Mes données dans un nuage ? — Oui mais le mien

Plutôt que de se résigner à l'usage de services en ligne n'offrant aucune garantie réelle de confidentialité, Frank Karlitschek a décidé de ne pas se contenter de prêcher la bonne parole mais de passer à l'acte en élaborant (avec d'autres) un projet qui remporte un succès grandissant : un logiciel libre et open source *de stockage de données. En revenant sur l'historique du projet ownCloud, il nous rappelle au passage les clés de la réussite (ne perdons pas de vue la proportion importante de projets open source qui n'aboutissent jamais) : développement collaboratif du code ouvert, prenant appui sur des outils et choix techniques ayant déjà une large base de développeurs, flexibilité, compatibilité multi-plateforme…*

*Cet article donne quelques indications plus précises sur les technologies mises en œuvre qui peuvent laisser perplexe le lecteur non développeur, mais la démarche et la philosophie de l'*open source *y apparaîtront pour tous avec clarté. L'enjeu, c'est de rendre à l'utilisateur le contrôle de ses données.*

Au fait, Framasoft dispose depuis un an de son propre ownCloud [1], pourquoi pas vous ?

Pourquoi j'ai créé OwnCloud et l'ai rendu *open source*

par **Frank Karlitschek**, fondateur de ownCloud et mainteneur de l'architecture globale du projet.

Article original : Why I Built OwnCloud and Made It Open Source Traduction Framalang : Asta, r0u, KoS, Wan, Omegax, goofy, Diab

Il y a 4 ans, j'étais au CampKDE à San Diego, je donnais une conférence sur la protection des données, mettant en garde le public sur les risques pour leur vie privée auprès des fournisseurs de *cloud* – en particulier Dropbox. « – Eh bien fais-le toi-même », m'a-t-on dit. Bien sûr, j'avais déjà créé des choses dans le passé, alors bien sûr, j'ai dit que j'allais le faire. Et c'est là que j'ai commencé mon odyssée, en premier lieu pour me protéger moi-même, mes amis et mes collègues de l'espionnage des gouvernements et d'autres méchants, et plus tard – quand j'ai vu l'intérêt croître dans le monde – pour concevoir un projet concret et efficace.

> *je n'avais pas envie d'envoyer mes données à un service tiers pour qu'il les stocke on ne sait où*

Évidemment, je devais décider d'un certain nombre de choses avant de commencer, notamment ce que je voulais que fasse le logiciel, quelle plateforme de développement utiliser, comment le structurer et bien sûr il fallait que je lui trouve un nom : ownCloud (NdT : littéralement, « le nuage qu'on possède »).

Mes amis et moi avions besoin d'un moyen de synchroniser nos images, nos documents et même nos vidéos en passant d'un appareil à l'autre (au lieu d'utiliser une clé USB), nous voulions aussi partager ces fichiers avec nos amis et nos proches. À l'époque, Dropbox devenait très populaire, mais je n'avais pas envie d'envoyer mes données à un service tiers pour qu'il les stocke on ne sait où. Je voulais créer une plateforme que mes amis puissent utiliser sur les espaces de stockage qu'ils avaient déjà, à la différence du *cloud*, pas seulement pour synchroniser et partager, mais aussi une plateforme assez flexible pour qu'on puisse y créer des applications.

Bien sûr ownCloud allait être *open source*.

Je faisais déjà partie de la communauté *open source*, mais ce

n’est pas la seule raison. En faisant de l’*open source* je concevais un code qui serait complètement transparent (et donc aurait peu de risques de comporter des « portes dérobées » pour entrer dans mes données). De plus je pouvais compter sur un grand nombre de personnes animées des mêmes convictions pour m’aider à créer ownCloud, je n’étais donc pas tout seul. Et je pouvais réutiliser les technologies d’autres projets. Comme SABREDAV, qui est le framework que nous utilisons pour la communication WebDAV du serveur (CalDAV, CardDAV et WebDAV sont tous utilisés par ownCloud), et nous utilisons aussi jQuery. Nous avons également utilisé csync pour les capacités de synchronisation bi-directionnelle du client de bureau et Qt pour l’interface utilisateur multi-plateforme. Je n’ai pas eu à réinventer la roue une fois de plus, je n’ai eu qu’à assembler ce qui existait déjà pour que tout fonctionne.

Mais comme je l’ai déjà dit, je savais ce que je voulais : ownCloud devait être plus qu’une « app ». Bien sûr, stocker les données d’une manière sûre et sécurisée est une chose importante. Mais en fin de compte, les gens veulent faire quelque chose de leurs données, alors j’ai voulu ajouter davantage de fonctionnalités à travers les applications ownCloud. Les applications sont des extensions qui peuvent implémenter des fonctionnalités telles que la détection de virus, la journalisation des accès et des changements de fichiers, le *versionnage*, le chiffrement, l’édition de fichiers et bien d’autres choses. Ce genre d’intégration du stockage de fichiers avec d’autres services est essentiel pour le développement futur.

Je voulais que mon projet soit flexible, de sorte que les gens puissent s’appuyer sur ownCloud (et beaucoup l’ont fait, avec une application type « Google News », un *streamer* de vidéos, un lecteur de musique, un calendrier – et plus encore) et que ownCloud puisse s’intégrer dans de nombreux environnements. Par exemple, n’importe quel client WebDAV devait pouvoir accéder à ownCloud dès le départ et le concept d’applications

internes est là aussi depuis le début du projet.

Bien entendu, nous sommes plus avancés à présent – il y a des API de partage et d'administration, des API internes pour les applications utilisant OCS, il existe des bibliothèques pour mobile (que nous avons rendues *open source*) et qui permettent l'intégration à d'autres applications mobiles, une base de données clés-valeurs pour un usage général de stockage de données, de synchronisation, et davantage encore. Ensuite, il y a l'intégration de systèmes de stockage externe comme FTP, S3, SWIFT, CIFS, iRODS et beaucoup d'autres. Mais même à l'époque où nous avons commencé, les intentions étaient claires – construire quelque chose d'assez flexible pour que les gens puissent créer des solutions auxquelles nous n'avions pas pensé.

