

"La régulation de la connaissance : arbitrage sur la taille et gestion aux frontières dans la communauté Debian"

Nicolas Auray (ENST-EHESS)

résumé :

Les communautés vouées à l'échange des connaissances affrontent un problème constant : la nécessité de réguler leur taille, afin d'arbitrer entre l'effet club et l'effet congestion. Cet arbitrage intègre en fait une pluralité complexe de choix de régulation. Quelle doit être l'amplitude de l'ouverture de nouveaux entrants ? Selon quelle proportion répartir, pour chaque développeur, le temps consacré à sélectionner ou auditionner les candidats, et le temps passé à envoyer un conseil à la communauté ? Selon le statut ou l'ancienneté du membre, comment doit-il répartir son effort entre l'assistance aux novices et la coopération avec les autres experts ? L'article fait l'hypothèse que les modalités de la régulation diffèrent selon le contenu cognitif des flux d'idées. C'est à partir de l'analyse sociologique d'une communauté complexe ayant existé que l'article tente de répondre à ces questions.

The regulation of knowledge: size and scope control for the Debian community

abstract :

The epistemic communities are confronted with a recurrent problem : they must regulate their scope, so as to arbitrate between the club effect and the congestion effect. This compromise integrate actually a complex plurality of regulation choices. What should be the scale of the openness for the new entrants ? How to proportionate, for each developer, the time allowed to select the applicants and the time allowed to send an advice to the community ? How shall the member of the community share out his effort between assistance for newbies and cooperation amongst experts ? The hypothesis of the paper is that the characteristics of the regulation depends on the cognitive content of the flow of ideas. It is with a sociological analysis of a complex community that the article tries to answer to those questions.

Parce qu'elles constituent des collectifs voués à l'échange de connaissances, les communautés médiatisées de consommateurs sont animées par un souci permanent de contrôle de la *qualité* des informations échangées. Ce « crible de la qualité » fonctionne selon une configuration générique, celle d'un double tri : il vise à écarter sur le long terme à la fois les connaissances fortement répétitives, et celles ayant une incertitude trop grande. Bien que les seuils en soient mobiles et continuellement ajustables, la régulation de la connaissance et de sa qualité repose ainsi sur le principe de recherche d'un « optimum de redondance » (Brillouin 1959) : cependant, cette contrainte d'optimisation, cantonnée par la théorie de l'information à la sphère de la transmission technique, doit être étendue à la sphère de la compréhension, étudiée par les cognitivistes. Elle a alors des expressions multiples. Un compromis porte sur l'originalité : l'énoncé doit différer suffisamment des attentes de l'interlocuteur, sous peine d'être un pléonasme, mais il doit rester proche de son cadre conceptuel, sous peine d'apparaître dû à un bruit. Un compromis porte sur la plausibilité : l'énoncé doit être doté d'une probabilité *supérieure* à celles de toutes ses alternatives, il doit

trancher¹ sur elles, mais dans le même temps il doit rester suffisamment *improbable* pour rester intéressant. Enfin, un compromis porte sur l'ambiguïté sémantique : l'énoncé doit être suffisamment clair pour permettre un savoir commun, mais assez ambivalent pour ménager une diversité de points de vue.

La reconnaissance de la dimension épistémique des activités cognitives, qui va de pair avec l'explicitation d'une visée de qualité sur la production de connaissances nouvelles, conduit à renforcer encore cette norme d'*ouverture* portant sur l'encadrement de l'échange. L'obéissance à la tradition, l'intériorisation des normes, le mimétisme conduits seuls sont une « pathologie » de l'apprentissage social. L'ordonnement autour de la qualité des communautés d'échanges de connaissance va par conséquent de pair avec certaines propriétés organisationnelles. Les interactions sociales s'inscrivent dans la logique des seuils de sociabilité décrits par Simmel : les caractéristiques objectives des partenaires ainsi que les sujets « trop intimement personnels » sont exclus de son domaine, et les éléments de présentation personnelle doivent être limités à des sujets qui facilitent l'échange (Beaudoin & Velkovska 2001). Paradoxalement, ce qui soude les « communautés » d'échanges de connaissance, ce sont les propriétés précises qui les soustraient à l'acceptation classique envisagée par Tönnies (1922) : la proximité affective, l'obéissance et la tradition, apparaissent plutôt comme des menaces qui, en augmentant le coût de participation, les détruisent. C'est d'ailleurs une explication majeure au fait que les liens électroniques, plus protecteurs pour la personne, sont le support privilégié par ces collectifs.

Parce que la visée de qualité impose une approche a priori laxiste en terme d'accueil, un problème structurel des communautés de connaissance consiste dès lors à bien régler leur *degré d'ouverture* : comment arbitrer entre l'effet club, qui bénéficie de la grande taille, et l'effet de congestion, qui incite à la clôture ? Quelle taille donner à la base des abonnés de la liste de discussion ? Comment gérer les hors sujets ? Faut-il scinder le forum en deux ? Ces questions récurrentes portent à la fois sur le choix du « diaphragme » reliant la communauté à l'extérieur et sur la gestion des « soupapes » qui permettent, à l'intérieur, de fixer la taille optimum pour chaque compartiment de connaissance. Les paramètres qui déterminent la régulation se répartissent selon trois axes principaux : un *axe de niveau* permet d'opposer une sélection dont le point d'application est « en amont » et passe par la sélection des candidats à une sélection « en aval » qui passe par la sanction après coup ou l'ostracisme en cas de récidive ; un *axe de déploiement* discrimine une sélection explicite qui repose sur des blâmes ou des exclusions, et une sélection implicite qui s'appuie sur des disciplines ; un *axe de ciblage* distingue un mécanisme de régulation qui porte sur les membres de la communauté, à partir de l'identification de *profils*, et un mécanisme de sélection qui porte plutôt sur les textes envoyés, en fonction de *normes objectives* de rédaction.

Tout en étant cruciale, la régulation de l'ouverture se heurte à des difficultés intrinsèques. Les communautés de connaissance sont fondées sur le contrôle latéral par les pairs, et l'acquisition d'autorité y est donc instable et révoquant : de ce fait, les pairs hésitent à endosser les coûts de l'exercice du contrôle car cela a un coût relationnel (Lazega 1999). D'autre part, une modalité importante de sélection est la sanction diffuse (sarcasme, blâme). Or, celle-ci peut facilement se retourner contre l'expéditeur, échappant ainsi à tout pronostic (Ogien 1998). Il n'est ainsi pas rare qu'un rappel à l'ordre s'exprimant sous la forme du sarcasme passe pour un manque de contrôle, du fait d'une critique portant sur le « ton » du sarcasme, dénoncé pour son excès de férocité.

¹ « La phrase qui menace de détruire tous les énoncés (et les carrières) prend la forme conditionnelle “ mais tu pourrais aussi bien dire que c'est... ” et suit une liste d'énoncés également probables. L'issue de cette formulation est souvent la dissolution de l'énoncé en bruit » (Latour 1988 p.258).

