Des relations de travail sans règles ?

Didier Demazière, François Horn, Marc Zune

Didier Demazière/François Horn/Marc Zune

DES RELATIONS DE TRAVAIL SANS RÈGLES ? L'ÉNIGME DE LA PRODUCTION DES LOGICIELS LIBRES

Résumé : Le développement de logiciels libres repose sur les engagements individuels et volontaires des participants. Ceux-ci réalisent des contributions potentiellement instables et désordonnées qui, pour déboucher sur la production d'un logiciel cohérent et efficace doivent être agencées et coordonnées. Cette configuration productive particulière délimite l'énigme du travail dans cet univers de production : comment des contributions sont produites, comment le projet perdure dans le temps, comment la qualité du produit est-elle assurée, comment les défaillances sont-elles gérées… ? L'analyse longitudinale et approfondie d'un projet qui a connu un succès rapide permet de montrer comment de telles communautés se structurent, avec l'émergence progressive de phénomènes de division du travail, de différenciation des positions, de sélection des contributions, de recrutement d'animateurs, de délégation de responsabilités. Cette organisation du travail présente cependant des traits originaux, du fait de l'absence de contractualisation des relations, de rétribution monétaire, de règle de subordination, de procédure d'éviction. Elle demeure en grande partie implicite, mais elle est nécessaire afin d'articuler deux objectifs partiellement contradictoires : mobiliser les contributeurs et susciter des investissements dans la fabrication du logiciel d'une part, maintenir la cohésion d'un collectif de travail en expansion et préserver l'identité du logiciel d'autre part.

Pour le sociologue du travail, la mise au point des logiciels libres, et plus largement leur diffusion et leur usage, est une activité sociale et collective qui intrigue. Il s'agit certes d'une activité productive, puisqu'elle consiste à fabriquer des biens qui sont destinés à autrui et rencontrent des usages, et, plus encore, qui ont un niveau de performance et de fiabilité élevés et entrent en concurrence avec des logiciels (dits propriétaires) commercialisés par des grandes firmes informatiques. Mais cette activité est assurée par des producteurs qui y participent de manière volontaire et bénévole, entretiennent des relations virtuelles médiatisées par le réseau Internet et ne partagent aucune inscription organisationnelle commune. Ce mode de production original ne présente aucune des caractéristiques attachées habituellement aux organisations productives : contractualisation des engagements, rémunération des contributions, coprésence des travailleurs, contrôle hiérarchique, etc. Plus largement, il est dépourvu d'instruments habituels de gestion des activités et de ceux qui les effectuent, tels que des procédures de prescription des tâches,

des systèmes de division du travail, des protocoles de sanctions. Il n'est pas même encadré par un dispositif juridique, relevant du droit du travail ou du droit commercial, si ce n'est une licence d'utilisation spécifique qui permet la libre modification et redistribution du logiciel.

Dans ces conditions, comment parvient-on, en quelque sorte malgré tout, à fabriquer un logiciel ? Comment celui-ci peut-il être le produit de multiples contributions individuelles et dispersées, et présenter une cohérence indispensable à son utilité ? Comment peut-il avoir une pérennité temporelle impliquant mises à jour et évolutions permanentes tout en étant le résultat de participations réversibles et potentiellement instables ? Apporter des réponses à ces questions suppose la réalisation d'enquêtes approfondies, permettant de décrire finement les modalités d'accomplissement de cette activité collective originale, et qui ne peut plus être considérée comme marginale ou exotique. Mais il s'agit, aussi, de faire avancer la théorisation sur les mutations contemporaines des activités de travail et sur l'émergence de nouvelles formes de coordination, irréductibles aux organisations formelles ou aux réseaux structurés, caractérisées par une faible formalisation, une grande plasticité et une efficacité productive notable.

SUR LA PISTE DES PROCESSUS D'ORGANISATION DU TRAVAIL

Ces spécificités sont d'une certaine manière revendiquées par les individus et groupes qui participent de ce monde de production original (Horn, 2004), puisqu'ils désignent celui-ci par le terme de « communauté », signalant ainsi le caractère volontaire des engagements individuels et la singularité des liens qui soudent les participants. Le point de vue indigène affirme et valorise le caractère alternatif de ce mode de production, qualifié de « modèle du bazar » (Raymond, 1998). En opposition au « modèle de la cathédrale », associé à l'industrie du logiciel propriétaire, il est fondé sur l'existence d'un réseau égalitaire de développeurs affranchis de toute organisation hiérarchisée et de tout contrôle centralisé. Il valorise la publication régulière de versions du logiciel, la mobilisation continue du réseau de contributeurs suivant la formule « distribuez vite et souvent, déléguez tout ce que vous pouvez déléguer, soyez ouvert jusqu'à la promiscuité ». Il s'agit, dans cette optique, de jouer sur un effet de masse de développeurs volontaires et non sélectionnés afin de provoquer une multiplicité d'opinions, d'intérêts, d'idées, et d'agendas. Ceci est à la fois défendu comme un principe d'efficacité

Un réseau égalitaire de développeurs affranchis de toute organisation hiérarchisée.

(« étant donnés suffisamment d'observateurs, tous les bogues [1] sautent aux yeux »), mais également de foisonnement d'idées par lequel se constitue un *bazar* « grouillant de rituels et d'approches différentes (...) à partir duquel un système stable et cohérent ne pourrait apparemment émerger que par une succession de miracles ». Mais la valeur heuristique de cette modélisation est contestable, même si on la considère comme un idéaltype. Car si un grand nombre de projets de développement de logiciels libres témoigne de l'effervescence et de la mobilisation autour de cette activité, la plupart n'aboutissent pas à la mise au point d'un logiciel opératoire [2] (Healy et Schussman, 2003). Aussi la mobilisation d'un nombre significatif de contributeurs, la fédération de leur activité autour d'un produit cohérent, le maintien de leur implication dans le temps, demeurent une énigme. Notre hypothèse est que le succès de certains projets de développement de logiciels libres ne résulte pas de quelque amalgame mystérieux, de quelque accident heureux, de quelque circonstance magique, mais s'explique par un ensemble de processus de mobilisation d'individus, de constitution de collectifs de travail, de définition de règles de coopération et de structuration des échanges.

La littérature relative au développement des logiciels libres reste très pauvre sur cette question notamment parce que les enquêtes empiriques approfondies sont rares. Une part importante des travaux porte sur la question des motivations des développeurs. Celles-ci relèveraient d'un utilitarisme direct (l'usage du logiciel pour ses propres besoins justifie le souhait de voir ses propres modifications intégrées, la confrontation à la communauté permet d'apprendre et développer ses compétences), ou indirect (l'exhibition des compétences agit comme un signalement vis-à-vis du marché du travail « réel »), ou encore une diversité de facteurs sociaux (le partage, l'amusement) (Ghosh *et al*. 2002 ; Lerner et Tirole, 2002 ; Elliot et Scacchi, 2004 ; Lakhani et Wolf, 2005). Pour d'autres travaux, la participation à des projets de logiciels libres serait avant tout l'expression des orientations idéologiques et normatives d'informaticiens férus de technique et engagés dans un combat pour la liberté de l'information et de la connaissance (des *hackers*) (Glass, 2003 ; Holtgrewe et Werle, 2001). De rares études empiriques centrées sur telle ou telle « communauté » montrent que la fabrication d'un logiciel s'appuie sur la réalisation de tâches différenciées (écriture du code [3],

1/ Un bogue (ou bug) est une anomalie dans un programme informatique l'empêchant de fonctionner correctement.

2/ Le site de développement collaboratif *SourceForge.net*, qui fédère très largement le monde du logiciel libre, héberge 117 000 projets. Ce chiffre est sans commune mesure avec le nombre, limité, de logiciels libres qui circulent et sont utilisés. De fait, la plupart des projets recensés sont inactifs ou en sommeil.

3/ Le code est le texte des instructions du logiciel, écrit dans un langage de programmation.

documentation, traduction, maintenance...) qui font invariablement l'objet de spécialisations (Basset, 2003). Cependant les mécanismes de distribution, d'attribution, d'affectation des tâches ne sont guère renseignés avec précision, de même que les glissements d'une division fonctionnelle du travail vers une hiérarchie, informelle ou structurée, des travailleurs.

