

Vers une société contributive de pair à pair -2

Et si le pair-à-pair devenait le modèle et le moteur d'une nouvelle organisation sociale ? – Deuxième volet de la réflexion de Michel Bauwens ([si vous avez raté le début, c'est par ici](#)).

Source : [Blueprint for P2P Society par Michel Bauwens](#)

Traduction Framalang : Fabrice, goofy, jums, CLC, avec l'aimable contribution de Maïa Dereva.

2. Les relations entre la communauté et la coalition d'entrepreneurs

Quelles sont les relations entre cette coalition d'entrepreneurs et les communs dont ces entrepreneurs retirent leur valeur ? La coalition subvient aux besoins vitaux des « commoners » et soutient parfois financièrement l'institution à but lucratif. IBM, par exemple, verse un salaire aux développeurs/commoners qui contribuent à l'environnement Linux ainsi que des aides à l'association à but non lucratif (la Fondation Linux). Ainsi, les coalitions entrepreneuriales co-produisent et financent les biens communs sur lesquels leur succès est bâti.

Il est vrai qu'en agissant de la sorte, ils font par ailleurs de Linux un « commun d'entreprises », comme l'a expliqué Doc Searls :

Le rédacteur en chef du Linux Journal explique que « Linux est devenue une entreprise économique commune (une *joint venture*) composée d'un certain nombre de sociétés, tout comme Visa est une entreprise commune à un certain nombre de sociétés financières. Comme le montre le rapport de la Fondation Linux, ces sociétés participent au projet pour des raisons

commerciales diverses et variées ».

Dans un rapport de la Fondation Linux sur le noyau de Linux, il est dit clairement :

Plus de 70 % des développements du noyau sont visiblement réalisés par des développeurs qui sont rémunérés pour ce travail. Plus de 14 % vient de contributions de développeurs qui sont connus pour ne pas être rémunérés et être indépendants, et 13 % sont produits par des gens qui peuvent ou non être rémunérés, donc la contribution faite par des travailleurs rémunérés peut atteindre jusqu'à 85 %. Par conséquent, le noyau Linux est largement produit par des professionnels, et non par des bénévoles.

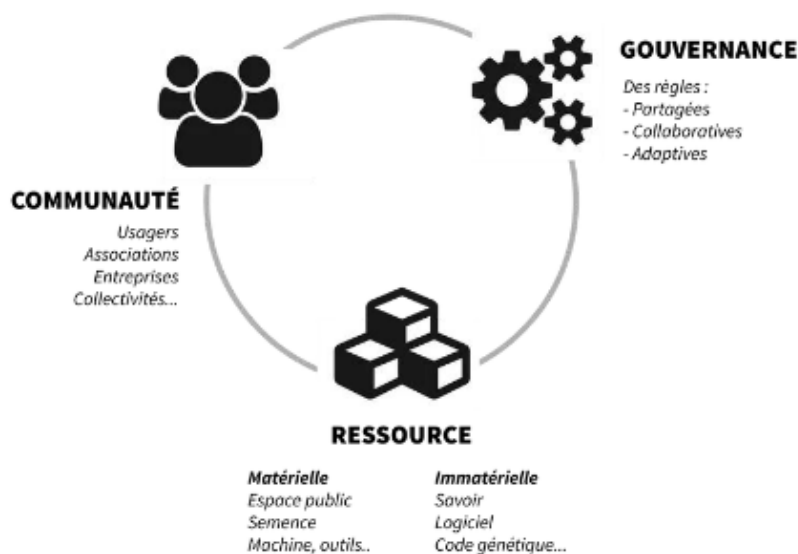
Mais ce n'est pas là toute l'histoire. Timothy Lee explique que la transformation de Linux en entreprise n'a pas changé son modèle d'organisation sous-jacente:

... l'important est la manière dont les projets open source sont organisés en interne. Dans un projet logiciel traditionnel, il y a un responsable projet qui décide des fonctionnalités dont bénéficiera le produit, et affecte du personnel pour travailler sur ces différentes fonctionnalités. En revanche, personne ne dirige le développement général du noyau Linux. Oui, Linus Torvalds et ses lieutenants décident quels correctifs iront finalement dans le noyau, mais les employés de Red Hat, IBM et Novell qui travaillent sur le noyau Linux ne reçoivent pas d'ordre de leur part. Ils travaillent sur ce qu'ils (et leurs clients respectifs) pensent être le plus important, et la seule autorité que possède Torvalds est celle de décider si le correctif qu'ils soumettent est suffisamment bon pour être intégré au noyau.

Clay Shirky, auteur de « [Here Comes Everybody: The Power of Organizing Without Organisations](#) [NdT : Voici venir tout le

monde: le pouvoir de s'organiser sans les organisations] souligne que les entreprises qui travaillent avec Linux, comme IBM, « ont abandonné le droit de gérer les projets pour lesquels ils payent, et que leurs concurrents ont accès immédiatement à tout ce qu'ils font. Ce n'est pas un produit IBM. »

C'est donc là où je veux en venir : même avec des sociétés d'actionnaires alliées à la production entre pairs, la création de valeur de la communauté reste toujours au cœur du processus, et la coalition entrepreneuriale, jusqu'à un certain point, suit déjà cette nouvelle logique, dans laquelle la communauté prime, et où le business est secondaire. Dans ce modèle, la logique d'entreprise doit s'accommoder de la logique sociale. En d'autres termes, c'est avant tout une « économie éthique ».



D'après une diapositive exposée par M. Dereva à l'occasion de la manifestation Le cloud de Numérique en Commun[s] – sept. 2018 (CC BY-NC-SA 4.0)

3. La logique démocratique des institutions à but lucratif

La production entre pairs repose aussi sur une infrastructure de coopération parfois coûteuse. Wikipédia n'existerait pas sans le financement de ses serveurs, pas non plus de logiciel

libre ou de matériel ouvert sans mécanisme de support similaire. C'est pour cela que les communautés *open source* ont créé une nouvelle institution sociale : les associations à but lucratif.

Encore une fois, c'est une innovation sociale importante car, contrairement aux institutions à but non-lucratif ou non-gouvernementales, elles ne fonctionnent pas du point de vue de la rareté. Les ONG classiques fonctionnent encore comme d'autres institutions industrielles à l'instar de l'entreprise ou de l'état-marché, car elles estiment que les ressources doivent être mobilisées et gérées.

À l'inverse, celles qui ont un but lucratif ont uniquement un rôle actif qui permet et favorise la coopération au sein de la communauté, qui fournit les infrastructures, sans pour en diriger les processus de production. Ces associations existent dans le seul but de bénéficier à la communauté dont elles sont l'expression, et c'est la bonne nouvelle, elles sont souvent gérées de manière démocratique. Et elles doivent l'être, car une institution non démocratique découragerait les contributions de sa communauté de participants.

Maintenant, le hic est de savoir comment appeler une institution responsable du bien commun de tous les participants, en l'occurrence, pas les habitants d'un territoire, mais les personnes impliqués dans un projet similaire ? Je rétorquerais que ce type d'institution à but lucratif possède une fonction très similaire aux fonctions normalement dévolues à l'État.

Bien que la forme étatique soit toujours aussi une institution de classe qui défend un arrangement particulier de privilèges sociaux, elle ne peut jamais être un simple instrument de règle de privilégié à elle seule, mais doit aussi gérer le commun. Si l'on considère cette dernière option, la plupart des gens la verrait comme acceptable, voire bonne. En revanche, si l'on considère que l'État échoue dans cette

gestion, alors il perd sa légitimité et il est vu de plus en plus comme une source d'oppression par une minorité.

En général, un État reflète l'équilibre des forces à l'œuvre dans une société donnée. L'État providence était une forme acceptable puisqu'il reposait sur un compromis et sur la force d'un puissant mouvement de travailleurs ouvriers, alors que « la peur de Dieu » était instillée dans les milieux privilégiés par la possibilité d'un modèle alternatif d'État qui aurait pu faire disparaître la loyauté de leurs citoyens.

Cette alternative s'est effondrée en 1989, et avec elle les mouvements sociaux occidentaux. Elle a d'autant plus été affaiblie par les choix sociaux, politiques et économiques de désindustrialiser le Nord depuis les années 1980. Depuis, l'État providence a peu à peu laissé sa place à l'État providence contemporain des multinationales (parfois appelé « l'État marché »), qui aide uniquement les privilégiés, détruit les mécanismes de solidarité sociaux, et appauvrit la majorité de la population, et a fortiori affaiblit fatalement la classe moyenne.

Malheureusement, un tel système ne peut avoir aucune légitimité à long terme, et rompt tout contrat social qui peut garantir la paix sociale. Il est compliqué d'établir une loyauté sur la perspective d'une souffrance toujours plus grande !

Cela signifie que nous assistons non seulement à la mort réelle de l'État providence social, mais aussi à la mort et à l'impossibilité logique de l'État-marché. Nous pourrions ajouter que même l'État providence est devenu problématique. La raison principale en est que sa base sociale, la classe ouvrière occidentale et ses mouvements sociaux, sont devenues des minorités démographiques en Occident, et que ses mécanismes, même lorsqu'ils fonctionnaient, ne contribueraient pas beaucoup à aider la majorité sociale actuelle, c'est-à-dire les travailleurs de la connaissance et des services,

souvent indépendants et précaires.

De plus, le fonctionnement paternaliste et bureaucratique de beaucoup d'États providence devient inacceptable face à la demande émergente d'autonomie sociale et personnelle qui est l'un des principaux désirs sociaux de la nouvelle classe des travailleurs de la connaissance. La plupart des autres fonctions sociales de l'État providence ont été affaiblies par les réformes néolibérales du « *New Labour* » qui tendent à introduire la logique du secteur privé dans le monde du secteur public.

Des routes et des ponts (10) – Les enjeux de la démocratisation de l'open source

L'*open source* est de plus en plus populaire et répandu parmi les développeurs. Grâce à des plate-formes comme GitHub, qui ont standardisé la manière de contribuer à un projet, l'*open source* est devenu plus accessible, au point de devenir une norme. Mais cette nouvelle configuration n'est pas sans poser certains problèmes.

Dans ce nouveau chapitre de son ouvrage [Des Routes et des ponts](#) (traduit [chapitre après chapitre](#) par l'équipe Framalang), [Nadia Eghbal](#) s'intéresse aux enjeux de la standardisation et de la démocratisation de l'open source – notamment à l'inflation parfois anarchique du nombre de projets et de contributeurs : pour elle, l'enjeu est d'éviter que l'écosystème numérique ne se transforme en un fragile

château de cartes.



Photo de [fdecomite](#) / [CC BY 2.0](#)

Pourquoi les problèmes de support des infrastructures numériques sont de plus en plus pressants

Traduction Framalang : goudron, Penguin, serici, goofy, Rozmador, xi, Lumibd, teromene, xi, Opsylac, et 3 anonymes.

L'*open source*, grâce à ses points forts [cités plus tôt dans ce rapport](#), est rapidement en train de devenir un standard pour les projets d'infrastructure numérique et dans le développement logiciel en général. Black Duck, une entreprise qui aide ses clients à gérer des programmes *open source*, dirige une enquête annuelle qui interroge les entreprises sur leur utilisation de l'*open source*. (Cette enquête est l'un des rares projets de banque de données qui existe sur le sujet.) Dans leur étude de 2015, 78% des 1300 entreprises interrogées déclarent que les logiciels qu'elles ont créés pour leurs clients sont construits grâce à l'*open source*, soit presque le double du chiffre de 2010.

L'*open source* a vu sa popularité s'accroître de manière impressionnante ces cinq dernières années, pas seulement grâce à ses avantages évidents pour les développeurs et les consommateurs, mais également grâce à de nouveaux outils qui rendent la collaboration plus facile. Pour comprendre pourquoi les infrastructures numériques rencontrent des problèmes de support grandissants, nous devons comprendre la manière dont le développement de logiciels *open source* prolifère.

GitHub, un espace standardisé pour collaborer sur du code

On n'insistera jamais trop sur le rôle clé de GitHub dans la diffusion de l'*open source* auprès d'une audience grand public. L'*open source* a beau exister depuis près de 30 ans, jusqu'en 2008, contribuer à des projets *open source* n'était pas si facile. Le développeur motivé devait d'abord découvrir qui était le mainteneur du projet, trouver une manière de le contacter, puis proposer ses changements en utilisant le format choisi par le mainteneur (par exemple une liste courriel ou un forum). GitHub a standardisé ces méthodes de communication : les mainteneurs sont listés de façon transparente sur la page du projet, et les discussions sur les changements proposés ont lieu sur la plate-forme GitHub.

GitHub a aussi créé un vocabulaire qui est désormais standard parmi les contributeurs à l'*open source*, tels que la « pull request » (où un développeur soumet à l'examen de ses pairs une modification à un projet), et changé le sens du terme « fork » (historiquement, créer une copie d'un projet et le modifier pour le transformer en un nouveau projet) [littéralement « fork » signifie « bifurcation », Ndt.]. Avant GitHub, forker un projet revenait à dire qu'il y avait un différend irréconciliable au sujet de la direction qu'un projet devrait prendre. Forker était considéré comme une action grave : si un groupe de développeurs décidait de forker un projet, cela signifiait qu'il se scindait en deux factions idéologiques. Forker était aussi utilisé pour développer un nouveau projet qui pouvait avoir une utilisation radicalement différente du projet initial.

