

Le travail des développeurs de logiciels libres.
D. Demazière / F. Horn / N. Jullien

**Communication à la journée du CLERSE
« La représentation économique de l'acteur au travail »
Villeneuve d'Ascq, les 20-21 novembre 2003**

Le travail des développeurs de logiciels libres.

La mobilisation dans des « communautés distantes »

Didier DEMAZIERE

(CNRS, laboratoire Printemps, UVSQ)

François HORN

(CLERSE, IFRESI)

Nicolas JULLIEN

(GET-ENST Bretagne / Marsouin)

(version provisoire)

Le travail des développeurs de logiciels libres.
D. Demazière / F. Horn / N. Jullien

Les *logiciels libres* sont des logiciels distribués avec leur code source (texte du programme écrit dans un langage de programmation compréhensible par l'être humain) et avec l'autorisation de les modifier et de les redistribuer librement, ce qui les différencie radicalement des logiciels dits privés ou propriétaires. La libre disposition du code source implique la gratuité de fait de ces logiciels. Et si en apparence des sociétés vendent des logiciels libres, elles commercialisent en réalité des activités liées au logiciel (sélection de logiciels, réalisation de la copie, distribution, installation, garantie, maintenance, assistance...) et non le logiciel lui-même que n'importe quel utilisateur peut copier ou télécharger librement et gratuitement.

Compte tenu de ces caractéristiques, les analyses économiques ou sociologiques portant sur les logiciels libres ont tendance à mettre l'accent sur les spécificités du produit lui-même ou sur les particularités de sa diffusion (Horn, 2004). Pour notre part, nous adopterons un autre point de vue, en focalisant notre regard sur la production des logiciels libres, sur les activités des développeurs informatiques, sur les modalités de mise au point et de fabrication de ces biens. Car ceux-ci offrent un terrain propice pour interroger le travail, les acteurs au travail. En effet, ce travail présente, de prime abord, de fortes singularités, soulignées à l'envi par les promoteurs « du logiciel libre » (l'emploi du singulier visant à construire par la rhétorique l'unité d'un objet et à en accroître ainsi l'importance et la taille) : le développement des logiciels libres est fondé sur le volontariat et le bénévolat des participants, dans un mode d'organisation coopératif qui s'appuie largement sur les commodités organisationnelles issues d'Internet. Ces propriétés sont d'autant plus remarquables qu'elles caractérisent une activité productive, orientée vers la fabrication de produits répondant aux besoins d'utilisateurs, et par conséquent soumise à des contraintes d'efficacité et de qualité.

Cette configuration conduit à s'interroger sur les caractéristiques de l'action collective qui permet de passer d'engagements individuels volontaires, et potentiellement volatiles et

instables, à la réalisation d'une production collective, impliquant continuité et pérennité : quels mécanismes de régulation permettent de répondre aux exigences des utilisateurs, quelles modalités d'organisation du travail permettent d'obtenir une production de qualité, comment les contributions des développeurs sont-elles agencées en un produit cohérent, comment une action collective est-elle produite ? La production de logiciels libres ne peut être considérée comme le résultat contingent de la rencontre spontanée entre des engagements individuels indépendants. Elle suppose certaines formes de mobilisation d'acteurs au travail, à même d'assurer une certaine continuité de leurs engagements et d'organiser un agencement de leurs contributions. Quelques observations empiriques liminaires alimentent et soutiennent cette piste d'investigation.

La production de logiciels libres apparaît comme une activité non profitable puisque les produits ne sont pas vendus mais diffusés gratuitement, de sorte qu'elle est souvent décrite sous le registre du bénévolat ou du volontariat. Pourtant les développeurs occupent des positions statutaires très diverses (étudiants, salariés d'universités ou de centres de recherche publics, salariés d'entreprises privées dont l'activité est liée à des logiciels libres, autres salariés d'entreprises informatiques ou de services informatiques d'entreprises diverses...), qui induisent des connexions hétérogènes entre l'activité de développement de logiciels libres et l'activité pour laquelle ils reçoivent une rémunération. La première peut s'effectuer hors du temps de travail (rémunéré), à titre exclusif ou non, mais elle peut aussi s'effectuer sur le temps de travail (rémunéré) et peut alors selon les cas être cachée, tolérée, officieuse, officielle, prescrite, reconnue, valorisée. Ainsi le développement de logiciels libres est inscrit dans des régimes juridiques et des régimes temporels pluriels.

Ces figures hétérogènes débordent largement le bénévolat et le volontariat, et elles indiquent aussi un autre enjeu de cette activité productive : celui de la coopération entre contributeurs, sans laquelle il n'y a pas de mise au point d'un produit utilisable. Or, de manière générale, ces contributeurs ne sont pas inscrits dans la même organisation, sont dispersés, ont des relations médiatisées par le réseau Internet, ne sont pas reliés par les fils d'un organigramme

quelconque. Cette absence d'interactions directes, codifiées et prescrites entre les producteurs est contrebalancée par le partage d'un sentiment d'appartenance à un collectif spécifique, à l'identité fortement marquée. Du moins c'est ainsi que l'on peut interpréter les références récurrentes aux « communautés du libre » dans les propos des contributeurs. Cette terminologie indigène ne livre pas d'emblée sa signification, mais elle indique une piste pour comprendre les manières dont l'action collective est réalisée en l'absence des leviers organisationnels qui encadrent habituellement les activités de travail et les acteurs au travail.

Le travail des développeurs de logiciels libres est donc à la fois une activité individuelle réalisée dans des conditions fort hétérogènes et une action collective aux modalités de production originales. Nous proposons d'analyser le travail des développeurs de logiciels libres à partir de la notion paradoxale de « communauté distante », qui vise à rendre compte de la tension entre d'une part la force du sentiment d'appartenance à un monde spécifique repérable dans le discours des acteurs et d'autre part les distances qui séparent les contributeurs d'un point de vue relationnel, statutaire, biographique. Ce faisant, l'objectif est de parvenir à une description, nécessairement plurielle, de formes de « communautés distantes » qui permettent la production d'un bien dans des conditions sociales et organisationnelles originales. De manière plus générale, cette notion met sur la piste de modes de coordination associant deux formes d'action collective habituellement opposées et antagoniques : une forme communautaire fondée sur le sentiment subjectif d'appartenir à une même communauté, et une forme sociétaire fondée sur la coordination d'intérêts et le partage d'objectifs motivés (Tönnies, 1887, Weber, 1921).

Dans un premier temps nous examinerons les manières dont les acteurs individuels s'organisent pour contribuer à une production, et nous mettrons l'accent sur les formes de coopération et de coordination mises en œuvre pour satisfaire les contraintes d'efficacité et de qualité associées à la diffusion d'un produit. Dans un deuxième temps nous renverserons la perspective pour examiner les manières dont les acteurs individuels se mobilisent, et nous soulignerons les mécanismes d'engagement et de participation qui rendent compte de leur contribution à la production de logiciels libres. Ces deux dimensions, pour nous indissociables, sont explorées à partir d'une enquête réalisée auprès de développeurs de

Le travail des développeurs de logiciels libres.
D. Demazière / F. Horn / N. Jullien

logiciels libres français (et aussi des personnalités défendant cette forme de production, les « figures » du libre). Cette enquête comporte deux volets articulés : le premier vise à comprendre les processus d'engagement et de participation individuels dans cette activité, le second vise à saisir les modalités d'organisation et de régulation sociales et économiques de cette activité. Nous avons recours à des méthodes variées : l'approche biographique fondée sur des entretiens, en particulier pour le premier volet, l'étude de cas de projets et l'enquête par questionnaire visant à reconstituer les réseaux relationnels entre contributeurs, en particulier pour le second volet¹. La collecte des matériaux empiriques est encore loin d'être achevée, et nous ne pouvons nous appuyer ici que sur des éléments partiels.

¹ A cet égard notre enquête se différencie nettement d'autres études déjà réalisées sur les « motivations » des développeurs, appuyées sur des questionnaires (FLOSS, 2002). Si nous perdons en extension statistique, nous pouvons saisir de manière plus précise les processus de participation des développeurs en les articulant avec les règles de fonctionnement des groupes et projets dans lesquels ils s'inscrivent.

1. Un travail collectif : organiser une production à distance

La production de logiciels libres est souvent réalisée selon un schéma où un seul individu écrit la totalité d'un programme, forcément de taille limitée. Même dans ce cas, les mises à jours, la correction des bogues, les développements ultérieurs peuvent être socialisés. Et dans les projets plus ambitieux, ce qui est le cas de la plupart des logiciels libres qui atteignent une certaine renommée, la coopération de plusieurs développeurs écrivant des fragments de programme est requise : des règles doivent être édictées et des instances prévues pour organiser les interfaces, pour distribuer l'activité, pour combiner les contributions, pour mettre en forme le produit final. Dans les entreprises produisant des logiciels privés (ou propriétaires) l'organisation du travail est codifiée dans des organigrammes et confiée à une hiérarchie qui contrôle la réalisation des tâches et agence le travail des développeurs. La production de logiciels libres a été analysée comme exacerbant certaines tendances observées dans les sociétés qui se sont imposées sur le marché des logiciels, en étant fondée sur un « ensemble de coutumes de coopération opposé à la direction par coercition » (Raymond, 1998), sur des stratégies de libre coopération, dites de « donnant-donnant » (Printz, 1998). Ces formes organisationnelles faiblement hiérarchisées et peu formalisées constituent un « modèle du bazar » en opposition au « modèle de la cathédrale » (Raymond, 1998) et renvoient à un modèle émergent plus général souvent appelé « connaissances distribuées » (Thévenot, 1997).