Les analyses mettent aussi l'accent sur la mise en place de mécanismes de sélection, visant à préserver la qualité du logiciel, mais aussi sa cohérence et son identité. Cette régulation mixe des procédures d'évaluation en aval des contributions, qui ne sont pas automatiquement prises en compte, et des mécanismes de sélection en amont des contributeurs, qui ne sont pas systématiquement admis et intégrés dans le groupe (Auray, 2004 ; Conein, 2004 ; Weber, 2005). Mais si les mécanismes d'ouverture et fermeture sont décrits pour les gros projets procurant reconnaissance et réputation à ceux qui parviennent à s'y insérer, ils ne sont pas explorés pour des collectifs ayant à la fois une certaine taille et une moindre visibilité : ne se caractérisent-ils pas par une informalité plus grande et par un risque de défection et de fuite des participants qui est un enjeu aussi important que la gestion des entrants et de leur production ? De plus, les collectifs de développement de logiciels libres sont rarement considérés dans leur dimension processuelle, c'est-à-dire en tant que projet inscrit dans une dynamique, temporelle et sociale, d'évolution. Procéder à une telle mise en perspective génétique met sur la piste de nouvelles conceptualisations de ce mode d'organisation du travail. Celui-ci doit alors être appréhendé, c'est notre seconde hypothèse, comme une configuration en actes, non stabilisée, ce qui ne signifie pas sans ordre, et non planifiée, ce qui ne signifie pas sans continuité. Il doit être saisi comme un processus de reconfiguration réactive et adaptative, répondant aux problèmes successifs et aux enjeux renouvelés produits par sa propre histoire. L'analyse empirique doit donc prendre en compte la dynamique d'évolution de cette activité et la plasticité de son organisation, ce qui a des implications directes sur la démarche d'enquête et les options méthodologiques associées.

Le terrain empirique choisi est particulièrement pertinent pour l'analyse des processus d'organisation de la production car il concerne un projet qui est inscrit dans une dynamique de structuration et de réussite. Ainsi ce projet a dépassé le stade du démarrage et de la confidentialité. Il rassemble 300 développeurs (recensés sur le site de développement Spip-Dev) et plus de 2 000 utilisateurs (enregistrés sur le site Spip-User) disséminés dans plusieurs dizaines de pays. Il est particulièrement emblématique de l'idéal promu par les partisans du logiciel libre : il fonctionne grâce à l'apport d'individus dispersés

L'ENQUÊTE DE TERRAIN

Nous nous appuierons principalement sur l'étude approfondie d'une « communauté » de développeurs, cristallisée autour de la mise au point d'un logiciel de publication collective sur Internet : le logiciel SPIP. Nous avons combiné plusieurs méthodes d'investigation qui ont permis de collecter des matériaux variés : entretiens approfondis portant sur les contributions individuelles au logiciel, sur le fonctionnement collectif et sur les parcours biographiques, avec vingt-sept participants ayant des activités et des engagements différenciés (initiateurs du projet, codeurs [4] réguliers, contributeurs épisodiques, traducteurs, débogueurs…) ; observations directes de multiples réunions rassemblant des nombres variables de contributeurs (apéritifs et petits-déjeuners d'échanges, ateliers de travail, journées de rencontre autour du logiciel) ; analyse des messages électroniques échangés sur des mailing-lists, et sur des forums de discussion (IRC) en privilégiant des moments et conjonctures où des questions vives étaient en débat. L'enquête s'est étalée sur une durée de treize mois (entre février 2005 et mars 2006), pendant laquelle nous nous sommes immergés dans ce monde social. Nous nous y sommes progressivement intégrés, en procédant à plusieurs restitutions partielles de nos analyses en cours ou en multipliant les interactions plus ponctuelles avec nombre de nos interlocuteurs. La variété des relations nouées avec les enquêtés, la réciprocité ou le renversement des sollicitations, les échanges entre enquêtés sur notre enquête, les mises à l'épreuve qui nous ont été proposées, les allusions à peine voilées à ce que tel ou tel nous avait raconté, tous ces indices montrent assez combien ces collectifs structurés, typiquement, par des relations à distance et médiatisés par Internet, sont aussi marqués par l'interconnaissance et la rapidité des échanges. Dans un tel contexte, notre enquête a nombre de traits communs avec les démarches des ethnographes qui s'installent dans des communautés villageoises de taille restreinte.

et non reliés par quelque arrangement institutionnel quand nombre de projets sont pilotés de manière plus ou moins ouverte par des institutions ou entreprises (Demazière *et al.*, 2006) ; il est enraciné dans des orientations idéologiques contestataires ou anticapitalistes partagées par ses initiateurs et bénéficie d'une forte image dans les milieux associatifs alternatifs ; il est délibérément orienté vers les utilisateurs non informaticiens et privilégie pour cela une bonne accessibilité, servie par des interfaces commodes et des fonctionnalités simples.

En privilégiant une perspective génétique, nous examinerons successivement les enjeux relatifs au lancement du projet et à la constitution d'un premier groupe de production, puis au succès du produit et au nécessaire élargissement du cercle des contributeurs, enfin au maintien de l'identité du logiciel et à la gestion de la cohésion d'un collectif qui s'élargit. Ces trois éclairages permettront de renseigner les modes de régulation de la production logicielle : comment

4/ Le codeur écrit le texte du logiciel, i.e. les instructions du programme informatique.

s'ouvrir au-delà du cercle des initiateurs, comment mobiliser des investissements durables et de qualité dans le projet, comment faire face à l'inflation des sollicitations et des tâches issues du succès du produit, comment distribuer des tâches et même des responsabilités, comment réaliser des recrutements fiables et adéquats, comment gérer les défaillances, comment prendre en compte le droit de chacun au retrait, comment limiter le caractère hétéroclite des contributions quand les développeurs deviennent plus nombreux et dissemblables, etc. ?

LANCEMENT D'UN PRODUIT ET MOBILISATION DE PARTICIPANTS

Le démarrage d'un projet de développement de logiciel libre est antérieur à toute forme d'organisation du travail, dans le sens où ce n'est pas l'action collective qui fait le logiciel, mais plutôt l'inverse, le logiciel qui cristallise des engagements et fédère des contributeurs isolés et anonymes. À cet égard il est nécessaire qu'un individu ou un groupe d'interconnaissance aient enclenché une première production. Mais il est tout aussi indispensable à la réussite et au succès du produit que ce cercle initial soit renforcé par des contributions libres et volontaires. Cet enchaînement ne se fait pas automatiquement, et la plupart des projets en restent au stade de l'idée initiale, du produit incomplet et inutilisable, ou du logiciel confidentiel connu de ses seuls promoteurs. Il implique la mise en action de diverses opérations destinées à faire connaître le produit, acquérir une crédibilité, conquérir des utilisateurs, mobiliser des contributeurs, mais aussi gérer les échanges, coordonner les apports, favoriser les discussions, et, ce faisant, constituer progressivement un collectif, de taille variable, qui se reconnaîtra dans le produit et dont les membres se définiront comme une « communauté ».

L'ENRACINEMENT DANS UNE COMMUNAUTÉ D'EXPÉRIENCES

Le logiciel SPIP est déjà le résultat d'une histoire, de plusieurs années, pendant laquelle des liens forts vont se tisser entre quelques personnes qui seront les initiateurs du projet, et qui, auparavant, partagaient des expériences entremêlant expérimentations sur internet, orientations politiques et bricolages informatiques. Les initiateurs du projet participent au milieu des années 1990 à différentes activités collectives orientées vers l'auto-publication sur Internet. Ces

activités ont une composante technique puisqu'il s'agit de fabriquer des sites adéquats, permettant d'accueillir des textes et articles. Ce sont également des formes renouvelées de militantisme et de combat idéologique, car il s'agit de favoriser de nouvelles possibilités d'expression collective, en dehors des canaux médiatiques habituels, et de promouvoir ainsi un *« web indépendant »*, une *« expression libre »* [5]. L'objectif est de disposer d'un outil performant et simple d'utilisation pour gérer des publications collectives, de manière à contrecarrer la montée en puissance des industries de contenu qui risque de ravaler les internautes au rang de consommateurs passifs d'informations formatées.

