



Nadia Eghbal

Sur quoi reposent nos infrastructures numériques ? Le travail invisible des faiseurs du web

OpenEdition Press

Une esquisse du tableau

DOI : 10.4000/books.oep.1823
Éditeur : OpenEdition Press, Framabook
Lieu d'édition : OpenEdition Press,
Framabook
Année d'édition : 2017
Collection : Encyclopédie numérique
ISBN électronique : 9782821894938



<http://books.openedition.org>

Référence électronique

EGHBAL, Nadia. *Une esquisse du tableau* In : *Sur quoi reposent nos infrastructures numériques ? Le travail invisible des faiseurs du web* [en ligne]. Marseille : OpenEdition Press, 2017 (généré le 26 octobre 2017). Disponible sur Internet : <<http://books.openedition.org/oep/1823>>. ISBN : 9782821894938. DOI : 10.4000/books.oep.1823.

UNE ESQUISSE DU TABLEAU

Il est trop tôt pour dire à quoi devrait ressembler le soutien institutionnel à long terme d'un point de vue prospectif, mais il y a plusieurs domaines de travail critiques qui peuvent nous aider à le déterminer. Les propositions suivantes se rattachent à trois domaines :

- **traiter les infrastructures numériques comme un bien commun essentiel** et les élever au rang d'acteur intersectoriel clé ;
- **travailler avec des projets** pour améliorer les standards, la sécurité et les flux de production ;
- **augmenter la taille du groupe de contributeurs** de manière à ce que davantage de personnes, et encore plus de personnes de types différents, puissent élaborer et soutenir ensemble les logiciels publics.

Conscientiser et éduquer les acteurs clés

Comme nous l'avons relevé dans cette étude, beaucoup d'acteurs clés – dont les *startups*, les gouvernements, et les sociétés de capital-risque – pensent à tort que les logiciels publics « fonctionnent, tout simplement » et ne requièrent pas de maintenance supplémentaire. Pour entretenir correctement l'écosystème de nos infrastructures numériques, ces populations devraient être les premières à être informées du problème. Les infrastructures numériques ont besoin de porte-parole qui soient affranchis de toute contrainte politique ou commerciale et qui puissent comprendre et communiquer les besoins de l'écosystème.

Traiter les infrastructures numériques comme des biens communs essentiels pourrait également motiver l'investissement direct dans la construction de meilleurs systèmes en

partant de zéro. Par exemple, aux États-Unis, les autoroutes inter-États et le réseau de bibliothèques publiques furent dès l'origine conçus comme des ressources collectives. Les unes et les autres ont eu leur champion (respectivement le président Dwight D. Eisenhower et le philanthrope Andrew Carnegie) qui ont clairement argumenté en faveur du bénéfice social et financier qui résulterait de tels projets.

Un réseau national d'autoroutes ne sert pas uniquement à nous relier en tant qu'individus en facilitant les déplacements d'un endroit à un autre, mais il apporte aussi la prospérité financière dans tous les coins du pays, grâce à l'usage commercial des voies rapides pour acheminer les marchandises. Dans les bibliothèques Andrew Carnegie, publiques et gratuites, les livres étaient accessibles et non stockés en magasin, pour permettre aux personnes de les feuilleter et d'y trouver elles-mêmes l'information sans avoir recours à un(e) bibliothécaire. Cette pratique a aidé à démocratiser l'information et à donner aux gens les moyens de s'éduquer eux-mêmes.

Une meilleure éducation et une meilleure prise de conscience pourraient s'étendre jusqu'aux gouvernements, dont certains ont rendu, par la loi, les infrastructures numériques difficiles à soutenir et qui ne sont peut-être pas familiers des normes culturelles et de l'histoire de l'*open source*. Aux États-Unis, l'IRS (Internal Revenue Service, service des impôts américain) a une définition très restrictive des activités caritatives, et comme l'*open source* est mal comprise, son impact positif sur la société demeure ignoré¹. Cela complique l'institutionnalisation de plus gros projets à travers une fondation ou une association professionnelle.

Mesurer l'utilisation et l'impact de l'infrastructure numérique

L'impact de l'infrastructure numérique est encore très difficile à mesurer. Les indicateurs habituels sont soit très imprécis, soit simplement indisponibles. Ce n'est pas un problème facile à

1. Voir partie 4, chapitre 4.

résoudre. Mais sans données relatives aux outils utilisés et à notre dépendance vis-à-vis d'eux, il est difficile de se faire une idée précise de l'état des sous-financements.

