

7 qualités de l'Open Source, entre mythes et réalités

« Nous avons tous intérêt à défendre l'Open Source contre les forces qui l'affaiblissent », nous affirme ici Matthew Butterick, juriste et passionné de typographie^[1].



Constatant que de nombreux projets se disent open source alors qu'ils n'en possèdent pas les caractéristiques, il nous propose ici de le définir en sept points, distinguant la réalité de ce qu'il appelle sa *dilution*.

Vous ne serez pas forcément d'accord, surtout si vous êtes plus « logiciel libre » (version canal historique Richard Stallman) que « Open Source ». Mais on devrait pouvoir en discuter sereinement dans les commentaires ☐

Sept qualités essentielles à l'Open Source

Seven essential qualities of open source

*Matthew Butterick – Janvier 2012 – TypographyForLawyers.com
(Traduction Framalang : Goofy, OranginaRouge, antistress)*

Les citations tronquées déforment les propos. Beaucoup connaissent la fameuse citation de Stewart Brand selon laquelle « l'information veut être gratuite » (NdT : « *information wants to be free.* »). Cette phrase, prise isolément, est souvent citée à l'appui de thèses selon lesquelles toute contrainte sur l'information numérique est futile ou immorale.

Mais peu ont entendu la deuxième partie du propos, rarement citée, qui prend le contrepied de la première partie : « l'information veut être hautement valorisée » (*NdT* : « *information wants to be expensive.* »). Lorsque l'on reconstitue le propos dans son entier, il apparaît que Brand veut illustrer la tension centrale de l'économie de l'information. Lorsque le propos est tronqué, sa signification en est changée.

Les citations tronquées ont aussi déformé le sens de « Open Source ». Au fur et à mesure que le monde de l'Open Source prenait de l'ampleur, il a attiré davantage de participants, depuis les programmeurs individuels jusqu'aux grandes entreprises. C'était prévisible. Tous ces participants ne sont pas d'accord sur le sens donné à l'Open Source. Ça aussi c'était prévisible. Certains participants influents ont tenté de diluer l'idée de l'Open Source, usant d'un raccourci réducteur pour décrire une démarche méthodique. Ça aussi, on pouvait s'y attendre.

Mais même si cela était à prévoir, il est toujours payant de combattre cette dilution. Sur le long terme, cela est néfaste à l'Open Source. Et pas seulement pour des raisons éthiques ou morales mais aussi pour des raisons pratiques et de viabilité. Si on laisse l'Open Source se diluer en acceptant de revoir à la baisse les attentes, ceux d'entre nous qui jouissent des avantages de l'Open Source seront perdants à terme. De la même façon, ceux qui contribuent à des projets pseudo-Open Source risquent de dépenser leur énergie au profit de causes douteuses. C'est pourquoi nous avons tous intérêt à défendre l'Open Source contre les forces qui l'affaiblissent.

Dans le même esprit, voici sept qualités que je considère comme essentielles pour définir l'identité de l'Open Source, par opposition aux formes diluées que revêt généralement ce mouvement. L'objectif de cet exercice n'est pas d'offrir une caractérisation univoque de l'Open Source. Ce serait à la fois présomptueux et impossible. L'Open Source est hétérogène par essence, comme tout ce qui apparaît sur la place publique.

Pourtant, le fait que nous puissions parler de l'Open Source en général signifie qu'il doit y avoir des caractéristiques irréductibles. Des gens raisonnables, par exemple, peuvent avoir des conceptions différentes de l'art. Mais qui irait contester la qualité de Mona Lisa ? De la même façon, s'il n'est pas raisonnable de parler de l'Open Source comme d'un bloc monolithique, il est tout aussi déraisonnable de prétendre que c'est un concept complètement subjectif. L'Open Source doit pouvoir être défini sinon le terme lui-même n'a pas de sens.

