The Functioning of a Free Software Community:

Entanglement of Three Regulation Modes – Control, Autonomous and Distributed

Didier Demazière, François Horn and Marc Zune

The ability to build solid and coherent software from spontaneous, sudden and evanescent involvement is viewed as an enigma by sociologists and economists. The internal heterogeneity of project contributors questions the functioning of collective action: how can commitments that are so dissimilar be put together? Our objective is to consider FLOSS communities as *going concerns* which necessitate a minimum of order and common, shared, social rules to function. Through an in-depth and diachronic analysis of the Spip project, we present two classical modes of social regulation: a control regulation centred on the product and an autonomous regulation reflecting the differentiated commitments. Our data shows that the meaning, value and legitimacy of contributors' involvements are defined and rated more collectively, through exchanges, judgments, and evaluations. A third regulation mode, called *distributed community regulation* and aimed at creating and transforming shared rules that produces recognition and stratification, is then presented.

Keywords: Free software, social regulation, Spip project, ethnographic approach

Free software can be considered as public property made available to all, with no acquisition costs and a user license that provides total freedom to use the code and a guarantee that no one will be able to monopolize the source code (West, 2003). The organizational forms supporting and supervising the activities surrounding the production of such software are very diverse, but most of them have several main characteristics in common: volunteer work by the participants, long-distance work, little direct interactions, and absence of salary (Demazière et al., 2007). In this context, free software developers and contributors accept to dedicate working time and freely supply their know-how, despite the fact that they are not able to choose, limit or control those who are going to benefit from their efforts (Bitzer & Schröder, 2005; Lerner & Tirole, 2002). This situation is rather unusual compared to the common production modes managed by salary or contract-based work relationships or even legal mechanisms framed by employment or commercial law (Gensollen, 2007; Horn, 2004). The only

contractual link is established by the open license, which acts as a barrier against opportunistic behaviours (Dalle & Jullien, 2003), beside other tactics aimed at preventing proprietary appropriation, such as foundations, legal and normative sanctions, trademark on logo, etc. (O'Mahony, 2003). Finally, voluntary participation is, in principle, not based on any structured organizational system, such as, e.g., decentralization, openness, absence of specifications, deadline constraints, or specific coordination (Holtgrewe & Werle, 2001), but on a series of shared action principles: meritocracy based on competences alone, participative leadership, freedom of involvement, etc. (Moon & Sproull, 2002; Raymond, 1999; Hertel et al., 2003).

The ability to achieve solid and coherent results from spontaneous, sudden and evanescent involvement is viewed as an enigma by sociologists and economists. The endogenous literature offers an answer to this enigma, by promoting the 'bazaar model'. Although this concept is a core belief in the free software community, empirical validation is not convincing: a simple comparison between thousands of free software programs is enough to observe an extreme heterogeneity in the organizational modes and action processes, as well as in the contributors' involvement processes (Crowston & Howison, 2005). The widespread use of the term 'community' to name the forms of collective action suggests another answer to that enigma: the individuals participating in the free software development projects share values, norms and action principles that are strong enough to weld the groups together and channel their activity towards a shared objective. The 'bazaar' or 'community' categories obviously represent an ideal focal point. However, they provide little information on the way free software projects actually function. Because the members interact at a distance in the world of free software, using the facilities of the Internet network, the classical communities studied by anthropologists do not constitute an appropriate model (Baym, 1995; von Krogh & von Hippel, 2003).

Necessity and plurality of the regulations

Sociological analyses of free software development projects have shown that the success of some of them is not the result of some kind of mysterious alchemy or supernatural circumstances (Lin, 2005). The software production activity cannot be conducted without the mobilisation of individuals or the coordination of a collective. In other words, it takes the form of an organized and regulated action, structuring the exchanges between participants. The production modes of these projects are also hybrid, combining private and collective innovation models (von Hippel & Krogh, 2003). In organizational modes, it results in a combination between the required constraint-generating coordination and valorisation of individual investments. Thus the social integration of individuals within the communities has specificities requiring the production of operational rules to govern the conduct of both individual and collective projects simultaneously (Wenger, 1998), despite the fact that the more influential or central individuals cannot mandate or enforce the work (Himanen, 2001). Ostrom (1990) has evidenced the resistance of individuals to the enforcement of operational rules generated by outside authorities, and to rules which have not been sufficiently elaborated or discussed collectively. Moreover, these rules are only viewed as legitimate when they are tailored to local

conditions, and when the members of the community monitor the compliance of their enforcement. This leads us to point to the specific characteristics of the social control of the members, which cannot rely on traditional tools like sanctions and exclusion (Kollock & Smith, 1996), and the specific modes of dividing work and allocating authority (Garcia & Steinmueller, 2003).

Empirical studies of free software development communities suggest that social control is less preponderant than in other forms of organisations (Demil & Lecocq, 2006). However, that does not mean that simply because contributions are voluntary and not remunerated, there is no control, since field studies have also indicated that these communities are structured and differentiated (Stewart, 2005). In our opinion, internal structuring and differentiation are essential, at least when communities succeed in attracting new members and when projects go beyond the initial launching phase. These communities are constantly facing irreducible tensions: between internal unity guaranteeing the coherence of the collective action's product and the increasing diversity due to the success of the product, as well as between maximum openness favouring the enrolment of new members and access filtering to control the quality of the applicants. Our hypothesis is that these tensions are managed via the combination of various mechanisms to regulate the activity of these communities, as well as by their adjustment to the events and episodes marking the history of each project.

Therefore, we can consider these projects as 'going concerns' (Commons, 1961), in other words collective actions with some stability and continuity, achieving results approved by the users and based on relationships characterized by a mix of dependency and conflict. But, for these 'going concerns' to function, no matter what their characteristics, a minimum of order and common, shared, social rules are necessary. This order itself is generated by collective action, and is elaborated and transformed during exchanges and transactions between members. In other words, these projects are highly processual: they are characterized by negotiations and permanent adjustments. They are organized actions in which rules are interpreted and adjusted, and individual motivations and preferences altered (Atkinson & Reed, 1992).

Our objective is to describe and analyse these rule-creation processes. Our starting point is that the rules are not exogenous to social relations, and that they are not generated by a transcendental order: society is wired together by the interactions between its members, and the rule is a collective production (Blumer, 1969). The fact that rules are considered as immanent (Paradeise, 1989) has major consequences for the sociological study. First, it means that we must address the source of these regulations, in essence the situations prevailing at their creation and the interactions through which they are transformed. It favours the analysis of the dynamics of social regulation as opposed to the study of set and rigid rules. It also moves the focus of attention to the production of collective regulations.

