

## ***Compétences dans la production d'un bien commun numérique: Le cas du recrutement des développeurs de logiciel libre.***

**Michaël Vicente**

Université de Technologie de Compiègne, COSTECH  
Centre Pierre Guillaumat  
60200 Compiègne  
[michael.vicente@utc.fr](mailto:michael.vicente@utc.fr)

Si le logiciel libre participe à la mise en commun de biens informationnels, il participe également à la mise en commun de compétences. À partir du constat selon lequel un certain nombre de développeurs sont rémunérés pour participer au logiciel libre, nous examinons les modalités de recrutement par des entreprises de ces développeurs. Nous pourrions ainsi discuter l'hypothèse selon laquelle le logiciel libre constitue un dispositif de marché et les communautés un vivier de compétences.

Logiciel libre, recrutement, professionnels de l'informatique

Free software involves the collectivisation of informational goods and a collectivisation of competences too. Since some free software developers are hired to participate to those software, we will examine their terms of recruiting by IT companies. We will discuss the hypothesis that Free Software can be a market device and the Free Software communities a talent pool.

Free Software, recruiting, IT professionals

## ***Compétences dans la production d'un bien commun numérique: Le cas du recrutement des développeurs de logiciel libre.***

Le logiciel libre par ses caractéristiques et sa diffusion est devenu l'archétype du bien commun numérique. Fruit de la coopération volontaire de communautés de développeurs, il est librement distribué et les contributions de chacun sont obligatoirement partagées. Tel est le contrat qu'impose le logiciel libre.<sup>1</sup> Ce mouvement né au milieu des années 1980 avec la création par Richard Stallman des licences GPL<sup>2</sup>, a depuis le début des années 2000 connu un succès qui l'a fait sortir de sa confidentialité. Aujourd'hui ces logiciels sont utilisés massivement et se retrouvent au centre si ce n'est de l'activité informatique, au moins de l'activité qui tourne autour d'Internet. Cette adoption massive a inévitablement entraîné une institutionnalisation; et cette transition s'illustre notamment par l'emploi de développeurs rémunérés pour participer au développement de ces logiciels libres. Ces développeurs qui normalement participent bénévolement peuvent ainsi être employés par des entreprises qui ont un intérêt dans le développement de certains logiciels. Comme le révélait un article du numéro inaugural de cette revue: le logiciel libre se situe certes dans le monde de la gratuité, mais il est également à la frontière de mondes antagonistes: celui du travail bénévole et celui de l'entreprise. [Demazière, Horn et Zune, 2008]

On comprend alors que cette combinaison et cette improbable collaboration nécessitent des ajustements de parts et d'autres. Du côté des développeurs, ils réalisent une "trajectoire de professionnalisation", depuis une activité bénévole vers une activité rémunérée. D'autre part, et c'est ce qui va nous intéresser ici, ceci implique, pour l'entreprise qui recrute, un certain nombre de d'ajustements. Ces ajustements concernent à la fois l'organisation de l'entreprise mais également sa manière de recruter. En effet, depuis le début des années 2000 un certain nombre d'études ont pu mettre en avant la présence des développeurs rémunérés par les entreprises, et

---

<sup>1</sup> Les licences de logiciels libres impliquent quatre libertés: La liberté d'exécuter le programme, pour tous les usages (liberté 0). La liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins (liberté 1). Pour ceci l'accès au code source est une condition requise. La liberté de redistribuer des copies, donc d'aider son voisin (liberté 2). La liberté d'améliorer le programme et de publier ses améliorations, pour en faire profiter toute la communauté (liberté 3). Pour ceci l'accès au code source est une condition requise.

<sup>2</sup> General Public License

ceci même dans les projets qui dans leur discours et leur positionnement sont des plus hostiles au modèle capitaliste. (Voir [Robles, Duenas, et Gonzalez-Barahona, 2006] pour la communauté Debian.)

## **1. Production de biens numériques communs et présence des entreprises.**

Si la construction d'un logiciel libre repose sur une collectivisation du code produit par les développeurs, la construction d'un tel bien commun correspond également à une sorte de collectivisation de connaissances et de compétences. C'est ce même processus que l'on retrouve dans d'autres communautés en ligne telles que Wikipedia ou dans un certain nombre de forums. Cependant, à partir du moment où dont les entreprises sont présentes au sein des ces communautés, cette collectivisation de connaissance et de compétences semble prendre un autre sens. Cette présence sonnerait le glas d'une expérience alternative réussie dans le monde numérique. Les entreprises en utilisant et développant du logiciel libre, piocheraient à la fois dans l'offre de logicielle librement disponible, mais également en recrutant un ensemble de développeurs ayant développés et même créés un ensemble de compétences nouvelles au sein de ces communautés, seraient les principales bénéficiaires de cette collectivisation. Dans ce sens, une hypothèse formulée par un certain nombre d'économistes [Lerner et Tirole, 2002] tend à avancer que l'engagement des volontaires dans le logiciel libre est avant tout motivé par des visées carriéristes (*career concerns*) et que cet engagement a pour objectif de signaler les compétences aux entreprises en vu d'un recrutement. (*signaling skills*). Les développeurs auraient donc intériorisé cet état de fait. Les entreprises sélectionneraient ces individus dans ces communautés, en souhaitant bénéficier des compétences techniques mais également acheter un portefeuille relationnel, comme cela peut être le cas dans la finance [Godechot, 2006]. Les relations dont il s'agit ici étant les liens que le développeur entretient avec la communauté. L'entreprise bénéficie ainsi de ce que les économistes appellent externalités positives.

Du point de vue du recrutement, pour l'entreprise et plus particulièrement la firme informatique, le logiciel libre viendrait s'ajouter aux différents dispositifs de recrutement qui existent déjà pour

le marché de l'informatique. Ces dispositifs classiques, qui vont de la petite annonce, au chasseur de tête, en passant par les sites Internet de recrutement ont en commun de reposer sur une codification des compétences [Eymard Duvernay et Marchal, 2000]. Un certain nombre de travaux ont pu mettre en avant que l'évolution de ces dispositifs de recrutement vers le numérique allait dans le sens d'une codification accrue des compétences [Marchal, Mellet et Rieucan, 2007]. Or, une particularité sous-jacente à l'adoption de licences libres est que l'activité de conception de logiciel libre laisse des traces, à la fois dans le code lui-même, où l'ouverture du code permet de voir "qui fait quoi", mais également dans les relations de conseil présentes dans les listes de discussions. L'activité des développeurs et encore plus, la compétence des développeurs qui est tacite par définition est ici, si ce n'est directement explicitée, du moins explicitable.

Dans ce contexte, on peut alors légitimement se demander si le logiciel libre ne constitue pas un tel dispositif de recrutement pour le marché et si les communautés du logiciel libre constituent un vivier de compétences pour ces entreprises. Cela implique de s'interroger sur la nature des ajustements singuliers à l'oeuvre, tant pour le mode de recrutement que dans la définition des compétences. Il nous semble alors que l'analyse des moments du recrutement pris du point de vue du recruteur, mais également du recruté, peuvent être des indicateurs de la pertinence d'une telle hypothèse sur la nature du dispositif qui se présente à nous.

