift/market, persons/objects, work/leisure, code/expression, material/immaterial, and discourse/practice. In this sense, Free Software is better approached as a *quasi-object* assuming different forms in intersecting recursive publics (Kelty 2008). Public administrators, for instance, may declare that FOSS is a tool for social change, given its potentiality in facilitating digital inclusion; among computer hackers, it is often defined as a highly valued expression of oneself and his or her technical competence; for artists and free culture activists, it is signified as a set of tools to empower cultural production.

This paper analyzes FOSS projects' participatory structures with informally negotiated or legally formalized aspects that relate to their growth over time. As suggested by Coleman (2005), "most FOSS projects in their infancy, including Debian, operated without formal procedures of governance and instead were guided by the technical judgments of a small group of participants" (Coleman 2005, p. 325). Formalization therefore typically comes about to address issues of scale and management. Riehle and Deshpande (2006) demonstrated that FOSS projects increased in size exponentially between 1998 and 2006, since "the total amount of source code and the total number of projects double about every 14 months" (Riehle and

Deshpandep 2006, p.11). As projects scale up, more is at stake between open source and proprietary development models (Lakhani 2007; West 200x). As FOSS projects grow they tend to organize their activities into businesses, NGOs and foundations to coordinate software development work and manage intellectual property rights, profit and fund-raising purposes. Spontaneous gatherings of half a dozen hackers become formal organizations over time, transforming substantially the very social fabric which constitutes software development projects.

In our analysis of Internet-based participatory projects more generally (Fish et al. 2012) we proposed two distinct entities which are generally present: first is the "Formal Social Enterprise" (FSE) - legal organizations with formal decision-making procedures that are composed of, at least, one contractually obligated employee. On the other end of the spectrum are the "Organized Publics" (OP), or the community of participants whose relation to the FSE is informal, voluntary, impermanent, and often financially uncompensated. In the case of Linux, for instance, the Linux Foundation is a legal entity, an FSE, while the Linux kernel hackers form its OP base. The foundation is responsible for managing the Linux trademark, manage general Intellectual Property disputes, organize conferences and training sessions across global locations and organize meetings with sponsors and partners. On the other hand, the OP is technically and structurally connected to the FSE in different ways (such as being board members), but are not contractual members of it - they can usually come and go as they like. As we have pointed out, most FLOSS projects start with a small OP then overtime develop a FSE. But there are cases of the opposite trajectory: a company launches a FOSS project and then gather a thriving OP over time. Also, the composition of power within and between these two groupings is not a foregone conclusion in any particular instance: the FSE for one FOSS project could be a powerful, hierarchical conglomerate, with its OP structured by decentralized, horizontal communication - Android and its Open Source Project falls here - while alternatively a not-for-profit FSE could be composed of one or two people who make little distinction between their efforts and the project's OP.

Table 1. FOSS Trajectories:

OP → X	OP → FSE	FSE → OP	FSE → X
Dyne.org SPIP Gimp	Linux Ourproject.org GNOME Apache **Debian** KDE Python PostgreSQL Drupal PHP Inkscape Tor	**Mozilla** Redhat Android MySQL LibreOffice Ubuntu SugarCRM Mediawiki Kaltura Eclipse Blender	Symbian OpenSolaris **Xaraxtreme**

Why do initially informal FOSS Organized Publics (OP) tend towards formalization into Formal Social Enterprises (FSEs)? In some cases foundations embody the formalization of FOSS projects at the managerial and decision-making level, often in efforts to broker commercial deals or tackle legal issues. In the discussion of the role of foundations in the FOSS context, O'Mahoney (2005) points out that the foundation emerges when projects become commercially relevant and profitable from the perspective of the industry, by producing software as service; foundations facilitate the interaction between community projects and corporations. O'Mahoney cites a Fortune 100 executive who once stated: "How do I make a deal with a webpage?" (O'Mahoney 2005, p. 396), referring to his doubts concerning the Apache project and its possibilities of establishing contracts with corporations. For non-profit projects, formalization as a 501(c)3 also allows projects to accept donations, apply for grants, and to centralize management of internal conflicts or intellectual property rights disputes.

But what about exceptions to this transformation towards formalization of Free Software projects? Below we introduce preliminary methods to analyze both typical cases as well as three other trajectories that FOSS projects can take as they develop (or fail to do so). We also propose five dimensions of comparison between these trajectories, as well as offer a theoretical approach for analyzing participants' social, legal and political capacities to participate in these projects. The running hypothesis is that these five categories can possibly unveil disjunctures of power between participants that tend to be overlooked: internal disputes and politics, questions of governance, language practices, history and genealogy, and the role of institution in helping or hampering the development of a software project. But, before outlining these methods, we need to consider how a normative theory of participation can be defined, by turning to historic normative definitions that will instruct a similarly nuanced approach fitting for online participation as it occurs today.

3. The Problem of Participation

The expression 'online participation' seems to be everywhere, in a flurry of scholarly and pop culture accounts that try to understand why communities form around blogs, file sharing, software development, internet forums, news portals, etc. The popular conception is of a massive, shared resource used by highly cooperative communities who put in significant time with no monetary gain. Yet closer inspection yields a very different portrait: the limits and possibilities of participation are structured both legally and sociotechnically and vary greatly across projects. If we want to obtain a finer picture of how participation occurs across FLOSS projects, we must advance a research agenda for empirical studies of participation as a core problem of the social ties binding projects, technical objects, and people. As a complementary effort, we seek to define a *normative* theory of participation, an otherwise ambiguous and overdetermined word. Instances of participation are in no way uniform, and conceptions of its scope form along a continuum of stronger to weaker degrees. This is crucial to define, since the strength of any adopted theory of participation will bear out directly on the technical and legal architectures developed by a project or organization.