The emergent central question from this orientation is the distribution of regulation capabilities. This distribution is described by sociological literature as lacking uniformity and instead depending on the position occupied and the frame of interactions it structures. In the extreme case, it is the actors occupying and monopolizing power positions who lead the organizations or the projects by producing and spreading the leading norms. This centralized regulation

model, usually called 'control regulation' contrasts with the aggregation of multiple 'autonomous regulations', which are mastered by many disjointed groups trying to get a grasp on the development conditions of their activity (Reynaud, 1988).

These analytical concepts have been tested on several fields but have never been put into play to study the functioning of collectives involved in free software production (De Terssac, 2003). In this specific case, autonomous regulation initially appears more significant than control regulation. In any case, it suggests the notion of 'communities'. However, from this regulation arises the problem, essential for software development, of the way product coherence is achieved and contributions are organized in an efficient, active digital text (Horn, 2004). Considering this constraint, we can support the idea of the coexistence of multiple forms of regulation in the activity of free software production. More specifically, we postulate that classical regulatory forms (control and autonomous) constitute an insufficient basis for the analysis, since these forms are too closely linked to formally hierarchical organizations. We hypothesize that in free software communities, the production of rules organizing collective action and structuring individual behaviours is widely socialized, with each participant being a potential producer of rules and controller of their application. Other, diffuse regulation mechanisms would thus come into play, completing the more classical forms. Our objective is thus to describe the plural regulations of the collective software production activity by opting for the analysis of the processes: how do rules prevail, and how are they discussed, how are they maintained? With a more theoretical perspective, the idea is to contribute to the theory of social regulation by showing the pertinence of its central concepts, but also their shortcomings, and by proposing a more articulate conceptualisation of the free software production.

A longitudinal and ethnographic approach

Our perspective led us to opt for a longitudinal ethnographic analysis of medium-size projects. Considering the fact that regulation processes are locally situated, an in-depth study must provide the opportunity to retrace the interaction systems leading to regulation by observing the exchanges between actors, the ways in which regulation or value conflicts are resolved, the organizational adjustments performed, the diverse forms of participation and the meanings of this experience. Moreover, since regulation is elaborated and iterative, a certain historicity should be considered. A longitudinal follow-up of the project is recommended in order to be able to grasp both the multiple temporalities characterizing the action, as well as the adjustment of the rules confronted by the evolution of the participation. In this context, medium-size projects represent perfect observation points. The projects involving very few individuals present very limited, or even absent forms of collective regulation. Large-scale projects instead develop more codified, stabilised and prescriptive forms of regulations, as shown in high-profile projects like Linux (Hertel et al., 2003 ; Lee & Cole, 2003), Apache (Franke & von Hippel, 2003) or Debian (Auray, 2004). The power of these projects in terms of reputation attracts numerous contributions, and favours the establishment of control, selection, and entry filtering procedures.

In this respect, Spip, an Internet publishing software, is an interesting project to analyze for several reasons.

First of all, it is a medium-size project (a few dozen active developers) that has reached the status of challenger within its niche. It is thus facing tough quality requirements, while the quality of contributions ranges from best to worst in projects (McKelvey, 2001). In addition, the Spip project is also characterized by the fact that it was initially supported by — and designed for — French left-wing activists, thus having a strong underlying political and ideological orientation. Our analysis of the project relies on empirical fieldwork, performed by the three authors between February 2005 and April 2007. In accordance with our research perspectives, we gradually plunged into this social world, like ethnographers who live for a while in the human communities they are studying. For the success of this 'virtual ethnography' (Hine, 2000), we had to get the contributors to trust us, since our objective was to harvest a broad range of material for closer examination to assess the functioning of the 'Spip community' and above all, to enquire into its regulation processes. In that context, running online questionnaires or using mailing lists were not the most appropriate investigation techniques. Our investigation had to favour the interactions, exchanges, and adjustments between participants in order to be able to identify the rules they structure and model.

To gain insight into the heart of the transactions, we had to establish close proximity to the developers, which made us choose an intensive, longer-term research design. This gave us the opportunity to combine several investigative methods, featuring the observation of on-line exchanges through multiple communication and cooperation channels (forums, IRC, CVS), 27 in-depth, one-on-one research interviews with participants involved in different activities and at different levels of commitment (initiators of the project, regular coders, occasional contributors, translators, debuggers, etc.) focusing on the individual contributions to the software, the collective functioning and the biographic itineraries, but also the direct observation of project meetings and the public restitution of the first analyses. The study helped describe the organization of the project and the developed activities, the characteristics of the participants and the meanings they assign to their contributions, the form of relationships developed between the contributors and the content of the exchanges — in other words, a sum of material that can be utilized to identify the rules supporting the collective that produces the Spip software.

Our argumentation will be developed in three phases. By retracing the genesis of the Spip project and its progressive structuring, we will examine the necessity for promoters to maintain coherence and preserve, over time, its conformity with a very strong original ideological project. However, this first form of control regulation requires that the autonomy of the most active contributors be acknowledged and encouraged. We will then focus on the approaches underlying the involvement of the participants occupying an intermediary position between the strategic core, including the initiators of the project, and the minor or volatile contributors, and show the specificities of autonomous regulation. Finally, we will show that contributor involvement does not derive exclusively from their individual activities, but that the meaning, value and legitimacy of this involvement are defined and rated more collectively, through exchanges, judgments, and evaluations between the participants of this project. This will

give us the opportunity to examine the 'distributed' aspect of this third form of regulation.

A control regulation focused on the product

The time-based analysis, focused on the maturing and the launch of the Spip software, underlines the stakes of the project in terms of growth, and maintenance of this growth. It indicates the production of rules, oriented towards two objectives: first, to mobilize many contributors who are willing and able to get involved in the development of the product, and then, to preserve the coherence of the software and safeguard its original identity. The convergence of these two objectives cannot be taken for granted; it requires the creation of specific regulations. Because of their position and their legitimacy acquired in the launch of the software, the initiators have strong capabilities in terms of initiative in these processes.