## **1.2. Méthode**

Cet article issu de notre travail de terrain mené entre 2006 et 2009 repose sur l'exploitation de 61 entretiens biographiques avec des développeurs de logiciel libre qui sont à un moment de leur parcours devenus des "professionnels du logiciel libre". Nous entendons ici par ce terme, des personnes qui d'une manière ou d'une autre ont été rémunérés pour leur participation au logiciel libre. Systématiquement nous nous sommes intéressés aux entreprises dans lesquelles ces développeurs travaillaient, et avons tenté de mettre à jour les conditions et modalités de leur recrutement.

Plus particulièrement, nous avons porté notre attention sur certaines entreprises qui sont totalement orientées vers le logiciel libre, et parmi celles-ci Canonical, qui supporte le logiciel Ubuntu. Ce cas nous semble être particulièrement pertinent puisque, d'une part, Ubuntu a connu

un incroyable succès auprès du public (il s'agit de la distribution logicielle la plus téléchargée<sup>3</sup>) et parce que, d'autre part, nous avons pu participer, lors de nos recherches, à un certain nombre de réunion concernant cette communauté et nous avons également pu réaliser des entretiens avec douze développeurs salariés de cette entreprise.

### **1.3. Le recrutement dans une entreprise Open Source: le cas de Canonical**

Le logiciel Ubuntu a été initié début 2004 par un ancien développeur Debian, Mark Shuttleworth, qui a fait fortune en vendant une *Dotcom*<sup>4</sup> juste avant l'éclatement de la bulle Internet. Cette importante rentrée d'argent lui a permis de mener une vie de rentier durant laquelle il se lance plusieurs défis personnels, il a ainsi été le second touriste de l'espace en 2002 et, l'année suivante, il fait partie d'une expédition en Antarctique, durant laquelle sera initié le projet Ubuntu :

"L'histoire, telle que Mark la raconte, c'est que je ne sais pour quelle raison, il s'est retrouvé dans l'Antarctique, et il avait avec lui six mois de *mailing lists* Debian, Et vu que, sur ce bateau, il n'y avait apparemment pas grand chose d'autre à faire, il s'est assis devant son ordinateur et a lu les archives ". ENT9

---

<sup>3</sup> Sur le site <http://distrowatch.com>, qui recense les téléchargements, on note que la distribution Ubuntu est en tête de classement depuis 2005.

<sup>4</sup> Il s'agissait d'une entreprise de sécurité informatique qui a été revendue à la société VeriSign.

Le principal problème du projet Debian à cette période est le manque de convivialité du logiciel. Mark Shuttleworth pense alors qu'il est nécessaire de créer un autre projet plus convivial, utilisable par l'utilisateur "Lambda". C'est ainsi que le projet Ubuntu voit le jour, avec pour slogan "Linux for Human Being ". Le projet est, dès le début, soutenu par l'entreprise Canonical, elle-même financée par la Fondation Shuttleworth. La viabilité économique escomptée du projet repose sur un modèle basé sur la vente de services d'assistance et de personnalisation en direction des entreprises. Après que Mark Shuttleworth en a eu l'idée, le projet Ubuntu se réalise rapidement, puisque les premières réunions de travail de l'équipe de développeurs ont eu lieu en mai-juin 2004, lors de la conférence Debian à Porto Allegre (au Brésil). D'un point de vue technique, le projet Ubuntu est une dérivation du projet Debian, c'est-à-dire qu'il est construit à partir du code initial de Debian<sup>5</sup>. L'entreprise Canonical se décrit comme une entreprise "distribuée". En effet, comptant à peine plus de 100 salariés, les développeurs employés par l'entreprise sont répartis dans 27 pays différents.

Sur le modèle de l'*open source*, les employés participent et se coordonnent d'une manière distribuée, ils utilisent un CVS<sup>6</sup>, des listes de discussion et la messagerie instantanée. L'entreprise organise également des rencontres physiques avec toute l'équipe, environ tous les trois mois.

Aussi, pour assurer cette coordination, les emplois au sein de l'entreprise ne sont pas uniquement techniques ou administratifs, un certain nombre de personnes sont également chargées d'animer la communauté, ce sont les "*Community Manager*".

---

<sup>5</sup> Il ne s'agit cependant pas d'un "fork " c'est-à-dire la construction d'un projet parallèle à partir du code disponible au moment de la rupture, mais d'une réelle dérivation car les implémentations et le code modifié sont par la suite retournés vers la communauté initiale. Des membres d'Ubuntu parlent ainsi de "spoon" (cuillère) par opposition au "fork" (fourche), dans le sens où il y a bien dérivation du projet, mais aussi un "retour" en direction du projet initial, ce qui fait penser à la courbure de la cuillère.

<sup>6</sup> *Control Versionning System*, c'est-à-dire un logiciel qui permet la coordination à distance dans le développement de code logiciel.

## **1.4. Comment s'articulent les liens entre entreprise et communauté?**

Dans son fonctionnement, l'entreprise interagit avec un ensemble de communautés, d'une part des communautés comme Debian dont le logiciel qu'ils portent est issu, et d'autre part la communauté Ubuntu qui est issu de la volonté de l'entreprise. On voit donc à l'œuvre un mode de fonctionnement original où les fonctionnements de la communauté et de l'entreprise tout en étant distincts sont tout de même dépendants et intimement liés.

En ce qui concerne le statut des développeurs, Ubuntu incarne en elle-même toute l'ambiguïté entre travail bénévole et travail salarié, puisqu'elle a la particularité de mêler des personnes salariées travaillant pour l'entreprise, et une plus vaste communauté de développeurs volontaires. Comme pour de nombreuses communautés, en suivant la métaphore de l'oignon<sup>7</sup>, on considère qu'il y a différents niveaux de participation et d'appartenance.

