

L'Open Source Hardware



open source hardware

**Travail de Diplôme réalisé en vue de l'obtention du Bachelor HES en
Informatique de Gestion**

par :

Jérémie Wenger

Conseiller au travail de Bachelor :

Alexandre de Banoff

Genève, le 12 Juillet 2013

Haute École de Gestion de Genève

Filière Informatique de Gestion

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre Bachelor HES en Informatique de Gestion. L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 12 Juillet 2013

Jérémie Wenger

Remerciements

En premier lieu, je tiens à remercier ma compagne Joëlle Martinet ainsi que son frère Yann Martinet pour leur soutien.

La société Lumens 8 SA et particulièrement Mr François Charles pour la mise à disposition du matériel et ses conseils dans la réalisation du projet.

Ma famille et mes amis pour lesquels j'ai été absent pendant les mois consacrés à la réalisation de ce travail.

Mr Théo Reichel de la société Complex IT sàrl pour avoir répondu à mes questions.

Sans oublier Alexandre De Banoff qui a cru en ce projet et décidé de m'accompagner dans sa réalisation. Ses conseils avisés et son soutien m'ont beaucoup aidé.

Sommaire

Il y a une quinzaine d'années l'Open Source commence à faire son apparition dans le monde de l'informatique. Pour beaucoup il s'agit d'un concept politique obscur porté par une poignée de développeurs hippies réfractaires à toute notion de propriété. Pour d'autres il s'agit d'une gamme de produits logiciels gratuits souvent difficiles à utiliser. Mais pour les initiés, il semble être une opportunité énorme de partage de connaissances et la possibilité de s'affranchir du dictat des gros éditeurs monopolisant le marché.

Aujourd'hui L'Open Source fait partie de notre quotidien il est présent dans nos téléviseurs, dans nos voitures, nos GPS, nos téléphones. Il dirige nos avions et nos trains sans que l'on y prête attention. Cette « révolution » logicielle a mis à disposition de tous des outils aujourd'hui indispensables pour la conception de certains produits.

Mais cette révolution est-elle réservée au monde logiciel ? Serait-il possible d'appliquer ces concepts à d'autres produits ?

C'est l'objectif de l'Open Hardware et ce travail tente de présenter son fonctionnement et de faire un état des lieux actuel (Juillet 2013) de ce concept. Par conséquent il se peut que les règles, principes et lois changent dans les années futures.

Résumé

Ce travail a pour but la description et l'analyse du concept de matériel libre (Open Hardware) au travers d'un travail de recherche et d'un projet de développement d'un produit basé sur des technologies Open Hardware.

L'étude commence par un aperçu de ce qu'est l'Open Source afin de bien comprendre son fonctionnement, ses lois et ses retombées tant sociales que financières.

Puis elle dresse l'état actuel de ce concept appliqué aux objets plus communément appelés Open Hardware.

Pour finir par la description et l'analyse de la conception d'un produit fini développé sur mesure.

Table des matières

Introduction	1
L'open Source	3
Historique	3
La licence logiciel	6
Les différents types de licences libres	8
Le libre et l'open source	10
Un modèle de développement (Linux, la cathédrale et le bazar)	11
Les différents modèles d'affaire (Business model)	16
<i>Le modèle de services ou de valorisation indirecte</i>	<i>18</i>
<i>Le modèle de distribution à valeur ajoutée</i>	<i>19</i>
<i>Le modèle de licence double ou licence open source commerciale</i>	<i>20</i>
<i>Le modèle de la mutualisation</i>	<i>22</i>
Quelques chiffres sur l'état actuel de l'open source	25
Le concept de l'open source en dehors du monde logiciel	31
L'open source appliqué aux objets	34
Introduction	34
En détail 35	
<i>Règles et licences</i>	<i>35</i>
<i>La communauté</i>	<i>37</i>
Quelques projets populaires	38
<i>Les imprimantes 3-D</i>	<i>38</i>
<i>L'automobile</i>	<i>39</i>
<i>Matériel électronique grand publique</i>	<i>40</i>
<i>Matériel informatique</i>	<i>40</i>
<i>Matériel médical libre</i>	<i>41</i>
<i>Agriculture</i>	<i>42</i>
Avantages et opportunités	43
Avenir et complications	45
Réalisation concrète d'un projet basé sur du matériel Libre	47
Descriptif du projet	47
<i>En résumé 47</i>	
<i>Technologies choisies</i>	<i>47</i>
Journal du projet	49
Présentation du résultat	50
Opportunités de production en série du prototype développé	52
Bilan 53	
<i>Conclusion du projet</i>	<i>53</i>
<i>Comparatif avec le milieu professionnel</i>	<i>53</i>
Conclusion	55
Lexique	58
Bibliographie	63