Origins and emergence

The Spip project started in the nineties. Its roots can be traced back to the French activist and associative community, experimenting ways by which Internet could contribute to enhancing freedom of expression. Grouped under the denomination *Minirézo*, some activists developed a publishing site with critical ideas about the development of Internet (*uzine.net*), and progressively reached an increasingly large group of actors with a manifesto published in February 1997, the 'Independent Web Manifesto'. Its promoters expressed their intent to fight the menace of commercial control on 'digital' freedom of speech. To help activists and sympathizers express themselves easily on the site, one of the initiators decided to develop a software tool to simplify Web page generation and site management. The project can be summarized as follows: to allow everyone, no matter what their level of technical skills and means, to be able to express his point of view on the Internet, freely and independently. This new software was initially designed exclusively for the operation of the *uzine.net* website. Upon request from sympathizers, it was later shared and then made public under GPL license. With this shift a software development phase emerged, one that ended up gathering a growing number of contributors and users.

Initially, the project was carried out by three persons, all closely involved with multiple activist networks: 1995 strikes, Act-Up, the movement for the defence of the illegal immigrants, etc. It was developed progressively and informally, without prior definition of explicit objectives, planning or anticipation of results:

> It's just a project, people were talking about something, and then we thought: instead of sending emails with a copy to everyone, we are going to build a list […] But actually, there was no organization, no big project, no master plan. It was just [that] among all projects, there was one, it was Spip. We were going to exchange our bits of code. (Gil, historical founder)

Then, slowly, exchanges intensified between three developers, who were contributing technical skills that were widely acquired by practice: one of them, trained as an airline pilot, was specialized in publishing scientific books; the second was a telecom engineer doing freelance work; and the third, a mathematician, was a webmaster for a major monthly opinion magazine. All three were involved in the 'Independent Web Manifesto'.

The core that was later to become the founders of Spip worked together on the definition of the software features, and agreed on the way to include political and ethical principles derived from their participation in *Minirezo*. So, the specificities of this publishing process on the web were established: projects proposals for articles within a private space, validation of articles by actors other than the editor (after debate on the forum), systematic addition of a forum below each article published, limitation of the number of decision levels, enhancement of the layout functionalities (and later, of language management), user-friendly interface, improved performance to allow installation on outdated servers, avoidance of user identification and tracking systems, etc. The objective was to translate the ideological orientations aimed at the support and promotion of freedom of speech on the Internet into technical choices.

After its launch, promoted by a wide communication campaign on the Internet, the software rapidly became a success, generating an increase, not only in the number of users, but also contributors. As a network of remote participants started to grow, and as the software extended its visibility beyond the world of activists, the motivations, behaviours, references and expectations started diversifying. This increasing heterogeneity corresponded both to a diversification of the institutional origins (simple individuals, associations, companies, public organizations, etc.), of the know-how proposed (translation, layout, documentation, coding, testing), or opportunities generated by the use of the software (increasingly close links with the professional activity, requests for services, the necessity to adapt the software to specific needs). Therefore, the question of the management of this diversity took on significant proportions, considering the strong hybridizing between the initial political project and the chosen development orientations. Ensuring software development requires attracting enough contributors to support the increasing extent and complexity of the project, but also necessitates channelling the input and to select the contributions as a function of the objectives pursued in terms of norms.

Growth and structuring

The success of a free software project rapidly generates a flow of questions: it calls for guidance and requests for advice, which must be addressed in order to reinforce the credibility of the product and the project. This increased load, corresponding to a growing user base, goes automatically together with increased pressure on the initiators and drivers of the project. It materializes in asymmetrical and unbalanced relationships, in which the recipient is assumed to master knowledge the other one is lacking. There is a non-negligible risk of these relationships imperceptibly getting closer to a client/supplier relationship and of seeing the development of a polarized world between a core of developers and a cloud of users.

In the case of Spip, there is indeed a central core, whose existence is not formally recognized, but widely known under the tag 'the four musketeers', composed of developers who have CVS access and the keepers of the politico-technical identity of the project. These core members are bound by strong interpersonal relationships, nurtured by frequent meetings outside the context of the project, shared philosophical or political orientations that also materialize in the support to other projects, actions, or activist campaigns. By contrast, the number of participants representing a

volatile, distant, uninvolved, anonymous or unstable relationship grew significantly. Unlike the members of the core, their investment was sporadic, irregular, and unpredictable. Their potential contributions are limited but not worthless: some documentation, bits of code, improvement ideas, identification of bugs are produced by them.

To face this situation (a few participants who are strongly involved in the development and increasing number of participants who only offer limited contributions), it is vital to reconfigure the collective action. The circle of significant contributors must be widened to be able to simultaneously ensure the continuous development of the software and to address the outside solicitations while concomitantly pushing back the boundaries to minimize the difference between simple users and contributors. It is essential to structure the mobilized collective around development, improvement, updates, and distribution of the software, without which the differentiation remains too strong between an active minority, limited to the circle of the initial producers, and a circle of users of varying size.

This structuring process, commanded by the central core, consisted in spotting the contributors who were particularly involved in order to delegate a few responsibilities to them, and simultaneously, to segment the internal organization into sub-spaces with specialized functions. Progressively, discussion lists appeared, dedicated to special themes, aimed at different types of contributors. Some of them were put in charge of the administration of these lists, and were thus placed in an intermediate position between the strategic core and the peripheral circle of the sporadic or volatile users and contributors. In most cases, this intermediate circle benefited from the approval and even the mandate from members of the strategic circle, in the form of providing advice, sharing confidential information, and establishing privileged contacts. The fact that they delegated responsibility for the animation of the community can also be perceived as granting a special status to reward the contributors with the strongest involvement, and acts as a motivation factor as well.

Autonomous regulations based on differentiated commitments

Control regulation is exerted on the software product. It consists in centralizing the activities related to the assessment, selection, and adaptation of contributions, in order to maintain their technical coherence and preserve their conformity with the original ideological project over time. It does not, however, result in the establishment of barriers to entry or taking social control of participant ideological orientation. An underlying prerequisite for the continuity of the Spip project is its capacity to mobilize and recruit new participants. The risk inherent to openness is to see the 'community' crumble or fragment into small entities, because of increasing internal heterogeneity and the emergence of conflicts, notably in terms of values. Even without considering participants who provide minor or occasional contributions, developers with significant and sustained investments produce widely divergent interpretations of their commitment and their participation. What is more, the analysis of the motives invoked, the interests pursued and the justifications put forward show the coexistence of contradictory and conflicting significations. These are not merely incentives, even if they provide a meaning, a purpose, rationales, to

the activity (Gosh et al., 2002). They can be considered as consistent sets of symbols, all the more resistant since they are strongly rooted in their biographic itineraries, generating autonomous action rules, and more or less disjointed from control regulation.

