V. 1.0

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. *One can observe a common trajectory in Free and Open Source Software (FOSS) projects with respect to public participation: they tend to start with a few core developers, then increase in code base size, complexity, and number of contributors and users, and in most cases form a formal organization to help coordinate the logistical aspects of the project. The question we discuss in this paper is "what are the characteristics of public participation in those projects that* do not *describe the common trajectory? 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 refused to constitute) a formal social enterprise. We highlight fundamental differences and similarities between projects: what makes them grow stronger or fail to attract, foster collaboration, and further forms of public participation. Four cases are explored (Dyne.org, Debian, Android, and Xara Extreme Linux) across five dimensions: 1) project genealogy; 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?). Our goal is to contribute to the studies of public participation, power, coordination, and collaboration by exploring the interplay between organizational forms, Intellectual Property* *licensing, and genealogy of FOSS projects.*[1]

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, then finally find it necessary to create a formal organization to help coordinate the development efforts, maintain hosting infrastructure, secure funding, manage donations, seek partnerships, and 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 this common trajectory—and how do they compare with the normal case?"

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 organization. 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 genealogy*; 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, Android, and Xara Extreme Linux.

This article 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 uses interpretive social science methods to explore and compare features of participation across a wide range of projects (not confined to Free and Open Source software). By using comparative case analysis, the project seeks to refine theoretical understandings and contribute to concept development in the general domain of participation after the widespread use of the Internet.

2. Free and Open Source Software Trajectories

For more than three decades Free and Open Source Software (FOSS) has generated an intense and intricate dispersion of technical objects and practices based on global collective efforts. Recent anthropological and sociological accounts of Free Software as a political, technical, and cultural practice further investigate the ongoing dispute

[1] Research for this paper was carried out under National Science Foundation (NSF) grant #1025569. Support was also provided by the Center for Society and Genetics (UCLA). Authors are listed in alphabetical order.

regarding the modernist notion of individual property over intangible goods and the opposition created by FOSS to the advancement of the transnational intellectual property regime (Coleman 2005; Kelty 2008; Leach 2009; Weber 2005). FOSS offered viable alternatives for remote coordination, distribution, and innovation in software development, made possible by the virtue of its licensing schemes: the constant rebuilding effort over a set of public software licenses that allow (re)distribution, free use and adaptation of software code. The resulting phenomena are situated in between, at least, two major registers: the general reciprocity oriented towards the free circulation of software as public good, and the market economy in which computer technicians offer their FOSS acquired expertise for remuneration.

From an anthropological standpoint, FOSS is curiously made up by boundary practices in a multitude of social ties and sociotechnical arrangements, bringing together persons, associations, and technical objects: it is a form of craft that is hard to analyze without problematizing the boundaries of established categories and their taken-for-granted oppositions, such as 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* (Serres 2007) assuming different forms but mainly organized around intersecting recursive publics (Kelty 2008). Public administrators, for instance, may advocate for FOSS as tool for social change, given its potential to foster 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. In the past five years, we experienced an implosion of FOSS, circulating under the new rubric of “Open Hardware”, “Open Access”, and “Open Data”.

This article 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 FOSS and proprietary development models (Lakhani 2007; West 2009). 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. 2011) 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, general Intellectual Property disputes, conferences, and training sessions across key global cities (such as San Francisco, Tokyo, and Sao Paulo) and meetings with sponsors and partners. On the other hand, the OP is technically and structurally connected to the FSE in various 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 FOSS 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 gathers a thriving OP over time. Also, the composition of power within and between these two formations 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 and corporate partnerships, while alternatively a not-for-profit FSE could be composed of a handful of people who make little distinction between their efforts and the project's OP.