Pour l'activité de production de logiciels libres le plus important est d'identifier avec précision les mécanismes de distribution du travail et de coordination des acteurs au travail. Car ceux-ci sont placés dans des configurations relationnelles atypiques au regard des finalités de production : les contributeurs n'appartiennent pas à la même organisation, n'y sont pas liés par une relation salariale, ne travaillent pas dans les mêmes locaux, mais participent à un projet dans des conditions fort diverses, s'y engagent pour des raisons variables, y dépensent des temps différents, poursuivent des objectifs multiples. Avant d'examiner le fonctionnement de collectifs concrets orientés vers la production de logiciels spécifiques, nous décrirons quelques propriétés plus transversales qui structurent et organisent le travail des développeurs de logiciels libres.

1.1. Relier des travailleurs isolés, agencer des productions individuelles

Les contributions à des logiciels libres reposent sur des engagements volontaires qui ne sont pas encadrés par des obligations juridiques. Mais pour que ces différentes contributions puissent constituer un logiciel, il est indispensable que la collaboration soit organisée, en raison même des propriétés du produit. Car si un logiciel est un texte, c'est un texte « actif », un texte qui agit dans la mesure où il se compose d'un ensemble d'instructions qui seront exécutées automatiquement par une machine, ce qui nécessite une cohérence extrêmement forte des différentes parties de ce texte. Et cette cohérence ne peut être obtenue naturellement ou spontanément car, même si certains projets de logiciels libres sont l'œuvre d'équipes restreintes, les logiciels libres les plus importants associent plusieurs centaines de développeurs et même plusieurs milliers pour les projets les plus ambitieux et connus.

Or l'essentiel de l'activité de contribution à un logiciel libre s'effectue de manière solitaire, comme le souligne Ernest², développeur Débian (une distribution Linux) : « ça reste en solitaire, comme la plupart des personnes au sein de Débian ; Débian c'est 1000 personnes qui, en solitaire font un ensemble ». Par cet aspect cette activité s'apparente à une production artisanale : Linus Torvalds, initiateur du projet Linux, souligne que « les logiciels libres sont faits par des artisans amoureux de leur art » (Le Monde, 27-28 septembre 1998). Ainsi donc, le développement des logiciels libres mêle de façon originale les caractéristiques d'une production nécessairement collective et d'un travail très individuel dans sa réalisation. L'œuvre collective est la combinaison de contributions individuelles qui n'acquiescent de signification que si elles parviennent à s'intégrer dans l'objet collectif qu'est le logiciel. L'agrégation des productions individuelles en un produit collectif signifiant et utile est permise par une série de mécanismes organisateurs de la production qui, s'ils se différencient des régulations institutionnelles, codifiées ou juridiques, n'en sont pas moins efficaces.

Le premier mécanisme réside dans le contrôle des productions individuelles. La mise en place d'instances de contrôle des contributions est systématique dans tous les projets. Leur mode de constitution et leur fonctionnement diffèrent, mais leur existence est bien la trace d'une organisation explicite et formelle. Cette organisation a nécessairement une structuration

² Les prénoms désignent des développeurs de logiciels libres que nous avons interviewés. Ils ont été modifiés de manière à préserver l'anonymat des personnes rencontrées.

hiérarchique –même s’il n’existe pas une hiérarchie qui contrôle étroitement les tâches et la coopération entre développeurs- une (ou plusieurs) instance(s) ayant un pouvoir d’arbitrage entre les différentes propositions et de sélection des développements qui seront validés pour être intégrés dans le logiciel. Bernard décrit ainsi un monde « très très structuré. Et puis il y a de la concurrence. Plusieurs personnes peuvent proposer différents modules pour résoudre tel problème, et c’est ce groupe de décideurs pour tel ou tel élément dans le logiciel, qui va les comparer et dire : on sélectionne celui-ci et pas l’autre. Donc la concurrence est ouverte en termes intellectuels, si j’ose dire, après il y a bien sélection ». Les productions individuelles ne sont pas prescrites ou commandées par un centre de décision, mais elles sont toujours évaluées et validées, ou rejetées.

Le deuxième mécanisme provient des propriétés du logiciel, et particulièrement de sa modularité. La structuration modulaire rigoureuse du logiciel (découpage en composants, respect scrupuleux des standards existants, définition précise des interfaces entre composants) autorise une fabrication émiettée, et une composition par assemblage de fragments écrits indépendamment les uns des autres. Cette architecture ouverte et modulaire est une nécessité du fait de l’absence de hiérarchie ayant le pouvoir de canaliser et de guider l’activité et les contributions des développeurs. Destinée à faciliter la coopération entre les acteurs, elle est aussi unanimement considérée comme un facteur décisif de qualité des logiciels, mais est rarement respectée par les entreprises commerciales (Jullien, 2001). De plus elle permet à « la majeure partie des phases d’architecture, d’implémentation et de réalisation d’un logiciel d’être menées en parallèles » (Brooks, 1996, p. 203) et à différentes propositions indépendantes d’être développées, l’une d’elles étant finalement retenue. Comme le souligne Himanen (2001), dans une telle forme d’organisation, fortement horizontale même si elle n’est pas un réseau totalement plat, les instances de décision ont une légitimité uniquement technique fondée sur la compétence reconnue par les autres développeurs et n’ont pas de pouvoir de nature économique par rapport à ceux-ci : l’absence d’appropriation privée du logiciel produit donne la possibilité pour un groupe de développeurs insatisfaits des décisions prises de développer un projet alternatif à partir du logiciel existant.

Le troisième mécanisme concerne la correction des erreurs. Le développement de logiciels sous forme de logiciels libres permet « d’engendrer des effets gigantesques d’apprentissage par l’usage, c’est-à-dire d’exploiter au mieux une fantastique *intelligence distribuée* : les

millions d'utilisateurs qui révèlent des problèmes et les milliers de programmeurs qui trouvent comment les éliminer » (Foray, Zimmermann, 2001, p. 6). Le fait de pouvoir mettre à contribution l'attention et la puissance de réflexion d'un ensemble de développeurs beaucoup plus étendue que celui de n'importe quelle entreprise commerciale est particulièrement efficace en raison de deux caractéristiques des logiciels : la complexité de cet objet technologique qui fait qu'un système composé de milliers de développeurs travaillant sur le même logiciel durant une longue période reste dans une phase de rendement croissant ; le caractère intangible du logiciel qui lui permet de circuler rapidement à un coût quasiment nul et qui explique que les améliorations apportées peuvent profiter directement à l'ensemble des utilisateurs sans investissement supplémentaire. En particulier, ce mode de développement est très performant pour éliminer les erreurs, tâche qui constitue fréquemment la majeure partie de l'activité de conception d'un logiciel : à la différence d'un logiciel privé qui est le plus souvent revu par des personnes très proches des auteurs et qui commettent les mêmes erreurs, un logiciel libre pourra être scruté par des personnes ayant des méthodes et des outils de travail très divers, ce qui a pour conséquence que « chaque problème sera rapidement isolé, et que sa solution semblera évidente à quelqu'un » (Raymond, 1998).

Le quatrième mécanisme est l'identification de l'auteur de chaque ligne de programme. En effet, quand l'activité de programmation s'effectue dans le cadre du développement d'un logiciel libre, la création de chaque personne est clairement spécifiée. L'identification des auteurs de logiciels libres est scrupuleusement garantie par leurs licences : les lignes de codes sont signées par leurs auteurs, le nom du développeur étant inscrit près des parties de code source sur lesquelles il a travaillé, et dans la plupart des logiciels libres il existe un fichier appelé « *credits* », qui recense les principaux contributeurs du logiciel et leurs apports. De plus, les licences des logiciels libres intègrent des dispositifs visant à préserver, en cas de modifications du logiciel, la réputation de l'auteur original. Dans un logiciel libre la partie qui a été effectuée par chaque développeur est exposée publiquement ce qui permet de juger de sa qualité. Ce point est important car les qualités d'un logiciel ne sont pas directement perceptibles à partir de son utilisation, dans la mesure où il s'agit d'un bien système, qui interagit avec d'autres logiciels et des composants matériels. C'est ce qui explique les très grandes difficultés en cas de défaillance d'un système informatique pour déterminer quel est le logiciel responsable. Le caractère non visible des qualités réelles d'un logiciel quand son

code source n'est pas public est illustré par le refus des éditeurs de logiciels privés de dévoiler le code source de leurs logiciels, y compris quand ces logiciels ne sont plus commercialisés. Dans un logiciel libre, « la disponibilité du code source met en jeu l'orgueil du programmeur, qui sait qu'il va être jugé par ses pairs. Et il existe pour un informaticien peu de satisfactions personnelles aussi grandes que celle d'avoir contribué à écrire un programme qui est apprécié, utilisé, repris et amélioré pendant dix ans par des milliers de programmeurs et des millions d'utilisateurs, le tout pour ses mérites propres » (Di Cosmo, Nora, 1998, p. 164).