Liste des Figures

Figure 1 : photo de l'arduino UNO prise sur le site http://www.semageek.com	1
Figure 2 : Photo de mon premier projet : l' alertator 4000.....	1
Figure 3 : Logo de la Free software foundation prise sur leur site : http://www.fsf.org	4
Figure 4 : logo de l'Open Source - pris sur le site - http://swissmediatools.ch	5
Figure 5 : Schéma de classification des CLUF (source : Wikipedia Licence de logiciel - http://fr.wikipedia.org/wiki/Licence_de_logiciel)	6
Figure 6 : Types de licences open source (Inspiré de source : faberNovel Consulting – Septembre 2007 http://fr.slideshare.net/gillesmu/business-model-open-source).....	9
Figure 7 : Logo de linux, le pinguin (appelé Tux) est la mascotte choisie par la communauté linux – il est souvent représenté sous différentes formes source de l'image : http://www.engineering.uiowa.edu	11
Figure 8 : Les différents modèles d'affaire(source : faberNovel Consulting – Septembre 2007 http://fr.slideshare.net/gillesmu/business-model-open-source).....	17
Figure 9 : Modèle de services simple (source : faberNovel Consulting – Septembre 2007)	18
Figure 10 : Modèle de distribution à valeur ajoutée (source : Fabernovel consulting – septembre 2007) ...	19
Figure 11 : Modèle de licence double (source : faberNovel Consulting – Septembre 2007)	20
Figure 12 : Modèle de licence double (source : faberNovel Consulting – Septembre 2007).....	21
Figure 13 : Modèle de la mutualisation (source : faberNovel Consulting – Septembre 2007).....	22
Figure 15 : Répartition des entreprises par nombre de salariés (source CNLL).....	25
Figure 15 : Répartition des entreprises par chiffre d'affaire (source CNLL)	25
Figure 16 : Répartition moyenne de l'activité d'une entreprise, selon la typologie de clients (source CNLL)	26
Figure 17 : Les principaux bénéfices attendus des solutions libres, par leurs clients – (source CNLL).....	26
Figure 18 : La croissance observée en 2009 – (source CNLL)	27
Figure 19: La croissance attendue en 2010 – (source : CNLL)	28
Figure 20 : Typologies d'activité pratiquées – (source CNLL)	28
Figure 21: Entreprises contribuant à un projet Libre – (source CNLL)	29
Figure 22: un des logos de l'open data - http://usgin.org	31
Figure 23 : Logo de creative commons - http://creativecommons.org	31
Figure 24 : Logo des Ressources éducatives libres - http://www.wikimedia.org	32
Figure 25 : logo Open Access - http://www.library.vcu.edu	32
Figure 26: un des logos de l'open document (odt) - http://openclipart.org	32
Figure 27 : logo Open Society Foundations - http://www.opensocietyfoundations.org/	33
Figure 28 : logo Open Source Hardware - http://oshwlogo.com	33
Figure 29 : logo de la TAPR open hardware licence - http://www.tapr.org/ohl.html	36
Figure 30 : Logo de la licence open hardwre du CERN - http://www.ohwr.org/	36
Figure 31 : logo communément choisi par bon nombre de FabLab - source de l'image : http://myglobalbordeaux.com	37
Figure 32 : Logo de rebrap - http://rebrap.org	38
Figure 33 : Logo de Makerbot - http://www.makerbot.com	39
Figure 34 : logo de riversimple - http://www.riversimple.com	39
Figure 35 : logo de wikispeed - http://www.wikispeed.com	39
Figure 36 : logo de la société Arduino - http://www.arduino.cc	40
Figure 37 : logo de raspberry pi avec une image du produit - http://www.raspberrypi.org	41
Figure 38 : logo créé par Open Prosthetics Project - http://openprosthetics.org	41
Figure 39 : logo de Whirlwind Wheelchair - http://www.whirlwindwheelchair.org	42
Figure 40 : logo du projet OpenEEG, symbolise le gnu (animal) avec un EEG connecté à un ordinateur, le gnu est l'emblème du projet GNU à l'origine de la licence GPL - http://openeeg.sourceforge.net	42
Figure 41 : logo du projet Open Source Ecology - http://opensourceecology.org	42
Figure 42 : logo de defence distributed, à l'origine de Defcad - http://defdist.org	46
Figure 43 : Schéma du projet de télécommande multimédia pour Lumens 8.....	48

Liste des Tableaux

Tableau 1 : Synthèse des différents business models (source : faberNovel Consulting – Septembre 2007
<http://fr.slideshare.net/gillesmu/business-model-open-source>).....24

Introduction



FIGURE 1 : PHOTO DE L'ARDUINO UNO PRISE SUR LE SITE [HTTP://WWW.SEMAGEEK.COM](http://www.semageek.com)

Adepté de bricolage électronique depuis toujours je découvrais en 2010 le potentiel de nouvelles petites cartes électroniques programmables appelées ArduinoTM dont les fonctionnalités semblaient illimitées. Après quelques recherches je découvrais

un nombre impressionnant de sites internet proposant différents tutoriaux¹, libres d'accès, décrivant des projets concrétisés à partir de ces cartes. Je décidais donc de m'en procurer une afin de pouvoir tester par moi-même son potentiel. Je ne fus pas déçu.

En quelques minutes j'avais installé le nécessaire pour la faire fonctionner et faisais clignoter une petite LED² sur ma carte. Puis je me suis posé la question de ce qui pourrait être mon prochain VRAI projet, quelque chose d'utile. Je trouvais vite une idée en parcourant le web et me focalisais sur un petit système d'alarme et de déclenchement d'une prise de courant via SMS. J'achetais les composants et me lançais dans le vif du sujet.