Un outil performant et simple d'utilisation pour gérer des publications collectives.

Le projet SPIP s'enracine ainsi dans une communauté d'expériences ayant des traits spécifiques : elle s'appuie sur des engagements individuels volontaires, elle se traduit dans des conduites orientées en valeurs, et elle soude des minorités actives. Il prend peu à peu sens, sans avoir été anticipé, programmé, porté sur les fonds baptismaux : *« il n'y a pas de congrès fondateur dans l'arrière-salle d'un café [...].C'est juste parmi tous les projets, il y en a un, c'est SPIP. On va échanger nos bouts de codes »* (un des fondateurs). Ce processus d'émergence a des conséquences décisives sur les caractéristiques du logiciel SPIP. Bien que celui-ci ne soit pas le produit d'une planification qui en aurait fixé les traits, il est néanmoins doté d'emblée d'une identité affirmée et durable. Les initiateurs de SPIP indiquent unanimement combien ce logiciel incorpore un vécu commun, comme si celui-ci avait été transféré et traduit dans un objet technique : non seulement l'interface est simple de manière à faciliter la prise en mains, mais le logiciel est économe en ressources techniques de manière à pouvoir être implanté sur des petits serveurs. Ainsi le recours à SPIP est accessible à de petites associations qui ne disposent ni de webmestres expérimentés ni d'une infrastructure informatique performante. De même les principes politiques et éthiques défendus en matière d'auto-publication s'incarnent dans les fonctionnalités du logiciel et en particulier dans les principes retenus pour l'administration des sites : limitation du nombre de niveaux décisionnels, mise en place de forums accompagnant chaque article publié, soin apporté dans les fonctionnalités typographiques, gestion du multilinguisme, absence de dispositif d'identification et de suivi des utilisateurs, etc. Ces articulations entre le projet politique originel et le produit logiciel donnent à celui-ci une identité particulièrement forte, puisque fixée dans des choix techniques, et revendiquée par les initiateurs. Obtenir une diffusion large du logiciel

5/ Les mots, expressions et phrases entre guillemets et en italique sont extraits des entretiens réalisés.

implique de faire partager cette identité, par des utilisateurs non chevronnés et par des contributeurs prêts à participer à son développement.

UNE CRÉDIBILITÉ TECHNIQUE ET POLITIQUE

La phase de lancement du logiciel a été initiée par une campagne de communication, alors que le logiciel n'est pas encore tout à fait opérationnel. Un des initiateurs multiplie les interviews et chroniques sur Internet. Il en ressort quelques messages percutants, comme « fini les webmestres avec SPIP », qui donne de la visibilité au logiciel. De plus, le contexte concurrentiel est favorable car s'il existe d'autres logiciels de conception de site web, aucun ne présente la facilité d'appropriation pour l'utilisateur et pour le contributeur, qui est au cœur de l'identité de SPIP et du projet, politique et technique, de ses concepteurs initiaux.

La publication du logiciel va permettre la mobilisation d'un troisième personnage-clé, engagé dans les mêmes activités de publication de webzines [6], dont les compétences informatiques sont plus pointues que celles des deux promoteurs initiaux. Ses compétences sont mobilisées par l'un des initiateurs, webmestre d'un journal mensuel de gauche, afin d'adapter le logiciel à la gestion du site Internet de ce mensuel, pionnier dans la mise en ligne de ses articles. Le passage du site de ce mensuel sous SPIP va renforcer son image et sa notoriété. Jusque là SPIP avait rencontré l'intérêt et la sympathie des webmestres d'associations de gauche et d'extrême gauche, qui participaient aux mêmes réseaux et milieux que les initiateurs du produit. Cette connivence perdure, mais s'y ajoute une crédibilité technique issue de la capacité du logiciel à fonctionner en grandeur réelle avec un site ayant beaucoup de trafic, alors qu'il a conservé sa simplicité de fonctionnement : *« il tenait la charge, et ça c'était le gros, gros truc. C'était le problème que l'on avait [...] Pourquoi tout le monde faisait des systèmes compliqués avec du statique, c'est parce qu'on pensait que quand les gens viendraient visiter un site important le serveur tomberait [...] Et on a quand même bien pris la charge [...] et tout de suite ça crédibilisait »*.

Le logiciel suscite alors un intérêt nouveau de la part de programmeurs chevronnés : il n'est plus (seulement) issu *« d'une bonne volonté gauchiste »*, mais est porté par *« les technos, qui s'intéressent*

6/ Un webzine est un magazine édité sur Internet.

plus à la qualité du produit ». Le logiciel change alors de statut : de *« produit amateur, programmé par un crétin »*, il devient *« un produit qui tient la route »*. Cette crédibilité technique ne se substitue pas à l'image politique du logiciel, elle ne la concurrence pas, mais la renforce. Cela tient principalement au fait que le premier médium important qui utilise SPIP pour son site Internet est un journal classé nettement à gauche. S'y ajoute une conjoncture favorable, issue de l'effondrement brutal de la « nouvelle économie ». Car cet épisode contribue à dissocier l'Internet de ses usages marchands, à changer l'image d'Internet dans les milieux de gauche, à démultiplier les montages ou rénovations de sites Internet : *« c'est devenu un truc qui se faisait, c'était bien de faire son site soi-même. C'était bien puisqu'on avait perdu l'image des crétins de la nouvelle économie »*. Si le lancement de SPIP est enraciné dans une communauté d'expériences partagées par la poignée d'initiateurs, son développement est aussi articulé à des groupes partageant des valeurs et des orientations idéologiques. Cet ancrage dans des réseaux qui ne se réduisent pas à des communautés virtuelles est un facteur favorable à la diffusion du logiciel, et cela d'autant plus que la crédibilité technique, acquise plus tardivement, pourra s'y greffer. Mais il reste que la prise de la greffe exige aussi de mobiliser des contributeurs plus nombreux et prêts à s'investir durablement dans le projet.

Cette crédibilité technique ne se substitue pas à l'image politique du logiciel, elle ne la concurrence pas, mais la renforce.

DES ENGAGEMENTS DIFFICILEMENT MAÎTRISABLES

La crédibilité acquise par SPIP n'est pas une ressource suffisante pour la construction d'une « communauté » de développeurs, même si elle en est une condition. Car la cristallisation d'une telle collectivité suppose de mobiliser et enrôler des personnes qui vont contribuer à la fabrication – et à l'évolution – du produit, quand cette crédibilité génère plutôt une demande, une clientèle. La multiplication d'utilisateurs d'un logiciel libre est un stimulant à la mobilisation de contributeurs – et vice-versa – mais elle ne s'y substitue pas, même si un des enjeux réside dans la conversion de simples utilisateurs en contributeurs ponctuels d'abord, plus réguliers ensuite. L'appui sur des réseaux de militance politique ou associative est un atout pour enclencher cette dynamique, qui mobilise nombre de webmestres de ces milieux. Cette diffusion génère rapidement un afflux de questions, demandes de conseils, sollicitations d'avis, postés sur le site du logiciel, et adressés, de fait, à la – petite – équipe initiale. Les trois personnes qui la composent sont alors contraintes d'augmenter leur investissement, qui s'ajoute à des activités professionnelles mobilisatrices même si elles ne sont pas dépourvues de liens avec leur engagement bénévole : l'une est webmestre d'un

journal mensuel d'opinion qui adopte rapidement Spip, une autre effectue des missions de développement informatique avec un statut d'indépendant, la troisième est graphiste spécialisé dans l'édition d'ouvrages scientifiques, et une quatrième qui les rejoindra plus tard est enseignant-chercheur en informatique. De fait, la montée en charge, correspondant à une multiplication des utilisateurs s'accompagne ainsi mécaniquement d'une montée des charges pesant sur les initiateurs, et pilotes, du projet. Mais elle offre aussi des opportunités de sensibiliser les utilisateurs, de susciter des contributions, de les mobiliser en tant que développeurs. Elle peut alors se traduire par des demandes explicites et adressées à des interlocuteurs identifiés, portant sur des contributions mineures sur le plan du temps de travail correspondant, mais très utiles pour le développement du logiciel : déboguage, écriture de documentation, traduction dans des langues étrangères, mise à disposition des maquettes de présentation des sites réalisés.

