

Chapitre 10 - Open Source et SaaS

Si le monde du logiciel a vu l'arrivée de nombreux cow-boys (sans doute du fait des faibles investissements initiaux), le mouvement open source représente le plus grand type de cow-boys que l'on n'ait jamais vu. Non seulement n'importe qui peut créer son propre logiciel, mais n'importe qui peut participer au développement d'un logiciel tiers ! Aucune méthodologie, aucune organisation, aucun contrôle centralisé, aucune tâche forcée.

Le mouvement Open Source

Depuis 1998, les logiciels à code source ouvert explosent sur le devant de la scène avec des hauts et des bas. Pour le moment, l'irruption du mouvement Open Source dans l'actualité du monde informatique est encore trop souvent considérée comme un phénomène de mode car l'engouement qui l'entoure oscille entre l'enthousiasme et l'incompréhension. Ce mouvement vient de loin car il y a toujours eu des logiciels qui étaient versés dans le "domaine public" et pour lesquels il n'y avait aucune licence, aucune restriction, même pas d'auteur identifié dans certains cas !

Mais, au-delà de la zone "où tout est permis" du domaine public, un programmeur américain, Richard Stallman, décida de formaliser une démarche rigoureuse où la liberté de copier, de modifier et de distribuer librement des logiciels serait préservée et garantie. Cette réflexion déboucha sur une licence spécifique, la GPL (General Public Licence).

=== La licence GPL

Extraits issus de http://fr.wikipedia.org/wiki/Licence_publicique_générale_GNU

La Licence publique générale GNU, ou GNU General Public License (son seul nom officiel en anglais, communément abrégé GNU GPL voire simplement couramment "GPL") est une licence qui fixe les conditions légales de distribution des logiciels libres du projet GNU. Richard Stallman et Eben Moglen, deux des grands acteurs de la Free Software Foundation, en furent les premiers rédacteurs. Sa dernière version est la GNU GPL version 3 publiée le 29 juin 2007. Elle a depuis été adoptée, en tant que document définissant le mode d'utilisation, donc d'usage et de diffusion, par de nombreux auteurs de logiciels libres.

La GPL met en œuvre la notion de copyleft, un jeu de mots anglais faisant référence à la notion de copyright (right en anglais signifie à la fois le droit, c'est-à-dire la règle juridique et la droite, qui marque une direction) que l'on peut transposer en français en parlant de gauche d'auteur par référence au droit d'auteur. Pour autant le copyleft n'est pas l'antithèse du copyright, bien au contraire, puisqu'en réalité le premier s'appuie sur le second. Ainsi le copyleft comme le copyright définissent et encadrent les droits des utilisateurs de façon contraignante. Si le mécanisme est le même, les objectifs diffèrent : là où le copyright garantit exclusivement les droits de l'auteur, le copyleft s'attarde en outre tout particulièrement sur les droits des utilisateurs et vise à préserver la liberté d'utiliser, d'étudier, de modifier et de diffuser le logiciel et ses versions dérivées. La GPL est la licence de logiciel libre la plus utilisée.

L'objectif de la licence GNU GPL, selon ses créateurs est de garantir à l'utilisateur les droits suivants (appelés libertés) sur un programme informatique :

- La liberté d'exécuter le logiciel, pour n'importe quel usage ;
- La liberté d'étudier le fonctionnement d'un programme et de l'adapter à ses besoins, ce qui passe par l'accès aux codes sources ;
- La liberté de redistribuer des copies ;

Cow-boys contre chemin de fer ou que savez-vous vraiment de l'histoire de l'informatique ?

- La liberté d'améliorer le programme et l'obligation de rendre publiques les modifications afin que l'ensemble de la communauté en bénéficie.

La GNU GPL (abrégé GPL) a été écrite par Richard Stallman pour être utilisée sur les programmes du projet GNU. Elle est basée sur l'assemblage des licences utilisées par GNU Emacs, GNU Debugger (GDB) et la GNU Compiler Collection (GCC). Ces licences contiennent des clauses identiques, mais elles sont spécifiques à chaque programme. Le but de Stallman est de produire une licence unique qui pourra être utilisée pour chaque projet et que cette licence permette au plus grand nombre de projets de partager leur code source. C'est ainsi que naquit la GPL version 1 en février 1983.