Peu de temps après j'avais une petite boîte qui m'avertissait lorsque certaines valeurs de ses capteurs³ devenaient critiques. Elle me répondait lorsque je lui demandais son état, allumait une prise de courant lorsque je lui en donnais l'ordre et tout ceci par le biais de quelques SMS. J'étais impressionné de m'apercevoir que pour environ 160 CHF et un peu de temps j'avais réussi à faire ce que je pensais être accessible uniquement à un ingénieur en électronique hautement qualifié avec du matériel hors de prix.

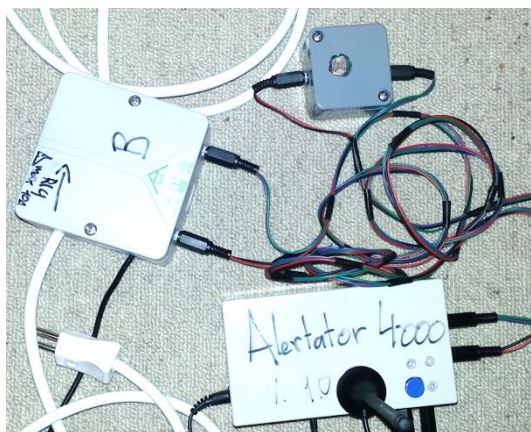


FIGURE 2 : PHOTO DE MON PREMIER PROJET : L' ALERTATOR 4000

¹ Le tutoriel désigne un guide d'apprentissage permettant à un débutant de se former de manière autonome.

² LED ou diode électroluminescente est un composant opto-électronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. (Wikipedia, Diode électroluminescente)

³ Dans ce projet : un capteur de lumière, un capteur d'humidité et un autre de température.

Je décidais donc d'aller plus loin en analysant la démarche des créateurs de ces petites cartes et là je découvrais un concept : celui d'appliquer les règles qui régissent le logiciel libre à des objets concrets, matériels. C'était simple, ça semblait évident mais il fallait y penser et surtout oser. Oser investir du temps, de l'énergie et de l'argent dans le développement de ces petites cartes pour lequel tout espoir de rentabilité semblait absurde car incompatible avec le fait d'en distribuer gratuitement les plans de conception. Contre toute attente la société ArduinoTM se porte à merveille⁴ et son chiffre d'affaire ne cesse de croître. Au lieu d'augmenter la concurrence, nuisible à son chiffre d'affaire, la distribution gratuite des plans avait accru considérablement la popularité de ces cartes ce qui a augmenté leurs ventes et avec elles le nombre d'utilisateurs. Tous entretiennent différents sites communautaires d'entraide où les plans des projets et les solutions aux problèmes rencontrés sont répertoriés, travaillant donc indirectement à l'amélioration du produit et à l'extension de la communauté.

La société ArduinoTM avait, par le biais de leurs petites cartes, créé une « brique » technologique diablement efficace et avait réussi à valoriser son produit en distribuant gratuitement ses plans.

Le concept, totalement novateur pour moi, avait éveillé ma curiosité. Existait-il d'autres projets tels que celui-ci ? Jusqu'où peut-on aller dans l'utilisation ou l'appropriation de ce type de produit afin de développer sa propre activité ? Le modèle d'affaire utilisé par la société ArduinoTM est-il un cas d'école ?

C'est à ces questions que j'ai tenté de répondre par le biais de cette analyse et de la concrétisation du projet mentionné plus bas. Étant un concept relativement novateur et basé sur les règles de l'Open Source j'ai jugé important de commencer par une analyse de ce qu'est l'Open Source, des règles qui le régissent et des différents modèles d'affaires utilisés dans ce type de marché.

⁴ De Septembre 2005 à Aout 2011 ils avaient vendus 314'461 cartes. En Août 2011 ils comptaient plus de 200 distributeurs à travers le monde et étaient utilisés par une 30aine d'universités. - <http://fr.slideshare.net/arduinoteam/open-source-hardware-summit-speech-2011>

L'open Source

Historique

Les ordinateurs des années 1960 étaient livrés avec des logiciels accompagnés de leur code source⁵ que les clients pouvaient modifier et étendre. Il aurait en effet été impossible de vendre un ordinateur sans son programme d'accompagnement et la plupart des clients estimaient trop aléatoire de faire fonctionner un logiciel dont ils ne pouvaient vérifier les caractéristiques internes. Les acquéreurs obtenaient donc sur simple demande le code source (y compris des systèmes d'exploitation⁶) qu'ils pouvaient modifier à leur convenance (⁷).

L'hétérogénéité des systèmes et leur incompatibilité rendaient impossible la valorisation autonome d'applications sur le marché. L'industrie informatique confondait donc le matériel et le logiciel en une seule et même économie et parmi les utilisateurs le partage du code était une pratique courante. Des praticiens témoignent que certains bâtissaient des programmes en se basant sur le travail des uns et des autres en partageant du code. L'essentiel des échanges était alors réalisé par le biais de supports matériels puisqu'il n'y avait pas encore d'infrastructure réseau permettant l'échange en ligne.

