

# ***Free Software and the Double Life of Computing Professionals: Some Biographical Insights in the Life Courses of Some Elder Developers.***

Auray, Nicolas. (ENST-EHESS, France) [auray@enst.fr](mailto:auray@enst.fr), Vicente, Michaël. (UTC Compiègne, France) [michael.vicente@utc.fr](mailto:michael.vicente@utc.fr)<sup>1</sup>

## **Abstract.**

The literature on motivation of Open source developers is ground in a dualistic distinction, some authors emphasise the interest oriented motivations, other stress on the hedonist motivations. In order to remedy this barren opposition and because we think that the real sense of motivation emerge from action and not simply from statement of intent, we proposed to do a qualitative and longitudinal study of these motivations, based on a questionnaire, interviews and observations with two well known communities, Debian and Open BSD. Ours main conclusions figure out the fact that in almost every cases this involvement is a reaction with a current professional situation and that we can distinguish two kinds of reactions, the first one monolithic and the second one dual. These trajectories are feed by external context, but are also determined by the nature of the developers courses within the project.

## **Keywords.**

Open Source software, biographical trajectories.

---

<sup>1</sup> This study has been done thanks to the European Calibre Program.

Open source software projects and development processes have spread rapidly and widely, and many thousands exist today; for example Weiss (2005) quantified only on Source Forge more than 27 000 different Open Source software projects. The number of developers participating in each project ranges from a few to many thousands, and the histogram distribution of the number of developers in a project follows a power law (von Krogh and von Hippel 2003). So, two different problematics have been explored by the social scientists. The first one deals with the *detection* of community and tries to understand the conditions to initiate the “commons”: what is the critical mass, how to struggle against free riders (Lerner Tirole 2000), how to discover a community structure within organizations (Tyler Wilkinson Huberman 2004). The second one deals with the regulation and the *management* of big community projects, which attracts a lot of user queries and face particularly spamming and over-supply problems : for example, they must split in separated “communities”, specially a user-centered one and a developer-centered one, which however must be subtly articulated or intertwined. They must also activate strong recruitment procedures to avoid the flow of those attracted by the signal incentives who could decrease the quality of the code (Auray 2003). The following paper falls in with this second series of reflexions. But it tries to shift the problematics from the projects themselves to the *developers* who built them, and to formulate a new methodology to understand the motivations of contributors to big international Open Source software projects. The approach was initiated during a deep ethnographical inquiry of two FOSS communities of developers (Vicente 2005)

Our approach is concerned with critics of the pertinence of the motivation theory: in fact, when applied to Free/Open Source involvements, the motivation theory leads to contradictory results<sup>2</sup>. Some authors,

---

<sup>2</sup> For a sharper critical analysis of motivation theory, see Vicente (2005).

especially Himanem (2001), or Torvalds (2002) coining the Maslow theory, dwell on pleasure and creativity to explain the prior motivation of free software developers; others, like Lerner or Tirole (2003), insist on reputation concerns and on cost benefit trade-offs. When interviewed on their motivations, the developers exhibit a very differencing view (Lakhani 2003). What it means is that the motivations seem not to correspond with declared intents, or the contrary they emerge from the course of action. So as pointed by the interactionist theory (Hughes 1937, Becker 1966), a series of tiny decisions can better explain the social involvement mechanisms than the clear expression of a deliberate will.

We focused on a study about the careers of involvement of developers in big international free software projects, involving an online survey to better understand the big steps of the involvement cycle. From the first contact in so far as a simple user until the admission in the “hat” of core developers, without forgetting the intermediary steps for example first post to developers-mailing list or first submission of code. We tried to identify some profiles of involvement, and so as to explain different significative logics for those profiles, we associated with those quantitative results a few long biographical interviews, using a special methodology for biographical analysis which underlies the *discontinuities* and the broken lines in the life courses.