To formulate a normative approach, we can turn to historical interpretations of what participation can and should be. In classical democratic theory, for instance, the standard debate over strong or weak participation falls over the issue of representation. Should citizens elect leaders whose job it is to make decisions on their behalf? Or should citizens collectively and directly make these choices about their society? Arnold S. Kaufman, writing in 1960, tackles Joseph Schumpeter's argument that direct democracy contradicts human nature, which Schumpeter sees as essentially irrational; people, in his view, simply can't sift intelligently through information or know what is best for themselves. Schumpeter is suspicious of crowd behavior, of clamoring, chaotic voices in aggregate, and instead favors representation by elected leaders. But Schumpeter does not provide a framework for *how* these leaders should be produced. Kaufman points out that direct democracy is precisely the mechanism needed to produce these rational leaders. His take is Rousseauian: crucial to Kaufman's strong interpretation of participatory democracy is that deliberation must occur, allowing each participant an equal say in the matter and so allowing them to accrue critical thinking skills through meaningful debate. For "it is only when men acquire direct responsibility for a certain range of decisions that social imagination breaks through its parochial barriers and envisiages larger psosibilities." (189) In this schema, while all voices will be considered of equal value, Kaufman also makes room for more or less influential voices - a person who "has special knowledge about a problem or...thinks more clearly or more imaginatively about certain issues." (192) While this person's formal power is on par with other participants, her influence may have more suasion on the final outcome. Kaufman's theory is normative because he believes that direct participation in democracy - a stronger conception than Schumpeter's representative sort - will produce better individuals.

The question becomes how to implement a strong theory of participation. Participatory democracy (PD), a concept first touted by the Students for Democratic Society (SDS) in their 1962 Port Huron statement, propounded two critical ideas that were tested on the ground during the Civil Rights struggles in the US. Like Kaufman the SDS claim that individuals must have a say in the decisions that determine their quality and choices in life. Also crucial is that societal structures should be set up, from the beginning, to encourage its citizens to offer their own opinions and help individuals feel comfortable expressing themselves when they enter into free political discussion. Just as importantly, society must provide the media platforms citizens need for self-expression. In keeping with these ideas, SDS ghetto projects in the North modified their own organizations by abolishing central offices, rotating leaders, and allowing executive committees to be checked by staff meetings. Equal say among citizens, not the decisions of a charismatic individual or representative middle person, should set the agenda.

Straughton Lynd points out the concept of 'parallelism' implicit in participatory democracy: as it operated in the 1960s, its radical structures moved alongside older, more conservative institutions. PD formed a challenge from within by building an alternate, steady enclave of critique that rejected conventional coalition politics. (6) But if these new institutions remain separate from the mainstream political process, Lynd questions if their lack of formalization will be able to address the such basic needs as feeding people and raising them from poverty. He concludes with a hopeful "perhaps": PD can possibly make change by increments through their suggestion of wider possibilities, as their more radical social constructions slowly transform the norms of the existing institutions they operate alongside.

Another direct application of democratic theory occurred in the industrial democracy movements of the late 19th and 20th centuries. Industrial democracy (ID) took inspiration from classic democratic theories on civic participation forged in political struggles a decade earlier in France, Germany, and the US. Articulated by thinkers such as Rousseau, John Stuart Mill and G. D. H. Cole, these theories produced concepts and strategies for workers inside late 19th and 20th century industrial factories who sought more equitable divisions of labor between themselves and management. ID, in essence, set out to modify the conventional hierarchy that traditionally allocated decision making entirely to management while denying workers any role beyond their labor tasks.

Industrial Democracy's proponents expressed ideas that fall on a strong-to-weak a continuum of participation. Carol Pateman's (1970) history of this movement spells out these distinctions, by calling attention to the levels of power allocation possible among workers and management. She distinguishes ID entirely from "pseudo participation," merely a persuasive style of management that gains workers support for decisions already made. Many so-called ID "participation" experiments, claims Pateman, took this spurious form. With partial democracy, in contrast, Pateman describes how workers have "influence" but not equal power to make final decisions, both over what goes on on the shop floor, and over the enterprise as a whole (such as matters of investment, marketing, etc). Finally, with full participation, workers are part of "a process where each individual member of a decision-making body has equal power to determine the outcome of decisions'." (71) Pateman finds examples of this system at work, at least at the lower shop-floor level, in the collective contracts found in mining and car industries, where workers operated in unsupervised, self-regulating groups to determine their everyday work environments. A fully socialized form of ID at the level of administration is only hinted at: most ideal is "a system of industrial democracy implies the opportunity for full higher level participation by employees in the formal organization." (71)

These theories of participation certainly do not graft directly onto the distributed labor of coders in contemporary societies. FOSS participants do not work under one roof (typically); they are not bargaining for better working conditions in a situation from which they have little recourse. What we *can* take from these theories is an appreciation for finer distinctions; the degrees between partial and full participation present a greater spectrum of possibility for laborers to bargain for autonomy and political efficacy. Below, we propose a set of metrics that define the spectrum of weak vs. strong participation in the online forms it takes today, by considering issues such as goal-setting, governance, and the availability of resources.