Table 1. Non-exhaustive list of FOSS projects' trajectories:

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

Why do initially informal FOSS Organized Publics (OP) tend towards formalization into Formal Social Enterprises (FSEs)? In some cases foundations formalize 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 often emerges when projects become commercially relevant and profitable from the perspective of the industry, delivering software as service; foundations facilitate the interaction between community projects and corporations. O'Mahoney cites a Fortune-100 executive who once questioned: "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. Formalization as a for-profit company or 501(c)3 also allows projects to accept donations and sponsorships, apply for grants, and 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 propose five dimensions of comparison between these trajectories that place them normatively along a continuum of strong vs. weak participation, to better understand how their formation over time might affect the levels of participation that can take place. Before we move on to outline these methods, we offer a theoretical approach for analyzing participants' social, legal, and technical capacities to engage with these projects by turning to historic definitions that can inform an analysis of participation as it occurs today.

3. The Problem of Participation

'Online participation' has been scrutinized in a rash of scholarly and pop culture accounts seeking to understand why communities form around blogs, file sharing, software development, Internet forums, independent news portals, etc. The popular conception of participation involves a shared resource used by a highly cooperative community that puts in significant time with no monetary gain. Yet closer inspection yields a very different picture: the limits and possibilities of participation are structured legally and sociotechnically and vary greatly across projects. If we want to obtain a finer picture of how participation occurs over time in FOSS specifically, we must advance a theoretical foundation for empirical research of participation as a core problem of social ties binding projects, technical systems, and people.

Participation and its theorists come in many forms (see eg. Carpentier 2011), but we review here a couple of key moments: democratic political theory, the practice of participatory democracy in the 1960s, and Industrial democracy—or worker participation—from the 19th century to the present.

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 make these choices about their society by participating directly in governance? The 20th century has seen the pendulum swing between strong “elite” understandings of democracy (such as those of Robert Michels or Joseph Schumpeter) and strong “participatory” theories of democracy (such as Arnold Kaufman, C.B. MacPherson, or Carole Pateman). Some of these debates hinge on definitions of “human nature” as when Kaufman, writing in 1960, critiqued Joseph Schumpeter's view that people can't sift intelligently through information or know what is in their best interest. Schumpeter was suspicious of crowd behavior and chaotic voices in aggregate, and instead favored representation by elected, learned leaders. Schumpeter however does not provide a framework for *how* these leaders should be formed, and Kaufman points out that direct democracy is precisely the mechanism needed to produce these rational leaders. Kaufman's take is Rousseauian: crucial to his strong interpretation of participatory democracy is the claim that deliberation must occur, allowing each participant an equal say in the matter and so allowing them to accrue critical skills through meaningful debate. For "it is only when [men/women] acquire direct responsibility for a certain range of decisions that social imagination breaks through its parochial barriers and envisages larger possibilities." (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" can have larger sway. (192)

The 1960s witnessed a flowering of attempts to strengthen or instantiate participatory democracy (PD). The term itself first appears with the Students for Democratic Society (SDS) in their 1962 Port Huron statement, where they proposed two critical ideas to be tested on the ground during the Civil Rights struggles in the US. Like Kaufman the SDS believe individuals must have a say in the decisions that determine their choices and quality of life. Also crucial is that societal structures are set up, from the beginning, to encourage critical thinking and help individuals comfortably express themselves when they enter into political discussion. Just as important, society must provide the media platforms citizens need for self-expression. In keeping with these ideas, SDS 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 that as these new institutions remained separate from the mainstream political process, their lack of formalization prevented them from addressing such basic needs as feeding people and raising them from poverty. But he takes comfort in the ideal of parallelism implicit in participatory democracy: as it operated in the 1960s, its radical structures operated 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) As such, PD can at least 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 (as many FOSS projects claim to do today).

A loosely related area of the theory and practice of participation is the industrial democracy movement 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. While it was influential and persistent in various places (most notably in Britain and the Scandinavian countries), it largely succumbed to the more class-antagonistic approaches of union labor movements that sought to keep the lines between workers and management as distinct and adversarial as possible.