Ce type de « fork de projet » existe toujours, mais GitHub a décidé d'utiliser le terme « fork » pour encourager à davantage d'activité sur sa plate-forme. Un fork GitHub, contrairement à un fork de projet, est une copie temporaire d'un projet sur laquelle on effectue des modifications, et qui

est généralement re-fusionnée au projet. Le fork en tant que pratique quotidienne sur GitHub a ajouté une connotation positive, légère au terme : c'est l'idée de prendre l'idée de quelqu'un et de l'améliorer.

GitHub a aussi aidé à standardiser l'utilisation d'un système de contrôle de version appelé Git. Les systèmes de contrôle de versions sont un outil qui permet de garder une trace de chaque contribution apportée sur un morceau de code précis. Par exemple, si le Développeur 1 et le Développeur 2 corrigent différentes parties du même code en même temps, enregistrer chaque changement dans un système de contrôle de version permet de faire en sorte que leurs changements n'entrent pas en conflit. Il existe plusieurs systèmes de contrôle de versions, par exemple Apache Subversion et Concurrent Versions System (CVS). Avant GitHub, Git était un système de contrôle de version assez méconnu. En 2010, Subversion était utilisé dans 60% des projets logiciels, contre 11% pour Git.

C'est Linus Torvalds, le créateur de Linux, qui a conçu Git en 2005. Son intention était de mettre à disposition un outil à la fois plus efficace et plus rapide, qui permette de gérer de multiples contributions apportées par de nombreux participants. Git était vraiment différent des systèmes de contrôle de version précédents, et donc pas forcément facile à adopter, mais son [workflow](#) décentralisé a résolu un vrai problème pour les développeurs.



GitHub a fourni une interface utilisateur intuitive pour les projets *open source* qui utilisent Git, ce qui rend l'apprentissage plus facile pour les développeurs. Plus les développeurs utilisent GitHub, plus cela les incite à continuer d'utiliser Git. Aujourd'hui, en 2016, Git est utilisé par 38% des projets de logiciels, tandis que la part de Subversion est tombée à 47%. Bien que Subversion soit

encore le système de contrôle de version le plus populaire, son usage décline. L'adoption généralisée de Git rend plus facile pour un développeur la démarche de se joindre à un projet sur GitHub, car la méthode pour faire des modifications et pour les communiquer est la même sur tous les projets. Apprendre à contribuer sur un seul des projets vous permet d'acquérir les compétences pour contribuer à des centaines d'autres. Ce n'était pas le cas avant GitHub, où des systèmes de contrôle de versions différents étaient utilisés pour chaque projet.

Enfin, GitHub a créé un espace sociabilité, qui permet de discuter et de tisser des liens au-delà de la stricte collaboration sur du code. La plate-forme est devenue *de facto* une sorte de communauté pour les développeurs, qui l'utilisent pour communiquer ensemble et exposer leur travail. Ils peuvent y démontrer leur influence et présenter un portfolio de leur travail comme jamais auparavant.

Les usages de GitHub sont un reflet de son ascension vertigineuse. En 2011 il n'y avait que 2 millions de dépôts (« repository »). Aujourd'hui, GitHub a 14 millions d'utilisateurs et plus de 35 millions de dépôts (ce qui inclut aussi les dépôts forkés, le compte des dépôts uniques est plutôt aux environs de 17 millions.) Brian Doll, de chez GitHub, a noté qu'il a fallu 4 ans pour atteindre le million de dépôts, mais que passer de neuf millions à dix millions n'a pris que 48 jours.

En comparaison, SourceForge, la plate-forme qui était la plus populaire pour héberger du code *open source* avant l'apparition de GitHub, avait 150 000 projets en 2008. Environ 18 000 projets étaient actifs.

Stack Overflow, un espace standard pour

s'entraider sur du code

L'une des autres plate-formes importantes de l'*open source* est Stack Overflow, un site de question/réponse populaire parmi les développeurs, créé en 2008 par Jeff Atwood (développeur déjà mentionné précédemment) et par le blogueur Joel Spolsky. En Avril 2014, Stack Overflow avait plus de 4 millions d'utilisateurs enregistrés et plus de 11 millions de questions résolues (à noter qu'il n'est pas nécessaire de s'enregistrer pour voir les questions ou leurs réponses).

Stack Overflow est devenu *de facto* une plate-forme d'entraide pour les développeurs, qui peuvent poser des questions de programmation, trouver des réponses à des problèmes de code spécifiques, ou juste échanger des conseils sur la meilleure façon de créer un aspect précis d'un logiciel. On pourrait définir la plate-forme comme un « support client » participatif pour les développeurs à travers le monde. Même si Stack Overflow n'est pas un endroit où l'on écrit directement du code, c'est un outil de collaboration essentiel pour les développeurs individuels, qui facilite grandement la résolution de problèmes et permet de coder plus efficacement. Cela signifie qu'un développeur individuel est capable de produire plus, en moins de temps, ce qui améliore le rendement global. Stack Overflow a également permis à certains utilisateurs d'apprendre de nouveaux concepts de développement (ou même de s'initier au code tout court), et a rendu le codage plus facile et plus accessible à tous.

Tendances macro dans un paysage en mutation constante

La popularité hors-normes de l'*open source* a amené à des changements significatifs dans la manière dont les développeurs d'aujourd'hui parlent, pensent et collaborent sur des logiciels.

Premièrement, les attentes et exigences en termes de licences ont changé, reflétant un monde qui considère désormais l'open source comme une norme, et pas l'exception : un triomphe sur l'univers propriétaire des années 1980. Les politiques de GitHub et de Stack Overflow reflètent toutes deux cette réalité.

Dès le départ, Stack Overflow a choisi d'utiliser une licence Creative Commons appelée CC-BY-SA pour tous les contenus postés sur son site. La licence était cependant limitante, car elle requérait des utilisateurs qu'ils mentionnent l'auteur de chaque morceau de code qu'ils utilisaient, et qu'ils placent leurs propres contributions sous la même licence.

Beaucoup d'utilisateurs choisissaient d'ignorer la licence ou n'étaient même pas au courant de ses restrictions, mais pour les développeurs travaillant avec des contraintes plus strictes (par exemple dans le cadre d'une entreprise), elle rendait Stack Overflow compliqué à utiliser. S'ils postaient une question demandant de l'aide sur leur code, et qu'une personne extérieure réglait le problème, alors légalement, ils devaient attribuer le code à cette personne.

En conséquence, les dirigeants de Stack Overflow ont annoncé leur volonté de déplacer toutes les nouvelles contributions de code sous la Licence MIT, qui est une licence *open source* comportant moins de restrictions. En Avril 2016, ils débattent encore activement et sollicitent des retours de leur communauté pour déterminer le meilleur moyen de mettre en œuvre un système plus permissif. Cette démarche est un encouragement à la fois pour la popularité de Stack Overflow et pour la prolifération de l'open source en général. Qu'un développeur travaillant dans une grosse entreprise de logiciel puisse légalement inclure le code d'une personne complètement extérieure dans un produit pour lequel il est rémunéré est en effet un accomplissement pour l'*open source*.

A l'inverse, GitHub fit initialement le choix de ne pas

attribuer de licence par défaut aux projets postés sur sa plateforme, peut-être par crainte que cela ne freine son adoption par les utilisateurs et sa croissance. Ainsi, les projets postés sur GitHub accordent le droit de consulter et de forker le projet, mais sont à part ça sous copyright, sauf si le développeur spécifie qu'il s'agit d'une licence *open source*.

En 2013, GitHub décida enfin de prendre davantage position sur la question des licences, avec notamment la création et la promotion d'un micro-site, choosealicense.com (« choisissez-une-licence »), pour aider les utilisateurs à choisir une licence pour leur projet. Ils encouragent aussi désormais leurs utilisateurs à choisir une licence parmi une liste d'options au moment de créer un nouveau « repository » (dépôt).

Ce qui est intéressant, cependant, c'est que la plupart des développeurs ne se préoccupaient pas de la question des licences : soit ignoraient que leurs projets « open source » n'étaient pas légalement protégés, soit ils s'en fichaient. Une étude informelle réalisée en 2013 par le Software Freedom Law Center (Centre du Droit de la Liberté des Logiciels) sur un échantillon de 1,6 million de dépôts GitHub révéla que seuls 15% d'entre eux avaient spécifié une licence. Aussi, les entretiens avec des développeurs réalisés pour ce rapport suggèrent que beaucoup se fichent de spécifier une licence, ou se disent que si quelqu'un demande, ils pourront toujours en ajouter une plus tard.

Ce manque d'intérêt pour les licences a amené James Governor, cofondateur de la firme d'analyse de développeurs Red Monk, à constater en 2012 que « les jeunes dévs aujourd'hui font du POSS – Post open source software. Envoie chier les licences et la gestion, contribue juste à GitHub » [en anglais « *commit to GitHub* » signifie littéralement « s'engager dans GitHub » mais fait également référence à la commande « *commit* » qui permet de valider les modifications apportées à un projet, NdT.]. En

d'autres termes, faire de l'information ouverte par défaut est devenu une telle évidence culturelle aujourd'hui que les développeurs ne s'imaginent plus faire les choses autrement – un contexte bien différent de celui des rebelles politisés du logiciel libre des années 1980. Ce retournement des valeurs, quoi qu'inspirant au niveau global, peut cependant amener à des complications légales pour les individus quand leurs projets gagnent en popularité ou sont utilisés à des fins commerciales.

Mais, en rendant le travail collaboratif sur le code aussi facile et standardisé, l'*open source* se retrouve aux prises avec une série d'externalités perverses.

L'*open source* a rendu le codage plus facile et plus accessible au monde. Cette accessibilité accrue, à son tour, a engendré une nouvelle catégorie de développeurs, moins expérimentés, mais qui savent comment utiliser les composants préfabriqués par d'autres pour construire ce dont ils ont besoin.

En 2012, Jeff Atwood, cofondateur de Stack Overflow, rédigea un article de blog intitulé ironiquement « Pitié n'apprenez pas à coder », où il se plaint de la mode des stages et des écoles de code. Tout en se félicitant du désir des personnes non-techniciennes de comprendre le code d'un point de vue conceptuel, Atwood met en garde contre l'idée qu'« introduire parmi la main-d'œuvre ces codeurs naïfs, novices, voire même-pas-vraiment-sûrs-d'aimer-ce-truc-de-programmeur, ait vraiment des effets positifs pour le monde ».

Dans ces circonstances, le modèle de développement de l'*open source* change de visage. Avant l'ascension de GitHub, il y avait moins de projets *open source*. Les développeurs étaient donc un groupe plus petit, mais en moyenne plus expérimenté : ceux qui utilisaient du code partagé par d'autres étaient susceptibles d'être également ceux qui contribuent en retour.

Aujourd'hui, l'intense développement de l'éducation au code

implique que de nombreux développeurs inexpérimentés inondent le marché. Cette dernière génération de développeurs novices emprunte du code libre pour écrire ce dont elle a besoin, mais elle est rarement capable, en retour, d'apporter des contributions substantielles aux projets. Beaucoup sont également habitués à se considérer comme des « utilisateurs » de projets *open source*, davantage que comme les membres d'une communauté. Les outils *open source* étant désormais plus standardisés et faciles à utiliser, il est bien plus simple aujourd'hui pour un néophyte de débarquer sur un forum GitHub et d'y faire un commentaire désobligeant ou une requête exigeante – ce qui épuise et exaspère les mainteneurs.

Cette évolution démographique a aussi conduit à un réseau de logiciels bien plus fragmenté, avec de nombreux développeurs qui publient de nouveaux projets et qui créent un réseau embrouillé d'interdépendances. Se qualifiant lui-même de « développeur-pie en rémission » [« pie » est un surnom pour les développeurs-opportunistes, d'après le nom de l'oiseau, la pie réputée voleuse, NdT], Drew Hamlett a écrit en janvier 2016 un post de blog devenu très populaire intitulé « Le triste état du développement web ». L'article traite de l'évolution du développement web, se référant spécifiquement à l'écosystème Node.js :

« Les gens qui sont restés dans la communauté Node ont sans aucun doute créé l'écosystème le plus techniquement compliqué [sic] qui ait jamais existé. Personne n'arrive à y créer une bibliothèque qui fasse quoi que ce soit. Chaque projet qui émerge est encore plus ambitieux que le précédent... mais personne ne construit rien qui fonctionne concrètement. Je ne comprends vraiment pas. La seule explication que j'ai trouvée, c'est que les gens sont juste continuellement en train d'écrire et de réécrire en boucle des applis Node.js. »

Aujourd'hui, il y a tellement de projets qui s'élaborent et se publient qu'il est tout simplement impossible pour chacun

d'eux de développer une communauté suffisamment importante et viable, avec des contributeurs réguliers qui discuteraient avec passion des modifications à apporter lors de débats approfondis sur des listes courriels. Au lieu de cela, beaucoup de projets sont maintenus par une ou deux personnes seulement, alors même que la demande des utilisateurs pour ces projets peut excéder le travail nécessaire à leur simple maintenance.

GitHub a rendu simples la création et la contribution à de nouveaux projets. Cela a été une bénédiction pour l'écosystème *open source*, car les projets se développent plus rapidement. Mais cela peut aussi parfois tourner à la malédiction pour les mainteneurs de projets, car davantage de personnes peuvent facilement signaler des problèmes ou réclamer de nouvelles fonctionnalités, sans pour autant contribuer elles-mêmes en retour. Ces interactions superficielles ne font qu'alourdir la charge de travail des mainteneurs, dont on attend qu'ils répondent à une quantité croissante de requêtes.