Variable	Strong	Pseudo or Weak
Decision-making	In goals, not only tasks	Solely in tasks designed or framed elsewhere
Availability	Collective control and/or individual access to the resource produced by participation	Expropriation and private ownership of resource produced
Governance	Capacity to exercise both exit (without penalty) and voice (without fear) vis-à-vis a known and addressable entity	No capacity to exercise both voice, or a risk of loss of some kind upon exit
Educative Dividend	An “educative” dividend from participation	No “educative” dividend from participation, or worse, an actual negative dividend (loss of privacy, loss of money, fraud, loss of reputation/trust)

With this spectrum of possibilities, we expect to gain a deeper understanding of what is at stake beyond issues of intellectual property that typically consume the legal analyses of FOSS projects. Conceptions of full participation - where participants take part in goal setting, governance decisions, and have full availability of the resources generated by the project, serve as a standard for strong participatory projects. Whether these standards bear on the project's success and development, as well as influence the structures it takes, we hope to illuminate in the aforementioned case studies.

4. Methods

For the analysis of FOSS project trajectories, we created a *corpus* of qualitative data composed of hypertext documents, mailing list archives, video and audio interviews, presentations regarding the project, and scholarly publications for four projects: Dyne.org (OP), Debian (OP to FSE), Mozilla (FSE to OP) and Xaraextreme (FSE).

For our purposes, the FSE/OP distinction will be used to analyze the aforementioned cases representing four different trajectories of FOSS projects. To establish parameters of comparison which highlight how participation occurs in FOSS projects, we consider five interdependent variables. The first two are genealogical, the remaining three account for patterns of social relationship and collective action:

> **a. Project Geneology:** a genealogical description of the project's origins and major shifts in its composition up to the present day.
>
> **b. Alliances:** who are partners and/or sponsors? Are they from the public sector, private sector, or both? Do these alliances have any formal or informal roles in the FSE?

The following variables bear directly on the political dynamics of participation in FLOSS projects and aim to distinguish the levels of privileges and types of roles granted to OPs by the FSE. The following dimensions can be placed on a normative continuum of 'weak to strong' architectures of participation, as defined above:

> **c. Tasks and Goals:** How are tasks defined, described, and distributed? Does the OP participate in goals, not only tasks? Can participants engage in discussion with leaders, managers, and administrators about what tasks should be pursued, how they should be structured, or how they should be measured? A strong theory of participation would at least grant publics access to the decisions made by FSEs through representation, if not direct voice and/or vote. It would also consider educative dividend: by participating in the tasks and/or goals, do participants learn what their own interests are? Do they develop civic virtue, a sense of responsibility or a refined sense of liberty by being directly involved?
>
> **d. Governance:** Is there a formal procedure for decision-making? If not, how are decisions made? Do participants have the capacity to exercise both exit (without penalty) and voice (without fear of reprimand) vis-à-vis a known and addressable entity? What constitutes having a real voice, and does it manifest in comments in forums, face to face, and in financial donations? Can participants leave without losing something, or protest and expect to be heard? What can you not exit from, such as formal or technical commitments to a website, a platform, an account, the Internet?
>
> **e. Availability:** The range of licenses and restrictions in these projects is

> considerable, from GNU to Creative Commons Licenses to moral economies, where no formal legal structure exists at all, to direct expropriation, where participants knowingly carry out free or underpaid labor and indirect expropriation, as when participants often unwittingly offer up their data for commercial expropriation. Which licenses are used? What is the rationale behind the decision of using a particular license? Is there collective control and/or individual access to the resource produced by participation? Can participants trust that what they give to a project will be returned to them in some form?

The methodological orientation we followed to gather data on these five variables is neither ethnographic, nor heavily driven by theory, but based on a middle range empirical approach. We set out as a primary goal to collect information on particular cases and compare between them. In the next four sections, we will present and analyze FOSS trajectories in order to provide a preliminary sketch of the range of participation.

5. FOSS Cases

5.1. Debian: from OP to FSE

One of the most important community-based Free Software projects, the Debian Project has more than a thousand volunteer developers working on almost twenty thousand software packages. The project was created in 1993 by Ian Murdock, during the early period of commercialization of Linux, in which several companies started Linux distribution projects. From 1994 to 1995, its development was sponsored by the Free Software Foundation, which guaranteed funding for Murdock to dedicate himself full time to the project. Bruce Perens, the successor of Murdock as the Debian project leader (DPL), founded the NGO Software in Public Interest in 1997 (with Tim Sailer and Ian Murdock), as a non-profit "umbrella organization for projects from the community" and the outfit that handles money donations to Debian.[1] Perens also wrote up Debian's “Social Contract” based on a month-long discussion on the Debian mailing lists, and included in it the “Debian Free Software Guidelines” to define the project's moral and technical commitments. Its core developers had ties with the IT industry in companies such as Sillicon Graphics, HP, and these ties helped the project obtain necessary hardware infrastructure, and Debian has received monetary support from Hewlett and Packard. Today Debian remains an independent decentralized organization, with a current stable release of over 29,000 software packages for 11 different computer architectures. (source: debian.org). Software in Public Interest remains the main foundation behind Debian as well as several other Free Software projects, providing money for conferences, accepting donations, and giving donations to Debian Developers for traveling to the annual Debian Conference.

Active members of Debian are generically identified with three basic titles that determine task (and skill) levels: Debian contributors, who are free software packagers; Debian Developers (DD), those who were socialized into the New Maintainer Process and are responsible for the quality of the packages that are included in the system (Coleman 2005); and Debian maintainer, those who dedicate less time to the project, but

[1] http://www.debian.org/donations

help debugging and packaging software. Package maintainers have control over their own packages, which are increasingly co-maintained. They manage the upstream source code version of FOSS projects and submit for packaging for quality assurance. Other tasks are usually handled by the domain of smaller, more collaborative groups of developers who perform, for instance, translation and internationalization work, upkeep of the IT infrastructure (software repositories, content delivery networks, IRC servers, mailing servers, physical CD distribution, merchandise, books, t-shirts), and documentation. Developers may make any technical or nontechnical decision with regard to their own work.

