Des routes et des ponts (13) — Des mécènes pour les projets open source

Chaque semaine, l'équipe Framalang vous propose la traduction d'un chapitre de Roads and Bridges de Nadia Eghbal, une enquête fouillée qui explore les problématiques des infrastructures numériques, et en particulier leur intrication avec l'écosystème open source.

Après avoir exploré dans le précédent chapitre différents types de modèles économiques adaptés aux projets open source (retrouvez ici tous les chapitres antérieurs), l'auteure examine ici les cas de projets s'appuyant sur les dons ou le mécénat : du financement participatif au soutien institutionnalisé d'une entreprise, elle analyse les avantages et les limites de chaque solution, et livre les témoignages de nombreux porteurs de projets ou contributeurs qui relatent leur expérience au cœur de projets aussi divers qu'OpenSSL, jQuery ou encore Node.js.

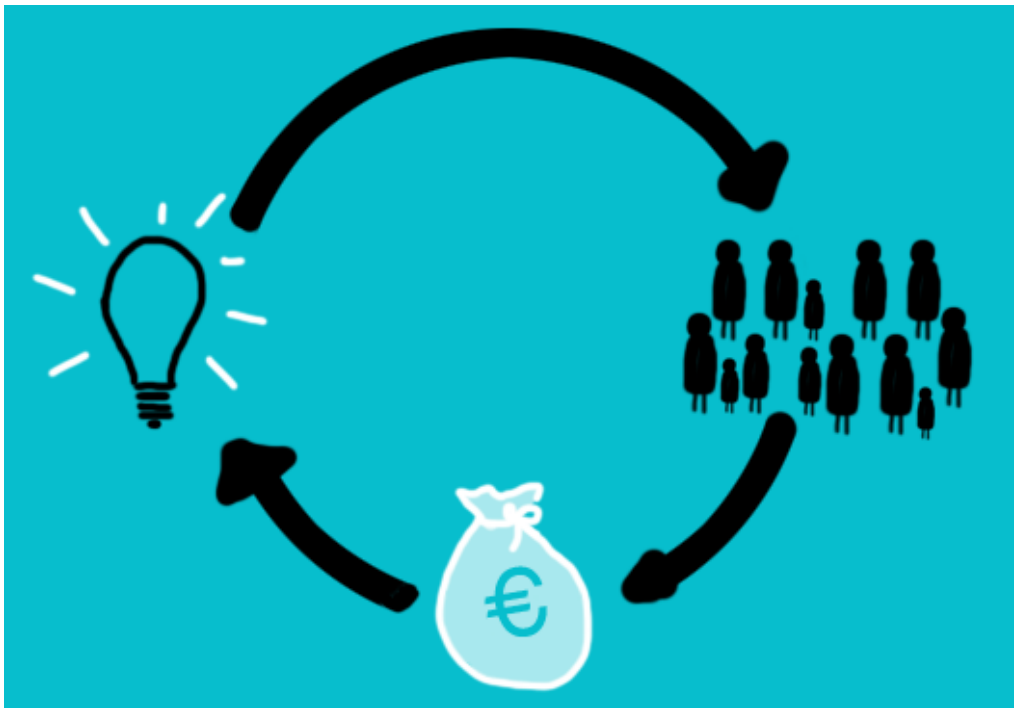

Image de Rocío Lara (CC BY-SA 2.0)

Trouver des mécènes ou des donateurs pour financer un projet d'infrastructure

Traduction Framalang : goofy, dominix, Opsylac, Rozmador, lyn, Julien, Penguin, Luc, serici, pasquin, et 2 anonymes

La deuxième option pour financer des projets d'infrastructure numérique consiste à trouver des mécènes ou des donateurs. Il s'agit d'une pratique courante dans les cas de figure suivants :

- Il n'existe pas de demande client facturable pour les services proposés par le projet.
- Rendre l'accès payant empêcherait l'adoption (on ne pourrait pas, par exemple, faire payer l'utilisation d'un langage de programmation comme Python, car personne ne l'utiliserait ; ce serait comme si parler anglais étant payant).
- Le projet n'a pas les moyens de financer des emplois rémunérés, ou bien il n'y a pas de volonté de la part du développeur de s'occuper des questions commerciales.
- La neutralité et le refus de la commercialisation sont considérés comme des principes importants en termes de gouvernance.

Dans ce type de situation, un porteur de projet cherchera des mécènes qui croient en la valeur de son travail et qui sont disposés à le soutenir financièrement. À l'heure actuelle, il existe deux sources principales de financement : les entreprises de logiciel et les autres développeurs.