En 1990, il était devenu évident qu'une licence moins restrictive serait utile pour quelques bibliothèques logicielles. Ainsi, quand la version 2 de la GPL apparut en juin 1991, une nouvelle licence fut créée, la GNU Library General Public License (abrégé GNU LGPL ou LGPL) prenant elle aussi la version 2 pour marquer leur lien de parenté. Les numéros de versions sont devenus différents en 1999 quand la version 2.1 de LGPL est arrivée. La LGPL a changé de nom en même temps afin de mieux refléter sa place par rapport à l'esprit GNU : elle s'appelle désormais la GNU Lesser General Public License (toujours abrégé GNU LGPL ou LGPL).

En 2005, Richard Stallman a commencé à écrire la version 3 de la GPL.

...
===

"La cathédrale et le bazar", l'article fondateur

Au début des années quatre-vingt-dix, Linux commença à apparaître et certains au sein du mouvement du logiciel libre voulaient s'affranchir de la tutelle de Richard Stallman, ce fut le cas notamment d'Eric Raymond qui fut à l'origine du mouvement Open Source qui diffère subtilement du mouvement logiciel libre. Dans le livre "Richard Stallman et la révolution du logiciel libre", Sam Williams, Richard Stallman & Christophe Masutti racontent ainsi le moment où Eric Raymond rédigea son fameux article "La cathédrale et le bazar" qui fut à l'origine du mouvement Open source : Raymond s'aperçut que cette méthode permettait en outre de contourner la loi de Brooks. Énoncée pour la première fois par Fred P. Brooks, responsable du projet OS/360 chez IBM et auteur du livre The Mythical Man-Month en 1975, cette loi postulait que l'ajout de développeurs à un projet ne faisait qu'entraîner des retards supplémentaires. Croyant comme la plupart de ses confrères programmeurs que le logiciel, à l'instar de la soupe, a tout avantage à être préparé par un nombre limité de cuisiniers, Raymond sentit que quelque chose de révolutionnaire était à l'œuvre. En invitant toujours plus de cuisiniers à la cuisine, Torvalds avait effectivement réussi à rendre le logiciel "meilleur".

La loi de Brooks est la forme courte de la citation suivante, tirée du livre de l'auteur : Puisque la fabrication de logiciels est intrinsèquement le fruit de l'effort conjugué de différents systèmes -un exercice d'interactions complexes- l'effort de communication est grand et il contrebalance rapidement la diminution du temps de travail individuel rendue possible par le partage des tâches. Au lieu de réduire les délais, ajouter plus de personnel les allonge. Raymond coucha ses observations sur le papier et les mit en forme dans un discours, qu'il lut rapidement devant un groupe d'amis et de voisins, dans le comté de Chester, en Pennsylvanie. Intitulé "La cathédrale et le bazar", le texte établissait un contraste entre le management de type "bazar" initié par Torvalds, et celui de style "cathédrale", utilisé habituellement. D'après Raymond, les réactions furent enthousiastes et elles le furent encore plus au printemps suivant, lors du Linux Kongress de 1997 qui réunissait des utilisateurs GNU/Linux allemands. Raymond finit par convertir ce discours en article, intitulé lui- aussi "La cathédrale et le bazar". Le texte tirait son nom de l'analogie centrale de Raymond. Auparavant, les programmes étaient des "cathédrales", monuments impressionnants dont la construction était planifiée de façon centralisée et conçus pour résister aux assauts du temps. À l'opposé, Linux ressemblait davantage à "un grand bazar bruyant", un projet logiciel développé grâce aux dynamiques floues et décentralisées de l'Internet.

== on peut retrouver l'article "La cathédrale et le bazar" (The Cathedral and the Bazaar) complet à http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar_monoblock.html ==

Face à une innovation majeure qui accumule les succès, les sceptiques rétorquent toujours que la portée est limitée. Il y a quinze ans, les gens disaient « les bricoleurs de la

Cow-boys contre chemin de fer ou que savez-vous vraiment de l'histoire de l'informatique ?