## ***A longitudinal study of Open Source developers:***

A general weakness of the inquiries on Free/Open Source Software projects is the excessive focalisation on the electronic *logs* and the lack of interest for the articulation between the involvement in the project and the context of the *rest of their life* : how is software involvement impacted by the evolution of the professional activity, or the transformation of the private life (matrimonial project, birth of a child)?

Because the extraction of statistical regularities, either from mailing-list activity (Mockus Herbsleb 2002) or from CVS logs (Robles 2006), seems easier, social sciences analyse these projects display with a hemiplegics view of the developers activity. Indeed those logs-obsessed inquiries have provided some great results indeed, about the ecology of knowledge creation in collaborative production of complex informational goods for example (Lanzara Monner). They even showed two series of results about the *individual involvement of developers*. They pointed out that a key result of those big projects is the interiorisation of humbleness norms (Lee Cole 2005), which finds a worthy expression in the “bias towards action rather than coordination” (Yamauchi et alii 2000). Participation to huge open source projects is also a mean to “overhear” a large range of up to date technical information (Gutwin Penner Schneider 2004) : consequently, the involvement in FOSS is explicitly used by individual developers as an efficient apprenticeship of project management. Log practices are also subtly interpreted so as to exhibit the formation of small social networks like teams or “cliques” in the big projects, so as showed in the “halo effect” hypothesis (Mockus Fielding Herbsleb 2002). The involvement in those software projects also has the means to create, even beyond reputation effects, *social capital*. But how are those apprenticeship and social capital wishes connected with the biographical course of each developer? What are the *involvement patterns*, on the mid or long term, of each individual developer? How does he articulate his life cycle with his benevolent status in FOSS?

The methodology that we used was grounded on a survey addressed to developers from two huge free software projects, Debian Devel and Open BSD, which we upload on a webpage hosted by our Engineering School Telecom Paris<sup>3</sup>. We broadcasted the

---

<sup>3</sup> See <http://socdev.enst.fr>

existence of our survey on the discussion list “Debian-devel”, and then obtained 68 answers from Debian maintainers (from a total of 700 maintainers, i.e. an answering rate of 10%<sup>4</sup>). We completed the survey with long biographical interviews of 14 elder<sup>5</sup> Debian maintainers, that is to say Debian members who succeeded in the admission process and had a regular-signed PGP key, and who had also uploaded their first package in the distribution for more than 3 years<sup>6</sup>. Most of those interviews took place in public events which hosted Debian conferences: 8 of them were held during the 2005 Debconf in Helsinki between July the 9<sup>th</sup> and the 17<sup>th</sup> of 2005, 4 of them were done during the 2005 “Rencontres mondiales du logiciel libre” in Dijon, France between July the 5<sup>th</sup> and the 8<sup>th</sup>. We also completed this survey with 10 interviews with OpenBSD developers, which is another more confidential Open Source distribution: 5 of them were held during the 2005 “What the hack!” meeting, between July the 28<sup>th</sup> and July the 31<sup>th</sup> in Netherlands.

Some general results of the survey are useful to frame our investigation. A first result is that most Debian elder maintainers are computing graduated. Free software involvement is articulated to the labor market of computing professionals. 72% of the developers who replied have a degree in computing : 39% have a BA, 50% a MS and 11% a PHD. A second result is that the activity in the Debian project is mostly take from the domestic life : 100% of the

---

<sup>4</sup> For the BSD maintainers, there were too few answers, so we did not use..

<sup>5</sup> Elder doesn't mean they are old people : the mean age of the Debian developers who replied to our study is 29 years.

<sup>6</sup> We know thanks to the statistical longitudinal tracks of benevolent careers in Open BSD showed by a Open BSD developer on his personal website that the mean longevity is about 5.5 years. Confirmation of that result is showed for Gnome and Open BSD in Robles (2005). Some samples on Debian developers bring us to the opinion that it could be generalized to Debian benevolent careers.

developers who replied use a Debian at home (generally the unstable version : 56%) when only 80% of them use a Debian at work (generally a stable one : 75%).