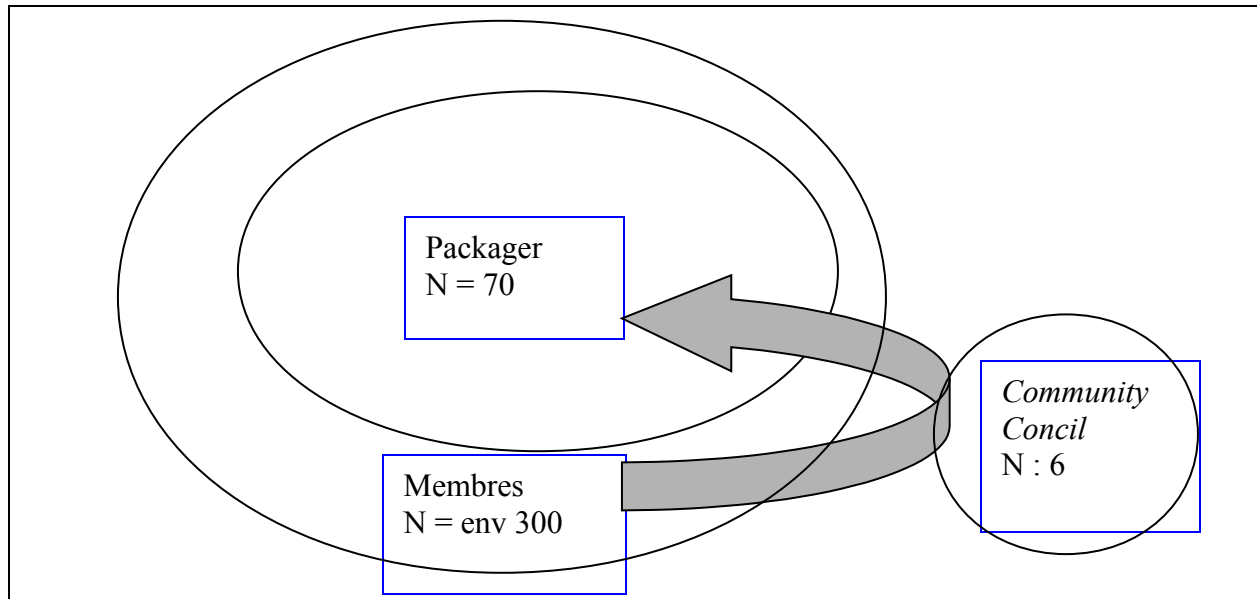
" Oui, en quelque sorte, au sein d'Ubuntu, il y a plusieurs niveaux: au premier rang, il y a les membres, ces personnes peuvent participer sans être packageur, mais au niveau au-dessus, il y a les *packagers*, ces personnes sont 70, et doivent être approuvées par le *Community Concil.*( ...) On est six, je suis dans le comité, mais je ne suis pas développeur. " ENT31

---

<sup>7</sup> Ce modèle a été développé pour la première fois dans [Crowston et Howison, 2003]

Nous pouvons alors résumer la forme de la communauté par le schéma ci-dessous:

### *Organisation de la communauté Ubuntu*



Le *Community Concil* joue un rôle prépondérant et est composé de développeurs appartenant à la communauté Ubuntu. Ainsi, parmi les développeurs élus en 2009, seul deux d'entre eux sont salariés de Canonical. L'élection se fait parmi les membres au sens large, c'est-à-dire qu'il n'est pas nécessaire de faire partie de l'équipe des *packagers* pour voter et être élu. Les *packagers* sont, quant à eux, soixante-dix et doivent être approuvés par le *Community Concil*. Ce sont eux qui ont accès au logiciel de contrôle de version. Parmi les membres du *Community Concil*, seule une minorité sont des développeurs employés par Canonical. De la même manière, être salarié de Canonical n'implique pas nécessairement de faire partie des *packagers* officiels. Le mode de gouvernance de la communauté est de l'entreprise sont donc étanches. Ceci d'une pour écarter les critiques potentielles de développeurs de la communauté et d'éviter un phénomène de démotivation généralisée plus connu sous le nom de *Crowding Out Effect*<sup>8</sup>. À ce risque de

<sup>8</sup> Le phénomène de *Crowding Out* est bien connu des économistes, il est explicite dans le cas de la régulation



démotivation due à la présence de personnes rémunérées, s'ajoute la potentielle fuite de développeurs craintifs que la communauté soient sous le contrôle d'une entreprise unique.

### **1.5. Le recrutement des salariés chez Canonical: une évolution du type de compétences**

Ce fonctionnement original implique également un mode de recrutement particulier, notamment dans son articulation entre communauté et entreprise. On observe qu'au cours du temps, l'entreprise, tout en gardant des relations privilégiées avec les développeurs de la communauté initiale, recrute des développeurs venant d'horizons différents.

Le projet étant issu de Debian, l'ensemble des technologies d'origine appartiennent à Debian. Il n'est donc pas étonnant que, dès le début, l'entreprise ait recruté un ensemble de développeurs issus de la communauté Debian,

"C'est vrai que la plupart des personnes que l'on embauche font partie de communautés telles qu'Ubuntu ou Debian. Mais après ces premières embauches, on a employé des personnes qui venaient d'ailleurs. Je pense que la première année 90% des personnes venaient de Debian". ENT44

Parmi ces recrues, on compte un ancien *Release Manager*<sup>9</sup> de Debian, mais aussi des développeurs qui ont des positions moins centrales. Initialement, tous les développeurs étaient directement recrutés par Mark Shuttleworth, puis le recrutement a été effectué par les développeurs eux-mêmes, élargissant ainsi l'éventail des communautés au sein desquelles sont recrutés les développeurs. Steven<sup>10</sup>, ancien *Release Manager* de Debian, a ainsi été directement

---

monétaire ; par exemple, une augmentation du taux d'intérêt amoindrit (*crowd-out*) l'investissement privé. Placé dans un contexte d'échange hors marché, le "*Crowding Out Effect*" repose sur l'idée contre intuitive selon laquelle la compensation financière, au lieu d'inciter à la participation, au contraire, la diminuerait. Selon Frey et Jegen (2001), le "*Crowding Out Effect*" implique qu'une incitation externe, par exemple l'incitation monétaire ou la punition, diminue la hauteur de l'implication dans les domaines qui sont normalement régis par l'action volontaire.

<sup>9</sup> Personne centrale dans le projet assurant la finalisation de la distribution.

<sup>10</sup> Les noms ont été modifiés.

recruté par Mark Shuttleworth en tant qu'administrateur système des serveurs de la communauté Ubuntu.

"Il est revenu de son périple avec une liste de personnes qu'il voulait recruter. Et pour je ne sais quelle raison, j'étais sur cette liste ".ENT9

Il s'agit ici de noms issus des archives de la *mailing list* Debian, que Mark Shuttleworth avait longuement étudié lors de son voyage en Antarctique.

"Il m'a téléphoné et on a ensuite discuté pas mal de temps. Au début, je n'ai pas réalisé qu'il me téléphonait pour essayer de me recruter. On était juste en train de parler du projet Ubuntu, voir ce qu'il était possible de faire... enfin, à l'époque, ça ne s'appelait pas Ubuntu, on parlait simplement d'une possible version dérivée de Debian. Et à la fin de notre conversation, il m'a demandé : "de combien aurais-tu besoin ? " et donc je l'ai fait répéter, il m'a dit : "veux-tu un travail ? ", il m'a complètement surpris. Et donc, j'ai dit "OK", un job pour travailler sur de l'*open source*, ça n'arrive pas souvent, surtout lorsqu'il s'agit de travailler sur Debian." ENT9

En faisant dans un premier temps exclusivement appel à des développeurs Debian, Canonical souhaitait mettre en place une interface entre les deux communautés afin de pouvoir exploiter la distribution Debian, sans que cela soit source de problèmes. Puisque les développeurs ont une connaissance précise du projet à la fois dans ses dimensions technique et organisationnelle, l'entreprise, en les recrutant, sait qu'ils ont une aptitude technique, mais également qu'ils sont capables d'interagir avec la communauté. En effet, ces développeurs connaissent les autres développeurs de la communauté et ils en maîtrisent parfaitement les règles et l'histoire.