Et c'est justement ça, la puissance de l'*open source*.

Nous (ma communauté grandissante et moi) avons évalué différentes options pour trouver la bonne technologie qui pourrait tourner sur chaque plateforme, du micro serveur jusqu'à des clusters de serveurs, qui aurait toutes les fonctionnalités et serait connue d'un grand nombre de développeurs. C'est pourquoi nous avons opté pour PHP et JS pour la partie serveur, C++ pour la synchronisation des Clients, Objective-C pour iOS et Java pour Android.

Il y avait plusieurs critères architecturaux à remplir dès le départ : multiplateforme, facilité d'extension, support des infrastructures, haute disponibilité basée sur les composants les plus largement utilisés. Donc, nous avons choisi PHP, pour cibler la pile « LAMP » (Linux / Apache / MySQL / PHP) qui est la plus répandue et éprouvée des plateformes permettant tout cela.

C'est également un projet *open source* et PHP est disponible gratuitement, facile à trouver, et multiplateforme (variantes Windows et Linux, IIS, Apache et autres serveurs Linux). Il

bénéficie d'une communauté massive de développeurs dont beaucoup sont très expérimentés. Enfin, c'est un langage facilement accessible pour la communauté. Avec tout ça, c'était une évidence.

> *« L'open source est la seule solution pour un stockage de données réellement sécurisé »*

Comme j'ai commencé ce projet par une conférence sur la sécurité et la confidentialité, il était essentiel d'avoir la meilleure sécurité possible pour les API. J'ai choisi un chiffrement SSL fort pour toutes les API WebDAV et REST. L'authentification est faite via la méthode basique, qui est très simple et facile à gérer. On peut également utiliser SAML, fourni au travers de son implémentation Shibboleth. En complément OAuth et l'authentification à deux facteurs sont disponibles, et nous profitons même de la flexibilité de ownCloud pour intégrer un backend personnalisé, en utilisant des jetons à la place des mots de passe standards.

Je suis convaincu que le stockage de fichiers n'est pas seulement un service web ou une infrastructure informatique de plus. C'est là où les gens et les entreprises stockent et gèrent leurs données les plus importantes. C'est pourquoi il est essentiel de le rendre aussi sécurisé que possible. Avec un logiciel propriétaire, vous ne pouvez jamais être sûr qu'il n'y a pas une porte dérobée ou d'autres problèmes de sécurité. L'open source est la seule solution pour un stockage de

données réellement sécurisé. Voilà ce que j'ai fait et pourquoi je l'ai fait. J'ai mis à ce travail toute ma passion pour l'*open source* et il a aussi demandé beaucoup de soin !

- Le site du projet ownCloud
- Contribuer au développement d'ownCloud

Notes

[1] Tiens par exemple, vous voulez de quoi imprimer de chouettes posters qui expliquent ce qu'est le logiciel libre ? C'est par là

Passer de l'exercice scolaire à la maintenance des paquets (Libres conseils 28/42)

Chaque jeudi à 21h, rendez-vous sur le framapad de traduction, le travail collaboratif sera ensuite publié ici même.

Traduction Framalang : satanas_g, Sphinx, Sky, Julius22, peupleLà, lamessen, goofy

Du débutant au professionnel

Jonathan Riddell

Jonathan Riddell est développeur KDE et Kubuntu, actuellement employé par Canonical. Quand il n'est pas devant un ordinateur, il fait du canoë sur les rivières d'Écosse.

Il y avait un bogue dans le code. Un bien méchant en plus : un

plantage sans enregistrement des données. C'est bien là le problème dès qu'on regarde le code, on trouve des trucs à réparer. C'est facile de s'impliquer dans le logiciel libre ; le plus dur est d'en sortir. Après le premier bogue réparé, il y en a d'autres, et de plus en plus, tous à portée de main. Les corrections de bogues mènent à l'ajout de fonctionnalités, ce qui mène à la maintenance de projet, ce qui mène à faire fonctionner une communauté.

Tout a commencé en lisant Slashdot, cette masse d'actualité geek et technique peu filtrée avec des commentaires de quiconque peut recharger assez vite pour être en haut de liste. Chaque actualité était intéressante et excitante, apportait un éclairage nouveau sur le monde de la technologie qui finissait par me fasciner. Je n'avais plus à accepter ce qui m'était donné par de grandes entreprises de logiciels, je pouvais voir là, dans la communauté du logiciel libre, le code se développer devant moi.

En tant qu'étudiant, il était possible de finir les exercices donnés par les professeurs très rapidement. Mais les exercices ne sont pas des programmes terminés. Je voulais savoir comment appliquer les compétences basiques qu'ils m'avaient données dans le monde réel en écrivant des programmes résolvant des problèmes réels pour les gens. J'ai donc recherché du code, qui n'était pas difficile à trouver, il se trouvait là, sur Internet, en fait. En regardant le code des programmes que j'utilisais de plus près, j'y ai décelé de la beauté. Non pas parce que le code était parfaitement soigné ou bien structuré, mais parce que je pouvais le comprendre avec les concepts que j'avais déjà appris. Ces classes, méthodes et variables étaient bien en place, me permettant de résoudre les problèmes pertinents. Le logiciel libre est le meilleur moyen de franchir le pas entre savoir comment finir ses exercices de cours et comprendre comment de vrais programmes sont écrits.