Industrial Democracy's proponents expressed ideas that fall on a strong-to-weak continuum of participation. Carol Pateman's (1970) analysis 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 only 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 "implies the opportunity for full higher level participation by employees in the formal organization," though this ideal is never realized. (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 distinctions: the degrees between partial and full participation present a spectrum of possibilities allowing participants autonomy, share in resources, 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.

Table 2. Weak *versus s*trong forms of participation:

Variable	Strong	Pseudo or Weak
Tasks vs. Goals	Decision making in goals, not only tasks	Solely in tasks designed or framed elsewhere

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
Availability	Collective control and/or individual access to the resource produced by participation	Expropriation and private ownership of resource produced

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 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 following case studies.

4. Methods

For the analysis of participation in FOSS projects, we created a *corpus* of qualitative data composed of hypertext documents, mailing-list archives, video and audio interviews, presentations, and scholarly publications for four projects: Dyne.org, Debian, Android, and Xaraextreme.

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

a. Project Formation: a genealogical description of the project's origins and major shifts in its composition up to the present day.

b. Alliances: who are the partners and/or sponsors? Are they from the public sector, private sector, or both? Do these alliances have any formal or informal role in the FSE?

The following variables bear directly on the political dynamics of participation in FOSS projects and aim to distinguish the levels of privileges and types of roles granted to OPs by the FSE. These dimensions can be placed on a normative continuum of 'weak to strong' structures 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, or 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 veto power, 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 how 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, etc.?

e. Availability: The range of licenses and restrictions in these projects is considerable, from GNU General Public License 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 (credit, authorship, legal rights, access to resources)?

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 a continuum of public 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, a computer scientist during the early period of commercialization of GNU/Linux in which several companies started Free Software distribution projects. From 1994 to 1995, Debian 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" handling donations to Debian.[2] Perens also wrote up Debian's “Social Contract” based on a months-long discussion on Debian mailing lists, and included in it the “Debian Free Software Guidelines” to establish the project's moral and technical commitments. Debian core developers had ties with the IT industry in companies such as Silicon Graphics and HP, and these connections helped the project obtain necessary hardware infrastructure as well as monetary support. Today Debian remains an independent, decentralized organization with a formal process of decision making and appointed project leader, managing a complete operating system release of over 29,000 software packages for a handful of distinct computer architectures. Software in Public Interest is still 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 travel to the annual Debian Conference, which takes place in different parts of the world where Debian has local volunteer developers.

Based on ethnographic data and a corpus of qualitative data, O'Mahony and Ferraro (2007) identified four successive phases in the development of Debian that 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 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). Auray (2003), Coleman (2005), and O'Neil (2009) provide in-depth analyses 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 admitting new developers called "New Member 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 that, by engaging in the process, newcomers incorporate the technical skills and cultivate the ethical values demanded from Debian Developers.

Currently this formalized process of membership acquisition is one of the most structured FOSS community projects, with a very clear process of collective governance and training of its novice participants. 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). Collaborators who 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

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

is necessary to ask for sponsoring from another Debian Developer and to participate in his/her web of trust (reinforced through GPG key signing parties – sharing of cryptographic fingerprints for future exchange of data and information in encrypted and secure form); it is even required 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 are supposed to work.

Other active members of Debian are generically identified with three basic titles that determine task, skill, and responsibility levels: Debian contributors, who are Free Software packagers; Debian Developers (DD), who are socialized through the New Member Process and responsible for the quality of the packages that are included in the system (Coleman 2005); and Debian Maintainers, who dedicate less time to the project but help with debugging and packaging software. Maintainers have control over their own packages, which are increasingly co-maintained; they help to manage the upstream source code version of FOSS projects and submit for packaging and quality assurance. Other tasks are usually handled by the domain of smaller, more collaborative groups of developers who perform 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, considered as have a prestigious position within the FOSS community, make independent decisions regarding their own work, but generally request comments, suggestions, and help from other members of the project. Within a project, roles are self-assigned or submitted to the voting system (in the case of the election for DPL - Debian Project Leader). Maintenance of packages is a volunteer task self-attribution, and can also occur through the list of orphan packages (packages that are abandoned by their former Debian maintainers and are up for grabs by volunteers).