Pourquoi rédiger un essai sur l'Open Source sur un site Web traitant de typographie ? Premièrement parce que les outils open source prennent une part de plus en plus importante dans la conception des polices (citons par exemple RoboFab). Deuxièmement parce que des polices soi-disant open source sont en train d'être produites à une cadence accélérée. Troisièmement parce que l'approche utilisée pour l'Open Source est de plus en plus appliquée aux projets issus des domaines du design ou du droit. Quatrièmement parce que je m'intéresse personnellement de longue date à l'Open Source, ayant par exemple travaillé pour Red Hat (ajoutons que ce site tourne avec WordPress, un moteur de blog open source).

Qualité essentielle n°1

- **Dilution : L'Open Source émane d'un esprit de liberté et de coopération.**
- **Réalité : L'Open Source émane d'un esprit de compétition capitaliste.**

L'Open Source, en tant que méthode de conception d'un logiciel, permet l'émergence de produits compétitifs sans les besoins en capitaux et main-d'œuvre inhérents aux méthodes traditionnelles de développement des logiciels propriétaires. La plupart des projets open source à succès sont conçus comme des substituts de logiciels propriétaires à succès. Il n'y a pas de coïncidence. La demande en logiciels propriétaires est

aussi ce qui crée la demande pour des alternatives open source.

De plus, le succès d'un projet open source dépend de sa capacité à rivaliser avec l'alternative propriétaire. Le temps, c'est de l'argent. Les logiciels open source qui ne remplissent pas leur office ne font pas des bonnes affaires au final. Bien que certains choisissent le logiciel open source pour des raisons purement politiques, les clients rationnels se prononceront sur la base d'un bilan coûts/avantages.

Qualité essentielle n°2

- **Dilution : Les développeurs de logiciels open source travaillent gratuitement.**
- **Réalité : Les développeurs de logiciels open source sont rémunérés.**

Personne ne travaille sur des projets open source gratuitement. Peut-être un petit groupe de développeurs contribue-t-il à des projets open source pour se distraire au lieu de collectionner des timbres par exemple. Il s'agit là d'une minorité. La plupart du travail open source est réalisé par des développeurs professionnels qui sont rémunérés à un tarif professionnel.

C'est forcément vrai, et ce pour deux raisons.

Premièrement, les développeurs ne sont pas altruistes. Comme n'importe qui d'autre sur le marché du travail, ce sont des acteurs rationnels, et bien payés avec ça. Il n'y a aucune raison pour que ces développeurs se consacrent au développement de logiciels open source moyennant un salaire de misère alors qu'il existe plein d'entreprises qui accepteraient de les payer pour le même travail.

Deuxièmement, ainsi que nous l'avons déjà mentionné, le logiciel open source ne peut réussir que s'il a les moyens de rivaliser avec les alternatives propriétaires. Et il ne peut

rivaliser avec les logiciels propriétaires que s'il parvient à attirer des développeurs de même niveau. Et la seule façon d'attirer ces développeurs est de les rémunérer au prix du marché. De la même façon qu'il n'y a pas de déjeuner gratuit, il n'y a pas d'avantage de logiciel gratuit.

Qualité essentielle n°3

- Dilution : **L'Open Source rend les choses gratuites.**
- Réalité : **L'Open Source redéfinit ce qui a de la valeur.**

Les développeurs de logiciels open source ne travaillent pas gratuitement, mais il existe un corollaire à cette affirmation : les projets open source bénéficient, pour une valeur équivalente, à ceux qui financent ce travail de développement – habituellement les employeurs des développeurs.

Si vous croyez que les entreprises de technologies sont des acteurs rationnels de l'économie, vous avez certainement raison. Une société investira son capital dans les activités les plus rentables qu'elle pourra trouver. Si une entreprise finance des projets open source, elle en espère un retour sur investissement supérieur aux autres options.

