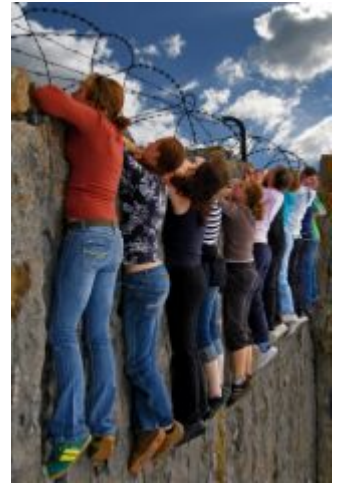


Comment détruire votre communauté en 10 leçons

Si vous avez le *malheur* de développer un projet « open source » au sein de votre entreprise alors vous courrez le risque de voir arriver une « communauté » qui peut à tout moment s'agréger autour du code source de votre logiciel et en menacer sa *bonne* gouvernance.



Heureusement le développeur Josh Berkus est là pour vous expliquer point par point comment faire pour être certain de ruiner et dissoudre toute velléité communautaire (au cours d'une intervention donnée il y a un mois à la Linux.Conf.au et relatée ici par Jonathan Corbet)^[1].

Un article évidemment ironique (qui détourne les [howto](#)), mais qui donne à réfléchir sur les relations subtiles et complexes qui peuvent exister entre les communautés et les entreprises qui œuvrent sur un même projet.

Pas toujours facile de se comprendre en effet quand les uns disent plutôt « logiciel libre » et les autres plutôt « open source » (voire même parfois carrément « [fauxopen source](#) »).

Et puis, c'est pratique, puisqu'on dispose ainsi de tout ce qu'il ne faut pas faire pour réussir un projet communautaire.

On notera que Josh Berkus avait la société [Sun Microsystems](#) en tête lorsqu'il a énoncé son propos (Sun soutient notamment MySQL et OpenOffice.org). Mais comme il le [précise](#) lui-même a posteriori sur son blog, cela peut s'appliquer à n'importe quelle « corporate open source » et de citer alors entre

autres Red Hat, Microsoft, IBM, Cisco, SugarCRM, Novell, Compiere, Borland, Google, ou encore Apple. « Si vous avez à faire à une compagnie qui possède un projet open source, vous pouvez être sûr à 95% qu'elle suit au moins un des dix points mentionnés ci-dessous »

Et de conclure positivement sur la nécessité de cultiver l'un des mots clés les plus importants de la communauté : la confiance.

Comment détruire votre communauté : mode d'emploi

[How to destroy your community](#)

Jonathan Corbet – 18 janvier 2010 – LWN.net

(Traduction Framalang : Olivier, Daria et Don Rico)

Le réputation de Josh Berkus en tant que hacker de [PostgreSQL](#) n'est plus à faire, mais ce n'est pas sa seule compétence, puisqu'il a aussi acquis une précieuse expérience durant sa pige au « Laboratoire de Destruction des Communautés », plus connu sous le nom de Sun Microsystems. Il y a suivi « l'enseignement breveté en 10 étapes » pour apprendre à débarrasser un projet de toute ingérence de la communauté.

La présentation très dynamique qu'a donnée Josh à la Linux.Conf.au sur le sujet était la première discussion de la miniconf [L'économie de l'open source](#) ; l'audience lui a réservé un accueil chaleureux.

Si vous êtes développeur dans une grosse entreprise, vous vous rendrez rapidement compte que les communautés de développement des logiciels libres sont une plaie. Dites adieu à vos plans marketing, par exemple, car elle se chargera d'introduire le logiciel dans des pays où vous êtes absents et pour lesquels vous n'avez pas de plan. Ils flanqueront par terre vos prévisions produits en sortant des innovations non-prévues, en

implémentant des fonctionnalités des années avant ce que vous aviez planifié, ou pire encore, des fonctionnalités qui devaient être réservées à la version propriétaire de votre logiciel. Les communautés de logiciels libres sont d'éternelles insatisfaites, elles n'ont de cesse de vouloir améliorer les choses. Elles ont tendance à redéfinir les relations avec vos partenaires et vos clients, et vos commerciaux ne savent plus à quel saint se vouer. Et sans arrêt elles vous dérangent : un e-mail par ci, une conférence à laquelle vous devriez assister par là, etc.

Mais heureusement, des solutions existent pour vous débarrasser de la menace que représente cette communauté. Il suffit d'appliquer une ou plusieurs des étapes suivantes.

1. Rendez le projet dépendant d'outils complexes

Il a remarqué qu'en général, les entreprises n'ont pas de problème avec cette technique puisqu'elles aiment s'appuyer sur leurs propres outils. Pour les projets où la communauté n'est pas la bienvenue, il faut par exemple employer des systèmes singuliers qu'on ne trouve nulle part ailleurs. Un [système de contrôle de version](#) propriétaire (*NdT : version control system*) est absolument obligatoire. Mieux encore, un [outil de suivi des problèmes](#) (*NdT : issue tracking system*) avec un nombre limité de licences, afin que tout le monde doive s'y connecter avec le même compte.

N'oubliez pas de mettre en place un site Web qui respecte la parité : 50% du temps planté, 50% du temps opérationnel. Ne pas mettre de site à disposition ne suffira pas : dans de telles situations, la communauté a la fâcheuse habitude de créer le sien. Avec un site bancal, en revanche, vous vous en prémunirez et vous assurerez que l'information restera bien cachée.