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 take part in governance by 1) proposing or sponsoring draft General Resolutions; 2) proposing themselves as a Project Leader candidate in elections; 3) voting on General Resolutions and in Leadership elections. By crafting the General Resolution, developers also address 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. With a hybrid mode of governance, composed by democratic majoritarian rule, meritocracy, and *ad-hoc* process of rough consensus, Coleman (2005) argues that most of the conflicts and crises within the project emerge from this hybridity.

The observance of ethical standards, and the technical skills of individuals are also 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 and perform a public display of commitment between technical advancement and software freedom; they also spell out Debian's goal to build a solid, community-developed operating system.[3]

Based on our schema for public participation, Debian has a normatively strong level of participation, particularly at the level of software development and project management. 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. Volunteer developers not only gain technical skill and cultivate moral values, but they are encouraged to further develop technical capacities and foster collaborative relationships the more tightly involved they become.

5.2. Dyne.org: OP *against* FSE

Dyne is a radically decentralized group of self-identified hackers and activists who engage in Free Software development, Internet activism, digital inclusion, and electronic art projects. A self-described "nomadic network" born as a protest against hierarchical institutions (Jaromil 2009), they have refuse to crystallize into a Formal Social Enterprise (FSE). As a loose software forge, they have legal status in the form of a Dutch Stichting (non-profit), they have institutional affiliation with the Nederlands Instituut voor MediaKunst, which has provided their server space, but they have no formal contractual structure of attributions such as board members, paid staff, and meetings. Over the years the project has also received support from GNU and the Free Software Foundation.

Founded in 2000 by Dennis Rojo Jaromil and Tatiana de la O, the project evolved out of the online Spanish language-based network Hackmeeting.org, a forum composed of political and human rights activists and hackers. The project first centered around a Linux distribution dedicated for multimedia, Dynebolic. Dyne then evolved into a set of applications for FOSS multimedia, including a highly-encrypted OS that could be used by political activists working in repressive societies or contexts of potential surveillance, as well as for VJing and projects devoted to issues of the “digital divide”. Dynebolic was designed to allow these software projects to run on older computers, in tactical opposition to consumerist approaches that require software and hardware upgrades. Dyne also organizes projects around changing the legal structures of copyright more broadly, such as allowing legal modification of the chips of formerly

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

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 of distributed labor among a community of artists and activists in the anti-corporate globalization movement. Most of the software contributors also have experience with radical political collectives and the Dutch squatter movement; the projects' internal procedures reflect these beliefs in cooperation rather than competition as a philosophical critique of capitalist economy. The word “dyne” itself comes from the Greek word *dynamis*, meaning power or self-generated motion, and connotes a leaderless effort whose direction will be determined by the sum of contributions of everyone involved. The mission statement is also 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] A 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 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) The website accordingly contains no information specifying formalization of governance. Communication on the project is instead done through listserves, wikis, and forums, which currently have 1000 subscribers and about 400 active users. All tasks and goal-setting are self-initiated, collaborative, and *ad hoc:* whoever takes an initiative can also set goals, though often community consensus inspires goal setting. In this setting, tasks and goals are intertwined, as administrators share goal setting with others using the infrastructure, depending on whoever is involved at any particular time.