Alors que les défenseurs des logiciels libres ou open source tentent parfois d'en illustrer l'idée avec ces comparaisons : libre comme dans « entrée libre » ou comme dans « expression libre » (*NdT : « free as in beer » et « free as in speech », deux images souvent utilisées en anglais pour distinguer les deux sens du mot « free », respectivement gratuit et libre*), un ingénieur de Red Hat me l'a un jour plus exactement décrit de cette façon : « gratuit comme un chiot ». Certes vous ne payez pas les logiciels open source. Mais vous ne bénéficiez pas des avantages habituels des logiciels propriétaires : facilité d'installation, support d'utilisation, documentation, etc. Soit vous payez avec votre temps, soit vous payez quelqu'un pour ces services. Le résultat est que le coût des

services liés au logiciel est déplacé à l'extérieur du logiciel lui-même au lieu d'être inclus dans son prix. Mais ce coût est seulement déplacé, et non supprimé.

Qualité essentielle n°4

- Dilution : **Il n'y a pas de barrière pour participer à l'Open Source.**
- Réalité : **L'Open Source s'appuie sur la méritocratie.**

L'Open Source ne pourrait pas fonctionner sans un filtrage méritocratique. Ce principe découle de l'idée que l'Open Source est une méthode pour créer des produits compétitifs. Pour obtenir des résultats de haute qualité, les projets open source doivent mettre l'accent sur les contributions de haute qualité, et rejeter le reste. Les projets open source sont ouverts à tous dans le sens où n'importe qui peut suggérer des changements dans le code source. Mais ces changements peuvent toujours être rejetés ou annulés.

L'idée de l'Open Source est mal employée quand elle est appliquée à des projets qui n'ont pas ce filtrage méritocratique et auxquels n'importe qui peut contribuer. Ces projets sont plutôt décrits comme du « partage de fichiers ». Ce qui distingue la méthode open source est son appui sur une communauté de développeurs pour trouver les meilleures idées. Cela signifie donc que la plupart des idées sont rejetées.

Qualité essentielle n°5

- Dilution : **L'Open Source est démocratique.**
- Réalité : **L'Open Source s'appuie sur des dictateurs bienveillants.**

Les projets open source sont menés par des développeurs qui sont parfois appelés des « dictateurs bienveillants ». Habituellement il s'agit de ceux qui ont démarré le projet et qui sont considérés comme détenant la vision de l'avenir du projet.

Ils ne sont pas élus mais, d'un autre côté, il ne sont autorisés à rester au pouvoir que tant que les autres participants y consentent. C'est le principe de l'Open Source : n'importe qui peut récupérer le code source et l'utiliser pour démarrer un nouveau projet (cette pratique a pour nom le « forking »).

Cela arrive rarement. En fin de compte, les participants à un projet open source ont davantage à gagner à conserver le projet intact sous la direction d'un dictateur bienveillant plutôt que le fragmenter en de multiples projets. (notons ici encore l'influence des incitations rationnelles). De même, le dictateur a intérêt à rester bienveillant puisque le projet pourrait à tout moment se dérober sous ses pieds.

Qualité essentielle n°6

- **Dilution : Un projet open source peut n'avoir qu'un développeur.**
- **Réalité : Un projet open source nécessite plusieurs développeurs.**

Cette exigence découle de l'idée que l'Open Source est une méritocratie. Il ne peut y avoir de filtrage méritocratique si toutes les contributions viennent d'une seule personne. Parfois certains publieront leur projet personnel et annonceront « Hé, c'est maintenant open source ».

Cela ne le rend pas pour autant open source, pas plus qu'acheter un pack de bières et des chips suffisent à faire une soirée. Vous aurez toujours besoin d'invités. De même pour créer une pression méritocratique, les projets open source ont besoin de plusieurs développeurs (pour créer une émulsion d'idées) et d'un dictateur éclairé (pour choisir parmi ces idées). Sans cette pression, on a affaire à du partage de fichiers, pas de l'Open Source.

