

Les réalités économiques du logiciel libre (Libres conseils 40/42)

* Aujourd'hui 28 mars, dernière séance de traduction collaborative sur ce projet avec l'épisode n°42 !

Traduction Framalang : [tcit](#), [Julius22](#), [Sphinx](#), [goofy](#), [peupleLà](#), [merlin8282](#), [lamessen](#), [BAud](#), [Jej](#), [Alpha](#)

Modèles économiques basés sur le libre et l'open source

Carlo Daffara

Carlo Daffara est chercheur dans le domaine des modèles économiques basés sur l'open source, le développement collaboratif d'objets numériques et l'utilisation de logiciels open source dans les entreprises. Il fait partie du comité éditorial de relecture du journal international des logiciels et processus open source (International Journal of Open Source Software & Processes : IJOSSP), est membre du comité technique de deux centres régionaux de compétences open source et est également membre du réseau juridique européen FSFE (fondation européenne pour le logiciel libre). Il a pris part aux comités SC34 et JTC1 pour la branche italienne de l'ISO, UNINFO et a travaillé au sein du groupe de travail de la société Internet du logiciel public (Internet Society Public Software) ainsi que pour beaucoup d'autres initiatives liées à la normalisation.

Auparavant, Carlo Daffara était le représentant italien dans le groupe de travail européen sur le logiciel libre, la première initiative de l'Union européenne afin de soutenir l'open source et le logiciel libre. Il a présidé le groupe de travail SME du groupe d'étude de l'UE sur la compétitivité et le groupe de travail IEEE des intergiciels open source du comité technique sur le calcul évolutif. Il a travaillé en tant qu'examineur du projet pour la commission Européenne dans le domaine de la coopération internationale, l'ingénierie logicielle, l'open source et les

systemes distribués et a été directeur de recherche dans plusieurs projets de recherche de l'Union européenne.

Introduction

« Comment gagner de l'argent avec le logiciel libre ? » était une question très courante, il y a encore seulement quelques années. Désormais, cette question s'est transformée en « Quelles sont les stratégies commerciales pouvant être mises en œuvre en se basant sur le logiciel libre et *open source* ? ». Cette question n'est pas aussi gratuite qu'elle peut paraître, puisque de nombreux chercheurs universitaires écrivent encore ce genre de textes : « le logiciel *open source* est délibérément développé hors de tout mécanisme de marché... il échoue à contribuer à la création de valeur aux développements, contrairement au marché du logiciel commercial... il ne génère pas de profit, de revenus, d'emplois ou de taxes...

Les licences *open source* sur les logiciels visent à supprimer les droits d'auteurs sur le logiciel et empêchent d'établir un prix pour le logiciel. Au final, les logiciels développés ne peuvent être utilisés pour générer des profits. » [Koot 03] ou [Eng 10] indiquent que « des économistes ont montré que les collaborations *open source* dans le monde réel s'appuient sur plusieurs incitations différentes telles qu'enseigner, se démarquer et se créer une réputation » (sans parler des incitations économiques). Cette vue purement « sociale » du logiciel libre et *open source* est partielle et fautive. Et nous démontrerons qu'il y a des raisons économiques liées au succès des métiers du libre et de l'*open source* qui vont au-delà des collaborations purement bénévoles.

Le logiciel libre et *open source* face aux réalités économiques

Dans la plupart des domaines, l'utilisation d'un logiciel libre et *open source* apporte un avantage économique substantiel, grâce aux développements partagés et aux coûts de maintenance, déjà décrits par des chercheurs comme Gosh, qui a estimé une réduction de coût de 36 % en R&D (« Recherche et Développement », NdT). La vaste part de marché des déploiements « internes » de logiciels libres et *open source* explique pourquoi certains des bénéfices économiques ne sont pas

directement visibles sur le marché des services commerciaux.