Tous les étudiants en informatique devraient travailler sur du logiciel libre comme sujet de leur mémoire. Sinon, vous avez

de grandes chances d'y passer six mois à un an pour qu'il finisse au sous-sol d'une bibliothèque sans être jamais plus consulté. Seul le logiciel libre permet d'exceller en faisant ce qui va de soi : vouloir apprendre comment résoudre des problèmes intéressants. À la fin de mon projet, des programmeurs de la NASA utilisaient mon outil de création de diagrammes en **UML** (NdT : langage de modélisation unifié) et il reçut des prix au cours de réceptions somptueuses. Avec le logiciel libre, on peut résoudre de vrais problèmes pour de vrais utilisateurs.

La communauté des développeurs est remplie de personnes formidables, passionnées et dévouées à leur travail, sans espoir autre de récompense qu'un programme d'ordinateur couronné de succès. La communauté des utilisateurs est également incroyable. Il est satisfaisant de savoir qu'on a aidé quelqu'un à résoudre un problème. Et j'apprécie les messages de remerciement que je reçois.

Après avoir écrit un logiciel utile, il faut le mettre à la disposition du plus grand nombre. Le code source ne va pas fonctionner pour la plupart des gens, il doit être compilé. Avant d'être impliqué, je trouvais que le fait de compiler était une manière un peu paresseuse de contribuer au logiciel libre. Vous vous attirez la plus grande partie de la reconnaissance sans rien avoir à coder. C'est, quelque part, quelque chose d'injuste. De même, la gestion de la communauté nécessaire pour porter un projet de logiciel libre peut aussi être vue comme une façon de s'attirer la reconnaissance sans faire de code.

Les utilisateurs dépendent beaucoup des *packagers* (NdT : les « empaqueteurs » qui préparent et maintiennent les paquets logiciels). Il est nécessaire que leur travail soit à la fois rapide, pour satisfaire ceux qui veulent la dernière version, et fiable, pour ceux qui veulent la stabilité (autant dire tout le monde). La partie la plus délicate, c'est que cela implique de travailler avec les logiciels des autres, qui sont

toujours « cassés ». Une fois que le logiciel est lâché dans la nature, commencent à emerger des problèmes qui n'étaient pas repérables sur l'ordinateur de l'auteur. Il est possible que le code ne puisse pas être compilé avec une version de compilateur différente, peut-être que la licence n'est pas claire et ne permet pas de le copier, peut-être que la gestion des versions est incohérente et qu'une mise à jour mineure est incompatible, ou encore que la taille de l'écran est différente, les environnements de bureau peuvent aussi l'affecter, quelquefois, des bibliothèques tierces nécessaires ne sont pas encore à jour. De nos jours, le logiciel doit pouvoir tourner sur différentes architectures. Les processeurs 64 bits ont occasionné pas mal de problèmes quand ils sont devenus courants. Aujourd'hui, ce sont les processeurs ARM qui déjouent les calculs des codeurs. Les *packagers* doivent régler tous ces problèmes pour donner aux utilisateurs quelque chose qui fonctionne de façon fiable.

Nous avons une règle chez Ubuntu selon laquelle les paquets avec des tests unitaires doivent inclure ces mêmes tests dans le processus de la création des paquets. Souvent, ils échouent et l'auteur du logiciel nous dit que les tests sont uniquement à son usage. Malheureusement, quand il s'agit de logiciel, il n'est jamais assez fiable de le tester soi-même, il doit aussi être testé par d'autres. Un test unique est rarement suffisant, il faut une approche à plusieurs niveaux. Les tests unitaires du programme original devraient être le point de départ, ensuite, le *packager* les teste sur son propre ordinateur, il faut ensuite que d'autres personnes les testent aussi. L'installation automatique et les tests de mise à jour peuvent être scriptés assez correctement sur les services d'informatique dans le nuage. L'envoyer dans la branche de développement d'une distribution permet d'effectuer plus de tests avant de le voir distribué en masse quelques mois après. À chaque étape, des problèmes peuvent être et seront découverts, ils devront être corrigés, puis ces correctifs eux-mêmes devront être testés. Il n'y a donc pas forcément à

écrire beaucoup de code, mais il y a pas mal de travail pour passer le logiciel de 95 % à 100 % prêt. Ces 5 % sont la partie la plus difficile, un lent et délicat processus qui demande une grande attention pendant tout son cours.

Vous ne pouvez pas faire de paquets sans une bonne communication avec les développeurs en amont. Quand des bogues se produisent, il est vital de pouvoir trouver la bonne personne à laquelle parler rapidement. Il est important d'apprendre à bien les connaître comme des amis et des collègues. Les conférences sont vitales pour cela, car rencontrer quelqu'un apporte beaucoup plus de contexte à un message sur une liste de diffusion qu'une année entière de messages.

Une des faces cachées du monde du logiciel libre réside dans la communication par les canaux IRC privés utilisés par les principaux membres d'un projet. Tous les grands projets en ont. Quelque part, Linus Torvalds a un moyen de discuter avec Andrew Morton et les autres sur ce qui est bon et sur ce qui est mauvais dans Linux. Ils sont plus sociaux que techniques et, quand on en abuse, ils peuvent être très antisociaux pour la communauté en général. Mais pour les moments où on a besoin d'un canal de communication rapide sans bruit parasite, ils fonctionnent bien.

Tenir un blog est un autre moyen de communication important dans la communauté du logiciel libre. C'est notre principale méthode pour promouvoir à la fois le logiciel que nous produisons et nous-mêmes. Non pas que ce soit utilisé éhontément pour de l'auto-promotion (il est inutile de prétendre que vous sauverez des vies avec votre blog…), mais parler de votre travail sur le logiciel libre aide à construire une communauté. Cela peut même vous valoir de trouver un travail ou d'être reconnu dans la rue.