On assiste ici à une première période d'adaptation et de mise en place de routines et de règles en partie copiées sur la communauté Debian.

Dans un second temps, les développeurs proviennent d'autres communautés *open source*. On retrouve effectivement par la suite des développeurs venant de communautés fonctionnant sur un mode assez proche de celui d'Ubuntu. c'est par exemple le cas des développeurs *Mozilla* ou

*Gnome*. Les logiciels libres forment ainsi une sorte d'écologie où les projets sont interdépendants, l'ajout de composantes émanant d'autres communautés est constamment présent. Ainsi, le développeur *Mozilla* a été recruté parce qu'il participait activement à une composante de celui-ci (*Bugzilla*, un logiciel de gestion des *Bugs*).

Plus étonnant, dans un troisième temps, le recrutement s'est étendu aux communautés hors *open source*. C'est ce dont témoigne ici Martin, l'une des premières personnes recrutées par la communauté et qui a, par la suite, recruté d'autres développeurs.

"Et plus récemment, on recrute des personnes très compétentes, qui ne sont pas forcément impliquées dans l'*open source*. Personnellement, la personne que j'ai embauchée est très bonne, elle a de bonnes compétences, elle a une implication dans l'*open source*, mais qui n'est pas énorme. " ENT48

Cette dernière catégorie de développeurs, qui n'appartiennent pas formellement à une communauté *open source*, jouit cependant d'une reconnaissance de ses compétences dans d'autres types de communautés, que nous pouvons appeler au sens large les "communautés professionnelles".

C'est le cas de Tim, qui a longtemps travaillé en tant qu'indépendant sur les questions du WIFI et qui s'est forgé une bonne réputation dans son domaine. Il pense qu'il a été recruté par Canonical sur la base de cette réputation. Aussi, il reste à noter qu'une partie significative des recrutements effectués hors communauté concerne des postes dont les compétences ne sont pas forcément présentes au sein des communautés *open source*, à l'instar des postes liés au marketing ou aux fonctions commerciales. Ainsi, Julien, un Canadien de Montréal, a été recruté parce qu'il avait une connaissance du tissu économique de la région, et aussi parce qu'il avait une expérience dans le support à Montréal. Compétences et relations qu'il a développées lorsque qu'il travaillait dans une boutique de matériel informatique puis chez un fournisseur d'accès Internet québécois.

On voit donc que l'ensemble des compétences demandées n'est pas uniquement lié à une technologie propre au libre. Viennent en effet s'ajouter à ces compétences spécifiques, d'autres

compétences complémentaires, qui sont difficilement codifiables vues sous le seul spectre du développement technique du logiciel libre.

## **2. Les communautés du libre: un vivier de compétences pour les entreprises?**

À travers l'exemple de Canonical, nous avons pu mettre en avant que le recrutement des entreprises *open source* reposait sur une articulation entre entreprise et communauté. Cet exemple peut nous renvoyer l'image d'un "réservoir de compétences" dans lequel les entreprises viendraient piocher, selon leurs besoins spécifiques. En effet, on peut très bien représenter cette situation comme étant un véritable marché avec, d'un côté, des compétences et, de l'autre, des demandes spécifiques.

Nous détaillerons ici dans un premier temps les éléments objectifs qui peuvent faire penser que le cas des développeurs correspond bien à un "marché de compétences".

Dans un second temps, nous discuterons d'un point de vue théorique cette notion de "marché de compétences" pour lui préférer celle de "dispositif d'appariement", où la confiance et la réputation jouent un rôle central. Enfin, dans un troisième temps, nous verrons les limites empiriques qui s'imposent à l'hypothèse d'un marché de compétences, limites que nous identifierons ainsi :

- Il y a une persistance des intermédiaires classiques
- Les négociations ne sont pas que salariales
- Lorsque l'on parle de recrutement, il est nécessaire de faire une distinction entre le salarié et le contractuel

### **2.1 Un marché de compétences?**

Comme le font remarquer Arthur et Rousseau (1996) ou encore Cappelli (2001), le marché du

travail est très favorable aux informaticiens, et au niveau international se pose même la question de leur pénurie. La mobilité des développeurs est elle-même largement concernée par ce phénomène et l'utilisation exponentielle de l'*open source* fait que les développeurs de logiciel libre sont eux aussi trop peu nombreux par rapport au marché potentiel. Lors de notre travail de terrain en Californie mais également en France, nous avons pu faire le constat que les développeurs *open source* se retrouvent eux aussi dans une position privilégiée sur le marché du travail.

"Oui, je connais même des entreprises qui mendient pour trouver des personnes. L'utilisation de Linux a explosé. Mais le plus dur, c'est de trouver des gens qui ont une expérience professionnelle avec Linux."ENT41

Cette niche spécifique au logiciel libre fait que ces développeurs constituent une sorte de "marché d'excellence".

"Je vois de plus en plus de personnes travailler avec du logiciel libre, écrire du code en Python, en Perl, des gens qui créent des applications dans les ASV. Je pense que l'on va voir de plus en plus de personnes. C'est un marché du travail qui est boosté. Et l'aspect ironique, c'est que la qualité doit aussi évoluer, on doit trouver plus de bons développeurs, travailler avec de bons développeurs. Nous, par exemple chez Canonical, on demande l'excellence, ils doivent être très très bons. "

Cette forte demande sur le marché du travail se traduit également par des évolutions significatives de carrières avec, bien souvent, un salaire conséquent pour les collaborateurs des firmes.

"La plupart des personnes que je connais et qui se sont engagées dans l'*open source* en 2000 ont un bon boulot maintenant " "Si tu es un contributeur majeur aux États-unis, si tu n'es pas employé à un bon salaire, c'est parce que tu ne le veux pas". ENT45

Ce lien entre compétences et rémunération que nous identifions dans le libre, va ici dans le sens

de la thèse d'un "marché de compétences", où ces compétences spécifiques et rares seraient valorisées. Cette notion peut néanmoins être discutée.

