

Un logiciel libre n'est pas toujours collaboratif et de qualité

Voici un titre étrange pour un blog comme le nôtre.

Oui il existe des logiciels libres de mauvaise qualité qui ne souffrent pas la comparaison avec leurs concurrents propriétaires ! Et oui encore la majorité des logiciels libres sont uniquement développés par un seul et unique contributeur : leur créateur !

Face à de tels logiciels, les partisans de *l'open source* pleurent car ils détruisent aussi bien leur argumentaire pratico-technique que le mythe de la collaboration spontanée. Les partisans de *logiciel libre* envisagent quant à eux les choses différemment car ce qu'ils voient avant tout c'est que le logiciel est libre.

Le logiciel libre n'est pas meilleur *en pratique* mais il est libre *en théorie* et c'est bien ça le plus important...

Remarque : Cette traduction est le fruit d'une coopération entre Framasoft (et son énergie plurielle présente sur Framalang et les réseaux sociaux) et l'April (via son équipe de traduction du site GNU.org)



Quand le logiciel libre n'est pas meilleur, en pratique

When Free Software Isn't (Practically) Better

Benjamin Mako Hill - GNU.org

Licence Creative Commons By-Nd - Version du 6 octobre 2012

(Traduction : Framalang et l'équipe Trad-GNU de l'April)

Les objectifs affichés par l'*Open Source Initiative* sont les suivants : « L'open source est une méthode de développement logiciel qui exploite la puissance d'une évaluation décentralisée, par les pairs, et la transparence des processus. Les promesses de l'open source sont une meilleure qualité, une plus grande fiabilité, davantage de flexibilité, un moindre coût et la fin d'une situation permettant à des fournisseurs rapaces de verrouiller leurs produits. »

Depuis plus de dix ans maintenant, la *Free Software Foundation* ne cesse d'argumenter contre la qualification d'« open source » dont on affuble le mouvement du logiciel libre. Si nous, les partisans du logiciel libre, réfutons ce qualificatif d'« open source », c'est surtout parce que nous considérons qu'il s'agit

d'un effort volontaire pour réduire la portée de notre message de liberté et masquer le rôle de notre mouvement dans le succès du logiciel que nous avons bâti. Si nous disons que le terme « open source » est mauvais, c'est fondamentalement parce qu'il tente d'éviter toute discussion à propos de la liberté du logiciel. Mais il y a une autre raison pour laquelle nous devrions nous méfier du cadre « open source ». L'argument fondamental de l'open source, tel qu'il est défini dans la déclaration ci-dessus, est souvent incorrect.

Malgré la suggestion de l'*Open Source Initiative*, que « la promesse de l'open source est une meilleure qualité, une plus grande fiabilité, plus de flexibilité », cette promesse n'est pas toujours honorée. Bien que nous ne le mettions pas souvent en avant, tout utilisateur d'un logiciel libre aux premiers stades de son développement peut expliquer que ce logiciel n'est pas toujours aussi pratique, sur le plan purement fonctionnel, que ses concurrents privés^[1] Un logiciel libre est parfois de piètre qualité. Il n'est pas toujours très fiable. La souplesse lui fait parfois défaut. Si les gens prennent les arguments en faveur de l'open source au sérieux, ils doivent expliquer pourquoi l'open source n'a pas tenu ses « promesses » et conclure que des outils privés seraient un meilleur choix. Il n'y a aucune raison pour que nous fassions de même.

Richard Stallman parle de cela dans son article « Pourquoi l'open source passe à côté du problème que soulève le logiciel libre » lorsqu'il explique : « L'open source repose sur l'idée qu'en permettant aux utilisateurs de changer et redistribuer le logiciel, celui-ci en sortira plus puissant et plus fiable. Mais cela n'est pas garanti. Les développeurs de logiciels privés ne sont pas forcément incompetents. Parfois ils produisent un programme qui est puissant et fiable, même s'il ne respecte pas la liberté de l'utilisateur. »

Pour l'open source, la mauvaise qualité d'un logiciel est un problème à analyser ou une raison de fuir ce logiciel. Pour le libre, c'est un problème à résoudre. Pour les partisans du libre, les bogues et les fonctionnalités manquantes ne sont jamais une raison d'avoir honte. Tout logiciel qui respecte la liberté de ses utilisateurs possède un avantage inhérent sur son concurrent privé. Même s'il a ses propres problèmes, un logiciel libre a toujours la liberté.