Debian's goal is to build a "universal" operating system, as articulated in its Constitution and Social Contract.[2] At the software development level, Debian's Developer OP is involved in goal setting through both representation by the Project Leader and majority vote (which is computed using the Condorcet Method): developers may 1) propose or sponsor draft General Resolutions; 2) propose themselves as a Project Leader candidate in elections; 3) vote on General Resolutions and in Leadership elections. General Resolutions address, generally, the appointment of the Project Leader; amendments to the constitution provided they agree with a 3:1 majority; decisions authorizing the powers of the Project Leader or Delegate as well as the Technical Committee; proposals and amendments for nontechnical policy documents and statements; and, in case of a conflict, the secretary appointment.

The Debian Project uses a formalized process of membership acquisition; it is one of the most structured community projects with a very clear process of collective governance. To be accepted as a DD is an index of prestige within the FOSS community. The observance of the ethical standards and the enskillment of individuals is mediated by three important community documents: the Debian Constitution, which describes the organizational structure for formal decision-making within the project and enumerates the powers and responsibilities of the Debian Project Leader (DPL), the Debian Project Secretary, and the Debian Developers generally; the Social Contract; and the Debian Free Software Guidelines (DFSG). These three documents codify the notion of freedom in the scope of the Debian project, as well as perform a public display of a commitment between technical advancement and software freedom.

Coleman (2005) and Auray (2003) provided an in-depth analysis of the internal regulation of Debian and the maintenance of its boundaries. As a form of control over the growth and the technical quality of the project, the Debian community created a formal process for the admission of new developers called "New Maintainer Process" (NMP), which stands as a solution for the problem of integration and trust among remote international collaborators (Coleman, 2005, p. 350). Coleman (2005) demonstrates – based on her ethnographic work – that by engaging in the process, newcomers incorporate the technical skills demanded from Debian Developers and ethically commit themselves to the project. According to Coleman (2005), Debian has a hybrid mode of governance, composed by democratic majoritarian rule, meritocracy and *ad-hoc* process of rough consensus. The author argues that most of the conflicts and crises within the project emerge from this hybridity. Based on ethnographic data and a corpus of qualitative data, O'Mahony and Ferraro (2007) identified four successive phases in the development of Debian which highlight the change in the orientation towards authority of the DPL: 1) authority exercised by the Debian founder (1993-97); 2) problem of succession and centralization of decisions about the future of Debian led

[2] http://www.debian.org/devel/constitution; http://www.debian.org/social_contract

the project to draft and ratify a constitution (1997-99); 3) with the approval of the constitution, the project implemented the new formal model of governance (1999-2003); 4) the stabilization of the formal process with the dispute for authority based on platforms for the future of the project (2003-2006).

As it is the case for most FOSS projects, Debian developers have the strong backing of a wide array of collaborators, those who provide bug reports, bug fixes and documentation (as well as translation and internationalization work). Those collaborators that have an active role in the community gain recognition, which is the first step to apply for membership in the Debian project. To become a developer it is necessary to ask for sponsoring from another Debian Developers and to participate in their web of trust (reinforced through GPG key signing parties – sharing of cryptographic fingerprints for future exchange of data and information in encrypted, secure form); it is even required that for a newcomer to be integrated in the project, a Debian Developer has to meet the person face to face and advocate for his or her membership acceptance. In the context of interpersonal relations, Coleman argues that "power is said to closely follow the heels of personal initiative and its close cousins, quality technical production and personal dedication to the project" (Coleman, 2005, p. 341). Sponsors are important agents in the NMP, given that they must ensure that the newcomers learn how to work the way Debian Developers work. Within a project, roles are self-assigned or submitted to the voting system (in the case of the election for DPL - Debian Project Leader). The roles are specified as such: maintainer, release manager, and technical committee member, who mediates debates on technical issues.

Debian achieves a strict adherence to the philosophies of UNIX and Free Software, and its Free Software and Open Source licenses enforce this compliance. The Debian Free Software Guidelines define what is permissible in the distribution: free redistribution that includes source code and allows modifications and derived works, to be distributed under the same license as the original. However, non-Free Software are distributed separately from the main distribution and can be included in the system**.**

Based on our schema for public participation, Debian has a normatively strong level of participation, particularly at the level of software development. Developers are allowed a say in the structure and goals of the project as a whole, not only in their individual and collective tasks. Like most FOSS projects, the resource's licenses put it under collective control and allow individual access to code. Its governance structure provides a mechanism for participants to move up within the hierarchy in order to exercise a voice in high-level decisions, and participants can exit without penalty. And there is an educative dividend: volunteer developers not only gain technical skill alongside the development Debian software products, but they are encouraged to develop technical skills and collaborative relationships the more tightly involved they become.