Qualité essentielle n°7

- Dilution : **Un projet logiciel peut devenir open source à tout moment.**
- Réalité : **L'Open Source est inscrit dans l'ADN du projet ou ne l'est pas.**

Comme l'Open Source rencontre un succès grandissant, de plus en plus de projets de logiciels propriétaires ont été « convertis » en logiciels open source. Cela revient à greffer des ailes à un éléphant dans l'espoir qu'il volera.

L'Open Source est une manière de produire des logiciels (entre autres). Cela inclut certaines valeurs et en exclut d'autres. Cela n'est ni intrinsèquement meilleur que le développement de logiciels propriétaires ni applicable à tous les projets. Mais la décision de développer un projet open source n'a du sens qu'au démarrage du projet. De cette manière, le dirigeant, la communauté des développeurs et le code pourront évoluer autour de ces principes.

Un projet propriétaire démarre avec des principes radicalement différents et évolue autour de ces principes. Et il ne pourra pas devenir plus tard un projet open source, pas plus qu'un éléphant ne peut devenir un aigle. Néanmoins, ceux qui convertissent des projets propriétaires en projets open source suggèrent le plus souvent que cela offre le meilleur des deux mondes : une manière de partager les bénéfices d'un développement propriétaire avec une communauté open source.

Mais cela offre presque toujours le pire des deux mondes : l'ouverture des sources n'est qu'une manière cynique d'exporter les problèmes d'un projet propriétaire. Malheureusement c'est ce qui se produit la plupart du temps car des projets propriétaires qui fonctionnent n'ont rien à gagner à devenir open source. Les développeurs n'adoptent pas votre technologie propriétaire ? Ramenez son prix à zéro et renommez le « Open Source ». Votre logiciel propriétaire est

rempli de mystérieux bogues insolubles ? Rendez le « Open Source » et peut-être que quelqu'un d'autre résoudra ces problèmes. Votre technologie est sur le point de devenir obsolète car vous avez été trop lent à la mettre à jour ? Peut-être que la rendre open source allongera sa durée de vie. Les entreprises refourguent leurs logiciels à la communauté open source quand elles n'ont plus rien à perdre.

Cependant cette technique n'a jamais payé. Les projets open source qui fonctionnent ont appliqué cette méthodologie très tôt et s'y sont tenu. Le simple engagement en faveur de l'open source n'est pas un gage de succès bien qu'il soit nécessaire.

La plupart de mes explications ci-dessus ont été formulées en termes de développement logiciel. Mais ce message a une portée qui va au delà du simple logiciel. Je pense que « l'Open Source » vient du monde du logiciel, et restera certainement plutôt adapté à ce type de projet, mais il n'y a aucune raison à ce qu'il ne convienne pas à d'autres types de projets. C'est déjà ce qui se passe.

La principale difficulté est d'appliquer de façon méticuleuse et réfléchie le modèle open source. Au fur et à mesure que l'Open Source s'étend et s'éloigne de ses racines traditionnelles, il va devoir faire face à un risque grandissant de dilution. Encore une fois, je ne prêche pas une définition canonique de l'Open Source. Peut-être que le meilleur que nous pouvons espérer est que ceux qui souhaitent qualifier leur projet d'open source apprendront d'eux-mêmes les caractéristiques qui font le succès de ce type de projet. Si vous n'aimez pas mon résumé de ces qualités essentielles, alors apprenez-en suffisamment à propos de l'Open Source pour venir avec le vôtre. Et ne vous sentez pas obligé, ne prenez pas cela comme un devoir à la maison – faites cela dans votre propre intérêt.

Ça ne peut pas faire de mal de se pencher sur les autres

projets open source pour comprendre comment ils ont réussi. Une fois cet examen réalisé, si vous voulez modifier les grands principes pour votre propre projet, allez-y ! Je ne m'en plaindrai pas.

Et accordez-moi une faveur : n'appellez pas cela Open Source.

Notes

[1] Crédit photo : UGArdener (Creative Commons By-Sa)