L'hypothèse de cet article est (1) que des équilibres stationnaires dans la régulation de la connaissance émergent sur le moyen terme, (2) que ces équilibres résultent d'un accord des colistiers sur le choix des paramètres-clefs, et (3) que ces équilibres stables dépendent étroitement du *contenu* de la connaissance échangée. Pour étudier la façon dont, selon les contenus épistémiques, les arbitrages sur le degré d'ouverture se stabilisent dans des accords différents, le dispositif choisi est l'observation d'une communauté maintenant très ouvert l'arc des échanges cognitifs, entre les conseils d'utilisateurs, les assistances et les coopérations. En raison de sa maturité de fonctionnement à l'aide d'outils exclusivement informatique, nous avons choisi de décrire la communauté Debian, un groupe de 900 développeurs répartis dans une trentaine de pays au niveau mondial. Debian est une distribution complète de logiciels libres, depuis le noyau jusqu'aux applicatifs. Au lieu d'être développée par une personne isolée, par une entreprise commerciale, ou par un noyau fermé, comme les autres distributions de Linux, Debian est codée avec un effort considérable d'ouverture, entièrement réalisée par des bénévoles coopérant en réseau via Internet. Elle fut créée en août 1993 par Ian Murdock, alors étudiant de premier cycle à l'Université de Purdue, à l'origine pour corriger les défauts d'une des premières distributions de logiciels libres de l'époque, qui n'était plus à jour. Pour l'anecdote, son nom provient de la contraction du prénom de sa petite amie (Deborah) et du sien. Financé au départ par la Free Software Foundation, le projet est désormais autofinancé grâce à l'apport de dons, individuels et moraux, d'apports en nature venus d'entreprise, soucieuses de monter des partenariats avec une distribution qui est parfois la seule existante pour certaines architectures, notamment dans les plate-forme ou les portables. Les principaux développeurs de Debian, ceux qui fabriquent ou maintiennent le plus de paquets, bénéficient parfois de clauses à leur contrat de travail qui les autorise à livrer une partie de leur temps à l'amélioration et à la maintenance de la distribution². Du fait de sa taille, 900 développeurs mais environ 1500 abonnés à la liste d'entraide utilisateurs, le problème de cette communauté consiste autant à gérer le niveau de flux autorisé à passer dans le diaphragme qui la relie à l'extérieur, qu'à décomposer son savoir en plusieurs compartiments de connaissance dessinés en fonction d'optimums de taille.

Le succès de la distribution Debian, sur la période considérée, constitue un bon tremplin pour notre analyse. Il va nous permettre en effet d'analyser la façon dont une communauté ayant atteint l'excellence a régulé la connaissance. Selon le type de connaissance échangé (conseils issus de retours d'expériences, contribution à la conception), comment se détermine le choix sur la *taille* du groupe? Comment s'effectue l'arbitrage entre flux à l'entrée et coût de sélection? Comment se stabilise, en fonction du statut du membre, la distribution optimale des temps consacrés à l'aide au novice et à la coopération entre experts ?

Nous envisagerons successivement la gestion de l'arbitrage dans chaque domaine épistémique: l'entraide mutuelle entre utilisateurs dans une première partie; la coopération dans la conception du code et la maintenance de la distribution dans une seconde partie. Chaque partie se consacrera à l'examen des polémiques sur la gestion aux frontières entre ces deux catégories d'interventions épistémiques.

1. Le support utilisateur et le rôle des « clusters »

Le support utilisateur de Debian est un système sophistiqué d'échange d'informations qui regroupe les personnes ayant installé et apprécié la distribution Linux.Debian et venant

² Sur la façon dont l'équilibre de coordination dans Debian est implémenté grâce à des mécanismes institutionnels, cf. Auray 2003.

faire part de leur expérience à d'autres utilisateurs qui s'interrogent sur sa qualité et son utilité. Ce système est ouvert à deux sortes d'actes de langage : d'une part des remarques et conseils entre utilisateurs, d'autre part des suggestions et critiques envoyés par les utilisateurs aux auteurs. Matériellement, il repose sur une liste de discussion (liste Debian Users) ouverte en lecture et en écriture sans restriction, et sur un système de « chasse aux bugs » (Bug Tracking System). Parce que la distribution Debian est un bien complexe, la façon de l'utiliser et de l'adapter à ses besoins demande une acculturation et une aide, et la construction de ce canal répond à cet objectif. Quelles formes prend l'effet club, et quelle est l'économie de l'entraide? Comment s'effectue la sélection sur ce support ?

Initialement, si l'on en reste aux préconisations d'utilisation, la liste Debian Users est apparemment construite selon le "modèle du tableau noir" (*blackboard model*) : le fonctionnement "individu-collectif" se caractérise par des interactions anonymes et publiques, l'ensemble du collectif ayant toujours une sorte de droit de regard et d'action sur les interactions. Cependant, l'analyse sociologique des relations entre utilisateurs sur la période 1997-2001 amène à modérer ce constat. Des sous-structures relationnelles existent à l'intérieur de la liste, marquées par une plus forte intensité de conseils, et elles opèrent partiellement en-dehors de la liste. Ainsi, une partie des aides ne sont pas posées directement sur la liste, mais le sont au sein de chacune de ces grappes (ou « clusters ») privilégiées d'utilisateurs. La possibilité de constituer des grappes est d'ailleurs encouragée par le dispositif logiciel qui relie la communauté, puisque celui-ci ouvre à un canal de messagerie synchrone qui permet à chacun de se constituer des listes prioritaires d'amis avec lesquels il peut interagir en direct. L'appartenance à une grappe d'utilisateurs ne concerne pas tous les utilisateurs, mais elle augmente avec l'intensité de contribution. C'est souvent par une détection fine, au moment de la lecture, des utilisateurs qui ont un niveau technique et des caractéristiques sociales qui le rendent proche de lui (proximité géographique, proximité professionnelle) que l'utilisateur constitue son cluster d'appartenance. Ce choix d'une insertion dans des grappes est peut-être lié à un souci d'efficacité : l'efficacité d'une aide est supérieure quand augmente la proximité culturelle et de niveau technique entre l'aideur et l'aidé, ce qui explique que la transmission dans le cadre de relations de « familiarité » entre aideur et aidé soit une condition du succès de la bonne transmission du soutien pédagogique en matière informatique (Gollac 2001). A l'intérieur d'un cluster, les relations peuvent aller jusqu'à l'établissement de rencontres physiques, mais ce phénomène est rare, et essentiellement limité aux clusters qui cumulent proximités géographique et professionnelle : ainsi, les collègues d'un même site et ayant une Debian ont l'habitude de partager entre eux leurs problèmes et n'envoient que les questions qu'ils ne sont pas arrivés à résoudre dans leur petit groupe (Delsalle 2001). Cette présence, au sein du tableau noir marqué par une publicisation forte de tous les commentaires, d'un modèle de grappe marqué par les « atomes crochus » aboutit à un constat partagé par tous que, au sein du groupe d'entraide entre utilisateurs, les relations sont plus denses qu'à l'accoutumé. « La liste debian fait un peu "village", c'est convivial. De plus, pas mal de gens de la liste se connaissent physiquement » (modérateur debian-user francophone mars 2002). Le nivellement des interventions depuis les clusters jusqu'au tableau noir est renforcé par l'existence de nombreuses sous-listes locales d'entraides, chacune dédiée à une langue d'utilisateurs. Le souci de proximité linguistique allonge les médiations existant entre les requérants et le tableau noir où sont affichées les réponses finales.