Ce processus d'émergence d'une collectivité mobilisée autour du développement, de l'amélioration, de la mise à jour, de la diffusion, du perfectionnement, des logiciels libres correspond à une étape obligatoire, faute de quoi la différenciation reste trop marquée entre une minorité active, limitée au cercle des producteurs initiaux, et un cercle, de taille variable, d'utilisateurs. Car la production des logiciels libres récuse, précisément, la césure radicale entre producteurs et clients. Les initiateurs de SPIP soulignent tous la montée rapide d'un besoin et d'une évidence : *« on a besoin que les gens s'investissent, s'investissent de leur temps »*, *« c'est apparu évident qu'il nous fallait des gens qui s'engagent avec nous »*. Les termes utilisés sont particulièrement significatifs de ce qui est attendu : à la fois une mobilisation fondée sur une adhésion au projet et au produit et impliquant d'y consacrer du temps, et un engagement qui soit aussi un investissement placé dans une entreprise collective et au service de celle-ci. En ce sens, la figure qui est invoquée n'est pas seulement celle du militant ou du bénévole qui soutient gratuitement une cause, c'est aussi celle du travailleur qui apporte sa pierre à une production collective. Ces travailleurs requis sont d'ailleurs caractérisés normativement par des traits relativement précis sinon classiques : ils doivent être *« en mesure de se débrouiller »*, *« capables de prendre des initiatives »*, mais aussi *« fiables »*, *« capables de tenir dans le temps »*.

Le contributeur idéal est par conséquent un travailleur autonome et fidèle, ce qui signifie que son engagement individuel ne doit pas être le produit d'un travail de mobilisation, mais qu'il est conçu - rêvé - comme la conséquence automatique d'une adhésion au projet. Cependant, il s'agit bien d'une figure rêvée, voire mythique, dans la

mesure où la conversion d'utilisateurs en contributeurs est marquée par l'incertitude, et où les engagements et investissements des développeurs ne sont pas si prévisibles. Les propos tenus par les membres du noyau qui pilotent le projet sont émaillés de commentaires exprimant ces écarts : *« mon impression c'est que les gens ils ont peur de s'investir [...] ils demandent tout de suite à être confortés »*, *« il y a un manque d'autonomie là-dedans, d'autodétermination, qui est flagrant »*, *« même ceux qui font, ils sont toujours à s'excuser d'être là, c'est très étonnant »*, *« beaucoup ne font que passer, ils sont enthousiastes, et puis, pfft, ça s'épuise vite »*.

Le terme de « communauté » ne rend pas compte de ce caractère gradué des participations.

Dès que le cercle des contributeurs s'élargit au-delà du périmètre, étroit, des initiateurs, le travail de production du logiciel devient plus hétérogène, mêlant des travailleurs historiques dont la stabilité est nécessaire à la réussite de l'entreprise, et des intermittents qui fournissent un travail peu prévisible, mais aussi des collaborateurs plus réguliers qui produisent des contributions attendues. Le terme de « communauté » ne rend pas compte de ce caractère gradué des participations, qui n'est pas contradictoire avec une identification partagée au produit. Surtout la question se pose de savoir comment ces différenciations sont appréhendées, prises en compte, organisées et maîtrisées, de manière à piloter dans la durée une production collective du logiciel libre.

STRUCTURATION DE LA PRODUCTION ET FIABILITÉ DES PARTICIPANTS

L'accroissement du nombre des utilisateurs et des contributeurs provoque une augmentation, qui peut être très rapide, des flux informationnels à traiter : demandes et sollicitations d'utilisateurs peu autonomes ou curieux, suggestions de développement et lignes de programmes adressés aux initiateurs du projet. Ceux-ci y répondent par une différenciation plus précise des tâches, mais aussi par des délégations de responsabilités qui introduisent des formes variées de hiérarchie interne. Ce mouvement fait émerger des questions classiques dans les mondes du travail, mais qui appellent ici des réponses spécifiques compte tenu de l'absence de contractualisation des relations de travail : comment identifier les compétences des travailleurs, sur quelle base recruter ceux à qui des responsabilités seront confiées, comment évaluer de manière anticipée la longévité des engagements dans l'activité, comment gérer les erreurs de recrutement, comment procéder à l'éviction des défaillants ?

DIFFÉRENCIATION INTERNE ET DÉLÉGATION

Le déséquilibre entre un faible nombre de contributeurs ayant un investissement important et un grand nombre de contributeurs ayant un investissement limité s'auto-entretient, et même a tendance à s'aggraver, puisque les contributions des seconds sont adressées aux premiers, qui doivent les gérer et y répondre. D'une certaine manière la constitution d'une collectivité active provoque un épuisement du noyau, par accumulation de tâches. Cette tension est d'autant plus sensible que la dynamique collective est largement dépendante de la réactivité à ces demandes. Car fournir des réponses rapides n'est pas considéré comme une activité mineure, mais a un caractère stratégique, comme l'indique un des fondateurs : *« animer la communauté c'est surtout répondre à des mails. Et ça c'est quand même une activité qui prend un temps absolument fou dès que le système a une certaine notoriété »*. La délégation de certaines activités s'impose comme une nécessité à mesure que grossit le flux d'informations et sollicitations remontant vers les initiateurs qui pilotent le projet. Elle concerne des échanges avec des utilisateurs et des développeurs sur des questions réputées peu cruciales mais pourtant chronophages : répondre à des demandes d'aide, donner des conseils, donner un avis sur une contribution, évaluer une traduction, proposer une documentation, etc. Elle suppose de s'appuyer sur des animateurs, dont la tâche est de susciter l'entraide mutuelle, de stimuler des contributions, de lancer des projets, d'organiser les échanges, de structurer des discussions, de réguler les interactions, voire de trancher dans les désaccords. Ces activités acquièrent une reconnaissance officielle dès lors qu'elles débouchent sur la différenciation de listes de discussion et de sites spécialisés sur certains aspects de SPIP, dont la gestion est confiée à ces animateurs, qui en deviennent les administrateurs dotés de droits spécifiques. La délégation de responsabilités s'accompagne ainsi d'une sectorisation du développement du logiciel et d'une multiplication des espaces de travail (cf. encadré).

La double dynamique de structuration d'un cercle d'animateurs-administrateurs et de spécialisation des activités présente plusieurs avantages. Distinguer des espaces de travail spécialisés permet de réduire l'hétérogénéité des participants, qui croit rapidement avec leur nombre. Car une trop grande disparité dans les attentes et les compétences a des effets démobilisateurs : les préoccupations d'un novice en informatique et d'un expert des langages de programmation ont peu de points d'intersection, et leur participation à une même liste de discussions risque de cantonner les échanges à des

LES ESPACES DE TRAVAIL DE SPIP

Le projet SPIP est organisé en espaces de travail et d'échanges spécialisés, dont le nombre croît avec le temps :

– le développement technique : outre la liste générale et historique, *Spip-dev*, sont apparus *Spip-contrib* (proposition d'ajouts de fonctionnalités optionnelles), *Spip-lab* (réflexion technique en matière de R&D), *Spip-zone* (site collaboratif de développement du logiciel et de plug-ins [7]).

– l'animation collective avec les sites *Spip-mag*, *Spip-party* (coordination des rencontres « réelles » entre contributeurs) et *Spip-Blog*.

– la documentation et la traduction avec notamment *Spip-trad* (gestion des traducteurs – couvrant une cinquantaine de langues – et support aux utilisateurs non-francophones).