L'étude FLOSSIMPACT a montré, en 2006, que les entreprises qui contribuent au code de projets de logiciels libres et *open source* ont, au total, au moins 570 000 employés et un chiffre d'affaires annuel de 263 milliards d'euros [Gosh 06], faisant ainsi du logiciel libre et *open source* l'un des phénomènes les plus importants des NTIC. Il est important aussi de reconnaître qu'un pourcentage non négligeable de cette valeur économique n'est pas directement perceptible du marché, vu que la majorité du logiciel n'est pas développée dans l'intention de le vendre (le soi-disant logiciel « prêt à l'emploi ») mais uniquement à usage interne. Comme le réseau thématique FISTERA EU l'a identifié, en réalité, la majorité du logiciel est développée seulement pour un usage interne.

Région	Licences de logiciels propriétaires	Services logiciel (développement personnalisation)	Développement interne
Union européenne	19 %	52 %	29 %
États-Unis	16 %	41 %	43 %
Japon	N/A	N/A	32 %

Il est clair que ce qui est appelé « le marché logiciel » est en réalité bien plus réduit que le vrai marché du logiciel et des services et que 80 % restent invisibles. Nous verrons que le FLOSS tient une place économique importante de ce marché, directement grâce à ce modèle de développement interne.

Modèles économiques et proposition de valorisation

L'idée de base d'un modèle économique est assez simple : j'ai quelque chose ou je peux faire quelque chose (la « proposition de valeur ») et c'est plus rentable de me payer ou d'obtenir ce quelque chose plutôt que de le faire soi-même (il est même parfois impossible de trouver des alternatives, comme dans le cas de monopoles naturels ou créés par l'homme, et l'idée même de le produire par soi-même n'est pas envisageable). Il y a deux sources possibles de valeur : une propriété (quelque chose qui peut être échangé) et l'efficacité (quelque chose propre à ce que fait une entreprise et la manière dont elle le fait).

Avec l'*open source*, la « propriété » est généralement non exclusive (à l'exception de ce qui est nommé « cœur ouvert », où une partie du code n'est pas libre du tout et cela sera abordé plus loin dans cet article). D'autres exemples de propriété concernent le droit des marques, les brevets, les licences... tout ce qui peut être transféré à une autre entité par contrat ou par une transaction légale. L'efficacité est la capacité à effectuer une action avec un coût moindre (qu'il soit tangible ou intangible) et cela correspond à la spécialisation dans un domaine d'application ou apparaît grâce à une nouvelle technologie.

Pour le premier cas, les exemples sont simplement la réduction du temps nécessaire pour réaliser une action quand vous augmentez votre expertise concernant ce sujet. La première fois que vous installez un système complexe, cela peut demander beaucoup d'efforts et cet effort diminue d'autant plus que vous connaissez les tâches nécessaires pour réaliser l'installation elle-même. Pour le second, cela peut être l'apparition d'outils qui simplifient le processus (par exemple, avec le clonage d'images) et introduisent une importante rupture, un « saut » dans la courbe efficacité-temps.

Ces deux aspects sont la base de tout modèle économique que nous avons analysé par le passé ; il est possible de montrer que tout ceux-ci échouent afin de garantir une continuité entre les propriétés et l'efficacité.

Parmi les résultats de notre précédent projet de recherche, nous avons trouvé que les projets basés sur un modèle propriétaire ont tendance à obtenir moins de contributions extérieures car cela nécessite une opération juridique pour faire partie des propriétés de l'entreprise, pensez par exemple aux licences doubles : afin que son code fasse partie du code du produit, un contributeur extérieur doit signer l'abandon des droits sur son code afin que l'entreprise puisse vendre la version commerciale ainsi que la version *open source*.

D'un autre côté, les modèles totalement orientés sur l'efficacité ont tendance à avoir plus de contributions et de visibilité mais des résultats financiers plus faibles. Je l'ai écrit plusieurs fois : il n'y a pas de modèle économique idéal mais un éventail de modèles possibles et les entreprises devraient s'adapter elles-mêmes pour changer les conditions du marché et aussi adapter leur modèle. Certaines entreprises débutent par des modèles entièrement axés sur l'efficacité puis construisent, avec le temps, une propriété en interne, d'autres ont commencé avec un modèle orienté vers la propriété et ont évolué différemment pour

augmenter les contributions et réduire les efforts d'ingénierie (ou développer la base d'utilisateurs afin de créer d'autres moyens d'avoir un retour financier grâce aux utilisateurs).