Le rejet d'une sélection en amont :

Les « user lists » sont *ouvertes au maximum*, et notamment sont autorisées en lecture (grâce à un archivage public) et en écriture même à des utilisateurs non abonnés. Seule existe une protection contre le postage multiple grâce à des filtres de spam. En effet, les menaces sont minimales sur ces listes, les participants n'ayant pas accès à la ressource critique de Debian, le code. La liste affiche une norme de réticence contre toute spécialisation : elle s'exprime par les formulations des membres insistant sur l'excitation curieuse : “feuilleter” la liste, “tomber sur” un article passionnant - “le tout, au hasard”- et en dernier lieu “s'ouvrir l'esprit”. L'ouverture curieuse est valorisée pour ses bénéfices épistémiques : elle permet de construire les conditions favorables à la découverte. Elle est de plus *morale*ment grandie, comme vaccin contre l'égoïsme, comme l'illustre l'idée baroque d'un contributeur : « là où je voulais en venir, c'était, encore et toujours, sur la spécialisation; la philosophie comtienne (d'Auguste Comte) nous enseigne que cela conduit à un développement excessif du sentiment égoïste. On en pense ce que l'on en veut, pour ma part, je me dis que ça ne doit pas être très bon... » (N.B., debian user fr mai 2001).

Les tentations d'alourdir le coût en écriture sur les listes sont rejetées. Ainsi, le 19 février 2002, un débieniste français propose de canaliser l'entrée en recommandant à tout requêteur qu'il formate sa question en remplissant les champs d'un formulaire avec de nombreuses questions nécessitant la citation de mots-clefs. Il suggérait de retourner automatiquement grâce à un moteur intelligent les réponses à ces mots clefs fournies par les archives des listes, et d'interdire techniquement l'envoi si cette lecture guidée des archives n'était pas faite. Le conseil de ce débieniste était proche d'une préconisation formulée par Van Hippel (1999) et qui vise, en soulageant le répondeur de l'obligation de trier les requêtes, à redistribuer le coût temporel de la participation du répondeur vers le requêteur. Elle vise à relever le niveau d'incitation à contribuer. Néanmoins, sa proposition est rejetée et les commentaires sont négatifs:

« Je ne suis pas sûr qu'un système automatique pour forwarder la question soit une bonne idée. Je préfère que les gens prennent le temps d'écrire eux-mêmes le message. Cependant, la page web pourrait donner un conseil améliorer sur comment envoyer et où» (Raphaël Hertzog, février 2002, liste *Debian Project*).

Cette volonté de minorer le coût en écriture est complétée par un effort pour maintenir très bas le coût d'accès en lecture. « Les newbies tout spécialement ne doivent pas avoir à chercher toutes les archives pour trouver une réponse avant de poster une question » (debian-project, mars 2001). Un effort essentiel est proclamé pour faciliter l'accueil du nouveau requêteur, pour qu'il se sente bien dans la communauté des utilisateurs, et y reste. Cet effort pour leur débrouiller le chemin va au-delà de la simple confection d'une FAQ : il porte sur la volonté de *désengorger la FAQ elle-même*, en la découpant en petits bulletins orientés chacun vers un thème précis. La mise au point de ce procédé résulte d'une discussion nourrie entre les membres du projet. Le 16 mai 2001, il est décidé, pour réduire l'engorgement de la FAQ, de créer une *seconde FAQ*, hebdomadaire, qui s'apparenterait à un bulletin. Mais, à l'époque, la proposition consiste à regrouper dans le bulletin périodique des thèmes de la semaine. La volonté a ensuite finalement été de remplacer la régularité de parution par l'homogénéité thématique, en publiant des FAQ sur des sujets locaux. Ainsi, lorsqu'un fil est récurrent, des membres bénévoles de la liste compilent les meilleurs morceaux des fils passés sur le même thème et rédigent ainsi un texte de synthèse dont la lecture sera suggérée aux nouveaux venus sur le forum, pour désengorger la liste.

Les échanges de conseils sont organisés dans une perspective d'ouverture. L'organisation de la remontée des remarques des utilisateurs auprès des développeurs répond

à un même souci d'ouverture auprès des nouveaux. Classiquement, les analystes soulignent la supériorité du développement distribué. Selon l'adage, « Given many eyeballs, all bugs are shallow ». La littérature ayant analysé dans le détail des systèmes de chasse aux erreurs, canalisant la remontée des remarques, mettent en avant la faiblesse des motivations d'utilisateurs, et, de ce fait, la nécessité cruciale d'une légèreté du coût administratif (que permet bien l'Internet). Ainsi, si on alourdit le processus nécessaire à l'utilisateur pour émettre une critique, on le décourage à en faire et on rend le processus déficient en le privant des remarques de nombreux utilisateurs découragés.

La liste de discussion comme propriété collective :

Les avantages liés à l'ouverture de la communauté ont cependant pour corollaire des effets de congestion. Pour soulager les utilisateurs chevronnés, prisonniers de l'encombrement du trafic par des questions de novices, certains développeurs eurent l'idée de faire une scission, en créant une liste d'utilisateurs chevronnés. L'événement de cette scission est riche d'enseignements. Le 17 août 2001, le modérateur de la liste de discussion des utilisateurs francophones annonce la création d'une liste spécialisée, en fait réservée à des entraides entre utilisateurs chevronnés de la distribution, pour la plupart déjà développeurs. Une vive polémique a alors vu le jour sur la liste historique à propos de la légitimité de cette scission. Assimilée à un "malencontreux fait de dissidence", la création d'une nouvelle liste est-elle un libre choix individuel, ou bien doit-elle être soumise à un vote préalable, comme une "décision" de la collectivité ?

```
« >3 Je trouve d'ailleurs que cette dernière décision a
malheureusement
> été
> prise de façon dictatoriale et autoritaire. Les propos d'une
> arrogance
Ce n'est pas une décision, c'est une requête. Tout le monde peut
demander la création d'une liste si son objet concorde avec les activités
de Debian. (fondateur de debian-french, 18 août 2001).
```

Les leçons de cette polémique sont que la décision de scission a été considérée comme illégitime parce qu'elle a été prise sans recherche préalable de consensus, et surtout sans débat suffisant. Ainsi, parmi les 706 membres en août 2001 abonnés à la liste d'utilisateurs francophones, seuls 21 avaient dit leur accord à la scission. De plus, le choix du mois d'août pour organiser le schisme, en pleine vacance estivale, identifie la prise de décision à un "putsch" témoignant d'un refus des règles démocratiques. Même s'il est modérateur d'une liste, aucun individu ne peut seul s'octroyer le droit de configurer les canaux de communication, qui demeurent propriété collective des membres.

Cependant, le « schismatique » corrige son geste pour le rendre finalement tolérable. Il maintient l'abonnement à la première liste de la totalité des membres inscrits à la seconde. Il se défend ainsi d'être un « sectaire » qui établirait des frontières étanches entre les deux communautés.

```
« L'ensemble des abonnés de debian-french a été abonné à debian-user-
french que je sache, donc cela ne fait aucune différence. Il s'agit juste
```

³ Le propos ici est extrait d'un fil polémique sur la liste historique Debian-French, et expose la position du fondateur de la nouvelle liste, qui est aussi le modérateur de l'ancienne. Les chevrons indiquent la citation du post précédent.

d'une tentative de redéfinir plus précisément les besoins en terme de liste. Personnellement, je n'ai pas forcément envie de recevoir tout le trafic occasionné par les listes d'utilisateurs et je ne vois pas en quoi ma satisfaction sur ce plan va entraîner ton insatisfaction » (N.B., août 2001)

L'action sur l'ouverture ou la fermeture du « robinet d'entrée » dans Debian est donc sévèrement limitée par la légitimité problématique de toute intervention sur le canal de communication, considéré comme propriété collective de Debian. Cette discussion sur le statut de la propriété du canal de discussion fut, sur la période 1997-2001 dans Debian, l'objet d'un seul débat à feu nourri : un débieniste, francophone en l'occurrence, proposa en effet sur la liste debian-devel que soit ratifié l'IRC comme un canal de communication officiel pour les développeurs, et archivées les communications qui s'y déroulent (octobre 2001). Il ressortait de cette discussion que la « propriété collective » de Debian dans son ensemble se limite aux listes qui sont hébergées par des serveurs dépendants de Debian. Elle ne comprend pas les listes « parallèles », très nombreuses entre utilisateurs, qui font du soutien technique Debian sur d'autres infrastructures. Par ailleurs, un excès prescriptif de Debian est redouté par les développeurs, comme le résume la phrase suivante du développeur qui eut le dernier mot dans la discussion : « les gens discutent aussi de debian au restaurant, au bar, dans le sport, sans que la charte Debian doive réguler la police de ces lieux (...) légiférer de là où vous pouvez et ne pouvez pas parler de Debian, c'est la *bureaucratie devenue folle* ».