– l'aide avec les listes *Spip-user* (gestion des demandes d'utilisateurs) et *Spip-Forums*.

polémiques peu productives (des *trolls* [8]). La spécialisation favorise le rassemblement de pairs, ou quasi-pairs, dans des espaces de travail bien différenciés. Par ailleurs ce mode de structuration a aussi une fonction de rétribution symbolique de certains travailleurs, puisque ceux qui sont désignés comme administrateurs de ces espaces bénéficient ainsi d'une reconnaissance de leur travail, de leur rôle, de leurs compétences, et se voient octroyer par délégation un pouvoir de régulation dans ces espaces. Cette différenciation des positions, qui se caractérise par la formation d'un cercle intermédiaire entre le noyau stratégique et les cercles périphériques (Demazière *et al*. 2006), est aussi un levier de mobilisation des participants les plus engagés, un moyen de stabilisation de ces fidèles, toujours exposés à la lassitude, un instrument pour officialiser la valeur de leur apport. Les rôles ainsi distribués ne se cristallisent pas dans des échelons hiérarchiques et n'alimentent pas des relations de subordination. Pourtant ils peuvent être considérés comme des quasi-statuts, et sont souvent vécus comme extrêmement valorisants. Mais ils définissent d'abord des responsabilités vis-à-vis du devenir du projet. C'est pourquoi la question du repérage des animateurs potentiels et de leur recrutement est un enjeu si important.

Formation d'un cercle intermédiaire entre le noyau stratégique et les cercles périphériques.

7/ Un plug-in est un module optionnel que l'utilisateur peut choisir d'installer avec le logiciel. Ceci permet de pouvoir bénéficier de fonctionnalités supplémentaires sans alourdir le logiciel de base.

8/ Un troll est une polémique initiée sur un forum de discussion avec l'objectif de générer des conflits entre les participants.

LES DIFFICULTÉS DU RECRUTEMENT

Les façons de recruter et la production de jugements de compétences sur les travailleurs sont marqués par de fortes incertitudes (Eymard-Duvernay, Marchal, 1997), car si les épreuves de jugements s'appuient sur des instruments d'évaluation multiples, l'interprétation des résultats et des indices qui en résultent demeure problématique (Bureau, Marchal, 2006). Ces difficultés sont décuplées dans le cas de la mobilisation de contributeurs à la production de logiciels libres, notamment parce que l'absence de codification de la prestation (absence de rémunération, de contrat, de règle juridique...) rapproche celle-ci des participations à des actions bénévoles faiblement institutionnalisées, volontaires et révisables à tout moment. Le contrôle social des engagements pesant habituellement sur les participants est d'ailleurs d'autant plus faible que les interactions sont médiatisées par le réseau Internet, que l'interconnaissance est très limitée, que le recours à des pseudonymes ou diminutifs est fréquent. La défection est donc peu coûteuse, et nombre de participations sont, de fait, ponctuelles, irrégulières ou erratiques. Mais les contraintes de l'activité sont différentes pour les animateurs et administrateurs, car leur fonction exige une certaine stabilité de leur participation. Et c'est pour cela que leur mobilisation, et en amont leur repérage par les membres du noyau stratégique, peut être considérée comme un protocole de recrutement. Ce recrutement est particulièrement délicat, puisque le caractère distant des relations complique la construction du jugement.

Une caractéristique constante de ce recrutement est qu'il prend la forme de promotions, dans le sens où les personnes concernées participent préalablement à la production du logiciel. C'est cette participation qui informe sur l'engagement et sur la compétence, c'est le travail réalisé qui fournit des indices de jugement. Pour autant la délégation de responsabilité ne se réduit pas à la reconnaissance de ce qui a été déjà fait ; car il ne s'agit pas de distribuer des *« bons points »*, des *« sucettes »*, des *« médailles »*, mais d'engager quelqu'un sur qui compter à l'avenir, de faire un pari raisonné sur un travail futur. L'évaluation doit s'accrocher à des activités constatées, et la capacité à avancer dans la concrétisation d'un projet semble ici décisive. Ainsi plusieurs des animateurs de différents espaces de travail font le récit de cheminements parallèles : ils ont eu une idée susceptible d'améliorer l'organisation du travail, ils l'ont exposée aux membres du noyau, ils ont eu en retour une incitation à la mettre en pratique, ils ont montré leur capacité à le faire, et finalement ils ont été intronisés comme administrateurs du nouvel espace de travail : *« au début on m'a dit'si tu veux participer il y a une liste de trucs*

à faire'. J'ai commencé à regarder un peu, mais le code de SPIP, c'est un peu hermétique pour moi [...] Et puis j'ai eu l'idée de monter un site pour les contrib., parce que ce n'était pas bien organisé [...]. Et donc on nous a dit, parce qu'on était deux à avoir cette idée, "et ben allez-y, faites-le". Et alors on a avancé, et on l'a fait. [...] et alors c'est devenu officiel, l'espace pour les contrib., et c'est comme ça qu'après on est devenus les administrateurs ». La capacité à faire, et à réussir un projet, est un indice important dans l'évaluation, même si celle-ci ne préserve jamais du risque de défection soudaine et imprévisible de personnes jusque là fortement impliquées dans la conduite de projet et la coordination d'espaces de travail.

Or une des qualités attendues des administrateurs est leur capacité à tenir le rôle dans la durée, car leur stabilité, ou plus exactement celle de leur engagement et de leur investissement, est une ressource pour la réussite de l'action collective : « *c'est difficile de juger, alors c'est vrai que de repérer des gens très actifs, très présents, qu'on connaît au bout d'un certain temps, qui ne font pas que passer. C'est un moyen de pas trop, enfin, en fait on ne sait jamais je crois* » (un des fondateurs). En effet, l'anticipation des comportements d'autrui est irrémédiablement fragile. Et si l'observation de l'activité antérieure peut fournir des indications, il faut trouver le bon compromis entre un temps d'observation suffisamment long pour minimiser les risques de défaillance après l'accès au cercle des administrateurs et une mise à l'épreuve suffisamment courte pour minimiser les risques de défection précoce pour défaut de reconnaissance, ou par découragement. Aussi la rencontre directe supplée dans nombre de cas l'incertitude inhérente au jugement à distance, d'autant qu'elle permet d'introduire un nouveau registre d'évaluation. Plus les positions concernées sont proches du noyau stratégique, plus le recours à des interactions face à face est systématique. Ces contacts directs permettent de vérifier une convergence de points de vue sur les orientations du projet logiciel, une adhésion aux principes de fonctionnement collectif, un partage de l'attachement à l'identité du logiciel, une proximité des significations associées à la participation, bref un ensemble d'affinités en valeurs, sinon une communauté d'expériences comparable à celle qui caractérise le petit groupe des initiateurs : « *c'est important de pouvoir avoir un peu plus de profondeur dans la relation, pour voir un peu si on a la même vision des choses, la philosophie de SPIP du moins. C'est un peu tout ce qui est derrière, même si on n'a pas besoin d'être d'accord sur tout, mais dans les grandes lignes, oui. Sinon on va aller au conflit avec des personnes qui ont des responsabilités, et ça va rejaillir sur toute la communauté* » (un des fondateurs). La rencontre directe est donc un moyen d'informer l'engagement dans le projet des possibles administrateurs, non plus à partir des traces concrètes et

matérielles de cet investissement telles qu'enregistrées dans les sites, forums de discussion, listes de messages, mais à partir de traces symboliques et immatérielles, renvoyant aux significations subjectives de cet investissement.

