

Des routes et des ponts (8) - ce qui motive les contributeurs

Participer à l'*open source* est souvent une activité bénévole donc non rémunérée mais qui peut parfois devenir chronophage et difficilement compatible avec une autre activité ou un emploi. Nadia Eghbal, dans ce nouveau chapitre nous donne à voir aujourd'hui les trois motivations principales des contributeurs de l'*open source*.

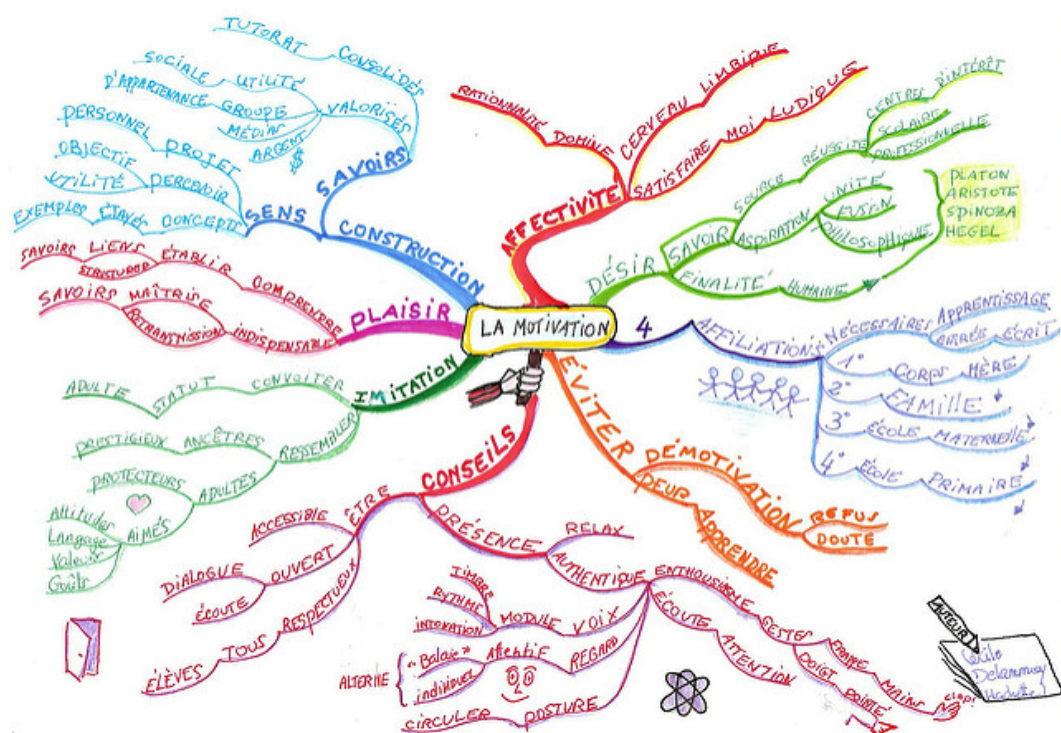


Image de Cécile Delannoy, Photo Philippe Boubovska CC-BY 2.0

Pourquoi les gens continuent-ils de contribuer à ces projets, alors qu'ils ne sont pas payés pour cela ?

Traduction Framalang : Adélie, goofy, xi, woof, Diane, Asta, Rozmador, Lumibd, Piup, teromene, mika, lyn. et deux anonymes

De nombreuses infrastructures numériques sont entretenues par des contributeurs individuels ou par une communauté de contributeurs. Dans la plupart des cas, ces contributeurs ne sont pas directement rémunérés pour ce

travail. En réalité, ils contribuent pour des raisons propres aux communautés *open source*, pour se construire une réputation, par exemple, ou dans un esprit de service public. Ce chapitre explorera certaines de ces motivations plus en détail.

Contribuer à l'open source pour se construire une réputation.

Se construire une réputation est peut-être la motivation la plus concrète pour contribuer à un projet *open source*. Pour les développeurs, rédacteurs techniques et autres, ces projets sont une occasion de faire leurs preuves en public, et leur offrent une chance de participer à quelque chose de grand et d'utile.

Dans le cadre d'un programme appelé *Google Summer of Code* (l'été du code de Google), Google propose des bourses d'été à des étudiants développeurs pour qu'ils contribuent à des projets *open source* populaires. Le programme fonctionne bien, parce que les développeurs sont des étudiants, novices dans le domaine de l'informatique et avides de démontrer leurs talents.

Les développeurs, en particulier, tirent profit de leurs expériences de contribution à l'*open source* pour se constituer un portfolio. De plus, en contribuant à des projets populaires dotés de communautés actives, un développeur a des chances de se construire une réputation en devenant « connu ».

GitHub, site web déjà cité, est une plateforme populaire pour l'élaboration collaborative de code. Quand un développeur contribue à un projet public de logiciel, ses contributions apparaissent sur son profil. Le profil GitHub d'un développeur peut faire office de portfolio pour des sociétés de logiciels, mais seules les contributions effectuées pour des projets publics (c'est-à-dire *open source*) sont visibles par tous.

Cependant, les motivations basées sur la réputation ne sont pas dénuées de risques, surtout parmi les jeunes développeurs. Un développeur dont la carrière est encore naissante peut contribuer à un projet *open source* dans le seul but de trouver un employeur, puis arrêter de contribuer une fois cet objectif atteint. De plus, un développeur cherchant uniquement à se construire un portfolio risque de proposer des contributions de moindre qualité, qui ne seront pas acceptées et qui pourront même ralentir le processus de développement du projet. Enfin, si le but d'une contribution publique de la part d'un développeur est de se construire une réputation, alors ce développeur sera tenté de contribuer uniquement aux projets

les plus populaires et attractifs (une extension du « syndrome de la pie » déjà évoqué), et de ce fait, les projets plus anciens auront du mal à trouver de nouveaux contributeurs.