At a basic level, tasks concern system administration, such as making sure the infrastructure runs, software development, bug fixing, documentation and support, participating in existing projects or forking, adapting and translating them. Dyne has no central 'help desk', but there does exist a core team of around 20 dedicated people, composed mostly of Italian developers (corresponding with its origins in Southern Italy), who respond to emergencies. According to Jaromil, this decentralization might mean that Dyne is less efficient. Efficiency, however, is less valued in this atmosphere than maintaining transparency, cooperation, and a personable spirit of exchange. Regarding licenses Dyne embraces the general orientation of the GPL license, allowing modification, redistribution, and commodification. Articulating this on Dyne's website

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

is the clause, "Verbatim copying and distribution is permitted in any medium, provided this notice is preserved."[5]

Currently, the Dyne website receives around 500 unique hits a day and has grown particularly in India and Brazil. However, according to Jaromil[6] the initiative is not self-sustaining, and its homepage now requires ads to pay the basic costs of running its infrastructure. Though the project has grown over the years, growth is now steady. Regarding its success as a FOSS project, Jaromil claims he does not know if dyne is a success, "or if a network should be a success. It's beyond this rhetoric of success and failure...also, exponential growth can be destructive." More important than growth, according to Jaromil, is that a community still orbits around the project as "an effort to write a collective history." The emphasis is instead on transparency of process, and on seeing participation in the project as a political and ideological stance in itself. If someone no longer wants to be connected to the project, he or she can still keep their work, and developers can leave anytime and rejoin if they want. Many in the current network are individuals who have applied the skills they learned in the early hacker movement to find jobs at other FOSS projects that can provide them salaries, including Android and Mozilla, but who collaborate together in Dyne's parallel community of sharing and openness. The motivation to stay a part of Dyne is clearly not commercial, but political.

As an OP *against* an FSE, Dyne.org has created infrastructure for reinforced horizontal 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. For Dyne, growth is not a metric for its success; rather, it values a self-reflexive exploration of the affordances of free software, and to alternative spaces that suggest radical political economies as an alternative to mainstream society.

5.3. Android: mediating between FSEs and an OP

Android Inc. was created in late 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White to develop a location-aware mobile device, later acquired by Google Inc. in 2005. The Android project was launched via the Open Handset Alliance (OHA) in 2007, a coalition composed of multi-national companies in the telephony, semiconductor, and software industries. As one of the biggest smartphone platforms worldwide, Android is also considered by the FOSS community as one of the most “closed” Open Source projects (see, for instance, “Open Governance Index” 2011). This observation is not only puzzling, but it captures adequately what is at stake in respect to public participation in Android: the fact that it integrates various Free and Open Source software projects while having a strong hold on high-level decisions, and, in doing so,

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

[6] Based on an interview dated November 4, 2012.

lagging behind in FOSS practices of transparency and openness. The controversy is related to process, more than product. The Android Open Source Project (AOSP) is composed of a software stack, combining different technologies which are released under different licenses, corresponding to different FOSS and non-FOSS development communities with and without FSE management.

For instance, Google only releases source code of completed versions, maintaining the company's complete control over the technological specifications that Android partners must then follow (Spreeuwenberg and Poell 2010). Google also provides ‘early access’ to the source code of new versions of the OS to specific third parties, allowing it to govern its relationship with manufactures, members of the OHA. As Spreeuwenberg and Poell (2012) have pointed out, Google additionally deploys its Android Compatibility Program, defined by the Compatibility Definition Document (CDD), to determine which hardware features can be allowed, so directing industry partners' production process in hardware development. Android's position towards public participation also affects more than the development side: users are asking for the right to own administrator rights over their Android devices (which to this date need to be modified to allow full access to the system and its functions, known in the community as “rooting”). One of the major consequences of this lack of control is that users do not share ownership of the metadata collected by Google nor have privileged access to implement enhanced security measurements.