The average time spent in the project is 10,4 hours per week. The time spent on the user list to read posts or eventually write an answer is quite low: more than 64% of the developers spend less than 1 hour per week on the user-list; and for 71% of them the frequency of reading the user-list is less than one a week.

The specific goal of our study was to characterize the different significative logics of involvement that could explain the *bearing* for years in the project and the *assiduity* in the contribution. We also eliminated from our biographical interviews all the developers that gave up before having passed the threshold of 3 years in the core.

To identify a life career in Debian, we use two types of temporalities (Conninck Godard 1990). The first one is the temporality of *process* : each biographical involvement in Debian passes through the same steps, which means : first installation of a Debian distro, first post to the user list, first post to the “Debian-devel” list (it is not needed to be an agreed maintainer to post to the “Debian-devel” list), appliance, possible “turning points” in the contributive patterns, and then *exit from Debian*. The second one is the temporality of *sequences*. The sequences are the specific moments where bifurcations happen: they are external factors, like getting married, child birth, employment (end of the student period). This methodological splitting into two temporalities can give us precious results about the importance of the sequences to explain the involvement cycles in the benevolent project: are there differences in terms of involvement in Debian between the Elder developers who have begun their involvement in Debian when have graduated and those who have begun when already gainfully employed ?

## **Two kinds of bibliographical trajectories.**

Motivations emerge in function of action, and not according to statement of intent. In fact these intentions are difficult to analyse. Motivations cannot be understood in a dynamic of evolution. Mobility and trajectories are a privileged means to size these practices. We cannot understand involvement, trajectories, or even regulation, without references to the context, in our case the professional context.

From the last studies in Open Source, we notice that most Open Source developers have an activity directly bound with IT. ( 89% according to the FLOSS study<sup>7</sup>, 83% according to BCG).

If we take into consideration the Lakhani and BCG study<sup>8</sup>, 68% of OS developers work on IT industry, ( 44% are coders, 7% are systems administrators, 7% are IT project leaders). We can add to these figure that 22% are students and that 7% are researchers, most of them in computer science. In conclusion only 17% have an activity not related to IT. These results are confirmed by our questionnaire, as 72% of Debian developers have a degree in computer science.

Taking into account their tie with a professional activity, we can clearly distinguish two kinds of situations, illustrating different trajectories in their OS developer “career”.

The problem is that the boundaries between professional and volunteer activities are not so clear, as 80% of the Debian developers interviewed use Debian at work, and that these developers consider that they do 65% of their code on Debian at home.

---

<sup>7</sup> *Free/Libre and Open Source Software: Survey and Study Floss* final report,  
<http://www.infonomics.nl/FLOSS/report/>

<sup>8</sup> *Boston Consulting Group Hacker Survey*,  
<http://downloads.planetmirror.com/pub/lca/2003/proceedings/papers/Hemos/Hemos.pdf>

However if we strictly consider the professional status, we perceive those who have a job directly attached to Open Source development, and those who develop Open Source purely as a volunteer and have a different professional activity.

The first group will have a monolithic involvement and the second group a dual involvement.

Both are present in our sample, and allow us to better understand the drives of this involvement, and the implication in every day life.

### **A dual involvement.**

If we take the results of questionnaires, half of the developers have a dual involvement in *Open Source*, considering that according to Lakhani & Wolf (2003), 40 % of Open Source developers are paid for their participation on Open Source, and according to our questionnaire 57% consider Debian as a professional activity. Moreover, we can consider that more developers have this dual involvement, indeed even if they are paid or if they consider Debian as a professional activity, it doesn't mean that they have a full job, in Open Source. IT professional are often driven to use and work on Open Source, even if it is not their main activity.

What is important, in this dual involvement, is that it allows us to perceive how they manage to bind professional activity and volunteer involvement, in other words how they manage a double life. Traditionally there is a strict separation between occupation and leisure, by working time, for example in an office and spare time at home. For Open Source developers, the separation is not that easy, thus several developers are used to develop Open Source, during their working time, when they have free time. This kind of activity is well known in the factories as « homers » or « government jobs » in the US, « fiddling » or « pilfering »

in the UK and «perruque» in France.<sup>9</sup> This behaviour is mostly present, for occupations which can afford some free time; it is especially the case for system administrators.

