

# Geektionnerd : Conseil National du Numérique

## CNN (PAS LA CHAÎNE)

Conseil National du Numérique mis en place cette semaine pour « conseiller » la politique à suivre en matière de numérique (en matière de 0 et de 1, donc).



Le ~~candidat~~ président Sarkozy en a profité pour reconnaître qu'Hadopi était une « maladresse » (c'est ce qu'on appelle avoir le sens de l'euphémisme).



Pendant ce temps-là, le site de l'Élysée précise qu'Hadopi n'est absolument pas remise en cause.



Bah quoi ? Moi je le trouve en plein dans le sujet, ce dessin. . .

Crédit : [Simon Gee Giraudot](#) (Creative Commons By-Sa)

## Montrez-nous les données ! Ce sont les nôtres après tout...

Le monde de plus en plus numérique qui se met en place présente le risque et le paradoxe de nous déposséder d'une part de nous-mêmes, celle qui laisse toujours plus de traces sur le réseau.



Dans un récent article publié par le New York Times, le professeur économiste [Richard Thaler](#) en appelle à plus de

transparence quant à l'utilisation commerciale de nos données<sup>[1]</sup>.

Mais il en appelle surtout à la possibilité de réutiliser nous-mêmes nos propres données, ce qui est loin d'être le cas actuellement.

Et la situation ne pourra évoluer que si les utilisateurs sont de plus en plus nombreux à prendre conscience du problème en mettant alors la pression sur les structures qui exploitent ces données.

## **Nos données nous appartiennent. Montrez-les-nous !**

[Show Us the Data. \(It's Ours, After All.\)](#)

*Richard Thaler – 23 avril 2011 – The New York Times*

*(Traduction Framalang : Goofy et Don Rico)*

Nul ne sait mieux que moi ce que j'aime.

Cette affirmation peut passer pour une évidence, mais la révolution des technologies de l'information a généré une liste croissante d'exceptions. Votre épicier sait ce que vous aimez manger et peut probablement vous donner des conseils judicieux et appropriés sur d'autres aliments qui pourraient vous plaire. Votre opérateur téléphonique sait qui vous appelez, et votre téléphone sait où vous êtes allé. Quant à votre moteur de recherche, il peut anticiper vos désirs avant même que vous ayez achevé de les saisir au clavier.

Les entreprises accumulent des masses considérables d'informations sur ce qui vous plaît ou pas. Mais ce n'est pas seulement parce que vous êtes digne d'intérêt. Plus elles en savent sur vous, plus cela leur rapporte d'argent.

La récolte et la diffusion de ces informations soulève une quantité de problèmes de confidentialité, bien entendu, et un

tandem de sénateurs des deux camps, [John Kerry](#) et [John McCain](#), a proposé de les régler avec leur [Commercial Privacy Bill of Rights](#) (*Ndt : Déclaration des Droits à la confidentialité dans le commerce*). Protéger notre vie privée est important, mais la démarche des deux sénateurs fait l'impasse sur un problème plus important : elle n'inclut pas le droit d'accès à nos propres données personnelles. Non seulement nos données devraient être protégées, mais elles devraient aussi être disponibles pour que nous puissions les utiliser selon nos propres besoins. Après tout, ces données nous appartiennent.

Voici un principe de base : si une entreprise commerciale collecte électroniquement les données des utilisateurs, elle devrait leur fournir une version de ces informations facile à télécharger et à exporter vers un autre site Web. On peut résumer cette démarche ainsi : vous prêtez vos données à une entreprise, et vous en voudriez une copie pour votre usage personnel.

Le gouvernement de la Grande-Bretagne [vient d'annoncer](#) une initiative intitulée « mydata » qui va dans ce sens (j'ai travaillé comme consultant pour ce projet). Bien que les lois britanniques demandent déjà aux entreprises de donner à leurs clients des informations sur l'utilisation de ces données, le programme vise à fournir des données accessibles via un ordinateur. Pour commencer, le gouvernement travaille en concertation avec plusieurs grandes banques, les émetteurs de cartes de crédit, les opérateurs et revendeurs de téléphones mobiles.

Pour comprendre comment un tel programme pourrait améliorer la façon dont fonctionne le marché, songez par exemple à la façon dont vous choisissez un nouvel abonnement à un service de téléphonie mobile. Deux études [ont démontré](#) que les consommateurs [pouvaient économiser](#) plus de 300 dollars chaque année en souscrivant un abonnement mieux adapté. Mais pour cela, il faut être capable d'estimer ses besoins en termes de services : SMS, médias sociaux, musique en streaming, envoi de

photos, etc.

Il se peut que vous ne soyez pas en mesure de traduire tout cela en mégaoctets, mais votre opérateur lui, en est capable. Bien que certaines informations soient déjà disponibles en ligne, elles ne se trouvent généralement pas encore disponibles dans un format exportable – vous ne pouvez pas les couper-coller facilement sur un autre site, un comparateur de prix par exemple – et elles ne se présentent pas de telle manière qu’il vous soit facile de calculer quel est le meilleur abonnement pour vous.

Si l’on suit la règle que je propose, votre opérateur vous donnerait accès à un fichier comprenant toutes les informations qu’il a récoltées depuis que vous avez un mobile, ainsi que toutes les factures en cours pour chacun des services que vous utilisez. Les données vous seraient remises dans un format utilisable par les créateurs d’applications, si bien que de nouveaux services pourraient voir le jour, proposant aux consommateurs des conseils pratiques (pensez à [Expedia](#), par exemple). Ainsi, ce cercle vertueux créerait des emplois pour ceux qui ne rêvent que de lancer ce genre de nouveaux sites Web.

Avant de se plaindre qu’il est difficile de se soumettre à une telle règle, les entreprises devraient jeter un coup d’œil à une initiative du gouvernement fédéral appelée [Blue Button](#). Cette procédure déjà en vigueur offre aux anciens combattants et bénéficiaires de [Medicare](#) (*NdT : programme de sécurité sociale pour personnes âgées*) la possibilité de transmettre leur dossier médical à un organisme de confiance (le nom « Blue Button » fait allusion au bouton bleu sur lequel peut cliquer l’utilisateur qui désire récupérer ses données).

L’initiative Blue Button se répand déjà dans les applications du secteur privé. Northrop Grumman a développé une application pour smartphone qui permet aux anciens combattants d’accéder à leur dossier médical et de recevoir sur leur téléphone des

conseils de santé pour être en bonne forme. [HealthVault](#), un site Microsoft pour l'organisation de ses soins de santé, permet également aux utilisateurs de Blue Button d'y rechercher leurs informations médicales. La possibilité d'accéder à ces différents types de services pourrait sauver des vies en cas d'urgence.

Si le gouvernement est capable de collecter et restituer des informations confidentielles de manière sécurisée et utile, les entreprises privées peuvent en faire autant, ce qui donnera davantage de chances aux consommateurs d'être des clients plus avisés.

Revenons à l'exemple des smartphones. Une fois que le propriétaire d'un téléphone fournit ses informations personnelles à des sites Web tiers, ceux-ci ([BillShrink](#), par exemple) peuvent l'aiguiller vers les abonnements de meilleur rapport qualité/prix. Vous envisagez de changer votre téléphone ? Les sites tiers peuvent vous avertir si votre utilisation risque de s'accroître, en se basant sur l'expérience d'utilisateurs qui ont fait avant vous le même changement.

Si les données personnelles sont accompagnées d'informations détaillées sur les coûts, comme je l'écrivais dans [mon dernier article](#), les consommateurs connaîtront mieux la façon dont ils utilisent vraiment les services, ainsi que leur coût réel. La tarification transparente, elle, donnera un avantage compétitif aux fournisseurs honnêtes et de qualité sur ceux qui ont des pratiques opaques. Ces éléments permettront une croissance économique saine.

Les applications possibles sont innombrables. Les supermarchés, par exemple, savent déjà qu'ils peuvent attirer plus de clients dans leurs clubs de consommateurs en offrant des réductions exclusives à ceux qui en font partie. Ce qui permet aux magasins de connaître les habitudes de consommation des clients et de cibler les bons de réduction d'après leurs

achats. Les clients peuvent se désinscrire – mais alors ils perdront leurs réductions.

Exigeons donc que ce soit à double sens. Pourquoi ne pas vous donner, à vous consommateur, quelque chose en échange de votre participation ? Exigez du supermarché qu'il vous fournisse l'historique de vos achats. Il ne se passera pas longtemps avant qu'un entrepreneur astucieux ne vous concocte une application capable de vous indiquer des solutions de remplacement moins coûteuses et plus saines, qui seront aussi bonnes pour votre ligne que pour votre compte en banque. Les applications ne servent pas qu'à économiser de l'argent ; elles pourraient aussi avertir les clients souffrant d'allergies, par exemple, qu'ils achètent des aliments contenant des ingrédients auxquels ils sont sensibles, comme les arachides ou le gluten.

La capacité qu'ont les entreprises à surveiller notre comportement fait déjà partie de notre quotidien, et ce n'est qu'un début. Nous devons évidemment protéger notre droit à la confidentialité, mais si nous sommes malins, nous utiliserons également les données qui sont collectées pour améliorer notre existence.