Il ne serait pas déraisonnable d'affirmer qu'un monde « post-open source » implique une réflexion non seulement autour des licences, ainsi que James Governor l'exprimait dans son commentaire originel, mais aussi autour du processus de développement lui-même.

Noah Kantrowitz, développeur Python de longue date et membre de la Python Software Foundation, a résumé ce changement dans un post de blog souvent cité :

« Dans les débuts du mouvement open source, il y avait assez peu de projets, et en général, la plupart des gens qui utilisaient un projet y contribuaient en retour d'une façon ou d'une autre. Ces deux choses ont changé à un point difficilement mesurable.

[...] Alors même que nous allons de plus en plus vers des outils de niche, il devient difficile de justifier l'investissement en temps requis pour devenir contributeur. »

Comblent son propre besoin » est toujours une excellente motivation, mais il est difficile de construire un écosystème là-dessus.

L'autre problème est le déséquilibre de plus en plus important entre producteurs et consommateurs. Avant, cela s'équilibrait à peu près. Tout le monde investissait du temps et des efforts dans les Communs et tout le monde en récoltait les bénéfices. Ces temps-ci, très peu de personnes font cet effort et la grande majorité ne fait que bénéficier du travail de ceux qui s'impliquent.

Ce déséquilibre s'est tellement enraciné qu'il est presque impensable pour une entreprise de rendre (en temps ou en argent) ne serait-ce qu'une petite fraction de la valeur qu'elle dérive des Communs. »

Cela ne veut pas dire qu'il n'existe plus de grands projets open source avec des communautés de contributeurs fortes (Node.js, dont on parlera plus tard dans ce rapport, est un exemple de projet qui est parvenu à ce statut.) Cela signifie qu'à côté de ces réussites, il y a une nouvelle catégorie de projets qui est défavorisée par les normes et les attentes actuelles de l'open source, et que le comportement qui dérive de ces nouvelles normes affecte même des projets plus gros et plus anciens.

Hynek Schlawack, [Fellow](#) de la Python Software Foundation et contributeur sur des projets d'infrastructure Python, exprime ses craintes au sujet d'un futur où il y aurait une demande plus forte, mais seulement une poignée de contributeurs solides :

« Ce qui me frustre le plus, c'est que nous n'avons jamais eu autant de développeurs Python et aussi peu de contributions de haute qualité. [...] Dès que des développeurs clefs comme Armin Ronacher ralentissent leur travail, la communauté toute entière le ressent aussitôt. Le jour où Paul Kehrer arrête de travailler sur PyCA, on est très mal. Si Hawkowl arrête son

travail de portage, Twisted ne sera jamais sur Python 3 et Git.

La communauté est en train de se faire saigner par des personnes qui créent plus de travail qu'elles n'en fournissent. [...] En ce moment, tout le monde bénéficie de ce qui a été construit mais la situation se détériore à cause du manque de financements et de contributions. Ça m'inquiète, parce Python est peut-être très populaire en ce moment mais une fois que les conséquences se feront sentir, les opportunistes partiront aussi vite qu'ils étaient arrivés. »

Pour la plupart des développeurs, il n'y a guère que 5 ans peut-être que l'*open source* est devenu populaire. La large communauté des concepteurs de logiciel débat rarement de la pérennité à long terme de l'*open source*, et n'a parfois même pas conscience du problème. Avec l'explosion du nombre de nouveaux développeurs qui utilisent du code partagé sans contribuer en retour, nous construisons des palaces sur une infrastructure en ruines.

Des routes et des ponts (9) – L'argent et l'open source

Nadia Eghbal a déjà évoqué plusieurs fois les liens entre l'argent et l'open source ([si vous avez manqué des épisodes](#)). Elle y revient dans ce chapitre, en insistant sur les questions fondamentales que pose l'argent aux communautés open source ainsi qu'à leurs membres.

Question de nature quasi-philosophique : l'open source peut-il perdre son âme à cause de l'argent ? Question de gouvernance : qui va décider de l'utilisation des fonds ? Et pour finir

question éthique et politique : jusqu'à où peut-on, doit-on accepter les requêtes des financeurs ?

La relation compliquée de l'*open source* avec l'argent

Traduction Framalang : goudron, Penguin, serici, goofy, Rozmador, xi, Lumibd, teromene, xi, Diane, et 3 anonymes



L'argent est un sujet tabou dans les projets *open source*, et ce depuis les premiers jours du mouvement du logiciel libre qui émergea en réponse directe aux pratiques commerciales des logiciels propriétaires. Dans le contexte du mouvement du logiciel libre, l'aversion pour l'argent est tout à fait compréhensible. L'argent est ce qui permettait de commercialiser les logiciels dans les années 1980 et il a fallu des décennies pour revenir sur cet état d'esprit et promouvoir les avantages liés à l'élaboration de logiciels qui soient libres d'utilisation, de distribution et de modification. Même si de nos jours, nous prenons les logiciels libres pour acquis, dans les années 1980, c'était une véritable contre-culture, un état d'esprit révolutionnaire.

Au sein même des communautés *open source*, il existe une croyance répandue selon laquelle l'argent est de nature à corrompre l'*open source*. Et en effet, le nombre de projets nés d'un « travail-passion » est assez incroyable. Aujourd'hui, le développement de logiciel est considéré comme un domaine lucratif, dont les écoles de programmation appâtent leurs futurs étudiants avec des promesses de premiers salaires en

dollars à six chiffres. Par contraste, il y a quelque chose de pur et d'admirable dans le fait de créer un logiciel simplement pour le plaisir.

D'un point de vue plus pratique, les projets *open source* se créent traditionnellement autour d'un besoin réel et identifiable. Quelqu'un estime qu'un projet pourrait être mieux fait, décide de le *forker*, effectue des améliorations, puis les diffuse pour qu'on en fasse usage. Le pragmatisme est au cœur de la culture *open source*, comme le prouve sa scission stratégique avec le mouvement du logiciel libre à la fin des années 1990. Certains contributeurs *open source* craignent, peut-être avec raison, que l'argent n'introduise un développement « artificiel » du système, avec des développeurs qui lancent de nouveaux projets simplement pour acquérir des financements, plutôt que pour répondre à un besoin réel.

David Heinemeier Hansson (aussi connu sous le pseudo de DHH), qui a créé le *framework* populaire *Ruby on Rails*, [mettait en garde en 2013](#) contre les mélanges entre *open source* et argent :

Si l'open source est une incroyable force pour la qualité et pour la communauté, c'est précisément parce qu'elle n'a pas été définie en termes de marché. Dans le cadre du marché, la plupart des projets open source n'auraient jamais eu leur chance.

Prenez Ruby on Rails. [...] C'est une réalisation colossale pour l'humanité ! Des milliers de gens, collaborant pendant une décennie entière pour produire une structure et un écosystème incroyablement aboutis, disponibles pour tous gratuitement. Prenez une seconde pour méditer sur l'ampleur de cette réussite. Pas seulement pour Rails, évidemment, mais pour de nombreux autres projets open source, encore plus grands, avec une filiation plus longue et encore plus de succès.

C'est en considérant ce fantastique succès, dû aux règles de vie d'une communauté, que nous devrions être extraordinairement prudents avant de laisser les lois du marché corrompre l'écosystème.

Structurellement, le meilleur atout de l'*open source* : son penchant pour la démocratie, est aussi sa faiblesse. Beaucoup de projets *open source* ne sont rien de plus qu'un dépôt numérique public où est stocké du code auquel un groupe de gens contribue régulièrement : l'équivalent d'une association officieuse sur un campus universitaire. Il n'y a pas de structure légale et il n'y a pas de propriétaire ou de chef clairement défini. Les « mainteneurs » ou les contributeurs principaux émergent souvent *de facto*, en fonction de qui a créé le projet, ou de qui y a investi beaucoup de temps ou d'efforts. Cependant, même dans ces cas-là, dans certains projets on répugne à introduire une hiérarchie favorisant clairement un contributeur par rapport à un autre.

En avril 2008, Jeff Atwood, un développeur .NET bien connu et [dont nous avons déjà parlé](#), a annoncé qu'il donnait 5 000 \$ au projet *open source* : *ScrewTurn Wiki*. *ScrewTurn Wiki* est un projet de wiki développé par Dario Solara, un autre développeur .NET, et maintenu par des volontaires. Atwood a dit à Dario que le don était « sans condition » : Solara pouvait utiliser l'argent de la manière qu'il jugerait la plus utile au projet.

Plusieurs mois plus tard, Atwood demanda à Solara comment il avait décidé de dépenser l'argent. Solara lui répondit que l'argent de la donation était « *encore intact. Ce n'est pas facile de l'utiliser... Que suggères-tu ?* » [Atwood a écrit](#) que cette réponse l'avait « terriblement déçu ».

La nature décentralisée du monde *open source* en a fait ce qu'il est : des logiciels produits de façon participative, que n'importe qui peut élaborer, partager, et améliorer. Mais quand vient le moment de discuter des besoins

organisationnels, ou de la viabilité, il peut être difficile de prendre des décisions faisant autorité.

Ces transitions vers une viabilité à long terme peuvent être interminables et douloureuses. Un des exemples les plus connus est le *noyau Linux*, un projet *open source* utilisé dans de nombreux systèmes d'exploitation à travers le monde, parmi lesquels Android et Chrome OS. Il a été créé en 1991 par Linus Torvalds, un étudiant en informatique .

Au fur et à mesure que le noyau Linux gagnait en popularité, Linus rechignait à discuter de l'organisation du développement du projet, préférant tout gérer tout seul. L'inquiétude et aussi la colère à l'égard de Torvalds grandirent chez les développeurs du projet, déclenchant de « vraies grosses disputes » selon Torvalds. Le conflit a atteint son apogée en 2002, on évoqua même un possible schisme.

Torvalds attribua ces conflits internes à un manque d'organisation, plutôt qu'à un quelconque problème technique :

Nous avons eu de vraies grosses disputes aux alentours de 2002, quand j'appliquais des correctifs à droite à gauche, et que les choses ne fonctionnaient vraiment pas. C'était très douloureux pour tout le monde, et également beaucoup pour moi. Personne n'aime vraiment les critiques, et il y avait beaucoup de critiques virulentes, et comme ce n'était pas un problème strictement technique, on ne pouvait pas juste montrer un correctif et dire : « Hé, regardez, ce patch améliore les performances de 15% » ou quoique ce soit de ce genre. Il n'y avait pas de solution technique. La solution a été d'utiliser de meilleurs outils, et d'avoir une organisation du travail qui nous permette de mieux distribuer les tâches.

La Fondation Linux a été créée en 2007 pour aider à protéger et à maintenir Linux et ses projets associés. Torvalds ne pilote pas la Fondation Linux lui-même, il a préféré recevoir

un salaire régulier en tant que « Compagnon Linux », et travailler sur ses projets en tant qu'ingénieur.

Malgré le fait que le logiciel *open source* soit admirablement ancré dans une culture du volontariat et de la collaboration relativement peu touchée par des motivations extérieures, la réalité est que notre économie et notre société, depuis les sociétés multimillionnaires jusqu'aux sites web gouvernementaux, dépendent de l'*open source*.

Dans l'ensemble, c'est probablement une évolution positive pour la société. Cela signifie que les logiciels ne sont plus limités à un développement privé et propriétaire, comme cela a été le cas pendant des dizaines d'années. Le fait que le gouvernement des États-Unis, ou un réseau social possédant des milliards d'utilisateurs, intègrent des logiciels construits par une communauté, annonce un futur optimiste pour la démocratie.

De plus, de nombreux projets fonctionnent très bien de manière communautaire lorsqu'ils sont d'une des deux tailles extrêmes possibles, c'est-à-dire soit des petits projets qui ne demandent pas de maintenance significative (comme dans l'exemple de Arash Payan et Appirater), soit de très gros projets qui reçoivent un soutien important de la part d'entreprises (comme Linux).

Cependant, beaucoup de projets sont coincés quelque part entre les deux : assez grands pour avoir besoin d'une maintenance significative, mais pas d'une taille suffisante pour que des entreprises déclarent leur offrir un soutien. Ces projets sont ceux dont l'histoire passe inaperçue, ceux dont on ne parle pas. Des deux côtés, on dit aux développeurs de ces projets « moyens » qu'ils sont le problème : du côté des « petits projets », on pense qu'ils devraient simplement mieux s'organiser et du côté des « gros projets », on pense que si leur projet était « assez bon », il aurait déjà reçu l'attention des soutiens institutionnels.

Il existe aussi des intérêts politiques autour de la question du soutien financier qui rendent encore plus difficile la prospection d'une source de financement fiable. On peut imaginer qu'une entreprise seule ne souhaite pas sponsoriser le développement d'un travail qui pourrait également bénéficier à son concurrent, qui lui n'aurait rien payé. Un mécène privé peut exiger des privilèges spécifiques qui menacent la neutralité d'un projet. Par exemple, dans les projets en lien avec la sécurité, le fait d'exiger d'être le seul à qui sont révélées les potentielles failles (c'est-à-dire payer pour être le seul à connaître les failles de sécurité plutôt que de les rendre publiques) est un type de requête controversé. Des gouvernements peuvent également avoir des raisons politiques pour financer le développement d'un projet en particulier, ou pour demander des faveurs spéciales comme une « *backdoor* » (une porte dérobée, c'est-à-dire un accès secret qui permet d'outrepasser les authentifications de sécurité), même si le projet est utilisé dans le monde entier.