In daily life, developers try to frame this border by a kind of strictness. As the case for this particular Debian developer who used to work on Debian during his daily travel between home and work( 3 hours), whereas at work he administrates Windows servers.

*« it's very difficult to find limit. If in the morning I have not finished, it's very easy to continue when I arrive. I have this asset, to have a job which allows me this flexibility, with a kind of tolerance, so when I need it, the Internet connexion is there. It is not a problem. But in a way, we have to limit this. We are not paid to work on Debian, it's clear »* French Debian developer, 44 years old, three children, dual involvement , (in author's translation).

This dual involvement is not easy for developers to take on in relation towards the employer. Indeed the threat to be fired always exists, and there are some legal considerations. In practice the employer can be the exclusive owner of all the production done during the working time, and even beyond.

*« I've done an internship at IBM, where it was not possible to contribute during six month because, I didn't want to affect the project (OpenBSD), with intellectual property problems. Because IBM make you sign something totally wacky, if I had invented an Ice-cream recipe, I think it would be theirs »* French OpenBSD developer, 28 years old, dual involvement, (in author's translation).

Consequently more and more developers want to have the legal control on their

---

<sup>9</sup> Michel Anteby « La perruque en usine : approche d'une pratique marginale, illégale et fuyante » Sociologie du travail . n°4, vol.45, octobre-décembre 2003

activity, so try to be very accurate when they sign employment contracts. They want to be sure that their Open Source activity could not be siphoned off by their employer.

This dual involvement and dual life is present in a professional context, but also in a familial one. Indeed it is not unusual to see an Open Source developer fixing a Windows personal computer, for their child or a member of their extended family.

### **The monolithic involvement. Three ways to render one involvement unified.**

A monolithic involvement means that people try to make coherence between their major spheres; employment, social life and family. In this context the two last spheres are not subordinated by the first one, but individuals act to change an unsatisfactory job. In this way occupation and leisure are part of the same existential goal. Occupation is an instrument for fulfilment and professional choices depend on the other spheres. However the professional sphere is not as flexible as the voluntary sphere. It responds to some structures which are more often imposed on the employed. Moreover, occupation is generally, a question of subsistence which leads to a kind of inertia, where every big change has a significant impact, notably in terms of earnings, and so in terms of well being. In this way, the dualistic involvement is more convenient.

Based on our qualitative study, we can bring out three ways that illustrate this kind of trajectory; these trajectories are tied with professional situation but also professional expectancies. It is also interesting to note that the involvement could change according to the generational belonging.

#### **To create ones own business.**

Since 1995, in France we have seen the growth of small business, specialized in

services bounded with Open Source Software, for example nowadays one estimate that there are 70<sup>10</sup> businesses of that kind. This new job market is seen as an opportunity to many Open Source volunteers, but most of these companies are indeed created by former developers of the main Open Source communities. For example, Ian Murdock, the Debian founder has also created Progeny, a Linux platform company. In France Christophe Le Bars, the first French Debian developer has created the first company of this kind, Alcôve. Recently Mark Shuttleworth, a South African millionaire and former Debian developer, created a Debian derivative distribution, Ubuntu, which is gaining in success.

The choice to create a company is feed by a reaction with a current professional context, as it is the case of Willem an OpenBSD developer, who after a qualifying in computer science engineering and a long experience in the Dutch telecom operator, has created his own company specializing in Open Source and particularly on Open BSD.

*« I think I was unhappy in my current job back then because we didn't use Open Source enough and I could see there were better alternatives available that I wanted to work with. So, it was a real motivator for me. It helped me cross the line to start my own company because that can be scary as well. »* Dutch OpenBSD developers 34 years old, Monolithic involvement.