La standardisation des infrastructures matérielles, des systèmes et des usages rendit possible la valorisation des logiciels ce qui créa une scission entre l'industrie des composants (hardware⁸) et celle des logiciels (⁹). Afin de conserver un certain rendement sur le fruit de leur travail l'industrie du logiciel se mis à « masquer » le fonctionnement de ses produits en fournissant à l'utilisateur uniquement une version « Langage machine »¹⁰ de ses applications. La gratuité des logiciels soulevait des

⁵ Le code source est un texte qui représente les instructions qui doivent être exécutées par un microprocesseur. Le code source est généralement écrit dans un langage de programmation permettant ainsi une meilleure compréhension par des humains. Une fois le code source écrit, il permet de générer une représentation binaire d'une séquence d'instructions code binaire (ou langage machine) exécutables par un microprocesseur. (Wikipedia, Code source)

⁶ Un système d'exploitation (OS pour Operating System en anglais) désigne un logiciel qui joue le rôle d'intermédiaire entre un utilisateur, ses programmes et le matériel sur lequel il est installé (exemple : Windows, Mac OS, Linux)

⁷ Wikipedia Open Source - https://fr.wikipedia.org/wiki/Open_source

⁸ Hardware signifie matériel, dans ce contexte il s'agit de matériel informatique

⁹ Benkeltoum, Nordine (2011), Gérer et comprendre l'open source

¹⁰ Le langage machine, ou code machine, est la suite de bits qui est interprétée par le processeur d'un ordinateur exécutant un programme informatique. C'est le langage natif d'un processeur, c'est-à-dire le seul qu'il puisse traiter. Il est composé d'instructions et de données à traiter codées en binaire. (Wikipedia, Language machine)

problèmes de concurrence déloyale et le partage du code source provoquait des problèmes de maintenance qui, fournie par les constructeurs, était rendue difficile à cause des modifications effectuées par les équipes informatiques. Les échanges de code source n'étaient alors plus pratiqués que par les universitaires et les chercheurs.

Parallèlement, la création d'Arpanet¹¹ facilitait le partage de fragments de codes des chercheurs et universitaires. Petit à petit il devint ce que l'on appelle aujourd'hui Internet¹². Les échanges de données se démocratisant, le partage de code s'étendit au-delà du milieu universitaire ou de la recherche pour toucher n'importe quel développeur. Cette phase de démocratisation s'est largement fondée sur le développement du net, à la fois le fruit et l'engrais du développement de l'open source. Le TCP/IP¹³, le HTML¹⁴ et BIND¹⁵ sont quelques-uns des standards libres sur lesquels internet est basé illustrant bien le fait que le « réseau des réseaux » est le produit du phénomène open source.

Afin de protéger les auteurs de ces « bouts de codes partagés » et garantir la gratuité et l'accessibilité de leurs travaux, la Free Software Foundation¹⁶ fondée par Richard Stallman¹⁷ en 1982 développa la notion de logiciel libre avec ses règles et ses licences. Le logiciel libre était né et avec lui une multitude de projets « communautaires ».



FIGURE 3 : LOGO DE LA FREE SOFTWARE FOUNDATION PRISE SUR LEUR SITE : [HTTP://WWW.FSF.ORG](http://www.fsf.org)

¹¹ ARPANET ou Arpanet (acronyme anglais de « Advanced Research Projects Agency Network », souvent typographié « ARPAnet ») est le premier réseau à transfert de paquets développé aux États-Unis par la DARPA. Le projet fut lancé en 1969 et la première démonstration officielle date d'octobre 1972. (Wikipedia, ARPANET)

¹² Internet est un système d'interconnexion de machines et constitue un réseau informatique mondial, utilisant un ensemble standardisé de protocoles de transfert de données. (Wikipedia, Internet)

¹³ Le TCP/IP (Transmission Control Protocol/Internet Protocol) est un protocole de communication conçu en 1974 afin de contourner la disparité des systèmes.

¹⁴ L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. (Wikipedia, HTML)

¹⁵ BIND (Berkeley Internet Name Domain) est un système d'adressage des sites internet permettant de ne pas avoir à entrer l'adresse au format numérique mais sous forme de noms de domaines. (Benkeltoum, 2011)

¹⁶ La Free Software Foundation (FSF) (littéralement « Fondation pour le logiciel libre »), est une organisation américaine à but non lucratif fondée par Richard Stallman le 4 octobre 1985, dont la mission mondiale est la promotion du logiciel libre et la défense des utilisateurs. (Wikipedia, Free Software Foundation)

¹⁷ Richard Matthew Stallman, est un programmeur et militant du logiciel libre. Il est à l'origine du projet GNU et de la licence publique générale GNU connue aussi sous l'acronyme GPL, qu'il a rédigée avec l'avocat Eben Moglen. Il a popularisé le terme anglais copyleft. (Wikipedia, Richard Stallman)

L'exemple le plus célèbre de ce type de projet est le projet "Linux"¹⁸ lancé en 1991 par Linus Torvalds¹⁹, un étudiant de l'université d'Helsinki. Celui-ci souhaitait développer un système d'exploitation libre compatible Unix²⁰ : il invita tous les programmeurs volontaires à participer au projet et reçut le soutien de centaines de participants.