Une typologie des modèles économiques

L'étude EU FLOSSMETRICS des modèles économiques basés sur le logiciel libre a identifié, après analyse de plus de 200 entreprises, une taxonomie des principaux modèles économiques utilisés par les entreprises *open source* ; les principaux modèles identifiés sur le marché sont :

- la double licence : le même code source logiciel distribué sous GPL et sous une licence propriétaire. Ce modèle est principalement utilisé par les producteurs de logiciel et outils pour développeurs et fonctionne grâce à une forte association de la GPL, qui requiert que les travaux dérivés et logiciels liés directement soient distribués sous la même licence. Les entreprises ne souhaitant pas distribuer leur propre logiciel sous GPL peuvent obtenir une licence propriétaire leur octroyant une exemption des conditions de la GPL, ce qui semble souhaitable à certains. L'inconvénient de cette licence double est que les contributeurs externes doivent accepter des conditions similaires et cela a révélé des réductions de contributions externes, se limitant à des corrections de bogues et des ajouts mineurs ;
- le modèle « cœur ouvert » (précédemment appelé « valeur ajoutée propriétaire » ou « séparation entre libre et propriétaire » * ? *\) : ce modèle se distingue entre un logiciel libre basique et une version propriétaire, basée sur la version libre mais avec l'ajout de greffons propriétaires. La plupart des entreprises qui suivent un tel modèle adoptent la Mozilla Public Licence, car elle permet explicitement cette forme de mélange et permet une plus grande participation des contributions externes sans les mêmes contraintes de consolidation du droit d'auteur comme dans l'usage de doubles licences. Ce modèle a l'inconvénient intrinsèque que le logiciel libre doit être de grande valeur pour être attractif pour les utilisateurs, i.e. il ne doit pas être réduit à une version aux possibilités limitées, tout comme, dans le même temps, il ne doit pas « cannibaliser » le produit propriétaire. Cet équilibre est difficile à atteindre et à maintenir dans la durée ; en outre, si le logiciel est de

grand intérêt, les développeurs peuvent essayer d'apporter les fonctionnalités manquantes dans le logiciel libre, réduisant ainsi l'intérêt de la version propriétaire et donnant potentiellement naissance à un logiciel concurrent entièrement libre qui ne souffrira pas des mêmes limitations ;

- les experts produits : des entreprises qui ont créé ou maintiennent un projet logiciel spécifique et utilisent une licence libre pour le distribuer. Les principaux revenus viennent du service, comme la formation ou l'expertise, et suivent la classification EUWG d'origine « le meilleur code vient d'ici » et « les meilleures compétences sont ici » [DB 00]. Cela conforte l'impression, courante, que les experts les plus compétents sur un logiciel sont ceux qui l'ont développé et qu'ils peuvent ainsi fournir des services au prix d'un démarchage minimal, s'appuyant sur la fourniture gratuite du code. L'inconvénient de ce modèle est que le coût d'entrée pour des concurrents potentiels est faible, vu que le seul investissement nécessaire est l'acquisition des compétences sur le logiciel lui-même ;
- les fournisseurs de plateforme : des entreprises qui apportent un ensemble de services, avec support et intégration de certains projets, constituant une plateforme cohérente et testée. En ce sens, même les distributions GNU/Linux sont classées en tant que plateforme ; une observation intéressante est que ces distributions sont distribuées en grande partie sous licence libre pour maximiser les contributions externes et s'appuyer sur la protection du droit d'auteur pour empêcher la copie sauvage sans empêcher les « déclinaisons » (suppression des particularités soumises à droit d'auteur comme les logos ou droit des marques, pour créer un nouveau produit). Des exemples de clones de Red Hat sont CentOS et Oracle Linux. La valeur ajoutée provient d'une qualité garantie, de la stabilité et de la fiabilité ainsi que d'une garantie de support pour les applications métier critiques ;
- les entreprises de conseil et de recrutement : les entreprises de cette catégorie ne font pas vraiment de développement mais fournissent des conseils de sélection et des services d'évaluation pour un large éventail de projets, d'une manière qui est proche du rôle de l'analyste. Ces entreprises ont tendance à avoir un impact très limité sur les communautés car les résultats de l'évaluation et du processus d'évaluation sont généralement des données propriétaires ;
- les fournisseurs de support global : des entreprises qui proposent un

support centralisé pour un ensemble de produits de logiciel libre, généralement en employant directement les développeurs ou en remontant les demandes de support ;