Les récents démêlés légaux entre le FBI et Apple sont un bon révélateur des tensions qui existent entre technologie et gouvernement, au-delà même des projets *open source*.

Le FBI a, de manière répétée, et à l'aide d'assignations en justice, demandé l'aide d'Apple pour déverrouiller des téléphones afin d'aider à résoudre des enquêtes criminelles. Apple a toujours refusé ces requêtes. En février 2016, le FBI a demandé l'aide d'Apple pour déverrouiller le téléphone d'un des tireurs d'une attaque terroriste récente à San Bernardino, en Californie. Apple a également refusé de les aider, et [a publié une lettre sur son site](#), déclarant :

Tout en croyant que les intentions du FBI sont bonnes, nous pensons qu'il serait mauvais pour le gouvernement de nous forcer à ajouter une « backdoor » dans nos produits. Et finalement, nous avons peur que cette demande mette en danger les libertés que notre gouvernement est censé protéger.

En mars 2016, le FBI a trouvé une tierce partie pour l'aider à déverrouiller l'iPhone et a laissé tomber l'affaire.

Une des plus grandes forces de l'*open source* est que le code est considéré comme un bien public, et beaucoup de projets prennent la gestion de ces projets au sérieux. Il est important à titre personnel, pour beaucoup de développeurs de projets, que personne ne puisse prendre seul le contrôle d'une chose que le public utilise et dont il bénéficie. Toutefois, cet engagement à rester neutre a un prix, puisque beaucoup de ressources disponibles pour les développeurs de nos jours (comme les capitaux-risques ou les donations d'entreprises) attendent en contrepartie d'influer sur le projet ou des retours sur investissement.

Le logiciel *open source* est créé et utilisé de nos jours à une vitesse jamais vue auparavant. Beaucoup de projets *open source* sont en train d'expérimenter la difficile transition d'une création désintéressée à une infrastructure publique essentielle.

Ces dépendances toujours plus nombreuses signifient que nous avons pour responsabilité partagée de garantir à ces projets le soutien dont ils ont besoin.



Crédits pour les 2 images [Eelke](#) (CC BY 2.0)

Quand Linus Torvalds se lâche sur Linux et Git

Vous prenez tranquillement un café à Portland, dans l'Oregon, lorsque tout d'un coup vous vous apercevez que c'est [Linus Torvalds](#) en personne qui se trouve à la table d'à côté.

Comme vous êtes 1. développeur et 2. pas timide, vous l'accostez gentiment et lui demandez s'il accepterait de répondre à quelques questions.

Et l'on découvre alors un Torvalds toujours aussi franc dans ses réponses mais un peu désabusé voire aigri..



Linus Torvalds se lâche sur Linux et Git

[Linus Torvalds goes off on Linux and Git](#)

*Greg Jorgensen – 25 septembre 2012 – Typical Programmer
(Traduction : lgnap, Cyrille L., greygghart, Gatitac,
@adericbourg, lgodard, Penguin, Diwann, Florck + Anonymes)*

J'étais dans un café à Portland dans l'Oregon quand j'ai remarqué Linus Torvalds assis seul à une table. J'ai demandé au créateur du [noyau Linux](#) et logiciel de [gestion de versions Git](#) si je pouvais me joindre à lui. Pendant 15 minutes, nous avons parlé de programmation et de programmeurs.

Typical Programmer Linux a été publié pour la première fois il y a maintenant 20 ans. C'est maintenant l'un des systèmes d'exploitation les plus répandus. Quel effet cela vous fait-il ?

Linus Torvalds Franchement, je suis très étonné. J'ai lancé ce projet comme un hobby et je n'en attendais pas grand chose. Au fur et à mesure que l'intérêt pour Linux croissait, j'ai

observé qu'il était utilisé principalement par les développeurs purs et durs et les geeks pour se différencier de la foule. Il y a beaucoup de personnes dans l'industrie logicielle qui aiment montrer qu'ils utilisent le logiciel ou le langage de programmation le plus récent. C'est une question de statut, un peu comme les personnes qui parlent de groupes de musique indépendants ou de films étrangers. Que ce soit Linux, [Haskell](#), [MongoDB](#) ou n'importe quoi d'autre, tout groupe comporte au moins une personne qui passe son temps à parler de [fermetures](#) ou à expliquer comment il est en train de migrer son blog vers [Node.js](#) pour qu'il puisse supporter une augmentation de la charge.

TP Linux est maintenant grand public. Est-ce devenu trop facile ?

Linus Je ne pense pas que ce soit plus facile, mais il y a beaucoup plus de ressources maintenant. Dans la plupart des cas, Linux est plus difficile à comprendre et à utiliser que Windows ou MacOS. Les personnes qui l'utilisent pour les serveurs avaient déjà l'habitude d'Unix, le changement n'était donc pas très important pour eux. Ils connaissaient les conflits de mises à jour et de dépendances et le cauchemar des [bibliothèques](#) partagées. Sur les postes de travail, je pense que les gens s'en sont désintéressés après quelques années.

TP Une perte d'intérêt ? de motivation ?

Linus Personne ne s'emballe plus pour les terminaux [à fenêtre transparente](#) , pour les [palettes de couleur de Kate](#) ou pour être le premier à expliquer sur Slashdot comment faire marcher telle mystérieuse carte son. Ça, c'était ce qui faisait passer les précurseurs de Windows à Linux. Maintenant Linux ressemble à Windows. Je peux installer Ubuntu sur le portable de ma grand-mère et elle ne voit même pas la différence tant que l'icône Facebook est sur son écran.

TP Et au sujet du nombre de distributions ? On dirait qu'il y

a plus de distributions que d'utilisateurs finaux sous Linux.

Linus Il y a plus de distributions Linux sur un seul disque *Linux Format* que toutes les versions Windows réunies. Mais elles sont toutes à peu près similaires. Ce sont toutes à peu près la même chose réchauffée. Seuls des noms plus ou moins ingénieux ou drôles les distinguent les unes des autres. À partir du moment où un livre [Linux pour les nuls](#) a été publié, j'ai commencé à m'en désintéresser.

TP Vous avez publié le système de gestion de versions Git il y a moins de dix ans. Git est rapidement devenu à la mode et semble être majoritaire parmi les systèmes de gestion de versions, ou au moins celui que les gens recommandent le plus sur Reddit et HackerNews.

Linus Git a repris le relais parce que Linux commençait à ne plus trop séparer les gourous des ignorants. Je ne m'attendais pas vraiment à ce que quiconque l'utilise parce que c'est tellement dur à utiliser, mais finalement cela s'est révélé être son principal attrait. Aucune technologie ne saura jamais être trop compliquée ou trop obscure pour les barbus à t-shirt noirs.

TP Je trouvais déjà que Subversion était difficile d'accès. Je ne me suis pas encore cassé la tête avec Git.

Linus Vous allez passer beaucoup de temps à essayer de vous familiariser avec, tout en étant ridiculisé par les experts sur GitHub et ailleurs. J'ai appris qu'aucune [toolchain](#) n'est jamais trop compliquée car le besoin de prestige et la sécurité de l'emploi sont trop forts. Au bout du compte, vous allez vous aussi découvrir l'[Easter egg](#) dans Git : toutes les opérations significatives peuvent être exprimées à l'aide la commande *rebase*.

TP Que pensez-vous de GitHub du coup ?

Linus Ça a commencé comme un cocon, un sorte de cimetière pour

projets non maintenus ou inutiles, et c'est toujours la majorité de ce qui y est hébergé. Mais c'est devenu aujourd'hui une sorte de grand *World of Warcraft* pour développeurs, où ils sont notés en fonction de leurs contributions et des projets sur lesquels ils ont les privilèges *trunk*. J'ai entendu dire que des boîtes informatiques recrutaient désormais à partir de votre réputation GitHub, alors j'imagine que si vous ne committez rien vous n'obtiendrez plus de boulot dans les start-ups les plus cools. Le bon vieux temps où l'on faisait passer le [test du FizzBuzz](#) et où on demandait [comment faire pour déplacer le mont Fuji](#) pendant les entretiens d'embauche est révolu.

TP Vous semblez un peu aigri au sujet de Git.

Linus Vous verrez, les premiers livres *Git pour les nuls* et *Git Visual Quickstart* vont paraître dans quelques mois et, en ce qui me concerne, ce sera le début de la fin. Ces livres marquent la fin d'une certaine expertise sur Git et de la réputation sur GitHub comme indicateurs fiables de la qualité du geek. Une fois qu'une technologie est adoptée par la masse, les geeks les plus geeks la délaissent au profit de quelque chose de plus ésotérique encore. Regardez ce qui est arrivé à [Ruby on Rails](#). Les gens qui se forment à partir de tutoriels, de « todo lists » Rails, n'ont même jamais entendu parler de [DHH](#).

TP Et pour la suite ?

Linus Je ne suis pas sûr. Difficile de prédire la prochaine mode technologique. J'ai travaillé sur un éditeur de texte que j'utilise moi-même, si compliqué qu'il ferait passer [Vim](#) pour Notepad, peut-être que je le publierai un de ces quatre.

Linus a terminé son café et a dû partir. J'ai apprécié le temps qu'il m'a consacré et de m'avoir fait voir Linux et Git sous un jour nouveau.

Crédit photo : [Murray Wayper](#) (Creative Commons By-Nc-Nd)

NdT : Oups, j'oubliais, Greg Jorgensen a publié son billet sous le tag « [satire](#) ». Je dis ça, je dis rien... Ça vous apprendra à ne pas lire jusqu'au bout ☐

Les 20 personnalités préférées de l'Internet Libre

L'excellent quotidien d'information britannique *The Guardian* (que nous traduisons souvent ici) a récemment proposé une liste restreinte et subjective de vingt « fighters for Internet freedom ».

Stallman, Torvalds, Wales, Lessig, Assange, Sunde, Berners-Lee, Anonymous... Il y en a que nous connaissons bien ici. D'autres moins, mais c'est l'occasion de les découvrir^[1].

Certains lecteurs regrettent ainsi par exemple l'absence de [Cory Doctorow](#), [Eben Moglen](#) ou Mitchell Baker.

Qu'en pensez-vous ? Qui rajouteriez-vous ? Et pourquoi aucun français ni francophone ?

D'ailleurs s'il fallait faire une liste *nationale* vous choisiriez qui ?

(Petite liste alphabétique qui s'édite en fonction des commentaires : Philippe Aigrain, Jean-Pierre Archambault, Benjamin Bayart, Stéphane Bortzmeyer, Guillaume Champeau, Frédéric Couchet, Laurent Chemla, Loïc Dachary, Florence Dévouart, François Elie, Fabrice Epelboin, Roberto Di Cosmo, Christophe Espern, Sam Hocevar, Antoine Moreau, Tristan Nitot, Valentin Lacambre, Bernard Lang, Olivier Laurelli, Jean-Marc Manach, Jean Peyratout, Bernard Stiegler, Thierry Stoehr,

Jérémie Zimmermann...)

Un billet à rapprocher de celui qui rendait hommage à quelques figures marquantes du monde du Libre : [Sur la place des grands hommes du logiciel libre.](#)



Top 20 du Guardian : ceux qui se battent pour un Internet libre

[The Guardian's Open 20: fighters for internet freedom](#)

Des hommes politiques et professeurs aux informaticiens en passant par le premier programmeur, les champions de l'Internet libre. Qui avons-nous oublié ? Faites votre propre liste [ici](#).

Anonymous

Légions, partout

([Wikipédia](#))

Le cri de ralliement des Anonymous – « Nous sommes Anonymes. Nous sommes Légion. Nous ne pardonnons pas. Nous n'oublions pas. Redoutez-nous » (*We are Anonymous. We are Legion. We do not forgive. We do not forget. Expect us*) – n'est pas du goût de tout le monde, mais est certainement très connu. À l'inverse de son groupe frère Lulzsec, le collectif Anonymous est vraiment fragmenté et sans leader et a donc pu continuer malgré les arrestations du FBI qui ont paralysé Lulzsec. Les pirates ont récemment mis brièvement hors ligne le site web du ministère de l'intérieur anglais et certains des sites web les plus visités au monde en protestation contre les propositions de lois de surveillance d'internet.

Jacob Appelbaum

Militant, chercheur et développeur, Projet Tor

([Wikipédia](#))

Appelbaum, un chercheur en informatique à l'Université de Washington, est l'un des principaux membres du projet Tor, qui permet de garantir l'anonymat de milliers d'internautes à travers le monde. Également connu comme le principal défenseur du groupe, Appelbaum a attiré l'attention du public après avoir été arrêté et fouillé à de nombreuses reprises par des douaniers américains, qui lui ont confisqué son matériel électronique, après qu'il a défendu Julian Assange pendant une conférence.

Julian Assange

Éditeur en chef, WikiLeaks

([Wikipédia](#) – [Framablog](#))

La force motrice derrière Wikileaks, Assange a coordonné la publication de documents secrets concernant les guerres en

Afghanistan et en Irak, de fichiers de prisonniers de la Baie de Guantanamo, et de 250 000 câbles diplomatiques. Assange est quelqu'un qui divise l'opinion, à cause des nombreuses polémiques qui l'entourent mais malgré cela (ou peut-être grâce à cela), il est sûrement le chef de file du mouvement pour un Internet libre et un interlocuteur important.

