

LES LEÇONS DE MANAGEMENT DE LA COMMUNAUTÉ LINUX

Michel Ferrary¹, Pascal Vidal²

INTRODUCTION

Depuis la fin des années quatre-vingt, l'industrie du logiciel informatique est dominée par Microsoft. L'entreprise américaine a su résister à ses concurrents sur ses différents segments de marché et aux différents procès anti-trust dont elle a été l'objet. Microsoft équipe plus de 90% du parc mondial de PC avec son système d'exploitation Windows sur lequel sa marge bénéficiaire est de 86%. Cette « vache à lait » dans son portefeuille de produits a permis à Microsoft de financer sa domination dans les logiciels bureautiques (Word, Excel, etc.), les logiciels de navigation sur Internet (Internet Explorer) et maintenant de financer sa pénétration sur de nouveaux marchés tels que les jeux vidéos (Xbox), les services sur Internet aux particuliers (MSN, Hailstorm, Expédia) et aux entreprises (.net). Aujourd'hui, Microsoft est la seconde capitalisation boursière au monde (plus de 260 milliards de dollars), son chiffre d'affaires annuel est de 28,4 milliards de dollars en 2002 (+15% sur un an) et ses bénéfices s'élèvent à 11,9 milliards de dollars.

Alors que son hégémonie semble totale, Microsoft se voit menacé sur le cœur même de son métier que constitue le système d'exploitation (O.S) Windows par un nouvel entrant appelé Linux. En 2003, Linux est le système d'exploitation de 13,7 % des serveurs informatiques fonctionnant à travers le monde dans un marché évalué à 51 milliards de dollars. Selon le Gartner Group, cette part devrait passer à 25,2 % en 2006. En ce qui concerne le marché des logiciels bureautiques, Linux pourrait connaître une forte expansion dans un avenir proche à travers la diffusion d'*OpenOffice* (l'équivalent du "*pack Office*" de Microsoft) développé par la communauté G.N.U./Linux (Linux détient actuellement 2,5% de ce marché).

¹ CERAM, Sophia Antipolis, France. Courriel : michel.ferrary@ceram.fr

² CERAM, Sophia Antipolis, France.

La qualité, la sécurité, la stabilité et le faible coût du système d'exploitation Linux conduisent de plus en plus d'entreprises à l'adopter au détriment des autres systèmes d'exploitation propriétaires (Unix, Windows, etc.). En 1998 (soit six ans après la première version du système d'exploitation), Linux arrivait en tête d'une enquête de satisfaction menée par le Gartner Group auprès de 829 responsables informatiques de grands groupes. L'O.S. libre de droits surclassait déjà ses concurrents dans les domaines de l'interopérabilité, du coût de possession, du prix, de la prise en compte d'Internet, de la disponibilité ou du support Java³. Une étude publiée par la banque Goldman Sachs montre que 39 % des grandes entreprises utilisent Linux.

Des grandes entreprises informatiques comme Intel, IBM., Hewlett-Packard, Dell ou Oracle soutiennent les solutions Linux en optimisant, sous Linux, les équipements informatiques et les logiciels qu'ils commercialisent et en distribuant des solutions informatiques basées sur ce système d'exploitation. Cette stratégie permet à ces grands groupes de s'affranchir partiellement de leur dépendance à Microsoft tout en apportant leur crédibilité aux solutions Linux. Par exemple, en 2000, IBM. a investi 1 milliard de dollars pour rendre compatible ses logiciels et ses serveurs avec Linux et a dédié 250 développeurs pour collaborer avec la communauté du logiciel libre.

Au-delà de l'analyse stratégique, la réussite de Linux interroge les pratiques traditionnelles de gestion. Du point de vue de l'organisation du travail, comment se définissent les missions de la communauté ? Comment se constituent les équipes projet ? comment se diffusent les innovations au sein de la communauté ? Du point de vue managérial, comment expliquer que des centaines d'individus à travers le monde contribuent bénévolement au développement de logiciels ? Quels sont les déterminants de leur motivation ? Quelle est leur rétribution ?

Notre analyse de la communauté Linux s'est faite à travers deux sources de données. La première est constituée par six entretiens semi-directifs avec des informaticiens contribuant à la communauté Linux. La seconde a consisté à explorer les sites de travail et d'échange de la communauté Linux, sachant que la coordination de cette communauté se fait essentiellement par l'intermédiaire du média électronique.

La capacité de Linux à concurrencer Windows ne s'explique pas uniquement par la qualité de ce logiciel, mais également par le mode original d'organisation du travail de développement du logiciel et par la philosophie du mouvement pour le

³ Le Monde Informatique, 03 juillet 1998

développement du logiciel libre (*Free Software Foundation*). Derrière la concurrence entre les deux systèmes d'exploitation se cachent deux logiques différentes de développement des logiciels.

Le succès de Linux s'appuie notamment sur une culture forte qui soude la communauté. Cette efficacité est liée à l'interaction auto-renforçante d'une dimension informationnelle et d'une dimension culturelle. La dimension informationnelle permet de comprendre comment la communauté se coordonne pour produire des logiciels de qualité. La dimension culturelle permet de comprendre pourquoi les individus s'identifient et contribuent à cette communauté. Il convient également de tirer des enseignements du fonctionnement de la communauté Linux pour rendre opérationnel le fonctionnement des communautés virtuelles de pratiques que les grandes entreprises industrielles tentent de mettre en place. La création de communautés de pratique virtuelles coordonnées par l'intermédiaire des réseaux électroniques devient un enjeu managérial important pour des entreprises géographiquement éclatée dans un système de production et de consommation globalisé (Wenger, 1998 ; Ferrary et Pesqueux, 2004).