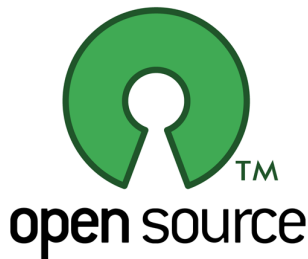


FIGURE 4 : LOGO DE L'OPEN SOURCE -
PRIS SUR LE SITE -
[HTTP://SWISSMEDIATOOLS.CH](http://swissmediatools.ch)

Plus tard, suite à une scission avec la communauté du logiciel libre, différents acteurs de la communauté (Bruce Perens²¹ et Eric Steven Raymond²²) développaient le mouvement Open Source²³. Ce mouvement a pour but d'une part de permettre une plus grande coopération entre les entreprises et la communauté et d'autre part une diffusion plus large des logiciels.

En 1998, un événement important marqua l'entrée des entreprises dans l'open source : la libération de NavigatorTM par NetscapeTM²⁴. La firme avait libéré son navigateur internet en réaction à la stratégie de MicrosoftTM qui embarquait gratuitement Internet Explorer²⁵ dans son système d'exploitation. À cette même période, la bulle spéculative liée aux NTIC²⁶ avait de sérieuses répercussions sur les start-up émergées lors du développement d'internet provoquant des faillites en chaîne. En revanche les communautés de l'open source ne furent pas touchées par l'éclatement de cette bulle

¹⁸ Linux est un système d'exploitation libre créé en 1991, sur l'initiative de Linus Torvalds.

¹⁹ Linus Benedict Torvalds est un informaticien américano-finlandais. Il est connu pour avoir créé en 1991 le noyau Linux dont il continue à diriger le développement. (Wikipedia, Linus Torvalds)

²⁰ Unix, est un système d'exploitation multitâche et multi-utilisateur créé en 1969, conceptuellement ouvert et fondé sur une approche par laquelle il offre de nombreux petits outils, chacun doté d'une mission spécifique. (Wikipedia, Unix)

²¹ Bruce Perens est un ancien leader du projet Debian, le premier à succéder au fondateur Ian Murdock, et un co-fondateur de l'Open Source Initiative. Il est détenteur de la marque Open source. (Wikipedia, Bruce Perens)

²² Eric Steven Raymond, est un hacker américain célèbre à qui l'on doit notamment la popularisation du terme open source par opposition à free software (ou logiciel libre). Il est l'auteur du livre La Cathédrale et le Bazar qui résume ses points de vue sur ce sujet. (Wikipedia, Eric Raymond)

²³ La désignation open source s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire la possibilité de libre redistribution, d'accès au code source et aux travaux dérivés. (Wikipedia, Open Source)

²⁴ Société éditeur de logiciels fondée en 1994, rachetée en 1998 par AOLTM puis dissolue en 2003.

²⁵ Logiciel de navigation internet (navigateur) développé par la société MicrosoftTM

²⁶ Nouvelles Technologies de l'Information et de la Communication : regroupent les techniques utilisées dans le traitement et la transmission des informations, principalement de l'informatique, de l'Internet et des télécommunications.

car, pour la plupart, n'avaient pas de modèle d'affaires et étaient encore moins cotées en bourse. ⁽²⁷⁾

Ces éléments augmentèrent la crédibilité et la popularité des projets Open Source et les sociétés éditrices de logiciels se mirent à participer activement aux communautés existantes d'utilisateurs-développeurs (CUD)²⁸ s'inspirant de leurs modèles de développement pour leurs propres intérêts. Elles sont aujourd'hui passées d'observateurs passifs au statut d'acteur ou d'investisseur.

La licence logiciel

Le logiciel, au même titre que les productions artistiques, est considéré comme une œuvre de l'esprit. Il est régi par le droit d'auteur ou le droit à la propriété intellectuelle. Pour pouvoir l'utiliser l'auteur doit céder une partie de ses droits. Cette cession se fait par l'intermédiaire d'une licence :

Une licence de logiciel est un contrat par lequel le titulaire des droits d'auteur sur un programme informatique définit avec son cocontractant (exploitant ou utilisateur) les conditions dans lesquelles ce programme peut être utilisé, diffusé ou modifié.