It is for the contributors engaged in these intermediate positions that the question of the involvement in the Spip project seems to be most interesting. They show a stronger and more sustained involvement than the peripheral participants who only produce sporadic and minor contributions (bug reports, for example), or mere users who flag errors that are obstacles to their own use of the software. At the same time, they differ from the members of the strategic core controlling the development of the software and form an inter-knowledge group bound together by common activist experiences. The intermediate position could be characterized by the intensity of the involvement and the responsibilities taken on, generating a significant workload together with a limited retribution due notably to the fact that the strategic core of the initiators remains closed. Light can be shed on the involvement of these contributors through the analysis of the biographic factors underlying their participation in the collective action. In this context, based on the biographic interviews performed (Strauss & Corbin, 1990; Demazière & Dubar, 1997), we were able to identify three types of trajectories that each combined special involvement outside the project, personal use of the software, as well as privileged projection spaces.

A politico-technological activist approach

For the first set of trajectories, the involvement in the Spip project, and more globally in the world of free software, is one in a series of activist experiences, materialized by memberships in associations, political parties, and social movements. These activities are variable, as a function of the individual's life history, but they situate the ideological orientations around the non-institutional left-wing community. Ernesto considers his political involvement as evidence and a direct consequence of his personal history: "I was born Uruguayan and I'm a political exile, thus I fell into political commitment without even knowing it, if you see what I mean". Ferdinand's political involvement is less steady, but is however part of his life trajectory: "in 2004, I decided to get back in touch with the political activist action I had abandoned for several years". Georges, on the other hand, has accumulated a very solid and diversified experience as an activist over his student years: "I always had multiple involvements, since you're always interested by many other things, it's the way you build your political conscience". There are many other examples indicating diversified but recurrent processes of socialization toward collective action. This presents a similitude with the activist approach of the initiators of the Spip project, but it is far more diversified and does not constitute a real community of shared experiences.

The first contact with the Spip project is often linked to other involvements. The typical case is the one in which the project to develop an Internet site for an activist association or organization leads to the use of Spip: an initial involvement goes through the question/answer routine on the user help site, then the answers to other users; slowly, the political dimension of the software comes forward, through individual exchanges, debates resonating with personal political sensibilities. As Georges points out: "There are very few

free software projects in which there is a political dimension that's so strongly defined. (…) we don't hesitate having politically- and ethically-oriented discussions, including their impact on features. (…) Here at Spip, we examine the notion of 'what model of society should we promote through this'". The Spip project becomes a *cause* to support, and the search for new collaborators becomes an objective: "very soon, we start to feel co-responsible for this thing, and to think that it's a good thing that this project can survive, I really would like to support it, and that's why we're answering the users (…) it's because we consider that's interesting for Spip to have a pool of tens of thousands of users. And yes, it's obvious; we feel that we are really part of the game".

The involvement in the project intersects other activist implications, since the participants concerned are then able to put into play the resources derived from their experience as activists, as well as the technical competence already developed for the creation of sites.

A 'technological fun' approach

A second approach to enter the project is shared by individuals who justify their initial commitment by mostly technological interest. In this case, technology is a stake in their personal and technological evolution. Their IT activity is considered as a mode of expression and creativity, favouring the resolution of problems by trial and error, patching, technical challenges, and more generally a passionate relationship with technology and innovation.

Initially, participation in the Spip project was for these people motivated by curiosity for the technical developments of the product: the goal is not to find a tool that can be used to develop a site, but rather to find a stimulating space to exert one's competence as a developer. Charles points out that before discovering the Spip project, he would wander from one small project to another (re-writing the features of a genealogy software package, developing a graphical interface for software comparing telephone rates, etc.), until the day he got interested in CMS. He followed the new trends, tried several software packages, without, however, looking for a particular use: "I had no particular need on … I do not develop web sites, I had one personal web site, so I had tried it in Spip, just like that, for the fun. Thus I tried Spip and liked it". For Nicolas, the logic is the same: "in my case, I fell for the technical side rather than the user side, since I am an IT professional, when I heard about that thing, I was more interested in discovering how it worked than discovering the possible applications, since at that time, I had no specific need for the product. It was more about technical curiosity".

Here the investment in the Spip project can be described as a type of hobby, the technical aspect of which constitutes the motor and the fuel. Participation is immediately materialized by proposing bits of code, 'contribs'. When Nicolas submits these 'bits of code' and these contributions are not selected, he is not discouraged, since the stakes are not high for him: "I do that mostly to kill time; I really had no serious needs". The choice of Spip is also motivated by the relatively small size of the project, in contrast with the 'big' projects, like Mozilla or Linux, "(…) actually, Spip still has an almost human size while being big enough to have a lively community, and that makes you want to chat with the people who are involved".

The contributors with this approach are IT professionals, but their professional activity has no real direct link to the Spip software, or more globally to free

software. Most of the time, their work revolves around proprietary technologies that they are in charge of implementing or maintaining for clients through service providers, for example in banking institutions. We could hypothesize that their involvement in the project provides the opportunity for those IT professionals to get back in touch with the roots of their job, while the technologies favoured in their professional context do not allow them access to the technical black box, to be able to understand by themselves how the software functions, to be able to hack appropriated technical solutions, to prove their creativity, their technical boldness, in short, to achieve a certain intimacy with technology (Kunda, 1992). The Spip project thus becomes a sort of *technological playground*, an opportunity to meet and develop projects, of which the usefulness is not evaluated in terms of answer to needs or a cost/benefit ratio, but rather according to criteria of 'technical beauty', challenge, and wit. We will refer to this trajectory as the 'technological fun' approach.

A business-oriented professionalism

Finally, for a third group of participants, participation in Spip is not linked to a practical interest deriving from activist activities, nor from a taste for technological challenges, but is more closely associated with their professional trajectory. More precisely, Spip is a resource to develop their professional and commercial activities — these people are usually freelancers. In this case, the objective pursued is very concretely to develop an economic model that allows them to bank on their software knowledge with clients while participating in the development of the product. With a marketing degree, Rene, a former radio presenter, would like to develop new models of (virtual) consumption for the cultural industry. The development of websites for clients provides him with the opportunity to develop a R&D activity, hoping that one day one of his ideas will capture the attention of investors. Unemployed, Damien took the time to train himself to the use of Spip and started a website development business mostly for public contractors: "in this unemployment period, it was vital for me to make money (…) thus I read the whole manual before whipping up my first Spip site (…) and then Spip became another way of continuing to make a living". Others with this approach are freelance workers in the domain of Internet technologies and consider the fact of developing an activity around Spip as a new tool in their tool kit. This is the case for Michel, who works with Spip for several clients: "there are people who want to sell tailor-made products for clients. And I do the same, to earn some money these days. There are more and more of us doing it, freelancers who have found Spip, and who have understood that it is a good tool".