I. PLANIFICATION OU AUTO-ORGANISATION DU DÉVELOPPEMENT LOGICIEL : LE DÉFI DE LA COMPLEXITÉ

En 1991, un étudiant en informatique Finlandais de 21 ans, Linus Benedict Torvalds commence à travailler sur un nouveau système d'exploitation baptisé Linux (pour Linus' Unix). Ce logiciel est gratuit et son code source est téléchargeable via Internet pour être utilisé ou amélioré par qui le souhaite. Torvalds a mis le code en accès libre en demandant explicitement aux autres internautes de l'aider à le transformer en un système d'exploitation opérationnel. Quelques informaticiens répondent à cet appel et partagent leurs contributions autour de ce noyau (le *kernel*). Celui-ci se développe rapidement et avec lui la communauté de programmeurs bénévoles qui dépasse rapidement le millier de personnes... En quelques années, grâce à la puissance de diffusion d'Internet ce mouvement spontané, auto-organisé, en dehors de toute organisation formelle a transformé Linux en l'un des systèmes d'exploitation les plus performants et fiables qui soient. Désormais, non seulement des programmeurs du monde entier contribuent à l'amélioration de Linux, mais nombre d'entre eux travaillent gratuitement au développement d'applications (traitements de texte, tableurs, gestion de bases de données, etc.) fonctionnant sous Linux.

Le paradoxe de la réussite de Linux c'est que celle-ci ne résulte pas d'une planification stratégique définie et mise en œuvre de façon formelle par une

grande entreprise, mais de la collaboration d'une communauté auto-organisée. Linux n'est pas une entreprise légalement constituée, mais une communauté informelle et les ressources humaines qui se sont mobilisées à l'origine du projet n'étaient pas des salariés d'une quelconque entreprise « Linux », mais des bénévoles contribuant à un projet utopique collectif.

Le développement d'un système d'exploitation, lorsqu'il est pris en charge par une « entreprise classique » de logiciels (Microsoft, SAP, Oracle, etc.), coûte des millions de dollars et demande la mise en place d'une structure de management conséquente, permettant d'administrer de multiples équipes projets. Les projets de production de systèmes d'exploitation sont d'une complexité croissante qui nécessite des équipes projets de plus en plus importantes et posent des problèmes de coordination des développeurs.

L'analyse de l'évolution du projet NT chez Microsoft montre que celui-ci démarra fin 1989 avec une équipe de six personnes. Lors de la sortie de la première version commerciale de « Windows NT » en 1993 l'équipe avait atteint 450 personnes et produit 4,5 millions de lignes de code. Deux années plus tard une équipe de taille similaire a réussi en un délai équivalent à produire un OS (Windows 95) qui comprenait 11 millions de lignes de code (Cusumano, 1997). Aujourd'hui, la suite du projet NT regroupe une équipe de huit à 10 000 personnes dont 5000 développeurs pour un nombre de lignes de code supérieur à 50 millions.

Si les délais de production n'ont pas changé, la taille des équipes a été multipliée par plus de dix entre Windows 95 et Windows Server 2003, et dans le même temps le nombre de lignes de code a été multiplié par « seulement » 5, conformément à la « loi de Brooks »⁴ (Brooks, 1995) selon laquelle l'augmentation du nombre de programmeurs sur un projet de développement informatique a tendance à ralentir celui-ci, en rendant plus difficile la communication et la coordination entre un nombre plus élevé d'acteurs, contribuant à la baisse de la productivité moyenne des développeurs.

Bien qu'ayant évolué depuis le milieu des années 1980 pour faire face à cette complexité, tout en améliorant le niveau de qualité et les délais de réalisation de ses produits, le modèle d'organisation de la production logicielle de Microsoft demeure un modèle de développement classique, structuré autour d'une étape initiale de planification (Cusumano, 1997) et un modèle de développement qui

⁴ Brooks explique que « les coûts de communication et de complexité d'un projet augmentent de manière quadratique avec le nombre de développeurs, alors que le travail réalisé n'augmente que linéairement. Depuis, cela est connu sous le nom de "loi de Brooks" ».

reste centralisé, puisque dans le cas du projet de Windows Server 2003, l'ensemble des équipes de développeurs, testeurs, chefs de projets sont sous la responsabilité d'un seul homme, Todd Wanke.

En même temps, la communauté Linux représente aujourd'hui un groupe de 40 000 programmeurs à travers le monde. Le modèle de développement adopté par cette communauté, au lieu des quatre phases d'un processus « classique » de production de logiciels (planification/analyse/conception/implémentation), se concentre, comme nous le rappelle Iannacci (2003), sur le codage (l'implémentation), le test du code produit et son *debogage*. Ceci est rendu possible par la modularité du noyau qui permet à ce système d'être développé sans conception d'ensemble.