People, who create this kind of company, emphasise that in a way in doing so they regain the liberty they found in their Open Source involvement. Nevertheless they all accept that their conditions are different from what they expected.

Besides, this kind of reaction is correlated with generational elements. According to our questionnaire, the average age of those

---

<sup>10</sup>

Le monde informatique, du 15/04/2006.

developers who works in their own company is 26, compared with the public sector; 31 and the other private sector; 30.5. We will see that this element is also correlated with the fact that the student period is a determinant one.

Furthermore, to have ones own company in open source, allows more time to be dedicating to Open Source development, thus they dedicate on average 16.8 hours a week to Debian, it is 10 hours for the whole community. Thus the creating a company is seen as a fulfilment of their passion.

### **To influence a company from inside.**

When a developer is already integrated into a company, one can react in attempting to include Open Source within the company's strategies. The initiative to integrate Open Source in the company's priorities does not come from the hierarchy but comes from the bottom, typically the engineers. For instance this Debian developer, working in a major French telecom provider, has managed to create his own position as Open Source coordinator.

*« In fact, the solution [Open Source], has been setting by itself, this solution could be generalised and used by everybody. So it became the benchmark solution. In addition I have changed my position at this moment (...), now my position is dedicated to configuration management on Open Source, and other activities related with Open Source. »* . French Debian developer 42 years old, two children, Monolithic involvement. (In author's translation).

Others are solicited because of their Open Source technical skills, gained in their Open Source project involvement. This is the case described by Lerner and Tirole to explain motivations, by professional interest, and Open Source participation as a professional signal of competences. For instance, Marius, a German Open BSD developer has been

hired by a security consultant after a degree in computer science, and now his position is dedicated to Open Source. Notwithstanding he notices that, even if he works exclusively on Open Source, there is a huge difference between working on Open BSD and his position in this company.

*« It was really different from Open BSD where people were engaged and they really are doing stuff. They are behind it, yes? But then at work they didn't care about anything. It has changed at the company now but some of the people there they were really ... You are trying to focus and work and they really don't care... Not all, but some they didn't care and it's not really fun. »* German Debian developer, 32 years old, one child, monolithic involvement.

### **To find or keep a professional activity in the public sector.**

According to our questionnaire 36% of Debian developers work in the public sector, that means that there is a surrepresentation compared with the whole population.

Within public sector, we can note as Lackani has done in his previous study, that most of them are researchers<sup>11</sup>. This can be explained by the proximity between both activities, as Himanen<sup>12</sup> has presented, the free software production is very close from the academic world.

For instance, the production of Open Source software, the mode of review among peers, or the mode the reputation is created.

Another way to act in the public sector is more political, in its institutional sense; that is the case of involvement in politics promoting Open Source, in local or national administrations. This kind of lobbying is supported by associations, such as April and

---

<sup>11</sup> Op. cit.

<sup>12</sup> Himanen, P., 2001, *The Hacker Ethic and the Spirit of the information age*, Secker and Warburg, London.

Aful in France, or the Free Software Foundation on an international scale. Most of them attempt to influence countries policies, by using typical lobbying. For a minority of them they can integrate political parties, for example as a specific adviser in technology policies.

Another monolithic involvement combines the two last points, that means a professional involvement and on the other hand a political one. This is typically the case of Michel, who has an engineering qualification; he discovered the Open Source when he was working for a public bio-medical research institute. Then he became one of the first Debian developers in France. Now he says, when he has to change a job, that the possibility to develop Open Source during his working time is determinant. He exclusively chooses an occupation where he can develop or work on Open Source.

*« For ten years, I have been trying to choose my position where there is only Open Source, if there were a company, in which I had to programme on Java with Oracle, actually I would be interested. (...), so yes I take it into accounts when I choose a job. »* French Debian developer, 54 years old, 2 children, monolithic involvement. (In author's translation).

This involvement lines in a political militancy, which came from his student period.