Ces histoires venant de Slashdot, à propos de développements de nouvelles technologies, ne concernent pas des personnalités

éloignées que vous ne rencontrerez jamais comme dans la presse *people*. Elles concernent des personnes qui ont trouvé un problème et qui l'ont résolu en utilisant l'ordinateur qu'elles avaient en face d'elles. Pendant quelques années, j'ai édité le site d'informations de KDE, trouvant les personnes qui résolvaient des problèmes, créaient des idées novatrices, s'acharnaient longuement à améliorer un logiciel jusqu'à ce qu'il soit d'une qualité suffisante, et j'en parlais au monde entier. Je n'ai jamais été à court d'histoires à raconter ni de personnes à présenter à tout le monde.

Mon dernier conseil est de conserver de la diversité. Il existe une telle richesse de projets intéressants à explorer, qui vous permettent d'apprendre et de progresser. Mais une fois que vous avez atteint une position de responsabilité, il peut être tentant d'y rester. Après avoir aidé à créer une communauté pour Kubuntu, je repars temporairement vers un travail sur Bazaar, un projet très différent, orienté sur les développeurs plutôt que sur des utilisateurs novices en technologies. Je peux à nouveau apprendre comment le code devient une réalité utile, comment une communauté communique, comment la qualité est maintenue. Ce sera un défi amusant et j'ai hâte de m'y attaquer.

Et si l'ignorance était enrichissante ? (Libres conseils 27/42)

Chaque jeudi à 21h, rendez-vous sur le framapad de traduction, le travail collaboratif sera ensuite publié ici même.

Traduction Framablog : Sphinx, purplepsycho, Cyrille L., lamessen

Ce que je suis contente de ne pas avoir su

Alexandra Leisse

Alexendra Leisse a quitté une scène pour en rejoindre une autre. Elle a transformé son autre passion (les logiciels et le Web) en un métier. Après une période de transition de douze mois en freelance dans le logiciel et l'opéra – et noyée par de nombreuses heures d'activités dédiées à KDE, elle a rejoint Nokia et le développement de la plateforme Qt en tant que gestionnaire de la communauté Web. C'est la femme derrière le réseau de développement Qt et les activités de sa communauté sur la toile. Bien qu'elle soit diplômée en art lyrique, elle refuse la plupart du temps de chanter en public.

Introduction

Quand Lydia m'a demandé de rejoindre son projet de livre sous-titré « les choses que j'aurais voulu savoir », mon esprit est resté vide. Les choses que j'aurais voulu savoir mais que je ne savais pas ? Rien ne me venait à l'esprit.

Je ne dis pas que je n'aie pas eu besoin de savoir quoi que ce soit, au contraire. J'ai eu beaucoup à apprendre et j'ai fait un nombre incalculable d'erreurs. Mais les situations ou les erreurs que j'aurais voulu éviter ? Je n'arrive pas à y penser.

Nous avons tous cette fâcheuse tendance à regarder les choses que nous pourrions mieux faire, les choses que nous ne savons pas, et nous les voyons comme des faiblesses. Mais que dire des faiblesses qui sont des atouts ?

Voici ma propre histoire sur l'ignorance, la naïveté, les mauvaises impressions et comme je suis heureuse de ne pas en avoir eu la moindre idée.

Les noms

Je n'avais aucune idée de qui était ce gars que j'avais rencontré lors de mon premier jour de travail. Il est entré dans la pièce, s'est présenté et a commencé à poser des questions me donnant l'impression que tout ce que je penserais serait insensé. Il était apparemment bien renseigné sur ce que je faisais sur KDE et les personnes que je côtoyais. Cependant, nos points de vue sur le sujet semblaient différents. À un moment, ses provocations ont fini par me fatiguer et j'ai perdu patience. Je lui ai alors dit qu'avec les personnes, ce n'était pas toujours aussi facile que les ingénieurs l'imaginent.

Juste après son départ après une heure de discussion, j'ai cherché son nom sur Google : Matthias Ettrich. Ce que j'ai lu m'a expliqué pourquoi il avait posé ces questions. Si j'avais su avant qu'il était un des fondateurs du projet KDE, j'aurais débattu avec lui d'une manière bien différente, voire pas du tout.

Ces dernières années, j'ai dû chercher quelques noms et à chaque fois, j'ai été heureuse de le faire *après* le premier contact.

C'est probablement mon idée la plus importante. Lorsque j'ai rencontré toutes ces personnalités du Libre et de l'*open source* pour la première fois, je n'avais jamais entendu leurs noms auparavant. Je ne savais rien de leurs histoires, mérites ou échecs. J'ai approché tout le monde de la même façon : le contact visuel. En étant ignorante (ou naïve selon certains), je ne me sentais pas inférieure par rapport aux personnes que je rencontrais lorsque j'ai commencé mon aventure au sein du Libre et de l'*open source*. Je savais que j'avais beaucoup à

apprendre mais je n'ai jamais eu l'impression d'avoir un rang inférieur aux autres en tant qu'individu.

« Projet de grande envergure »

Je n'avais pas suivi religieusement dot.kde.org ni PlanetKDE et encore moins ces innombrables publications liées au Libre et à l'*open source*, avant de commencer à m'intéresser à ce qui se passait sur les listes de diffusion KDE. Je voyais ces canaux avant tout comme moyen de communiquer avec un public choisi, principalement des utilisateurs et des contributeurs du projet en tant que tel.

Pendant un certain temps, je n'avais pas conscience que les articles que je publiais sur The Dot pourraient être repris par des journalistes. Je m'appliquais à les écrire parce que je voulais faire du bon boulot et non pas parce que j'avais peur de passer pour folle auprès du reste du monde. La liste de presse était maintenue par d'autres personnes et ce que j'écrivais ne me paraissait pas important non plus. Je voulais toucher certaines personnes. Pour cela les canaux officiels et mon propre blog me semblaient être les moyens les plus efficaces.