La supériorité de Linux est liée à la capacité de son mode d'organisation à faire face à la complexité croissante du développement d'un système d'exploitation, et remet en question la « loi de Brooks ». Hutchins (1991), à travers ses réflexions sur l'écologie cognitive, a montré que les technologies de l'information structurent les comportements et modifient les conditions d'interaction et de coordination entre les individus permettant de faire face à des environnements de travail complexes. Le développement de Linux nous enseigne que grâce aux NTIC et en particulier à Internet, il existe d'autres modalités de production collective de logiciels. La technologie apporte ici un support qui permet de diminuer les « coûts de coordination », permettant une production collective assurée par un nombre croissant de développeurs.

Dans la concurrence que se livrent les entreprises de logiciels propriétaires (dont Microsoft est la figure emblématique) et les communautés de logiciels libres, peut-être qu'à terme, la culture du logiciel dont le code source est ouvert triomphera, non pas parce qu'il est moralement bon de coopérer, non pas parce qu'il est moralement mal de « clôturer » le logiciel [...] mais simplement parce que le monde dont le code source est fermé ne peut pas gagner une course aux armements évolutive contre des communautés de logiciel libre, qui peuvent mettre sur le problème un temps humain cumulé plus important de plusieurs ordres de grandeurs. (Raymond, 1998)

Mais ce « temps humain » peut-il être efficient sans un minimum d'organisation ? Les nombreuses contributions individuelles au développement de Linux se déroulent-elles de façon totalement désorganisée au point de constituer un « bazar » par opposition à la « cathédrale », modèle d'organisation hiérarchisée représenté par un système classique d'entreprise de production de logiciels ?

II. CARACTÉRISTIQUES ORGANISATIONNELLES ET MODE DE FONCTIONNEMENT DE LA COMMUNAUTÉ LINUX

Derrière l'apparente anarchie de la communauté, « l'architecture modulaire de Linux est au contraire relativement hiérarchisée » (Narduzzo et Rossi 2003) et fonctionne sur la base d'un certain nombre de principes fondamentaux.

1. Principe de développement permanent

La première caractéristique de ce mode d'organisation réside dans le fait que le système Linux est développé de façon massivement parallèle et sans réel système de planification centralisé. Ce mode de développement est rendu possible par la modularité de la conception du noyau de ce système d'exploitation qui diminue le besoin de coordination.

Cette modularité permet un processus de développement quasi « permanent ». Les mises à jours sont extrêmement fréquentes, et le système d'exploitation intègre quasi-immédiatement toutes les améliorations apportées par les développeurs y compris celles destinées à tenir compte des évolutions matérielles (nouvelles imprimantes, etc.).

À un instant donné coexistent une version stable de Linux apte à satisfaire pleinement les utilisateurs qui veulent un système très fiable (notamment les entreprises) et une version, voire plusieurs, encore en cours d'expérimentation qui intègrent les derniers développements.

Cette particularité permet, compte tenu du nombre d'utilisateurs-contributeurs membres de la communauté, de rechercher des solutions innovantes pour les prochaines versions, tout en exploitant au mieux les capacités existantes dans l'environnement technologique du moment. On se trouve dans une situation d'exploration – exploitation (March, 1991) où les projets coexistent sans être en concurrence, à la différence d'une organisation classique où les ressources consacrées à une innovation ne sont pas utilisées pour l'exploitation de solutions techniques alternatives.

Le principe de développement permanent de Linux repose sur le fait que les utilisateurs de ce système d'exploitation sont aussi potentiellement des développeurs de celui-ci : « La plus grande innovation de linux réside dans le fait de considérer les utilisateurs comme des codéveloppeurs, ce qui constitue la solution la plus rapide pour l'amélioration des programmes et leur *débogage* » (Raymond, 1998). Les utilisateurs sont potentiellement intégrés dans le cycle de

développement, y compris ceux qui ne développent pas directement. Ils participent à ce processus par la détection de bogues (et éventuellement la proposition de solutions à ceux-ci), mais aussi par leur proposition en matière d'usage. Comme le note Linus Torvalds, « j'ai plus bénéficié du fait de disposer d'un grand nombre d'utilisateurs que du fait de disposer d'un grand nombre de développeurs, même si j'ai eu besoin de ces deux populations pour créer le système que Linux est aujourd'hui. Je n'avais aucune idée des spécificités dont les utilisateurs souhaiteraient disposer, et si j'avais continué à développer Linux de mon côté il aurait évolué en un système moins intéressant et complet. »⁵

La modularité du système Linux conjuguée avec un développement massivement parallèle et permanent, ainsi que la contribution d'un nombre important d'utilisateurs concepteurs/ contributeurs permet la détection et la correction rapide des bogues, selon ce que Raymond (1998) appelle *la loi de Linus* « étant donné suffisamment d'observateurs, tous les bogues sautent aux yeux. » Ce principe de « développement permanent », lié à la modularité de ce système d'exploitation, est synonyme de forte décentralisation du processus de développement. Mais cette décentralisation n'est pas synonyme de désorganisation, au contraire, on observe une certaine formalisation des pratiques.

2. Principe d'auto-organisation dans un système décentralisé

Une autre caractéristique essentielle du mode d'organisation de la communauté Linux est que n'importe quel programmeur ou presque⁶ peut de sa propre initiative essayer de prendre en charge le développement d'un module ou la résolution d'un problème technique.