John Perry Barlow

Co-fondateur, Electronic Frontier Foundation
([Wikipédia](#))

Fondée en 1990, l'EFF se décrit elle-même comme « la première ligne de défense » lorsque les libertés numériques sont menacées. Grâce à un mélange d'action directe, d'actions en justice et de lobbying dans le milieu politique, le groupe défend la liberté d'expression et se bat contre la surveillance et les problèmes de propriété intellectuelle. L'ancien parolier des Grateful Dead est un des membres fondateurs de l'EFF et a depuis été l'un de ses plus importants porte-parole.

Sir Tim Berners-Lee

Inventeur du world wide web
([Wikipédia](#) – [Framablog](#))

Étant l'inventeur d'une des parties les plus visible de l'Internet (le world wide web), la place de Berners-Lee dans l'histoire de l'Internet était déjà réservée. Il ne s'est cependant pas reposé sur ses lauriers: il joue un rôle important dans la promotion de l'open data jusque dans les hautes sphères des gouvernements à travers le monde, il fait également campagne contre un internet à « deux vitesses ». Il s'est aussi récemment excusé pour les deux slashes au début d'une adresse web (http://), reconnaissant qu'ils sont «?complètement inutiles?».

Heather Brooke

Journaliste et militante

([Wikipédia](#))

Militante pour la liberté d'information et contre la surveillance des états, Heather Brooke a eu un rôle capital dans les procès qui ont permis de révéler les abus dans le système de dépense du premier ministre britannique. Durant ses recherches sur la culture hacker et l'activisme en ligne, Brooke s'est procuré les câbles diplomatiques de Wikileaks et a été l'une des journalistes à travailler sur le projet. Elle siège au conseil de l'Open Rights Group et elle est professeur en résidence à la City University de Londres.

Bram Cohen

Scientifique en chef, BitTorrent

([Wikipédia](#))

Bram Cohen n'est pas un homme populaire à Hollywood. Cohen a non seulement inventé la technologie du peer-to-peer qui est derrière le réseau BitTorrent, mais a également mis au point le logiciel qui permet aux utilisateurs de partager leurs fichiers. La technologie revendique plus de 100 millions d'utilisateurs actifs chaque mois, téléchargeant à peu près 400 000 fichiers chaque jour – quelques-uns sont légaux, mais pour beaucoup, il s'agit de films, musiques et programmes télé protégés par le droit d'auteur.

Rickard Falkvinge

Fondateur, le parti Pirate

([Wikipédia](#) – [Framablog](#))

Falkvinge a fondé le parti Pirate Suédois en 2006 afin de se concentrer sur la réforme des lois sur le droit d'auteur, les brevets et le partage de fichiers. Le parti a maintenant une présence souvent marginale dans 22 pays, voire significative en Suède, où il dispose de deux députés européens, ainsi qu'en

Allemagne, où il est la troisième force politique du pays.

Birgitta Jonsdottir

Membre du Parlement, The Movement, Islande

([Wikipédia](#))

Une poétesse et militante devenue femme politique, Jonsdottir est élue au parlement islandais depuis 2009. Principalement connue pour son rôle dans la diffusion de la vidéo WikiLeaks Collateral Murder au grand public, Jonsdottir a également eu un rôle prépondérant dans l'effort qu'a fait l'Islande pour être un paradis de la liberté d'expression, et c'est l'une des plaignantes dans les poursuites contre le gouvernement américain concernant les pouvoirs de surveillance que permet la loi NDAA.

Dr Susan Landau

Chercheur en cyber-sécurité, Membre de Guggenheim

([Wikipédia](#))

Susan Landau est professeur invitée du département de sciences informatiques de l'Université d'Harvard avec plus de 30 ans de publications sur la cyber-sécurité, la surveillance et la cryptographie derrière elle. Elle milite pour le respect de la vie privée des utilisateurs et dénonce les systèmes de surveillance installés sur les canaux de communication. Elle combat également pour la cause des femmes dans les sciences, et dirige la mailing list ResearchHers. Elle a gagné le prix Women of Vision social impact en 2008.

Lawrence Lessig

Fondateur, Creative Commons

([Wikipédia](#) – [Framablog](#))

Tandis que beaucoup d'hacktivistes se contentent simplement d'ignorer les lois et en risquent donc les conséquences, Lawrence Lessig a suivi une approche plus douce,

en introduisant un type de licence qui autorise le partage de contenu sans crainte de poursuites. Les personnes qui créent des œuvres qui devraient normalement être soumises au droit d'auteur peuvent utiliser les licences Creative Commons (CC) qui autorisent une ré-utilisation libre, en y adjoignant une clause d'usage non-commercial ou de non-modification s'ils le souhaitent. Plus de 100 millions d'images sont déjà disponibles sur internet sous licences Creative Commons.

Ada Lovelace

Programmeuse

([Wikipédia](#))

Ada Lovelace, morte en 1852, sert d'inspiration à un internet libre. Lovelace travaillait avec Charles Babbage sur sa machine analytique, a écrit certains des premiers programmes informatiques, et elle est donc considérée comme la première programmeuse d'ordinateur. À l'opposé de Babbage, elle avait compris le rôle que pourraient jouer les ordinateurs pour faire de la musique, de l'art et plus encore. Ces dernières années, le jour Ada Lovelace est devenu une institution sur Internet, promouvant le rôle des femmes dans les sciences et technologies, et modifiant la vision que l'on a d'elles dans les médias.

Alex MacGillivray

Conseiller Général, Twitter

([Wikipédia](#))

Alex MacGillivray, l'avocat général de Twitter, est celui à qui l'on attribue le mantra de l'entreprise « la branche liberté d'expression du parti de la liberté d'expression » (*the free-speech wing of the free-speech party*), et a joué un rôle important dans les efforts qu'a faits le site pour rester le plus transparent possible sans pour autant se rendre hors-la-loi. Ses récents efforts en vue de limiter la censure non plus au niveau planétaire mais pays par pays, ont d'abord

provoqué une fronde mais ont finalement été vus par beaucoup comme une façon habile de n'adopter que le minimum des restrictions exigées par la législation.

Clay Shirky

Ecrivain, professeur assistant à l'Université de New York
([Wikipédia](#) – [Framablog](#))

Clay Shirky a été l'un des premiers ardents défenseurs du crowdsourcing, de la collaboration et de l'agrégation de contenus et du journalisme en ligne, et donc des institutions ouvertes nécessaires à leur développement. Shirky encourage des institutions à se remettre en question dans un monde toujours plus connecté, et il est crédité comme l'une des personnes qui a inspiré la politique de journalisme ouvert du Guardian.

Richard Stallman

Fondateur, Free Software Foundation
([Wikipédia](#) – [Framablog](#))

L'un des défenseurs les plus ardents du logiciel libre (et non open source, terme qu'il déteste), Stallman fait le tour du monde afin de montrer les avantages qu'apportent des logiciels libres d'utilisation et libres de modification. Cependant, Stallman est davantage qu'un porte-parole, il est également l'un des principaux programmeurs de GNU (un système d'exploitation qu'il a mis au point).

Peter Sunde

Co-fondateur, Pirate Bay
([Wikipédia](#) – [Framablog](#))

Peter Sunde était une des personnes responsables de Pirate Bay, un moteur de recherche permettant l'accès à plus de 4 millions de fichiers sur le réseau BitTorrent, et un portail clé pour toute personne échangeant des fichiers. Le site a

longtemps évité les tentatives de la justice pour l'interdire, mais Sunde est déjà passé à autre chose, en fondant Flattr, un site de micro-paiement qui a pour but de rétribuer financièrement et volontairement les sites indépendants et les blogs. Pendant ce temps, Pirate Bay prétend mettre au point des serveurs embarqués sur des drones, afin de s'assurer que le gouvernement ne sera jamais capable de les déconnecter.

Aaron Swartz

Programmeur, militant

([Wikipédia](#))

Si un vendeur sur un marché tunisien a lancé le printemps arabe, il est peut être juste de créditer Aaron Swartz d'avoir lancé le «?printemps universitaire?», si les accusations le concernant s'avèrent fondées. Swartz est accusé d'avoir téléchargé plus de 4 millions d'articles universitaires du site JSTOR afin d'améliorer l'accès à la littérature savante. Après son inculpation, et alors qu'il se dit innocent, d'autres personnes ont commencé à partager des milliers de papiers sur le net sans aucune permission. Maintenant, l'accès aux publications universitaires est en train d'être libéré par des moyens légaux et de plus en plus rapidement, avec comme principal appui financier le Wellcome Trust, et certains ministres du Royaume Uni qui soutiennent maintenant le libre accès.

Professeur Sebastian Thrun

Fondateur, Udacity

([Wikipédia](#))

Le professeur Thrun n'était pas un homme qui avait à se soucier de son prochain salaire?: en tant que professeur permanent en intelligence artificielle à la prestigieuse Stanford University, il avait un poste à vie. Non satisfait d'enseigner à un nombre restreint d'étudiants, il a donné accès à ses cours en ligne gratuitement, à travers un site

nommé Udacity. La première année, plus de 140 000 étudiants se sont inscrits à ses cours.

Linus Torvalds

Architecte en chef, Linux

([Wikipédia](#) – [Framablog](#))

Pour ceux qui ne seraient pas désireux d'avoir à choisir entre Microsoft Windows et Apple Mac OSX, une alternative libre existe, qui s'accompagne d'une valeur ajoutée (pour les programmeurs) : elle est ouverte aux modifications et personnalisations, cette alternative c'est Linux. Linus Torvalds est un développeur Finno-Américain qui a démarré ce projet, et a depuis lors montré la voie à de nombreuses distributions en tant que défenseur de l'open source et des logiciels libres.

Jimmy Wales

Fondateur et membre du Board, Fondation Wikimedia

([Wikipédia](#) – [Framablog](#))

Jimmy Wales est l'homme derrière Wikipédia, la plus grosse encyclopédie au monde (avec 21 millions d'articles), rédigé exclusivement par des bénévoles, grâce à son système d'édition ouvert. De plus, Wales a récemment gagné en légitimité en matière d'Internet ouvert en encourageant le comité du site à sortir de sa politique habituelle de neutralité afin d'organiser un black-out d'une journée de la version anglophone du site, en réponse à la proposition de loi SOPA contre la piraterie.

Qui avons-nous oublié ? [Faites-le nous savoir](#) en proposant vos propres choix dans les commentaires.

Notes

[1] Crédit photo : [Aaron Brown](#) (Creative Commons By-Nc)

Levons le tabou du stress, surmenage et burnout au sein des communautés

Voici un sujet dont on parle trop peu parce que ceux qui en sont victimes pratiquent souvent le déni et n'aiment pas apparaître vulnérables aux yeux de leurs pairs.



Quitte à ce que cela craque complètement un jour et qu'il n'y ait plus d'autre issue que de disparaître momentanément (ou pire définitivement) de la circulation.

Il s'agit du phénomène de **burnout** que Wikipédia traduit par [syndrome d'épuisement professionnel](#)^[1].

Il touche également les associations et l'activité bénévole car c'est avant tout d'un trop plein de travail et de responsabilités dont il est question. Et tous ceux qui me connaissent d'un peu près savent que j'en suis parfois passé par là au sein de Framasoft.

Parce que, oui, le Libre n'est pas épargné. Il serait même, à parcourir la traduction ci-dessous, parmi les plus durement touchés, à cause de ses spécificités mais aussi parce qu'Internet, aussi pratique soit-il, peut parfois manquer de patience, d'attention et de bienveillance.

Un article qui cherche à mieux comprendre pour mieux prévenir.

Et n'hésitez pas à laisser votre témoignage dans les

commentaires, histoire de libérer aussi la parole et aider ceux qui sont susceptibles de tomber dans ce piège que l'on se fabrique presque toujours tout seul.

Remarque : Nous avons choisi de traduire systématiquement ci dessous « burnout » par « surmenage », même si le mot anglais commence à se diffuser chez les francophones et évoque mieux le processus d'aller au bout de ses forces en épuisant ponctuellement ou durablement toute son énergie.

Linus Torvalds et d'autres à propos du surmenage dans les communautés

Des développeurs open source parlent du stress des codeurs dans le monde de Linux.

[Linus Torvalds and Others on Community Burnout](#)

Bruce Byfield – 30 août 2011 – Datamation

(Traduction Framalang : Yonnel, Deadalnx, Mammig2, Martin, Raphaelh, Penguin)

Traînez autant de temps que vous le voudrez dans la communauté du libre et de l'open source, et vous ne manquerez pas de rencontrer des exemples de surmenage. Un collègue prend trop de choses à sa charge, et d'un coup il ne travaille plus, avec de moins bons résultats.

Il a du mal à se concentrer sur son travail. Il néglige sa vie privée. Face aux contestations, il est sur la défensive et devient étrangement agressif. Et pour finir il s'en va, pour souvent ne jamais revenir.

Le surmenage n'est pas l'apanage de la seule communauté linuxienne, bien entendu. Et pourtant ce problème semble toucher la communauté telle une épidémie, et ses membres semblent réticents à en parler en public.

Selon le « community manager » d'Ubuntu [Jono Bacon](#) et la

journaliste et contributrice d'Ubuntu [Amber Graner](#), qui donnent des conférences adaptées de « [The Burnout Cycle](#) » d'Herbert Freudenberger et Gail North, les gens les contactent par la suite en privé pour parler de leur propre expérience de surmenage.