- la validation juridique et l'expertise : ces entreprises n'apportent pas de développements de code source mais fournissent une aide à la vérification de conformité aux licences, parfois en apportant une garantie et une assurance contre les attaques juridiques ; certaines entreprises utilisent des outils pour assurer que le code n'est pas réutilisé ;
- la formation et la documentation : des entreprises qui proposent de la formation, en ligne et en présentiel, des documentations et des manuels supplémentaires. Cela est généralement fourni dans le cadre d'un contrat de support, mais, récemment, quelques réseaux de centres de formation ont lancé des cours orientés spécifiquement vers le logiciel libre ;
- le partage des coûts de R&D : une entreprise ou une société peut avoir besoin d'une nouvelle version ou d'une amélioration d'un paquet logiciel et financer un consultant ou un développeur pour réaliser le travail. Plus tard, le logiciel développé est redistribué en *open source* pour bénéficier de l'ensemble des développeurs expérimentés pouvant le déboguer et l'améliorer. Un bon exemple est la plateforme Maemo, utilisée par Nokia pour ses smartphones (comme le N810) ; au sein de Maemo, seul 7,5 % du code est propriétaire, apportant une réduction des coûts estimée à 228 millions de dollars (et une réduction du temps de mise sur le marché d'un an). Un autre exemple est l'écosystème Eclipse, un environnement de développement intégré (EDI) distribué à l'origine par IBM comme logiciel libre puis ensuite géré par la fondation Eclipse. De nombreuses entreprises ont choisi Eclipse comme socle pour leur produit et ont ainsi réduit le coût global pour la création d'un logiciel fournissant une fonctionnalité pour les développeurs. Il y a un grand nombre d'entreprises, d'universités et de personnes qui participent à l'écosystème Eclipse. Comme récemment constaté, IBM contribue aux alentours de 46 % au projet, les contributeurs à titre personnel représentant 25 % et un grand nombre d'entreprises comme Oracle, Borland, Actuate et de nombreuses autres ayant des participations allant de 1 à 7 %. Ceci est semblable aux résultats obtenus grâce à l'analyse du noyau Linux et qui montre que, lorsqu'il y a un écosystème sain et de grande taille, le partage des tâches réduit de manière significative les coûts de maintenance, dans [Gosh 06], on estime qu'il est possible de faire des

économies de l'ordre de 36 % dans la recherche et la conception logicielle grâce à l'utilisation du logiciel libre, ces économies constituent en elles-mêmes le plus gros « marché » réel pour le logiciel libre, ce qui est démontré par le fait qu'au moins une partie du code des développeurs est basé sur du logiciel libre (56,2 % comme mentionné dans [ED 05]). Un autre excellent exemple de « coopération » inter-entreprises est le projet WebKit, le moteur de rendu HTML à la base du navigateur Google Chrome ainsi que d'Apple Safari et qui est utilisé dans la majorité des appareils mobiles. Dans ce projet, après un délai initial d'un an, le nombre de contributions externes a commencé à devenir significatif et, après un an et demi, il surpasse largement les contributions d'Apple — réduisant de fait les coûts de maintenance et d'ingénierie grâce à la répartition des tâches entre les co-développeurs ;