La délégation des responsabilités ne s'effectue pas seulement à partir du noyau stratégique, mais elle est aussi initiée par des administrateurs d'espaces spécialisés qui, eux aussi, sont débordés par leurs tâches. Dans ce cas, c'est l'intensité et la qualité de l'activité réalisée dans l'espace de travail considéré qui fonctionnent comme signal : *« j'avais moins le temps, alors quand j'ai vu qu'il avait proposé un certain nombre de choses que, il y avait quatre ou cinq articles, alors je me suis dit, ce gars il traduit bien. Alors je lui ai donné les droits d'administration de la zone en espagnol »*. La procédure d'évaluation qui sous-tend le recrutement est ici faiblement outillée, et le risque de faire de mauvais recrutements semble élevé : *« les gens attendent d'être administrateurs, et puis en fait quand ils deviennent administrateurs, ils ne font plus rien parce qu'ils ont le privilège et puis ça leur suffit. Il y en a plein des comme ça, il y en a même majoritairement des comme ça »*. C'est dans les espaces de travail organisés autour de l'activité de programmation, d'écriture du logiciel, que ces erreurs de recrutement semblent les plus fréquentes, parce que la reconnaissance du travail fourni est, ici plus qu'ailleurs, susceptible d'être utilisée et reconvertie dans la sphère marchande : *« On a fait des mauvais recrutements, des mauvais choix, on ne sait pas, il y a des gens qui étaient admin et puis en fait ça n'a servi qu'à favoriser leur carrière personnelle ou professionnelle. C'est juste une carte de visite, donc pas très intéressant. »*

LA GESTION DES DÉFAILLANCES

Les défaillances correspondent à des cas de figure où le travail attendu n'est pas – ou mal – réalisé, ce qui indique que cette catégorisation est produite par des personnes qui ont la légitimité de juger, et donc qui occupent dans le collectif de projet une position plus centrale que celle de la personne évaluée. Être défaillant c'est ne pas remplir ses engagements, lesquels sont plus implicites et moins impératifs à mesure que l'on passe du noyau stratégique au cercle intermédiaire et aux zones périphériques. En pratique, la défaillance est tolérable quand le travail est distribué entre un grand nombre de personnes qui contribuent chacune de manière limitée à la production collective. La question de la défaillance se pose donc pour ceux qui occupent des positions de responsabilité. En l'absence de rapports contractualisés ou salariaux, les instruments de sanction

sont rares, et l'éviction, sorte d'équivalent du licenciement dans les collectifs d'engagements volontaires, est un levier peu opératoire car dangereux : du fait de la faible institutionnalisation de l'action organisée, elle risque de détériorer la cohésion interne, indispensable à la production collective du logiciel. Pour les administrateurs, même les conduites les plus opportunistes, jugées déviantes par rapport aux règles, implicites, attachées à la participation et a fortiori à la prise de responsabilité, sont délicates à sanctionner : *« il est arrivé une fois que, des gens qui se sont vus admins, ils ont commencé par s'autovalider, enfin valider leurs contribs à eux. [...] On n'a pas gueulé mais, on a dit c'est pas le but du jeu. Devenir administrateur pour valider ses contrib., c'est un peu puéril »*. Les sanctions effectives, tel le retrait de droits d'administration, se sont souvent révélées inefficaces, en raison du caractère distribué des prérogatives : *« on a dit'tiens toi tu fais rien donc on t'enlève tes droits' (...(et puis il a dit'ça sert à rien' [...] C'est-à-dire, ça sert à rien de lui retirer les droits. [...] Il avait retrouvé un copain parmi la trentaine d'admin qui lui avait redonné ses droits. [...] Parce qu'un admin peut transformer quelqu'un en admin »*.

La gestion des défaillances passe rarement par la sanction directe des défaillants, mais utilise deux autres leviers.

Aussi la gestion des défaillances passe rarement par la sanction directe des défaillants, mais utilise deux autres leviers. Le premier vise à redynamiser un espace de travail dans lequel les administrateurs sont faiblement investis, en faisant l'annonce publique d'un retrait, manifestant un ras le bol face à la charge de travail et destiné à provoquer une remobilisation des animateurs défaillants. Le second levier, plus indirect encore, consiste à créer de nouveaux espaces de travail, dont les fonctions sont proches, mais pas similaires, à un espace existant qui est déréglé ou n'apporte plus de contribution suffisante au projet. L'objectif est d'ouvrir un nouvel espace d'engagement, quitte à ce que cela se fasse à partir d'un écrémage des participants défaillants, qui par le jeu de la multiplication des nouveaux espaces sont repoussés, de fait, dans des positions plus périphériques, moins stratégiques : *« une idée pour essayer de bouger la communauté, c'est Spip-zone [...] avoir en tout cas un endroit technique où les gens peuvent committer*[9]*, faire évoluer des choses, et par exemple des grosses contribs, qui sont en train de végéter dans l'espace privé de Spip-contrib »* (un fondateur d'un nouvel espace de travail).

Ces observations montrent le caractère fluide des relations de travail au sein des collectifs de production de logiciels, et indiquent que leur fonctionnement ne peut être retracé sans une mise en perspective temporelle. Car, ces actions organisées étant dépourvues des

9/ Committer : effectuer un commit i.e. une modification qui est intégrée dans le logiciel.

ressources mobilisables ordinairement pour la gestion des organisations de travail, elles sont l'objet de reconfigurations continues, qui en modifient la structuration interne et les contours externes. Compte tenu de ces processus, se posent avec acuité les questions du maintien dans le temps de l'identité du logiciel et de la consolidation de l'action collective qui le produit.

MAINTIEN DE L'IDENTITÉ DU PRODUIT ET ADHÉSION DES PARTICIPANTS

L'accroissement du nombre de développeurs et utilisateurs est une condition pour le succès du logiciel mais constitue également une menace, au moins potentielle, sur l'identité du produit et sur la cohésion d'un collectif en croissance. L'apparition de cette tension appelle des ajustements dans l'organisation interne : elle conduit en particulier à une intensification de la sélection des contributions, ce qui accroit le risque de démobilisation des contributeurs dont le travail n'est pas pris en compte, voire le danger de coalition de développeurs insatisfaits des décisions prises autour d'un projet alternatif. La prévention de ces risques consiste, notamment, à affirmer et maintenir une identité forte au logiciel, en assurant un pilotage stratégique clair et explicite du projet. Elle dépend aussi de la capacité à faire partager à tous les participants, quel que soit leur rôle, l'identité du produit, de sorte que celle-ci devienne un véritable ciment qui consolide le collectif.

ÉVALUER LE TRAVAIL ET SÉLECTIONNER LES PRODUCTIONS

La fiabilité du produit repose en effet sur la mutualisation rapide des résultats d'un grand nombre de tests en situation d'usage.

Un des rôles majeurs des utilisateurs du logiciel, est de signaler les bugs, et, si possible, de proposer des ajustements du code permettant de les résoudre. La fiabilité du produit repose en effet sur la mutualisation rapide des résultats d'un grand nombre de tests en situation d'usage. Si les corrections de bugs sont prises en compte, de manière quasi-automatique, d'autres contributions sont évaluées et font l'objet d'une sélection. Ce sont celles qui sont destinées à enrichir le logiciel, à le compléter, à ajouter des fonctionnalités. Ces contributions ont un tout autre statut : elles risquent d'affecter l'identité du produit, et sont, la plupart du temps rejetées par les leaders. Ceux-ci constituent un petit noyau formé par les trois fondateurs de SPIP, rejoints par des nouveaux membres qui ont apporté des contributions importantes, et qui adhèrent sans restriction aux principes sous-jacents au logiciel et partagent des orientations idéologiques

proches. Ce noyau a le monopole de la validation des contributions et des droits d'écriture. Le rejet est souvent argumenté au motif que les contributions concernent des usages spécifiques et des applications particulières : *« ce sont des grosses modifications qui n'intéressent qu'une fraction des utilisateurs »*. Parce qu'elles sont importantes et exigent un travail de conception conséquent, ces contributions sont souvent proposées par des entreprises de services informatiques qui ont fait des développements pour un client et qui demandent leur intégration dans SPIP. Le refus est alors systématique : *« une entreprise qui développe du SPIP, elle le développe pour ses clients et des besoins très spécifiques [...] nous on s'enquiquine systématiquement à généraliser tous les problèmes et donc les rapports deviennent assez rapidement tendus parce qu'une bonne idée pour un client n'est pas forcement une bonne idée pour les autres »* (un des fondateurs).