*«I was a left wing militant when I was at the uni, I have participated in the student movement against Devaquet laws, so yes my involvement doesn't come from nowhere. »*

Currently he pursues his involvement as project manager in the French Ican association, developing Open Source software for developing countries.

### ***Involvement trajectories.***

### **First steps into the project.**

The integration into an Open Source project involves a series of different steps. From the first use to the official integration, a long course has to be done, with different significant stages. With our questionnaire and interviews on Debian, we managed to isolate accurately these stages.

Primary the first use of the distribution indicates a relative interest for open source and particularly this distribution, and it can be at work or on their home computer.

Secondly the first installation, in most of the cases the first use and the first installation are simultaneous, according to our questionnaire on average only one and a half months separate these two steps. The great majority of users, only do this first step and don't participate more to the project, they can be describe as "free-riders", because they take advantage of the software without participating.

The third is the inscription on the User-list. The User-list is a mailing list on the use of Debian The subscription to the user list indicates an interest on the distribution, and on the evolution of this. On user-list, one can ask questions on the utilisation of the software one can also answer to the user requests.

The fourth is the subscription on Devel-list. Devel-list is the central tool of coordination for all the developers. On this list, developers are used to ask questions about the resolution of bugs, on technical requests or even on political questions about the project. Subscription on this mailing list involved an interest on the project, and also some technical skills. Developers and users who want to become official developer in the great majority subscribe on this mailing list.

The last step is the official acceptance. It is the result of an acceptance process, which was informal at the beginning of Debian,



when one just has to send an e-mail to the project leader and one received an account on the Debian CVS<sup>13</sup>. Now because of the increasing number of pretenders, the community have created a process, ( New Maintainer Process), which involve the acceptance by an official Debian developer as a tutor, and technical and philosophical tests. The latter is to assure that the new enterers agree with the Debian philosophy and mainly its charter ( Debian Policy Manual).

### Swiftness of courses.

The major result of this approach is that swiftness of this course, is quite constant, that means that if people are quite tardy on the first steps, there is a high probability that they will also be on the other steps. Developers who take more that a year to be accepted have a gap of 16 months between the subscription on Debian User-list and subscription on Devel-list. On the contrary, people who take less than six months to be integrated have a gap of two and a half months between these two inscriptions. It reveals that there are different modes of involvement, in terms of time spent, and in terms of progress but that the participation requires a constant involvement whatever the level.

The swiftness of this integration into the project is crucial to the rest of the “career”. It determines the future level of involvement, and also nature of this involvement.

Thus in this table we can correlate the period between the first contact and the acceptance, and the weekly time these people spend on Debian when they are Official Developers.

Time between first contact and official acceptance.	Average weekly hours spent on Debian.
Two weeks to one month.	6.4
Two months to three months.	7.6
Three to six months.	5.6
Six months to one year.	26.3
One year or more.	14.3

To have important and long-lasting involvement, the first steps into the project have to be progressive, only those who had taken more than six month between the first contact and the acceptance have spent a consequent amount of time.

This is a consequence of two essential aspects on Open Source.

-Open Source involvement, in elitist projects such as Debian or Open BSD, involves a significant level of competences. To gain these skills, a learning period is essential, and this learning period is typically portrayed by the first steps in the project. These projects are well known as meritocracy, developers are evaluated on the work they do.

-The second aspect deals with the socialisation into the project, software development is a technical issue as well as a social one<sup>14</sup>. Also in both projects we have studied, socialisation has a central importance. We can realise that they use mailing-list and IRC a lot, also they are used to meeting each other, in international conferences organized specifically to the project. Thus every year nearly the whole Open BSD community (around 70 participants) assemble at the project leader house in Calgary in Canada, they call this meeting the *Hackathon*. Concerning Debian, every year a similar event takes place, gathering 300 out of 1000 official developers. This sociability is central for

<sup>13</sup> Concurrent Versions System. For a definition see [http://en.wikipedia.org/wiki/Concurrent\\_Versions\\_System](http://en.wikipedia.org/wiki/Concurrent_Versions_System)

<sup>14</sup> Philip Kraft and Jacob Nørbjerg "Software Practice is Social Practice". In Yvonne Dittrich, Christiane Floyd, Ralf Klischewski (eds.), *Social Thinking - Software Practice*. MIT Press (Cambridge, MA: 2002).

long term involvement, and to feed the motivation to participate. As for the skills, one can gain sociability within the project through a long term initiation. As the Defconf supervisor indicate these meeting reinforce motivation.