Être citée sur ReadWriteWeb après avoir annoncé sur mon blog que je commencerai un nouveau boulot fut un choc pour moi. Non pas parce que j'ignorais que des gens lisaient ce que j'écrivais (j'espérais bien qu'ils le lisent !) mais je ne m'attendais pas à ce que ça soit un sujet d'une telle importance. Ce n'était même pas pendant les vacances d'été. Encore heureux que personne ne me l'ait dit, je n'aurais pas été capable de publier ne serait-ce qu'une seule ligne.

L'œil étranger

Il y a quelque temps, quand j'ai assisté à ma première conférence, j'avais la ferme conviction que j'étais différente

des autres participants. je me voyais comme une étrangère parce que je n'avais pas grand-chose en commun avec qui que ce soit à part un vague intérêt pour la technologie : je travaillais en freelance depuis quelques années déjà, après mon diplôme universitaire ; je n'avais aucune éducation pertinente dans le domaine, et j'étais mère d'un enfant de 10 ans. Sur le papier en tout cas, il ne pouvait pas y avoir plus éloignée des suspects habituels qu'on rencontre dans les projets FOSS.

En 2008 j'ai assisté à un *sprint* (NdT : phase de développement, généralement perçue comme intense, aboutissant à un produit fonctionnel) KOffice au sein de l'équipe promotion et marketing de KDE pour préparer la sortie de la version 2.0. L'idée initiale était d'esquisser une série d'activités promotionnelles autour de cette sortie afin de développer à la fois le support développeur et utilisateur. Pour celui-ci, nous étions trois à suivre un chemin parallèle à celui concernant les développeurs.

Nous avons essayé de comprendre comment nous pourrions positionner KOffice et adapter la communication au public ciblé. Très tôt dans le processus, nous avons découvert que nous devions faire marche arrière : à ce stade, le manque de maturité de la suite rendait impossible son positionnement comme option pour les utilisateurs non avertis. Nous devions nous en tenir aux développeurs et aux précurseurs. C'était difficile à vendre à certains développeurs, mais en tant qu'étrangers nous avions la chance de regarder le logiciel sans penser à tout le sang, la sueur et les larmes versés dans le code.

Pour beaucoup de projets, de n'importe quelle sorte, jeter un œil objectif à la situation donne du fil à retordre aux contributeurs principaux. Nous avons tendance à ne pas voir les grands succès quand nous sommes très concentrés sur des problèmes de détails et réciproquement. Parfois, nous manquons une occasion parce que nous pensons que ça n'a rien à voir

avec ce que nous faisons (ou, pour commencer, parce que personne ne voudrait que ça ait quelque chose à voir).

Dans tous ces cas, les contributeurs extérieurs au projet ont le potentiel pour apporter des points de vue différents à la discussion, particulièrement quand il s'agit de déterminer un ordre de priorité. C'est encore plus utile quand ce ne sont pas des développeurs : ils poseront différentes questions, sans ressentir de pression face à la connaissance et à la compréhension de tous les détails techniques ; ils peuvent aussi aider pour les décisions ou la communication sur un plan moins technique.

Conclusion

L'ignorance est une bénédiction. Ce n'est pas seulement vrai pour les individus qui profitent de l'insouciance qui en résulte mais aussi pour les projets que ces individus rejoignent. Ils apportent différents points de vues et expériences.

Et maintenant, filez et trouvez vous-même un projet qui vous intéresse, indépendamment de ce que vous pensez savoir.

Pour que le Libre aille vers l'Université (Libres conseils 6/42)

Université et communauté

Kevin Ottens

Kevin Ottens est un « hacker » de longue date au sein de la communauté KDE[1]. Il a largement contribué à la plateforme KDE, en particulier à la conception des API et frameworks. Diplômé en 2007, il est titulaire d'un doctorat en informatique qui l'a amené à travailler, en particulier, sur l'ingénierie des ontologies[2] et les systèmes multi-agents. Le travail de Kevin au KDAB inclut le développement de projets de recherche autour des technologies KDE. Il vit toujours à Toulouse, où il est enseignant à mi-temps dans son université d'origine.

Introduction

Les communautés du Libre sont principalement animées par l'effort de bénévoles. De plus, la plupart des personnes qui s'impliquent dans ces communautés le font lors de leur cursus universitaire. C'est la période idéale pour s'engager dans de telles aventures : on est jeune, plein d'énergie, curieux et l'on veut probablement façonner le monde à son image. Voilà tous les ingrédients d'un bon travail bénévole.

Mais, en même temps, être étudiant ne laisse pas forcément beaucoup de temps pour s'engager dans une communauté du Libre. En outre, la plupart de ces communautés sont assez vastes et les contacter peut faire peur.

Cela soulève évidement une question dérangeante : si les communautés du Libre ne réussissent pas à attirer la nouvelle génération de contributeurs talentueux, est-ce parce qu'elles ne cherchent pas activement à étendre leur activité dans les universités ? Cette question pertinente, nous avons essayé d'y répondre dans le contexte d'une communauté qui produit des logiciels, à savoir KDE. Dans cet article, nous nous concentrerons sur les aspects auxquels nous n'avions pas initialement pensé mais qu'il nous a fallu aborder pour répondre à cette question.

Construire un partenariat avec une université locale

Tout commence, réellement, par le contact avec les étudiants eux-mêmes. Et pour ça, rien de mieux que de se rendre directement dans leur université et de leur montrer à quel point les communautés du Libre peuvent être accueillantes. À cet effet, nous avons construit un partenariat avec l'université *Paul Sabatier* de Toulouse plus précisément, avec l'un de ses cursus – nommé IUP ISI (NdT : Ingénierie des

Systèmes Informatiques) – axé sur le développement logiciel.