Aucun programmeur n'est affecté à une tâche particulière par une quelconque organisation centrale. Chacun contribue à un module qu'il considère comme pouvant relever de ses compétences et constituer un challenge intéressant, proche de ses centres d'intérêts du moment. À la différence des organisations classiques, la priorité est laissée aux initiatives locales et individuelles.

Cependant le processus pour avoir une quelconque responsabilité dans un projet est bien défini. Avant d'être associé au développement d'un module, un développeur doit faire la preuve de la qualité de ses développements antérieurs auprès des membres de la communauté. Ensuite, son travail de développement doit être validé par le responsable du module. Markus et *al.* (2000) citent le cas

⁵ First Monday, mars 1998, traduit par Sébastien Blondel.

⁶ Il faut tout de même avoir des compétences en programmation pour prendre en charge de telles tâches.

d'Apache où n'importe quel programmeur volontaire est susceptible de contribuer à un projet pendant quelques mois. Pendant cette période la qualité de son travail ainsi que son adhésion à la philosophie de l'*Open Source* sont évalués. Après une « période probatoire », l'affectation d'une responsabilité particulière (d'un module par exemple) est décidée par consensus des fondateurs d'Apache.

Dans ce mode de fonctionnement, la capitalisation de la connaissance est omniprésente. Les projets sont coordonnés par un petit nombre d'individus fortement impliqués (un noyau d'individus stables) qui capitalisent sur l'ensemble des contributions des programmeurs volontaires au fil du temps. Les « responsables » qui dynamisent la communauté, structurent et coordonnent les efforts de l'ensemble de ses membres ne sont donc nullement nommés, mais émergent naturellement de par la qualité de leurs contributions et leur implication à la vie communautaire et également par la reconnaissance de leurs compétences par la communauté. Cette légitimité naturelle fonde leur « autorité » et leur position hiérarchique dans la communauté. Le principe de base est que le développeur qui a initialement pris en charge le développement d'un module en garde la responsabilité. Ces responsables deviennent de véritables leaders et pour certains d'entre eux des *gourous* pour l'ensemble de la communauté. Une telle position ne se décrète pas, elle émerge par le fonctionnement même de la communauté.

L'implication dans la communauté Linux peut aider à la résolution de problèmes techniques qu'un informaticien peut rencontrer dans sa pratique quotidienne de Linux (dans son activité professionnelle). La communauté joue ici à la fois le rôle de support technique mais plus encore favorise une dynamique d'interactions favorable à l'apprentissage. Ainsi, un informaticien qui a installé Linux dans l'entreprise qui l'emploie a été amené à utiliser Debian (une des distributions de Linux). Il a rencontré un problème technique particulier avec ce logiciel qu'il a soumis au forum technique de Debian organisé sur Internet, comme personne ne lui a apporté de solution, il s'est attelé à la tâche en le notifiant à la communauté par l'intermédiaire d'un envoi de courriels à l'ensemble de la liste de diffusion de la communauté Debian. Il est ainsi devenu de manière naturelle et informelle responsable d'un sous-projet (un module) dont la finalité est de résoudre ce problème. Dès qu'il a programmé une solution, il la met à la disposition de la communauté Debian. Les membres de la communauté qui sont intéressés par son utilisation vont tester ce programme, signaler au développeur les bogues éventuels et pour les plus compétents d'entre eux proposer des solutions. L'équipe projet se sera constituée naturellement, elle grandira et diminuera jusqu'à la résolution du problème, sachant que lorsque ce module sera terminé, le développeur qui en est à l'origine en restera responsable pour les années à venir et

pourra coordonner ou transmettre les informations nécessaires pour des évolutions futures.

Cependant si la contribution est libre et « ouverte à tous », l'intégration finale des modifications proposées au niveau du noyau est soumise à l'aval de Linus Torvalds ou de l'un de ses « lieutenants ». Une sorte de sélection naturelle s'opère alors. Si le code proposé par un programmeur n'est pas de bonne qualité, un autre en proposera rapidement un qui sera meilleur. Plusieurs programmeurs peuvent donc travailler sur la résolution du même problème de façon quasi concurrentielle. Mais la modalité de fonctionnement est ici encore, essentiellement coopérative, puisque les solutions proposées par les différents programmeurs sont immédiatement soumises à l'ensemble de la communauté pour avis et mises à sa disposition pour apporter des améliorations.

3. Principe de transparence

La transparence des échanges d'information constitue également un principe essentiel de fonctionnement de la communauté Linux et constitue un gage de sécurité, de fiabilité, de qualité et même de productivité. L'ensemble du processus de développement fonctionne dans la plus grande transparence : au niveau des contributions, des échanges de point de vue, des comportements, et des sanctions éventuellement mises en place (exclusion de la communauté...).

Dans le mode de fonctionnement de la communauté Linux, le responsable d'un module a tout intérêt à mettre toute nouvelle version le plus rapidement possible à la disposition des membres de la communauté à la fois pour avoir de nombreux commentaires, pour qu'on lui signale les bogues éventuels et qu'on lui propose des améliorations de son module. Une large diffusion contribue également à accroître le nombre d'utilisateurs et la satisfaction personnelle et la notoriété d'un responsable de projet dépend de la qualité de son logiciel et du nombre d'utilisateurs. Ainsi, E. Raymond dans son texte de référence pour la communauté Linux intitulé *The Cathedral and the Bazaar* explique que lorsqu'il développait l'application *Fetchmail* comme application de courrier électronique fonctionnant sous Linux, il mettait tous les jours une nouvelle version de *Fetchmail* à la disposition de la communauté et qu'en trois semaines le nombre de développeurs contribuant au projet à travers le monde a atteint rapidement 300 personnes, pour ensuite décroître dès lorsque l'application fut stabilisée.