Concerning Android project goals, influence by contributors is restricted to specific parts of the Android platform, and the extent of participation in software development is primarily determined by a contributor's place within a hierarchy managed by Google. AOSP separates tasks between “contributors”, who work on the Android source code, and “developers”, engineers who write applications that run on Android devices. Higher up are “verifiers” who test change requests. They are invited to this position and must have submitted a significant amount of high-quality code to the project. Still higher are “approvers”, who decide whether to include or exclude a change. At the top of this pyramid are “project leads”: senior contributors who oversee the engineering for individual Android projects and often work for Google. Once submitted, changes must be accepted by a designated approver, typically a Google employee. If a change is not approved, it is considered to be technically incompatible to the Android platform.[7]

The project is available under a myriad of licenses, which attest to its complexity and to its effort of integration and mediation not only between FSEs from the information and communication technologies industry, but also from community-based FOSS projects. Android uses Apache Software license 2.0, allowing corporate partners from the telephony and embedded systems industry to close-source their add-ons to the platform. To exemplify this feature to render Android close-source, Aaron Williamson – from Software Freedom Law Center – in his OSCON conference talk in 2010 demonstrated that, if all the proprietary software were to be removed from his HTC phone, he would lose the ability to use the GPS, camera, wifi, sensors, 3D, and

[7] http://source.android.com/source/roles.html

bluetooth. On the Free Software side, the Linux kernel and the library layer of the Android stack are licensed as GNU Public License and Lesser GNU Public License version 2 respectively. Its other key technologies are protected by proprietary licenses and Free Software licenses, such as variants of BSD and MIT. Most of the public participation is concentrated in the top applications layer of the platform, where independent developers and companies can use the developers kit to build applications for Android. Interest and attention has been drawn to Android via developer challenge programs and "hackathons" in which developers are awarded with large sums of cash for creating applications for the platform.

Weaker forms of public participation in Android therefor derive in part from Android's derivation from a for-profit business and project management model, which supplies much of the labor pool, rather than an originally unaffiliated Organized Public members, which was the case for Debian and Dyne.

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 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. 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.[8]

The main tasks for developers of the project involved porting XaraXtreme for GNU/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."[9] 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

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

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

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."[10] Like most FOSS projects, Xara Xtreme was released under the GPL (GNU Public License) version 2.

This is yet another example discussed in the FOSS literature (Schach 2009; Koch 2009) of the large proportion of FOSS projects that failed to attract public participation - “failures” because they could not foster an organized public necessary to take its development further. What is clear from the start, participation in Xara was weak: participants did not take part in goal setting and had no means to participate in certain parts of the project (task definition and goal setting). By reading the mailing list archives, one is able to see how the interactions between Xara employees and members of FOSS projects: the latter, by demanding access to code and suggesting widely shared FOSS tools (source code management tools, bug tracker systems) to conduct a more transparent and open development process. The resource at hand was not opened to the extend of becoming a public good: ten percent of the project, a small but central part, was withheld from developers of the public. In terms of governance, participants' voices were ignored. Finally, there was little to gain for the OP, given the lack of activity around the codebase and in the public mailing lists. We can see that a weak approach to participation had a detrimental effect on the attempt of Xara to foster an OP and develop a FOSS project.

6. Discussion

This paper aimed to provide a contribution to the study of public participation within a variety of FOSS projects. The cases we introduced describe 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	Android	Xaraextreme
Trajectory	OP *against* FSE	*Strong* OP *which formed one* FSE	*Mediation between* FSEs *to constitute one* OP	FSE *without* OP