Bien évidemment, tout logiciel libre doit commencer quelque part. Un nouveau programme, par exemple, a peu de chances d'offrir plus de fonctionnalités qu'un programme privé déjà établi. Un projet commence avec de nombreux bogues

et s'améliore avec le temps. Alors que les partisans de l'open source peuvent argumenter qu'un projet deviendra utile avec du temps et un peu de chance, un projet libre représente pour les partisans du logiciel libre une importante contribution, dès le premier jour. Chaque logiciel qui donne aux utilisateurs le contrôle sur leur technologie est un pas en avant. L'amélioration en qualité due à la maturation d'un projet n'est que la cerise sur le gâteau.

Un second point, peut-être plus accablant encore, est que le processus de développement collaboratif, distribué, évalué par les pairs, qui est au cœur de la définition de l'open source, ne ressemble que de loin à la manière dont sont développés en pratique la plupart des projets sous licence libre (ou « open source »).

Plusieurs études universitaires menées sur les sites d'hébergement de logiciels libres SourceForge et Savannah ont démontré ce que beaucoup de développeurs de logiciels libres ayant mis en ligne une base de code savent déjà : la grande majorité des projets libres ne sont pas particulièrement collaboratifs. Le nombre médian de contributeurs à un projet de logiciel libre sur SourceForge ? Un. Un développeur solitaire. Les projets de SourceForge du quatre-vingt-quinzième centile en termes de nombre de participants n'ont que cinq contributeurs. Plus de la moitié de ces projets libres, et même la plupart des projets qui ont fait plusieurs versions à succès et ont été téléchargés fréquemment sont l'œuvre d'un seul développeur avec un peu d'aide de l'extérieur.

En insistant sur la puissance du développement collaboratif et de « l'évaluation décentralisée par les pairs », l'approche open source semble ne pas avoir grand chose à dire, dans la majorité des cas, sur les raisons pour lesquelles on devrait contribuer à un projet libre ou se servir d'un logiciel en développement. Puisque les avantages supposés de la collaboration ne peuvent être constatés quand il n'y a pas de collaboration, la grande majorité des projets de développement libres n'ont pas d'avantage technique sur leurs concurrents privés.

Pour les partisans du logiciel libre, ces mêmes projets sont tous vus comme des succès importants. Comme chaque logiciel libre respecte la liberté de ses utilisateurs, les partisans du libre peuvent argumenter qu'il possède au départ un avantage éthique intrinsèque sur les concurrents privés - même sur ceux qui proposent plus de fonctionnalités. En insistant sur la liberté plutôt que sur les avantages pratiques, la défense du logiciel libre est ancrée dans la réalité

technique d'une façon qui manque souvent à l'open source. Quand le logiciel libre est meilleur, nous pouvons nous en réjouir. Quand il ne l'est pas, nous n'avons pas à considérer cela comme une attaque dirigée contre lui ni même comme un argument valable contre l'utilisation du logiciel en question.

Les partisans de l'open source doivent défendre leur thèse selon laquelle le logiciel développé librement devrait, ou devra avec le temps, être meilleur que le logiciel privateur. Les militants du logiciel libre peuvent quant à eux demander : « Comment peut-on rendre le logiciel libre meilleur ? » Dans le cadre du libre, les logiciels de haute qualité existent comme un moyen plutôt que comme une fin en soi. Les développeurs de logiciels libres doivent s'efforcer de créer des logiciels fonctionnels, flexibles, qui servent bien leurs utilisateurs. Mais ceci n'est pas le seul moyen de progresser vers la réalisation d'un objectif qui est à la fois plus simple et bien plus important : respecter et protéger leurs libertés.

Bien sûr, nous ne cherchons pas à nier que la collaboration joue un rôle important dans la création de logiciels de haute qualité. Dans la plupart des projets libres ayant réussi, ce fut d'ailleurs le cas. Il faut comprendre, soutenir et développer la collaboration, plutôt que de considérer dogmatiquement qu'elle va de soi, quand bien même les faits sont là pour montrer le contraire.

Crédit photo : James Rickwood (Creative Commons By)

Notes

[1] Autre traduction de *proprietary* : propriétaire