L'organisation de l'entraide technique auprès des utilisateurs répond donc au besoin de Debian d'ouvrir au maximum l'accueil vers les utilisateurs novices, au prix de solutions ingénieuses incluant les rapports de proximité. Dans des cas ponctuels, la fermeture du canal de communication passe par une sélection en aval plutôt qu'en amont, et passant plutôt par une restriction du canal à un public plus étroit de locuteurs.

La gestion du taux d'ouverture à la critique :

Cependant, le souci d'ouverture se heurte à des limites qui apparaissent d'abord au niveau du système de retours critiques. Une trop forte ouverture entraîne un risque de baisse de qualité des retours de bugs et crée une menace mortelle sur la distribution. Dès 2000, le succès de la distribution génère tellement de nouveaux membres que le niveau moyen de qualité des retours critiques sur le système s'est dégradé, mettant en péril le cycle de correction dans son ensemble. Ainsi, beaucoup d'anciens développeurs tirent la sonnette d'alarme en liant directement la défaillance critique à la *popularité* de Debian : cette thèse que Debian récolte « les rançons de la gloire » est bien illustrée par cette position de Bdale Garbee :

« J'aimerais voir le niveau d'intelligence (clue level) des rapports de bogues s'élever, et bien sûr j'aimerais que chaque rapport de bug inclue une suggestion de patch. Mais, je pense qu'un des prix du succès est que, au fur et à mesure que notre base d'utilisateurs s'élargit, nous allons rencontrer de plus en plus de gens pour lesquels l'informatique est un outil, non une obsession... et cela signifie qu'il est inévitable que le bruit du système de bug va s'amplifier » (debian-devel juin 2001).

D'un côté, il faut qu'il y ait suffisamment de monde qui envoie des rapports de bugs (pour breveter le système à l'épreuve de l'acuité de regard des utilisateurs), mais de l'autre il n'en faut pas trop sinon on abaisse la qualité des rapports de bugs et on étouffe le dispositif avec du bruit. Dès lors, éclate une polémique : quel est le niveau d'ouverture du serveur de bugs qui faciliterait son désengorgement?

C'est en juin 2001 que le problème est explicité dans les débats politiques de la communauté. Les indicateurs statistiques sont en fait au rouge depuis fin 2000, mais le débat a mis six mois à s'établir. Debian est en effet confronté depuis fin 2000 à une hausse brutale de son nombre de *bugs pendants* (en attente de correction), qui se met brutalement à progresser plus vite que le nombre de paquets : en juin 2001, le nombre moyen de bugs pendants par paquet est de 2,5 ; il dépasse le seuil critique de 3 en décembre 2001. Certes, le système de recherche de bogues de Debian est objet de fierté et d'un soin jaloux, car il constitue une énorme quantité de savoirs⁴. Le cent millième bug est fêté le 7 juin 2001 : quelques développeurs l'ont fêté sur IRC avec un bot spécial d'un membre, Joey Hess. Mais l'envolée de la part des bugs non corrigés, à partir de fin 2000 (de 15% en janvier 2000, il passe à 20% en 2001 et à 28% en janvier 2002), devient sujet de préoccupation dès juin 2001.

Cet accroissement de la présence des bugs pendants est certes partiellement causé par un effet d'optique : il traduit la hausse de vigilance de la part des utilisateurs, qui, étant plus nombreux, débusquent plus facilement les perles cachées. Il n'est toutefois pas un pur artefact de comptage : il s'ancre sur un retour de réalité, que l'on peut induire du rapprochement des expressions publiques de troubles, par les usagers les plus divers.

La principale raison invoquée à la hausse des bugs pendants est très vite l'ouverture à un trop grand nombre d'utilisateurs du système : le manque de qualité de la documentation des rapports d'enquête est expliqué par la chute du niveau des utilisateurs contributeurs. Le rapport de bug semble tendre de plus en plus vers le mauvais côté de l'échelle suivante : rustine testée et qui fonctionne, diagnostic complet et explication du problème, enregistrement des événements du bug, description basique du problème, commentaire disant qu'il y a un bug dans le paquet, erreur de l'utilisateur mal diagnostiquée.

« Si nous avons de plus en plus d'utilisateurs, ou de plus en plus de novices, ou de plus en plus d'utilisateurs qui veulent une « information appliance » à la place d'un ordinateur, je ne peux que prédire que cette situation empirera. Peut-on y faire quelque chose ? Peut-être qu'il faudrait d'abord encourager les gens qui peuvent seulement produire [le bas de l'échelle] à discuter d'abord dans une liste ou avec des membres de leur LUG local ? » (A.T., debian-devel juin 2001)

Les incitations à biaiser les rapports de bugs :

Le problème de l'envolée des bugs est aggravé par les réactions des participants qui s'ajustent à cette nouvelle situation : l'accumulation de bugs pendants entraîne logiquement une surestimation des déclarations de gravité du bug faites par les rapporteurs.

« Je ne fais que relater mon expérience de plus de deux ans de maintenance. La surévaluation des bugs est parallèle à leur accumulation : un peu comme si on allumait un projecteur juste au-dessous d'un autre, les bugs sont quasiment ignorés. C'est seulement si l'on incrémente leur gravité qu'on a quelque chose de voir quelqu'un s'en occuper » (S.P., octobre 2000, debian-devel).

⁴ La collection de ces bogues, chacun rapportant une description et étant chargé d'histoire, acquiert une valeur patrimoniale. Régulièrement sont relancés des concours pour découvrir le « bug le plus vieux », le « bug le plus célèbre », le « bug le plus amusant » etc... Ces occasions de découvrir son patrimoine sont autant d'incitations à épousseter le système en fixant au passage les plus simples qu'on a pu trouver.

En surestimant la gravité, les rapporteurs ont ainsi l'espoir de capter plus l'attention du mainteneur. Par ce biais, ils déclenchent une spirale inflationniste entretenue par des comportements de mimétisme et de surenchère. L'élévation de la gravité des bugs semble avoir aussi pour condition la difficulté objective qu'a un utilisateur pour évaluer l'importance d'un bug qui l'affecte personnellement - la plupart des bugs paraissent toujours plus sérieux envers les personnes qui en sont victimes. Ce biais peut difficilement être réparé : cela supposerait une supervision du rapport par des tiers, ce qui semble précisément impossible dans une situation d'excès de rapports. Elle engendre une crise de fiabilité du système.