FSF (Free Software Foundation) ont développé quelques belles démonstrations mais rien de sérieux ni d'utilisable » et le projet GNU démontra le contraire. Les détracteurs dirent alors « le toolkit GNU est intéressant mais on est encore loin d'un système complet et opérationnel » et Linux fit son apparition. Aujourd'hui, les sceptiques prétendent encore qu'il ne faut rien attendre au-delà des couches systèmes... Pourquoi devrions-nous croire ces pessimistes alors que le passé leur a toujours donné tort ?

Le fait est que, même pour des solutions très exigeantes comme les suites bureautiques, Open Office est désormais considérée comme une alternative viable à MS-Office ou GIMP comme un concurrent de Photoshop sans parler de Firefox dont la popularité ne cesse de grandir !

De même, si on trouve des logiciels libres de tout type, force est de constater que les projets qui ont eu le plus de succès ont été des projets d'informaticiens destinés (du moins au début) à des informaticiens. Rien de surprenant, car les meilleurs logiciels ont été développés par leurs utilisateurs. Ce n'est pas un hasard si les logiciels open source qui ont le plus pris ont été des systèmes d'exploitation (Linux, FreeBSD, Gnome), des outils de développement (GCC, Emacs, MySQL) ou des outils liés au Web (Apache, Firefox, PHP, GIMP). Comme l'a dit Peter Mattis, l'un des deux créateurs originaux de GIMP : « Vous devez comprendre que GIMP et GTK n'ont pas été écrits pour combler des trous des logiciels existants sous la GPL et la LGPL. GIMP a commencé parce que je voulais créer une page Web. GTK a commencé parce que l'on n'était pas satisfait de Motif et que l'on voulait voir ce que ça donnerait d'écrire une boîte d'outil pour interface graphique. Ces raisons sont purement égoïstes. C'est probablement pourquoi ces projets ont tant progressé et ont finalement eu du succès. ». En d'autres termes, GIMP a été créé pour les propres besoins de ses créateurs et non pas pour adresser un marché. Le concept du logiciel libre a été également utilisé par des consortiums (comme l'OS pour mobile Symbian). Là encore on assiste à un cas de logiciel créé par ses propres utilisateurs, même si les utilisateurs ne sont plus des particuliers mais des entreprises.

Donc, la généralisation de l'utilisation des projets open source est encore très partielle et va prendre du temps, encore beaucoup de temps. En attendant, Linus a réussi à horizontaliser le système d'exploitation, un peu comme IBM a dû le faire avec son PC. Linus a gardé la maintenance du noyau, mais la grande modularité de Linux a permis de distribuer le développement. Par exemple, les pilotes de périphériques ont pu être développés par des développeurs dans leur coin (par exemple, un développeur qui avait une carte graphique non supportée par Linux). X-Window a pu être développé séparément du reste. Et Gnome au-dessus de X-Window.

La théorie et la pratique

Comme tous les phénomènes, le logiciel libre a en pratique évolué d'une manière parfois différente de ce que les gens pensaient. La liberté n'a en effet pas toujours été au rendez-vous. Si en pratique les utilisateurs peuvent maintenir eux-mêmes leur propre version des logiciels libre, dans les faits un infime pourcentage n'est prêt à le faire. Si Linux a bénéficié d'énormément de développeurs de par le monde, tel n'est pas le cas pour beaucoup des projets open source. Bien souvent, un noyau dur de développeurs écrit la plus grande partie du code (c'est encore plus le cas avec les petits projets open source). Les statistiques montrent que les programmeurs vraiment actifs sont peu nombreux et que la plupart des contributions sont très simples (ce qui ne veut pas dire qu'elles ne sont pas précieuses : l'alarme qui indique une erreur est toujours bienvenue). Pour Apache par exemple, plus de 80 % des modifications proviendraient de seulement quinze programmeurs.

En fait, la gratuité a peut-être été au moins aussi importante que l'aspect ouvert pour expliquer le succès de certains logiciels libres. Tout comme certains éditeurs commerciaux ont fermé les yeux sur le piratage dans l'espoir de voir leur logiciel se faire connaître, un logiciel en open source est facilement disponible, ce qui favorise sa propagation. Car la gratuité permet à n'importe qui d'utiliser un tel logiciel. Un développeur en entreprise peut utiliser Linux, Apache ou MySQL sans avoir à demander quelconque autorisation ni avoir autorité pour débloquer un budget.