Et de même, la contributrice au noyau Linux et co-fondatrice d'Ada Initiative, [Valerie Aurora](#), se rappelle d'une discussion avec une douzaine de femmes activistes dans les nouvelles technologies, où elles ont découvert que toutes étaient soit en surmenage, soit en train de s'en remettre, soit l'avaient été.

Personne ne semble être à l'abri, pas même Linus Torvalds. Bien qu'il commence par dire « je n'ai jamais vraiment été victime de surmenage », il en vient ensuite à parler d'une situation qui a tout des premiers stades du surmenage :

« Nous avons de très grosses disputes vers 2002 ([cf « Linus ne sait pas ce qu'il demande »](#)), je posais des patches à droite et à gauche, et ça ne marchait pas vraiment. C'était pénible pour tout le monde, à commencer par moi.

Personne n'aime la critique, et il y avait beaucoup de descentes en flèche. Et comme ce n'était pas un problème strictement technique, on ne pouvait pas identifier un patch et dire « hé, regardez, ce patch améliore de 15 % la vitesse » ou autre chose du genre : il n'y avait donc pas de solution purement technique. À la fin, la solution était de meilleurs outils et une charge de travail mieux répartie ».

Quelle est la cause du surmenage, en particulier dans la communauté du libre ? Que faire pour l'éviter, à la fois individuellement et au niveau de la communauté ? Ces questions, de plus en plus de leaders du logiciel libre ont du mal à y répondre. Le surmenage commence seulement à être reconnu comme un problème sur lequel se pencher.

Les origines du surmenage

L'organisation dans le libre rend les membres de la communauté particulièrement exposés au stress. Selon Bacon, les contributeurs étant répartis autour de la planète et certains étant volontaires, chacun doit alors gérer lui-même sa charge de travail.

Mais lorsque quelqu'un peut travailler n'importe où sur un projet à n'importe quelle heure, fixer des limites devient alors ardu. Comme pour un jeu de simulation en temps réel, il n'y a pas de moment idéal pour arrêter. En réalité, à cause des réponses instantanées qui sont la norme sur Internet, certains peuvent s'agacer de ne pas voir les autres immédiatement disponibles.

Le stress peut augmenter car la première génération des membres de communautés sont maintenant largement d'âge mûr, et certains commencent à avoir des difficultés à travailler aux heures auxquelles ils étaient habitués, que ce soit dû à la fatigue ou aux obligations familiales.

D'autre part, selon Graner, certains membres de communautés ajoutent une couche à leur stress en prenant davantage de travail pour se prouver quelque chose à eux-mêmes. Elle observe, par exemple, que chez Ubuntu, ceux qui ne font pas de développement peuvent se sentir moins impliqués dans le projet que les développeurs, ou prennent davantage de responsabilités dans l'espoir que leurs sacrifices soient payants et qu'ils puissent accompagner les développeurs à l'Ubuntu Developer Summit.

« Ils pensent que s'ils n'en font pas toujours davantage et ne deviennent pas cette *super personne* de la communauté, alors les gens penseront qu'ils n'en font pas assez », affirme Graner.

Pourtant, comme le fait remarquer Torvalds, le surmenage n'est pas uniquement lié au stress. « Personnellement les grosses

engueulades ponctuelles ont tendance à me plaire et à me gonfler à bloc », affirme-t-il. « Cela peut être stressant, mais ça peut être aussi revigorant, et je pense même que, sans ces éruptions occasionnelles, votre projet a tendance à s'endormir, ou alors c'est que vous n'y croyez plus assez. »

Cependant il ajoute: « mais le stress continué peut aussi être vraiment usant. Pour moi, ça a toujours été plus ou moins un problème de gestion du flux de travail. Le stress vient du manque d'énergie suffisante (ou du nombre d'heures dans une journée) pour faire ce que je dois faire. C'est donc pour ça, au niveau du noyau Linux, que je pense que les gros moments de stress ont toujours tourné autour de problèmes d'organisation du flux de travail. »

Bacon perçoit le surmenage de manière similaire, en le définissant ainsi : « vous cumulez le stress des jours précédents, et cela augmente jusqu'à ce que vous ne puissiez plus le surmonter. »

À l'opposé, l'ancien leader de la communauté Fedora [Paul Frields](#) voit l'origine du surmenage dans les interactions dans un groupe :

« Les gens n'ont pas tous les mêmes attentes. Celle par exemple d'aspirer à ce que les autres membres de l'équipe aient le même degré d'implication que vous, sans tenir compte des capacités, du temps disponible ou des situations personnelles des autres. Ou peut également souhaiter, consciemment ou non, que tout le monde adore et adopte votre nouveau concept original et radical là, maintenant, tout de suite. Si ces attentes ne se concrétisent pas, et que vous continuez à ruminer longtemps là-dessus, il y a de très fortes chances pour que vous soyez bientôt en surmenage. »

Une autre source possible de surmenage pour les femmes en particulier est leur [sous-représentation](#) dans la communauté. Selon le projet, les femmes représentent en général de un à

cinq pour cent de la communauté. Non seulement doivent-elles subir les remarques sexistes, les présentations pornographiques, voire une hostilité pure et simple, mais elles ont le sentiment de se retrouver tout de suite en situation de faire leurs preuves – tout autant par rapport aux femmes déjà présentes que par rapport à la majorité masculine.

« C'est un peu comme à l'armée », dit Graner, qui a participé à la première guerre du Golfe. « Vous devez en faire dix pour cent de plus que n'importe qui d'autre pour être perçue comme aussi bonne qu'eux. »

Pour celles qui essaient vraiment de changer cette culture, le stress est encore plus intense. « Il y a tout simplement trop peu de femmes dans l'open source pour assumer tout le travail à faire de ce côté-là », constate Aurora. « Un seul pour cent au sein d'une communauté et c'est déjà mathématiquement le surmenage assuré. Vous êtes en situation précaire, vous recevez plein de messages disant que vous n'êtes pas à votre place, Vos heures et vos heures de temps libre pour ce militantisme n'y changeront rien. Vous vous en voulez de ne pas faire du code, et d'avoir des doutes tout court. »

De plus il n'y souvent qu'une seule figure de proue du féminisme à la fois. « Vous devenez une cible de choix pour les critiques et les menaces », explique Aurora. « Vous en payez sacrément le prix. À chaque fois que quelqu'un devient la représentante de la cause des femmes dans l'open source, sa carrière en pâtit. »

Pour compliquer encore les choses, tous genres confondus, le surmenage est un mal difficile à diagnostiquer ou à admettre. « On distingue fort bien les symptômes chez les autres sans nous apercevoir que bien souvent c'est notre propre reflet que l'on regarde », dit Graner. « Et comme le mot *surmenage* a souvent une connotation péjorative cela ne pousse pas à la confiance. »

Un tel déni est surtout fréquent chez les leaders en surmenage, soit parce qu'ils se considèrent essentiels, soit parce qu'ils ont plus l'habitude de venir en aide plutôt que d'avoir besoin qu'on les aide eux-mêmes. Mais dans tous les cas ce déni ne fait qu'aggraver la situation en rendant les gens plus réticents à faire ce qu'il faut pour s'en sortir.

Traiter le surmenage

Torvalds suggère que, pour lui, la clef pour se remettre d'un surmenage est :

« d'apprendre à laisser aller les choses. Si l'on ne le fait pas pour l'ensemble du projet, il faut au moins ne plus essayer de le contrôler entièrement. Avec le noyau Linux, je pourrais être le mainteneur principal, mais je fais simplement confiance aux autres pour faire ce qu'il faut. Il y a toujours des parties du projet que je suis de très près, mais même pour celles-ci, je suis vraiment content quand des personnes à qui je fais confiance font le travail à ma place.

Sinon, abandonnez purement et simplement le projet. C'est ce que j'ai fait pour [git](#) (le logiciel de gestion de versions décentralisée) : ça me plaisait vraiment, mais il me semblait aussi ne pas pouvoir prétendre être le mainteneur à plein temps dont le projet avait besoin. Et j'ai vraiment été ravi de trouver un très bon mainteneur ([Junio Hamano](#)). Cela restait quelque part mon bébé, mais en même temps, le mieux pour git était que quelqu'un d'autre le gère ».

Bien sûr, comme l'ajoute Torvalds, « les gens semblent parfois avoir du mal à lâcher prise, moi y compris. »

Pour répondre à la résistance naturelle au lâcher prise, Frields suggère: « il vous faut la volonté de faire un examen de conscience, de vérifier votre équilibre et votre capacité à vous engager pleinement pour votre accomplissement. Et plus de la volonté, il vous faut réellement et en toute conscience

prendre effectivement le temps de le faire. »

Bacon est encore plus précis. En partant de sa propre expérience de surmenage, qui s'est produite environ un an après avoir rejoint Canonical, il a beaucoup réfléchi à la façon d'organiser une vie équilibrée qui pourrait le rendre, lui et d'autres, plus résistants au surmenage.

Ne pas être célibataire est une des meilleures garanties contre le surmenage, selon Bacon, mais il fait remarquer que même les célibataires peuvent passer une soirée loin de la communauté et prendre du bon temps avec des amis. Il suggère également d'avoir d'autres activités (pour Bacon, c'est de faire de la musique avec son groupe [Severed Fifth](#)), de faire du sport régulièrement, de suivre un régime plus sain et moins calorique.

Dans ce régime, il préconise de réduire la dose de caféine, à laquelle de nombreux membres de la communauté sont littéralement accros ; Bacon lui-même décrit le sevrage de ses six canettes de Coca par nuit, avec tous les vomissements et les tremblements que cela a entraîné, comme « une des expériences les plus affreuses de ma vie », et fait figurer la restriction de caféine en haute place parmi les changements mis en place pour réduire les probabilités de surmenage.

Le surmenage peut aussi être régulé à l'intérieur de la communauté, en créant une culture commune qui rencontre l'adhésion de tous. Une culture où, selon les termes de Graner, « si vous n'êtes pas à cent pour cent, alors vous ne nous rendez pas service », et où l'on vous encourage à prendre régulièrement des pauses réparatrices et salvatrices.

Graner suggère également que le travail au sein du logiciel libre « doit être un effort collectif pour qu'aucune personne ne soit responsable de l'ensemble ». En se basant sur son expérience personnelle dans l'armée, elle conseille à chacun d'apprendre, ou tout du moins d'avoir de sérieuses notions,

sur la fonction et le rôle des autres participants au projet. Une telle rotation a le mérite de réduire la tendance à se sentir indispensable, et propose une diversité qui peut aider à diminuer toute sensation de surmenage. Cela implique qu'en cas de surmenage d'une personne, les autres membres du projet puissent en reprendre les rênes et les responsabilités avec un minimum d'adaptation.

Graner suggère aussi que les rôles d'un projet soient clairement définis, ce qui arrive rarement dans un projet distribué qui compte un grand nombre de bénévoles. De cette façon, les personnes seront moins susceptibles de prendre de nouvelles responsabilités.

À ces suggestions, Aurora ajoute que le surmenage peut aussi être atténué par « des expressions personnalisées de soutien venant de différentes personnes – envoyer un email qui dit *je pense que tu fais un très bon travail, et que tu as raison* peut souvent faire la différence ». En fait, Aurora explique que « n'importe quelle forme de reconnaissance » peut aider :

« Cela paraît trivial, mais toute forme de surmenage vient en partie du sentiment que ce que vous faites est insignifiant et n'est pas apprécié à sa juste valeur. Je pense qu'Internet est malheureusement un lieu propice à vous donner l'impression que ce que vous faites n'est pas apprécié. Les personnes critiques, voire mesquines, sont apparemment plus enclines à envoyer des reproches que les gentilles à formuler des remerciements, et l'absence de visages humains ou d'intonations de voix rend les incompréhensions courantes ».

Aurora cite sa propre expérience de mise en place, dans le cadre de son travail, d'une [politique anti-harcèlement](#) lors des conférences, entreprise dans un moment de quasi-surmenage, et qui a été accueillie avec tellement d'emails d'encouragement qu'elle aurait pu « en pleurer de joie ». Ce soutien aura été une reconnaissance fondamentale à un moment

crucial de sa vie.

Notons enfin le rôle important qu'ont les leaders de communautés pour améliorer les situations. Bacon suggère qu'un dirigeant ayant « un engagement quotidien au sein de sa communauté » est le mieux placé pour remarquer des signes de surmenage.

Bacon suggère également d'engager autour du surmenage une discussion franche et sincère, mais qui apporte également concrètement du soutien, quitte à proposer à certains de prendre un congé ou de réduire les responsabilités. Cette discussion devrait se faire en face à face si possible, au pire par téléphone, mais jamais par email ou chat, car le manque de signaux non-verbaux peut conduire à des incompréhensions, en particulier pour quelqu'un qui se sent déjà inadapté à son travail. Pendant cette conversation, le leader de la communauté doit bien faire comprendre qu'il n'est pas en train de réprimander, mais plutôt de donner impressions et suggestions dans l'intérêt de tous. Si nécessaire, on peut adoucir la perception de cette discussion en encourageant la personne à qui l'on parle à effectuer le même type d'intervention si le leader de communauté venait lui aussi à montrer des signes de surmenage.