« *The initial aim was only to meet. "Let's come together". This goal that I have and I also try to give to other people, and I think that other people ought to understand that, that's based on what I think is important and what happens and what I think about group dynamics.* » Swedish Debian supervisor of the Debconf. 35 years old. Monolithic involvement.

Also one can suppose that the privileged bound and interlocutor within the community would be the tutor, but according to our questionnaire it is not the case, 74% of Debian developers consider that they have the same relations with their former tutor as with the other developers. Developers have privileged links with their co-maintainer developers; that is to say with the developer they are used to working with, or when they gain responsibilities with other developers who are on the same committee.

### **Steps which leads to a monolithic involvement.**

From the last section, we have learned that developers who participate the most, have spent more time during the initial period working on Debian. This initial period is determinant in the constitution of a monolithic involvement.

From this point we can consider that there is a contradiction between this and the fact that a large part of developers who are concerned with a monolithic involvement are young, for instance those who created a business.

We can develop several elements arguing that it is not a contradiction.

First, the tendency of developers to angle their involvement toward a monolithic one,

is quite new, it is in part because the creation of the Open Source labour market depends on the diffusion of Open Source into society. As Michel Callon<sup>15</sup> suggests markets are socially constructed, and the Open Source Software market is a recent and significant example of this kind of social construction.

Secondly, among our interviews we noticed that the student period was a crucial one. Many developers have discovered Open Source during their student days. (14 out of 24 developers we met). During this period most of the developers have spare time, and are in a learning environment.

As a developer said, « *there is an "everlasting student" effect, because you are always learning, doing test, failing, doing it again.* », French OpenBSD developer. 30 years old, one child. Dual involvement. In other's translation.

In several cases the involvement into Open Source is directly a result from a student project; it is specially the case of the project Video Lane<sup>16</sup>. The project was created by several students, from a French engineering school (Ecole Centrale de Paris).

« *It was a 120 hours-credit project for students, during this year they have to make a prototype. But the success was so important within the student hall of residence and even outside, that we lobbied within the school, to be allowed to diffuse it in OpenSource.* » French Debian developer, 27 years old, monolithic involvement, (in author's translation).

As this project was also included within Debian, there is a core team of Video Lane developers from this school who are still into the Debian project.

---

<sup>15</sup> Callon, M. 1998. (Eds), *The Laws of the Market*, Blackwell Publishers, Oxford

<sup>16</sup> Basset, T, "Coordination and Social Structures in an Open Source Project: VideoLAN" in *Free/Open Source Software Development*, edited by Stefan Koch, published by Idea Group, Inc, July 2004, pp.125-151

Also concerning people who are still students, when they talk about their future most of them see the fact that working as a professional Open Source developer as the ultimate achievement. The monolithic involvement allows the matching of existential and professional desires.

«For job considerations, clearly, this will take precedent on the others. I know that there are several developers who has got a activity close from OpenBSD, it allows to unify the both, to work during the day on things that can provide benefit to OpenBSD, it is additional time to the project. So I was thinking about creating my own company, but it will be hard to realise. »

French OpenBSD developer, 24 years old, monolithic involvement, (in author's translation).

## Conclusion.

In this qualitative analysis of the developer involvement we combine professional context and technical courses within the projects. We can suppose that in the future, there will be a tendency to the monolithic involvement, with in parallel the creation of a real job market. Antagonisms, presents in the different nature of involvement, don't participate to a dissipation of professional identity, but on the contrary participate to its structuring. Thus the repetition of convergent trajectories, leads to the creation of a new professional identity which integrate the former critics.<sup>17</sup>

## Bibliography:

Auray, N.