- les revenus indirects : une entreprise peut choisir de financer des projets de logiciel libre si ces projets peuvent créer une source de revenus importante pour des produits dérivés, non liés directement au code source ou au logiciel. L'un des cas les plus courants correspond à l'écriture de logiciel nécessaire au fonctionnement de matériel, par exemple, les pilotes d'un système d'exploitation pour un matériel spécifique. En fait, de nombreux fabricants de matériel distribuent déjà gratuitement leurs pilotes logiciels. Certains d'entre eux distribuent déjà certains de leurs pilotes (surtout ceux pour le noyau Linux) sous une licence libre. Le modèle du produit d'appel est une stratégie commerciale traditionnelle, répandue même à l'extérieur du monde du logiciel : dans ce modèle, les efforts sont consacrés à un projet de logiciel libre et *open source* afin de créer ou d'étendre un autre marché dans des conditions différentes. Par exemple, les fournisseurs de composants matériels investissent dans le développement de pilotes logiciels pour des systèmes d'exploitation *open source* (comme GNU/Linux) pour s'étendre sur le marché spécifique des composants. D'autres modèles de revenus auxiliaires sont ceux, par exemple, de la fondation Mozilla qui réunit une somme d'argent non négligeable grâce à un partenariat avec Google sur le moteur de recherche (estimé à 72 millions de dollars en 2006), tandis que SourceForge/OSTG est financé en majorité par les recettes des ventes en ligne du site partenaire ThinkGeek.

Certaines entreprises ont plus d'un modèle principal et sont, par conséquent,

comptées en double ; notamment, la plupart des entreprises pratiquant une licence double vendent aussi du service de support. En outre, les experts d'un produit ne sont comptés que s'ils ont une partie visible de leur entreprise qui contribue au projet en tant que « commiter principal ». Autrement, le nombre d'experts serait bien plus élevé, du fait que certains projets sont au cœur du support commercial de nombreuses entreprises (de bons exemples sont OpenBravo et Zope).

Il faut aussi tenir compte du fait que les fournisseurs de plateforme, même s'ils sont limités en nombre, tendent à avoir des taux de facturation plus élevés que les experts ou que les entreprises à cœur ouvert. De nombreux chercheurs essaient d'identifier s'il y a un modèle plus « efficace » parmi ceux pris en compte ; ce que nous avons trouvé est que le futur le plus probable sera l'évolution d'un modèle à l'autre, avec une consolidation sur le long terme des consortiums de développement (comme les fondations Eclipse et Apache) qui fournissent une forte infrastructure légale et des avantages de développement ainsi que des spécialistes apportant des offres verticales pour des marchés spécifiques.

Conclusion

Le logiciel libre et *open source* permet non seulement une présence pérenne, et même très large, sur le marché (Red Hat est déjà proche du milliard de dollars de revenus annuels), mais aussi plusieurs modèles différents qui sont totalement impossibles avec le logiciel propriétaire. Le fait que le logiciel libre et *open source* est un bien non concurrent facilite aussi la coopération entre entreprises, tant pour accroître sa présence mondiale et pour signer des contrats à grande échelle pouvant demander des compétences multiples que sur le plan géographique (même produit ou service, région géographique différente) ; « verticalité » (entre produits) ou « horizontalité » (des domaines d'application). Cet adjuvant à créer de nouveaux écosystèmes est l'une des raisons expliquant que le logiciel libre et *open source* fait partie intégrante de la plupart des infrastructures informatiques dans le monde, enrichissant et aidant les entreprises et administrations publiques à réduire leurs coûts et à collaborer pour de meilleurs logiciels.

Bibliographie

- *[DB00] Daffara, C.Barahona, J.B. Free Software/Open Source: Information Society Opportunities for Europe working paper, <http://eu.conecta.it> paper, OSSEMP workshop, Third international conference on open source. Limerick 2007*
- *[ED05] Evans Data, Open Source Vision report, 2005*
- *[Eng10] Engelhardt S. Maurer S. The New (Commercial) Open Source: Does it Really Improve Social Welfare Goldman School of Public Policy Working Paper No.GSPP10-001, 2010*
- *[Gar06] Gartner Group, Open source going mainstream. Gartner report, 2006*
- *[Gosh06] Gosh, et al. Economic impact of FLOSS on innovation and competitiveness of the EU ICT sector. <http://bit.ly/cNwUzû>*
- *[Koot03] Kooths, S.Langenfurth, M.Kaiwey,N.Open-Source Software: An Economic Assessment Technical report, Muenster Institute for Computational Economics (MICE), University of Muenster*