La sécurisation des échanges est liée à la parfaite circulation de l'information, quiconque violerait les valeurs de la communauté en s'appropriant une partie des codes ou en intégrant volontairement un bogue dans un programme serait

immédiatement signalé à l'ensemble de la communauté et exclu de la liste de diffusion électronique et des échanges techniques virtuels.

La dimension informationnelle d'une communauté, en tant que groupe d'individus fortement socialisés, se caractérise par un fort degré d'interconnaissance des individus appartenant à la communauté, par une conscience collective de cette connaissance commune de chacun sur tous les autres et sur la promptitude de la diffusion d'une nouvelle information entre les membres du réseau que constitue la communauté. Cette connaissance commune sur tous les membres de la communauté modifie la nature et l'horizon d'optimisation des individus et crée une coercition sociale qui limite les comportements opportunistes pouvant conduire à l'exclusion de la communauté en cas de comportement non conformes aux règles de la communauté (appropriation d'une partie du logiciel, introduction volontaire de bogues, etc.).

Cette dimension informationnelle renforcée par la recherche de reconnaissance sociale des membres au sein de la communauté et l'existence d'une culture forte déterminent une dynamique vertueuse qui limite les risques de comportements déviants au sein de la communauté. La dynamique communautaire favorise la construction de la réputation des acteurs et fait de cette dernière un actif que les individus chercheront à accroître et à rentabiliser.

De plus, pour construire un lien de confiance transférable à l'échelle de la planète, les membres de la communauté s'identifient physiquement mutuellement à l'occasion de salons ou de congrès et échangent des codes d'identification qu'ils incluront ensuite dans leurs communications électroniques avec les autres membres de la communauté afin de signaler les noms des personnes qui les ont physiquement identifiés et ainsi rassurer leurs correspondants contre une éventuelle utilisation frauduleuse des adresses électroniques.⁷ Il y a une transitivity du lien de confiance qui conduit les membres de la communauté à être identifiés physiquement par le nombre le plus large possible de membres de la communauté pour que les messages électroniques soient ensuite reconnus comme étant fiables par la communauté à travers un processus d'apprentissage social collectif, ce qui permet de sécuriser les échanges.

⁷ Par exemple, si Jean a rencontré physiquement Patrick lors du salon du logiciel libre de Bordeaux, qu'ils se sont mutuellement identifiés physiquement et qu'ils ont échangé leur clé numérique d'identification, alors lorsque Jean enverra des messages électroniques aux autres membres de la communauté, il spécifiera qu'il a été identifié par Patrick, ainsi ses correspondants, par exemple Mike, qu'il n'a jamais rencontré mais qui a déjà identifié Patrick lors d'une rencontre physique, accordera une plus grande confiance au message envoyé par Jean car il a été identifié par Patrick et que lui-même, Mike, a eu l'occasion par le passé d'identifier Patrick, par exemple au salon Linux qui se tient chaque année à New-York.

4. Principe philosophico-juridique de l'Open Source

Il existe un dernier principe fondamental, sur lequel repose finalement tous les autres : le principe de licence *open source*. Celui-ci autorise à la fois une très grande transparence dans la circulation de l'information, mais aussi l'intégration réelle des utilisateurs dans le processus de développement, et enfin la « libre » participation d'un très grand nombre de développeurs à ce processus.

Linux est un logiciel *Open Source* dont l'utilisation est spécifiées par un code réglementaire défini par Richard Stallman intitulé G.N.U. G.P.L. (G.N.U. *General Public License*) déposé en 1991 par la F.S.F. (*Free Software Foundation*) qui stipule :

1. Le logiciel doit faire l'objet d'une redistribution libre et gratuite.
2. Le programme doit inclure le code source et doit autoriser la distribution du code source comme de l'exécutable compilé.
3. La licence doit autoriser les modifications et les travaux dérivés au programme et leur distribution dans les mêmes termes que la licence du logiciel d'origine.
4. L'intégrité du code source de l'auteur doit être respectée.
5. La licence ne doit pas être discriminative à l'encontre de personnes ou de groupes.
6. La licence ne doit pas restreindre ni interdire l'usage du logiciel à un quelconque domaine d'activité.
7. Les droits attachés au programme s'appliquent à tous ses utilisateurs sans qu'il leur soit besoin de se conformer à des termes de licence complémentaires.
8. La licence ne doit pas être spécifique à un programme.
9. La licence ne doit pas imposer de restrictions sur d'autres logiciels distribués avec le programme source⁸.

Cette dimension juridique constitue un principe philosophique à laquelle adhèrent les membres de la communauté, participe à la structuration de la communauté, et facilite la coordination au sein d'un espace cognitif collectif où les interactions sont essentiellement médiatisée par la technologie.