Préparer le retour

Tout comme le surmenage n'a pas de cause unique, il n'y a pas de remède rapide, unique et miracle pour l'éviter ou le soigner. Cela signifie qu'il est difficile de prédire si les victimes du surmenage pourront surmonter leurs problèmes et revenir ou si elles vont simplement disparaître de la communauté. D'autant que l'on commence tout juste à accepter d'en parler explicitement et à étudier des stratégies de prévention.

En se rappelant son propre surmenage, Graner raconte : « mon surmenage n'est pas arrivé du jour au lendemain, pas plus que

le retour. Il y a eu des moments où je me disais que si je ne revenais pas, si je ne reprenais pas toutes mes activités précédentes, j'aurais échoué. Mais personne ne le pensait sauf moi. J'étais la seule qui me faisait des reproches, à me mettre ainsi la pression. J'ai été obligée de me dire : « non, tu n'es pas en situation d'échec, au contraire tu es désormais plus responsable. »

La bonne nouvelle, c'est que ceux qui reviennent d'un surmenage ont souvent une conscience plus aiguë de ce qui s'est mal passé, et éviteront plus facilement un nouveau surmenage. « Une fois que vous avez subi un surmenage total, vous êtes mieux à même de prévenir le prochain et modifier avant votre comportement en conséquence », explique Graner. Une telle prise de conscience représente sûrement l'une des armes les plus efficaces contre le surmenage, car ces personnes reviennent mettre en pratique et témoigner de ce qu'elles ont appris de cette difficile expérience.

Notes

[1] Crédit photo : [Perry McKenna](#) (Creative Commons By)

Librologie 5 : Troll en libertés

Chers lecteurs et lectrices,

Après une parenthèse [recueillie](#) la semaine dernière, la chronique [Librologique](#) que je vous propose aujourd'hui reprend le cheminement que nous avons entamé, et s'aventure dans le domaine, **trollophile** s'il en est, des soi-disant *libertés*

numériques^[1].

(Cet épisode trahit aussi une de mes habitudes irréprouvables : j'adore me faire des amis un peu partout.)

Bonne lecture, et à la semaine prochaine...

V. Villenave.



Librologie 5 : Troll en libertés

Pas de mythologie digne de ce nom sans troll... en tout cas dans le domaine des communautés d'internautes. Le [troll](#) est un rôle traditionnel de toute communication de groupe, amplement favorisé par le confort (anonymat relatif, éloignement,...) des échanges sur Internet : plus ou moins [sciemment](#), un ou plusieurs participants à la discussion mettent en avant une opinion dont la nature ou la formulation bloque le débat à [tous les niveaux](#), le plus souvent sous forme d'une polémique fallacieuse (sur le même sujet, voir également [cet article](#) d'un mien [collègue](#)). Que l'on puisse les mettre au compte de l'ignorance, d'une pulsion narcissique, d'une volonté de nuire ou du pur opportunisme politique, les comportements de troll nous intéresseront ici moins par leur origine que de par ce qu'ils révèlent de présupposés et, lorsqu'ils fonctionnent à grande échelle, d'imaginaire collectif : ainsi de cette tendance [documentée](#) de toute discussion houleuse et trolloïde à dériver vers les figures contemporaines du Mal : Hitler,

Staline ou Ben Laden.

Le troll est une composante historique du mouvement du logiciel Libre, où presque chaque programme se définissait par opposition avec un autre, le plus souvent propriétaire (emacs contre vi, Linux contre Minix, GNOME contre KDE/Qt) mais pas toujours (GNU emacs contre Xemacs, Guile contre Tcl, OpenBSD contre, hum, le reste du monde, etc.). De nos jours encore, tout Libriste qui se respecte peut posséder une opinion bien tranchée en faveur de tel logiciel plutôt que tel autre, et se répandre en commentaires acerbes : Linus Torvalds n'est [pas le dernier](#) à en donner l'exemple.

Est-ce à dire que, de même qu'on a pu parler du christianisme comme d'[une](#) « secte qui a réussi », Richard Stallman ne serait qu'un trolleur qui a réussi ? C'est certainement une possibilité à laquelle se prêterait la personnalité volontiers facétieuse de l'intéressé – et de nombreux commentateurs ne se privent pas de le [dénoncer](#) comme tel. Et pourtant, débusquer, ou croire débusquer, un troll, ne doit [pas](#) servir à éluder de véritables divergences idéologiques : nous [avons vu](#) que la rigueur intellectuelle de Richard Stallman, toute folklorique qu'elle se présente, repose sur une pensée précise et méthodique. Comme dans le cas de rms, la mythologie du troll présente ainsi le danger de faire oublier, sous l'aspect rituel du débat, les présupposés idéologiques qui le soutendent : ainsi la dispute terminologique entre *open-source* et *logiciel Libre* n'est-elle pas une dissension entre, d'un côté les idéologues, de l'autre les pragmatistes, mais bel et bien entre deux idéologies dont seulement l'une [se pense](#) et se [revendique](#) comme telle... tandis que l'autre, proche des milieux entrepreneuriaux ou capitalistes, épouse peu ou prou le discours dominant du libéralisme économique.

Dans un même ordre d'idées, un sujet de dissension classique (et donc à fort potentiel trollogène) est l'affrontement entre différentes licences plus ou moins Libres – voire, comme nous l'a [montré](#) le trolleur renommé qu'est Linus Torvalds, entre

deux *versions* de la même licence ! Là encore, il est souvent possible de distinguer un affrontement idéologique sous les points d'apparence purement technique – nous y reviendrons prochainement, avec l'application des licences Libres au domaine culturel. En effet, si la substance du troll équivaut en général à un antagonisme de personnes, son essence est d'ordre idéologique : dire « je ne t'aime pas » ne suffit pas, il faut ajouter « je ne t'aime pas car je n'aime pas les apprentis-Staline » pour déplacer le débat sur un champ idéologique.

Cela n'est, cependant, que la première étape de la constitution du troll : la suivante incombe à son public lui-même, lorsque celui-ci entre dans la danse et qu'une polémique se crée – le succès du troll se mesurant au temps et à l'énergie qu'il consomme, directement ou indirectement. D'où la [maxime](#) *Don't feed the troll*, « prière de ne pas donner de nourriture aux trolls ». L'usage même de cette maxime, au demeurant, pose question : si c'est véritablement d'un troll qu'il s'agit, alors il devrait être suffisamment outrancier pour être aisément désamorcé : dans notre cas la personne à qui l'on s'adresse n'a que très peu de points communs avec un dictateur soviétique, et sera trivial de le démontrer et mettre ainsi fin à la discussion. Cependant cela demande de part et d'autre – outre un minimum de bonne foi – une rigueur conceptuelle, une finesse d'analyse et de compréhension, et une culture politique qui souvent fait, hélas, défaut. C'est donc sur cette faille que s'édifie le troll ; y compris dans le milieu dit de « défense des libertés numériques », sur lequel nous nous arrêterons plus longuement ci-dessous.

La « communauté » des internautes, au sens large, est habituée à recevoir des attaques émanant des classes politiques, médiatiques et, de façon générale, traditionnellement légitimées qui, telles un M. Jourdain se retrouvant sur AOL, « trollent sans le savoir » ([certains gouvernants français](#) s'en sont même fait une spécialité) : en général la thématique

mise à contribution est celle qui consiste à faire du réseau Internet le repaire des Ennemis de la société : le terroriste, le pédophile, le « pirate », l'islamiste, le nazi, et plus généralement, le jeune (adepte de jeux vidéos violents ou décérébrants, du « *tout-gratuit* » que j'ai [déjà abordé ailleurs](#)...). Ne disposant pas du bagage culturel, juridique et technique qui caractérise la plupart des internautes proches du mouvement Libre, de tels trolleurs-malgré-eux (pour rester chez Molière) n'hésitent pas à s'en prendre à des principes qui, pour le reste des citoyens, relèvent de l'évidence : [interopérabilité](#), [neutralité du réseau](#), [liberté](#) de choisir des licences alternatives... Pour officielles qu'elles soient, ces techniques de trollage n'en sont pas moins, parfois, d'une efficacité redoutable : au moment où je rédige ces lignes, les communautés de citoyens, consommateurs, artistes et Libristes de notre pays ont consacré la quasi-totalité de leur énergie et de leur attention, pendant quatre ans, au [plus gros Troll administratif](#) de la décennie : j'ai nommé la loi dite « Hadopi », sous ses différents avatars.

On aurait pu penser que des internautes habitués aux dissensions et discussions enflammées sauraient reconnaître, et échapper à, toute manœuvre de troll de la part du gouvernement – technique d'ailleurs bien connue de tous les politiciens, que celle qui consiste à lancer quelques propositions bien choquantes pour faire passer, sous les espèces du compromis, d'autres mesures plus pernicieuses. Mais comme dans tout contexte de lutte, la « communauté » se définit davantage par ce à quoi elle s'oppose ou ce qu'elle moque, que par des valeurs communes (même si elle s'en défend, nous y reviendrons précisément ci-dessous). De là vient qu'elle puisse quelquefois, à ses contempteurs, présenter l'illusion d'un front uni ; il n'en est pourtant rien, et les *trolls* abondent au sein du mouvement Libre et des « libertés numériques » comme de toute communauté sur Internet – peut-être même davantage, tant les sujets de dissension y sont nombreux et les individualités, disons, peu accommodantes.

Si j'emploie cette expression de « libertés numériques », ce n'est qu'avec la plus grande suspicion. On l'a vue, certes, [consciencieusement brandie](#) (et [brandée](#)) par le [gratin](#) des [activistes](#) et [entrepreneurs](#), jusqu'à servir d'unique justification à des [groupuscules](#) ou [coups publicitaires](#) aux implications [politiques](#) parfois [fumeuses](#). Mais que dit-on *vraiment* lorsque l'on parle « libertés numériques » ? Le mot Liberté est évidemment déjà éminemment chargé d'idéologie, plus encore lorsqu'il est au pluriel : la liberté ne serait donc pas un absolu ? (Un autre [collègue](#) me rappelle ici judicieusement qu'il en va de même pour le mot Laïcité, auquel d'aucuns gouvernants réactionnaires ont jugé bon d'[adjoindre des adjectifs](#) pour mieux le [vider de son sens](#).)

De [quelles](#) libertés parle-t-on alors ? Libertés civiques ? Collectives ? Individuelles ? Rien de tout cela en fait : puisqu'on vous dit qu'il s'agit simplement de libertés « numériques ». Ce qui permet finalement à chacun de regrouper sous un même parapluie ses propres présupposés idéologiques, parfois antinomiques – par exemple entre ceux pour qui la Liberté passe par un État garant de la cohésion sociale à ceux pour qui, au contraire, l'État est une entrave à l'aspiration de liberté individuelle totale. Comme si le domaine numérique n'était pas un *moyen* (de communication et d'expression, qui ouvre certes des espaces publics ou privés où la Liberté doit certainement être défendue comme partout ailleurs), mais un *enjeu* en lui-même : politique, commercial, consumériste... peu importe en fait : seul reste le slogan. « Libertés numériques ! »

Un slogan qui, en dernière analyse, ne séduit ni ne convainc... À tel point que je ne puis que me demander si ce n'est vraiment que par effet de mode qu'on l'emploie à l'envi. Le plus intéressant ne serait-il pas ici ce que l'on ne *dit pas* ? Parler de « libertés numériques », c'est éviter de se référer aux valeurs de *Liberté, Égalité, Fraternité* (pourtant chères à [Richard Stallman](#)), ou encore aux *Droits de l'Homme et du*

Citoyen ? (Cela expliquerait la présence, inexplicable autrement, parmi les défenseurs des « libertés numériques », de sympathisants de [formations politiques](#) ou de [polémistes réactionnaires](#) qui se plaisent à brocarder ceux qu'ils appellent les « droitsdelhommistes ».) Processus familier, somme toute : les « libertés numériques » sont finalement aux Droits de l'Homme ce que l'*open-source* est au mouvement Libre.

Pour maladroit ou imprécis qu'il puisse être, le paysage associatif français autour de ces prétendues « libertés numériques » n'en est pas moins un milieu attachant et nécessaire : dans leur grande majorité, ces associations sont mues par des personnes de bonne volonté dont l'éthos et le dévouement vont largement au-delà du troll-de-base. Ce milieu, je le fréquente moi-même depuis de nombreuses années et il m'a été donné d'y contribuer personnellement, en tant que sympathisant de nombreuses associations et, *last but not least*, spectateur puis animateur de l'un des [nids à trolls](#) les plus prometteurs de ces dernières années – et dont nous serons amenés à reparler dans ces chroniques.

Les querelles de chapelles sont monnaie courante dans un tel environnement, qui nous donne chaque semaine de nouveaux exemples de groupes (ou groupuscules) dont la principale raison d'être semble se résumer à taper sur des [collectifs](#) ou [projets](#) existants. Ces divisions (au sens [biologique](#) du terme) peuvent être mises en rapport avec la culture du [fork](#) dans le Logiciel Libre, qui permet à n'importe quel développeur de reprendre le code d'un programme pour le faire évoluer dans une direction différente. Si le phénomène n'est pas simple dans le monde informatique (le succès d'un *fork* dépend de sa qualité, du charisme de son initiateur, de l'éventuel mécontentement des utilisateurs vis-à-vis du projet d'origine...), dans le domaine des idées il touche à l'ésotérisme : entre 2006 et 2010, il m'a été donné de dénombrer pas moins de sept *forks* différents du [Parti Pirate](#), rien qu'en France !