5.2. Mozilla: from an FSE to an OP

Another important case is the open source initiative of the company Netscape Communications. In 1998, Netscape released the source code of its Internet browser, the Netscape Communicator and released it under its own open source software licenses (Netscape Public License and Mozilla Public License). 15 July 2003 saw the launching of the foundation Mozilla.org, a non-profit organization that is responsible for the maintenance of the code base of the Mozilla browser suite. Two years later the

Foundation started the Mozilla Corporation, described as "a taxable subsidiary that serves the non-profit, public benefit goals of its parent, the Mozilla Foundation, and that will be responsible for product development, marketing and distribution of Mozilla products." (mozilla.org 2008) In 2004, the Mozilla Foundation discontinued the Mozilla suite to focus entirely on Firefox and Thunderbird, though the suite, now called Sea Monkey, continues to be developed within the Mozilla community.

At a basic level individual tasks by Mozilla developers include bug-fixes or improvements to the source code in the repositories. Higher level tasks are delegated along a tight hierarchy of responsibility. Module owners, for instance, lead the development of a module of code as well as community activities. Super-reviewers review code for its effects on the overall state of the tree and adherence to Mozilla coding guidelines. Release drivers determine which bug fixes are important for a given release. Component owners review bug reports, reassign bugs to correct owners,and track the resolution of important fixes. At the top of this schema are the 'ultimate decision-makers', Brendan Eich and Mitchell Baker who have the final say in disputes.[3]

Mozilla has not set up any procedure for allowing developers to petition or modify goals, or to vote for people in leadership positions such as the 'ultimate decision-makers' and the module owners, who are tasked with effectively developing modules for products such as Firefox, Thunderbird, SeaMonkey, among others. According to Holck and Jorgensen (2005), Mozilla top-level management is done by a staff composed by not more than a dozen paid employees. In contrast to Debian, the staff members are not democratically elected by the community: "you do not apply to join the staff, you are invited to join" (Holck and Jorgensen, p.12). Mozilla coordinates the development work, mostly done by former Netscape engineers who "own" modules of the browser (Hamerly et al., 1999), and those owners define the contributions from the public that are going to be included. Holck and Jorgensen speculate that the undemocratic structure may derive from Mozilla's ties to Netscape, where most developers were employed by the same company whose mandates and procedures were already clear - making election of top-level management less attractive.

Mozilla also formalized the process of becoming a "committer," which is the general figure of the person who has the power (and the responsibility) to apply modifications to the source code of the project. As a common procedure for FOSS projects, those who want to engage effectively with the project have to demonstrate interest publicly, by participating in online interactions via mailing list and IRC, studying documentation and the source code, and sending bug reports and patches with bug fixes to officially assigned committers. With this demonstration of interest, aspiring committers get respect and recognition of the module owners, who can then vouch for them to get access and become an official Mozilla commiter. According to Hamerly et. all (1999), there are two steps in the evaluation of contributions to Mozilla: 1) evaluation of quality and 2) compatibility of licenses - contributions must not conflict with Netscape Public License or the Mozilla Public License. After sustained involvement and formal approval, committers sign a "committers agreement," a formal document explaining license terms and providing a statement that a participant's code complies with Mozilla license policy.

Mozilla projects are licensed using primarily the Mozilla Public License (MPL), but also NPL and GPL. The MPL allows covered source code to be mixed with other

[3] http://www.mozilla.org/about/roles.html

files under a different, even proprietary license. However, code files licensed under the MPL must remain under the MPL and freely available in source form. MPL-licensed code can be freely used, altered, and redistributed. Mozilla engineering practices also instituted the following conditions for the open enterprise of development: "effective version control; a well-defined protocol for integrating source code changes; a high degree of accountability for people who integrate this code; high modularity; custom development tools; good communication channels" (Reis and Fortes 2002, p. 5). All these conditions express the order of a development methodology that is, according to the authors, "bug-driven".

Mozilla Foundation owns the trademarks of its products and claims the right to deny the use of the name and other trademarks to unofficial builds. This has caused some friction and at least one fork; for instance, in 2006, the Mozilla Corporation, a for-profit subsidiary of the Foundation, disapproved of modifications made to the Thunderbird software by a Debian Project and invoked their Trademark to have the Debian developers comply with their standards before redistribution. In response, Debian simply rebranded the project, allowing them to distribute the software with modifications permitted, but without being bound by Mozilla's trademark requirements. In this Debian project, Firefox became Iceweasel, SeaMonkey became IceApe, and Thunderbird became Icedove. (Debian Bug Report Logs, 2006)

Unlike Debian, Mozilla's participants have little say over goal-setting and have no democratic control over their leaders. Participants have voice at the level of task-setting, but not over matters of governance. Like Debian, the resource is available to users that comply with the Foundation organizational schema, and there is a clear educative dividend in working on the collaborative modules. However, its orientation towards Trademark has allowed it to restrict certain projects, in one case leading to a fork by Debian of its software development. It seems possible that, given Mozilla's history, its weaker forms of participation derive in part from Mozilla's original attachment to a for-profit business, which supplied much of the initial labor pool, rather than an originally unaffiliated Organized Public who built formalization from the bottom up, as Debian did.

5.3. Dyne.org: OP without FSE

Dyne is a radically decentralized collection of self-identified hackers and activists who engage in electronic disturbance, digital inclusion, and art projects. A self-described "nomadic network" (Jaromil 2009) they have refuse crystallization of the project into a Formal Social Enterprise (FSE). As a loose software forge, they have legal status in the form of a Dutch Stichting (non-profit), but they have no formal contractual structure of attributions such as board members, paid staff, and meetings. They have had institutional affiliation with the Nederlands Instituut voor MediaKunst, which has provided their server space.