III. LES FACTEURS DE MOTIVATION ET LES CARACTÉRISTIQUES CULTURELLES DE LA COMMUNAUTÉ LINUX

⁸ www.opensource.org/docs/osd-français.php, cité par Le Monde Informatique, 6 février 2003

marquant cependant de ce fait une forme de hiérarchisation de la communauté. De même, le code de conduite stipule :

faites attention à bien utiliser la liste appropriée. En particulier, n'envoyez pas de questions concernant l'utilisation sur des sites de développement " et " si vous envoyez des messages à des listes auxquelles vous n'êtes pas abonné, indiquez toujours cela dans le corps de votre message .

Cela signifie à la fois une porosité limitée entre les groupes de la communauté mais également une obligation de signaler sa non-appartenance au groupe quand on souhaite communiquer avec lui. Que ces règles soient définies pour permettre un fonctionnement optimal de la communauté n'empêche pas qu'elles contribuent aussi à sa hiérarchisation sociale.

Cette hiérarchisation est une source de motivation, car ses différents échelons professionnels peuvent être gravis par les membres de la communauté. La communauté Linux est une méritocratie qui offre des possibilités d'ascension sociale à tout développeur faisant la preuve de sa compétence.

Le statut et la position sociale dans la communauté sont des facteurs de motivation importants pour les membres de la communauté. Comme le souligne Raymond (2000) dans *Homesteading the Noosphere* (ouvrage électronique décrivant le fonctionnement de la communauté du logiciel libre), la reconnaissance par les pairs de la contribution à une solution technique de qualité est un facteur de motivation à un point tel que les membres de la communauté ont inventé un néologisme pour qualifier ce désir de reconnaissance : « l'egoboo ». Les règles de la communauté *Open Source* protègent la paternité du code non pas pour des raisons économiques puisque le logiciel est gratuit, mais pour conférer une reconnaissance sociale, car les auteurs sont très fiers de leur œuvre et ne veulent pas que quelqu'un vienne supprimer leur nom de celui-ci ou prétende l'avoir écrit (www.debian.org). Le statut social d'un développeur se construit à travers sa capacité à fournir rapidement une réponse de qualité à un problème rencontré par un utilisateur. Dans le cadre des activités de développement, la libre publication du code augmente la valeur du développeur, qui voit sa notoriété augmenter. Il s'agit d'un jeu définissant un cercle vertueux de l'augmentation de la valeur personnelle par la valeur commune. En répondant rapidement sur les forums électroniques à un problème rencontré par un membre de la communauté, un développeur signale son expertise à l'ensemble de la communauté. Cet enjeu de positionnement social par la signalisation de son expertise conduit à une compétition entre les développeurs pour apporter rapidement une solution aux problèmes de la communauté. Il y a un risque d'apporter une solution de piètre

qualité et d'être publiquement humilié par un autre développeur qui, non seulement apportera une meilleure solution, mais en même temps, mettra en évidence la pauvreté de la solution proposée par le premier développeur et donc la faiblesse de ses compétences. En revanche, le second développeur acquerra ou renforcera, par ce moyen, son prestige et son statut social au sein de la communauté virtuelle des développeurs. Ce risque de sanction symbolique par les pairs favorise un processus d'autocensure des développeurs qui les conduit à bien vérifier leurs solutions avant de les proposer à la communauté.

D'un point de vue psychanalytique, le projet Linux offre à ses membres des vecteurs de cristallisation de leurs pulsions de vie et de leurs pulsions de mort. Freud (1968) dans sa théorie des pulsions montre que l'individu est porteur de pulsions de vie créatrices et de pulsions de mort destructrices qui se cristallisent nécessairement dans les activités humaines. Les individus cherchent naturellement des vecteurs d'expression de ces pulsions. Le projet de la communauté Linux permet l'expression de ces deux formes de pulsions. Contribuer au développement de Linux signifie participer à un projet humaniste permettant l'expression des pulsions de vie. La gratuité du logiciel d'exploitation et des applications s'intègre dans un projet social qui vise à donner l'accès à l'informatique au plus grand nombre d'utilisateurs. Linux est perçu comme un facteur de réduction de la fracture numérique comprise par le monde politique comme étant la nouvelle source d'inégalité sociale tant entre les classes sociales des pays occidentaux qu'entre les pays développés et les pays en voie de développement. Comme le note Linus Torvalds : *Nombreux sont ceux qui apprécient l'interaction sur Internet, tout simplement, ainsi que le sentiment d'appartenir à un groupe qui fait quelque chose d'intéressant : c'est ainsi que certains projets logiciels ont vu le jour.*

D'autre part, le fait que Linux soit le logiciel concurrent de Windows développé par Microsoft, entreprise symbole du capitalisme et de la logique, des codes propriétaires mobilisent les pulsions de mort des membres de la communauté Linux. L'identité des membres de la communauté se construit notamment contre cet ennemi diabolisé qui devient le bouc émissaire du groupe, et contribuer au développement des logiciels libres revient implicitement à combattre la logique commerciale de Microsoft. C'est cette haine fédératrice de Microsoft qui permet d'expliquer pourquoi, alors que des entreprises symboles du capitalisme telles qu'IBM, Intel, Dell, Hewlett-Packard ou Oracle se saisissent du phénomène Linux à des fins commerciales, les *hackers* à l'origine de la communauté Linux, les développeurs porteurs d'une culture de la marginalité en conflit avec les valeurs sociales établies continuent de contribuer aux différents projets de la communauté

Linux et à collaborer avec ces grandes entreprises au nom de l'union contre Microsoft.