L'IUP ISI était très orienté sur les connaissances « pratiques » et avait à ce titre un programme pré-établi pour les projets étudiants. Un point particulièrement intéressant de ce programme est le fait que les étudiants travaillent en équipe « inter-promotions ». Des étudiants de troisième et quatrième années, généralement en équipes de 7 à 10, apprennent à collaborer autour d'un objectif commun.

La première année de notre expérience, nous nous sommes raccrochés à ce programme en proposant de nouveaux sujets pour les projets, et en nous concentrant sur des logiciels développés au sein de la communauté KDE. Henri Massie, directeur du cursus, a très bien accueilli cette idée, et nous a laissés mettre cette expérience en place. Pour cette première année, nous nous sommes vu attribuer deux créneaux horaires pour les « projets KDE ».

Pour créer rapidement un climat de confiance, nous avons décidé, cette année-là, d'offrir quelques garanties concernant le travail des étudiants :

- Aider les professeurs à avoir confiance dans les sujets abordés : les projets sélectionnés étaient très proches des sujets enseignés à l'IUP ISI (c'est pourquoi, pour cette année, nous avons ciblé un outil de modélisation UML et un outil de gestion de projet) ;
- donner un maximum de visibilité aux professeurs : nous avons mis à leur disposition un serveur, sur lequel étaient régulièrement compilés les projets étudiants, accessible à distance à des fins de test ;
- faciliter la participation des étudiants à la communauté : les responsables des projets étaient désignés pour jouer le rôle du « client », soumettre leurs exigences aux étudiants et les aider à trouver leur chemin dans le dédale de la communauté ;
- enfin, pour mettre le pied à l'étrier aux étudiants,

nous leur avons donné un petit cours sur « Comment développer avec Qt et les autres frameworks produits par KDE ».

Au moment de l'écriture de ces pages, cela fait cinq ans que nous menons de tels projets. De petits ajustements dans l'organisation ont été apportés ici et là, mais la plupart des idées sous-jacentes sont restées les mêmes. La majorité des changements ont été le résultat d'un intérêt grandissant de la communauté, désireuse d'établir un partenariat avec les étudiants, et d'une plus grande liberté pour nous quant aux sujets que nous pourrions couvrir dans nos projets.

De plus, tout au long de ces années, le directeur nous a donné une aide et des encouragements constants, attribuant effectivement plus de créneaux pour les projets de la communauté du Libre et prouvant que notre stratégie d'intégration était juste : établir de la confiance dès le début est la clé d'un partenariat entre la communauté du Libre et l'Université.

Comprendre que l'enseignement est un processus interactif

Durant ces années à tisser des liens entre la communauté KDE et la filière IUP ISI, nous nous sommes retrouvés en situation d'enseignement afin d'assister les étudiants dans des tâches liées à leurs projets. Quand vous n'avez jamais enseigné à une classe pleine d'étudiants, vous avez sans doute encore une image de vous, il y a quelques années, assis dans une classe. En effet, la plupart des enseignants ont un jour été étudiants… Parfois même, pas le genre d'étudiant très discipliné ni attentif. Vous aviez sans doute l'impression d'être submergé : l'enseignant entrait dans la salle, faisait face aux étudiants, et déversait sur vous ses connaissances.

Ce stéréotype est ce que la plupart gardent à l'esprit de

leurs années d'études et la première fois qu'ils se retrouvent en situation d'enseigner, ils veulent reproduire ce stéréotype : arriver avec un savoir à transmettre.

La bonne nouvelle, c'est que rien n'est plus éloigné de la vérité que ce stéréotype. La mauvaise nouvelle, c'est que si vous essayez de reproduire ce stéréotype, vous allez très probablement faire fuir vos étudiants et ne ferez face qu'à un manque de motivation pour participer à la communauté. L'image que vous donnez de vous est la toute première chose dont ils vont se souvenir de la communauté : la première fois que vous entrez dans la salle de classe, vous êtes, pour eux, la communauté.

Pour éviter de tomber dans le piège de ce stéréotype, il vous faut prendre un peu de recul et réaliser ce que signifie réellement d'enseigner. Ce n'est pas un processus à sens unique où l'on livre la connaissance aux étudiants. Nous sommes arrivés à la conclusion que c'est en fait un processus à double sens : vous êtes amené à créer une relation symbiotique avec vos étudiants. Les étudiants et les enseignants doivent tous sortir de la salle de classe avec de nouvelles connaissances. Il vous faut livrer votre expertise, bien sûr, mais afin de le faire efficacement, vous devez en permanence vous adapter au cadre de référence de vos étudiants. C'est un travail qui rend très humble.

Cette prise de conscience génère pas mal de changements dans la manière d'entreprendre votre enseignement.

- Vous allez devoir comprendre la culture de vos étudiants. Ils ont probablement des expériences assez différentes des vôtres et vous allez devoir adapter votre discours à eux ; par exemple, les étudiants que nous avons formés font tous partie de la fameuse « génération Y » qui, en matière de leadership, loyauté et confiance, présente des caractéristiques assez différentes de la génération précédente.

- Vous allez devoir réévaluer votre propre expertise, puisque vous allez devoir adapter votre discours à leur culture. Vous aborderez vos propres connaissances selon un angle très différent de celui dont vous avez l'habitude, ce qui vous mènera inévitablement à des découvertes dans des domaines que vous pensiez maîtriser.
- Enfin, vous allez devoir vous forger des compétences en présentation ; l'enseignement consiste réellement à sortir de votre zone de confort afin de présenter vos propres connaissances tout en les gardant intéressantes et divertissantes pour votre audience. Cela fera de vous un meilleur présentateur.

Ainsi, vous deviendrez un meilleur enseignant. De plus, vous remplirez mieux vos objectifs : des étudiants bien formés, dont certains s'engageront dans la communauté du Libre.