Le cinquième mécanisme est la compétition qui colore les rapports entre contributeurs. La personnalisation des contributions est au principe de la reconnaissance du travail réalisé. Chaque développeur peut juger de la qualité de son travail et de sa reconnaissance : sélection de sa proposition de contribution à un logiciel, choix de sa suggestion de correction, intégration de son logiciel dans une distribution, nombre de téléchargements de son logiciel. Eric Raymond (1998) insiste sur « la perspective autogratifiante de prendre part à l'action et d'être récompensés par la vue constante (et même quotidienne) des améliorations de leur travail ». Cette visibilité des contributeurs organise une concurrence et crée « une situation où la seule évaluation possible de la réussite dans cette compétition est la réputation que chacun acquiert auprès de ses pairs (...). Les participants rivalisent pour le prestige en donnant du temps, de l'énergie, et de la créativité » (Raymond, 2000, p.10 et 23). Compte tenu de l'hétérogénéité des régimes juridiques et des régimes temporels dans lesquels les développeurs sont inscrits, il n'est pas sûr que cette compétition conduise toujours à une intensification des engagements, à un accroissement du temps et de l'énergie dépensés par chaque participant. Mais du moins elle contribue à réguler l'accès et le maintien dans ce travail, et elle participe à la production de la qualité. D'une certaine manière le modèle du logiciel libre est organisé selon les mêmes principes que la recherche scientifique : circulation libre de l'information qui est critiquée publiquement, contrôle par les pairs, proposition de solutions alternatives, concurrence acharnée entre les équipes (Lang, 1999).

L'activité de production de logiciels libres est organisée autour de règles qui permettent d'assurer la réalisation de contributions individuelles, leur agencement dans des programmes, et la qualité des logiciels ainsi réalisés. Ces mécanismes régulateurs assurent l'agrégation de personnes isolées ou distantes autour de projets collectifs. Mais ils diffèrent selon les projets,

et configurent alors des modes organisationnels différenciés que nous pouvons explorer maintenant.

1.2. Des modes organisationnels variés, des groupes sociaux divers

Si les logiciels libres peuvent être, comme nous l'avons noté de manière liminaire, aisément spécifiés par rapport aux logiciels privés, ils forment néanmoins un ensemble hétérogène, allant du produit complexe sur le plan technique, visant à satisfaire les besoins d'utilisateurs extrêmement nombreux, nécessitant la collaboration d'un ensemble important de contributeurs, comme ce peut être le cas d'un système d'exploitation (Linux) ou d'une offre bureautique (OpenOffice), jusqu'à un produit plus simple à élaborer, remplissant une fonction limitée, réalisable par un seul et même individu, comme par exemple un utilitaire spécifique à un périphérique (appareil photographique numérique) ou un pilote d'imprimante intégrant des caractères spéciaux. Cette hétérogénéité du produit trouve son pendant dans les modes de production : nombre et caractéristiques des contributeurs, organisation de la coopération, place et intérêt accordés aux utilisateurs potentiels. Autrement dit, les mécanismes généraux identifiés précédemment trouvent des traductions particulières dans chaque projet et chaque opération de développement.

Les manières dont les tâches sont distribuées, dont la qualité des programmes est évaluée, dont les erreurs sont détectées et corrigées, dont les contributeurs sont recrutés, correspondent chaque fois à des configurations particulières. Et chacune de ces configurations peut être considérée comme une tentative pour construire une coopération efficace, et au-delà une cohésion minimale, entre des travailleurs séparés les uns des autres, œuvrant à distance, privés d'interactions directes. Un volet de notre recherche consiste précisément à étudier de manière approfondie des projets diversifiés, en réalisant des entretiens croisés avec des participants, en distribuant des questionnaires par voie électronique, en observant le fonctionnement des listes de diffusion, et en collectant des informations accessibles par d'autres canaux. L'objectif est d'identifier des modes organisationnels typiques, qui constitueraient une description systématisée des formes concrètes de ce que nous avons appelé les « communautés distantes ».

L'enquête n'est pas suffisamment avancée pour esquisser une telle typologie, mais du moins certaines caractéristiques apparaissent d'ores et déjà saillantes pour rendre compte de la diversité organisationnelle : la taille du cercle des contributeurs principaux (qui peut d'ailleurs varier fortement au cours de l'évolution du projet), mais aussi celle des autres cercles (contributeurs secondaires proposant des corrections mineures, utilisateurs signalant des erreurs) ; les caractéristiques des initiateurs du projet, qui peuvent être des individus ou des institutions publiques ou privées de taille variable ; les propriétés des liens qui les fédèrent, qui peuvent être limités à la participation au projet ou avoir été noués antérieurement (réseau d'anciens d'une filière de formation ou de collègues, consortium d'entreprises constitué sur d'autres finalités, etc.) ; la nature des objectifs et perspectives qui les réunissent et qui peuvent osciller entre des composantes multiples et non exclusives (défi technique à relever, créneau marchand à occuper, valeurs à défendre, etc.) ; les origines et les circonstances du lancement du projet (amélioration de fonctionnalités précises, reprise d'un projet en déclin, planification d'objectifs ambitieux, etc.)... Ces propriétés ont des incidences sur l'organisation, et tout spécialement sur la division du travail, la prescription des tâches, le contrôle des résultats, la validation des productions, le recrutement de travailleurs, le système de rétribution, les qualités privilégiées du produit, l'importance accordée aux utilisateurs. Nous pouvons présenter des éléments de quelques cas, qui suffisent à suggérer l'ampleur des contrastes dans les modes organisationnels.

Un cas de figure rencontré fréquemment, en particulier pour les projets de taille réduite, se caractérise par une hiérarchie étanche et fixe, qui confine à une monopolisation par un individu des décisions. Le fonctionnement est conçu pour marquer et maintenir la distance, non pas seulement spatiale mais aussi sociale, entre le décideur et les autres contributeurs. Cette configuration s'appuie toujours sur une histoire singulière, celle d'un individu qui écrit un logiciel et le propose sur un serveur (Sourceforge). Son produit rencontre alors des utilisateurs, qui dans certains cas peuvent être très nombreux, et qui pour certains d'entre eux ne manquent pas de proposer des corrections, extensions, développements. Mais l'initiateur tente de conserver le monopole de la validation des développements ultérieurs, et en quelque sorte de cantonner les autres développeurs dans des contributions secondaires (signalement d'erreurs, fonctions périphériques au module initial). Tant que ces contributions restent limitées et ponctuelles, la frontière reste très marquée entre des participants ponctuels et un

initiateur garant du produit. Celui-ci peut ainsi renforcer sa légitimité et sa reconnaissance et conserver le monopole, résultant de son initiative personnelle initiale, sur le logiciel libre qu'il a créé. La multiplication du nombre des utilisateurs et des contributeurs, qui indique le succès croissant du logiciel, ne modifie pas forcément ce schéma, car l'initiateur peut constituer une petite équipe, en associant certains développeurs plus réguliers ou déterminants, qui contrôlera les contributions, mais aussi canaliser les contributeurs. Un exemple proche de ce schéma est fourni par le projet Tex, créé, développé, maintenu et expliqué par Donald E. Knuth, qui permet ensuite le développement de plusieurs logiciels de composition typographique, dont le plus connu est LaTeX.

Un autre cas de figure correspond à des projets lancés et pilotés, au moins dans la phase initiale, par un groupe caractérisé par l'interconnaissance entre des personnes ayant des expériences communes et des parcours proches. Cette proximité, sociale et spatiale, des initiateurs, qui par exemple travaillent dans la même entreprise ou partagent les mêmes activités informatiques, est fréquemment associée à une organisation spécifique dont le socle est une représentation d'eux-mêmes comme des professionnels du libre. Cette définition de soi est d'autant plus solide qu'elle s'appuie sur l'inscription des activités de développement dans un cadre professionnel (salarial dans des entreprises informatiques, création d'entreprise, etc.). Elle devient alors très opérante et structurante pour le sentiment d'appartenance au groupe et pour la définition d'exigences en matière de qualité des produits : des professionnels se doivent de mettre à disposition des logiciels qui fonctionnent de manière optimale. Ce souci du produit est une manière d'inscrire l'utilisateur dans le travail de production, mais comme destinataire et non comme partenaire, ce qui introduit une différenciation entre les développeurs, et les utilisateurs, les simples utilisateurs, considérés en quelque sorte comme des profanes. Cette frontière est à la fois marquée et perméable, puisque le groupe des développeurs n'est pas fermé : des utilisateurs proposant des contributions peuvent y entrer, mais cela ne remet pas en cause la netteté des contours du groupe. Dans ce cas la procédure d'entrée s'appuie sur une appréciation des compétences techniques des outsiders, que les contributions qu'ils ont faites permettent de juger, et sur une prise de décision collective, souvent appuyée sur un vote des membres du groupe. Ceux-ci organisent, entre eux, la répartition du travail, la spécialisation des tâches, la distribution des responsabilités sur tel ou tel module du logiciel, selon un modèle d'organisation par projet

assez classique. Mais ce noyau reste néanmoins toujours ouvert à des apports extérieurs, par intégration de contributions mineures d'outsiders ou par élection des outsiders. Le projet Apache (serveur http) est proche de ce cas de figure. Il a été lancé par un groupe d'informaticiens, administrateurs réseau et utilisateurs du logiciel de serveur web du NCSA, lorsque ce dernier a annoncé qu'il se désengageait de ce produit et en stoppait la maintenance. Ces membres sont des professionnels de l'informatique et même du libre, qui travaillent sur le projet dans le cadre de leur activité professionnelle. Leurs motivations s'inscrivent dans un registre technique (le libre permet d'aboutir à des produits plus fiables), et ils ont adopté rapidement une organisation par sous-projets, permise par la modularité logicielle, de manière à responsabiliser les membres du groupe, tant en termes de qualité des produits que de rythmes de production. De nouveaux arrivants ayant apporté des contributions ont rejoint le groupe après cooptation validée par un vote des membres.