J'espère que les entreprises américaines suivront l'exemple de leurs homologues britanniques et coopéreront au programme « mydata ». Sinon, nous exigerons des entreprises qu'elles nous indiquent ce qu'elles savent déjà sur nous. Pour paraphraser Moïse, demandons-leur : « laisse aller mes données, afin qu'elles me servent »

*Richard H. Thaler est professeur d'économie et des sciences du comportement à l'école de commerce Booth de l'Université de Chicago.*

## Notes

[1] Crédit photo : [Ernst Vikne](#) (Creative Commons By-Sa)

---

# Vies parallèles : une BD pour semer la liberté chez les enfants

Deux familles voisines entrent dans un magasin d'informatique pour s'acheter un ordinateur. Le choix des uns diffère radicalement de celui des autres. Mais heureusement, ça se termine bien à la fin et la grande soeur conclure : « Regardez les enfants, utiliser des logiciels libres donne envie aux gens d'aider les autres et de collaborer ! ».



Il existe [une réelle dynamique](#) en Amérique latine autour du logiciel libre.

Ayant compris son intérêt éducatif, plusieurs initiatives ont vu le jour. En Argentine, l'ONG [Gleducar](#), composée d'enseignants, d'étudiants et d'activistes du logiciel libre, s'est fixée pour mission, depuis sa création en 2002, de favoriser le libre dans le monde de l'éducation.

Pour atteindre cet objectif, l'association développe de nombreux projets :

- le don aux écoles et aux bibliothèques d'ordinateurs recyclés livrés avec des logiciels libres





---

# Geektionnerd : OpenOffice Libéré

[OpenOffice.org](http://OpenOffice.org) n'est pas mort, Vive LibreOffice ! titrions-nous il y a trois mois.

Nouveau rebondissement cette semaine : [Oracle met fin à l'exploitation commerciale d'OpenOffice](#).

La suite au prochain épisode...

## OPENOFFICE LIBÉRÉ

Suite dans la grande saga Oracle au Pays des Logiciels Libres, l'entreprise a décidé de laisser OpenOffice à la communauté.



La question se pose donc : que va-t-il advenir de LibreOffice, la dépendance à Oracle étant la raison du fork ? Va-t-on encore chambouler l'organisation du tout ?



Crédit : [Simon Gee Giraudot](#) (Creative Commons By-Sa)

## Le temps de cerveau disponible des utilisateurs de Google Chrome

Nous nous faisons régulièrement l'écho de la [concurrence](#) entre les navigateurs Mozilla Firefox et Google [Chrome](#), et ce [au sein même](#) de la communauté du *Libre*.



Faites comme vous voulez mais ce petit rappel sur les réelles motivations de Google n'est pas inutile, car on comprend alors

mieux pourquoi Google fournit tant d'efforts pour développer et faire connaître son navigateur<sup>[1]</sup>.

Et nous sommes tout d'un coup bien loin du [Mozilla Manifesto](#)...

## **Le véritable intérêt de Chrome pour Google : l'utilisateur captif**

[Why is Chrome so important to Google? It's a 'locked-in user'](#)

*Larry Dignan – 14 avril 2011 – ZDNet*

*(Traduction Framalang : Vincent, Don Rico et Goofy)*

Au départ, le navigateur Chrome n'est apparu que comme un projet secondaire de plus pour l'entreprise Google. Bien sûr, c'est un navigateur rapide. Bien sûr, Google a engrangé d'impressionnantes parts de marché en un temps record. Et bien sûr Google a donné un coup de fouet au développement des navigateurs.

Mais l'importance véritable de Chrome – qui explique pourquoi Google fait un tel battage publicitaire pour son navigateur dans le monde entier –, se résume à la valeur du capital client (NdT : [customer lifetime value](#)) de l'utilisateur de Chrome.

Au cours d'une conférence de presse sur les revenus de l'entreprise, on a demandé au directeur financier [Patrick Pichette](#) quelle était l'importance de Chrome, et sa réponse est édifiante.

*En fait, Chrome a deux aspects distincts. Côté pile c'est une arme tactique, côté face c'est une arme stratégique. Chrome bouscule vraiment le Web et il offre un avantage extraordinaire : dès qu'un internaute l'adopte, il accède directement à ses requêtes Google via la barre d'adresses au lieu d'avoir à trouver la barre de recherche. D'un point de vue stratégique, il est intégré dans le système*

d'exploitation [Chrome OS](#). D'un point de vue tactique, nous avons la garantie que tous les utilisateurs de Chrome restent captifs du moteur de recherche Google.

Nikesh Arora, directeur commercial chez Google, a lui aussi donné son analyse du retour sur investissement du marketing de Chrome.

*Nous avons plus de 120 millions d'utilisateurs quotidiens et 40% d'entre eux sont arrivés récemment après notre campagne de publicité. L'usage de Chrome a progressé de 30% d'un trimestre à l'autre. Je pense donc que la stratégie de Chrome est la bonne. Voilà comment nous diffusons Chrome : soit les gens l'adoptent d'eux-mêmes, soit ils le font suite à nos efforts de marketing, ou bien encore nous travaillons avec des partenaires qui nous aident à le promouvoir.*

*Dans ce contexte, nous avons donc estimé que la campagne marketing produit finalement un retour sur investissement bien meilleur que lorsque nous devons passer des accords de partenariat. Vous verrez un jour ou l'autre que les coûts d'acquisition du trafic et celui du marketing de Chrome sont interchangeables. Quand nous investissons dans le marketing de Chrome, c'est autant que nous n'avons pas à sortir pour le coût d'acquisition du trafic. Vous pouvez donc vous attendre à ce que nous continuions à mener cette stratégie pour Chrome parce qu'il ne s'agit pas seulement pour nous d'un avantage spécifique de Chrome, mais parce que nos autres produits liés à Chrome en bénéficient. Le temps consacré à Chrome par chaque utilisateur a donc pour nous une valeur phénoménale.*

Si l'on extrait l'essentiel de ces commentaires, voici ce qu'on peut retenir de Chrome :

- Chaque utilisateur de Chrome contribue par son usage à l'amélioration de la qualité de la recherche.
- Chaque utilisateur de Chrome utilisera Google de façon

massive et presque exclusive.

- Google peut dépenser moins en coût d'acquisition du trafic.
- L'entreprise n'a pas besoin de dépenser autant pour la diffusion de son produit – pensez par exemple au budget consacré à la boîte de recherche dans Firefox (*NdT : et qui fait vivre en grande partie Mozilla*).
- Enfin, il y a de grandes chances que l'utilisateur captif augmente encore sa consommation de produits Google à mesure que l'entreprise développe ses applications pour Chrome et les fonctionnalités sociales qui gravitent autour.

Au passage, Chrome permet aussi à Google de mieux faire respecter les standards des navigateurs, mais ce retour sur investissement semble décidément bien naïf comparé au profit que représente le temps de cerveau disponible de l'utilisateur de Chrome.

## Notes

[1] Crédit photo : [Elliott Brown](#) (Creative Commons By-Sa)

---

# Geektionnerd : Blender 2.57

[Blender](#) 2.57 vient de [voir le jour](#). Pour une visite guidée c'est [ici](#).

# BLENDER 2.57

Le célèbre logiciel libre de modélisation 3D fait peau neuve.

(Et c'est pas du luxe !)

Je me souviens encore du menu planqué en haut et qu'il fallait trouver en faisant un glisser-déposer JUSTE sur le pixel en-dessous de la barre de menu...



Oui je sais, il y a aussi plein de nouvelles fonctionnalités géniales, mais étant un novice, c'est l'interface que je vois. . .

15000 raccourcis claviers à retenir, pour arriver à faire le moindre truc, c'était un petit peu décourageant, z'ont bien fait de simplifier.

Et c'est un emacsien qui dit ça...

Duais mais quand t'apprends la commande pour faire copier-coller sur Emacs, tu t'en sers tout le temps après.

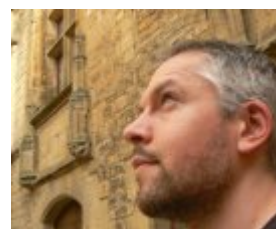
Alors que le raccourci pour appliquer un torseur cinétique sur le maillage de ton modèle 3D, faut être motivé pour s'en souvenir...

15/04/11  
gee

Crédit : [Simon Gee Giraudot](#) (Creative Commons By-Sa)

# J'ai signé la lettre pour l'open data à Toulouse

L'[open data](#) est dans tous ces états actuellement !



Il y a deux ans, nous avons publié un article intitulé [S'il te plaît... dessine-moi une ville libre](#). Il s'agissait en l'occurrence de Vancouver mais depuis de nombreuses [autres cités](#) se sont emparées de la question, et commencent à y apporter des réponses en impliquant leurs administrés.

Quitte à ce que ce soit les administrés eux-mêmes qui suggèrent fortement l'idée à leurs leaders politiques locaux, comme c'est le cas ici pour Toulouse, qui souhaite emboîter le pas à Rennes, Paris ou Nantes pour faire vivre ses données publiques.

## Open Data Toulouse – Entretien avec David Chemouil

**Bonjour pouvez-vous vous présenter succinctement ?**

J'ai 36 ans, je suis chercheur en informatique à Toulouse, où je vis avec ma compagne et ma fille. Très sensible à la question du logiciel libre (je suis tombé dedans il y a un peu plus de 15 ans pendant mes études), je suis aussi administrateur de l'April depuis cette année.

Mais dans le cas présent, je suis surtout point de contact d'un collectif s'intéressant à une question connexe, celle de [l'open data à Toulouse et sa région](#).

**Qu'est-ce que l'open data ?**



L'open data, c'est un mouvement international qui promeut la libération des données publiques non-nominatives.

Les données publiques, ce sont en gros les informations exploitées et conservées par les services publics (collectivités locales, État, mais on peut aussi penser à certaines sociétés privées travaillant directement pour les services publics). Ces données concernent des sujets très variés : cela peut aller d'informations sur la pollution dans une ville à la position des arbres entretenus par les jardiniers de la ville, ou encore le budget d'une ville ou d'une région et sur quels postes il est utilisé... Ou bien la position des bus en temps-réel...