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

Project Demographics	Free Software development for artistic and political purposes. Hacker activists, Tactical Media Artists, small number of active contributors	Free Software development by professional and hobbyist programmers, large number of active members, entirely based on volunteer work	ICT companies' software developers and engineers, freelance programmers, small mobile phone application developers	Professional Programmers employed by the FSE; small number of contributors, sporadic participation, all active members are paid employees
Alliances	Self-funded and sponsored with donations from the Dutch government; partners with Free Software projects and hacktivist groups	Donations from IT corporations and companies	Close to one hundred big ICT industry partners	None
Tasks and Goals	Tasks and goals are consensually defined and performed	Tasks are defined by developers, maintainers, and collaborators, and discussed actively in mailing lists; do-ocracy and meritocracy count in the definition of goals and tasks	Tasks are defined by the top engineers of FSEs (which are part of the Open Handheld Consortium) and distributed down to contributors (OP), who have the ability to submit patches, but not have *commit* rights.	Contributions in software development for certain parts of the program (not allowed in others). No commit rights and privileges to members of the OP, only to the members of the FSE
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 and procedures are defined and coded in the legal documents of the community	Control and coordination between one central FSE and several other FSEs to foster and advance the project. Members of the public have the ability to participate by developing applications to the platform. Few contributions of the public make to the official codebase of the project	Corporate control over the Open Source project, no accountability
Availability	Free Software Licenses (GPL version 2 and 3). Intellectual Property Rights are kept by the contributors	Free and Open Source Licenses which comply to the DFSG (Debian Free Software Guidelines). Contributions are kept as intellectual property of the contributors	Mixed licensing: Apache 2.0 (preferred), variants of BSD, GPL v2 (Linux Kernel), plus proprietary licenses. Contributions become part of Android, owned by the principal FSE and the partner FSEs	Mixed licensing: proprietary and GPL version 2

The comparison among these four representative cases reveals important differences in how participation is structured. To be clear, this approach does not reveal much about individual motivations, frustrations or desires, but it does create the outlines of a normative theory of participation that might help us understand either the success of a project or the reasons for differing motivations or experiences in different projects, based on their relationship (or not) to a formalized entity.

Participation is not a neutral experience for most people—it comes with values and implies a sense of worth and an understanding of context that are not always present in an organization. Generations of scholars have explored the role of worker participation in corporations and factories in order to try to understand how participation affects everything from worker happiness to raw productivity to shareholder returns. FOSS projects, however, provide a unique "natural experiment" in the reconstruction of organizational forms. The specific details of the genealogy and trajectory of different projects can be analyzed in order to understand the key features in the table above.

The two features that are most illuminating in this study are the tasks and goals of these projects and the availability of resources. Participation in goals as well as tasks creates a stronger sense of belonging, a more robust sense of deliberative or participatory democracy and a sense of obligation that is otherwise absent when participation in tasks is the only option. This can be seen both in the explicit politicization of this aspect by Dyne, but also in the different attitudes and forms of engagement present in Debian *versus* those in Android.

Availability is another important dimension, corroborated by the analysis of Santos Jr. et Al (2011) of attraction to FOSS projects, which demonstrates how closely linked it is with licensing decisions. Participation decisions are conditioned upon how a resource is managed and maintained, as well as on its legal availability. This difference is most stark in the case of Android, which must walk a fine and complex line between legal availability and managerial and strategic control—with obvious implications for the level and kind of participation they can expect (and manage).

There are important aspects of FOSS projects that were left out of our exposition and should be incorporated in a future version of 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 transformation) is more complex than the relation between these two types of organizational spheres (FSE and OP), given their internal variability and patterns of 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), moral, and political conditions for engagement in FOSS projects should also be taken into account.

Our conclusions here do not settle the question of how to increase participation, or improve the outcomes of FOSS projects—rather they decompose the notion of participation into its component parts in order to ask how one might distinguish between strong and weak participation and for whom. Too often the label's Free Software or Open Source are taken to imply a valued form of participation—but this is an unwarranted implication. Rather, it is necessary to explore why participation is valuable, how it has succeeded or failed in the past, and what new forms of organization, technology or politics are giving it a new lease on life.

7. References

Android Open Source Project. (2011) "Android Compatibility.", accessed on 03/08/2012, available at: http://source.android.com/compatibility/index.html

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 6th, 2011. accessed on 01/20/2013, available at: 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.

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.

Serres, M. (2007) *The Parasite (Posthumanities)*. Minneapolis: University of Minnesota Press.

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.

VisionMobile (2011) “Open Governance Index”, accessed on 09/11/2012, available at: http://www.visionmobile.com/product/open-governance-index/

Willis, N. (2007) "Lessons learned from open source Xara's failure." Linux.com, 13

October. http://archive09.linux.com/feature/119790.