Pourtant des contributions jugées *« significatives et importantes »*, c'est à dire ayant des conséquences sur les caractéristiques et l'identité du logiciel, sont parfois prises en compte par le noyau stratégique, mais c'est alors au terme de longs échanges avec le développeur, de négociations sur le code proposé, de réécriture du programme, et même d'une validation préalable de l'idée, bref d'un processus de coproduction, comme l'indique un développeur : *« sinon, après il y a eu d'autres contribs que j'ai faites qui sont mieux passées. Il y a notamment eu les "champs extra". C'était moi qui étais à l'origine de ça à l'époque. Donc j'avais codé une première passe pour dire, montrer un peu la faisabilité, voire un peu si ça permettait d'aller loin ou pas, là dessus [un des membres du noyau] avait dit qu'effectivement c'était pas mal mais après, ce qu'il fallait, c'était l'intégrer [...]. Là on a commencé à avoir un dialogue entre lui et moi pour dire d'amener le projet jusqu'au bout. Et puis après, il l'a intégré dans SPIP mais, il l'a pas mal recodé en passant »*. Ce type de contribution est assez rare, et n'est jamais accepté d'emblée, car sa compatibilité avec les développements envisagés ou engagés par le noyau stratégique doit être préalablement vérifiée. Cela suppose une concertation et une collaboration avec des membres du noyau, ce qui correspond à un long processus d'ajustements rendant la prise en compte de la contribution *in fine* évidente et indiscutable. Ce processus est particulièrement visible dans un cas, qualifié par certains de *« parcours du combattant »*, par d'autres de *« contre-exemple »*, qui concerne une contribution *« majeure »*, proposé par une personne qui a ensuite été intégrée au noyau stratégique et est devenue, au terme de ce processus, le *« quatrième mousquetaire »*. À partir du moment où celui-ci propose un compilateur, s'engage une série d'échanges avec des membres du noyau qui s'étale sur une année, pendant laquelle il fait évoluer sa contribution, travaille avec acharnement, propose

Long processus d'ajustements rendant la prise en compte de la contribution *in fine* évidente et indiscutable.

six versions successives, ce qui fait dire à un des animateurs du site de contributions *« je crois que pendant un an en fait, enfin il a eu le courage et la force de re-proposer à chaque fois sa contrib et, voilà, pour finir ça a été intégré au noyau »*. Finalement, à partir du moment où le principe de sa contribution est retenu, il travaille plus de six mois avec un des membres du noyau pour réécrire le code proposé, *« la tête dans le guidon »*, *« comme des dingues »*.

Différents mécanismes conduisent à ce que le noyau stratégique exerce une sorte de monopole sur le développement : validation des contributions mineures, contrôle de l'accès en écriture, implication dans la production des contributions externes jugées significatives. La différenciation entre ce noyau et l'ensemble des autres participants apparaît ainsi très tranchée, même si le noyau n'est pas un cercle fermé puisque de nouveaux membres y sont intégrés. Ce clivage apparaît comme un moyen de préserver l'identité du logiciel dans un contexte de croissance et de diversification des participants. Mais il conduit à poser le problème, conjoint, de la cohésion du groupe.

LES CIRCUITS DE DÉCISION ET DE DISCUSSION

La sélection des contributions, ou le monopole des droits d'accès en écriture, ne sont pas des leviers suffisants pour maintenir l'identité du logiciel produit, car celle-ci est également dépendante de l'adhésion de membres à un produit et à un projet. En ce sens, la fabrication d'une « communauté » est une dimension aussi importante que la fabrication d'un logiciel. Une part non négligeable de la participation se traduit dans des échanges, débats, discussions (sur *les mailing list*, les forums, le *chat*...), de sorte que le travail réalisé ne se réduit pas à l'addition de contributions individuelles, mais que sa signification réside aussi dans sa dimension délibérative. La mobilisation et la participation sont composites : à la fois du débat et de la délibération, et en même temps de la production et de la fabrication, ce que certains participants résument en différenciant les *« trolleurs »* et les *« codeurs »*. Les « communautés » sont traversées par cette tension (qui peut aussi cliver les individus eux-mêmes), et la recherche de compromis et d'équilibres est indispensable au maintien de ces collectifs.

Dans le cas de SPIP la fabrication de la cohérence du logiciel et de la cohésion du collectif s'articule à une gestion très fine des débats et discussions, qui peut être formulée ainsi : *« tout est public, sauf ce qu'il faut vraiment pas que ce soit public. C'est plutôt ça que l'inverse,*

que tout est privé et de temps en temps on fait une annonce de comm ». (un des fondateurs). Toutes les listes de discussion, c'est un principe général dans la production des logiciels libres, sont publiques (n'importe qui peut s'y inscrire) et archivées, ce qui garantit la *« transparence »*. Mais ces listes ne sont pas les seuls vecteurs d'échanges, d'autres canaux, privés, peuvent être utilisés : échanges de mails personnalisés, coups de téléphones, rencontres directes. Les échanges non publics sont fréquents à l'intérieur du noyau stratégique : *« on participe à des conversations, à des discussions qui ne sont pas, qui sont... comment dire, secrètes, qui sont en avance, qui ne sont enfin pas secrètes mais discrètes »*. Les justifications qui en sont produites sont toutes orientées vers l'objectif de préservation de la *« communauté »* et de son projet. Il s'agit d'abord d'un moyen de régler des désaccords internes, de traiter de *« divergences »*, de *« re-synchroniser un peu nos objectifs »*, sans étaler publiquement les différends, afin de montrer l'unité du noyau. Il s'agit de produire des décisions collectives (entre membres du noyau) concernant le pilotage stratégique du projet, les développements futurs du logiciel, les chantiers à engager, les options à privilégier : *« il y a beaucoup, beaucoup de discussions effectivement en interne [...] Y compris sur liste privée, enfin ce n'est pas une liste dont on va parler. Il y a pas mal de choses sur les grandes orientations, de choses vraiment lourdes, enfin des choses qu'il faut mettre en chantier. Et il faut qu'on sache si on va le faire, si on va pas le faire, si on va s'orienter vers là, parce que c'est lourd »*. Cette discrétion est congruente avec le monopole des membres du noyau dans les procédures de décision de prise en compte de contributions externes. Elle est aussi un élément de gestion de l'action organisée, dans la mesure où elle est destinée à éviter tout risque de démobilisation lié à des effets d'annonce sans lendemain, au renoncement à des pistes perçues par certains comme passionnantes : *« si on lance ça sur les sites publics tout de suite on va faire du vaporware* [10]*, on va avoir des gens qui vont s'enthousiasmer pour un truc qui va pas être, qu'on va laisser peut-être en plan plus tard »*.

Le souci de resserrer en permanence le travail des contributeurs autour d'un produit à la cohérence et à l'identité fortes est au principe de ces concertations internes au noyau stratégique, à l'abri des autres. Poussée à l'extrême cette logique peut aussi être perçue comme une culture du secret et favoriser une démobilisation de certains participants : *« on a l'impression qu'il y a un petit noyau dans son coin qui décide des trucs et qu'il n'y a pas moyen d'y accéder ou de même simplement savoir ce qu'ils font »* (un développeur). Il est donc nécessaire

10/ Un vaporware (ou logiciel fantôme) est un logiciel annoncé longtemps, toujours retardé, et qui parfois ne voit jamais le jour.

d'articuler canaux privés et canaux publics de manière à renforcer l'adhésion au projet collectif et l'identification au produit logiciel. Les possibilités de jeu sont ici importantes. Ainsi les échanges internes au noyau ne sont souvent qu'une première étape dans une chaîne d'opérations qui vont amener de nouveaux développements et vont conduire à modifier le produit. La mise en discussion publique de pistes de travail est un moyen d'obtenir une large adhésion par élargissement du socle des personnes informées et concernées avant l'annonce de la décision : *« alors après il y a des moments où on se dit : "tiens, cette discussion-là, en fait ce serait plus intéressant qu'elle ait lieu en public, sur la liste des développeurs ; et effectivement celle-là il vaut mieux qu'elle reste privée parce que c'est une perspective qui est beaucoup trop lointaine et ça ne sert à rien d'aller polluer la liste avec ça". Mais après, il ne s'agit pas d'aller faire des cachotteries entre utilisateurs ou aux autres contributeurs »* (un membre du noyau). Les échanges ouverts et publics, parfois assortis de consultations discrètes de tel ou tel administrateur ou développeur, alimentent la réflexion des membres du noyau, et participent à la légitimation de leurs décisions, en amont de celles-ci ou du moins de leur annonce publique. Celle-ci est aussi un moment privilégié d'explication et justification des décisions, en particulier en mobilisant l'argument de la cohérence du produit : *« nous on insiste pour avoir une certaine cohérence et une logique d'intégration et que c'est un produit intégré, les gens comprennent assez bien. On a pas de gros choc, après que le type débarque, trollant sur la liste pour dire : "c'est de la merde, ça fait pas ça". Tout le monde s'en fout [...] grosso modo la communauté comprend qu'il y a une logique, un produit à préserver »* (un des fondateurs). Cette méthodologie de la décision est destinée à entretenir l'adhésion du plus grand nombre au projet, à répondre aux objections enracinées dans des positions particulières, à situer les choix effectués au sein d'une stratégie globale qui les justifie. Le noyau stratégique est à la fois au point d'origine et au point d'arrivée, il est à la source des débats et à leur conclusion, il encadre, dans tous les sens du terme, la décision, selon une figure qu'un administrateur, observateur extérieur mais privilégié de ce fonctionnement, décrit de manière suggestive : *« Je dirai d'une certaine manière ça se passe de manière totalement autocratique en définitive quelque part. Mais il faut bien le justifier, mais il faut bien le discuter, mais discuter publiquement donc, il ne faut pas être en contradiction avec tout ce qu'on dit. Donc en fait, c'est comme les décisions à la japonaise où finalement le chef il a le pouvoir de tout mais, mais il doit le faire émerger de sa base, il ne peut pas prendre une position qui soit totalement contraire à l'ensemble des gens qui sont autour sinon il est tout seul. »*