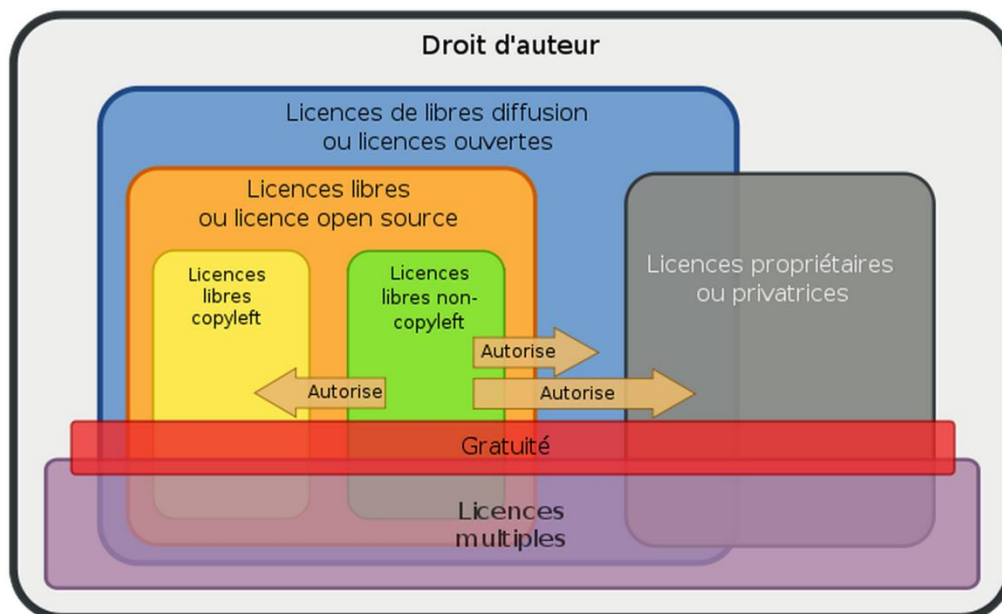


FIGURE 5 : SCHÉMA DE CLASSIFICATION DES CLUF (SOURCE : WIKIPEDIA LICENCE DE LOGICIEL - [HTTP://FR.WIKIPEDIA.ORG/WIKI/LICENCE_DE_LOGICIEL](http://fr.wikipedia.org/wiki/Licence_de_logiciel))

²⁷ Benkeltoum, Nordine (2011), Gérer et comprendre l'open source

²⁸ CUD ou Communauté d'utilisateurs développeurs désignent une communauté internet d'utilisateurs et/ou de développeurs d'un logiciel libre. Chaque projet de logiciel libre a sa propre communauté d'utilisateurs développeurs. (Benkeltoum, 2011)

De manière globale existe 3 grandes familles de licences logicielles :

Logiciel dans le domaine public, comme pour tout ce qui est dans le domaine public, le produit est considéré comme appartenant à tout le monde. C'est une caractéristique juridique qui n'a pas besoin de licence du fait que le logiciel n'a aucun ayant droit (auteur) officiel. Théoriquement, tout logiciel tombe dans le domaine public une fois les droits d'auteur échus. Toutefois, la durée de protection des droits d'auteur est bien plus longue que le plus ancien des logiciels, par exemple soixante-dix ans après la mort de l'auteur dans tous les pays de l'Union européenne. On ne trouve donc dans le domaine public que des logiciels qui y ont été placés suite à une renonciation aux droits. ⁽²⁹⁾

De plus le fait qu'un produit tombe dans le domaine public n'impose pas le fait de distribuer ses plans de fabrications ou, dans le cas d'un logiciel, son code source. C'est pourquoi, dans certains cas, un programme exécutable (code machine) peut être dans le domaine public sans que son code source soit disponible.

Logiciel propriétaire est un logiciel dont la duplication, la modification ou l'usage est limité. Ceci se matérialise par des limitations légales, matérielles, ou logicielles dans son utilisation, sa diffusion ou sa modification.

Ces limitations sont choisies par les ayants droits (auteurs) et sont souvent encadrées par un contrat de licence utilisateur final (CLUF)³⁰, nommé alors licence propriétaire. ⁽³¹⁾.

Logiciel sous licence libre est un logiciel qui garantit à l'utilisateur les quatre libertés suivantes qui sont :

- la liberté d'exécuter le programme pour tous les usages.
- La liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins.
- La liberté de redistribuer des copies, avec ou sans contrepartie financière.

²⁹ Wikipedia, Logiciel libre

³⁰ Le terme Contrat de Licence Utilisateur Final (CLUF) est une traduction du terme anglais EULA, End User License Agreement. Il s'agit d'un contrat liant une personne installant un logiciel affecté par ce type de licence sur un/son ordinateur et l'éditeur du logiciel. La plupart d'entre elles, que l'on dénomme aussi licences de logiciels propriétaires, limitent le nombre de machines sur lesquelles on peut installer le logiciel, le nombre d'utilisateurs qui peuvent utiliser le logiciel, et contiennent d'autres limitations qui ne sont pas inhérentes à la technologie. (Wikipedia, License de logiciel)

³¹ Wikipedia, logiciel propriétaire

- La liberté d'améliorer le programme et de publier ses améliorations. Nous verrons ci-après les différentes variantes de licence libre.

Les différents types de licences libres

Richard Stallman (décrit comme le père des logiciels libres) considère que le droit d'auteur, en interdisant d'aider ses amis par une simple copie ou en privilégiant l'auteur aux dépens du monde entier, est nuisible pour la société. Pour illustrer le principe du logiciel libre face aux autres logiciels, il compare volontiers cela à la recette de cuisine d'un gâteau :

- Selon le principe du libre : vous avez obtenu légalement cette recette par n'importe quel moyen (revue, bouche à oreille...). Vous avez le droit de redistribuer cette recette à qui vous voulez et vous pouvez la modifier puis la redistribuer comme il vous plaît.
- Selon le principe du logiciel non libre : vous n'avez pas accès à la recette, mais uniquement au gâteau déjà fait. Vous ne pouvez manger le gâteau que dans une seule cuisine, et personne d'autre que vous ne peut en manger. Quand bien même la recette serait fournie avec le gâteau, toute copie ou modification serait interdite⁽³²⁾.