Founded by Dennis Rojo Jaromil and Tatiana de la O, the project first centered around a Linux distribution dedicated for multimedia: Dynebolic. It then evolved into a set of applications for FOSS multimedia, including a highly-encrypted OS that could be used by political activists, particularly those working in repressive societies or contexts of potential surveillance, as well as for V-Jaying and projects devoted to issues of the “digital divide”. Dynebolic was designed to allow all these software projects to run on older computers, in tactical opposition to consumerist approaches; instead, their multipurpose Operating System could operate on platforms that might be older than the latest device. Consequently, Dyne organizes projects around changing the legal structures of copyright more broadly, such as allowing legal modification of the chips of formerly closed game consoles. (Jaromil, 2009)

Structurally Dyne.org operates as a "hub" connecting different groups that are geographically dispersed - from free radio to internet radio, software development groups, artists and independent journalists. Its software development has a clear political agenda and involves artists and active participants in the anti-corporate globalization movement. Most of the software contributors have experience with radical political collectives and the Dutch squatters movement.

Participants' tasks include both software development, documentation, and support. Their mission statement is pointedly political: to "promote the idea and practice of open source knowledge sharing within civil society; to open the participation to on-line and on-site communities, leveraging the democratic and horizontal access to technology, lowering the economical requisites to its accessibility; to foster employment of FOSS in artistic creation: exploring new forms of expression and interaction, disseminating new languages that can be freely adopted and re-elaborated by everyone, insuring the long term conservation of digital artworks; being software a socially relevant media it should not be invented and maintained only on the basis of its merchantability".[4]

The website contains no information specifying formalization of governance. Communication on the project is instead done through listserves, wikis, and forums, and all tasks and goal-setting are self-initiated, collaborative, and *ad hoc*. This lack of formal structure complies with the anarchist orientations of Dyne.org members, who see Free Software as inextricably linked to social justice activism, the gift economy, privacy, and free speech. According to one of Dyne.org's founder, Jaromil, this decentralized structure is a political mirroring of the potential network structure of software projects themselves: "We do not want to have an institutional role; we are a network." (Jaromil, 2009) In particular, Jaromil emphasizes that this is a hacker network of individuals who themselves may be part of multi-nationals, government owned telecoms, and other non-profits, but who collaborate together in a parallel community of sharing and openness that share similar political goals.

Dyne.org's embraces the general orientation of the GPL license, allowing modification, redistribution, and commodification. Articulating this on Dyne's website is the clause, "Verbatim copying and distribution is permitted in any medium, provided this notice is preserved."[5]

As an OP without an FSE, Dyne.org has created infrastructure for very strong participation. Developers engage in tasks as well as goal setting, and the resource is collectively available provided its distribution remains copyleft. Governance is formally decentralized and given no written specifications, and all involve gain an educative dividend both by developing the technology while finding solidarity in an activist community.

[4] http://www.dyne.org/mission/.

[5] http://www.dyne.org/

5.4. Xara Extreme on Linux: an FSE without an OP

Xara Extreme on Linux is a vector graphics and photo editing software, the open source version of Xara Xtreme, originally released in 1992 under the name "Artworks". The open source version was released for the Linux platform by its owner Xara Ltd in 2006 at the inaugural Libre Graphics Meeting in Lyon, France. However, the port from Windows to Linux immediately stalled when Xara refused to release a central piece of the code as Open Source, namely the the application's core rendering library CDraw - a situation that failed to attract volunteer developers. In 2007, Xara Ltd sold to new owners Magix, who similarly worried about developers compiling an open source Windows version. Later that year the company announced it was pulling its own developers off the open source version, to concentrate on the release of its next Windows product and major cash cow. (O'Neil n.d.) Today the project is called “Xara Xtreme for Linux” and, according to its website, which was last updated in 2008, the port still hasn't completed.[6]

The main tasks for developers of the project involve porting XaraXtreme for Linux. The goals, as stated on the website, are "to create a new cross-platform industry standard, To change the graphics landscape forever, To create the best drawing / vector graphics software that has ever existed, At the same time create a genuinely useful, general-purpose graphics tool for everyone."[7] Magix the company sets the developmental goals for the project; because it failed to develop a structure for collaboration or situation of trust, the developing community fell away. According to Linux.com journalist Nathan Willis (2009), Xara LX will "begin to suffer from bit-rot as core system libraries evolve. It will stop working at some point, and become just like the thousands of other abandoned applications still available through SourceForge.net and other project hosting services."

The company Magix's top-down manner gave developers no voice in the matter of its withheld code or in determining the quality and shape of the product. According to their website's FAQs, "we are going to manage the official version. Anything branded Xara will be the official version that has our direct backing, undergone our fanatical quality control to ensure not just they are as reliable and as fast, but that they continue to provide the slick ease of use that Xara are renowned for. Assuming we continue to manage the project, and develop the product as the user community want, to the high standard we have in the past, we would hope to have an active and critical role in the future direction of the product."[8] Like most FOSS projects Xara Xtreme was released under the GPL (GNU Public License) version 2.

This is an example of one out of thousands of failed open source projects, defined as such because it could not foster an OP necessary to take its development further. From the start, participation was thin: participants did not take part in goal setting and had no means to participate in certain parts of the project (task definition, goal settings). The resource itself was not collectively owned: ten percent of the project, a small but core part, was withheld from developers. In terms of governance, participants' voices were ignored. Finally, there was little creative dividend, given the lack of activity to begin with. We can see that a weak approach to participation had a detrimental effect on this project.