L'efficacité de ce type d'organisation suscite des tentatives pour le reproduire avec un noyau initiateur qui n'est plus constitué d'individus mais d'institutions diverses (entreprises, centres de recherches...). La fondation de tels consortiums, encouragée par les pouvoirs publics, regroupent des partenaires qui avaient déjà des relations entre eux. L'organisation des développements est encore plus structurée que dans le cas d'un groupe d'individus. Ici également, la frontière est très marquée entre les utilisateurs et le noyau des développeurs, mais le succès des premiers développements peut permettre de recruter dans le consortium de nouveaux partenaires permettant d'amplifier le projet et de renforcer sa crédibilité. Une illustration est le consortium de développement d'une plate-forme de middleware ObjectWeb initié par des grandes entreprises et des centres de recherche français, qui s'est élargi récemment à des entreprises américaine et allemande.

Un troisième cas de figure est organisé autour d'une institution centrale (entreprise privée, publique, ou laboratoire de recherche), qui initie le projet, y consacre des ressources propres (en temps de travail rémunéré), et pilote son développement. Toutefois celui-ci ne reste pas confiné dans les limites de l'institution et le cercle de ses salariés, puisque le principe même de la production de logiciels libres est la mise à disposition du code source du logiciel et de ce fait la possibilité pour tout utilisateur d'apporter sa propre contribution. Le choix de développer librement le logiciel correspond d'ailleurs à une volonté de l'institution initiatrice de bénéficier de contributions extérieures. L'institution porteuse conserve cependant un rôle

central vis-à-vis des différents cercles de développeurs. Son contrôle est direct, et permanent, sur les principaux développeurs, qui sont rémunérés et liés par une relation salariale contractuelle, et dont elle organise les activités. Quant aux contributions secondaires des utilisateurs, elles sont examinées et évaluées selon des procédures formalisées : généralement la participation de contributeurs externes s'effectue sur des sites web et des listes de diffusion consacrées au logiciel, et peut être structurée par la tenue de conférences. Les évolutions du logiciel sont ainsi d'autant mieux maîtrisées que l'adjonction de contributeurs externes particulièrement productifs et remarquables au noyau décisionnel des développeurs passe par leur recrutement dans l'institution ou leur embauche dans l'entreprise responsable du logiciel. Les groupes sont donc nettement segmentés, les relations entre les membres du noyau sont enkystées dans un rapport salarial. Les exemples proches de ce cas de figure où le projet est initié et piloté par une institution, qui en est une sorte de propriétaire symbolique, sont multiples, qu'il s'agisse d'un centre de recherche (l'INRIA avec le projet Scyla), d'une université (Paris VII et les logiciels Alliance), d'une entreprise (le logiciel Zope développé par la société éponyme aux Etats-Unis, en France le logiciel Glassnost par Entr'Oouvert, le logiciel ERP5 par Nexedi, le logiciel CPS de Nuxeo...). Il arrive aussi qu'une entreprise éditant un logiciel privé décide de le transformer en logiciel libre (Open Cascade pour Matradatavision, CodeAster pour EDF).

Un dernier cas de figure correspond à des groupes plus larges, plus étendus et plus hétérogènes, qui ont fait évoluer leurs règles de fonctionnement à mesure qu'ils grossissaient et agrégeaient des membres dispersés d'un point de vue spatial et non reliés entre eux par des interactions sociales. Les initiateurs, qui forment noyau, participent, à des degrés divers, aux mêmes réseaux relationnels constitués pendant les études notamment, mais quand le groupe s'étend cette communauté d'expériences disparaît, et la cohésion sociale peut être menacée. Cette croissance du groupe est à la fois la résultante du succès d'un produit, qui intéresse de nombreux utilisateurs dont certains deviennent développeurs, et le signe d'une stratégie d'ouverture de la part des initiateurs. Des groupes de développeurs peuvent ainsi atteindre plusieurs centaines de membres, ce qui pose des problèmes spécifiques de régulation de l'activité productive, et indissociablement de préservation de l'identité même du groupe. Dans ce cas, la pérennité du groupe et du projet est assurée par des barrières à l'entrée, de sorte que l'on observe un paradoxe : les groupes les plus ouverts apparemment, c'est-à-dire les plus

nombreux, sont aussi les plus fermés, c'est-à-dire les plus sélectifs. Le recrutement s'appuie, c'est une constante des « communautés du libre » sur la cooptation et sur l'apport déjà réalisé au produit, mais des épreuves spécifiques sont également instituées, mêlant aspects techniques et aspects idéologiques. Cela permet de garantir que tous les membres partagent des compétences techniques et des valeurs, comme si cette proximité de dispositions compensait la distance entre les positions occupées. Il reste que cette équation improbable entre extension du groupe et sélectivité à l'entrée suppose que le logiciel produit ait un attrait particulier et rencontre un intérêt plus large qu'un intérêt d'usage. Par ailleurs, ces barrières à l'entrée permettent de maintenir une faible division du travail à l'intérieur du groupe et une sorte d'égalité de situation ou de statut, de sorte que tout membre peut prendre en charge l'organisation du développement de tel ou tel module. Le projet Debian peut être rapproché de ce cas de figure. Il rassemble plus de mille personnes, qui partagent tous le statut de « développeur-mainteneur », sans hiérarchie (un « project leader » élu une fois par an représente le projet auprès des partenaires extérieurs mais n'a pas de fonction interne). Seules des personnes, à l'exclusion de toute institution, peuvent adhérer à Debian, et les demandes d'adhésion sont très nombreuses. Une procédure formalisée et longue a donc été progressivement mise en place, qui comporte plusieurs étapes : parrainage par un membre du groupe, test d'aptitude technique, test de connaissance de l'histoire, des principes et des valeurs du groupe. Une équipe est chargée de sélectionner les candidats, ce qui conduit à ce que tous les membres du groupe partagent le même ensemble de valeurs par rapport aux logiciels libres (refus de tout brevet, assimilation de la production logicielle à la production intellectuelle, promotion de la licence GPL, etc.).

Ces quelques exemples montrent que les solutions adoptées pour organiser une production à distance sont contrastées et traduisent des contraintes propres aux projets développés, prolongent les dynamiques de lancement de ces projets, expriment les orientations des initiateurs. L'enjeu sous-jacent à des modes organisationnels variés est constant : il s'agit de constituer un collectif à partir d'individus séparés et distants. On a pu entrevoir que cette distance pouvait revêtir des significations fort différentes, et que le collectif ou le groupe produit correspondaient à des configurations fort diverses. Ainsi la catégorie de « communauté distante » doit être dépliée pour être précisément renseignée. Pour en

Le travail des développeurs de logiciels libres.
D. Demazière / F. Horn / N. Jullien

prolonger l'exploration il faut aussi comprendre ce qui amène des individus à participer à l'activité de développement de logiciels libres.

2. Les processus d'engagement individuel

La plupart des études économiques de la participation à des logiciels libres estime que l'engagement s'appuie sur des incitations économiques « classiques », qui se traduiraient par la valorisation pécuniaire ultérieure des compétences des contributeurs à des logiciels libres qui connaissent un certain succès : embauche sur un poste intéressant, accès privilégié à des sources de financement. L'argument repose sur le fait que les dispositifs qui identifient précisément la contribution de chaque personne à un logiciel libre permettent de se constituer un capital de réputation qui agit comme un signal puissant de compétences difficilement évaluables directement (Foray, Zimmermann, 2001). Nos investigations empiriques mettent en valeur des processus de mobilisation plus complexes, et confirment plutôt ce qu'écrit Eric S. Raymond : certes, « parfois la réputation acquise (...) peut se répandre dans le monde réel et avoir des répercussions financières significatives [par] l'accès à une offre d'emploi plus intéressante, à un contrat de consultant, ou aiguïser l'intérêt d'un éditeur », mais « ce type d'effet de bord est rare et marginal (...) ce qui est insuffisant pour en faire une explication convaincante » (2000, p. 8-9).

Nous avons plutôt rencontré des informaticiens pour qui l'engagement dans les logiciels libres avait des conséquences neutres, voire négatives sur le plan matériel, le cas extrême étant celui de Ernest, jeune informaticien qui a quitté un emploi de consultant rémunéré 400 € par jour, pour s'engager dans une SSSL (Société de Services en Logiciels Libres) où il peut consacrer toute son activité à développer des logiciels libres... pour un salaire mensuel de 1200 €. Certes, on peut argumenter qu'il s'agit d'un investissement qui serait rentable à terme, mais il apparaît que même quand des opportunités de valorisation pécuniaire existent, celles-ci ne sont pas systématiquement saisies, comme le montre l'expérience de Richard, gérant d'une des premières SSSL lors du boom de la net économie : « Imaginez qu'à cette époque là, comme toutes les autres entreprises du libre, on ne se payait pas ou on se payait au SMIC. Il y a [...] BNP et AXA, qui arrivent en disant : voilà, vous êtes une entreprise dans le monde du libre. Est-ce que vous voudriez... ? Donc nous, on a dit non. Mais on a un peu hésité d'ailleurs [...] il y avait un moyen pour moi, qui avait 49% des parts, de récupérer quelques centaines de milliers de francs [...] Et puis il y a les Américains [...] VA Linux et Linux Care

[...] qui sont venus nous voir ! [...] Et c'était très dur de résister à ces sirènes là. Nous on a résisté uniquement parce qu'on voulait créer une boîte différente ».

Surtout la validité de l'hypothèse sur la motivation par des incitations économiques repose sur le présupposé d'une contribution basée sur un choix calculé, anticipant des effets à long terme sur l'avenir professionnel. Or, ce que montre les entretiens c'est qu'il s'agit plus d'un processus d'engagement progressif, soutenu par une familiarité croissante avec l'activité de programmation et le « monde social » des développeurs (Strauss, 1978), et scandé par des expériences marquantes au cours desquelles les informaticiens construisent le sens de leur participation en interaction avec d'autres développeurs de logiciels libres. Si les individus ont une production propre et individualisable, celle-ci est le maillon de chaînes de coopération qui, bien sûr, organisent des savoirs techniques spécifiques, mais, surtout, s'incarnent dans des habitudes de travail, des catégories de perception, des univers de discours (Becker, 1988). Dès lors, la mobilisation dans le développement de logiciels libres est intelligible en termes de carrière.