Cette métaphore résume bien la différence fondamentale entre un logiciel libre et un logiciel propriétaire et les problèmes qui en découlent :

Serait-ce juste, une fois la recette connue, de vendre des gâteaux sans en distribuer la recette ou sans en mentionner l'auteur ? En partant de cette recette légèrement modifiée par mes soins aurais-je le droit de me déclarer comme en étant l'auteur ?

C'est pourquoi la communauté du logiciel libre a développé le Copyleft³³ (aussi appelé licence « virale » à cause de l'aspect « contagieux » de sa mise en application) permettant d'assurer la pérennité des « droits » accordés à l'utilisateur d'un produit malgré sa modification ou son appropriation par autrui sans pour autant rentrer dans le domaine public et ainsi en protéger l'auteur.

³² Metycea

³³ Le copyleft est l'autorisation donnée par l'auteur d'un travail soumis au droit d'auteur (œuvre d'art, texte, programme informatique ou autre) d'utiliser, d'étudier, de modifier et de copier son œuvre, dans la mesure où cette autorisation reste préservée. L'auteur refuse donc que son travail puisse évoluer avec une restriction du droit à la copie. De ce fait, le contributeur apportant une modification (correction, ajout, réutilisation, etc.) est contraint de redistribuer ses propres contributions avec les mêmes conditions d'utilisation que l'original. Autrement dit, les créations réalisées à partir d'éléments sous copyleft héritent de facto ce copyleft. (Wikipedia, Copyleft)

L'enjeu étant non pas de contrôler la diffusion du logiciel dans un but de rentabilité, contrairement au logiciel de type « propriétaire », mais de protéger les droits accordés à un utilisateur pour l'utilisation d'un logiciel malgré son évolution ou son utilisation dans d'autres produits.

Cependant cette autorisation impose à tout nouveau composant associé ou vendu avec un logiciel soumis au Copyleft l'adoption de celui-ci. Ce qui est très contraignant car cela élimine toute possibilité de rentabilité sur le développement de produits logiciels complémentaires ou développés grâce à un produit soumis au Copyleft.

C'est pour cela qu'il existe une multitude de licences libres (plus d'une centaine) qui modèrent (ou non) l'imposition du Copyleft aux produits dérivés sous certaines conditions. Nous n'aborderons ici qu'une partie de celles-ci.

Il existe 3 familles de licences libres :

- **Copyleft fort** : la redistribution ou l'intégration du logiciel avec ou sans modification peut se faire mais toujours sous la licence initiale (GNU GPL³⁴).
- **Copyleft faible** : la redistribution du logiciel avec ou sans modification peut se faire mais toujours sous la licence initiale. Il est cependant possible pour des produits dérivés ou complémentaires que le code de ces composants soit ajouté sous d'autres licences (éventuellement propriétaires) (PGP³⁵, GPG³⁶).
- **Sans Copyleft** : la redistribution du logiciel avec ou sans modification peut se faire sous une autre licence, éventuellement propriétaire. (New BSD³⁷)

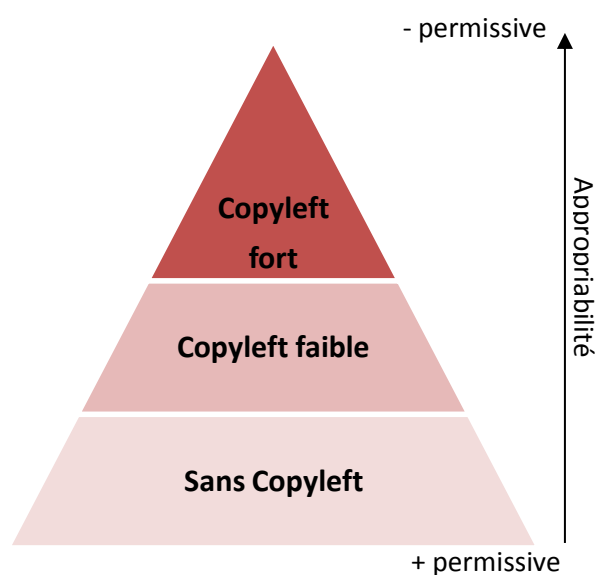


FIGURE 6 : TYPES DE LICENCES OPEN SOURCE (INSPIRÉ DE SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007
HTTP://FR.SLIDESHARE.NET/GILLESMU/BUSINESS-MODEL-OPEN-SOURCE)

³⁴ <http://www.gnu.org/licenses/gpl.html>

³⁵ <http://openpgp.org/>

³⁶ <http://www.gnupg.org/index.fr.html>

³⁷ <http://www.freebsd.org/copyright/license.html>

D'un côté, une licence soumise au Copyleft (GNU GPL) permet de s'assurer que tout restera dans le domaine libre. C'est ce qui fait le succès de cette licence (3/4 des projets open source sont sous GNU GPL).

De l'autre, une licence non soumise au copyleft (new BSD) permet une grande liberté d'action pour celui qui en apporte des modifications.

Ainsi, on peut partir d'une base BSD pour redistribuer en GPL mais on ne peut pas partir d'une base GPL pour redistribuer en BSD. ⁽³⁸⁾

Le libre et l'open source

Les termes « logiciel libre » et « Open Source » sont, pour la majorité des gens, similaires. Cependant il existe de vives polémiques.