Or, la plupart de ces informations se trouve de nos jours sous forme informatisée dans diverses bases de données. Vient alors l'idée, assez naturelle, que ces données pourraient la plupart du temps être rendues publiques (en tenant compte de la loi, bien sûr, d'où l'exclusion des données nominatives du champ de l'open data). Pour quoi faire ? Eh bien, pour des utilisations très variées.

Sur le plan socio-économique, cela permet d'imaginer des applications que les acteurs publics n'ont pas nécessairement les moyens de mettre en œuvre, ou alors dont personne n'a l'idée sauf une personne ingénieuse. Cela permet aussi de lever des barrières organisationnelles (par exemple, le fait que les informations relatives à des moyens de transport différents ne sont pas reliées, pas forcément à cause d'une mauvaise volonté mais plutôt pour des raisons structurelles). Parmi les applications déjà proposées dans des initiatives similaires, on a par exemple des sites web calculant des itinéraires adaptés aux personnes à mobilité réduite, empruntant des voies sur lesquelles il n'y a pas de gravier, les trottoirs sont tous équipés de passages *bateau*, etc. Ou alors des sites proposant des itinéraires empruntant à la fois la marche, le vélo de location, le bus et le train...

Sur un plan plus citoyen et politique, on peut utiliser les données pour évaluer les politiques publiques. De ce côté là, l'association [Regards Citoyens](#) propose plusieurs applications, notamment une intéressante sur le redécoupage de la carte électorale, qui permet d'en estimer son impact sur la représentation des partis politiques.

Outre ces raisons, on peut aussi penser qu'il n'est pas anormal que ces données, dont la collecte et la maintenance sont financées par les impôts, soient accessibles aux citoyens.

### **Pourquoi l'open data et les logiciels libres vont très bien ensemble ?**

J'y vois au moins deux raisons.

En premier lieu, on sent bien qu'il y a une parenté entre ces deux domaines, avec l'idée que des objets techniques porteurs de connaissances -les logiciels, les données- ont vocation à être librement accessibles, reproductibles, diffusables, modifiables, etc. On parle souvent à leur propos de bien non-rivaux, ce qui signifie que les donner à quelqu'un n'enlève pas de connaissance à celui qui a donné.

Toutefois, il y a aussi des différences entre logiciels et données. Pour cette raison, des licences spécifiques ont été proposées pour ces dernières. Celle que nous préconisons pour Toulouse est l'[Open Database License v1.0](#). Il n'y aucune originalité de notre part là dedans, c'est la licence préconisée par de nombreux tenants de l'open data. En gros, elle permet le partage des données, la création d'œuvres à partir des données, leur adaptation. Elle impose aussi des devoirs : l'origine des données ou œuvres basées sur les données doit être clairement tracée, la licence doit être préservée (clause de réciprocité analogue au copyleft des licences GNU) et l'ouverture doit être préservée (il s'agit là d'une clause qui permet de se protéger des menottes numériques

que sont les DRM).

Je disais qu'il y avait au moins deux raisons. Pour la seconde, il me semble qu'un certain nombre de logiciels faisant usage des données publiques ont vocation à être libres. Pour les raisons usuelles qu'invoquent les défenseurs du logiciel libre bien sûr, mais aussi parce que certaines applications à teneur politique doivent pouvoir être étudiées. Afin de vérifier, sinon l'objectivité des calculs, au moins étudier les choix faits par les concepteurs de l'application.

**Qui est open data Toulouse, pourquoi avoir créé ce collectif et que souhaitez-vous à court et à long terme ? quels sont les éventuels obstacles à lever ?**

Open Data Toulouse, c'est un collectif qui s'est créé un peu par hasard. Un collègue et moi-même nous intéressions au départ à la question du déplacement dans l'agglomération, en particulier pour favoriser les déplacements domicile-travail pas trop, voire pas du tout, polluants. En particulier, comment encourager les usagers à ne pas avoir envie de prendre leur voiture ? Il nous est vite apparu qu'il manquait à Toulouse des informations publiques relatives aux transports, mises à jour en temps-réel. Par exemple, si je dois attendre le bus 30 minutes sous la pluie, ce n'est pas la même chose que si je reçois un avertissement à mon bureau me disant que le bus sera là dans 5 minutes. De même, si le train a une panne, inutile que j'aille jusqu'à ma gare habituelle, mieux vaut que je prenne un bus spécifique. Mais il faut pouvoir en être averti.

De fil en aiguille, nous avons pris conscience que le problème était plus général et rejoignait la question de l'open data. Nous avons donc rencontré diverses personnes, associations et sociétés et mis au point une lettre commune, fondée sur celle élaborée pour la même raison à Nantes par l'association [LibertIC](#) (merci à cette dernière pour nous avoir permis cette réutilisation). Nous avons adressé cette lettre le 4 avril à

M. [Pierre Cohen](#), Président du [Grand Toulouse](#).

Par cette lettre, nous souhaitons surtout montrer à M. Cohen et son administration que cette question remporte l'adhésion de beaucoup de monde. Nous avons quelques contacts au Grand Toulouse et il semble que cette idée soit accueillie positivement. Évidemment, tout reste à faire, et il ne suffira pas, le cas échéant, de déclarer l'ouverture des données. La mise en œuvre de cette ouverture nécessite, encore une fois, une licence satisfaisante (comme l'[ODbL v1.0](#)) et implique aussi des efforts (mise à disposition des données sur un serveur, mise à jour régulière, etc.). La réunion de ces facteurs, c'est la liste des obstacles potentiels. Pour les surmonter, nous comptons sur des réunions de sensibilisation, mais il nous faut aussi, pour être pris au sérieux, de nombreuses signatures.

Pour finir, nous visons à court terme le lancement de l'open data dans le Grand Toulouse. Mais il s'agit seulement d'un début, bien sûr. Nous comptons à plus long terme sur l'effet d'entraînement du Grand Toulouse vis-à-vis des autres collectivités : les autres communes environnantes mais aussi le département, la région, d'autres services publics, etc.

**On parle également d'open data à [Rennes](#), [Nantes](#) ou [Paris](#). Vous pouvez-nous en dire plus et êtes-vous en relations avec les acteurs de ces villes pour fédérer les actions ?**

Nous avons été un peu en contact avec LiberTIC à Nantes qui a été moteur dans l'open data sur place. Et nos homologues parisiens nous ont contacté aujourd'hui ! De fait, nous sommes intéressés par tout retour d'expérience et éventuellement collaboration sur le sujet.

**Peut-on signer la lettre quand bien même nous ne sommes pas Toulousain ?**

Tout à fait, la question de l'open data concerne tout le monde. Ceci dit, nous espérons tout de même une forte

participation des habitants du Grand Toulouse ainsi que des environs (je songe en particulier à toutes les personnes qui travaillent dans l'agglomération toulousaine mais n'y habitent pas). C'est évidemment la première chose que regarderont M. Cohen et l'administration du Grand Toulouse.

J'ajoute, pour les personnes qui signeraient, qu'elles ne doivent pas hésiter à sensibiliser leur entourage [à signer aussi](#) !

---

## L'expérience Sugar Labs préfigure-t-elle une révolution éducative du XXI<sup>e</sup> siècle ?

Du projet [One Laptop per Child](#) (ou OLPC) les grands médias ont surtout retenu qu'il s'agissait de mettre un ordinateur entre les mains des enfants des pays défavorisés. Confondant la fin et les moyens ils sont alors souvent passés totalement à côté de son intérêt principal qui est pédagogique. [Negroponte](#) n'a de cesse à juste titre de le répéter : « le projet OLPC n'est pas un projet informatique, c'est un projet éducatif ».



Lorsqu'une écolière Uruguayenne et un écolier Uruguayen allument leur petit ordinateur vert, ils se retrouvent sur une interface qui est fort différente du classique environnement

graphique d'un Mac, Windows ou d'une distribution GNU/Linux.

Ici on abandonne la métaphore du bureau. Applications et fichiers sont bien entendu toujours présents mais ce qui est mis en avant c'est l'interaction avec les autres, ce qui apparaîtra de suite à l'écran c'est la présence du camarade, ce sur quoi il travaille, sachant qu'il est alors facile de le rejoindre pour collaborer.

Cette interface innovante et pleine de promesses s'appelle [Sugar](#) (cf [vidéo](#)). Elle est déjà massivement utilisée dans des pays comme l'Uruguay (cf [vidéo](#)) et nous voici alors projetés à des années-lumière de ce qu'une école française peut proposer non seulement comme outil mais aussi et surtout comme conception générale de sa fonction et de ses missions<sup>[1]</sup>.

En matière d'éducation et de nouvelles technologies, il y a ceux qui pensent qu'il est important de savoir comment mettre en gras dans Word, c'est-à-dire apprendre le mode d'emploi d'un logiciel propriétaire, et il y a ceux qui veulent en profiter pour... changer le monde !

Le créateur de Sugar, [Walter Bender](#), est de ceux-là. Simon Descarpentries l'a rencontré pour nous à Paris à l'occasion de l'[Open World Forum](#) 2010 et il a gentiment accepté de nous livrer un texte inédit nous présentant la jeune fondation Sugar Labs, sa philosophie, ses objectifs et ses réalisations.

Il ne s'agit que d'un témoignage mais c'est un témoignage important car il est bien possible que se trouve là l'une des pistes possibles et souhaitables pour l'éducation de demain. Et il n'est guère étonnant de constater la convergence entre une conception dynamique, créative et collective de l'apprentissage et le logiciel libre et sa culture.