2.1. La carrière de développeur de logiciels libres

Une telle perspective vise à souligner plusieurs traits saillants du travail des développeurs de logiciels libres : il est ordonné en séquences correspondant à une succession de positions dans le monde social correspondant ; la mobilité d'une position à l'autre est le produit de la rencontre de motivations personnelles et de milieux sociaux intégrateurs, l'avancée dans la carrière correspond à un comportement qui devient stable et public et à un renforcement des liens de coopération (Becker, 1963). La progression dans la carrière ne signifie pas seulement l'enrichissement des compétences techniques, mais aussi l'accumulation de compétences sociales impliquant des manières de faire, des façons de voir, des codes de conduite propres à chaque monde social (Hughes, 1958). Nous avons tenté d'identifier des séquences successives, correspondant à diverses modifications : dans les comportements et les activités de l'individu, dans les perspectives et significations qu'il attribue à son activité, dans les interactions et les relations noués avec les autres développeurs.

Accéder au code source : un approfondissement technique

L'activité de développement de logiciels libres ne concerne que des « passionnés de l'informatique », qui se désignent comme tels, c'est-à-dire des personnes qui non seulement ont des connaissances spécialisées et ésotériques approfondies, acquises par l'utilisation intensive des outils informatiques et presque toujours la poursuite d'études supérieures informatiques, mais qui éprouvent aussi un engouement pour l'activité de programmation. Celle-ci conduit fréquemment à la volonté d'accéder au code source de tel ou tel logiciel, soit pour approfondir ses connaissances et explorer plus avant le cœur des logiciels, soit, de manière plus fortuite, pour résoudre des difficultés techniques. Les situations qui imposent techniquement l'accès au code source sont fréquentes. Un cas emblématique est celui de Richard Stallman « inventeur » des logiciels libres en réaction à une imprimante qui se bloquait et où il était dans l'impossibilité de modifier le logiciel qui pilotait l'imprimante pour résoudre le problème. De façon générale, les défauts d'un programme se manifestant par des anomalies de fonctionnement, les connexions entre des logiciels et des composants matériels de plus en plus divers, les adaptations non prévues à des situations précises constituent de multiples occasions où les personnes ayant la compétence pour agir doivent disposer du code source du logiciel. Une motivation complémentaire est l'ambition de comprendre comment un logiciel fonctionne pour se former soi-même à la programmation. Ainsi Pascal explique : « ce qu'il faut admettre quand même, c'est que la prise de conscience de l'importance du phénomène, de l'importance des licences, etc., n'est pas venu du premier coup. C'est à dire que dans un premier temps, ce qui m'intéressait c'était uniquement d'avoir accès, de pouvoir faire des choses avec. Je ne me posais pas à l'époque la question du développement coopératif [...] Quand j'étais en première année à Normale Sup., on avait un cours de programmation-système et j'avais demandé à l'enseignant [...] si par hasard on ne pouvait pas avoir les sources du shell Unix pour voir un peu comment c'était fait ». De même, Ernest raconte : « en entrant à l'université, je me suis dit : tiens, à l'université on va devoir utiliser Unix, tant qu'à faire ça serait bien de voir un petit peu par soi-même avant, à quoi ça peut ressembler. Et il y a Linux, qui ressemble à Unix, qui est du logiciel libre, que je dois pouvoir installer, sur mon ordinateur ». Symétriquement, pour de nombreux enseignants la formation à la programmation nécessite de pouvoir montrer comment les programmes sont construits.

L'examen du code source d'un logiciel apparaît comme normal à la plupart des informaticiens. Mais il est impossible dans le cas des logiciels privés. De ce fait, des informaticiens se tournent vers des logiciels libres pour répondre à leurs besoins ou satisfaire leur curiosité. Car l'utilisation d'un logiciel libre permet d'accéder à sa caractéristique déterminante, à savoir son code source ouvert. Cette première phase d'acculturation au logiciel libre est souvent favorisée par la fréquentation de certaines institutions, notamment universitaires, historiquement favorables aux logiciels libres. Si donc elle s'effectue dans un contexte social organisé, il n'en reste pas moins qu'elle répond encore à ce stade à un besoin personnel et parfois très ponctuel, et qu'elle reste largement déconnecté de l'apprentissage des significations associés aux logiciels libres et de la connaissance des modalités selon lesquelles ils sont produits.

Produire une contribution et la diffuser graduellement

Il reste que cette acculturation s'effectue de manière collective, même quand la géométrie des groupes impliqués est limitée à des étudiants suivant le même cursus et à leurs enseignants. Une partie des membres de ces groupes, qui souvent n'ont encore connaissance que d'un logiciel libre en particulier, va jouer un rôle plus actif. Ce processus est en général très progressif : il commence le plus souvent par la fréquentation du site web du logiciel pour suivre son évolution, puis se poursuit par la participation aux listes de diffusion, rendue souvent indispensable par les difficultés initiales pour tirer partie d'un logiciel libre. Cette participation, qui consiste d'abord à adresser des questions et peut se poursuivre par la proposition de réponses à des questions formulées par d'autres, permet de développer des interactions, à distance, au-delà du cercle initial des proches. Les premières contributions sont souvent périphériques : signalement de bogues, traductions et enrichissement de la documentation...

Ces collaborations épisodiques et ces partages d'expérience permettent l'intégration à un collectif et l'imprégnation de son discours qui donnent du sens aux actions effectuées et peuvent susciter l'envie pour ceux qui en ont les compétences d'approfondir sa participation en proposant des corrections et en écrivant des modules d'importance croissante. La diffusion de ces premières contributions se fait de manière graduelle, en gagnant des cercles de plus en plus larges à mesure que la valeur de la production est reconnue : les premiers destinataires

sont les pairs les plus proches, puis des collègues un peu plus éloignés mais connus personnellement, puis des pairs distants à travers la mise à disposition sur le site. Cette gradation dans la diffusion constitue une sorte d'initiation articulant mise à l'épreuve du novice et validation de sa production. Paul décrit selon ce schéma son expérience d'engagement dans un logiciel libre anglophone de composition typographique : « Petit à petit, je me suis intéressé... Il y avait des choses que je trouvais, notamment en tant que français, qui ne marchaient pas comme je l'aurai voulu, au niveau typographique. Donc, j'ai eu envie de faire des choses et puis [...], j'ai commencé à développer des choses, puis à en parler à des collègues, « Ah ! Ben c'est bien. Tiens, tu peux nous le filer. » Au départ, ça se passe de manière pas officielle. Ce sont des collègues qu'on connaît, « Ben, tu as ça ? Ben, oui, tiens. Ben files-le moi. » Enfin ce n'est pas public, c'est échanger entre des gens qui se connaissent, disons, au niveau humain. Et après, on dépose ça sur des serveurs publics, donc c'est récupéré par des gens qu'on ne connaît pas forcément. Mais ça, c'est une deuxième étape [...]. Ce n'est pas quand on débute... Bon, forcément, les premiers trucs qu'on fait, c'est comme les peintres, ils ne font pas la Joconde du premier coup. Donc, on n'aime pas trop déposer des machins qui vont être critiqués par des gens plus compétents. Je crois que c'est au bout d'un certain temps qu'on se dit, « Ben, tiens. Ca, ça vaut peut-être le coup. » Enfin, ce sont souvent les collègues qui disent, « Si, tu devrais le mettre, franchement... ».

Dans les premières étapes de la carrière il existe donc un mécanisme de contrôle par les réseaux de proximité de la qualité de la production. Quand celle-ci est rendue publique et mise à disposition de tous les utilisateurs, celui qui l'a réalisée devient véritablement contributeur, car il est parvenu à participer aux mécanismes de réciprocité et de socialisation des produits qui sont au principe du logiciel libre. Il en assimile alors les significations et les implications sur ses propres conduites. Cette évolution est permise par la nature des logiciels et le mode d'organisation des logiciels libres, puisque les améliorations suggérées et retenues peuvent profiter directement à l'ensemble des utilisateurs, le logiciel modifié pouvant être directement utilisable sans investissement supplémentaire. Mais au-delà de la condition technique, c'est bien un processus graduel et socialement réglé qui permet d'occuper le statut de contributeur au logiciel libre. Alors, cela semble naturel de faire profiter les autres de sa contribution personnelle quand on a soi-même profité du travail des autres développeurs. Paul explique : « J'ai commencé à l'utiliser, je crois comme beaucoup d'utilisateurs de logiciels

libres, c'est un truc qui est disponible, gratuit... En plus sympathique parce que, ce n'est même pas le fait que ce soit gratuit, c'est le fait que c'est ouvert, c'est-à-dire que s'il n'y a pas exactement les fonctionnalités que vous voulez dans le logiciel, vous pouvez les rajouter, le modifier, évidemment, ça paraît normal de faire profiter la communauté de... Si vous avez rajouté quelque chose qui peut être utile aux autres, ça paraît normal que... On le met dans le pot commun en fait, c'est évident ».