[6] http://www.xaraxtreme.org/

[7] http://www.xaraxtreme.org/faqs.html

[8] http://www.xaraxtreme.org/faqs.html

6. Discussion

This paper aimed to provide a contribution to the study of participation within a variety of FOSS projects. The cases introduce a variety of trajectories in which participation assumes different forms with some overlapping characteristics. In the table below, we offer a systematic evaluation of our cases, comparing them along the five variables we discussed in the previous sections:

Variable/Project	Dyne.org	Debian	Mozilla	Xaraextreme
Trajectory	OP without FSE	Strong OP with FSE	Strong FSE with OP	FSE without OP
Project Demographics	Free Software development for artistic and political purposes. Hacktivists, Tactical Media Artists, small number of active contributors, sporadic paticipation	Free Software development by professional and hobbyist programmers, large number of active members, entirely based on volunteer work	Open Source for open standards for the web, Professional Programmers, Designers, Community Managers; large user base, small number of FSE members / employees	Open source for profit and as a strategy to avoid bankrupcy. Professional Programmers, small number of contributors, sporadic participation, most active members are paid employees
Alliances	Self-funded and sponsored with donations; partners with Free Software projects and hacktivist groups from Europe	HP and donations from IT corporations and companies	Occasional partnership with IT corporations for "Internet Freedom" causes	None
Tasks and Goals	Tasks and goals are consensually defined and performed	Tasks are defined by Debian Developers and discussed in mailing lists, do-ocracy and meritocracy count in the definition of goals and tasks (after a period of active participation in the project) Tasks are picked by contributors, but defined collectively by the OP	Community organizing, software development, tasks are picked by volunteers, but defined by the Mozilla foundation	Software development for certain parts of the program (not allowed in others)

Governance	Consensus driven and meritocracy combined. No formal procedure, *ad-hoc* decision making, authority based on technical expertise and political trajectory	Meritocracy combined with representative democracy. The most active members of the OP can apply to become leaders of the project. Formal political structures are defined and coded in the legal documents of the community	FSE managing of OP by community managers; prestigious members of the OP get to participate in the board of the FSE	Corporate control over the Open Source project
Availability	Free Software Licenses (GPL version 2 and 3). Intellectual Property Rights are given to contributors	Free and Open Source Licenses. Contributions become property of the contributors	Mozilla Public License (Open Source License), NPL and GLP. Contributions to Mozilla become property of Mozilla foundation	Mixed licensing: proprietary and open source

By looking at different cases of participation, project formation and development, we have identified initially 2 key features: 1) *project geneology* (antecendents of the project, either structured as a company or loosely organized around political goals or particular technology of wider application beyond the immediate project,); 2) *public participation in tasks and goals* define the intensity of public engagement around Free Software (corporate controlled or heavy FSE coordination and mediation tends to fail to attract public participation). Availability is another important dimension. According to Santos Jr. et Al (2011) attraction to FOSS projects is closely linked with licensing decisions:

There are important aspects of FOSS projects that were left out of our exposition and should be incorporated in our study. The temporal dimension of our model (mostly OP to FSE, but also FSE to OP and, sometimes, only OP and FSE without further transformations) is more complex than the relation between these two types of organizational spheres, given their internal variability and alliance with other entities that are not FOSS-based. On a larger scale, the distinction OP and FSE captures well the formation and transformation of organizational spheres within FOSS projects. On a micro sociological scale, important questions of trust (Antikainen et al. 2007) and subjective conditions for engagement in FOSS projects should be taken into account. What is left to be done in future work is to further explore the interplay between the macroscale and the microscale of analysis in order to capture finer grained details of the experience of the participants: what can be said about the level of technical expertise, experience with other FOSS projects or other types of organization that are brought to a project? By pursuing this question, we might be able to further explore the relation between OPs and FSEs. To answer this question also means to describe the conditions of possibility for the four major project formations we identified: OP without FSE; OP constituting over time an FSE; FSE fostering an OP; and FSE without OP.

7. References

Antikainen, M., Aaltonen, T., Vaisanen, J., Feller, J., Fitzgerald, B., Schcchi, W. and Silitti, A. “The Role of Trust in OSS Communities—A Case Linux Kernel community.” In *Open Source Development, Adoption and Innovation*, 223–228. New York: Springer, 2007.

Antoniades, IP et al., “Dynamical simulation models of the Open Source Development process,” in *Free/Open Source Software Development*, ed. S Koch (Hershey, PA: Idea Group Publishing, 2004), 174-202.

Auray, N. (2003) “Communautes epistemiques d'innovation - La regulation de la connaissance : arbitrage sur la taille et gestion aux frontieres dans la communaute Debian,” *Revue d'économie politique.* 11: 161.

Barcellini, F. et al. (2005) “A Study of Online Discussions in an Open-Source Software Community,” in *Communities and Technologies,* 320, 301.

Browne, C B. (1998) “Linux and Decentralized Development,” *First Monday* 3, no. 3, available at: http://www.firstmonday.org/issues/issue3_3/browne/index.html.

Coleman, G. (2005) The Social Construction of Freedom in the Free and Open Source Software: Hackers, Ethics, and the Liberal Tradition. Dissertation, University of Chicago, 2005.

Coleman, G. and B. Hill, (2004)“The Social Production of Ethics in Debian and Free Software Communities: Anthropological Lessons for Vocational Ethic,” in *Free/Open Source Software Development*, ed. S Koch (Hershey, PA: Idea Group Publishing), 273-295.

Crowston, K. and Howison, J.. (2006) “Hierarchy and centralization in free and open source software team communications,” *Knowledge, Technology & Policy* 18, no. 4: 65-85.

Debian.org. "Debian 6.0 'Squeeze' released". February 6, 2011. http://www.debian.org/News/2011/20110205a.