Avec de meilleurs indicateurs, nous pourrions décrire l'impact économique de l'infrastructure numérique, identifier les projets essentiels qui manquent de financement et comprendre les dépendances entre les projets et les personnes. Pour le moment, il est impossible de dire qui utilise un projet *open source* à moins que l'utilisateur, individu ou entreprise, ne le révèle. Pour déterminer quel projet a besoin de plus de soutien, nous ne disposons que d'informations anecdotiques.

De meilleures statistiques pourraient également nous aider à identifier les « contributeurs clés de voûte ». En biologie environnementale, une « espèce clé » est une espèce animale qui a un impact disproportionné sur son environnement au regard de ses effectifs². Dans la même idée, un « contributeur clé » pourrait être un développeur qui contribue à plusieurs projets essentiels, le seul responsable d'un projet crucial, ou généralement perçu comme influent et digne de confiance. Les « contributeurs clés » sont des défenseurs essentiels, les valoriser en leur fournissant les ressources dont ils ont besoin pourrait améliorer le système dans son ensemble. Comprendre les relations entre les communautés *open source* et les « contributeurs clés » pourrait aider à identifier rapidement les secteurs qui auront besoin de soutien supplémentaire.

On dispose également de peu de données sur les contributeurs eux-mêmes : qui contribue à l'*open source*, quelles conditions leur permettent de le faire, et quelles sont les contributions effectuées. Les femmes, les non-anglophones et les nouveaux contributeurs à l'*open source* sont des exemples de population qui devraient être suivis dans le temps, en particulier pour mesurer l'impact des programmes de soutien.

Les seules statistiques disponibles sur les dépôts GitHub sont le nombre de personnes ayant *étoilé* (action semblable à *liker*), vu (c'est-à-dire qu'elles reçoivent des nouvelles du projet) ou « *forké* » un projet. Ces chiffres permettent de fournir

2. Pour en savoir plus sur les espèces clés, voir l'article « Espèce clé de voûte » sur Wikipédia.

des indicateurs concernant la popularité, mais ils peuvent être trompeurs. Beaucoup de personnes peuvent étoiler un projet, parce qu'il a une conception intéressante par exemple, sans toutefois l'intégrer à leur propre code.

Certains gestionnaires de paquets tel npm (qui est celui de Node.js) suivent les téléchargements. Le « *popularity contest* » de Debian piste les téléchargements du système d'exploitation libre Debian. Néanmoins, chaque gestionnaire de paquets est limité à un écosystème particulier, et aucun de ces gestionnaires ne peut donner une image du système dans son ensemble. Plusieurs projets ne sont pas inclus dans un gestionnaire de paquets et ne sont pas suivis. Libraries.io, un site web créé par Andrew Nesbitt, est une tentative pour agréger des données des projets *open source* en fonction de leur usage, il piste environ 1,3 million de bibliothèques *open source* sur 32 gestionnaires de paquets³.

Travailler avec les projets pour moderniser l'organisation de travail

Beaucoup de projets peinent et pas seulement à cause d'un manque de financement, mais aussi parce qu'il est difficile d'y contribuer, ou encore parce qu'il existe un goulot d'étranglement au niveau des mainteneurs qui traitent les demandes de modification (*pull requests*) de la communauté. C'est vrai, en particulier, pour les plus anciens projets qui ont été bâtis avec des outils de développement, des langages et des processus qui ne sont plus aussi répandus (ceux qui par exemple utilisent un système de contrôle de version autre que Git, dont la popularité croît chez les développeurs).

On peut faire beaucoup de choses pour faciliter la contribution à un projet, depuis la migration vers un flux de travail (*workflow*) plus moderne, le nettoyage du code, la fermeture des *pull requests* délaissées, jusqu'à la mise en place d'une politique claire pour les contributions. Certains projets ont essayé

3. Ces chiffres relevés sur le site Libraries.io, sont constamment mis à jour. En janvier 2017, on compte environ 2,1 millions de bibliothèques *open source* recensées sur 33 gestionnaires de paquets.

de rendre les contributions plus simples. Le développeur Felix Geisendörfer, par exemple, a suggéré que chaque personne qui soumet une modification du code devrait avoir une permission de *commit* afin de réduire l'engorgement au niveau de l'unique mainteneur vérifiant et approuvant ces changements. Felix a estimé que « cette approche est un moyen fantastique d'éviter que le projet ne se ratatine en transformant le projet d'un seul homme en celui d'une communauté »⁴.