Conclusion

Au bout du compte, pourquoi feriez-vous tous ces efforts pour établir une relation de confiance avec une université et sortir de votre zone de confort en améliorant votre manière d'enseigner ? Eh bien, cela se résume vraiment à la question initiale à laquelle nous avons tenté de répondre :

Si les communautés du Libre ne réussissent pas à attirer de nouveaux contributeurs venus des universités, est-ce simplement dû à leur inaction ?

D'après notre expérience, la réponse est oui. Au cours de ces cinq années passées à bâtir un partenariat avec l'IUP ISI, nous avons attiré deux étudiants par année en moyenne. Certains d'entre eux nous ont quittés après quelque temps, mais certains sont devenus des contributeurs très actifs. Les autres gardent encore une certaine nostalgie de cette période de leur vie et continuent de nous soutenir même s'ils ne contribuent pas directement. En ce moment même, nous avons une

équipe locale KDE qui a réussi à organiser efficacement une conférence de deux jours pour notre dernière « release party » (NdT : soirée de lancement).

Parmi ces anciens étudiants, pas un seul ne se serait impliqué dans le projet KDE sans ces projets universitaires. Nous serions passés complètement à côté de ces talents. Par chance, nous n'avons pas été inactifs.

[1] http://www.kde.org

[2] https://fr.wikipedia.org/wiki/Ontologie_(informatique)

Université et communauté

Kevin Ottens

Kevin Ottens est un « hacker » de longue date au sein de la communauté KDE[1]. Il a largement contribué à la plateforme KDE, en particulier à la conception des API et frameworks. Diplômé en 2007, il est titulaire d'un doctorat en informatique qui l'a amené à travailler, en particulier, sur l'ingénierie des ontologies[2] et les systèmes multi-agents. Le travail de Kevin au KDAB inclut le développement de projets de recherche autour des technologies KDE. Il vit toujours à Toulouse, où il est enseignant à mi-temps dans son université d'origine.

Introduction

Les communautés du Libre sont principalement animées par l'effort de bénévoles. De plus, la plupart des personnes qui s'impliquent dans ces communautés le font lors de leur cursus universitaire. C'est la période idéale pour s'engager dans de telles aventures : on est jeune, plein d'énergie, curieux et l'on veut probablement façonner le monde à son image. Voilà tous les ingrédients d'un bon travail bénévole.

Mais, en même temps, être étudiant ne laisse pas forcément beaucoup de temps pour s'engager dans une communauté du Libre. En outre, la plupart de ces communautés sont assez vastes et les contacter peut faire peur.

Cela soulève évidement une question dérangeante : si les communautés du Libre ne réussissent pas à attirer la nouvelle génération de contributeurs talentueux, est-ce parce qu'elles

ne cherchent pas activement à étendre leur activité dans les universités ? Cette question pertinente, nous avons essayé d'y répondre dans le contexte d'une communauté qui produit des logiciels, à savoir KDE. Dans cet article, nous nous concentrerons sur les aspects auxquels nous n'avions pas initialement pensé mais qu'il nous a fallu aborder pour répondre à cette question.

Construire un partenariat avec une université locale

Tout commence, réellement, par le contact avec les étudiants eux-mêmes. Et pour ça, rien de mieux que de se rendre directement dans leur université et de leur montrer à quel point les communautés du Libre peuvent être accueillantes. À cet effet, nous avons construit un partenariat avec l'université *Paul Sabatier* de Toulouse plus précisément, avec l'un de ses cursus – nommé IUP ISI (NdT : Ingénierie des Systèmes Informatiques) – axé sur le développement logiciel.

L'IUP ISI était très orienté sur les connaissances « pratiques » et avait à ce titre un programme pré-établi pour les projets étudiants. Un point particulièrement intéressant de ce programme est le fait que les étudiants travaillent en équipe « inter-promotions ». Des étudiants de troisième et quatrième années, généralement en équipes de 7 à 10, apprennent à collaborer autour d'un objectif commun.

La première année de notre expérience, nous nous sommes raccrochés à ce programme en proposant de nouveaux sujets pour les projets, et en nous concentrant sur des logiciels développés au sein de la communauté KDE. Henri Massie, directeur du cursus, a très bien accueilli cette idée, et nous a laissés mettre cette expérience en place. Pour cette première année, nous nous sommes vu attribuer deux créneaux horaires pour les « projets KDE ».

Pour créer rapidement un climat de confiance, nous avons décidé, cette année-là, d'offrir quelques garanties concernant le travail des étudiants :

- Aider les professeurs à avoir confiance dans les sujets abordés : les projets sélectionnés étaient très proches des sujets enseignés à l'IUP ISI (c'est pourquoi, pour cette année, nous avons ciblé un outil de modélisation UML et un outil de gestion de projet) ;
- donner un maximum de visibilité aux professeurs : nous avons mis à leur disposition un serveur, sur lequel étaient régulièrement compilés les projets étudiants, accessible à distance à des fins de test ;
- faciliter la participation des étudiants à la communauté : les responsables des projets étaient désignés pour jouer le rôle du « client », soumettre leurs exigences aux étudiants et les aider à trouver leur chemin dans le dédale de la communauté ;
- enfin, pour mettre le pied à l'étrier aux étudiants, nous leur avons donné un petit cours sur « Comment développer avec Qt et les autres frameworks produits par KDE ».

Au moment de l'écriture de ces pages, cela fait cinq ans que nous menons de tels projets. De petits ajustements dans l'organisation ont été apportés ici et là, mais la plupart des idées sous-jacentes sont restées les mêmes. La majorité des changements ont été le résultat d'un intérêt grandissant de la communauté, désireuse d'établir un partenariat avec les étudiants, et d'une plus grande liberté pour nous quant aux sujets que nous pourrions couvrir dans nos projets.

De plus, tout au long de ces années, le directeur nous a donné une aide et des encouragements constants, attribuant effectivement plus de créneaux pour les projets de la communauté du Libre et prouvant que notre stratégie d'intégration était juste : établir de la confiance dès le début est la clé d'un partenariat entre la communauté du Libre

et l’Université.