**Culture communautaire : l'expérience**

# Sugar Labs

## Community culture: The experience of Sugar Labs

*Walter Bender – décembre 2010 – Licence Creative Commons By-Sa  
(Traduction Framalang : Siltaar, Goofy, Seb seb, Zitor, Julien  
et Barbidule)*

Dans un article publié il y a 30 ans et intitulé [« Critique de l'ordinateur contre pensée technocentrique »](#), [Seymour Papert](#) écrivait : « le contexte du développement de l'homme est toujours la culture, jamais une technologie isolée ». Dans un autre passage du même article, Papert offre un aperçu de ce qui est nécessaire pour fonder une culture de l'apprentissage : « Si vous vous demandez que doit savoir un pratiquant averti du [LOGO](#), la réponse va au-delà de la capacité à utiliser et enseigner le LOGO. L'adepte doit être capable de parler du LOGO, d'en faire la critique, et de discuter des critiques émises par d'autres personnes ».

## 30 ans après, remplaçons « LOGO » par « Sugar »

[Sugar](#) est une plateforme logicielle destinée à l'éducation des enfants. Sugar est développé et maintenu par Sugar Labs, une communauté mondiale de développeurs et d'éducateurs bénévoles. Notre objectif est l'émergence d'une génération de penseurs critiques et de gens capables d'inventer des solutions. À travers Sugar, nous nous efforçons de procurer à chaque enfant une chance d'apprendre et d'apprendre à apprendre, dans un contexte qui va lui permettre à la fois d'entamer un échange dynamique avec d'autres et de développer des moyens indépendants pour atteindre ses objectifs personnels.

Que devraient apprendre les enfants et comment devraient-ils apprendre ? Ceux qui apprennent devraient avoir accès aux idées qui nourrissent leur culture locale de même qu'aux idées puissantes qui constituent l'héritage global de l'humanité. Mais ils devraient aussi s'exercer à l'exploration et à la collaboration, et s'approprier des connaissances en menant une

démarche authentiquement ouverte de recherche de solutions. Ce qui peut être réalisé au sein d'une communauté éducative construite autour d'une structure de responsabilités, c'est-à-dire avec des apprenants qui s'impliquent dans un processus d'expression, de critique et de réflexion par eux-mêmes. Qu'est-ce que j'apprends ? Comment l'ai-je appris ? Pourquoi est-ce important ? Puis-je l'enseigner à d'autres ? Est-ce que j'en ai une connaissance approfondie en l'enseignant ?

Dans cet essai, je compte exposer la façon dont Sugar nourrit une culture éducative par l'association de deux communautés – les développeurs de Sugar et ceux qui apprennent – participant à créer un « contexte favorable au développement humain » et un changement de culture scolaire.

## **La culture du logiciel libre**

La culture du [logiciel libre](#) a influencé le développement de Sugar. Les développeurs du Libre vont au-delà du produit de consommation, ils créent et partagent leurs créations ; ils « débattent » du logiciel libre, ils en font la « critique », et ils « discutent le point de vue critique des autres ». Il ne prennent rien pour argent comptant. Les points communs entre le projet Sugar et le mouvement du logiciel libre sont les suivants : des outils pour s'exprimer, car les enfants créent des contenus autant qu'ils les consomment ; et la collaboration, car les enfants partagent leurs réalisations, s'aident mutuellement, et se lancent dans un processus de réflexion sur eux-mêmes et de critique collective.

Le projet Sugar s'inspire également de la façon dont les acteurs de la communauté du logiciel libre collaborent. Tout comme les développeurs de logiciels, les enfants discutent, se socialisent, jouent ensemble, partagent des médias, s'associent pour créer de nouveaux médias et des programmes, s'observent les uns les autres, dans un cadre à la fois formel et informel. Le projet Sugar facilite le partage, la collaboration et la critique. Les développeurs de logiciels



libres et ceux qui apprennent avec Sugar rédigent des documents, échangent des livres et des images, créent de la musique ou écrivent du code ensemble. Les deux communautés s'investissent dans une « pratique de réflexion » : il s'agit de mettre en pratique leur expérience tout en étant guidé et épaulé par des « spécialistes » d'un domaine (ils peuvent être professeurs, parents, membres de la communauté dans un salon de discussion, ou encore de camarades étudiants investis dans un échange critique soutenu).

De la même façon qu'avec le logiciel libre, Sugar encourage chaque enfant à être une force créative au sein de sa communauté. L'apprentissage avec Sugar n'est pas un acte passif où l'enfant reçoit le savoir. Il est actif. On parle de créativité, d'aisance, d'innovation, et de résolution de problèmes, tout ce qui implique l'expression personnelle et les liens forts à la communauté. Sugar apporte les outils d'expression à portée des enfants pour qu'ils soient libres d'agir à l'intérieur de leur communauté et à travers leurs actions, de changer le monde. Le logiciel libre est une condition nécessaire pour établir cette culture de l'expression et de l'émancipation. Le mot d'ordre de la génération suivante d'élèves sera « montre-moi le code, que je puisse en tirer un apprentissage et l'améliorer. »

## Réalisations et défis

Depuis que nous avons établi les Sugar Labs en tant que projet dans le cadre du [Software Freedom Conservancy](#) (*NdT : lit. Protection des Libertés Logicielles*) en 2008, nous avons démontré notre engagement à un ensemble de valeurs fondamentales qui comprennent la liberté et l'ouverture ; nous sommes devenus dans une large mesure indépendants de tout matériel et distribution (lorsque nous avons commencé, nous étions liés à une seule plateforme – le netbook [XO](#) du projet [One Laptop per Child \(OLPC\)](#)) ; nous avons énormément avancé sur le chemin qui conduit à une version logicielle stable 1.0 ; nous sommes fiers d'une vaste communauté qui comprend

près de 2 millions d'élèves utilisateurs ainsi que, bien entendu, des développeurs de logiciels et de nombreux professeurs et étudiants qui ont leur franc-parler.

Alors que nous nous débattons quotidiennement avec des défis techniques, notre défi principal est l'un des engagements avec notre communauté : comment pouvons-nous nous assurer qu'il y a un dialogue fructueux entre le développeur et les communautés éducatives liées à Sugar ? En d'autres termes, comment pouvons-nous transmettre à la communauté éducative la culture de la collaboration et de l'esprit critique qui est essentielle au développement de la plateforme Sugar, et à mieux nous permettre d'apprendre de nos utilisateurs finaux ? L'un des rôles que joue la communauté Sugar est de sensibiliser l'ensemble de l'écosystème du logiciel libre aux besoins des enseignants. Un autre rôle est de sensibiliser l'ensemble de l'écosystème éducatif au pouvoir de l'expression, de la critique et de l'auto-critique. Dans nos interactions avec les deux communautés, nous prenons grand soin de nous demander nous-mêmes : « Quel effet cela a-t-il sur l'apprentissage ? ».

Afin d'élargir nos efforts, un équilibre entre la fréquence des déploiements Sugar et la fréquence des nouveautés apportées par les Sugar Labs doit être maintenu. Nous avons un bon bilan dans notre réactivité aux besoins identifiés par les déploiements ; dans le même temps, nous sommes pro-actifs en sollicitant une plus grande participation de la communauté.

Les Sugar Labs sont aussi axés sur les besoins des enseignants. Nous avons des discussions régulières sur la façon de solliciter leurs retours. Certaines initiatives, tel qu'une [liste de discussions](#) fréquentée par des enseignants et des [conversations](#) hebdomadaires sur la pédagogie sont très productives. Un exemple de notre succès est que des enseignants commencent à apporter des modifications à Sugar et à ses activités. Un autre exemple est que des professeurs d'université enseignent l'informatique avec des logiciels

libres dont Sugar.

## **Sugar Labs se décline au pluriel**

Sugar Labs est une communauté globale qui se charge de définir des objectifs clairs et de maintenir l'infrastructure dont a besoin le projet dans son ensemble. Mais la communauté Sugar encourage et facilite également la création de « labs locaux » qui apportent leurs spécificités et une autonomie pour les déploiements régionaux, y compris en partenariat avec des entreprises locales à but lucratif, ce que le Sugar Labs « central » ne peut pas faire.

Ces labs locaux :

- adaptent la technologie et la pédagogie à la culture et aux ressources locales (ex : développement d'activités et de contenus spécifiques à une région) ;
- aident à traduire Sugar en langues régionales ;
- gèrent les déploiements Sugar dans les écoles de la région ;
- créent des communautés locales adhérentes aux principes des Sugar Labs, rendant Sugar plus ouvert et autonome ;
- permettent la communication entre ces communautés locales et la communauté mondiale Sugar Labs ;
- hébergent, co-hébergent ou s'associent dans l'organisation de conférences, ateliers, discussions et rencontres relatifs à l'utilisation et au développement de Sugar.

Avec le temps, la charge technique se répartit sur les labs locaux (la sortie récente de [« Dextrose »](#), pour les OLPC X0 construits au Paraguay, est un exemple de comment les labs locaux – menés par une communauté de volontaires – peuvent travailler ensemble pour résoudre des défis techniques et pédagogiques).

## En « amont » et en « aval »

Marco Presenti Gritti, développeur Sugar et co-fondateur des Sugar Labs, me rappelait que lorsque nous avons créé les Sugar Labs, nous avons pris une décision réfléchie sur l'étendue du développement. « En suivant le modèle de l'environnement graphique [GNOME](#), nous n'allions pas tout créer et gérer nous-même, mais nous allions nous intégrer et nous appuyer sur les distributions GNU/Linux et le projet OLPC pour le faire ».