Le règlement de contribution de Node.js, qui peut être adopté par les autres projets Node, met l'accent sur l'augmentation du nombre de contributeurs et sur leur autonomisation dans la prise de décision, plutôt que de désigner les mainteneurs comme seule autorité approbatrice. Leurs règles de contribution expliquent comment soumettre et valider des *pull requests*, comment consigner des bugs, etc. Les mainteneurs Node.js ont constaté qu'adopter de meilleures règles les avait aidés à gérer leur charge de travail et à faire évoluer leur communauté vers un projet plus sain et actif⁵. Des études sont à réaliser pour déterminer quels projets doivent avancer en priorité. Autrement dit, définir à quoi ressemble un « projet à succès », aussi bien en termes de financement et de modèles de gouvernance, que dans l'équilibre à trouver entre mainteneurs, contributeurs et usagers ! La réponse peut varier en fonction des différents types de projets et de leur ampleur.

Encourager les standards communs dans les projets *open source*

Bien que GitHub soit en train de devenir une plateforme standard pour la collaboration sur le code, de nombreux aspects des projets *open source* ne sont pas encore standardisés, notamment l'ampleur et la richesse de la documentation, des licences et des guides de contribution, ainsi que le style de code et le formatage.

4. Voir Felix Geisendörfer, « The Pull Request Hack », sur felixge.de (site personnel), 11/03/2013.

5. Voir Mikeal, « Healthy Open Source », Medium.com, 22/02/2016.

Encourager l'adoption de standards de projets pourrait faciliter, pour les mainteneurs, la gestion des contributions, tout en réduisant pour les contributeurs les obstacles à la participation.

Parmi les exemples de standardisation croissante, on trouve le code de conduite, qui est un règlement détaillant les attentes en termes d'attitude et de communication.

Ces dernières années, des codes de conduite ont été adoptés par un nombre croissant de communautés de projets, notamment Node.js, Django et Ruby. Bien que le processus d'adoption ait pu donner lieu à d'intenses débats au sein de certaines communautés, leur prolifération révèle un intérêt croissant pour la responsabilisation du comportement des communautés.

Augmenter le nombre de contributeurs et contributrices *open source*

Comme nous l'avons évoqué dans un chapitre précédent de cette étude, l'industrie du logiciel est florissante, avec un nombre croissant de nouveaux développeurs mais aussi d'autres talents variés : il y a du travail à faire pour encourager ces nouveaux arrivants à contribuer à l'*open source*. Augmenter le nombre de contributeurs permet aux projets *open source* d'être plus durables, car davantage de personnes participent à leur développement. Permettre à encore plus de personnes de contribuer à l'*open source* accroît également l'empathie et la communication entre les « utilisateurs » de l'*open source* et les projets dont ils dépendent.

Your First PR⁶ est un exemple d'initiative, développée par la programmeuse Charlotte Spencer, qui aide les nouveaux venus à effectuer leur première contribution à l'*open source*. First Timers Only⁷ (Réservé aux débutants) et Make a Pull Request⁸ (Faites une *pull request*) sont deux autres exemples de ressources populaires qui initient les néophytes à l'*open source*. Certains projets *open source* utilisent également des étiquettes telles que *first*

6. Voir le compte Twitter Your First PR@yourfirstpr. « Votre première PR », PR pour *pull request*, (NdT).

7. Voir Firsttimersonly.com.

8. Voir Makeapullrequest.com.

bug ou *contributor friendly* pour signaler les problèmes susceptibles d'être résolus par des contributeurs moins expérimentés. Il serait également bénéfique d'encourager les contributions à l'*open source* autres que le code, comme la rédaction de documentation technique, la gestion des tâches et des flux de travail, ou la création d'un site internet pour le projet.

En plus d'accroître la proportion de techniciens talentueux contribuant à l'*open source*, se présente l'opportunité de puiser dans un vivier plus large de contributeurs. Faire en sorte que les non-anglophones se sentent bienvenus dans les communautés *open source*, par exemple, pourrait rendre la technologie plus accessible à travers le monde. Et comme beaucoup de recruteurs utilisent les travaux *open source* comme un portfolio au moment d'embaucher un développeur, une communauté *open source* plus diverse encouragerait l'apparition d'un personnel technique globalement plus intégré.