Le financement participatif

Certains travaux de développement obtiennent des fonds grâce à des campagnes de financement participatif (« crowdfunding ») via des plateformes telles que Kickstarter ou Indiegogo. Bountysource, le site de récompenses dont nous parlions dans un chapitre précédent, possède également une plateforme appelée Salt dédiée au financement participatif de projets open source.

Andrew Godwin, un développeur du noyau Django résidant à Londres, a ainsi réussi à récolter sur Kickstarter 17952£ (environ 21000€) de la part de 507 contributeurs, afin de financer des travaux de base de données pour Django. Le projet a été entièrement financé en moins de quatre heures.

Pour expliquer sa décision de lever des fonds pour un projet open source, Godwin écrit :

> *« Une quantité importante de code open source est écrit gratuitement. Cependant, mon temps libre est limité. Je dispose actuellement d'une seule journée libre par semaine pour travailler, et j'adorerais la consacrer à l'amélioration de Django, plutôt qu'à du conseil ou à de la sous-traitance.*
>
> *L'objectif est double : d'une part, garantir au projet un temps de travail conséquent et au moins 80 heures environ de temps de codage ; et d'autre part prouver au monde que les logiciels open source peuvent réellement rémunérer le temps de travail des développeurs. »*

À l'instar des récompenses, le financement participatif s'avère utile pour financer de nouvelles fonctionnalités, ou des développements aboutissant à un résultat clair et tangible. Par ailleurs, le financement participatif a moins d'effets pervers que les récompenses, notamment parce qu'organiser une campagne de financement demande plus d'efforts que de poster une offre de récompense, et parce que

le succès du financement repose en grande partie sur la confiance qu'a le public dans la capacité du porteur de projet à réaliser le travail annoncé. Dans le cas de Godwin, il était l'un des principaux contributeurs au projet Django depuis six ans et était largement reconnu dans la communauté.

Toutefois, le financement participatif ne répond pas à la nécessité de financer les frais de fonctionnement et les frais généraux. Ce n'est pas une source de capital régulière. En outre, planifier et mettre en œuvre une campagne de financement participatif demande à chaque fois un investissement important en temps et en énergie. Enfin, les donateurs pour ces projets sont souvent eux-mêmes des développeurs ou des petites entreprises – et un porteur de projet ne peut pas éternellement aller toquer à la même porte pour financer ses projets.

Avec le recul, Godwin a commenté sa propre expérience :

> *« Je ne suis pas sûr que le financement participatif soit totalement compatible avec le développement open source en général ; non seulement c'est un apport ponctuel, mais en plus l'idée de rétribution est souvent inadéquate car elle nécessite de promettre quelque chose que l'on puisse garantir et décrire a priori.*
>
> *S'en remettre uniquement à la bonne volonté du public, cela ne fonctionnera pas. On risque de finir par s'appuyer de manière disproportionnée sur des développeurs, indépendants ou non, à un niveau personnel – et je ne pense pas que ce soit viable. »*

À côté des campagnes de financement participatif, plusieurs plateformes ont émergé pour encourager la pratique du « pourboire » (*tipping* en anglais) pour les contributeurs *open source* : cela consiste à verser une petite somme de revenu régulier à un contributeur, en signe de soutien à son travail. Deux plateformes populaires se distinguent : Patreon (qui ne

se limite pas exclusivement aux contributeurs open source) et Gratipay (qui tend à fédérer une communauté plus technique).

L'idée d'un revenu régulier est alléchante, mais souffre de certains problèmes communs avec le financement participatif. On remarque notamment que les parrains (*patrons* ou *tippers* en anglais) sont souvent eux-mêmes des développeurs, avec une quantité limitée de capital à se promettre les uns aux autres. Les dons ont généralement la réputation de pouvoir financer une bière, mais pas un loyer. Gratipay rassemble 122 équipes sur sa plateforme, qui reçoivent collectivement 1000 $ par semaine, ce qui signifie qu'un projet touche en moyenne moins de 40$ par mois.

Même les très gros projets tels que OpenSSL ne généraient que 2000$ de dons annuels avant la faille Heartbleed. Comme expliqué précédemment, après Heartbleed, Steve Marquess, membre de l'équipe, a remarqué « un déferlement de soutien de la part de la base de la communauté OpenSSL » : la première vague de dons a rassemblé environ 200 donateurs pour un total de 9000$.