Microsoft joue ainsi le rôle de catalyseur des pulsions de mort des membres de la communauté Linux et laisse s'exprimer la pulsion de vie créatrice au sein de la communauté.

2. Les caractéristiques de la culture de la communauté Linux

Peters et Waterman (1983) dans leur célèbre ouvrage *Le prix de l'excellence* avaient mis en évidence le rôle de la culture dans la performance des entreprises. Bien que Linux ne soit pas une entreprise légalement constituée, elle n'en demeure pas moins une entité dotée d'une forte culture porteuse d'efficacité productive.

La culture d'une communauté se caractérise par son langage, ses valeurs, ses règles formelles, ses rites, ses normes, ses symboles, son fondateur, ses histoires, ses héros, ses leaders, ses parias et sa stratification sociale qui font de la communauté une réalité sociale à laquelle s'identifie les membres de la communauté et qui détermine leurs comportements (Schein, 1985). Dans cette approche, la notion de culture met l'accent sur ce qui est partagé par les membres d'une communauté et la culture évoque un tout doté d'une grande cohérence. Ce sont les connaissances, schémas mentaux ou structures de pensée qui sont partagés, et la culture intervient inconsciemment dans la manière d'agir des individus. La communauté sera d'autant plus une réalité sociale que les individus qui la composent s'y identifieront et qu'ils auront conscience d'y appartenir. La communauté se caractérise, en tant que telle, par une communauté d'information, d'intérêts et de valeurs partagés qui est auto-entretenu par les pratiques et les comportements des membres de cette communauté.

Linux possède toutes les caractéristiques d'une communauté sous-tendue par une culture, au sens anthropologique du terme. Elle a un fondateur : Linus Torvalds, des valeurs : celle de la mise à disposition du monde de logiciels fiables et gratuits, des règles : celles de l'*Open Source* définies par Richard Stallman de la *Free Software Foundation* et la licence G.P.L. Elle a ses rites de passage dans la communauté, comme être admis dans un projet, se voir confier un « module » à développer ou avoir une adresse électronique d'un projet. Elle a ses symboles comme le pingouin Tux qui représente Linux, son langage spécifique avec l'acronyme récursif G.N.U. (*GNU is not Unix*) ou le terme *copyleft* pour qualifier juridiquement les logiciels libres par opposition du terme « copyright » qui protège les logiciels propriétaires. Elle a ses héros, notamment les chefs de projets réputés comme Ian Murdock le responsable du projet Debian, Miguel de Icaza, le

programmeur mexicain responsable du projet GNOME ou encore Linus Torvalds qui reste responsable du *kernel*, c'est-à-dire le cœur du système d'exploitation Linux (Linus Torvalds se qualifiant lui-même de « dictateur bénévole » qui contrôle les développement de Linux). Elle a son philosophe législateur en la personne de Richard Stallman, fondateur de *Free Software Foundation*, porteur de l'utopie philosophique de l'*Open Source* et qui en assure le prosélytisme. Elle a son chroniqueur historien avec Eric Raymond qui a rédigé les textes fondamentaux de la communauté *Open Source*, en l'occurrence *The Cathedral and the Bazaar*, *Homesteading the Noosphere* et *The Magic Cauldron*. La communauté a ses lieux de rencontres virtuelles (les forums sur Internet et les listes de diffusion électroniques) et physiques (les salons et les congrès).

On retrouve une logique d'échange par le don dans la communauté Linux. Raymond (1999) illustre cette logique d'échange dans *The Magic Cauldron*. Quand un programmeur Lambda a trouvé une solution à un bogue, il a deux possibilités : garder le correctif pour lui, ou en faire gracieusement bénéficier les autres. La première possibilité ne lui apporte rien. Le second choix ne lui apportera peut-être rien, mais il peut aussi encourager d'autres à donner des correctifs qui résoudront certains des problèmes que le programmeur Lambda pourra rencontrer à l'avenir. Ainsi, tout le monde travaille pour améliorer Linux, et chacun profite des efforts de tous les autres. Et c'est ce qui rend Linux si vertueux : on y met quelque chose, et cet effort est multiplié par la contribution des autres membres de la communauté. Comme le note Linus Torvalds :

Imaginez dix personnes accordant chacune une heure quotidienne au projet. Chacun contribue à hauteur d'une heure de travail, mais puisqu'ils partagent le résultat final, ils récupèrent gratuitement neuf heures du travail des autres ... Remarquez que je ne prétends pas que tous les utilisateurs se comportent ainsi : on trouvera assurément des utilisateurs qui ne font rien remonter. Mais le logiciel a ceci de particulier qu'il ne coûte rien aux programmeurs de laisser de tels utilisateurs disposer du programme malgré tout. Ce qui est important, c'est d'encourager un nombre suffisant d'utilisateurs à faire des commentaires, afin de disposer d'un taux de retour correct.

Malone et Laubacher (1998) envisagent l'organisation du XXI^e siècle non pas comme une entité contrôlée par une « chaîne managériale » stable, mais plutôt comme un réseaux d'agents « indépendants » interconnectés au travers de réseaux personnels et électroniques. Les grandes entreprises sont caractérisées par un éclatement géographique de leurs activités qui pose le problème de la coordination de leurs acteurs. Pour améliorer la coordination des entreprises éclatées, les choix managériaux ont porté sur des investissements informatiques importants en matière de systèmes d'information afin d'améliorer la communication entre les individus et de créer des communautés virtuelles de pratiques.