Classiquement, un projet en amont<sup>[2]</sup> développe du code et un processus de publication. En aval, les distributions créent des paquets avec des personnalisations et distribuent un produit pour l'utilisateur final (cela implique habituellement un [processus QA](#) bien défini et un mécanisme de support).

Le spécificité éducative de notre projet a nécessité d'élargir le modèle et les communautés impliquées. Le développement et les déploiements de Sugar sont évidemment engagés dans la construction d'images, de QA, des tests, dans la recherche d'erreurs à corriger, dans la documentation, le support... qui relèvent de programmeurs experts. Mais, comme mentionné précédemment, nous travaillons également avec des étudiants et lycéens et à l'occasion un professeur qui connaît suffisamment bien le Python peut contribuer aux correctifs.

Afin de créer un produit viable et gérable, nous devons établir un équilibre entre notre travail comme projet logiciel « en amont » et les efforts « en aval » des distributeurs GNU/Linux. C'est ainsi que nous travaillons activement avec la communauté [Fedora](#) (laquelle a pris à son compte une grosse partie de la charge associée au support du matériel OLPC), la communauté [Debian](#), [openSUSE](#), [Trisquel](#), [Mandriva](#), [Ubuntu](#) (ex : le Sugar Ubuntu remixé), etc.. À l'occasion nous devons assumer un rôle de leader, comme quand nous avons pris à bras-le-corps les initiatives naissantes pour créer un [Live USB](#) – « [Sugar on a Stick](#) ».

## Optimisé pour la communauté

À la conférence [LIBREPLANET](#) en 2010, Eben Moglen a accordé un entretien sur tout ce qui avait été accompli par la communauté du logiciel libre. Le logiciel libre n'est plus une possibilité ; il est « indispensable », a-t-il affirmé. Ce logiciel « fiable et qui a un coût de production quasi nul » présente de nouvelles et nombreuses opportunités, en particulier dans le secteur de l'éducation, qui est toujours grevé par un budget serré. Seul le logiciel libre est « écrit une fois mais exécuté partout ».

Nous voulons aussi écrire du code fiable qui permette à Sugar d'être exécuté « partout », et nous avons réalisé de grands progrès en suivant les pas de la grande communauté GNU/Linux. Mais la communauté Sugar a un objectif supplémentaire : nous souhaitons que nos utilisateurs finaux participent également à l'amélioration du code, parce que cela participe de l'apprentissage. Si tout le monde est capable d'écrire du code et si ce code est écrit avec les modifications des utilisateurs finaux en tête, nous aurons un monde dans lequel chacun est engagé dans le « débogage », ce que Cynthia Solomon a décrit une fois comme [« l'une des grandes opportunités éducatives du XXIe siècle »](#).

Oui la licence [GPL](#) (General Public License) utilisée par les Sugar Labs garantit que le logiciel peut être modifié par l'utilisateur final. Mais, pour la plupart des utilisateurs, ceci n'est qu'une liberté théorique si la complexité du logiciel représente une barrière insurmontable. Par conséquent, les critères habituels (fiabilité, efficacité, maintenance, etc.) sont nécessaires mais non suffisants pour l'éducation.

Aux Sugar Labs, nous faisons un pas supplémentaire en nous assurant que notre code est à la fois libre et ouvert, mais également « ouvert à la manipulation des utilisateurs finaux ».

Voici quelques actions entreprises par Sugar Labs pour encourager et faciliter les modifications des utilisateurs finaux :

- Susciter des attentes et des envies en établissant une culture dans laquelle c'est la norme d'utiliser les libertés permises par le logiciel libre et articuler la liberté pour modifier les aspects du logiciel libre (1ère liberté).
- Offrir des outils qui facilitent l'accès aux sources (ex : un menu « voir les sources » toujours disponible, rendant la source de chaque application à portée d'un « clic de souris »).
- Utiliser des langages de script (Python, Javascript, et SmallTalk dans le cas de Sugar) pour que ces changements puissent être immédiats et faits directement.
- Mettre en place des paliers pour permettre à l'utilisateur final de commencer en faisant des petits pas (alors que le langage de programmation C peut avoir une « couche haute », il n'a pas de très « basse couche »).
- Réduire le risque associé aux erreurs en proposant des « zones tampons » ; si en touchant au code vous introduisez des bugs collatéraux ou irréversibles alors les gens seront vite conditionnés à ne pas se livrer à des comportements à « risque » en modifiant le code.
- Fournir de « vrais » outils : s'assurez-vous que la vraie version puisse être modifiée et non une version répliquée indépendante mais peu motivante.
- Être une communauté de soutien ; on peut dire à juste titre de la communauté Sugar qu'elle est accueillante et tolérante avec les « nouveaux venus », poser une question c'est déjà devenir membre de la communauté, nous sommes pointilleux pour ce qui concerne l'octroi de privilèges sur le « projet principal » mais nous donnons les droits pour encourager la création de branches expérimentales.

Quand on m'a demandé combien de correctifs ont été fournis par les utilisateurs de Sugar, j'ai répondu que des membres de la communauté ont contribué aux correctifs mais que je n'avais pas connaissance de correctifs apportés par des enfants. Encore faut-il faire la distinction entre correctifs envoyés et acceptés, car l'apprentissage commence en créant le correctif, en le soumettant, et en le partageant avec d'autres même lorsqu'il ne se retrouve pas accepté. Sugar a inculqué aux enfants et à leurs professeurs le sentiment qu'ils peuvent être créatifs et utiles avec l'informatique.

Cependant, après deux années d'expérience concrète de Sugar, nous commençons à voir des contributeurs émerger de sa communauté d'utilisateurs. Par exemple, en Uruguay, qui a été le premier pays à fournir des outils éducatifs libres à chaque enfant, quelques préadolescents sont en train de coder activement (un enfant de 12 ans d'une petite ville à des heures de Montevideo fréquente notre canal IRC, y pose des questions et poste du code, à la mi-décembre 2010, il a déjà envoyé huit activités sur notre portail). Quand le président uruguayen José Mujica a entendu parler de ces réalisations, il a souri et a dit avec une voix remplie de fierté : « Nous avons des hackers ». Il y a peut-être 12 enfants qui développent du logiciel libre aujourd'hui en Uruguay. L'an prochain ils seront 100. Dans 2 ans, ils seront 1000. L'Uruguay est en train d'expérimenter un changement de culture lié à un changement dans les attentes que le pays a pour ses enfants, un changement accéléré par la culture du logiciel libre.

## **Maximiser nos efforts**

Qu'est-ce qui motive nos contributeurs et qu'est-ce qui motive les professeurs (que nous aimerions voir adopter Sugar) ?

Pour tenter d'y répondre je me suis appuyé sur l'article [L'économie comportementale : les sept principes des décideurs](#) publié par le *New Economics Foundation* :

- **Le comportement des autres personnes compte.** Nous devons sensibiliser les professeurs aux meilleures pratiques de Sugar pour qu'ils puissent faire des émules. Pouvons-nous identifier les « génies », « contacts », « commerciaux » dans nos communautés cibles ? Quelles ressources pouvons-nous mettre en place pour les inciter à adopter Sugar ? Ainsi je travaille avec une petite école de quartier dans la ville de Boston dont l'exemple est suivi par d'autres quartiers bien plus importants. Si nous pouvons avoir une influence sur un professeur « génie » du quartier, nous pourrions avoir un gros avantage. Cela signifie également que nous devons être vigilants quant à la qualité pédagogiques de nos activités proposées.
- **Les habitudes sont importantes.** Ces habitudes qui participent au status quo ne doivent pas être négligées. Qu'est-ce qui motive et encourage le changement ? Quelles actions pouvons-nous mener pour soutenir et engager les changements dans les pratiques et les comportements ?
- **Les gens sont motivés pour « faire ce qu'il faut ».** Mettons alors cette notion de « faire ce qu'il faut » (*NdT : do the right thing*) en débat avec les enseignants, essayons de voir avec eux si leurs conceptions peuvent évoluer. En géométrie, il n'y a pas de chemin réservé aux rois, disait Euclide.
- **Les attentes des gens influencent leur comportement : ils veulent que leurs actions soient en phase avec leurs valeurs et leurs engagements.** C'est un travail de longue haleine pour nous car nous ne sommes pas toujours en phase au départ avec ces attentes. Cependant, tant que nous respectons et sommes fidèles à nos valeurs, nous pouvons convaincre et avoir de l'influence.
- **Les gens sont réticents au changement de peur de perdre**



**ce qu'ils possèdent.** Utiliser Sugar à partir d'un clé USB (« Sugar on a Stick », qui emprunte seulement un ordinateur sans rien modifier dedans) n'implique aucune changement irréversible tout en permettant de faire une nouvelle expérience pédagogique.

- **Les gens hésitent souvent lorsqu'il s'agit de prendre de grandes décisions.** Ils sont souvent intimidés par les perspectives d'apprentissage de nouvelles choses (jusqu'à vraiment les faire). De plus es pertes immédiates peuvent décourager et faire perdre de vue les récompenses à long terme. Nous devons accorder une grande importance à ce moment crucial du démarrage en accompagnant ceux qui acceptent de prendre un tel risque.
- **Les gens ont besoin de se sentir écoutés et impliqués pour s'engager dans le changement.** Nous avons une communauté qui tente d'accorder le plus grand soin à l'accueil des participants et à l'examen de leurs contributions. Ceci est une de nos grandes forces.

## **Est-ce que cela fonctionne ?**

L'évaluation de projets éducatifs a toujours été difficile, en partie parce qu'il est difficile d'arriver à un concensus sur les mesures d'évaluation.