## **2.2. Marché de compétences ou dispositif d'appariement?**

Les liens au sein des communautés du libre sont très lâches et ne reposent que sur la participation libre à un projet (c'est-à-dire la liberté de participer mais également la liberté de quitter), la confiance reste donc un élément essentiel au maintien dans la communauté. Il s'agit ici d'une confiance plus tacite qu'organisationnelle. Il n'y a pas de règles explicites ni de contrats formels, mais plutôt un ensemble de conventions et de règles implicites qui visent à maintenir les relations entre les développeurs, dans le but de mener à bien un projet commun. Ces règles concernent la réciprocité, la disponibilité et les différents rites de passage, présents dans le processus d'apprentissage. Dans la théorie des contrats, la confiance a pour rôle de combler les manquements résultant de l'incomplétude de ces contrats. Selon Williamson (1985), cette dernière est due à la singularité des actifs mis en jeu. Pour reprendre Lucien Karpik (2007), "les produits singuliers sont des entités incommensurables: ils sont caractérisés par des constellations de qualités et de dimensions, dont les significations sont inscrites dans leurs relations mutuelles"<sup>11</sup>. La plupart des produits singuliers (les grands vins, les conseils d'un avocat ou la psychanalyse) comportent une grande incertitude quant à leur signification, leur qualité et leur dimension. Il en est de même lorsque l'on fait appel à un développeur. Celui-ci peut promettre à une entreprise de résoudre certains problèmes techniques particuliers sans que le client soit a priori certain de l'identité et des compétences du développeur qui les résoudra et donc, par là même, s'ils seront réellement solutionnés ou non. La variable centrale de ces indéterminations est donc la compétence des acteurs, ici des développeurs. Selon Karpik, le seul moyen de lever ces indéterminations est de mettre en place des dispositifs de jugement et de confiance. Le logiciel libre, notamment grâce à sa transparence, fait office de dispositif à la fois de jugement et de confiance, et ceci de manière plus ou moins inductive. Dans ce dispositif de confiance, la transparence est ainsi aidée par le respect mutuel des normes éthiques autour du logiciel. Ce dispositif repose donc d'une certaine manière sur un modèle de "confiance réputation" (Kreps,

---

<sup>11</sup> [Karpik, 2007] p. 39

1990). La littérature en sciences économiques montre très clairement que cette confiance est centrale dans les relations contractuelles de travail, et a un effet économique certain. Akerlof et Yellen (1986) remettant en cause la vision *walrasienne* du marché du travail indiquent que le rapport salarial et le niveau des salaires ne dépendent pas seulement de la confrontation entre l'offre et la demande, mais qu'ils reposent sur un contrat implicite, les employés fournissant plus d'efforts et de loyauté que ce dont il était convenu dans leur contrat, en échange d'une rémunération supérieure à celle du marché. Ce contrat implicite est ainsi essentiellement basé sur la confiance entre les deux contractants et sur une réduction des incertitudes.

En effet, conformément à la théorie des coûts de transaction, ces contrats sont toujours sources d'incertitude. D'un côté, l'employeur ne peut pas être assuré de la loyauté et des compétences de son employé, et de l'autre, l'employé ne connaît pas quelles seront ses futures conditions de travail. Pour les entreprises qui participent au logiciel, et qui ont donc une prise avec les communautés soit par la participation de leur salarié soit par d'autres formes de soutien, le lien entre employé et employeur est établi. L'entreprise devient un acteur de la communauté et, via ses développeurs rémunérés, elle coopère avec les autres membres. Lorsque ces entreprises recrutent les développeurs appartenant à la communauté, le passage au statut d'employé/salarié est simplifié. On passe ainsi d'une coopération informelle à un type de coopération qui a pris une forme contractuelle, celle du contrat de travail.

Dans le cas des recrutements intra-communautaires, les incertitudes sont considérablement réduites. D'un côté, l'employeur connaît très précisément les compétences, l'effort et les capacités d'organisation de son futur employé, la transparence permise par l'*open source* offrant la possibilité de pouvoir observer avec exactitude le travail effectué par ces personnes. D'un autre côté, l'employé développeur de logiciel libre connaît très précisément le mode de fonctionnement de l'entreprise pour avoir coopéré avec elle de manière non-contractuelle au cours de son engagement dans le libre.

### **2.3. Confiance et réputation au moment du recrutement**

Au sein des communautés, le recrutement des développeurs et l'appréciation de leurs compétences ne se fait pas par l'examen précis de leur code, mais de manière plus intuitive, à

partir du "capital réputation" que les développeurs semblent avoir acquis. Ainsi, Julia, ancienne responsable de recrutement de développeurs *open source* au sein de l'entreprise Google, nous indique comment elle effectuait ce recrutement:

"L'*open source* est basé sur la réputation, donc lorsque l'on a une bonne réputation, cela signifie que l'on a de bonnes compétences. Et donc, c'est aussi compétitif, si tu écris quelque chose, tout le monde te connaît sur ce que tu fais, et si personne n'utilise ce que tu as fait parce que c'est mauvais ou inutile, tu n'auras jamais de réputation. "ENT47

Elle nous indique que, lorsqu'elle était chargée du recrutement, sa perception de la réputation se construisait de manière très intuitive et pratique, et pas simplement sur la base d'un CV.

"Ici, chez Google, on a un système d'entretiens particulier, tu as d'abord un entretien classique avec les RH, mais après, il y a encore un entretien sur ce que tu as réellement fait, qui est plus technique, basé sur le code, et là on voit la réputation, c'est très clair. "ENT47

On retrouve les mêmes considérations pour ce responsable informatique de Morgan Stanley, qui régulièrement est amené à recruter des développeurs *open source*.

"Embaucher des gens, c'est compliqué et coûteux. Les talents sont très difficiles à attirer et à garder. Donc, nous employons beaucoup de développeurs *open source* qui sont très connus dans leurs domaines, et on les paye pour qu'ils participent aussi sur leurs hobbies. Par exemple, la personne de Linux Window Manager, il a travaillé à Tokyo avec nous, il est très très brillant, mais on ne l'aurait jamais connu s'il ne travaillait pas dans le monde *open source*. "ENT49

Ce recruteur vérifie la réputation des développeurs en se basant sur leur visibilité au sein des communautés, principalement en observant les *mailing lists*.

"Non, mais ça n'a pas d'importance, ils te connaissent, si tu participes dans les *mailing*



*lists*, on voit ton nom et plus on voit ton nom et plus on considère que tu es quelqu'un de sérieux".ENT49

D'autres, comme cet ancien directeur technique d'une SSSL française, vont plutôt se référer à l'acceptation ou non du code au sein des logiciels. La revue par les pairs est ici un gage de réputation au sein de la communauté.

"Il y a la revue des pairs dans le logiciel libre, la revue des pairs c'est quand on regarde ce que font les autres pour voir si c'est bien ou pas, donc les gens ont une réputation assez rapidement. " ENT2

Ce dispositif d'appariement fonctionne dans les deux sens, les entreprises devant elles aussi faire la preuve de leur réputation. Dans cette optique, des entreprises comme Google entretiennent des relations de confiance en finançant des communautés via la subvention d'équipements ou d'événements. La plupart du temps, ce sont les développeurs *open source* qui sollicitent directement les ingénieurs Google. Ces derniers jouissant pour la plupart d'une visibilité au sein des communautés, ils avalisent de manière informelle l'accord pour l'octroi des subventions.