S'inscrire dans des groupes et devenir un professionnel reconnu

L'étape ultime du processus, parcourue par une minorité des contributeurs, consiste à devenir ce que l'on peut appeler un « professionnel » du libre, c'est-à-dire quelqu'un qui collabore au libre sur son temps de travail rémunéré, qu'il soit expressément et officiellement chargé de cette tâche, de manière exclusive ou non, ou bien qu'il parvienne à consacrer, de manière plus ou moins officieuse ou clandestine, une part significative de son temps de travail à cette activité. De ce fait, les « professionnels » du libre ont un engagement temporel plus important (tant en termes de durée que de stabilité) et constituent le « noyau dur » des communautés qui assure les fonctions de régulation, d'organisation, de structuration décrites précédemment.

Cette professionnalisation implique la participation à un projet de grande ampleur et porté par un groupe ou une institution solides ; elle passe par l'obtention d'une position professionnelle compatible avec un engagement continu et stable dans cette activité collective. De fait le cadre de l'emploi est varié : entreprises privées, entreprises publiques, administrations, universités, centres de recherche, etc. L'installation dans ces emplois peut résulter d'une transformation progressive du contenu d'un emploi existant permettant de consacrer une fraction croissante de son temps de travail au logiciel libre, ou de la recherche d'un nouvel emploi en adéquation avec la participation au libre, parfois après une période de chômage souvent bien indemnisée. Dans le monde marchand ce peut être le choix de travailler dans une SSSL ou la création d'une SSSL ou plus récemment un emploi consacré totalement ou partiellement au libre dans une entreprise informatique « classique ».

Cette professionnalisation n'est pas seulement une institutionnalisation. Elle correspond à l'acquisition de références symboliques partagées et à l'adhésion à des valeurs et croyances spécifiques, qui caractérisent ce monde social. Cet engagement dans le développement de logiciels libres est rétribué, mais il est aussi un engagement pour le logiciel libre. Ce sont

Le travail des développeurs de logiciels libres.
D. Demazière / F. Horn / N. Jullien

donc des convictions fortes qui motivent un engagement quasi-professionnel en faveur du logiciel libre, comme l'exprime par exemple Alain qui, après avoir travaillé dans une grande SSII classique s'est engagé dans une SSLL, et occupe actuellement un emploi dans une université où il consacre l'essentiel de son temps au logiciel libre : « disons, pour quelqu'un qui a un profil technique, le Libre c'est génial, parce que ça te permet d'avoir une prise sur la société. Tu peux avoir un rôle politique, tu peux essayer de changer le monde par le biais de quelque chose qui est dans ton domaine de compétence [...] Etre dans une association de logiciels libres, faire du Libre, c'est un moyen concret de faire changer les choses et te dire que tu n'es pas en train de gâcher ta vie quoi. Donc ça, c'est mon moteur principal. Je pense que c'est le moteur principal de pas mal de gens. ». D'un autre côté, l'hétérogénéité des positions des professionnels du libre suggère une différenciation de leurs parcours, de leur travail et des significations qu'ils lui attribuent. C'est ce point que nous allons examiner maintenant.

2.2. Des logiques d'engagement contrastées

Les développeurs de logiciels libres ont surtout été étudiés sous l'angle de la diversité de leurs motivations idéologiques (Blondeau, Latrive, 2000). Florent Latrive, dans une présentation d'un recueil des écrits des figures emblématiques du libre estime qu'ils constituent une « coalition improbable » composée de « néolibéraux, libertariens, tiers-mondistes, protomarxistes ». Le principal point commun semble être la volonté de défendre les libertés des utilisateurs de logiciels, et de promouvoir ainsi des usages individuels et collectifs spécifiques : « la liberté d'utiliser le programme, quel que soit l'usage ; la liberté d'étudier le fonctionnement du programme et de l'adapter à vos besoins ; la liberté de redistribuer des copies, et donc d'aider votre voisin ; la liberté d'améliorer le programme et de diffuser vos améliorations au public, de telle sorte que la communauté tout entière en bénéficie » (Stallman, 1998). Ces différents courants idéologiques se rejoignent également dans la lutte contre des monopoles au premier rang desquels Microsoft. En France, cette variété des justifications se retrouve dans l'existence de plusieurs associations de promotion des logiciels libres (APRIL, AFUL, FSF...), dans la vigueur des échanges (pas seulement techniques) qui circulent sur leurs listes et entre ces associations et dans les multiples manifestations rassemblant des publics nombreux où sont à la fois présentés des objets techniques (logiciels libres) et menés des débats souvent vifs.

Ainsi le monde social des logiciels libres n'est pas uniforme, et les carrières peuvent y suivre des chemins variés. Nous allons explorer cette diversité des parcours, et des significations qui y sont associées, à partir de l'exploitation de quatre entretiens avec des professionnels, choisis pour les contrastes qu'ils présentent, dans les positions occupées, les activités réalisées, les valeurs revendiquées, les croyances défendues, les réseaux d'appartenance. Il ne s'agit pas de dresser une typologie raisonnée, mais, dans une première étape d'identifier des lignes de différenciation parmi ceux qui ont atteint l'étape ultime de la carrière.

Une activité désintéressée équivalente à la recherche publique

Paul est un universitaire mathématicien. Son premier contact avec le logiciel libre provient de ses besoins d'un logiciel typographique permettant d'éditer des caractères mathématiques. Dès 1978, bien avant que la notion de logiciel libre apparaisse, un universitaire américain,

Knuth avait développé Tex, pour répondre à l'abandon de la typographie au plomb et à l'absence de logiciel de PAO de bonne qualité. Knuth a gardé une mainmise complète sur Tex (il est le seul à décider des évolutions et des corrections proposées qu'il intègre au code) mais comme il avait mis Tex dans le domaine public, de nombreux logiciels ont été créés à partir de cette base, dont le plus connu est LaTeX. LaTeX est un logiciel libre contrôlé par une équipe restreinte mais évolutive (principalement américaine au départ, elle est maintenant exclusivement européenne) et composée d'universitaires et d'informaticiens travaillant chez des éditeurs scientifiques. Paul est séduit par certaines fonctionnalités de logiciel, mais celui-ci étant anglophone, il est nécessaire de l'adapter aux particularités de la typographie française. Paul commence par effectuer quelques petits développements répondant à ce problème, et les montre à ses collègues, qui l'encouragent à les diffuser parmi les utilisateurs. C'est ainsi qu'il entre en contact avec le responsable de l'interface multilingue de LaTeX. Celui-ci est hollandais, ne parle pas français et est ravi d'avoir des contributions d'un francophone avec qui il noue une étroite collaboration. Progressivement Paul se retrouve à s'occuper des modules de francisation puis à développer d'autres modules. Cette activité lui prend de plus en plus de temps, auquel il faut ajouter une implication et une prise de responsabilité dans Gutenberg, association des utilisateurs francophones de Tex, qui publie les *Cahiers Gutenberg*.

Sa contribution à LaTeX est étroitement imbriquée avec son activité professionnelle. Elle s'effectue dans le cadre de sa profession d'enseignant chercheur mais également en dehors : « Est-ce que j'étais dans mon temps de travail ou dans mon temps de loisirs, c'est indécidable. Mais, après tout, même si c'est dans mon temps de travail, si c'est utile à la communauté, ce n'est pas plus inutile que les réflexions que je peux faire sur des maths. Je ne pense pas avoir volé l'Etat si j'en ai fait dans mon temps de travail. Et d'un autre côté si je l'ai fait dans mes loisirs, comme je l'ai fait avec plaisir, et qu'en contrepartie j'ai récupéré aussi tout ce que les autres ont fait bénévolement, je trouve que je ne suis pas volé ». Cette interpénétration, voire cette confusion, des temps de travail et de hors-travail a deux significations différentes mais convergentes. D'une part l'activité logicielle est une activité intellectuelle qui devrait relever du « domaine public », « exactement comme de la recherche [...] avec l'Etat qui paierait des universitaires ou autres, pour faire du développement de logiciels libres ». D'autre part son travail de développeur de LaTeX lui fournit une satisfaction et une reconnaissance quasi-

professionnelles qu'il définit comme plus « gratifiante » que la recherche en mathématique : « A la limite, je trouve plus gratifiant de faire un développement dont les gens se servent que de pondre un théorème dont personne ne se servira ou peut-être 30 ans après ma mort [...] Moi j'ai le plaisir, c'est vrai que quelquefois des gens me disent : Ah ! c'est vous qui avez fait ça ? Je m'en sers, je suis content de voir la tête que vous avez. ».

Paul défend donc un modèle de développement et de diffusion des logiciels libres qu'il qualifie de « convivial » et efficace parce qu'il permet de produire des logiciels de meilleure qualité, « les plus exempts de bogues, les plus stables ». Il l'oppose aussi très nettement à l'économie de marché, à laquelle la production et la distribution des logiciels devraient selon lui échapper, car il y voit la possibilité d'instaurer « d'autres rapports entre les gens. On vient me voir, on ne m'achète rien. Je peux rendre service et quelqu'un d'autre me rendra service. On peut dire que c'est une économie de troc, on peut dire ce que l'on veut. Mais c'est quand même plus sympathique ». Ses points de vue ne sont pas partagés par tous les membres de la communauté des utilisateurs de LaTeX : ainsi il s'oppose frontalement à un des responsables de Gutenberg qui voulait commercialiser une extension de francisation de LaTeX, ce que Paul considère comme « une trahison de l'esprit dans lequel on travaillait tous ». Il incarne une position de rejet catégorique du marché pour les logiciels et de refus d'utiliser les logiciels propriétaires, et se qualifie avec humour de « sectaire » : « je ne veux pas de ça chez moi. Y compris sur les machines que j'administre pour l'université. Si vous voulez du Windows, vous allez vous faire administrer ça par qui vous voulez, mais pas par moi. C'est contraire à mes principes. Je suis pour le logiciel libre, donc chez moi, il y a des logiciels libres. Si vous avez besoin d'autre chose, et bien allez voir d'autres [...] Donc, ça j'ai un côté un peu sectaire je reconnais, sur ce truc là ».