Comprendre que l’enseignement est un processus interactif

Durant ces années à tisser des liens entre la communauté KDE et la filière IUP ISI, nous nous sommes retrouvés en situation d’enseignement afin d’assister les étudiants dans des tâches liées à leurs projets. Quand vous n’avez jamais enseigné à une classe pleine d’étudiants, vous avez sans doute encore une image de vous, il y a quelques années, assis dans une classe. En effet, la plupart des enseignants ont un jour été étudiants… Parfois même, pas le genre d’étudiant très discipliné ni attentif. Vous aviez sans doute l’impression d’être submergé : l’enseignant entrait dans la salle, faisait face aux étudiants, et déversait sur vous ses connaissances.

Ce stéréotype est ce que la plupart gardent à l’esprit de leurs années d’études et la première fois qu’ils se retrouvent en situation d’enseigner, ils veulent reproduire ce stéréotype : arriver avec un savoir à transmettre.

La bonne nouvelle, c’est que rien n’est plus éloigné de la vérité que ce stéréotype. La mauvaise nouvelle, c’est que si vous essayez de reproduire ce stéréotype, vous allez très probablement faire fuir vos étudiants et ne ferez face qu’à un manque de motivation pour participer à la communauté. L’image que vous donnez de vous est la toute première chose dont ils vont se souvenir de la communauté : la première fois que vous entrez dans la salle de classe, vous êtes, pour eux, la communauté.

Pour éviter de tomber dans le piège de ce stéréotype, il vous faut prendre un peu de recul et réaliser ce que signifie réellement d’enseigner. Ce n’est pas un processus à sens unique où l’on livre la connaissance aux étudiants. Nous sommes arrivés à la conclusion que c’est en fait un processus

à double sens : vous êtes amené à créer une relation symbiotique avec vos étudiants. Les étudiants et les enseignants doivent tous sortir de la salle de classe avec de nouvelles connaissances. Il vous faut livrer votre expertise, bien sûr, mais afin de le faire efficacement, vous devez en permanence vous adapter au cadre de référence de vos étudiants. C'est un travail qui rend très humble.

Cette prise de conscience génère pas mal de changements dans la manière d'entreprendre votre enseignement.

- Vous allez devoir comprendre la culture de vos étudiants. Ils ont probablement des expériences assez différentes des vôtres et vous allez devoir adapter votre discours à eux ; par exemple, les étudiants que nous avons formés font tous partie de la fameuse « génération Y » qui, en matière de leadership, loyauté et confiance, présente des caractéristiques assez différentes de la génération précédente.
- Vous allez devoir réévaluer votre propre expertise, puisque vous allez devoir adapter votre discours à leur culture. Vous aborderez vos propres connaissances selon un angle très différent de celui dont vous avez l'habitude, ce qui vous mènera inévitablement à des découvertes dans des domaines que vous pensiez maîtriser.
- Enfin, vous allez devoir vous forger des compétences en présentation ; l'enseignement consiste réellement à sortir de votre zone de confort afin de présenter vos propres connaissances tout en les gardant intéressantes et divertissantes pour votre audience. Cela fera de vous un meilleur présentateur.

Ainsi, vous deviendrez un meilleur enseignant. De plus, vous remplirez mieux vos objectifs : des étudiants bien formés, dont certains s'engageront dans la communauté du Libre.

Conclusion

Au bout du compte, pourquoi feriez-vous tous ces efforts pour établir une relation de confiance avec une université et sortir de votre zone de confort en améliorant votre manière d'enseigner ? Eh bien, cela se résume vraiment à la question initiale à laquelle nous avons tenté de répondre :

Si les communautés du Libre ne réussissent pas à attirer de nouveaux contributeurs venus des universités, est-ce simplement dû à leur inaction ?

D'après notre expérience, la réponse est oui. Au cours de ces cinq années passées à bâtir un partenariat avec l'IUP ISI, nous avons attiré deux étudiants par année en moyenne. Certains d'entre eux nous ont quittés après quelque temps, mais certains sont devenus des contributeurs très actifs. Les autres gardent encore une certaine nostalgie de cette période de leur vie et continuent de nous soutenir même s'ils ne contribuent pas directement. En ce moment même, nous avons une équipe locale KDE qui a réussi à organiser efficacement une conférence de deux jours pour notre dernière « release party » (NdT : soirée de lancement).

Parmi ces anciens étudiants, pas un seul ne se serait impliqué dans le projet KDE sans ces projets universitaires. Nous serions passés complètement à côté de ces talents. Par chance, nous n'avons pas été inactifs.

[1] http://www.kde.org

[2] https://fr.wikipedia.org/wiki/Ontologie_(informatique)

Quand les passants de Sydney adorent le nouveau Windows

Lu sur LinuxFr et, tout juste, sous-titré par Framalang :

« ZDNet Australie a réalisé une vidéo dans les rues de Sydney en faisant semblant de montrer une vidéo de démonstration du prochain Windows 7, alors qu'il s'agit en réalité de… KDE 4 !

En effet, il a été de très nombreuses fois constatées que le bureau de la version bêta (Build 7000) du prochain Windows présente une ressemblance troublante à cet autre environnement très présent dans le monde de Linux qu'est KDE. Mais au-delà de l'apparence du bureau, on se rend compte que toutes ces fonctionnalités de KDE plaisent énormément aux personnes passées dans la vidéo, et ne se doutent pas (apparemment) qu'il ne s'agit pas de Windows. La plupart en revanche, critiquent Vista pour diverses raisons.

Alors, KDE4 est-il Windows 7, ou Windows est-il KDE 4, ou simplement un peu des deux finalement ou ils n'ont rien à voir ? À vous de juger… »

–> La vidéo au format webm