"En fait, je ne les choisis pas, il suffit qu'ils le demandent bien souvent et l'on participe. On est très faciles à trouver. Ils me demandent à moi, ou à Steve, ou à nos ingénieurs. On sait que ce sont des gens sérieux, comme c'est basé sur la réputation, nous on finance, et ça permet d'améliorer notre propre réputation. Je fais confiance à cette personne, parce qu'un des ingénieurs lui fait confiance. " ENT47

Cette réciprocité du modèle de confiance nous confirme ainsi que le recrutement repose sur la forme du dispositif d'appariement et non pas sur la forme du marché formel.

## **2.4. La persistance des intermédiaires**

Cette critique du modèle du marché du travail avec, d'un côté, les demandeurs d'emploi avec

leurs propres compétences et, de l'autre, les employeurs à la recherche de ces mêmes compétences, avait déjà été formulée par la sociologie économique. Comme le montre l'étude de Granovetter (1973), dans la majorité des cas, le meilleur moyen de trouver un emploi n'est pas de passer par le marché du travail à proprement parler, mais de passer par des intermédiaires. Il différencie ainsi "les liens forts" des "liens faibles". Les liens forts correspondent aux liens familiaux, ou à ce que l'on appelle communément "les pistons", les liens faibles correspondent quant à eux aux réseaux étendus des personnes.

Si l'on fait la synthèse des ressources sociales mobilisées dans l'accès à la professionnalisation, on peut montrer que les liens forts sont sensiblement marginaux. Ceux qui ont trouvé leur travail via des liens amicaux ne représentent que 22% des cas et il s'agit très rarement d'un lien fort. Nous n'en avons rencontré qu'un unique cas : le développeur avait été recruté par l'entreprise dans laquelle travaillait sa femme. Nous avons cependant observé d'autres cas de recrutements qui correspondent à des liens familiaux plus faibles, c'est-à-dire avec un degré de filiation plus lointain.

*Types de liens ayant mené à la professionnalisation dans le secteur du logiciel libre*

<b>Type de liens</b>	
<b>Universitaires</b>	4,92%
<b>Professionnels</b>	22,95%
<b>Annonces</b>	9,84%
<b>Amicaux</b>	22,95%
<b>Communautés</b>	39,34%

N=61

On observe également que les intermédiaires classiques du marché du travail (c'est-à-dire les annonces) représentent une part assez marginale, moins de 10%. À l'inverse, les liens faibles (via la communauté) sont primordiaux, puisqu'ils concernent près de 40% des personnes. Des liens faibles plus classiques tels que les réseaux professionnels sont eux aussi présents.

**2.5. Les négociations ne sont pas exclusivement salariales**

Le recrutement implique bien évidemment des négociations entre le futur employé et l'employeur concernant les termes du contrat qui les liera et notamment le salaire, qui est un point déterminant. Dans le cas du logiciel libre, si l'on reprend l'argument de l'opportunisme défendu par Lerner et Tirole, ceci impliquerait qu'au moment du recrutement dans l'entreprise dont l'activité est centrée autour du logiciel libre, il y a un effet d'opportunité, les nouveaux salaires proposés étant substantiellement supérieurs à ceux de l'entreprise que les développeurs quittent ou de celui auquel ils peuvent prétendre à compétence égale dans une autre firme.

Or, c'est le phénomène inverse qui se produit : d'une part, le premier salaire dans une entreprise du logiciel libre est inférieur à ce qu'il pourrait être dans une autre entreprise et, d'autre part, le recrutement dans le monde du libre implique un certain nombre de risques au vu de la vulnérabilité de cette activité émergente.

"À l'époque, j'aurais gagné beaucoup plus en travaillant avec une SSII classique mais le travail était moins excitant, et ils auraient pu m'embaucher en me payant bien plus. "

ENT51

La plupart des développeurs qui ont été recrutés par des entreprises *open source* conviennent que le salaire est inférieur de 15 à 20 % à celui des entreprises classiques. Mais cette diminution de salaire semble être compensée chez les développeurs par la dimension plus intéressante et stimulante du travail proposé.

"Cela devait être 15% de moins, (...) Mais maintenant, je pense que mon salaire est compétitif. " ENT37

On voit que cette différence tend à se résorber avec le temps. Ceci est dû à un double effet: d'un côté, parce que l'industrie du logiciel libre est en pleine croissance et, de l'autre, parce ces développeurs parviennent à valoriser leurs compétences individuelles au sein de l'entreprise.

L'appariement entre employeurs et employés ne repose donc pas en priorité sur des éléments salariaux, mais concerne en premier lieu la convivialité des relations de travail. Comme nous

l'avons vu avec l'exemple de Canonical, les entreprises *open source* tentent ainsi de reproduire les conditions de travail des communautés au sein de l'entreprise. Pour les développeurs, la reproduction de ces conditions de travail semble alors plus importante que le salaire. En important ce mode de travail, les entreprises *open source* intègrent également les principes d'autonomie.

Le principe le plus évident est celui de l'autonomisation de l'objet de production, en participant au libre, les développeurs autonomisent l'objet de la production à la fois du point de vue de la propriété intellectuelle mais aussi d'un point de vue subjectif.

L'autonomie concerne également le rapport à la hiérarchie. En effet une forte opposition existe entre le principe hiérarchique propre à la firme et les principes d'autorité qui ont cours au sein des communautés. L'entreprise, en adoptant un mode de fonctionnement libre, tend à affaiblir cette pression hiérarchique tout en confiant davantage de responsabilités et d'autonomie aux développeurs.

Ainsi les entreprises de l'*open source* encouragent le travail à domicile de leurs employés, comme c'est le cas chez Canonical puisque seuls quelques développeurs travaillent dans les bureaux situés à Montréal et à Londres. Plus étonnant, le travail à domicile des développeurs *open source* concerne aussi des entreprises plus traditionnelles, comme c'est par exemple le cas dans la banque Morgan Stanley:

"On évalue sur le travail qui est fait, pas sur le temps passé à le faire. On paie les personnes assez bien, et cela peu importe le temps qu'ils passent au bureau. Ils peuvent aussi travailler depuis chez eux, et ils viennent au bureau 2 ou 3 fois par semaine. Moi, j'habite dans la même ville que là où je travaille, donc je dois rester chez moi, peut-être une fois par semaine. Par exemple, quelqu'un a dû déménager pour l'Afrique du Sud pour des raisons personnelles, et il continue à travailler pour nous depuis l'Afrique du Sud. C'était ça ou on perdait cette personne donc..."ENT49

Dans les recrutements, lors des négociations avec l'employeur, la question de la localisation du

travail a toute son importance et est un élément souvent plus important que la rémunération. Les entreprises peuvent alors être amenées à faire d'importants sacrifices en ce qui concerne la localisation géographique des développeurs.