Améliorer les relations entre projets et acteurs extérieurs

Les entreprises sont un élément incontournable de l'écosystème *open source*, et leur rôle ne fait que gagner en importance à mesure qu'un plus grand nombre d'entre elles adoptent les logiciels *open source*. Faciliter la collaboration entre entreprises et projets, ainsi qu'aider les entreprises à comprendre les besoins des communautés *open source*, pourrait débloquer le soutien des entreprises susceptibles de devenir mécènes ou promoteurs de l'*open source*.

Selon l'étude annuelle des entreprises *open source* réalisée par Black Duck⁹, seulement 27 % d'entre elles ont un règlement formel concernant les contributions de leurs employés à l'*open source*. Clarifier la possibilité ou non pour les employés de contribuer à l'*open source* sur leur temps de travail, et les encourager à le faire, pourrait grandement améliorer le soutien des entreprises aux projets *open source*.

9. Voir le rapport de la société BlackDuck *The Ninth Annual Future of Open Source Survey*, sur blackducksoftware.com, 2015.

En 2014, un groupement d'entreprises a fondé le TODO Group¹⁰, pour partager les bonnes pratiques autour de la participation corporative à l'*open source*. Parmi les membres de ce groupe, on trouve Box, Facebook, Dropbox, Twitter et Stripe. En mars 2016, le TODO Group a annoncé qu'il serait hébergé par la Fondation Linux en tant que projet collaboratif¹¹.

Les entreprises peuvent également fournir un soutien financier aux projets, mais il est parfois difficile pour elles de trouver comment formaliser leur mécénat. Créer des budgets dédiés au sponsoring en direction des équipes d'ingénieurs ou des employés, ou encore créer des documents permettant aux projets de « facturer » plus facilement leurs services aux entreprises, sont autant d'initiatives qui pourraient augmenter les contributions financières à l'*open source*.

Poul-Henning Kamp, par exemple, travaille sur un projet *open source* nommé Varnish¹², utilisé par un dixième des sites les plus visités d'Internet, notamment Facebook, Twitter, Tumblr, *The New York Times* et *The Guardian*. Pour financer ce travail, il a créé la Varnish Moral License pour faciliter la subvention du projet par les entreprises.

Même si en pratique la relation est un mécénat, Poul-Henning utilise une terminologie familière aux entreprises, avec des termes tels que « facturation » et « licences », pour réduire les obstacles à la participation¹³.

Augmenter le soutien aux compétences diverses et aux fonctions hors codage

Dans un passé pas si lointain, les *startups* de logiciels étaient fortement centrées sur les compétences techniques. Les autres rôles, comme le marketing et le design, étaient considérés comme secondaires par rapport au code. Aujourd'hui, avec la création et la consommation rapide de logiciels, cette

10. Voir Todogroup.org.

11. Voir TheTodoGroup, « TODO Becomes A Linux Foundation Collaborative Project », sur le site Todo.org, 30/03/2016.

12. Pour en savoir plus sur Varnish, voir l'article « Varnish » sur Wikipédia.

13. Voir Poul-Henning Kamp, « The Varnish Moral License », sur la page personnelle de l'auteur, 30/06/2015.

conception ne tient plus. Les *startups* sont en concurrence pour capter l'attention de leurs clients. L'identité de la marque est devenue l'un des principaux facteurs de différenciation.

Ces cinq dernières années ont été celles de l'essor du développeur *full stack* (polyvalent) : des développeurs plus généralistes que spécialisés, capables de travailler sur plusieurs domaines d'un logiciel complexe, et qui sont susceptibles d'avoir des compétences dans la conception et la production. Les équipes de développement sont plus soudées, elles utilisent les méthodes agiles avec des approches de conception d'architecture logicielle (où le livrable est élaboré en faisant des navettes entre les équipes de techniciens, designers et commerciaux), plutôt qu'une approche en cascade (où chaque équipe apporte sa pièce à l'édifice avant de la transmettre au groupe de travail suivant).

Les logiciels *open source* ont connu peu d'évolutions de ce genre, malgré notre dépendance croissante vis-à-vis d'eux. On comprend aisément que le code soit au cœur d'un projet *open source*, il est en quelque sorte le « produit final » ou le livrable. Les fonctions relatives à la gestion de la communauté, à la documentation ou à la promotion du projet, qui sont la marque d'une organisation saine et durable, sont moins valorisées. Il en découle que les projets sont déséquilibrés. Beaucoup de choses pourraient être entreprises pour financer et soutenir les contributions autres que le code, des dons en nature pour payer les serveurs par exemple, ou des avantages comme une assurance maladie. Disposer de soutiens de ce type permettrait de réduire notablement la charge des développeurs.