Debian Bug report logs. "Debian Bug report logs - #354622. Uses Mozilla Firefox trademark without permission" 27 February 2006. http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=354622.

Dinh-Trong, T T and J M Bieman (2005) “The FreeBSD Project: A Replication Case Study of Open Source Development,” *IEEE Trans. Softw. Eng.* 31, no. 6: 481-494.

Fielding, R T. “Shared Leadership in the Apache Project,” *Communications of the ACM* 42, no. 4 (April 1999): 42-43.

Franke, N. and E V Hippel, “Satisfying Heterogenous User Needs via Innovation Toolkits: The Case of Apache Security Software,” *Research Policy* 32, no. 7 (2003): 1199-1215.

Gonzalez-Barahona, J. et al. (2004) “Analyzing the anatomy of GNU/Linux distributions: methodology and case studies (Red Hat and Debian),” in *Free/Open Source Software Development*, ed. S Koch (Hershey, PA: Idea Group Publishing, 27-58.

Gonzalez-Barahona, J. et al.. (2009) “Macro-level software evolution: a case study of a large software compilation," *Empirical Software Engineering* 14, no. 3: 262-285.

Hamerly, J, T Paquin, and S Walton (1999) "Freeing the Source: The Story of Mozilla," in *Open Sources: Voices from the Open Source Revolution*, ed. C DiBona, S Ockman, and M Stone. Cambridge, Massachusetts: O'Reilly and Associates.

Holck, J. and N Jorgensen (2004) "Do not Check in on Red: Control Meets Anarchy in Two Open Source Projects," in *Free/Open Source Software Development*, ed. S Koch (Hershey, PA: Idea Group Publishing), 1-26.

Jaromil. (2009) "Interview with Jaromil (dyne.org) by Soenke Zehle." Posted by the Institute of Network Cultures. 3 March 2009. http://vimeo.com/4165209.

Jorgensen, N. (2001) "Putting it All in the Trunk: Incremental Software Engineering in the FreeBSD Open Source Project," *Information Systems Journal* 11, no. 4 (2001): 322.

Kaufman, A. S. (1969) "Human Nature and Participatory Democracy: Ten Years Later." In *The Bias of Pluralism* ed. William E. Connolly. New York: Atherton Press, pp. 178-200.

Leach, J. 2009. "Freedom Imagined: Morality and Aesthetics in Open Source Software Design." *Ethnos* 74 (1) (March 1): 51–71.

Lakhani, K., E V Hippel, and K Lakhani. (2003) "How Open Source Software Works: Free User-to-User Assistance," *RESEARCH POLICY* 32: 923—943.

Lynd, S. (1965) The New Radicals and "Participatory Democracy." Printed for Students for a Democratic Society. Reprinted from DISSENT, Summer.

Mateos-Garcia, J. and W E Steinmueller (2008) "The institutions of open source software: Examining the Debian community," *Information Economics and Policy* 20, no. 4 (December): 333-344.

Matthew, R. (2003). The Pressure of Openness: the hybrid work of the Linux Free/Open Source Kernel Developers. Dissertation, University of California, San Diego.

McKusick, M K. (1999) "Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable," in *Open Sources: Voices from the Open Source Revolution*, ed. C DiBona, S Ockman, and M Stone. Cambridge, Massachusetts: O'Reilly and Associates.

Mockus, A, R Fielding, and J Herbsleb. (2002) "A Case Study of Open Source Software Development: The Apache Server," in *Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000)*, 263-272.

Mozilla.org (2008) "Mozilla Foundation Reorganization." 21 April. http://www-archive.mozilla.org/reorganization.

O'Mahoney, S. (2005). Nonprofit foundations and their role in community-firm software collaboration. In Perspectives on free and open source software, edited by J. Feller, B. Fitzgerald, S. A. Hissam, and K. R. Lakhani, 393-413. Cambridge, MA: MIT Press.

O'Mahony and Ferraro, 2007. The emergence of governance in an open source community. Academy of Management Journal, Vol. 50, No. 5, pp. 1079-1106.

O'Neil, D. (n.d.) "Xara Xtreme History." Digitalred.com, http://www.digitalred.com/oldsite/xara-xtreme-history.php.

Pateman, C. (1970) *Participation and Democratic Theory*. Cambridge: Cambridge University Press.

Reis, C.R. and de Mattos Forte, R.P. (2002) An overview of the software engineering process and tools in the Mozilla project. Proc. The Open Source Software Development Workshop, 155--175.

Robles, G., J Gonzalez-Barahona, and J Merelo (2006) “Beyond source code: The importance of other artifacts in software development (a case study),” *Journal of Systems and Software* 79, no. 9: 1233-1248.

Santos,Jr., Carlos Denner, Marcos Bonci Cavalca, Fabio Kon, Julio Singer, Victor Ritter, Damaris Regina, and Tamy Tsujimoto. 2011. “Intellectual Property Policy and Attractiveness: a Longitudinal Study of Free and Open Source Software Projects.” In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, 705–708. CSCW’11. New York, NY, USA: ACM.

Smith, A. and Jones, B. (1999). On the complexity of computing. In *Advances in Computer Science*, pages 555–566. Publishing Press.

Toral, S. (2009) “Virtual communities as a resource for the development of OSS projects: the case of Linux ports to embedded processors,” *Behaviour & Information Technology* 28, no. 5: 405-419.

Willis, N. (2007) "Lessons learned from open source Xara's failure." Linux.com, 13 October. http://archive09.linux.com/feature/119790.