Dans ce milieu où les discussions s'enflamment vite, le troll n'est donc jamais loin, non seulement d'un groupe à l'autre (nous le voyions à l'instant) mais au sein même du groupe. C'est que la nature des thématiques abordées prête souvent aux confrontations politiques et au trollage idéologique ; à cela s'ajoute ce que je nommerais le « troll structurel », qui me semble ontologiquement propre à ce milieu. En effet, le mode de fonctionnement des communautés en ligne (particulièrement dans le mouvement Libre), que l'on a pu (trop facilement, à mon sens) décrire comme « horizontal », « anarchique » ou « méritocratique », repose sur des règles non-écrites, diffuses, changeantes, et des valeurs inquantifiables telles que la confiance mutuelle, la sympathie ou le degré de conviction qu'inspire chaque membre. À l'inverse, toute association républicaine se doit structurellement de présenter un édifice « vertical », « représentatif » ou « démocratique » : des élections, des assemblées, une hiérarchie de commandement et de responsabilités... Je n'irai pas jusqu'à dire que concilier ces deux modèles est *impossible* ; je n'hésiterai pas à dire, en revanche, qu'à l'heure actuelle *personne*, à ma connaissance, n'y est parvenu. L'attitude adoptée par la plupart des responsables associatifs consiste à simplement ignorer le problème, en laissant se développer une « communauté » plus ou moins appariée avec l'association, et qui peut constituer un vivier de ressources bénévoles – à moins que ce ne soit l'association qui se donne pour mission d'« animer » la communauté. Une telle ambiguïté ne peut que profiter à l'émergence des trolls, particulièrement dès que croissent le nombre d'adhérents ou les ressources financières. L'embauche d'un salarié permanent constitue, à ce titre, un cap révélateur : dans beaucoup de cas l'association le recrutera parmi ses propres fondateurs, comme en gage d'ancienneté et de reconnaissance, dans d'autres cas l'on recrutera un jeune « battant » issu de Sciences-Po ou d'une école de commerce ; c'est seulement dans le plus rare des cas que l'on recrute parmi la « communauté », au moyen de critères qui ne peuvent que faire question : de quelque façon

qu'elle s'y prenne, l'association qui franchit ce cap trahit de ce fait son aspiration de légitimation sociale, et par là les *habitus* de ses membres même.

Je ne saurais trop souligner la grande diversité de toutes ces structures, qui peuvent aller du groupuscule informel à de véritables institutions, et qu'il serait pour le moins hasardeux de prétendre décrire en quelques phrases. Cependant, quelle que soit leur taille, il me semble pouvoir distinguer quelques traits récurrents : le plus frappant et le plus répandu étant sans doute, leur étrange pudeur sur le plan politique, que l'on peut voir, soit comme un simple manque de courage (j'y reviens dans un instant) soit comme un idéologème caché : nous [avons vu](#) combien la parole « dépolitisée » que dénonce Barthes peut masquer une démarche activement réactionnaire (au sens contre-révolutionnaire du terme), et il n'est [plus à démontrer](#) qu'[apolitique veut dire « de droite »](#).

Ainsi de cette insistance répétée, largement répandue et, en fin de compte, presque suspecte, qu'a le milieu du logiciel Libre et des « libertés numériques » à... *ne pas faire de politique*. Dans un exemple frappant, l'[Aful](#), pour critiquer l'April, nous explique-telle doctement qu'elle ne se place pas « sur le plan des principes »... avant, dans la *phrase suivante*, de faire sienne la notion de « concurrence libre et non faussée », qui a depuis longtemps [cessé](#) d'être neutre politiquement ! D'autres associations, si elles se défendent également de *faire de la politique*, ont des liens [plus](#) ou [moins](#) ténus avec [certaines formations politiques](#) existantes, des sponsors [inattendus](#), une organisation souvent [opaque](#), ou des [motivations](#) quelquefois peu claires – au point que l'on soit parfois amené à se demander si dans certains cas, le travail des bénévoles et les (nombreuses et insistantes) campagnes d'appel au don, toujours au nom de l'Intérêt Général, n'ont pas pour finalité véritable de financer les émoluments et ambitions politiques ou entrepreneuriales de quelques individus.

Ambitions qui, je m'empresse de le dire, ne sont pas nécessairement antinomiques d'un engagement par ailleurs sincère et d'un travail de qualité. D'un point de vue intellectuel, l'ambition personnelle ou même le simple besoin de devoir faire bouillir ses spaghettis ne me paraissent pas moins dignes de respect que la défense du Bien Commun – tant que l'on n'invoque pas celui-ci pour masquer celui-là.



Quelle que puisse être la part de calcul, de romantisme ou d'idéalisme de leurs figures de proue, l'effervescence des associations et communautés de l'Internet Libriste ou « citoyen » d'aujourd'hui ne doit pas faire oublier la relative pauvreté de leur réflexion politique et de leur culture idéologique. J'en veux pour illustration la danse du ventre des courants politiques traditionnels qui se pressent devant l'électorat *geek* : aux partis traditionnels il faut ajouter les formations écologistes, centristes, ultra-libérales, nationalistes... pour ne citer que celles qui se sont achetée une image « libertés numériques-*friendly* » en s'opposant (au moins en apparence) à ce troll dit « Hadopi » que j'évoquais plus haut. Et le mythe est ici d'autant plus aisé, d'autant plus séduisant, que l'éthos de son « cœur de cible » *geek* reste mal défini, et sa terminologie, mal conceptualisée.

Ainsi pourrait s'expliquer, également, le succès de certaines personnalités politiques « traditionnelles », qui font à l'occasion l'objet d'un engouement soudain et inattendu auprès des geeks et internautes, quasi indépendamment de leurs valeurs : François Bayrou lors de la fondation du bien-nommé [MoDem](#), Nathalie Kosciusko-Morizet sur [Twitter](#), Michel Rocard sur le [logiciel Libre](#), ou bien sûr l'[accession au pouvoir](#) du candidat Obama. Venant d'une communauté habituellement si critique et moqueuse, la chose a de quoi surprendre : j'y verrai même des exemples de trolls *positifs*.

Ici se trouve sans doute une justification de cette posture « apolitique » que je critiquais plus haut... et qui explique d'ailleurs peut-être le succès de certains discours « ni de gauche, ni de droite » (bien au contraire) tels que celui de François Bayrou en 2007 ou d'Europe-Écologie en 2009 auprès des geeks français) : de même que la grande majorité des Libristes (à l'exception de rms, nous l'avons vu) et, plus encore, des partisans de *l'open-source* fuit explicitement toute ambition philosophique ou idéologique, beaucoup de groupements de citoyens sur Internet se détournent avec dégoût ou terreur de la politique en tant que telle. Certes, cela n'est ni nouveau, ni inhérent aux communautés en ligne : les associations traditionnelles (y compris, d'ailleurs, au sein des formations politiques ou syndicales) ne sont pas, pour la plupart, des clubs de réflexion ni des regroupements d'idéologues ou d'intellectuels – nulle raison pour qu'il en soit autrement sur Internet.

Attitude compréhensible, quoiqu'un tantinet hypocrite : comment en effet promouvoir des modèles inédits de diffusion culturelle, de lien social et d'équilibre économique, sans en tirer les conclusions politiques et institutionnelles inévitables ? En évitant, sinon les trolls (nous avons vu que c'est peu ou prou impossible), du moins les « sujets qui fâchent », ne risque-t-on pas d'abdiquer certaines convictions, et de se contenter (comme nous le voyions plus

haut) de remâcher en fait le discours idéologique dominant, fût-ce sous une forme dégradée ?

Cette stratégie d'évitement (qu'elle soit assumée ou non) en laquelle je vois une certaine hypocrisie, me frappe d'autant plus qu'elle vient (majoritairement) de classes sociales qui ont pourtant accès à la connaissance (notamment historique), à des sources d'information multiples et à des modes de pensée favorisant l'esprit critique – non seulement vis-à-vis du Système politico-médiatique traditionnel (ce point semble acquis), mais également au sein même de ce nouveau *mainstream* qu'elles créent, comme [nous le mentionnions](#) en préambule de ces chroniques. On a pu souligner l'importance humanitaire de l'éducation dans les pays soumis aux guerres, famines, épidémies ; faudrait-il, dans nos contrées virtuelles ravagées par les trolls – fléau certes autrement moins grave – envisager comme prérequis à l'émergence d'une représentation politique efficace de leurs modèles de société, la nécessité pour les internautes Libristes d'une culture politique et d'une certaine rigueur intellectuelle ?

La chasse au troll n'est donc pas plus neutre idéologiquement que le troll lui-même : tous deux participent d'une même ritualisation des débats d'idées, qui les vide peu à peu de tout contenu politique et de toute aptitude à influencer sur l'évolution des choses. Si le troll a un effet parasite et immobilisant, sa répression laisse quant à elle entendre qu'aucune divergence idéologique de fond n'est efficace, justifiée ou même envisageable. La possibilité même d'un débat rigoureux et approfondi, s'étiôle : c'est de cet étiolement que le troll est indice.

Notes

[1] Crédit illustrations sur Wikimedia Commons : [Internet Troll](#) (licence Art Libre) et [Troll nicht fuettern](#) (Creative Commons By-Sa)

La vidéo francisée des 20 ans du noyau Linux

Le noyau Linux a 20 ans. C'était en effet le 25 août 1991 que [Linus Torvalds](#) (qui fait justement l'[objet de la librologie de la semaine](#)) a en effet posté son [célèbre message](#).

Pour l'occasion la [Linux Foundation](#) nous a proposé un petit clip anniversaire que nous avons non seulement sous-titré (merci Framalang) mais également doublé en français (merci Padoup-Padoup).

La vidéo en VO STFR



- > La [vidéo](#) au format webm
- > Le fichier de [sous-titres](#)

La vidéo en VF



- > La [vidéo](#) au format webm

Sur le même sujet, on pourra également lire la très intéressante [interview](#) donnée par Torvalds à LinuxFr.

Transcript

[URL d'origine du document](#)

L'histoire de Linux (à l'occasion de son 20e anniversaire)

Notre histoire commence il y a vingt ans. Boris Elstine prenait ses fonctions, Jay Leno remplaçait Johny Carson au Tonight Show et les téléphones portables étaient très très

gros.

C'est en août 1991 qu'un étudiant en informatique de 20 ans nommé Linus Torvalds s'est assis devant son écran à Helsinki pour envoyer un des plus célèbres messages de l'histoire de l'informatique

« Salut tout le monde... Je suis en train de faire un système d'exploitation libre (c'est juste un passe-temps, rien d'aussi professionnel ni énorme que GNU) il ne prendra sûrement jamais rien d'autre en charge que des disques durs AT puisque c'est tout ce que j'ai »

Et bien, son annonce de projet open source a vite fait le tour du monde et des développeurs de tous les coins ont contribué au code.

Linus a baptisé le noyau de son système d'exploitation Linux en choisissant un manchot comme mascotte après un petit incident au zoo.

Il a pris assez vite une décision qui allait être déterminante pour l'avenir de Linux autant que sa technologie. Il a choisi la licence GPL, créée par un visionnaire nommé Richard Stallman.

Le noyau Linux, accompagné de la licence GPL et d'autres composants GNU, ont révolutionné l'industrie informatique avec une liste de libertés très simples mais très importantes : La liberté d'utiliser le logiciel à son gré. La liberté de modifier le logiciel pour l'adapter à ses besoins. La liberté de partager le logiciel avec ses amis et ses voisins. Et la liberté de distribuer les modifications qu'on a faites

Ces principes radicaux ont propulsé la diffusion de Linux partout dans le monde et de façon un peu paradoxale Linux qui était une expérience d'amateur est devenu la base d'un vaste écosystème commercial qui se développe.

Des entreprises ont basé leur activité sur Linux. En 1999 le cours de l'action de Red Hat a triplé en devenant la première entreprise à utiliser Linux sur le marché. Cette même année IBM a dépensé un milliard de dollars pour améliorer et promouvoir Linux

(Et comment il s'appelle ? – son nom est Linux)

Rapidement, Linux a bousculé les poids lourds de l'industrie et a permis le développement accéléré d'Internet avec ses logiciels libres.

Bref : Linux a révolutionné le monde de l'informatique.

Bien sûr, dès qu'un phénomène aussi novateur s'impose, il essuie un feu croisé de critiques mais Linux ne s'est pas contenté de survivre, il a pris de l'ampleur. Aujourd'hui la communauté de développement du noyau compte des milliers de membres, et des centaines d'entreprises qui collaborent ensemble au développement de Linux. Tous les trois mois une nouvelle version de Linux voit le jour.

Donc où en est Linux aujourd'hui ? Il permet 75% des transactions boursières dans le monde, Il fait tourner les serveurs d'Amazon, de Facebook, de Twitter, d'eBay, et de Google. Vous vous servez de Linux littéralement à chaque fois que vous allez sur internet. C'est dans votre téléphone, votre télé, sur 95% des superordinateurs, et dans beaucoup d'autres appareils que vous utilisez tous les jours. Linux est partout.

Et qu'est devenu le programmeur d'Helsinki qui a tout commencé ? Il orchestre le travail de cette armée mondiale de développeurs depuis le bureau de sa maison à Portland en Oregon, en tant que membre de la Fondation Linux.

Alors que nous célébrons les 20 ans de Linux, nous pouvons nous retrouver dans cette histoire.

Merci d'avoir contribué à cette saga tout au long de ces 20

ans.