---

<sup>17</sup> Renaud Sainseaulieu, stress the role of the conflict in the access to the professional identity. « If there are collective identities, it is because individuals share a same actor logic in the access to the professional identity » p 303. Sainseaulieu, R ( 1988), *L'identité au travail*. Paris FNSP.

2004, "La régulation de la connaissance : arbitrage sur la taille et gestion aux frontières dans la communauté Debian", *Revue d'économie politique*, n°113, pp.160-182, Paris.

Basset, T,  
2004."Coordination and Social Structures in an Open Source Project: VideoLAN" in *Free/Open Source Software Development*, edited by Stefan Koch, published by Idea Group, Inc, pp.125-151

Becker, H.  
1966 *Outsiders: Studies in the Sociology of Deviance*, Free Press/MacMillan.

Callon, M (Eds).  
1998. *The Laws of the Market*, Blackwell Publishers, Oxford

Conninck, F .Godard,F.  
1990 « L'approche biographique à l'épreuve de l'interprétation. Les formes temporelles de la causalité », *Revue française de sociologie* 31-1, pp.322-337.

Gutwin,C, Penner,R, Schneider,K  
2004 "Group Awareness in Distributed Software Development", CSCW.

Himanen, P.  
2001 *The Hacker Ethic and the Spirit of the information age*, Secker and Warburg, London.

Hughes, E.C.  
1937, « Institutional Office and the Person », *American Journal of sociology*, vol. XLIII, p.408-410.

Kraft, P ,Nørbjerg,J.  
2002, "Software Practice is Social Practice". In Yvonne Dittrich, Christiane Floyd, Ralf Klischewski (eds.). *Social Thinking - Software Practice*. MIT Press (Cambridge, MA)

Lakhani, K.  
2003, "Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects", MIT Sloan School of Management, Working Paper 4425-03.

- Lakhani K, Wolf R.  
2005, "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects" in *Perspectives on Free and Open Source Software* edited by J. Feller, B. Fitzgerald, S. Hissam, and K. R. Lakhani (MIT Press)
- Lanzara, G.F Morner,M.  
2003, "The Knowledge ecology of open-source software projects", Conference at the London School of Economics.
- Lee, G.K., Cole, R.E.,  
2003, "The linux kernel development as a model of knowledge creation", *Organization Science*, 14/6 : 633-649.
- Mockus,A, Fielding,RT Herbsleb, J.  
2002, "Two Cases Studies of Open Source Software Development : Apache and Mozilla", Proceedings of ACM to SEM, 11, 3; pp.309-346.
- Robles, G.  
2006, *Empirical Engineering Software Research on Libre Software : Data sources, Methodologies and Results*, PhD Thesis.
- Lerner, J., Tirole, J.,  
2000, "The simple Economics of Open Source", *Working Paper Harvard Business School*.
- Torvalds, L., Diamond, D.,  
2002, *Just for fun : the story of an Accidental Revolutionary*, paperback.
- Tyler, J., Wilkinson, D., Huberman, B.,  
2004, "Email as Spectroscopy : Automated Discovery of Community structure within Organizations", *Hewlett Packard Working Paper*.
- Vicente, M.,  
2005, "Motivations à l'engagement dans le développement de logiciel libre. Les cas de Debian et Open BSD", UTC (Compiègne University of Technology), Master Thesis.
- Von Krogh,G von Hippel, E,  
2003, "Special Issue on Open Source Software Development", *Research Policy*, Volume 32, Issue 7, pp.1149-1157.
- Yamauchi,Y Yokozawa,M Shinohara,T Ishida,T  
2000, "How Cooperation opensource succeeds", *ACM Conference on Computer Supported Cooperative Work (CSCW2000)*, Philadelphia, PA.
- Weiss, D.  
2005, "Quantitative analysis of Open Source projects on Source Forge", *Working Paper*, Poznan University.