Il semble plus facile de prendre le problème par la négative où le consensus sur ce qu'il ne faut pas faire est plus facile à trouver. Ainsi [Michael Trucano](#), qui blogue sur le portail éducation de la Banque mondiale, a publié un « top 10 » des pires pratiques de l'utilisation des nouvelles technologies dans l'éducation. Liste que je prends ici comme référence négative pour le projet Sugar avec comme exemples probants et prometteurs les deux déploiements d'envergure que sont le [Paraguay Educa](#) et le [Plan Ceibal](#) en Uruguay.

## **1. Parachuter du matériel dans les écoles et espérer qu'un miracle se produise.**

C'est une critique souvent entendue pour le projet One Laptop per Child (un ordinateur portable par enfant), mais dans le faits, il y avait d'importants mécanismes d'aide et de mise en place en Uruguay et au Paraguay avant même que le matériel ne soit livré. En Uruguay, en plus du vaste support proposé directement par le gouvernement (incluant un programme de formation des professeurs, un centre d'appel, une vidéothèque des bonnes pratiques, etc.), deux initiatives communautaires au niveau national ont vu le jour : [Ceibal Jam](#), qui fournit des logiciels et du contenu local aux enfants d'Uruguay, et Red de Apoyo al Plan Ceibal ([RAP-Ceibal](#)), qui assure un réseau d'aide pour les professeurs. Paraguay Educa a une équipe de conseillers qui travaille à temps plein dans les écoles, en aidant les professeurs. Et les éducateurs des deux pays participent régulièrement à des forums mondiaux.

## **2. Concevoir via l'OCDE des environnements d'apprentissage à implémenter partout.**

Les « pays développés » proposent du contenu et quelques règles de bonnes pratiques, mais ce sont avant tout les équipes pédagogiques locales en Uruguay et au Paraguay qui échangent et conçoivent leurs propres matériels et programmes pour répondre à leurs besoins locaux (par exemple, un professeur de la campagne péruvienne a écrit un [livre](#) sur l'utilisation de Sugar en salle de classe qui est internationalement lu et reconnu par les autres professeurs).

## **3. Penser les contenus éducatifs après la mise en place du matériel.**

En Uruguay et au Paraguay, c'est la pédagogie qui a guidé la vitesse de déploiement d'un projet vu avant tout comme une plateforme d'apprentissage (incluant les ordinateurs portables, la connectivité, les serveurs, la formation, la documentation, le support, l'assistance de la communauté,

etc.).

#### **4. Supposer que vous pouvez uniquement importer du contenu venu d'ailleurs.**

Le mot clé ici est « uniquement ». L'Uruguay et le Paraguay profitent bien entendu des contenus créés ailleurs (comme par exemple ceux de la communauté [Etoys](#)) mais ils n'oublient de favoriser la production de ressources locales, qu'il s'agisse de nouveaux contenus ou de contenus modifiés à partir de ceux récupérés ailleurs.

#### **5. Ne pas surveiller, ne pas évaluer.**

À Plan Ceibal, ils ont un fonctionnement étendu pour surveiller l'état du réseau, des serveurs, et des ordinateurs portables lors du déploiement. Il y a beaucoup d'évaluations en cours du programme, aussi bien internes qu'externes. Paraguay Educa a été l'objet d'une évaluation externe par la Banque Interaméricaine de Développement (IDB Inter-American Development Bank).

#### **6. Faire un gros pari sur une technologie qui n'a pas fait ses preuves.**

C'est en particulier le cas lorsque l'on se base sur un unique distributeur et sur des standards fermés et/ou propriétaires. C'est alors une épée de Damoclès qui pèse sur l'avenir du projet. Les deux programmes mentionnés ci-dessous ont fait l'objet d'appels d'offre public et ont plusieurs distributeurs. Les deux utilisent abondamment des logiciels libres.

#### **7. Ne pas être transparent sur le coût global de l'opération.**

L'Uruguay a été assidue en publiant les chiffres de leur coût total de possession, maintenance et services du projet (chiffres, basés sur les coûts mesurés sur le terrain, qui se sont avérés plus bas que ce que certains avis pessimistes

avaient prévu).

## **8. Négliger les problèmes d'équité.**

En Uruguay ce sont avant tout les familles modestes qui sont ainsi équipées en informatique avec un accès Internet gratuit.

## **9. Ne pas former vos professeurs (ni votre directeur d'école).**

Le plus gros investissement dans le programme au Paraguay a été la formation des professeurs. C'est sûrement la principale clé de la réussite du projet et nous veillons à ce que cette formation soit toujours plus efficace et adaptée aux réalités du terrain.

Trucano laisse le point numéro 10 comme exercice ouvert pour le lecteur. J'ajouterais :

## **10. Ne pas impliquer la communauté.**

Dans les deux communautés uruguayenne et paraguayenne l'implication fait partie du projet par nature. Pour ce qui concerne Sugar, c'est un effort d'une communauté globale qui implique des centaines d'ingénieurs et des milliers de professeurs. Un résultat remarquable est le degré d'implication des parents dans les programmes.

## **Regarder vers le futur**

Comme il est de mise avec chaque projet piloté par une communauté, il y a un débat permanent sur la vision de Sugar. Il peut y avoir des divergences d'opinion sur l'étendue de la mission des Sugar Labs (allant d'un point d'attention particulier sur les outils de collaboration à une vision plus large sur tout ce qui est nécessaire pour des déploiements réussis de l'OLPC). Mais tout le monde s'accord à dire qu'il y a une communauté Sugar de développeurs et d'apprenants pleine de vie et d'énergie et que les plateformes d'apprentissage basées sur des logiciels libres encouragent l'appropriation du savoir quel que soit le domaine que l'apprenant explore :

musique, navigation sur internet, lecture, écriture, programmation, dessins, etc.

[Carla Gomez Monroy](#), une pédagogue qui a participé à de nos nombreux déploiements, décrit Sugar comme « un environnement émergent et collaboratif, où la communauté identifie, code, utilise, innove, conçoit et re-conçoit ses propres outils » Les membres de la communauté d'apprentissage de Sugar s'engagent dans le *débogage* de leur créativité et des outils mis en place pour exprimer cette créativité. Ils investissent Sugar en tant que technologie mais aussi et surtout comme une culture de l'apprentissage passant par l'expression et la critique collective.

L'expérience Sugar Labs est « une participation collaborative pour *apprendre à apprendre* avec des outils qui nous correspondent ».

*Walter Bender est le fondateur et le directeur exécutif de Sugar Labs, une fondation à but non lucratif. En 2006, Bender a co-fondé « One Laptop per Child », une organisation à but non lucratif avec Nicholas Negroponte et Seymour Papert.*

## Notes

[1] Crédit photo : [Danishkanavin](#) (Creative Commons By-Sa)

[2] Dans le développement logiciel, la métaphore de la rivière est utilisée pour décrire où les différentes activités et responsabilités se situent dans l'écosystème. L'« Amont » fait référence aux auteurs et mainteneurs du logiciel. L'« Aval » fait référence aux distributeurs et aux utilisateurs du logiciel.

---

# Un manuel sur Thunderbird pondu collégialement en 48h chrono

Il est finalement assez rare de voir des acteurs du logiciel libre préserver la règle théâtrale classique de l'unité de temps, de lieu et d'action.



Ce qui l'est moins, c'est de voir différentes structures ([Mozilla Europe](#), [FrenchMozilla](#), [FLOSS Manuals Francophone](#) et Framasoft) mettre leur force en commun pour un noble objectif, en l'occurrence proposer aux utilisateurs francophones de Thunderbird un [tutoriel](#) de qualité.

Vous trouverez ci-dessous la reproduction de la page « À propos » que j'ai eu l'honneur de rédiger ainsi que le billet blog de Goofy qui annonce l'évènement.

## À propos de ce manuel

Qu'est-ce qui peut bien pousser un Tourangeau, un Niçois, une Francilienne, d'autres Franciliens et même un Romain à se retrouver l'espace d'un week-end à Paris pour y travailler bénévolement sur leur temps libre ?

Un BookSprint bien sûr !

Votre mission si vous l'acceptez : traduire du début à la fin un livre de plus de cent pages en moins de trois jours ! Voilà

une tâche qui s'annonçait si ce n'est prométhéenne tout du moins rébarbative.

Sauf s'il s'agit d'un livre sur Thunderbird et qu'en ce temps où nos données personnelles se promènent toujours plus nombreuses sur le Web, il n'est pas inutile de rappeler les intérêts et avantages à utiliser sur son ordinateur cet excellent client de messagerie.

Sauf si l'on s'y met tout ensemble dans la joie et dans la bonne humeur pour apporter nous aussi notre modeste pierre francophone au logiciel libre en général et à ce logiciel libre en particulier.

Un objectif motivant, la perspective d'un convivial travail collaboratif in the real life et la confiance de placer le fruit de nos efforts sous licence libre, il n'en fallait pas plus pour trouver une bonne dizaine de volontaires prêts à relever le défi.

Un défi concrétisé le 19 et 20 mars 2011 à Paris dans les bureaux de Mozilla Europe avec le soutien de FrenchMozilla, Framasoft et FLOSS Manuals Francophone.

Quant à la version originale, vous ne serez pas surpris d'apprendre qu'elle avait été conçue cinq mois plus tôt lors d'un... BookSprint à Toronto !

Un BookSprint en suit un autre, et à en juger par les mines fatiguées mais réjouies des participants, le mouvement n'est pas prêt de s'arrêter...



## Votre courrier avec Thunderbird – un manuel en français

[URL d'origine du document](#)

*Goofy – 9 avril 2011 – FrenchMozilla*

### Le fruit d'une traduction collaborative

La version originale en anglais de ce manuel pour Thunderbird a été réalisée pendant un booksprint à Toronto au Canada à l'occasion d'une semaine de l'open source en novembre 2010 : pendant deux jours, une vingtaine de participants y ont contribué, à l'instigation de Mozilla Messaging et de FLOSS Manuals.