La véritable solution envisagée par une majorité de développeurs consiste à encourager une modalité plus « collective » de rapport de bugs : poser sa question d'abord à un groupe de pairs et à des anciens, au sein d'un « cluster » donc, avant d'envoyer le rapport. La mise en discussion du soupçon de bug sur une liste user locale apparaît la meilleure solution, et la discussion aboutit à un accord pour qu'un avertissement de la messagerie du système de chasse aux bugs mentionne cette exigence. La nécessité est de maintenir cette exigence à l'état de recommandation, sans la rendre obligatoire, sous peine de perdre complètement les rapports d'utilisateurs ("un post de novice qui n'obtient pas de réponse et l'utilisateur ne le renouvellera jamais"), ou de perdre la discussion.

Les réticences communautaires devant le contrôle de l'accès :

Comment arbitrer entre ouverture et qualité minimale des rapports de bugs ? Vaut-il mieux sacrifier des critiques d'utilisateurs pour sauvegarder la qualité ? Ou bien, à l'inverse, faut-il accepter la dégradation inévitable des critiques car elle permet de maintenir la richesse d'avis divers ? La question a surgi de manière vitale pour la communauté, et cela rend d'autant plus saisissantes les conclusions qui ont été apportées : malgré ce péril mortel, les membres ont réitéré leur confiance pour un maintien de la priorité à l'ouverture et pour un rejet de la sélection en amont.

Dans ce débat, la polémique s'est déplacée sur plusieurs terrains. Un premier terrain consistait à déterminer le seuil optimal de formalisation acceptable dans la rédaction des critiques. Elever le degré de formalisation permettait d'améliorer la lisibilité et la qualité des rapports, mais ce gain avait comme contrepartie d'une part une lenteur de rédaction, et entraînait le déplacement d'une partie de l'effort de conception dans Debian vers un effort de rédaction. D'autre part, la formalisation risquait d'avoir un effet d'éviction sur les chercheurs de bugs, découragés par la complexité du formulaire. Elle risquait de priver la communauté de volontés. Un autre terrain consistait à déterminer la tâche devant être privilégiée par Debian : doit-elle être la tâche de *critique*, ou bien ne faut-il pas privilégier plutôt la *conception* ou la *correction* ? A quoi sert-il alors d'améliorer la qualité des critiques, si cela ponctionne les énergies des concepteurs et des correcteurs ?

Ainsi, en octobre 2000, la discussion est lancée sur l'opportunité de renforcer la sélection à l'entrée des rapporteurs de critique. Elle est tranchée dans le sens d'un refus d'une trop lourde normalisation des rapports. L'occasion de ce refus a été fournie là encore par une proposition d'un développeur, le 30 juin 2001. Il voulait rendre obligatoire, pour toute critique, le remplissage de trois champs : un champ de *sévérité* indiquant la gravité du bug, un champ de *statut* indiquant son niveau de réalité (une seule occurrence, reproductible, vérifié, doté de correction), et un champ de *domaine* où il fallait écrire une opinion sur l'origine supposée du bug. Il souhaitait ainsi inclure dans tout rapport de bug des drapeaux obligatoires, sous la forme d'un formulaire avec des cases à cocher et des questions ouvertes. Sa proposition s'est heurtée globalement à la critique, et la discussion fut tranchée dans le sens d'un refus de la formalisation excessive.

Porte-parole de la position majoritaire, le *project leader* Bdale Garbee déclare ainsi en août 2001 : « Alors que je ne suis pas contre des changements précautionneusement considérés, j'aimerais dire qu'une des raisons pour lesquelles le BTS de Debian a eu tant de succès à la fois pour nous et pour d'autres groupes qui travaillent avec est qu'il a été ajusté à notre usage... et que précisément, il n'est *pas* comme les autres systèmes de rapport de bugs. Il me semble que le temps passé à remplir les informations de champ du formulaire serait mieux dépensé à travailler effectivement à résoudre les bugs directement. Même notre mécanisme de drapeau actuel n'est pas très utilisé.. et si c'était de mon ressort, nous aurions encore moins de niveaux de sévérité définis. Trop de choix augmente la prise de tête (the process overhead) pour ceux qui sont fondamentalement des bénévoles à rendre meilleur le monde des bits » (debian-devel).

La solution consistant à *élever le coût d'accès* au système a été, malgré la situation d'engorgement, écartée. La priorité donnée à la collecte des remarques semble transcender même les situations les plus marquées de pléthore.

2. La conception des logiciels et la gouvernance des équipes:

Une autre structure de coordination organise le travail de *conception*. La conception comprend trois types d'activités cognitives relativement autonomes les unes par rapport aux autres : l'activité de *programmation* qui s'apparente à la phase de créativité proprement dite, une activité de *vérification* qui s'apparente à du test, et une activité de *maintenance* qui comprend les mises à jour liées à la modification d'autres éléments du système. L'effet de la taille sur la qualité est analogue dans ces trois cas : une certaine amplitude est nécessaire à la qualité du projet (effet club), mais au-delà d'un seuil la qualité se dégrade rapidement (congestion).

La nécessité de réguler la taille au niveau de la conception est vitale car il existe une pression naturelle à l'entrée sur le groupe, dès lors qu'il a atteint un équilibre convenable de fonctionnement. Ainsi, si l'on suppose qu'une communauté de connaissance en bonne santé est celle qui a maintenu un niveau élevé de qualité, il est inévitable qu'elle entraîne un afflux de candidats. En effet, la qualité élevée de l'objet *propage tous azimuts* la renommée de la communauté, ce qui attire vers elle un double flot de volontaires. Un motif d'adhésion est l'intérêt sincère pour participer à la dynamique d'apprentissage du bien d'expérience de qualité dans ce groupe prestigieux. Un autre motif d'adhésion est la volonté de réfléchir vers soi une part du prestige du groupe pour se constituer une réputation professionnelle. Dès lors, un problème structurel des communautés de connaissance est la *gestion de leur ouverture en amont*.

Les quatre régimes typiques de fonctionnement :

Les effets de congestion sont beaucoup plus prégnants au niveau de la conception que des échanges entre utilisateurs. L'existence de « seuil » signifie que la dégradation de la qualité n'est pas linéaire mais correspond à un changement de braquet brutal lorsqu'est franchi un certain niveau. Mais le terme de « congestion » recouvre une réalité complexe. Quatre seuils correspondant chacun à un point de rebroussement particulier du niveau de qualité peuvent être distingués. Au-delà d'un premier seuil, l'excès de développeurs multiplie dramatiquement les problèmes de coordination et le coût d'administration. Un deuxième seuil

est lié à l'apparition de développeurs tellement médiocres qu'ils contaminent l'ensemble du système.

« Serais-tu d'accord que le péquin moyen (sans offenser personne) soit *responsable* d'un soft qui pourrait être très sensible sur ton système ? (...)Comment être sûr que personne ne tente d'envoyer un troyen dans la distribution via des canaux publics ? »

Un troisième seuil est lié à l'apparition d'un *risque de malveillance*. L'apparition de ce risque est liée à la concomitance de deux phénomènes : avec l'augmentation de la taille s'accroît l'anonymisation ; parallèlement, l'augmentation de taille permet une hausse de la dimension du service fourni, couvrant une gamme plus large de fonctions : cela développe les usages professionnels. Dès lors apparaît la nécessité de se couvrir contre ce nouveau risque. Le risque d'accident, qui s'exprime par la narration de scénarios alarmistes, a une force rhétorique qui surclasse la fidélité aux grands principes, qui n'est positivement évaluée qu'en référence à des situations d'usage plus banales. L'accident est plus facile à retenir car il repose sur une heuristique de la disponibilité : son exemple frappe plus facilement les esprits que celui d'une routine ordinaire.