The fact that Spip has been considered from the onset as a professional tool designed to be proposed to clients, leads them to develop arguments indicating the merits, specificities and qualities of the software and to justify their specialization in that niche, as for example Damien puts it: "one very important thing is that Spip is a very good product. It provides excellent opportunities to develop a client base". In that context, participation in the project, whether it involves helping users, proposing contributions, discussing features, correcting bugs, is motivated by a will to achieve a fine grasp of the software and its specificities, its features and its essential flaws. For them, it is an opportunity to build up and sharpen their know-how, to acquire true professional expertise on the product, and demonstrate their reliability to

clients by showing off their competence, with the self-interested objective of a return on investment under the form of commercial contracts.

The diversity of these involvement trajectories clearly demonstrates that the most active contributors who have joined the intermediate circle of the Spip project do not share the same community background with the members of the central core. More specifically, some of these regular contributors get close to it, while others are indifferent or ignore this dimension of the project, and still others keep their distance or are even hostile. In fact, these contributors go as far as defining their own exchange and activity space (pure R&D trials, information exchanges for freelancers, etc.), thus defining their alternative action rules and producing an autonomous regulation.

A distributed community regulation based on judgements

How can commitments that are so dissimilar be put together? A closer look at the crossroads between control and autonomous regulation modes is necessary to understand how the organized construction of heterogeneity takes place. This implies a shift of focus to the interactions between participants. In this perspective, the difference between the statutes and the legitimacies conferred to the contributors appeared as processes shedding a revealing light on the creation and transformation of shared rules.

Our observations of the social exchanges developed within this project, through online exchanges on forums and on IRC, but also during group meetings helped us identify two distinct mechanisms through which contributors are differentiated: one — *recognizing the deserving* — is involved in the distribution of the roles associated with positions in the organizational chart of the project. The other — *building reputations* — is more informal and concerns building the opinions on which reputations are built. In both cases, the regulation based on the principle of these two differentiations relies on the production of judgements by other participants who take part in the exchanges, give their opinion, and adapt to the stands taken by others. This regulation is consequently widely distributed within the group.

Recognizing the deserving

Delegation of responsibilities involved an acknowledgement of efforts and input, establishing a value for the contributions where the individual contributors have given the best of themselves: “I think it’s normal that at a certain point, they tell you: you’re doing a good job, you’re going to get more to do, to organize things”. Sometimes, it is even perceived as the achievement of an actual position in the organization of the project, symbolizing even a promotion, “we do that for free, but that doesn’t mean that there shouldn’t be any recognition, so we can feel we are moving up within this thing”. These interpretations are criticized by the members of the strategic circle, but they concede that the delegation of responsibilities is appropriated as a powerful recognition marker, to signify a change of status, a sort of ennoblement: “for people, it’s a bit like a lollipop, it’s more than a reward, it’s like being dubbed a knight. It’s not really our mentality, but it is perceived as such”. In a context of activity where exclusion or eviction is not an option since one of the levers for success resides in the capacity to attract and enrol a large number of contributors, these forms of promotion appear as one of the few possible positive sanctions available.

Rewards are the consequence of significant contributions, which means that they combine several qualities: they correspond to major efforts that alleviate the core member's workload, they involve a significant investment in terms of time, and they occur over a sufficiently large period of time. These qualities are continuous characteristics, so that their identification requires an activity of assessment and evaluation. But estimates can be erroneous: "what matters for us really is the investment in terms of time. We watch, we observe, we spot […] but it is a delicate matter to judge someone's reliability, since you'll have people who will do a great job, and then when they become administrator, they no longer do anything, they feel like they have made it".

So, beyond the quality of the contributions, it is the reliability of the contributor that becomes the subject of assessments and that constitutes pertinent or even decisive information for the transfer of responsibility. But what does this reliability involve? It is related to the position taken towards the project, which can take two quite distinct and widely independent forms. The first is linked to participation and production, to the capacity to materialize contributions or finalize projects that require persistence. In this context, reliability means meeting one's commitments, even implicit, and being autonomous enough to produce tangible contributions. It is the stability of the involvement, assessed in an action-oriented framework, which is at play here. The second dimension involves the sense of commitment, and no longer its result. It refers to the compliance to the spirit of the project — rather than to its production — which rested on clearly stated political colour and ideological identity. Reliability here is expressed in terms of ideological affinities, particularly concerning the explicit agreement to the principles of 'the Independent Web Manifesto'. This does not imply involvement or participation in activist or political organizations, but it does require the expression of orientations in terms of values justifying the participation in the project.

Recognition through values is not always expressed explicitly, but it is materialized by the will to better know the contributors who seem to be reliable through their production. Informal or private contacts are then organized (telephone conversation, direct meeting, lunch, etc.) in order to evaluate — or rather to check — several values behind the involvement: "someone who does such a job can't be a freebie, can't be a dangerous freak either. We know lots of cool people, but you can't count on them for big jobs […] So then, we try to have lunch together, talk it over, to get to know them better, you know". This second test, after the contribution criteria, is all the more determinant when the responsibilities to be delegated are important and bring people closer to the core. They also discriminate between the involvements rooted in trajectories, similar to those we refer to as politico-technological activist action.

However, important responsibilities are also conferred to contributors who justify their participation by their taste for computer technology rather than by political or ethical values. These are individuals whose trajectory is related to the 'technological fun' approach, but who slowly have shown signs of socialization and adhesion, for example by taking part in non-technical discussions (during meetings with contributors or on IRC channels). More generally, the most technically competent contributors benefit from extra consideration, since their production is precious, even if the sense of their involvement is unclear.

They are granted positions in which they can exert their creativity. However, they are not on a promotional track. Instead, they reveal a horizontal work division: some mostly technical spaces are dedicated to them. They are thus subject to specific processes of recognition and valorisation.