Pour vous offrir la version française, nous avons aussi joint nos forces. Mozilla Europe a pris en charge le financement et l'intendance de l'opération et nous a accueillis généreusement dans ses locaux parisiens (merci Tristan et Pascal !), **Floss manuels fr** a fourni la plateforme de traduction et sa compétence attentive (merci Élisabeth), les traducteurs du milieu du Libre se sont associés : ceux de **frenchmozilla**, ceux de **frama-lang** venus en force et d'autres encore que l'aventure tentait pour cette traduction « ouverte ».

Mieux que les crédits détaillés qui figurent dans les dernières pages, c'est un grand merci qu'il faut donner à tous ceux



qui ont consacré bénévolement du temps libre et de l'énergie à cette opération. Le week-end (des 19-20 mars 2011) a été à la fois fiévreux et détendu, c'était aussi une occasion de se rencontrer, de manger, boire et s'amuser ensemble... Et les textes du manuel ont été intégralement traduits et relus une première fois en 48 heures. Vous le verrez, le manuel est aussi riche en illustrations : les nombreuses captures d'écran ont demandé un peu plus de temps, car nous tenions à ce qu'elles montrent l'interface française.

## **Un manuel destiné à tous**

Nous pensons que ce manuel peut être utile en particulier à ceux qui découvrent Thunderbird ou hésitent encore à l'utiliser. Il les aidera à installer, faire les premiers pas et découvrir les possibilités de l'application.

Il peut être également une référence pratique pour tous les bénévoles des forums de Geckozone qui assurent vaillamment l'aide et le support de Thunderbird et qui doivent souvent répéter les mêmes réponses. Un lien vers tel ou tel chapitre peut éviter de trop longues explications.

Ceux qui utilisent déjà Thunderbird et qui ont besoin d'aller plus loin ont l'habitude de consulter la base de connaissances de Thunderbird (merci Vincent qui la maintient avec brio !). mais ils pourront découvrir ici, outre quelques trucs et astuces, les fonctionnalités d'une extension toute récente : il s'agit de « Conversations », dont Jonathan Protzenko, son développeur, a fourni le chapitre-tutoriel qui n'existait pas encore dans la version anglaise. C'est un des « plus » non négligeables de ce manuel !

## **Pas parfait mais parfaitement utilisable et évolutif**

Il est libre et open source. Conformément à l'engagement des co-rédacteurs participant à la plateforme Flossmanuals, ce manuel sous copyright est sous licence GNU GPL : il peut notam-

ment être lu et copié librement.

Plus encore, vous pouvez y contribuer ! Il vous suffit de vous enregistrer sur la plateforme de traduction et vous pourrez relire/réviser/corriger (la chasse aux coquilles est ouverte) mais aussi bien sûr ajouter des phrases explicatives voire des chapitres entiers, compléter avec de nouvelles captures d'écran etc.

Nous serons attentifs aux développements de Thunderbird et nous ferons en sorte de tenir ce guide à jour lorsque de nouvelles fonctionnalités de l'application apparaîtront.

Le manuel est à vous !

## Liens utiles

- [Le manuel Thunderbird, disponible en HTML et en PDF](#)
- [Thunderbird en français sur la page officielle de téléchargement](#)
- [Le forum d'aide pour Thunderbird sur Geckozone](#)
- [La base de connaissances pour Thunderbird](#)
- [FlossManuals France](#) Manuels libres pour logiciels libres
- [Framalang](#) et ses traducteurs fous
- [Les photos de la tradAction par Elisa](#)

---

**6 questions à Karl Fogel,  
auteur de Produire du  
logiciel libre**

À l'occasion de la sortie du framabook [Produire du logiciel libre](#) (dont notre secret espoir est qu'il suscite des vocations chez les jeunes et les moins jeunes), nous avons posé quelques questions à son auteur [Karl Fogel](#).



Est-ce que la situation a évolué depuis la première version du livre, en particulier avec les nouvelles forges comme [GitHub](#) (qui repose entre autres la question du [fork](#)) ? Est-ce un problème d'héberger des logiciels libres sur des plateformes propriétaires ? Est-ce que l'informatique devrait être enseignée en tant que telle aujourd'hui à l'école ?

Autant de questions auxquelles il apporte de très intéressantes réponses.

## Entretien avec Karl Fogel

[L'interview en version originale anglaise sur le blog de Karl \(intéressants commentaires inside\)](#)

*(Traduction Framalang : Don Rico pour les questions et Olivier Rosseler pour les réponses)*

**La version française de POSS vient tout juste d'être publié et votre livre a été traduit, ou est en cours de traduction, dans d'autres langues. Que pensez-vous de ces adaptations de votre œuvre, rendues possibles par le choix de le placer sous licence libre ?**

Je suis absolument ravi. Je n'y vois vraiment aucun inconvénient. Les traductions permettent une diffusion plus large du livre, et c'est exactement ce que je souhaite.

Je suis extrêmement reconnaissant envers les traducteurs.

**Si vous deviez écrire une deuxième version de POSS aujourd'hui, qu'est-ce que vous changeriez ou ajouteriez ? Et d'ailleurs, est-ce qu'une deuxième version est prévue ?**

Et bien, en fait, j'y apporte toujours des petites modifications, à mesure que les pratiques de l'open source évoluent. La version en ligne change constamment. On pourra peut-être la nommer officiellement « Version 2.0 » à un moment donné, mais au fond, c'est vraiment un processus continu.

Par exemple, il y a cinq ou six ans, presque tous les projets avaient leur propre infrastructure de développement. Chacun avait son serveur, son système de contrôle de versions, son système de suivi de bogues, un responsable de la liste de diffusion, un wiki peut-être, c'étaient les outils de développement.

Mais depuis, on a assisté à des regroupements. De nos jours, seuls les très gros et les très petits projets possèdent leur propre infrastructure. La majorité des projets choisissent des sites pré-conçus, comme [GitHub](#), [Google Code Hosting](#), [SourceForce](#), [Launchpad](#), etc. La plupart des développeurs open source se sont familiarisés avec ces environnements.

Et par conséquent, j'ai mis à jour la partie du livre traitant des infrastructures d'hébergement, pour enrichir la section « Les sites Web » et parler des sites comme ceux mentionnés ci-dessus, plutôt que de ré-inventer la roue à chaque projet. Les gens se rendent bien compte qu'administrer son propre hébergement requiert énormément de ressources, malgré les avantages que l'on peut en tirer, et que donc, externaliser cette tâche est devenu presque une obligation si on veut avoir un peu de temps pour effectivement travailler sur le projet.

J'ai également mis le livre à jour pour parler des nouvelles versions des licences open source (comme la [GNU General Public](#)

[License 3](#), qui est sortie après que le livre ait été publié), et j'ai également revu mes recommandations vis à vis de certains logiciels, car les temps changent. Par exemple, [Git](#) est de bien meilleure qualité aujourd'hui qu'à l'époque où j'ai rédigé la toute première édition.

**La manière de produire des logiciels libres n'a pas tellement changée en cinq ans. Mais de nouvelles forges sont apparues, sur un modèle un peu différent de SourceForge. Je pense à Google Code mais surtout à GitHub. GitHub serait un peu le « Facebook des forges open source », avec ses fonctions de réseau social, son édition à même le navigateur... Son slogan est « Fork me on GitHub ». La notion de fork semble ne plus être tout à fait la même qu'avant. Que pensez-vous de tout cela ?**

En fait, je pense que la notion de fork n'a pas changé. La terminologie, peut-être, mais pas le concept.

Si je me penche sur les dynamiques des rouages des projets open source, je ne vois pas de différences fondamentales selon que le projet utilise une forge ou l'autre. GitHub propose un produit fantastique, mais ils ont aussi un marketing fantastique. Ils encouragent les projets à inviter leurs utilisateurs à « créer une fork sur GitHub », c'est à dire « créer une copie pour jouer un peu avec ».

Et même si en un sens la copie d'un projet hébergé sur Git peut techniquement s'appeler un « fork », en pratique ça n'en est pas un. Le concept de fork est avant tout politique, pas technique.

À l'origine, initier un fork signifiait élever la voix pour dire : « nous pensons que le projet ne prend pas la bonne direction, nous avons pris la décision d'en faire une copie pour le poursuivre dans la bonne direction, que tout ceux qui partagent ce point de vue se joignent à nous ». Et les deux projets se retrouvaient alors publiquement en concurrence, à

l'attention des développeurs et des utilisateurs, parfois aussi pour des questions d'argent. Parfois l'un des deux l'emporte, parfois ils fusionnent pour ne former à nouveau qu'un seul projet. Mais quelle qu'en soit l'issue, c'est avant tout un processus politique : susciter des adhésions pour continuer ensemble le projet.

Cette dynamique est toujours d'actualité, elle se poursuit tous les jours. Qu'on parle de « fork » pour designer quelque chose de différent, pourquoi pas, mais ça ne change pas la réalité, on utilise juste un terme différent pour décrire la réalité.

GitHub a commencé à parler de « fork » pour dire « créer une copie à bidouiller ». Maintenant, c'est vrai qu'avec ce genre de copie il est facile de s'éloigner du projet originel pour re-fusionner plus tard, c'est l'une des caractéristiques de Git et de tous les [systèmes de contrôle de version](#) décentralisés. Et c'est vrai que s'éloigner pour re-fusionner est plus compliqué avec les systèmes de contrôle de version centralisés comme [Subversion](#) et [CVS](#). Mais tous ces « forks » créés sur Git ne sont pas des forks au sens premier du terme. En général, lorsqu'un développeur se fait une copie sur Git et la modifie, c'est en espérant que ses changements seront fusionnés dans la copie « maîtresse ». Et quand je dis « maîtresse », ce n'est pas au sens technique, mais bien au sens politique : la copie maîtresse est celle que la plupart des utilisateurs suivent.