Enfin, un quatrième seuil est lié au blocage des *mécanismes de solidarité* qui fondent l'efficacité de la communauté épistémique. Une taille excessive empêche la démocratie directe de fonctionner et remet en cause le modèle républicainiste (Pettit 1999) et sa conception très pointilleuse de la liberté comme non-domination. Une polémique instiguée par Branden Robinson en mars 2001 l'illustre : rappelant la Constitution⁵, il rappelait que le seul moyen de remettre en cause une « rule » du chef de projet ou du comité technique, c'est une « résolution générale » de l'ensemble des développeurs. Ainsi, pour qu'une proposition soit adoptée, elle doit d'abord être considérée comme soutenable et pour cela être secondée par un quorum de « la motié de la racine carrée du nombre de développeurs » ; puis, elle doit être acceptée par vote après avoir respecté une période de discussion active de 3 semaines. La publicité et la longueur du débat sont des exigences centrales, dans ce modèle, pour que chaque programmeur puisse faire son choix. Or, avec la hausse de la taille, la démocratie directe est devenue un luxe coûteux du fait de l'excès des saisines et de la pollution du serveur par des propositions multiples.

La sélection des candidats développeurs comme élément central de la régulation:

La taille est une variable névralgique pour le travail de conception. La hausse en taille des effectifs exerce des effets indirects à la *baisse* sur la qualité : à moyens de sélection constants, cette hausse des effectifs entraîne mécaniquement en effet une baisse de sévérité de la sélection des membres. Dès lors, elle doit être régulée. Le seul moyen de contrecarrer la liaison dramatique entre la hausse de la taille et la chute de la qualité est un effort accru de sélection. Si l'on examine le fonctionnement de la communauté qui nous sert de cadre d'observation, la communauté Debian, sur la période 1997-2001, la choix a constamment été de réaliser une sélection en amont, portant sur les membres : le tri des candidats. Ce choix d'une sélection en amont se justifie pour des raisons épistémiques : du fait de la difficulté inhérente à la sanction dans les organisations à contrôle latéral, il est très difficile de sanctionner publiquement un membre à part entière (Auray 2003). Dès lors, la régulation de la taille se transforme presque toujours, dans les débats, en un arbitrage sur le *montant d'effort* à consacrer à la tâche de sélection. Faut-il augmenter l'effort de sélection au risque d'imposer sur chaque membre un lourd tribut, ou bien faut-il à l'inverse maintenir bas cet effort, au

⁵ <http://www.debian.org/devel/constitution.fr.html>

risque de diminuer le niveau de compétence des nouveaux membres de la communauté ? Quel est le degré d'altruisme intracommunautaire exigible des membres, pour qu'ils consacrent du temps à la qualification du groupe ?

Les conditions à la vitalité de la communauté :

Dans ce dilemme, un paramètre central est très vite apparu : la *vitalité* de la communauté. En ce sens, deux écueils sont à éviter, qui se dégagent de manière claire du très grand nombre de discours initié par cette polémique sur le mode optimal de candidature. D'une part, un effort trop lourd de sélection crée une chute de vitalité. Dans certaines conditions, l'exigence de qualité en effet ralentit le cycle de sélection, et aboutit à l'éviction de candidats compétents qui refusent d'être traités de la sorte. La prise de conscience de la tension entre qualité et vitalité est récente, car pendant longtemps les deux exigences étaient supposées compatibles : la condescendance avec laquelle étaient traités les nouveaux était alors assimilée à un rite de bizutage, et établie au rang de cibles optimum : forgeant l'humilité, testant la motivation. Si la sélection est trop lente, ce qui est mécaniquement le cas quand la pression à l'entrée s'accroît, cela finit par bloquer la dynamique communautaire en décourageant les vocations : lorsque le processus de sélection est très lent, voire nul, les meilleurs candidats fuient. C'est ce qui s'est passé de octobre 1999 à février 2000, suite à une décision du chef de projet Wichert Akkerman. Cette auto-régulation cependant n'est pas souhaitable, car elle aboutit à une anti-sélection : les meilleurs candidats fuient, ce sont les moins bons qui restent. Un afflux trop faible d'entrées génère un départ massif des anciens développeurs.

Symétriquement, un effort trop léger de sélection crée un afflux de nouveaux venus de niveau médiocre, engendre une baisse de qualité sur les paquets, et génère un découragement des anciens développeurs, qui quittent la communauté. Un afflux trop important d'entrées génère un départ massif des développeurs. C'est donc entre ces deux écueils de l'afflux trop important et de l'afflux trop faible que se coordonne l'activité de régulation de la connaissance.

Tableau 1. Etat des candidats (mai 2001, sur un nombre total de 900 développeurs)

<i>phase</i>	nombre	délai en jours (mode)	délai en jours (médiane)	délai en jours (moyenne)	délai en jours (maximum)
en attente de sponsors	13	0	1	8	37
en phase d'oral face au jury	106	10	156	144	460

Il en résulte un choix de maintenir un engorgement, ainsi que l'illustre le tableau 1 ci-dessus : un nombre structurel de 1,5% des développeurs sont en attente de parrain, et donc clairement stoppés dans le processus d'adhésion. Loin de correspondre à un déséquilibre, ce chiffre structurel permet de maintenir une bonne vitalité du groupe, loin des deux écueils que l'on vient de mentionner.

La détermination d'un grain optimal pour la production:

Une grande partie des arbitrages concernant la taille, au niveau de la conception, ont à voir avec la détermination du niveau de coordination auquel doivent être traitées les diverses tâches : doit-on privilégier une organisation avec beaucoup de petites *task forces* réalisant chacune de petits « modules », ou bien avec peu de grosses *task forces* mais réalisant chacune un travail fortement intégré ? A quel niveau doit-on fixer la taille et la division des groupes, pour maîtriser à la fois le coût de coordination et le délai de conception ? Ce problème d'obtention de la bonne *limite de décomposition* des différentes unités qui constituent une équipe structurée est central dans tout projet d'envergure.

Ces questions ne sont pas nouvelles. Deux grands types de travaux peuvent être distingués. Les premiers se concentrent sur la décomposition « horizontale », qui amène plusieurs équipes à travailler en parallèle sur différents secteurs d'un même gros projet. Considérant par exemple le cas de la réalisation d'un projet d'envergure, le noyau Linux, Tuomi (2000) montre que la spécificité de l'ingénierie informatique est qu'une prime est donnée à la « modularité » - c'est sa modularité qui explique le succès de Linux sur Hurd-mais qu'existe une forte instabilité des modules et que les modules les plus petits sont voués à fusionner avec d'autres pour survivre.

Le second type de travaux se consacre à la décomposition « verticale », qui amène à déterminer quelle est la date optimale pour faire la « release » d'un produit : doit-on diffuser le travail alors qu'il est encore buggé mais en étant très réactif aux usagers et en faisant beaucoup de releases, ou bien au contraire diffuser après une vérification ? Comment doit être le profil d'incrémentation du produit : à pas haut mais lent, ou à pas bas mais rapide ? Le débat entre ces deux positions est vif en informatique, et loin d'être tranché : une thèse soutient que pour optimiser le produit, il faut diffuser tout de suite et mettre à jour souvent (modèle du bazar) ; une autre thèse défend au contraire que, pour désengorger la fabrication, il faut corriger avant de diffuser. Yamauchi et alii (2000) montrent que, contrairement aux règles de coordination usuelles dans les projets à plusieurs, dans les projets informatiques la déclaration publique de focalisation sur une tâche intervient après sa réalisation complète.