Ainsi, Brian, un développeur *Apache* et *Subversion*, nous a relaté les conditions de son recrutement chez Google: Il a fait carrière chez Apple puis chez CollabNet, une entreprise directement issue du logiciel libre, avant d'être recruté par Google. Il a été approché par un ancien collègue:

"J'ai été approché par Google, un de mes amis de Subversion, Steve a été recruté par Google, et auparavant, il travaillait aussi pour CollabNet. " " Il n'est resté que six mois, mais on se connaissait depuis longtemps sur le projet, et au bout d'un an chez Google, il m'a approché. "ENT27

C'est en partie parce que Brian participait au projet Subversion que Google a décidé de le recruter. A l'époque, Google met en effet en place un outils appelé "Big Table" qui utilise le *Contrôle de Versions*, ce qui est la particularité technique de Subversion.

De son côté, s'il a choisi d'accepter le travail, c'est parce que le projet est intéressant.

*"Et pourquoi avez-vous choisi de travailler chez Google?"*

Parce que je connaissais des gens qui travaillaient chez Google, le boulot est très intéressant. Et je pense que l'entreprise fait des choses très intéressantes pour le monde, pour la science, et pour l'*open source*. Ils ont aussi de bons produits." "Vous savez, des personnes très bien travaillent chez Google, des personnes étonnantes, des pointures..."  
ENT27

L'une des conditions inhérente à son recrutement était de rester à Chicago et donc, de ne pas rejoindre le campus de Mountainview où tous les développeurs *open source* sont réunis.

"Parce que si je dois partir de Chicago, je dois, à la fois, trouver une nouvelle maison, mais aussi une nouvelle femme. (*rires*). J'ai rencontré ma femme à Chicago, elle est de Chicago et nous aimons beaucoup Chicago, j'ai fait mes études là-bas. "ENT27

On peut noter que l'entreprise a ainsi recruté un ensemble de développeurs qui travaillaient précédemment sur le projet Subversion, parmi lesquels trois sont amis et vivent à Chicago.

Nous avons par ailleurs évoqué ce même recrutement avec la personne responsable chez Google, elle affirmait avoir confiance dans la capacité d'autonomie au travail de l'équipe d'amis développeurs de Chicago.

*"Mais pourquoi y a-t-il une équipe à Chicago?"*

Parce qu'ils y vivaient, et voulaient rester à Chicago. Oui.

*"C'était leur condition?"*

Oui et on a accepté, mais en fait pour ces personnes, ce n'est pas un problème, ils ont l'habitude de travailler dans un environnement distribué. On savait donc que ça pouvait marcher comme ça, parce qu'ils avaient l'habitude auparavant". ENT47

### **3. Le cas des développeurs indépendants: un marché de compétences ?**

On pourrait objecter que l'un des effets du logiciel libre sur le marché du travail des informaticiens a été une plus grande segmentation de celui-ci et que de plus en plus de développeurs exercent leur activité en tant que développeurs indépendants et donc par conséquent

ne sont pas directement concernés par le recrutement. Ainsi, si les éléments que nous avons mis en avant tendent à montrer que le marché de compétences ne correspond pas à ce qu'il se passe réellement pour les développeurs de logiciel libre salariés, on pourrait néanmoins penser que cela correspond à la réalité pour les développeurs indépendants. En effet, en tant qu'indépendants, l'objectif est de conclure des contrats avec des entreprises clientes pour la réalisation de divers travaux. Barley, S.R Kunda, G. Evans, J. (2002) mettent en lumière cette tendance à la contractualisation des informaticiens. Ils y voient dans la tradition classique, une meilleure adéquation des compétences et une opportunité pour les développeurs de percevoir des rémunérations plus importantes.

Pour ces travailleurs indépendants, les éléments concernant la rémunération peuvent effectivement nous amener à penser qu'il existe une similitude avec le modèle du marché de compétences, et c'est ce que nous révèle ce développeur français:

"En fait, j'ai pas mal de petits boulots liés à *Firefox* et des petits jobs qui rapportent pas mal. Au début, c'est 500 € la journée puis aujourd'hui j'en suis à 1200 € la journée. Ça a évolué. "ENT51

Il y aurait ainsi un véritable marché de compétences, basé sur la demande (le nombre d'utilisateurs) et l'offre (le nombre de développeurs ayant les compétences précises).

"C'est assez simple, des experts *Mozilla* en France, on n'a pas trop le choix, ..., je les connais tous. Sur les quatre qui existent, il y a en deux qui ont déjà leur entreprise qui s'appelle "xxx", dédiée à *Firefox*, et ils ont déjà trop de boulot par rapport à la demande. (...) il y a 30 000, 50 000 utilisateurs qui ne peuvent pas utiliser le Web, ce sont des gens qui veulent reprendre l'application *Firefox*,... des gens composant d'extension pour leur site Web, ça fait beaucoup "ENT51

Ces compétences peuvent aussi rapidement devenir obsolètes :

"Il y a trois ans, lorsque j'ai commencé et maintenant, ...Aujourd'hui, des gens qui avaient

mon niveau de trois ans, il y en a maintenant près de 700. " ENT51

"Il y a un site Web qui recense tout ce qui se passe sur *Mozilla* en France, je me rends compte de la quantité de personnes qui sont capables de faire ce que je faisais il y a trois ans mais à l'époque on n'était que cinq. Donc moi, j'ai continué à évoluer en fait, tout le monde évolue donc je suis obligé d'évoluer aussi et donc, si je travaille sur autre chose à côté, c'est aussi pour gagner en compétences, pour toucher des choses auxquelles je n'aurais jamais pu toucher dans le cadre professionnel. "ENT51

Si, effectivement, pour ces indépendants, la situation est proche du modèle du marché de compétences, il n'en reste pas moins que dans les faits, il subsiste un ensemble de d'interactions hors marché entre ces mêmes développeurs indépendants; une coopération qui passe principalement via la communauté.

Les développeurs indépendants, qui doivent répondre à des commandes précises émanant d'autres entreprises, peuvent être conduits à diviser le travail entre une multitude de développeurs, selon leurs compétences, expériences et disponibilités. Le travail se disperse ainsi selon les relations sociales existantes entre les développeurs indépendants.

"Lorsqu'on demande des choses simples, je "forward" à des gens que je connais ou des gens qui veulent se lancer dans le métier. " ENT51

Autre exemple, Jérôme, de la communauté *Gozza*, devait, au moment de l'entretien, effectuer une tâche sur les serveurs d'EDF, il tombe alors sur un problème qu'il n'a jamais rencontré auparavant et fait appel au chef de la communauté.

"Avec le chef du projet, on collabore sur beaucoup de trucs, là par exemple, je me rends à Munich, parce qu'il y a les gens qui font la migration, ce sont ces gens qui font le programme *Gozza*. Munich a déjà dépensé énormément d'argent pour faire cette migration. Je vais le voir parce que j'ai besoin de conseils, et on va s'occuper d'ajouter des développements pour que je puisse les déposer chez EDF.