Je trouve que ces fonctionnalités de Git et de GitHub sont géniales, et j'aime bien les utiliser, mais il n'y a rien de révolutionnaire ici. Il y a peut-être une évolution de la terminologie, mais la vraie dynamique des projets open source ne varie pas : les développeurs fournissent de gros efforts pour que leurs modifications soient intégrées à la distribution principale, car ils ne veulent pas s'embarrasser avec une copie privée qu'ils auraient à entretenir. Git réduit la pénibilité liée à la maintenance de

modifications indépendantes, mais pas encore suffisamment pour que cet effort soit négligeable. Les développeurs intelligents forment des communautés et tentent de conserver un code de base unifié, car c'est la meilleure chose à faire. Ça n'est pas près de changer.

**En juin 2010, Benjamin Mako Hill remarque dans son article [Free Software Needs Free Tools](#) (traduit [ici](#) sur le Framablog) qu'héberger un projet libre sur une plateforme propriétaire pose problème. À votre avis, quelle est l'importance de ce problème ?**

Et bien, je connais Mako Hill, je l'apprécie et j'éprouve beaucoup de respect pour lui. Mais je dois dire que je ne partage pas son avis sur ce point, et ce, pour plusieurs raisons.

D'abord, il faut être réaliste. On ne peut pas être un développeur logiciel sans outils propriétaires de nos jours. Réduire arbitrairement la notion de « plateforme » n'est qu'un artifice pour croire qu'on travaille dans un milieu entièrement libre. Par exemple, je peux héberger mon projet chez Launchpad, qui est un logiciel libre, mais est-ce que je peux vraiment écrire du code sans utiliser le moteur de recherche de Google, qui n'est pas libre ? Bien sur que non. Tous les bons programmeurs utilisent en continu Google, ou un autre moteur de recherche propriétaire. Il faut inclure ces recherches Google dans la « plateforme », impossible de se voiler la face.

Mais on peut pousser la réflexion plus loin :

Qu'attendez-vous de l'hébergeur de votre projet, quelles sont les libertés importantes ? Vous utilisez une plateforme et vous demandez aux autres de l'utiliser aussi pour collaborer avec vous, donc, idéalement, la plateforme devrait être libre.

Ainsi, si vous souhaitez y apporter des modifications, vous pouvez : si quelqu'un veut créer un fork de votre projet (au

sens ancien, politique, du terme), ils peuvent reproduire l'infrastructure d'hébergement ailleurs, où ils la contrôleront, si nécessaire. Alors, en théorie tout cela est très bien et très joli, mais honnêtement, même si le code source de Google Code, par exemple, était libre, vous ne pourriez pas reproduire Google Code Hosting. Il vous manquerait encore le personnel, le service, les [data center](#) de Google... toute l'infrastructure qui n'a rien à voir avec le code source. Ça n'est pas réalistiquement faisable.

Vous pouvez *forker* le projet, mais en général vous ne pouvez pas reproduire son hébergement, cela demande trop de ressources. Et puisque ça n'est pas votre propre service, vous ne pouvez pas l'adapter à votre convenance ; ce sont les gens qui font tourner les serveurs matériels qui décident de quels ajustements sont acceptables ou pas. Donc dans la pratique, vous ne disposez pas de ces libertés.

(Certains services d'hébergement tentent d'octroyer autant de libertés que possible à leurs utilisateurs. Par exemple, le code de Launchpad est open source, et ils intègrent les correctifs de leurs membres. Mais l'entreprise qui héberge Launchpad doit quand même approuver chaque modification puisque ce sont eux qui font tourner les serveurs. Je crois que SourceForge veut tenter la même expérience, si l'on en croit l'[annonce](#) faite récemment à propos d'Allura.)

Alors, en fonction de tout cela, quelles sont les libertés possibles ?

Il vous reste la liberté de faire entrer et sortir vos données. En d'autres termes, le noeud du problème se situe au niveau de la possibilité qu'on les interface de programmations ([API](#) pour Application Programming Interfaces) de déplacer les données d'un service à l'autre, de manière fiable et automatique. Si je peux écrire un programme qui peut récupérer toutes les données de mon projet



depuis une forge pour les transférer à une autre, c'est une liberté utile. Je ne suis pas pieds et poings liés. Ça n'est pas la seule liberté qui compte, on est même loin d'une liberté idéale. Mais c'est une liberté utile dont on dispose dans un monde où utiliser ses propres serveurs est devenu inabordable.

Ce n'est pas que cette conclusion m'enchante. Mais les choses sont ainsi. La période de « chasseur/cueilleur » dans l'open source est terminée, nous sommes entrés dans l'ère agricole et urbaine. Vous ne pouvez plus creuser vos propres sillons d'irrigation ou votre propre système d'évacuation des eaux usées. C'est trop compliqué. Mais, au moins, si vous n'êtes pas satisfait du service rendu par un hébergeur, vous pouvez déménager chez un autre plus efficace grâce à la portabilité des données.

Donc ça m'importe assez peu de savoir que la plateforme GitHub est propriétaire, par exemple. Evidemment, ça serait mieux si elle était entièrement open source, mais le fait qu'elle ne le soit pas n'est pas vraiment un énorme problème. Le premier critère auquel je fais attention lorsque j'évalue un service d'hébergement est la richesse de leurs APIs. Est-ce que je peux récupérer toutes mes données si besoin ? Si leurs APIs sont riches, c'est bon signe, ils feront leur travail pour maintenir un service de qualité, car c'est le critère qui leur permettra de conserver leurs utilisateurs.

**En France, les élevés de collège et de lycée ne suivent pas de cours d'informatique. Pensez-vous que l'informatique devrait être une matière à part entière, et pas seulement un outil pour les autres matières ?**

Evidemment. La compréhension des données et du calcul formel est très importante désormais. C'est une forme d'alphabétisme. Sans aller jusqu'à maîtriser la programmation, il faut savoir comment les données fonctionnent. Cela fait écho à une discussion récente où je me suis rendu compte du gouffre qui

peut exister.

J'étais chez le docteur, pour faire quelques tests. L'un d'eux consistait à filmer les battements de mon cœur grâce aux ultra-sons et toute la séquence était enregistrée. C'était incroyable à voir ! Et donc, une fois terminé, je demande à l'accueil si je pouvais avoir les données. Pour être précis, j'ai demandé : « Est-ce que je pourrai avoir les données de l'échocardiogramme ? » L'assistante m'a répondu qu'ils pouvaient m'imprimer des images basse-résolution. J'ai alors répondu : « Merci, mais ce sont les données que je veux ». Elle m'a répondu que c'est bien ce qu'elle me proposait. Pour elle, le mot « données » n'avait pas la même signification précise que pour ceux qui ont appris ce que sont les données. Ma question impliquait évidemment que je voulais toutes les données qu'ils avaient enregistrées. C'est bien ce que signifie « Toutes les données », non ? Il ne devrait pas y avoir de perte d'information : c'est une copie bit par bit. Mais cela ne lui parlait pas. Pour elle, les données, c'est « quelque chose qui ressemble à ce que j'ai demandé ». Je parlais d'information, d'informatique, elle me parlait de perception.

Je suis bien conscient que mon point de vue est radical, mais je trouve que c'est une forme d'illettrisme de nos jours. Vous devez savoir faire la différence entre les vraies informations et les fausses informations et vous devez comprendre l'énorme différence d'application qui existe entre les deux. Si je me rends chez un autre médecin, vous imaginez bien la différence que ça fait si je lui présente la vidéo complète sur clé USB par rapport à des copies basse résolution d'images fixes. L'une est utile, l'autre ne sert strictement à rien.

Les entreprises qui comprennent le mieux la valeur des données, de données nous concernant, ont de plus en plus de moyens d'utiliser ces données à leur avantage, mais pas nécessairement dans le vôtre. Les cours d'informatique

sont une forme de défense contre ceci, une réponse immunitaire à un monde dans lequel la possession et la manipulation des données se transforme de plus en plus en pouvoir. Vous êtes mieux à même de comprendre comment les données peuvent être utilisées si vous les avez déjà manipulées vous-même.

Donc oui, je suis pour les cours d'informatique... mais pas seulement comme moyen de défense :-). C'est aussi une formidable occasion pour les écoles de réaliser quelque chose de collaboratif. L'enseignement se focalise trop souvent sur des apprentissages « individuels ». D'ailleurs, la coopération à l'école est souvent prohibée et on appelle cela de la triche. Or en cours d'informatique, la chose la plus naturelle est d'initier des projets open source ou de participer à des projets open source.

Bien sûr, tous les étudiants ne seront pas forcément doués ou hyper motivés pour cela, mais c'est la même choses dans toutes les autres matières. Je pense donc que les cours d'informatique sont une bonne opportunité d'exposer les élèves aux plaisirs du développement collaboratif. Ces cours devraient avoir un impact incroyable sur certains élèves, comme, par exemple, les cours de musique.

**Une toute dernière question : quel conseil donneriez-vous au programmeur en herbe qui souhaite découvrir la communauté des logiciels libres et open source ? Essayez de répondre en une phrase, pas avec un livre entier ☐**

Trouvez un projet ouvert que vous appréciez (et, idéalement, que vous utilisez) et commencez à y participer ; vous ne le regretterez pas !