Building reputations

Another mechanism of differential involvement valorisation is based on a discursive production of judgments that tends to control deviant individuals. It takes the form of various types of social control (reminders of behavioural rules, mocking, sarcastic remarks, etc.), which are very widespread within the collective. This regulation strongly contributes to the production and the maintenance of group cohesion, by expressing connivance and complicity between its members, all the while being already the result of this cohesion. We know that the Spip software originated in a project to develop means of free expression on the Internet. This objective, political, ethical, or ideological aim, is included in the product itself, both in the principles it edicts in order to structure the web sites and in its low requirements in terms of IT skills, which makes it accessible to a wide range of users. It contributes to the construction of legitimacy orders that are used as references for the differential valorisation of attitudes and behaviours of the contributors. Also, the roots in a left-wing scene, reinforced by the affirmation of a principle of selflessness, automatically de-valorises any commercial use of the software. In such a cognitive context, the contributors, whose involvement we refer to as *business-oriented professionalism*, are out of alignment. Therefore, one of the stakes of the upkeep of the software identity and the project survival is to contain the participation of these contributors while benefiting from their contribution.

What characterizes this type of project is that the social control here is widely distributed between the members and is not generated by a powerful strategic circle being able to sanction, ban, or limit the intervention of deviant individuals. Here, this process is collective, and above all extremely dispersed. This regulation is strongly integrated into ordinary exchanges, as if embedded in the group's life, and is thus all the more effective. Continuous observation of these interactions has helped us identify several mechanisms contributing to this regulation.

The first mechanism is based on an explicit reminder of the initial reference to the project. It is not targeted at any type of participants in particular, but it contributes to the diffusion of legitimate and essential values. Examples are plenty. So in 2005, the launch of a new development space (Spip-zone) provided the opportunity to put the emphasis on the original values. Various actors interested by the launch of this new sub-space started to discuss, in parallel, the formalization of a charter, the purpose of which was to manage participation and strongly reaffirm the identity of the project. Accordingly, this charter states:

> Let us remind you that Spip is free software, and that any person who uses it can do whatever he wants with it; however, the participation to the Spip-zone must take place within the framework of the goals and values promoted by the initial Minirézo project, and notably to promote and defend freedom of speech for all on the Internet, to remain defiant towards financial interests, and to respect the identity of each and everyone. This implies an effort to

> internationalize one's contributions, make sure that the language and functioning modes chosen are not sexist, giving priority to the needs of the associative world rather than business needs, etc. This site is not a development platform for military or business-oriented versions of Spip, which would change its nature. Neither is it designed to be used as a communication or advertising media for consultants.

The objective is explicitly to restrain the intervention of 'business people', those whose interest is to 'bang on their own drum' by multiplying their personal signatures in all discussion threads. The choice of elaborating a charter to regulate the participation to this sub-workspace is both linked to a desire to make an initial selection of the participants (avoiding consultants in search of reputation) and to promote practices of other sub-spaces that are worth supporting (awareness of internationalization, non-sexism). However, the terms of this charter are also subject to interpretation: individuals adjust their individual practices and strategies to work with these norms. So, some contributors who exert an independent activity around Spip, learn how to feel the limits that should not be trespassed, and adopt, for example, signature strategies to avoid any reference to the business side of their activities: "within the community, some aspects were ill-perceived […] as if my behaviour consisted in looking for clients at all costs, far and wide, non-stop, to dream of clients, to see potential clients everywhere. Thus, when I understood that the community felt bad about it, I shortened my signature in the developers' group and now I simply sign Damien".

Another mechanism for the production of opinion is both more collective and more personalized: it is driven by a series of contributors during public exchanges, and it is oriented towards certain explicitly targeted participants. In this context, instant messaging (IRC) constitutes a major resource for the diffusion of values and rules of conduct, the production of judgments and reputations. The targets of these exchanges are always those who deviate from the boundaries of the involvement or those who are suspected of straying from the rules. The contributors concerned are active as computer consultants, and their contributions, even when they are very solid, are always interpreted as business strategies rather than selfless contributions to the collective project. Here, call to order are less formalized than in the previous case, since regulation is totally embedded in the flow of exchanges and discussions: triggered by one of the forum participants in reaction to an opinion expressed or a proposed initiative, it is relayed and amplified by others. Expression styles like jabs, mockery, or irony are privileged since they allow incisive reactions while saving, to a certain extent, the face of the person criticized, in order not to compromise his/her participation to the project.

The examples are numerous. For instance, Alex, known as IT freelancer, faced a fierce argument. One of his technical contributions was strongly criticized. It consisted of a *plug-in* to Spip, i.e., an optional complementary module implementing advanced typographic features, called the *typo bar*. The product code was thought to 'trap' the user, making it very difficult, once the module is installed, to get back to the initial status.

> <NeroSR> it's a trap; it is useless
> <Kriss> an "anti-typo bar" plug-in should be developed ;-)
> <NeroSR> it's really stupid to design a thing that gets you trapped
> <NeroSR> it should be deleted so no one gets caught again
> <Boggy> it's a fork, it's a trap, it's the «Alex trap»
> <Boggy> Alex: your typo bar, you can keep it to yourself, you have put enough people in trouble already with that thing
> <Carla> Boggy: why do you call it a fork?
> <Boggy> once installed, people are linked to his bar, and thus to him. It makes the content of the site fork.
> (Spip IRC channel, February 7, 2007)

Alex's answering strategy consisted in justifying the code produced by technical difficulties. The others retorted in unison that "you should not confuse requirements with techniques", and to "rewrite the whole thing, obviously". This argument resumed after Alex posted a tasteless comment with homophobic innuendos, as a reply to a request for technical help on the user's list. Reactions were strong, condemning the comments as incompatible with the community's values. Parallels were also established with Alex's positions published on his personal site. On the IRC channel, other participants again started to criticize his contributions, and make fun of his user-trapping strategy. Aspirin's sardonic comment was: "apparently it's not easy to make big money with the typo bar". In other words, beyond the expression of technical difficulties, it is a condemnation of the commercial strategy of making users captive. This leads Alex to testify, during an interview, "this taught me a first lesson, or rather, a conclusion: to participate in the Spip community, you need a lot of abnegation. Which means taking a few hard knocks to the head and still holding on".

'Real life' meetings during which the members of the group physically meet each other are another occasion to develop distributed regulation activities. These meetings often take different forms, from regular meetings organized by city where contributors and new users discuss their project (the Apéro-Spip), to annual 'technical' meetings, hosting the main contributors, or project parties open to a wide circle of contributors, sympathizers and ordinary users.