Une activité alternative, transposée dans les services marchands

Richard est un passionné de programmation informatique, à laquelle il s'est initié grâce à sa première calculatrice programmable, et à laquelle il s'est perfectionné lors d'une formation supérieure en informatique. Après des emplois d'informaticien classique dans des grandes entreprises, il crée en 1993, avec deux « commerciaux », une entreprise qui développait des logiciels « totalement propriétaires » qui permettait de rapatrier l'information des PDA de Newton sur des serveurs d'entreprises. Parallèlement, il suit le développement de Linux

(développeur Apple, il « achète un PC rien que pour ça, pour regarder ce que c'était ») et commence à utiliser Linux d'abord personnellement puis sur les serveurs d'entreprise.

L'événement qui va le faire basculer définitivement dans le logiciel libre est la décision d'Apple d'arrêter Newton au début de l'année 1998, condamnant l'activité de son entreprise : « le jour où Apple a arrêté Newton, je me suis dit : je ne travaille plus jamais pour un constructeur. Je ne travaille plus jamais sur du propriétaire ». Avec ses associés, ils décident de recentrer leurs activités de services sur les logiciels libres, et lancent ainsi une des trois ou quatre premières entreprises en France (et la seule qui existe encore de manière indépendante) focalisées sur cette activité. Ils effectuent deux types de travaux : l'installation de serveurs et l'administration système à partir de logiciels libres –ce qui les conduit à développer des « drivers » pour Linux-, et des développements de logiciels spécifiques pour des clients.

Richard, était gérant de cette société et fondateur d'un réseau d'entreprises basées sur le logiciel libre ce qui lui laissait peu de temps pour effectuer du développement. Toutefois, il continue de développer sur son temps libre un logiciel de vote électronique et de publication collaborative qui répond à des besoins internes à l'entreprise. Ce projet qu'il fait « pour [se] faire plaisir » prend de l'ampleur et il y consacre bientôt tout son temps, en étant rémunéré par les Assedic, après son départ de l'entreprise motivé par la chute de l'activité. Le logiciel commencé il y a deux ans bénéficie du travail de 7 ou 8 développeurs et une première version en logiciel libre va être publiée bientôt. Parallèlement, Richard a créé une nouvelle société qui commercialise des services autour de ce logiciel auprès d'associations (pour leur prise de décision), de collectivités locales (pour associer les citoyens) et d'entreprises. Même si ce n'est pas le marché visé prioritairement, le créneau actuellement le plus porteur est constitué par certaines entreprises qui l'utilisent pour développer de nouvelles pratiques d'évaluation des salariés basées sur la notation par les collègues.

S'il travaille dans le monde marchand où il a développé des activités de production et de distribution de logiciels libres, Richard se revendique d'un modèle productif alternatif. D'ailleurs il raconte volontiers son passé de militant d'extrême gauche, et considère le logiciel libre comme « un enjeu politique » : « c'est quand même la première ressource, le premier produit qui ne soit pas en voie de privatisation mais en voie de socialisation. On est entrain de privatiser l'eau, l'air bientôt, quand il sera pollué... Bon, là il y a un truc qui se crée, et on dit : voilà, ça appartient à la société ». Ses convictions politiques sont étroitement associées à

ses activités professionnelles, comme si elles étaient mises en œuvre, transposées, concrétisées. Ainsi ses deux sociétés sont la propriété des salariés (une SCOP et une SARL dont les parts sont détenues par l'association de salariés) et les salaires sont uniformes. Au-delà, il a favorisé la constitution d'un réseau d'entreprises liées au logiciel libre, partageant des valeurs identiques (notamment être détenue par ses salariés), et mettant en commun « toutes les informations, aussi bien comptables que financières qu'économiques, clients ». Cet échange et ce partage d'informations sont revendiqués comme une transposition au monde de l'entreprise du mode de fonctionnement du logiciel libre. Car, de même que Richard est convaincu que « le logiciel libre va supplanter les autres logiciels » en raison de l'efficacité de son mode de développement, il pense qu'un réseau d'entreprises détenu par les salariés constitue un modèle économique qui sera à terme gagnant par rapport aux entreprises classiques. Il en donne déjà pour preuve la plus grande résistance des entreprises de ce type à la récente crise traversée par les entreprises constituées autour des logiciels libres.

Une activité innovante, correspondant à un créneau commercial

Bernard a une approche différente du logiciel libre. S'il est également fondateur d'une société basée sur des logiciels libres, il insiste sur les similitudes avec des entreprises « classiques ». Très concerné dans son activité professionnelle initiale d'informaticien d'entreprise par les questions d'infrastructures réseaux, il assiste à l'essor d'Internet dont le « fondement même [...] est le développement du logiciel libre » : avec le succès d'Internet le monde informatique des infrastructures réseaux, bâti autour des éditeurs logiciels classiques propriétaires, a été conquis par le logiciel libre, et celui-ci, « créé pour faire du réseau », va envahir progressivement les différentes « couches » de l'informatique et « tout doucement gagner, par un processus viral, tout le système d'information des entreprises [...] et éjecter du marché le logiciel propriétaire », depuis le système d'exploitation en passant par le « middleware » et jusqu'aux applicatifs ou outils métiers (pour la gestion, le management...). Il en déduit une progression inéluctable de la diffusion des logiciels libres, et voit dans cette activité en émergence un créneau de développement à privilégier.

Mais il ne parvient pas à faire partager ses intuitions à sa hiérarchie : « ce sont des problèmes d'organisation classique d'entreprise, de jeu de pouvoir, de stratégie d'entreprise qui font qu'à un moment donné [...] vous êtes beaucoup trop en avance par rapport à ce qui se passe, et

que, non ça ne marchera pas ». Il retrouve par hasard d'anciennes connaissances avec lesquelles il avait fait de l'informatique dans un cadre non professionnel, qui partagent sa vision de l'évolution de l'informatique, mais se heurtent à la même incompréhension de leurs employeurs que ce soient des grands groupes ou des petites structures. Constatant qu'ils ont le même « état d'esprit créateur, de création d'entreprise, de satisfaction personnelle par rapport à la créativité, la nouveauté », ils décident de fonder, en 1999, une société basée sur le logiciel libre qui comprend aujourd'hui une dizaine de personnes. Leur activité principale est l'intégration système et les réseaux. Ils commercialisent des services en utilisant les nombreux outils logiciels libres existants. Ils participent aux communautés créées autour de ces outils en soumettant des « patches correctifs » et les modules logiciels qu'ils ont développés.

En parallèle à ces activités, l'entreprise a développé une plateforme logicielle qui permet de faire dialoguer l'ensemble des applications d'une entreprise quelles que soient leur fonction et leur statut (logiciel libre ou non). Cette plateforme est un logiciel libre et l'entreprise est « en train de discuter avec un autre groupe qui voudrait le reprendre pour aller plus vite et qui permettrait d'avoir une communauté un peu plus étoffée ». Bernard considère que « ce qui fait la force du logiciel libre aujourd'hui » c'est qu'il constitue « un nouveau mode de production du logiciel » : « les entreprises qui ne l'ont pas encore compris vont se trouver très très mal au fur et à mesure qu'on va avancer dans le temps, à savoir que ça revient à partager les coûts de RD qu'il pouvait y avoir dans le logiciel ; auparavant, il fallait mettre peut-être vingt développeurs en ligne pour obtenir un soft. Aujourd'hui, on a besoin que d'une personne, voire deux, sachant que vous avez la communauté qui bosse avec vous sur ce logiciel ». Si les groupes de production de logiciels libres fonctionnent de « manière informelle », ce qui « n'est pas du tout rassurant pour les esprits rationnels qui ne jurent que par les labels Iso 9001, 9002 », ils sont, selon Bernard, « plus innovants et plus efficaces » que les organisations classiques (« un bug est corrigé en trois jours en moyenne aujourd'hui »).

Il revendique son appartenance à la « sphère économique » du logiciel libre qu'il oppose à la « sphère philosophique » qu'il juge « sectaire ». Les discours sur le libre lui semblent contre-productifs par rapport aux entreprises clientes (« le libre, c'est vraiment un problème entre informaticiens ») et son attitude pragmatique le conduit à « insérer du libre dans des architectures propriétaires » ce qui a choqué les « purs du libre » (« on pactise avec le diable »). Les entreprises sont toutes équipées avec de nombreux logiciels propriétaires et

« on passe pour un rigolo si on arrive chez un client, pour lui dire « monsieur je vais tout changer, tout casser puis vous mettre du libre, vous allez voir c'est mieux ». La seule stratégie rentable c'est de « convaincre que le logiciel libre a un intérêt pécuniaire, fonctionnel [...], d'intégrer un petit bout de logiciel libre dans son architecture propriétaire, [...] et que on sache lui démontrer que petit à petit, on peut lui insérer un maximum de logiciels libres dans son architecture, dans son infrastructure réseau et informatique ».