2. Attirez les participants nocifs et optimisez

les dégâts qu'ils peuvent engendrer

Ce cas particulier nécessite quelques étapes :

1. Prenez sur vous et engagez-vous dans de longs débats avec ces personnes et dénoncez-les sur les listes du projet.
2. Au bout d'un certain temps, bannissez-les par décret ; évitez à tout prix tout processus communautaire.
3. Les gens bannis déverseront leur bile ailleurs. Vous devez les suivre et poursuivre votre débat sur ces sites externes.
4. Enfin, la communauté se plaindra de ce comportement. Votre réponse sera simple : réintégrez les enquiquineurs à la communauté. Reprenez à la première étape et recommencez.

D'après Josh, un casse-pieds bien pris en main peut annihiler une communauté de plusieurs centaines de membres.

3. Ne fournissez pas de documentation

Aucune information utile ne devrait être disponible, ni pour le code, ni sur les méthodes de compilation, rien sur le processus de soumission de correctif, ni sur le processus de sortie, rien de rien. Puis, quand on demandera de l'aide, répondez « Lis la notice, bordel ! » (*NdT : RTFM pour Read the fucking manual*)

4. Prenez les décisions relatives au projet en petit comité

Pour bien commencer, vous pouvez organiser vos réunions en ligne en ne prévenant les participants que très peu de temps à l'avance. Pour que cette technique soit vraiment efficace, prévoyez ces réunions à des heures incompatibles avec le fuseau horaire commun à la plupart des membres de la communauté.

Le mieux est encore de tout faire en visioconférence : vous

exclurez de fait environ un tiers de la planète pour qui elle se déroule de nuit, de plus, les gens ont un boulot, tant pis pour eux aussi. Mais le must reste encore d'organiser les réunions en personne au siège de la société.

5. Sortez la grosse artillerie juridique

S'impliquer dans le projet devrait rimer avec accords de participation alambiqués, contrats de licence protégeant le contenu du site Web, accords de confidentialité, marques déposées, etc. Pour ne pas faire les choses à moitié, tous ces documents devraient être modifiés en catimini à peu près tous les deux mois.

6. Choisissez avec soin l'agent de liaison avec la communauté

Votre meilleur candidat : quelqu'un de solitaire, quelqu'un qui n'a pas d'amis et qui n'apprécie pas vraiment les autres. Si vous n'en avez pas sous la main, prenez le membre de votre équipe qui a le plus de travail, quelqu'un avec des responsabilités en développement et en gestion, et qui travaille déjà au minimum 70 heures par semaine. Dans ce cas, il est primordial de ne pas le décharger de la moindre de ses responsabilités.

Quelqu'un qui ne maîtrise pas la technologie fera aussi l'affaire. Prenez un spécialiste de Java pour assurer la liaison dans un projet en Perl. Ou, si vraiment ces solutions ne sont pas possibles, laissez simplement la place vacante pendant des mois.

7. Rendez opaques les prises de décision

Les entreprises réfractaires aux communautés devraient, d'après Josh, s'inspirer des Nations Unies et créer des processus longs et complexes. Personne ne doit savoir qui prend réellement les décisions, c'est très bon pour transformer les contributeurs en éléments nocifs. Il va de soi

que les règles devraient être immuables ou presque.

8. Faites n'importe quoi avec les licences

Les membres de la communauté étant souvent à cheval sur la question des licences, modifiez la licence et vous avez de bonnes chances de les faire fuir. Évoquer des changements de licence sans jamais vraiment rien modifier peut se révéler encore plus efficace : vous faites fuir les contributeurs actuels qui apprécient la licence choisie sans pour autant en attirer d'autres, adeptes eux de la future licence supposée.

9. N'accordez jamais, au grand jamais, l'accès au commit à quelqu'un d'extérieur à l'entreprise

Ce devrait être une règle (tacite évidemment) : seuls les employés peuvent avoir les droits de commit (*NdT : avoir ces droits revient à avoir accès et contrôle sur le code source du dépôt officiel du logiciel*). Vos réponses doivent être évasives, « problèmes légaux, on y travaille » est une bonne carte à jouer. Afin que cette mesure prenne tout son effet, choisissez un employé qui n'écrit pas de code et confiez lui l'accès au commit sur le projet.

10. Réfugiez-vous dans le silence

Laissez les questions sans réponse, ne dites rien. Maîtriser cette technique peut rendre, à elle seule, toutes les autres inutiles. C'est le meilleur moyen de détruire une communauté qui existe.

En conclusion, Josh ajoute que grâce à Sun, il peut témoigner de l'efficacité de toutes ces techniques. Mais Sun est loin d'être la seule entreprise dans cette situation. Un ancien du *X Consortium* a avoué à Josh qu'eux aussi avaient un jour recouru à chacune de ces méthodes. Ces compétences de destruction de communauté sont monnaie courante dans l'industrie du logiciel.

Mais que faire si votre entreprise veut au contraire se bâtir

une communauté ?

Il paraît évident qu'elle devrait alors s'employer à appliquer à l'inverse les méthodes énumérées ci-dessus. Mais d'après Josh, la clé de voûte du système reste la confiance. À l'instar du mariage, développer une communauté peut prendre des années, mais une seule infidélité détruira la confiance qui en constitue le socle. Ainsi, une entreprise peut perdre la moitié de sa communauté en un week-end. Pour ne pas connaître ce triste sort, il faut avoir confiance en la communauté et agir de sorte que cette confiance soit réciproque.

Notes

[1] Crédit photo : [Giuseppe Bognanni](#) (Creative Commons By)