While some of the participants do not know each other beforehand, other than through their online nicknames, all have a reputation based on their contributions (one is known as the person who re-wrote this part of the software, translated the documentation into this language, or worked for a questionable client), or from the opinions they have expressed on the forums (this is the one who reacted most vigorously against that other participant, that one animated the debate on that subject). These meetings then become opportunities to reinforce connivance and rejection, and thus to support judgment-based distributed regulation. These meetings themselves provide the occasion to apply these overlapping judgements and therefore constitute significant regulation spaces, as indicated by the observation notes taken during a so-called technical meeting. It shows that interactions are created to reinforce the sharing of views and interpretations, to generate communion in one single vision of the collective project:

> We were sitting around a table during a Spip workshop. The table was U-shaped; the speaker displayed the results of his work and notes comments on the whiteboard. The vast

> majority of the 30-odd participants were connected to the project's IRC channel through their laptops, while listening to the presentation. At a certain point, muffled laughter was heard. My neighbour showed me a picture of a dead, squashed squirrel [the squirrel is the symbol of the Spip project]. He then showed me the content of the attached IRC message saying "Figo, can you show this to your neighbour, it's the Agora logo". This muffled laughter and the accompanying glances pointed to an individual present in the room, sitting next to Figo. He was the representative of the Agora project, an aborted attempt for a Spip fork. He came to the meeting to try to favour a merge of the contributions of the two projects. He was unaware that he was being made fun of by almost the entire room. (Spip 'coding party', Lille, France, March 2006).

Regulation by and within the interactions between participants takes multiple and ever-renewed forms, but is based on a discursive production that leads to the establishment of norms, by stigmatizing specific behaviours, by reinforcing socialization, and in some extreme cases, by rejecting any individual who attempts to ignore the collective ban. It is mainly based on two types of proofs according to the principle of contributor differentiation and legitimacy: the contribution as proof revealing competence, both technical and managerial, and the proven signification revealing their orientations, in terms of both value and ideology. These proofs are powerful mechanisms to grant recognition and legitimacy to the contributors. This regulation is continuous, since it is instilled within the exchanges and activities which make the project progress. It is also public since it takes place in spaces that are visited, at least potentially, by all participants: everyone's contribution to the development of the product, investment in the collective action, as well as points of view are converted into value and legitimacy, to the extent that they are readable, perceptible, public. Consequently, all the participants contribute to the production of these rules, either explicitly by intervening in forums and IRC or during one-on-one meetings, or implicitly by maintaining the consensus on legitimacy or status of one or the other participants. The meaning of these commitments is also a collective production since they interact with the values associated with the collective action, and since it is subject to the perceptions, interpretations and judgments produced by the other participants.

Conclusions

The in-depth ethnography of the Spip project indicates a multiplicity of mechanisms regulating collective action that we labelled with three terms: control, autonomous and distributed. Beyond this case study, this diversity is probably typical of medium-sized communities, since they face permanent uncertainty regarding the quality of the participants and the reliability of their commitments (Stewart, 2005), and lack adequate resources to define routines and stabilize interactions. In contrast with the most highly reputed projects with a strong power of attraction, they cannot rely mainly on codified control regulation tools without the risk of provoking the desertion of many participants. On the other hand, unlike the smaller-scale projects supported by small affinity-based communities, they cannot rely only on the autonomous regulations rooted in the commitments of the participants

alone, because it could result in the disintegration of the community.

We were able to establish that these two types of regulation are not contradictory, but instead reinforce each other. Close and centralized control exerted on software development guarantees — and that is actually the explicit objective — its coherence over time, without which the success in terms of distribution and use is impossible or short-lived. This regulation based on a more or less complex and codified hierarchical structure is observed in many free software communities (West, 2003; Lakhani & von Hippel, 2003; Lee & Cole, 2003). Our contribution is to show that they do not constitute a constraint for the participants: it rather appears as a guarantee that their efforts will not be in vain, but instead that they will be rewarded by the success of the product and of the project. Whether Spip contributors are independent web supporters, computer technology enthusiasts, or consultants looking for a market, they all share, beyond the diversity of their commitments and their motivations to act, a strong interest in the success of the project. The centralized control on the project constitutes a defence of this interest since it guarantees the coherence of the result.

The combination between these two classical forms of regulation is necessary in order to manage the irreducible tension between unity and diversity, coherence and abundance. But this combination is not sufficient since the interdependence between control regulation and autonomous regulation does not cancel out the tensions and the contradictions rooted in the double constraint of result coherence and significant heterogeneity of the contributors. We have introduced a third, original form of regulation of the community and its activity, which we called distributed regulation. This does not mean that the rule-making power is distributed equally among the participants — our observations have clearly ruled out this eventuality — but that it is dispersed throughout the group. It also means that it is adapted according to vague criteria regarding the forms of presence of each participant in the interactions. This presence can be assessed, depending on the case, in terms of number of contributions, time dedicated to the project, legitimacy of the interventions, accumulated status — all various ingredients that can be subject to multiple interpretations, but on which the reputation of each participant is based, as well as his symbolic position within the group.

These results coincide with other works that have examined the hybrid and entangled nature of classifying the structuring of productive organizations (market, hierarchy or network) (Powell, 1990) or of free software production (Demil & Lecocq, 2006). The ethnographic approach that we chose underlined the interdependence between the three regulation modes: they acted as a system and it is their hybridization that supports the development of the Spip community. First of all, the control regulation, focused on the software produced collectively, is vital in order to avoid the dispersion of contributions or forking. Autonomous regulation is necessary to allow the integration of new participants who contribute to the success of the common action without sharing the values and the founding experiences, and to allow initiative taking, which is an essential element of the vitality of collective activities. Finally, distributed regulation ensures an equilibrium between opposed or even conflicting dynamics and guarantees that none of the two other regulation forms wins out over the other, neither formalizing and rigidifying

the organization, nor dispersing and diluting it. It represents a flexible form of production and implementation of social rules, which tend to delimit the attitudes and behaviours acceptable for the group. It is therefore a vector of social control, diffuse but nevertheless effective, since the exchanges and interactions take place in public or semi-public spaces (the forums, virtual or not). In that sense, the attention brought to the situations in which rules are created and transformed is a path that deserves deeper analysis, by developing methods to observe the exchanges and the functioning of these 'communities', derived from anthropologic work.

References

Atkinson, Glen & Mike Reed (1992) 'The individual in a going concern', Journal of Economic Issues 26 (2): 469-476.