La négociation des frontières vis-à-vis de l'amont : création et test

Dans le cas de Debian, l'arbitrage sur la délimitation entre ce qui ressortit à la création (et doit être mené dans une task force séparée) et ce qui ressortit au test (et exige d'être discuté publiquement par tous les développeurs) a été l'occasion de polémiques virulentes. L'une de celle-ci a émergé, comme toujours, à l'occasion d'une proposition hérétique d'un développeur qui voulait objectiver par un indicateur facilement extractible la qualité du travail des développeurs responsables d'un paquet. La proposition consista à instituer un « index de vitalité » pour évaluer l'activité des développeurs :

« Package's index could be computed using the following factors: a. How many bugs have been resolved by the current maintainer? b. How long has the package been maintained by the current maintainer? c. How many releases have the current maintainer made? d. How long ago was it that the package was last updated by the current maintainer? Those factors can be molded into a value thusly:

$$\text{vitality} = ((a + 1) * b * c) / d$$

•(a + 1) is because otherwise packages that haven't had any bugs fixed will all have zero vitality. Also, I imagine the vitality as being for a certain package with a certain maintainer; if a package switches maintainer the vitality score should be reset (as the new

maintainer can be better / worse than the previous one) » (8 janvier 2002, debian-devel)

Cette position se heurta à de virulentes réactions, visant à bien distinguer « vitalité » et « volatilité » d'un paquet. Ainsi, les arbitrages rendus par un autre développeur, responsable du projet Debian-Junior, privilégient l'effort de conception, et dénoncent l'accommodement à l'erreur qu'introduit un tel index mettant l'accent sur la réactivité.

« I'm sure nobody thinks doing any of the above is a bad thing, but the above tasks fall under the general label "accommodation for failure". While these tasks are all necessary evils which I will do my best to support, I think task number one needs to be an effort to reduce bugs, failed builds, etc. in the first place. The model that I am working with in Debian Jr. is to try to find the best possible people to work with, including some promising new maintainers, work directly with them via peer review as they build their packages, and learn together how to build quality in from the beginning. Education is going to take us much farther than fixing problems after the fact » (B.A., 11 janvier 2002, debian-devel).

Debian est organisée en sous-projets modulaires qui repoussent vers le terme le moment de communication. L'exemple de l'interaction entre l'un de ces sous-projets et le niveau d'ensemble offre une représentation fidèle de la façon dont est externalisée, en amont, une bonne partie du travail de conception. Comment s'organise, à l'inverse, la communication vers l'aval, vers l'utilisateur final ?

La négociation des frontières vers l'aval : le cas de la traduction

Les frontières vers l'aval du projet Debian sont soumises à discussion. Une occasion d'embrasement fut la proposition d'inclure dans la distribution une traduction en plusieurs langues, proposition qui aboutira en fait à un rejet. Cela montre la conception restrictive de la tâche employée par les membres du projet.

A partir du printemps 2001, face à la diversification géographique des usagers, est projetée l'internationalisation de Debian : il s'agit de traduire non seulement la documentation des logiciels, mais aussi leur interface avec l'utilisateur, les séquences d'installation, les boîtes de dialogue, les menus, les messages d'erreur. La tâche est complexe car elle suppose un tripatouillage du code des logiciels. Traduire les boîtes de dialogue par exemple suppose de regarder dans les scripts d'installation, ce qui a des conséquences potentiellement dommageables à des endroits très éloignés de là où porte l'intervention initiale. Globalement, à cause de la concision de l'anglais, la traduction crée un risque critique de dépassement de la limite en taille des chaînes de caractère.

Le 9 août 2001, par un long envoi sur plusieurs listes, Michael Bramer annonce officiellement son projet de rendre polyglotte la prochaine version diffusée. Cette déclaration suscite une violence critique, car elle exacerbe la crainte des programmeurs, bien réticents à l'idée de risquer d'entamer deux de leurs ressources les plus rares : le temps et l'espace. D'une part, le projet de traduction exacerberait la pénurie de temps disponible des développeurs bénévoles. En effet, pour que le projet soit viable, il faut que la traduction soit élevée au rang central. Chaque fois qu'un mainteneur finit un paquet ou une mise à jour, une traduction pour chaque langue doit lui être retournée pour qu'il l'insère dans son paquet à un répertoire approprié. Il reçoit une notification automatique qui l'en avertit. La très grosse concentration de la maintenance de paquets – les développeurs pivots dans Debian contrôlent jusqu'à 60 paquets chacun – renforce l'animosité de certains membres du premier cercle.

Beaucoup de développeurs se sont ainsi plaint en identifiant les messages du système de traduction à du courrier non sollicité : certains renvoyaient les mails du système de courrier

sur la liste des développeurs et feignent d'engager la procédure qu'on applique habituellement pour les informations publicitaires qui envahissent les boîtes aux lettres :

Renvoyant sur la liste un extrait du message de notification du système de Bramer, un développeur ajoute : « Ceci est un mail automatique que je n'ai pas requis : je considère cela comme du spam. Enlevez-moi immédiatement de vos listes. Je suis plutôt agressif avec les mails de spam non garantis ». Un autre : « aux USA, des lois prohibent le courrier non sollicité ».

La critique dérape en colère. Les développeurs mettent beaucoup de cruauté contre Michael Bramer : des attaques personnelles subtiles sont tentées, alors qu'apparemment le sujet est maintenu :

```
« Mercredi, Sep 05, 2001 à 08:46:12PM +0200, Michael Bramer a écrit:
> The maintainers need not do anything. Maybe he don't
> know translation. The user only use this. This need
> only the translators.
```

Pendant qu'on est sur ce sujet, pouvez-vous trouver quelqu'un pour traduire vos mails dans un anglais compréhensible ? »

Cette férocité attire à Bramer des défenseurs, notamment ses compatriotes allemands (Schulze, Atterer, Christian Kunz), qui tentent bien de ramener le calme :

```
« Adam, je pense que tu t'empportes un peu vite. Comme il a été noté par
d'autres, ce "spam" est en fait le résultat de quelqu'un qui dépense son
temps pour améliorer ton paquet ».
```

Mais, progressivement l'espace de discussion devient un brasier difficile à éteindre.

La situation a connu une escalade le 4 octobre 2001 avec la proposition de résolution générale d'exclusion à l'encontre de Michael Bramer. Malgré l'agressivité, le requérant est secondé par quelques pairs.

Cherchant le compromis, Bramer propose le filtre d'une liste sur laquelle on s'inscrit pour ne pas recevoir les paquets. Il s'abonnaient à un filtre si ils ne souhaitent pas recevoir de mail concernant la mise à jour pour d'autres langages de leurs paquets (opt out). Les mainteneurs ont trouvé cela trop intrusif et ont proposé à l'inverse la création d'une liste où le mainteneur fait la démarche volontaire de s'inscrire s'il veut recevoir les paquets (opt in).

Mais, à la suite de ces avancées conciliantes de Michael Bramer, prêt à rechercher des compromis, de nouveaux développeurs se plaignent d'une nouvelle menace : même si elle ne coûte rien au développeur, la traduction va signer une perte de contrôle du mainteneur sur son paquet, et serait donc une *usurpation*. Le mainteneur selon eux doit garder la pleine responsabilité sur son paquet.

```
« Cela dessert Debian de vouloir prendre en otage des parties du paquet du
mainteneur, en lui en extorquant le contrôle ! Or ceci, comme vous le
noterez, est la prémisse fondamentale de la proposition de Michaël Bramer.
Il ne veut pas voir le mainteneur engagé, excepté s'il reste coopératif et
accepte d'être envahi par des mails lui notifiant que son paquet a été
modifié ».
```

A leur tour, les pro-traduction disent qu'une traduction n'est pas une modification de fonctionnalité d'un paquet, juste du texte en plus, donc qu'en le déchargeant des traductions on ne porte pas atteinte aux responsabilités du mainteneur. Il faut distinguer la description « canonique » et les autres, qui ne sont que ses reflets sous un autre éclairage, et qui ne font

pas partie du paquet, qui n'en sont que des émanations : dès lors, selon cette minorité de débianistes, la traduction n'est pas une partie nécessaire du paquet.