*Le chef du projet a donc lui-même sa société?*

Oui, il a lui-même sa société (...) je vais à Munich pour le voir, comme ça pendant trois jours, on discute, ça permet de pouvoir tourner autour du truc. (...). Par exemple, je peux lui dire : " je suis chez tel client, il faudrait ça mais je n'ai aucune idée ". Par exemple lorsque l'on a décroché le contrat EDF, je lui ai envoyé le cahier des charges que l'on a reçu, je lui ai envoyé le dossier. Je lui ai posé des questions et il m'a dit : " ça, ça et ça on peut le faire". Puis je suis parti chez EDF (...) donc avec lui on dialogue, on est sur IRC, et on discute, on s'envoie des bouts de code. Et donc, dans le cadre d'EDF, on a ajouté des fonctionnalités à Gozza qui vont être intégrées dans la branche classique mais qui seront en fait payées par EDF. Notre modèle, il ressemble plus à ça. " ENT26

On voit ici qu'il y a un système d'entraide entre membres de la communauté qui exercent la profession d'indépendant. Le hors marché est donc présent dans la collaboration, mais également dans la répartition temps libre/ temps travaillé. En effet, un développeur indépendant de logiciel libre ne peut pas se contenter de répondre aux commandes de ses clients.

"Oui, c'est ce que je disais tout à l'heure je passe 40 % du temps à faire autre chose et puis, si je ne faisais que ce que nous demandent les clients, mon niveau n'augmenterait pas "

ENT51

*Le développeur entrepreneur* doit en effet obligatoirement consacrer une partie de son temps libre à améliorer ses compétences et à garder des liens forts avec la communauté, pour maintenir un rapport passionné avec l'activité.

"Disons, que si j'étais au chômage, je ferais quand même ce que je fais aujourd'hui "

ENT51

Ces divers éléments nous indiquent que le modèle du marché de compétences est également difficilement applicable aux développeurs indépendants. Plus que la coopération autour de

l'activité de codage, des solidarités et l'entraide entre les développeurs que l'on retrouve dans le développement du logiciel libre persistent, même pour des personnes qui par leur statut d'entrepreneur sont mises en concurrence.

#### **4. Conclusion:**

La collectivisation de la production numérique présente dans le logiciel libre et la collectivisation des compétences qu'elle induit ne se concentre donc pas uniquement dans le cadre d'amateur et de bénévole dans lequel elle a émergé. Les interactions avec les entreprises sont bel et bien présentes, elles se concrétisent notamment par les recrutements de développeurs. La validation de ces compétences par le marché passe indéniablement par le recrutement par les entreprises qui interviennent dans ce secteur. À travers l'exemple de plusieurs trajectoires et plus particulièrement les recrutements par Canonical, nous pouvons discuter la nature des compétences et la nature du dispositif ici à l'œuvre.

Nos conclusions tendraient à infléchir l'hypothèse instrumentale qui peut être accolée au logiciel libre. Certes le logiciel libre a créé un marché du travail spécifique hautement qualifié, certes les entreprises savent apprécier leurs compétences et recrutent ces développeurs, cependant nous ne pouvons pas affirmer que les communautés constituent un vivier de compétences pour ces entreprises. Le logiciel libre qui est en position intermédiaire entre communauté et entreprise n'est pas pour autant un dispositif de recrutement.

Il semble cependant que d'une certaine manière le logiciel libre a inauguré une certaine tendance à l'ouverture et la transparence. Au contact de l'entreprise on a ainsi pu voir un glissement depuis l'ouverture du contenu (le code et le travail effectué) vers une interprétation en termes de pratiques et compétences. Dans le cas du logiciel libre, la persistance des communautés professionnelles et un contexte économique favorable font qu'il y a résistance à l'instrumentalisation d'un tel dispositif. Les modalités que nous avons pu mettre en lumière font plus référence à un dispositif d'appariement qu'à un marché de compétences. Et ces utilisations telles que l'on les observe concrètement vont plutôt dans le sens d'une méritocratie, valeur défendue par les communautés du libre.

## **Bibliographie:**

Akerlof, G. Yellen, J. (1986) *Efficiency Wage Models of the Labor Market*, Orlando: Florida Academic Press, 178p.

Arthur M.B. Rousseau, D.M. (Ed.) *The boundaryless career. A new employment principle for a new organizational era*, Oxford University Press, 394p.

Barley, S.R. Kunda, G. Evans, J. (2002) "Why do Contractors Contract? The Experience of Highly Skilled Technical Professionals in a Contingent Labor Market." *Industrial and Labor Relations Review*, Vol 55, pp. 234-261.

Cappelli P. (2001) " Why Is It So Hard to Find Information Technology Workers? " *Organizational Dynamics*, Vol. 30, No. 2, pp. 87-99.

Crowston, K. Howison, J. (2003) "The social structure of open source software development teams". In *Proceedings of the International Conference on Information Systems*, Seattle, WA, USA

Demazière, D. Horn, F. Zune, M. (2008) "Les mondes de la gratuité à l'ère du numérique : une convergence problématique sur les logiciels libres" *Revue Française de Socio-économie*. Vol 1, N°1, pp 47-65.

Eymard Duvernay F. et Marchal E. (2000) "Qui calcule trop finit par déraisonner : les experts du marché du travail ", *Sociologie du travail*, 42, pp .411-432.

Frey B.S. Jegen R. (2001) "Motivation Crowding Theory:A Survey of Empirical Evidence" *Journal of Economic Surveys* Vol 15, No5, pp. 214-239

Granovetter, M. (1973) "The Strength of Weak Ties", *American Journal of Sociology*, Vol 78, No 6, pp. 1360-1380.

Godechot O. (2006) "Hold-up en finance Les conditions de possibilité des bonus élevés dans l'industrie financière" , *Revue française de sociologie*, vol. 47, n°2, p. 341-371.

Karpik L.(2007) *L'économie des singularités*, Paris: Gallimard, 373p.

Kerps D. (1990) "Corporate Culture and Economic Theory" in J E Alt et K.A Shepsle (éd) *Perspectives on Possitivs Political Economy*, Cambridge: Cambridge University Press pp. 90-143.

Lerner J. et Tirole, J. (2002) "Some Simple Economics of Open Source" *Journal of Industrial Economics*, Vol. 50, No 2, pp. 197-234

Marchal E. Mellet K. et Rieucan G. (2007) "Job Board toolkit: Internet matching and changes in job advertisements" *Human Relations*, Vol 60, N°7, pp1091-1113.

Robles, R. Duenas, S. et Gonzalez-Barahona (2007) "Corporate involvement of libre software: Study of presence in Debian code over time", in: J. Feller, B. Fitzgerald, W. Scacchi, A. Sillitti (eds.), *Open Source Software* , vol. 234 of IFIP, Springer,

Williamson, O.E. (1985) *The Economic Institutions of Capitalism*, NY: Free Press, 450p.