Auray, Nicolas (2004) 'La régulation de la connaissance : arbitrage sur la taille et gestion aux frontières dans la communauté Debian', Revue d'économie politique 113: 74-99.

Baym, Nancy (1995) 'The emergence of community in computer-mediated communication', in S. Jones (ed) Cybersociety: Computer-mediated Communication and Community (Thousand Oaks, CA: Sage): 138-163.

Bitzer, Jurgen & Philipp Schröder (2005) 'Bug-fixing and code-writing: the private provision of open source software', Information Economics and Policy 17 (3): 389-406.

Blumer, Herbert (1969) Symbolic Interactionism: Perspective and Method (Englewood Cliffs, NJ: Prentice-Hall).

Commons, John (1961) Institutional Economics: Its Place in Political Economy (Madison: The University of Wisconsin Press).

Crowston, Kevin & James Howison (2005) 'The social structure of free and open source software development', First Monday 10 (2).

Dalle, Jean-Michel & Nicolas Jullien (2003) 'Libre' software: Turning fads into institutions?', Research Policy 32 (1): 1-11.

De Terssac, Gilbert (eds) (2003) La Théorie de la Régulation Sociale de J.D. Reynaud (Paris: La Découverte).

Demazière, Didier & Claude Dubar (1997) Analyser les Entretiens Biographiques. L'exemple des récits d'insertion (Paris: Nathan).

Demazière, Didier, François Horn & Marc Zune (2007) 'Des relations de travail sans règles ? L'énigme de la production des logiciels libres', Sociétés contemporaines 66(02): 101-125.

Demil, Benoît & Xavier Lecocq (2006) 'Neither market nor hierarchy or network: the emergence of bazaar gouvernance', Organization Studies 27(10): 1447-1466.

Franke, Nikolaus & Eric von Hippel (2003) 'Satisfying heterogeneous user needs via innovation toolkits: The case of Apache security software', Research Policy 32(7): 1199-1215.

Garcia, Juan Mateos & Eduard Steinmueller (2003) 'The open source way of working: A new paradigm for the division of labour in software development?', SPRU Electronic Working Papers Series (92).

Gensollen, Michel (2007) 'Information goods and online communities', in E. Brousseau & N. Curien (eds) Internet and Digital Economics (New York: Cambridge University Press).

Ghosh, Rishab, Ruedigger Glott, Bernhard Krieger, & Gregorio Robles (2002) The Free/Libre and F/OSS Software Developers Survey and Study — FLOSS Final Report (University of Maastricht).

Hertel, Guido, Sven Niednet & Stefanie Herrmann (2003) 'Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel', Research policy 32(7): 1159-1177.

Himanen, Pekka, Linus Torvalds & Manuel Castells (2001) The Hacker Ethic (New York: Random House).

Hine, Christine (2000) Virtual Ethnography (London: Sage).

Holtgrewe, Ursula & Raymund Werle (2001) 'De-commodifying software? Open source software between business strategy and social movement', Science Studies 14(2): 43-65.

Horn, François (2004) L'Économie du Logiciel (Paris: La Découverte).

Kollock, Peter & Marc Smith (1996) 'Managing the virtual commons: Cooperation and conflict in computer communities', in S. Herring (eds) Computer-Mediated Communication: Linguistic, Social and Cross-Cultural Perspectives (Amsterdam: John Benjamins): 109-128.

Kunda, Gideon (1992) Engineering Culture: Control and Commitment in a High Tech Corporation (Philadelphia: Temple University Press).

Lakhani, Karim & Eric von Hippel (2003) 'How open source software works: 'Free' user-to-user assistance', Research Policy 32(6): 923-943.

Lee, Gwendolyn & Robert Cole (2003) 'From a form-based to a community-based model of knowledge creation: the case of the Linux kernel development', Organization Science 14(6): 603-649.

Lerner, Josh & Jean Tirole (2002) 'Some simple economics of open source', Journal of Industrial Economics 50(2): 197-234.

Lin, Yuwei (2005) 'The future of sociology of Floss', First Monday, Special Issue #2.

McKelvey, Maureen (2001) 'The economic dynamics of software: three competing business models exemplified through Microsoft, Netscape and Linux', Economics of Innovation and New Technology 10(2-3): 199-236.

Moon, Jae Yun & Lee Sproull (2002) 'Essence of distributed work: The case of the Linux kernel', in P. Hinds & S. Kiesler (eds) Distributed Work (Cambridge: MIT Press): 381-404.

O'Mahony, Siobhan (2003) 'Guarding the commons: how community managed software projects protect their work', Research Policy 32(7): 1179-1198.

Ostrom, Elinor (1990) Governing the Commons: The evolution of institutions for collective action (New York: Cambridge University Press).

Paradeise, Catherine (1989) 'Autour des règles du jeu', Cahiers de recherche (Lyon: GLYSI).

Powell, Walter (1990) 'Neither market nor hierarchy: Network forms of organization', in B. Staw & L.L. Cummings (eds) Research in Organizational Behaviour 12: 295-336.

Raymond, Eric (1999) The Cathedral & The Bazaar (Cambridge, MA: O'Reilly).

Reynaud, Jean-Daniel (1988) 'Les régulations dans les organisations: régulation de contrôle et régulation autonome', Revue française de sociologie 29(1): 5-18.

Stewart, Daniel (2005) 'Social status in an open-source community', American Sociological Review 70(5): 823-842.

Strauss, Anselm & Juliet Corbin (1990) Basics of Qualitative Research: Grounded Theory procedures and techniques (Newbury Park, CA: Sage).

von Hippel, Eric & Georg von Krogh (2003) 'Open source software and the 'private-collective' innovation model: Issues for organization science', Organization Science 14(2): 209-223.

von Krogh, Georg & Eric von Hippel (2003) 'Editorial : Special issue on open source software development', Research Policy 32(7): 1149-1157.

Wenger, Etienne (1998) Communities of Practice (Cambridge: Cambridge University Press).

West, Joel (2003) 'How open is open enough? Melding proprietary and open-source platform strategies', Research Policy 32(7): 1259-1285.

Didier Demazière
Laboratoire Printemps, CNRS
University of Versailles
Saint-Quentin-en-Yvelines

François Horn
CLERSE - CNRS, University of Lille III

Marc Zune
FNRS - Free University of Brussels &
Management Centre, Queen Mary -
University of London
mzune@ulb.ac.be