Ce que montre l'exemple de la communauté Linux c'est que le réseau informatique est une condition nécessaire mais non suffisante à l'existence de communautés de pratiques. Pour que collaborent des individus dont *a priori* les intérêts convergent, il est nécessaire qu'ils se constituent en une communauté avec une culture et des pratiques communes. *Les communautés de pratique sont des groupes d'individus partageant le même intérêt, le même ensemble de problèmes, ou une passion autour d'un thème spécifique et qui approfondissent leur connaissance et leur expertise en interagissant régulièrement* (Wenger, 1998). En ce sens, Linux constitue une communauté de pratiques. Mais de façon plus générale il s'agit également d'une communauté « sociale » avec ses règles, ses valeurs, ses symboles et ses leaders. Les principes de développement permanent, de décentralisation dans un système ouvert et de transparence qui structurent le fonctionnement de cette communauté, ne fonctionneraient pas, sans le principe quasi philosophique de l'open source et une identité communautaire très forte.

Si la communauté Linux montre qu'une communauté virtuelle d'individus électroniquement reliés ne peut exister sans une dimension sociale forte, elle montre aussi que cette relation sociale se construit à travers les liens électroniques, car les systèmes d'information ne sont pas uniquement un vecteur de circulation d'informations économiques et industrielles, mais également un vecteur d'apprentissage social et d'engagements émotionnels permettant l'émergence d'une communauté virtuelle sous-tendant l'efficacité productrice de l'entreprise. Dans l'interaction sociale, il faut considérer le lien électronique comme un lien de « communalisation » (Weber, 1971). Le lien électronique constitue le nécessaire lien faible du processus d'apprentissage social qui permet la construction et la pérennisation de liens sociaux forts dans la réalité virtuelle ou la réalité physique et favorise l'émergence d'une communauté.

L'erreur serait d'affirmer que le lien fort constitué par un engagement émotionnel suppose nécessairement une proximité physique, un face à face entre les individus. Or, outre le fait que des liens sociaux faibles peuvent se nouer par l'intermédiaire de systèmes d'échange électroniques (internet ou autres), il faut comprendre qu'une nouvelle génération d'individus émerge qui vit de façon plus ou moins large sa réalité sociale dans le « monde électronique » dans lequel son éducation et ses mécanismes psychiques la dispose à nouer et à entretenir des liens sociaux forts porteurs d'engagements émotionnels intenses.

À l'aune de la communauté Linux, le débat qui oppose « monde réel » et « monde virtuel » est vide de sens dans la mesure où ils constituent deux dimensions interagissantes de la même réalité sociale à laquelle appartiennent les individus. Les réseaux électroniques et les réseaux sociaux ne sont pas deux réalités cloisonnées et différentes, mais deux dimensions de la même réalité sociale dont les interactions contribuent au renforcement du processus de socialisation. L'exemple de Linux montre qu'Internet est un facteur de socialisation, car il permet d'identifier des communautés d'intérêt, de communiquer à moindre coût avec les membres de cette communauté à l'échelle planétaire, d'être protégé dans les échanges et de rompre ou d'entretenir aisément le lien social dans un processus d'apprentissage social.

RÉFÉRENCES

- Brooks F.** 1995. *The Mythical Man-Month : Essays On Software Engineering*, Addison-Wesley
- Cusumano.** 1997. « How Microsoft makes large teams work like small teams », *Sloan Management Review*, Fall
- Ferrary.M.** et **Y. Pesqueux.** 2004 *L'organisation en réseau. Mythes et réalités*, PUF
- Freud S.** 1968. *Métapsychologie*, Gallimard, 184 p.
- Hutchins E.** 1991. « Organizing Work by Adaptation », *Organization Science*, vol. 2, n°1, February
- Iannacci F.** 2003. *The Linux Model*, Working Paper, London School of Economics
- Malone et Laubacher.** 1998. "The Dawn of the e-lance economy", *Harvard Business Review*, Sept/Oct., Vol. 76, Issue 5
- March J.G.** 1991. *Décisions et organisations*, Les Editions d'Organisation
- Mockus A., R.T Fielding et J. Herbsleb.** 2000. « A case study of open source Software Development : The Apache Server ACM », *Proceedings of ICSE*
- Maslow A.** 1954. *Motivation and Personality*, Harper and Row, 288 p.
- Narduzzo A. et A. Rossi.** 2003. « Modularity in action : GNU/Linux and Free/open source software development model unleashed », *Quaderno Disa*, n°78
- Peters T. et R. Waterman.** 1983. *Le prix de l'excellence*, Interéditions

- Raymond E.** 1998. « The Cathedral and the Baazar » (version électronique)
- Raymond E.** 2000. *Homesteading the Noosphere* (version électronique)
- Raymond E.** 1999. *The Magic Cauldron* (version électronique)
- Schein E.** 1985. *Organizational Culture and Leadership*, Jossey-Bass
- Wenger E.** 1998. *Communities of Practice : Learning, Meaning and Identity*, Cambridge University Press
- Weber M.** 1971. *Economie et Société*, Plon