La mise en discussion publique de pistes de travail est un moyen d'obtenir une large adhésion.

Une caractéristique constante de la « communauté » SPIP au cours du temps et au long de sa croissance réside dans l'autorité et

la légitimité du noyau stratégique. Ce crédit, technique et politique, lui permet de jouer, et de tenir dans le temps, un rôle décisif dans le pilotage du projet, c'est-à-dire à la fois la production du logiciel et la mobilisation des acteurs. Cette coordination présente des traits apparemment contradictoires : elle est centralisée tout en sollicitant les membres à travers les listes ouvertes, elle est directive tout en s'appuyant sur des débats publics, elle est autocratique tout en s'alimentant à des consultations démocratiques. Cette alchimie improbable permet à la fois de maintenir dans le temps la cohérence et l'identité du produit logiciel, et d'autoriser une collectivité – caractérisée par une hétérogénéité croissante des contributeurs – à s'approprier cette production et à s'y identifier.

CONCLUSIONS

Une spécificité, manifeste, des collectifs de production des logiciels libres réside dans l'absence de tout encadrement des relations de travail par des règles appuyées sur des ressources exogènes, telles celles du droit commercial ou du travail. On chercherait en vain une super-règle qui organiserait une telle activité productive qui associe des participants travaillant à distance les uns des autres, reliés par des dispositifs techniques utilisant les ressources d'internet. Mais l'objectif de production ne peut être atteint sans l'implantation de mécanismes de coordination, extrêmement variés. Ainsi, l'activité, collective, de production est traversée et structurée par une multitude de régulations qui contribuent à la mobilisation des travailleurs, à la spécialisation des tâches, à la production de jugements, à la mise en œuvre de sélection, à la différenciation de positions statutaires, à la distribution d'incitations, à l'invention de sanctions. Les caractéristiques de ces règles sont ajustées à la dynamique de chaque projet et à sa trajectoire de développement. Dans le cas de SPIP, le noyau stratégique initial se renforce progressivement par un cercle intermédiaire d'animateurs et d'administrateurs qui assurent la gestion et la régulation de zones périphériques qui elles-mêmes s'élargissent et se diversifient. Cette structuration est à la fois une nécessité fonctionnelle et comporte en même temps le risque de désintégration des collectifs de projet.

L'objectif de production ne peut être atteint sans l'implantation de mécanismes de coordination.

La différenciation des attributions et des positions dans la division du travail, comme la distribution inégale du pouvoir de décision, montrent que ces groupes ne sont pas égalitaires et invalident le modèle du bazar. Ils se caractérisent plutôt par une plasticité organisationnelle, dont témoignent bien la spécialisation continue des

espaces de travail ou la cooptation de nouveaux arrivants dans les positions les plus centrales (noyau, administrateurs). Cette fluidité, dont le débit est contrôlé par les fondateurs et ceux qui sont devenus leurs pairs, permet de répondre aux exigences de maintien de la mobilisation des contributeurs les plus engagés dans le projet, et aux risques latents et structurels de défection. Finalement ces collectifs apparaissent comme des configurations productives originales, dont ne rend pas compte la dénomination usuelle de « communautés du libre ». Ils sont à la fois distants (faibles interactions directes entre travailleurs et absence de super-règle), clivés (différenciation des positions et attributions et diversité des régulations internes), mais aussi soudés (rétribution des engagements et identification positive à un produit bien reconnaissable). Cette organisation du travail, qui demeure en grande partie implicite, permet d'articuler deux objectifs partiellement contradictoires : mobiliser les contributeurs et susciter des investissements dans la fabrication du logiciel d'une part, maintenir la cohésion d'un collectif de travail en expansion et préserver l'identité du logiciel d'autre part. Elle définit un modèle productif alternatif à la fabrication de logiciels propriétaires, qui n'apparaît pas sans atout, du moins quand il atteint, comme dans le cas étudié ici, un tel degré de structuration et de souplesse à la fois.

Didier Demazière (CNRS, laboratoire Printemps-UVSQ)
didier.demaziere@printemps.uvsq.fr
François Horn (CLERSE, IFRESI, Université de Lille 3)
Marc Zune (FNRS-CSTEF, Université Libre de Bruxelles)

RÉFÉRENCES BIBLIOGRAPHIQUES

AURAY N., 2004 — La régulation de la connaissance : arbitrage sur la taille et gestion aux frontières dans la communauté Debian, *Revue d'économie politique*, n° 113, p. 74-99.

BASSET T, 2003 — *Monographie d'un logiciel libre : VideoLAN*, IEP de Paris, DEA de sociologie de l'action organisée.

BUREAU M-C., MARCHAL E. (éds), 2006 — *Au risque de l'évaluation. Salariés et candidats à l'emploi soumis aux aléas du jugement*, Villeneuve d'Ascq : Presses universitaires du Septentrion.

CONEIN B., 2004 — Communautés épistémiques et réseaux cognitifs : coopération et cognition, *Revue d'économie politique*, n° 113, p. 141-159.

DEMAZIÈRE D., HORN F., ZUNE M., 2006 — La dynamique de développement des « communautés » du logiciel libre : conditions d'émergence et régulation des tensions, *Terminal*, n° 97-98, p. 71-84.

ELLIOT M., SCACCHI W., 2004 — Free Software Development : Cooperation and Conflict in a Virtual Organizational Culture, in : Koch S. (ed.), *Free/Open Source Software Development*, Pittsburgh, PA, Idea Group Publishing, p. 152-172.

EYMARD-DUVERNAY F., MARCHAL E., 1997 — *Façons de recruter : le jugement des compétences sur le marché du travail*, Paris : Éditions Métailié.

GHOSH R.A., GLOTT R., KRIEGER B., ROBLES G., 2002 — *The Free/Libre and F/OSS Software Developers Survey and Study-FLOSS Final Report,* University of Maastricht.

GLASS R.L., 2003 — A socio-political look at open source, *Communications of the ACM*, nº 46, p. 21-23.

HEALY K., SCHUSSMAN A., 2003 — The Ecology of Open-Source Software Development, http ://opensource.mit.edu/papers/healyschussman.pdf.

Holtgrewe U., Werle R., 2001 — De-Commodifying Software ? Open Source Software Between Business Strategy and Social Movement, *Science Studies*, nº 15, p. 43-65.

HORN F., 2004 — *L'économie des logiciels*, Paris : La Découverte.

LAKHANI K., WOLF R., 2005 — Why hackers do what they do : understanding motivation and effort in Free/Open source software projects, in : Feller J., Fitzgerald B., Hissam S., Lakhani K. (eds), *Perspectives on Free and Open Source Software*.

LERNER J., TIROLE J., 2002 — Some simple economics of open source, *Journal of Industrial Economics*, nº 52, p. 197-234.

RAYMOND E.S., 1998 — *La cathédrale et le bazar*. Traduction de Blondeel S., http ://www.lifl.fr/~blondeel/traduc/Cathedral-bazaar/Main_file.html.

WEBER S., 2005 — *The success of open source*, Harvard : Harvard University Press.