Le projet est devenu très populaire de manière inattendue, et son développeur se sent obligé d'en assurer le suivi.

Des entreprises, des particuliers ou des organisations peuvent devenir dépendants d'un projet *open source* populaire. En d'autres termes, le code est utilisé dans des logiciels opérationnels, écrits et déployés par d'autres, ces logiciels peuvent servir pour un tas de choses : achats en ligne ou soins médicaux. En raison de la complexité du réseau de dépendances (dont beaucoup ne sont même pas connues de l'auteur du projet, puisqu'il ne peut pas savoir exactement qui utilise son code), la personne qui maintient le projet peut se sentir moralement obligée de continuer à l'entretenir.

Arash Payan, le développeur d'Appirater mentionné au début de ce rapport, a publié son projet en 2009. Au sujet de sa décision de continuer à maintenir le projet, il déclare :

« Ce n'est pas quelque chose de très excitant, mais il y a tellement de gens qui utilisent le projet (qui en dépendent, même ?) pour leurs applications, que je me sens obligé d'en prendre soin correctement. Personnellement, j'ai quitté iOS, donc maintenir une bibliothèque iOS n'est pas exactement la première chose que je veux faire de mon temps libre. »

Payan estime que maintenir le projet à jour lui prend 1 à 2 heures par mois maximum, donc cela ne le dérange pas.

Mais certains projets devenus populaires de façon inattendue prennent plus de temps à maintenir. Andrey Petrov est un développeur indépendant qui a écrit une bibliothèque Python appelée *urllib3*. Sa publication en 2008 fut une avancée majeure pour la bibliothèque standard déjà existante, et elle est devenue populaire parmi les développeurs Python. Aujourd'hui, tous les utilisateurs de Python en dépendent.

Andrey a rendu le projet *open source* dans l'espoir que d'autres gens soutiendraient son développement et sa maintenance. Il est développeur

indépendant ; bien qu'il apprécie de maintenir *urllib3*, il ne peut s'en occuper que pendant son temps libre puisqu'il n'est pas payé pour ce travail. Cory Benfield, qui est employé par la *Hewlett Packard Enterprise* pour aider à maintenir des bibliothèques Python d'importance critique (que HPE utilise et dont HPE dépend), est désormais affecté à la maintenance de *urllib3* dans le cadre de son travail ; cela a allégé la charge de travail d'Andrey.

Le projet est une passion plus qu'un travail

Eric Holsher est l'un des créateurs de *Read the Docs*, une plateforme qui héberge de la documentation sur des logiciels. La documentation est l'équivalent d'un mode d'emploi. De même qu'on a besoin d'un mode d'emploi pour monter un meuble, un développeur a besoin de documentation pour savoir comment implémenter un projet. Sans la documentation adéquate, il serait difficile pour un développeur de savoir par où commencer.

Read the Docs fournit de la documentation pour 18 000 logiciels, y compris pour des entreprises clientes, et compte plus de 15 millions de pages consultées chaque mois.

Bien qu'il génère des revenus grâce à de grosses sociétés clientes, le projet *Read the Docs* est toujours majoritairement financé par les dons de ses utilisateurs. Le coût des serveurs est quant à lui couvert grâce au mécénat de l'entreprise Rackspace.

Eric et son cofondateur, Anthony Johnson, s'occupent de la maintenance du projet et n'en tirent pas de revenus réguliers bien qu'ils s'y consacrent à temps plein. Une subvention ponctuelle de 48 000 \$ de la Fondation Mozilla, reçue en décembre 2015, les aidera à court terme à couvrir leurs salaires. Ils expérimentent actuellement un modèle publicitaire (qui ne piste pas ses utilisateurs) qui rendrait le modèle viable.

Eric remarque que la difficulté ne réside pas uniquement dans le travail de développement, mais aussi dans les fonctions extérieures au code, comme le support client, qui nécessite qu'un mainteneur soit d'astreinte chaque week-end en cas de problème. Pour expliquer pourquoi il continuait de maintenir le projet, Eric l'a qualifié de « travail-passion ».

« Soit les humains sont irrationnels, soit ils ne sont pas intéressés seulement par l'argent. Il y a clairement une autre motivation pour moi dans ce cas. C'est un travail-passion. Si je le voulais, je pourrais clore ce projet demain et ne plus

jamais y revenir, mais je travaille dessus depuis 5 ans et je n'ai aucune envie que ça se termine. »

Eric est motivé pour travailler sur *Read the Docs* parce qu'il perçoit la valeur que cela crée pour les autres. Pour beaucoup de mainteneurs de projets, cet impact sur autrui est la première motivation, parce qu'ils peuvent directement constater l'effet positif que leur travail a sur la vie d'autres personnes. En ce sens, l'*open source* a beaucoup de points communs avec le secteur à but non-lucratif. Cependant, tout comme dans le secteur à but non-lucratif, cette mentalité de « travail-passion » peut augmenter la difficulté qu'ont les communautés *open source* à aborder le sujet qui fâche : comment pérenniser des projets qui nécessitent davantage de ressources et d'attention que ce que les contributeurs actuels peuvent fournir ?