La majorité des débianistes a donc deux exigences : d'une part ne plus être dérangée par les traducteurs, d'autre part garder le contrôle sur leur paquet. Une proposition, suggérée par un développeur, est de créer un sous-répertoire pour les traductions dont le mainteneur souhaite garder le contrôle, et qui correspondent aux langues étrangères qu'il maîtrise, un autre sous-répertoire pour celles dont il accepte de ne rien savoir.

Qu'est-ce qui fait que la traduction est considérée par de nombreux développeurs comme une attaque mortelle ? Elle abaisse drastiquement le niveau d'implication exigé pour faire partie de Debian. En effet, l'existence d'un service de traduction suppose l'arrivée dans Debian de « relecteurs de traduction » qui les notent et corrigent. Or, la relecture repose sur un faible engagement (5 minutes par semaine), et fait donc entrer dans le projet central des personnes faiblement mobilisées, rompant l'équilibre de l'ensemble.

Conclusion

L'étude sociologique d'une communauté épistémique nous permet de montrer que la gestion de l'ouverture s'effectue selon des modalités très différentes selon les contenus épistémiques. Tout d'abord, la gestion des conseils d'utilisateurs répond à une priorité d'abaissement des barrières d'entrée, au prix de solutions ingénieuses incluant les rapports de proximité. Les solutions visant à augmenter la sélection en amont sont rejetées. La priorité donnée à la collecte des remarques semble transcender même les situations les plus marquées de pléthore. A l'inverse, dans le travail de conception, la sélection s'effectue plutôt en amont. Pour rendre possibles augmentation de la taille et augmentation de la qualité, Debian met l'accent sur un effort partagé de contribution à la sélection. Pour éviter au maximum les problèmes de taille, le groupe choisit de repousser le plus possible vers son terme le moment de communication des compartiments de connaissance. Il organise ainsi une gestion du « grain » qui privilégie l'approche modulaire et déconcentrée, externalisant vers l'amont une bonne partie du travail de conception. Symétriquement, tout ce qui concerne la gestion du rapport avec l'utilisateur final est externalisé, comme l'atteste l'exclusion retentissante des tâches de traduction hors des obligations de maintenance des développeurs.

La régulation se fait en amont pour ce qui concerne la conception, et en aval pour ce qui concerne l'utilisation. Mais, au-delà de ce contraste fort, deux séries de conclusions doivent être mises en évidence. D'une part, un rôle central est assuré, pour le fonctionnement de la communauté, à plusieurs sous-groupes qui sont faiblement visibles sur la liste. Ainsi, dans l'échange de conseils, les « clusters » d'utilisateurs constituent un niveau d'intervention et de coopération privilégié. A l'inverse, dans le travail de conception, ce sont les « grains » réunissant autour d'une tâche homogène une équipe de concepteurs qui sont privilégiés. Une grande partie de la communication sur le « blackboard » se trouve ainsi déplacée vers ces deux sous-structures relationnelles. Des individus attirés jouent le rôle de jonction entre chacune de ces sous-structures et le groupe général : ce sont les porte-parole de clusters et les managers de projets.

Un deuxième type de remarques concerne la superposition problématique des mécanismes opposés de sélection qui affectent utilisateurs et développeurs. Cela génère des problèmes de compatibilité qu'il est intéressant d'analyser. Deux métaproblèmes sont apparus particulièrement. D'une part, il existe un déséquilibre fort entre les vitesses d'acceptation de nouveaux utilisateurs et de nouveaux développeurs. Les nouveaux utilisateurs sont acceptés

très rapidement, tandis que les nouveaux développeurs sont très longs à être acceptés. Ainsi, il y a tendanciellement, dans toute communauté complexe réunissant ces deux réservoirs, un problème de charge de sélection des nouveaux développeurs. Si le groupe a une bonne qualité, il attire de nombreux utilisateurs. Ceux-ci ont tendance à vouloir devenir développeurs. Ils sont freinés dans l'engouement par le frein sélectif nécessaire au maintien de la qualité de la conception. Ce frein sélectif doit être tenu avec tact pour ne pas entraver, ni vers le bas, ni vers le haut, la vitalité de la communauté.

Une catégorie passionnante de métaproblèmes concerne la relation de dépendance entre les paramètres objets du contrôle communautaire. Une articulation forte semble s'établir entre le rythme des entrées dans la communauté et celui des sorties. Une loi semble même avoir été dégagée : lorsque l'afflux est trop faible, et lorsqu'il est trop élevé, la vitalité chute et la communauté se meurt. Mais, entre ces deux écueils, il semble exister seulement deux rythmes d'afflux correspondant à un équilibre stable. Le premier correspond à un afflux « soutenu » et correspond un renouvellement important de la communauté. Le second correspond à un afflux « parcimonieux », et va de pair avec une appartenance à la communauté plus durable. Il semble qu'on ait le choix entre seulement deux types de communautés stables, l'une à fort renouvellement marquée par une génération rapide de nouveaux membres, l'autre à faible renouvellement marquée par un maintien plus durable des anciens.

bibliographie

Auray, N., 2003, « Le sens du juste dans un noyau d'experts. Debian et le puritanisme civique », in Conein, B., Massit-Folléa, F., Proulx, S., *Internet. Une utopie limitée*, Presses Universitaires de l'Université du Québec à Montreal.

Beaudoin, V. Velkovska, J., 2001, « Présentation de soi et structures d'une communauté sur Internet »,

Brillouin, D., 1959, *La science et la théorie de l'information*, éd. J.Gabey, Paris.

Ciborra, C., 2001, « In the Mood for Knowledge : a New Study of Improvisation », working paper, Department of Information Systems, London School of Economics.

Delsalle 2001, *Coopération et interdépendance cognitive autour de Debian User French*, mémoire de DEA sous la direction de Bernard Conein, Lille III.

Jones, P., 2000, « Brook's Law and open source : The more the merrier ? Does open source development method defy the adage about cooks in the kitchen ? », *Developer Works* 21.

Lakhani, K., van Hippel, E., 2000, « How Open Source Software Works : Free « user to user » assistance », MIT Sloan School of Management, working paper #4117.

Latour, B., 1998, *La vie de laboratoire*, La Découverte, Paris.

Lazega, E., 2000, « Rule Enforcement Among Peers : A Lateral Control Regime », *Organization Studies*, 21/1, pp.193-214.

Pettit, Ph., 1999, *Republicanism. A theory of freedom and government*, Oxford, Oxford University Press.

Tönnies, F., 1977 [1922], *Communauté et société. Catégories fondamentales de la sociologie pure*, trad. fr. J.Leif, PUF, Paris.

Tuomi, I., 2000, « Internet, Innovation, and Open Source. Actors in the Network », *First Monday*, 6, pp.17-41.

Yamauchi, Y., Yokozawa, M., Shinohara, T., Ishida, T., « Collaboration with Lean Media : How Open-Source Software Succeeds ? », 2000, ACM Conference on Computer Supported Cooperative Work, décembre, Philadelphia.