Marquess a remercié la communauté pour son soutien mais a également ajouté :

> *« Même si ces donations continuent à arriver au même rythme indéfiniment (ce ne sera pas le cas), et même si chaque centime de ces dons allait directement aux membres de l'équipe OpenSSL, nous serions encore loin de ce qu'il faudrait pour financer correctement le niveau de main-d'œuvre humaine nécessaire à la maintenance d'un projet aussi complexe et aussi crucial. Même s'il est vrai que le projet « appartient au peuple », il ne serait ni réaliste ni correct d'attendre de quelques centaines, ou même de quelques milliers d'individus seulement, qu'ils le financent à eux seuls. Ceux qui devraient apporter les vraies ressources, ce sont les entreprises lucratives et les gouvernements qui utilisent OpenSSL massivement et qui le considèrent comme un*

acquis. »

(À l'appui de l'argument de Marquess, les dons de la part des entreprises furent par la suite plus importants, les sociétés ayant davantage à donner que les particuliers. La plus grosse donation provint d'un fabricant de téléphone chinois, Smartisan, pour un montant de 160000$. Depuis, Smartisan a continué de faire des dons substantiels au projet OpenSSL.)

Au bout du compte, la réalité est la suivante : il y a trop de projets, tous qualitatifs ou cruciaux à leur manière, et pas assez de donateurs, pour que la communauté technique (entreprises ou individus) soit en mesure de prêter attention et de contribuer significativement à chacun d'eux.

Le mécénat d'entreprises pour les projets d'infrastructure

À plus grande échelle, dans certains cas, la valeur d'un projet devient si largement reconnue qu'une entreprise finit par recruter un contributeur pour travailler à plein temps à son développement.

John Resig est l'auteur de jQuery, une bibliothèque de programmation JavaScript qui est utilisée par près des 2/3 du million de sites web les plus visités au monde. John Resig a développé et publié jQuery en 2006, sous la forme d'un projet personnel. Il a rejoint Mozilla en 2007 en tant que développeur évangéliste, se spécialisant notamment dans les bibliothèques JavaScript.

La popularité de jQuery allant croissante, il est devenu clair qu'en plus des aspects liés au développement technique, il allait falloir formaliser certains aspects liés à la gouvernance du projet. Mozilla a alors proposé à John de travailler à plein temps sur jQuery entre 2009 et 2011, ce qu'il a fait.

À propos de cette expérience, John Resig a écrit :

> *« Pendant l'année et demi qui vient de s'écouler, Mozilla m'a permis de travailler à plein temps sur jQuery. Cela a abouti à la publication de 9 versions de jQuery… et à une amélioration drastique de l'organisation du projet (nous appartenons désormais à l'organisation à but non lucratif Software Freedom Conservancy, nous avons des réunions d'équipe régulières, des votes publics, fournissons des états des lieux publics et encourageons activement la participation au projet). Heureusement, le projet jQuery se poursuit sans encombre à l'heure actuelle, ce qui me permet de réduire mon implication à un niveau plus raisonnable et de participer à d'autres travaux de développement. »*

Après avoir passé du temps chez Mozilla pour donner à jQuery le support organisationnel dont il avait besoin, John a annoncé qu'il rejoindrait la Khan Academy afin de se concentrer sur de nouveaux projets.

Cory Benfield, développeur Python, a suivi un chemin similaire. Après avoir contribué à plusieurs projets open source sur son temps libre, il est devenu un développeur-clé pour une bibliothèque essentielle de Python intitulée Requests. Cory Benfield note que :

> *« Cette bibliothèque a une importance comparable à celle de Django, dans la mesure où les deux sont des « infrastructures critiques » pour les développeurs Python. Et pourtant, avant que j'arrive sur le projet, elle était essentiellement maintenue par une seule personne. »*

Benfield estime qu'il a travaillé bénévolement sur le projet environ 12 heures par semaine pendant presque quatre ans, en plus de son travail à plein temps. Personne n'était payé pour travailler sur Requests.

Pendant ce temps, HP embauchait un employé, Donald Stufft, pour se consacrer spécifiquement aux projets en rapport avec Python, un langage qu'il considère comme indispensable à ses logiciels. (Donald est le développeur cité précédemment qui est payé à plein temps pour travailler sur le packaging Python). Donald a alors convaincu son supérieur d'embaucher Cory pour qu'il travaille à temps plein sur des projets Python. Il y travaille toujours.