Une activité porteuse, soutenue par un militantisme intense

Pascal a eu un premier contact précoce avec le code source d'un logiciel quand en première année à l'ENS, des élèves de deuxième année lui font découvrir un logiciel de jeu qui avait la particularité tout à fait étonnante à l'époque d'être livré avec son code source, ce qui permettait de comprendre comment le jeu avait été programmé. Il prend connaissance ensuite de Minix, un système d'exploitation développé par un universitaire et dont le code source était public. Minix est rapidement supplanté par Linux, auquel il s'intéresse immédiatement et qui lui fait prendre conscience de la force d'un « modèle réellement coopératif » par rapport au développement par un « individu isolé, quelque soit son talent et ses compétences professionnelles et intellectuelles ». Ses premières contributions à un logiciel libre s'effectuent dans le cadre de son premier emploi de chercheur en mathématiques. Dans son domaine de recherche avait été développé par une équipe de chercheurs et leurs étudiants une librairie logicielle d'algorithmes extrêmement avancés. Toutefois elle comprenait pas mal de bogues, et Pascal va proposer des corrections et développer des améliorations pour l'utilisation de la librairie.

En 2000, le succès d'un certain nombre de start-up qui étaient dans le domaine du logiciel libre l'incite à créer une société. Le retournement de conjoncture réduit les possibilités de mobiliser du capital-risque et c'est sous la forme d'une PME traditionnelle, qu'il fonde finalement une société qui comprend actuellement 15 personnes. A posteriori, il estime que c'est une bonne chose, car une croissance très rapide basée sur un financement externe aurait pu entraîner l'entreprise sur des options qui ne se seraient pas révélées viables à long terme. L'entreprise développe des applications pour des clients (en particulier les administrations) en utilisant un serveur d'applications libre, qui a lui-même été développé avec un langage de programmation également libre. Dans ce cadre, les salariés de l'entreprise proposent des

corrections et des contributions à la plateforme et au langage sur lesquels sont basés ses prestations et contribuent à les populariser. A partir des développements effectués pour les clients, l'entreprise a réalisé un « framework » qu'elle diffuse sous forme de logiciel libre. Autour de ce logiciel s'est créée une communauté d'utilisateurs et de contributeurs : entre 1000 et 2000 téléchargements du logiciel sont effectués chaque mois, plusieurs centaines de personnes sont inscrites aux « mailing-lists » et une trentaine de contributions externes ont enrichi la plate-forme.

En plus de son rôle de gestion de la société et d'animateur de la communauté créée autour du logiciel, Pascal a un engagement important et des responsabilités élevées dans une des principales associations de promotion des logiciels libres, dont il a été un des fondateurs. Cette association s'est créée dans la foulée de la première grande manifestation consacrée au « libre » en France et portant essentiellement sur l'intérêt économique du « libre », manifestation qu'il avait organisé avec d'autres personnes, et notamment des élèves de différentes ENS. Comme il l'explique, « l'enjeu au départ c'était de faire connaître quelque chose qui m'intéressait sur le plan technique, me passionnait même, et puis progressivement, c'est une activité professionnelle ». Ce travail de « popularisation du libre, d'évangélisation à l'attention de managers ou de décideurs [...] d'aider à contrer les attaques qui peuvent arriver contre le libre » est complémentaire de son activité professionnelle dans son entreprise qui « a intérêt à encourager le développement du libre à tous les niveaux ». Dans la même perspective de développement de son activité professionnelle, il revendique, et tente de légitimer, une approche pragmatique du client : il estime ainsi qu'après avoir effectué une percée significative dans le domaine des serveurs et du logiciel embarqué, le logiciel libre ne pourra s'imposer sur les postes de travail qu'en acceptant de s'intégrer à des logiciels propriétaires contre les partisans d'une utilisation exclusive des logiciels libres. Enfin, il critique les développeurs de logiciels libres qui ne pensent qu'à la perfection technique de ce qu'ils créent sans se soucier des besoins réels des utilisateurs. Ce faisant il ne désigne pas seulement les individus qui téléchargent les logiciels sur les serveurs, mais aussi, et probablement surtout, les institutions qui achètent des services autour de logiciels libres. Il se réjouit d'ailleurs de la progression des manières de faire du libre qui articulent « à la fois une approche business et une approche technique ».

Le travail des développeurs de logiciels libres.
D. Demazière / F. Horn / N. Jullien

Enfin, les parcours de Paul, Richard, Bernard et Pascal sont singuliers : au-delà de la diversité des processus selon lesquels ils s'investissent dans le développement de logiciels libres, ils attribuent des significations différenciées à cette activité, qui est menée dans des conditions biographiques et institutionnelles disparates. Chacun travaille à sa manière au développement des logiciels libres. Le partage d'un socle minimal de compétences (en particulier techniques), de croyances (en l'efficacité du travail coopératif), d'appartenances (à un même monde social qu'ils appellent « le libre ») n'efface pas les différences. Celles-ci suggèrent, ici encore, des variations importantes dans les formes concrètes que prennent les « communautés du libre », qui ne concernent pas seulement des règles de fonctionnement et des histoires collectives, mais aussi des parcours individuels et des inscriptions sociales.

3. Conclusion : des « communautés distantes » ?

Les personnes engagées dans le développement de logiciels libres utilisent systématiquement le terme « communauté » pour désigner le monde social auquel elles participent, et qui est un monde paradoxal puisqu'il est extrêmement ouvert et accessible via le réseau Internet et en même temps extrêmement sélectif et distinctif du fait des compétences requises. Nos premiers résultats empiriques permettent de conclure à une grande disparité des principes et règles d'organisation sociale de ces groupes d'une part, des ressorts et significations des appartenances d'autre part. Cette diversité englobe pourtant un problème commun : comment produire ensemble quand on est séparé ; comment fabriquer de la cohésion avec de la distance ? Si l'on privilégie les activités de production, le cas des logiciels libres met en évidence un travail spécifique, irréductible au télétravail ou au travail à distance de salariés d'une même organisation, caractérisée par la coopération entre travailleurs distants et affranchis de contraintes qui seraient issues d'une autorité extérieure au collectif constitué par la mise en réseau.

Nous avons entrepris d'en éclairer deux enjeux cruciaux. Le premier concerne la fabrication de la coopération. Nous avons identifié des mécanismes transversaux qui assurent un contrôle du travail et des travailleurs. Ils trouvent néanmoins des traductions différenciées, en fonction de l'histoire des projets et des groupes qui les initient et les développent. Le second concerne la production des engagements. Nous avons identifié des processus généraux qui construisent une carrière du développeur de logiciels libres. Et cette carrière s'inscrit dans des cheminements différenciés dans les biographies individuelles. Ainsi la réduction de la distance entre membres du groupe passe par des formes sociales multiples ; et symétriquement l'appartenance à un groupe de production passe par des liens sociaux multiples.

L'éclairage successif, mais séparé, de ces deux versants permet de renseigner la tension entre d'un côté l'activité collaborative et le sentiment d'appartenance (à un groupe, un monde, une communauté) qui résulte de cette participation, et de l'autre la distance relationnelle et l'individualisation des engagements qui résulte de cet isolement. Les constats dressés sont provisoires, mais il apparaît en tous cas nécessaire de croiser ces deux dimensions pour pouvoir dégager des figures distinctes du paradoxe que nous avons désigné par « communauté

distante » et identifier des segmentations du monde du logiciel libre organisées autour de formes organisationnelles et de mobilisations individuelles.

Références bibliographiques citées

- Becker H. S., 1963 (trad. 1985), *Outsiders. Etude de sociologie de la déviance*, Paris, A-M. Métailié.
- Becker H.S., 1982 (trad. 1988), *Les mondes de l'art*, Paris, Flammarion.
- Blondeau O., Latrive F. (eds), 2000, *Libres enfants du savoir numérique*, Paris, L'Eclat.
- Brooks F. P., 1995 (trad. 1996), *Le mythe du mois-homme : Essais sur le génie logiciel*, International Thomson Publishing.
- Di Cosmo R., Nora D., 1998, *Le hold-up planétaire : La face cachée de Microsoft*, Paris, Calmann-Lévy.
- FLOSS, 2002, *Free/Libre and Open Source Software : Survey and Study, Final Report*, <http://www.infonomics.nl/FLOSS/report>
- Foray D., Zimmermann J.-B., 2001, L'économie du logiciel libre : organisation coopérative et incitation à l'innovation, *Revue Economique*, 52, 77-93.
- Himanen P., 2001, *L'éthique hacker*, Exils
- Horn F., 2004, *L'économie du logiciel*, Paris, La Découverte.
- Hughes E.C., 1958, *Men and their work*, Glencoe, Free Press.
- Jullien N., 2001, *Impact du logiciel libre sur l'industrie informatique*, Thèse de doctorat en économie de l'Université de Bretagne Occidentale, 315 pages, http://www-eco.enst-bretagne.fr/Etudes_projets/RNTL/documents_universitaires.html
- Lang B., 1999, « Ressources libres et indépendance technologique dans les secteurs de l'information », *Technique et science informatique*, 18, 8, 901-914.
- Printz J., 1998, *Puissance et limites des systèmes informatisés*, Paris, Hermès.
- Raymond E. S., 1998, *La cathédrale et le bazar*, traduction de Blondeel S., http://www.lifl.fr/~blondeel/traduc/Cathedral-bazaar/Main_file.html
- Raymond E. S., 2000, « A la conquête de la noosphère », in Blondeau O., Latrive F. (eds), op. cit.

Le travail des développeurs de logiciels libres.
D. Demazière / F. Horn / N. Jullien

Strauss A., 1978, “A world social perspective”, in Denzin N (ed.), *Studies in Symbolic Interaction*, volume 1, Greenwich, JAI Press.

Thevenot L. 1997, « Un gouvernement par les normes. Pratiques et politiques des formats d'information », in Conein B., Thevenot L. (eds), *Cognition et information en société, Raisons Pratiques*, 8, 205-242.

Tönnies F., 1887 (trad. 1965), *Communauté et société*, Paris, Reitz.

Weber M., 1921 (trad. 1971), *Economie et société*, Paris, Plon.