Les entreprises sont des acteurs tout désignés pour soutenir financièrement les projets bénévoles qu'elles considèrent comme indispensables à leurs activités, et quand des cas comme ceux de John Resig ou de Cory Benfield surviennent, ils sont chaleureusement accueillis. Cependant, il y a des complications.

Premièrement, aucune entreprise n'est obligée d'embaucher quelqu'un pour travailler sur des projets en demande de soutien ; ces embauches ont tendance à advenir par hasard de la part de mécènes bienveillants. Et même une fois qu'un employé est embauché, il y a toujours la possibilité de perdre ce financement, notamment parce que l'employé ne contribue pas directement au résultat net de l'entreprise. Une telle situation est particulièrement périlleuse si la viabilité d'un projet dépend entièrement d'un seul contributeur employé à plein temps. Dans le cas de Requests, Cory est le seul contributeur à plein temps (on compte deux autres contributeurs à temps partiel, Ian Cordasco et Kenneth Reitz).

Une telle situation s'est déjà produite dans le cas de « rvm », un composant critique de l'infrastructure Ruby. Michal Papis, son auteur principal, a été engagé par Engine Yard entre 2011 et 2013 pour soutenir le développement de rvm. Mais quand ce parrainage s'est terminé, Papis a dû lancer une campagne de financement participatif pour continuer de financer le développement de rvm.

Le problème, c'est que cela ne concernait pas seulement rvm.

Engine Yard avait embauché plusieurs mainteneurs de projets d'infrastructure Ruby, qui travaillaient notamment sur JRuby, Ruby on Rails 3 et bundler. Quand les responsables d'Engine Yard ont été obligés de faire le choix réaliste qui s'imposait pour la viabilité de leur entreprise, c'est-à-dire réduire leur soutien financier, tous ces projets ont perdu leurs mainteneurs à temps plein, et presque tous en même temps.

L'une des autres craintes est qu'une entreprise unique finisse par avoir une influence disproportionnée sur un projet, puisqu'elle en est *de facto* le seul mécène. Cory Benfield note également que le contributeur ou la contributrice lui-même peut avoir une influence disproportionnée sur le projet, puisqu'il ou elle dispose de beaucoup plus de temps que les autres pour faire des contributions. De fait, une telle décision peut même être prise par une entreprise et un mainteneur, sans consulter le reste de la communauté du projet.

On peut en voir un exemple avec le cas d'Express.js, un framework important pour l'écosystème Node.js. Quand l'auteur du projet a décidé de passer à autre chose, il en a transféré les actifs (en particulier le dépôt du code source et le nom de domaine) à une société appelée StrongLoop dont les employés avaient accepté de continuer à maintenir le projet. Cependant StrongLoop n'a pas fourni le soutien qu'attendait la communauté, et comme les employés de StrongLoop étaient les seuls à avoir un accès administrateur, il est devenu difficile pour la communauté de faire des contributions. Doug Wilson, l'un des principaux mainteneurs (non-affilié à StrongLoop), disposait encore d'un accès commit et a continué de traiter la charge de travail du projet, essayant tant bien que mal de tout gérer à lui seul.

Après l'acquisition de StrongLoop par IBM, Doug déclara que StrongLoop avait bel et bien tué la communauté des contributeurs.

> *« Au moment où on est passé à StrongLoop, il y avait des membres actifs comme @Fishrock123 qui travaillaient à créer… de la documentation. Et puis tout à coup, je me suis retrouvé tout seul à faire ça sur mon temps libre alors que les demandes de support ne faisaient que se multiplier… et pendant tout ce temps, je me suis tué à la tâche, je me suis engagé pour le compte StrongLoop. Quoi qu'il arrive, jamais plus je ne contribuerai à aucun dépôt logiciel appartenant à StrongLoop. »*

Finalement, le projet Express.js a été transféré de StrongLoop à la fondation Node.js, qui aide à piloter des projets appartenant à l'écosystème technologique Node.js.

En revanche, pour les projets open source qui ont davantage d'ampleur et de notoriété, il n'est pas rare d'embaucher des développeurs. La Fondation Linux a fait savoir, par exemple, que 80% du développement du noyau Linux est effectué par des développeurs rémunérés pour leur travail. La fondation Linux emploie également des Fellows [« compagnons » selon un titre consacré, NdT] payés pour travailler à plein temps sur les projets d'infrastructure, notamment Greg Kroah-Hartman, un développeur du noyau Linux, et Linus Torvalds lui-même, le créateur de Linux.