Le logiciel libre, défendu par la « Free Software Foundation », se définit par les libertés accordées à l'utilisateur. Un logiciel est considéré comme libre (au sens de la FSF ¹²) s'il confère à son utilisateur quatre libertés qui sont :

- La liberté d'exécuter le programme, pour tous les usages
- La liberté d'étudier le fonctionnement du programme et de l'adapter à ses besoins (ce qui implique la mise à disposition du code source)
- La liberté de redistribuer des copies du programme (ce qui implique la possibilité aussi bien de donner que de vendre des copies)
- La liberté d'améliorer le programme et de distribuer ces améliorations au public (ce qui implique la mise à disposition du code source).

Si la nature du logiciel libre facilite et encourage son partage, ce qui tend à le rendre gratuit, elle ne s'oppose pas pour autant à sa rentabilité principalement via des services associés. Les rémunérations sont liées par exemple à des travaux de création, de développement, de mise à disposition et de soutien technique.

Le logiciel libre ne doit cependant pas être confondu avec les freewares (gratuciels) ³⁹ dans le sens où ils ne correspondent pas à ces quatre libertés car ne permettent pas de modifications du code (code source non fourni). ⁽⁴⁰⁾

³⁸ Metycea

³⁹ Un freeware, graticiel ou gratuiciel, est un logiciel propriétaire distribué gratuitement sans toutefois conférer à l'utilisateur certaines libertés d'usage associées au logiciel libre. (Wikipedia, Freeware)

⁴⁰ Wikipedia, Logiciel libre

L'Open Source, défendu par l'Open Source Initiative⁴¹, défend une politique jugée plus adaptée aux réalités économiques et techniques. Le terme Open Source a été suggéré afin de lever l'ambiguïté sémantique du mot anglais free qui signifie libre au sens de « liberté », mais également libre au sens de « libre accès, gratuité » et par le biais de cette rectification sémantique, indiquer aux utilisateurs qu'un logiciel a un coût.

En pratique, la plupart des licences de l'open source satisfont aux critères du libre selon la Free Software Foundation, les différentes subtilités qui les distinguent étant principalement d'ordre philosophique et commercial.

En résumé, la communauté du logiciel libre accuse l'open source d'être mû par la dynamique et les ressources financières de multinationales, l'opposant au logiciel libre mû par des idéaux d'ordre philosophique et politique. (⁴²).

Il s'agit d'une « guerre » idéologique entre les partisans du copyleft « coûte que coûte » et ceux pour des solutions sous copyleft « faible » ou sans copyleft.

Un modèle de développement (Linux, la cathédrale et le bazar)

Ce chapitre est inspiré des expériences d'Eric S. Raymond relatées dans son article « The Cathedral and the Bazaar » sorti en 1998 (traduit ici par Sébastien Blondeel). Il présente le modèle de développement utilisé pour le projet Linux, ce modèle n'est pas systématiquement appliqué à tous les projets open source cependant cette méthode est très populaire et aujourd'hui un grand nombre de projets open source l'utilisent.

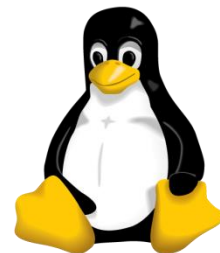


FIGURE 7 : LOGO DE LINUX, LE PINGUIN (APPELÉ TUX) EST LA MASCOTTE CHOISIE PAR LA COMMUNAUTÉ LINUX – IL EST SOUVENT REPRÉSENTÉ SOUS DIFFÉRENTES FORMES SOURCE DE L'IMAGE : [HTTP://WWW.ENGINEERING.UIOWA.EDU](http://www.engineering.uiowa.edu)

En 1993 Eric S. Raymond est impliqué dans des projets de développement sous Unix⁴³, convaincu qu'il existe une certaine complexité critique au-delà de laquelle une approche centralisée, hiérarchisée, organisée est indispensable (Il nomme ce mode de

⁴¹ L'Open Source Initiative est une organisation dévouée à la promotion des logiciels open source. (Wikipedia, Open Source Initiative)

⁴² Wikipedia, Open Source Initiative

⁴³ Unix est un système d'exploitation multitâche et multi-utilisateur créé en 1969, conceptuellement ouvert et fondé sur une approche par laquelle il offre de nombreux petits outils, chacun doté d'une mission spécifique. (Wikipedia, Unix)

développement « cathédrale »), que les logiciels les plus importants (comme les systèmes d'exploitation ou les gros outils) doivent être conçus de la sorte :

« Soigneusement élaborés par des sorciers isolés ou des petits groupes de mages travaillant à l'écart du monde, sans qu'aucune version bêta ne voie le jour avant que son heure ne soit venue. » (Raymond, 1998)

Ayant été l'un des premiers contributeurs du projet GNU⁴⁴ en 1980, Eric pense connaître le mode de développement du logiciel libre. Il s'implique dans le projet Linux (dirigé par Linus Torvalds) qui promeut un style de développement radicalement différent :

« Distribuez vite et souvent, déléguiez tout ce que vous pouvez déléguer, soyez ouvert jusqu'à la promiscuité. » (Linus Torvalds)

À l'opposé de la construction de cathédrales, silencieuse et pleine de vénération, la communauté Linux paraissait plutôt ressembler à un bazar, grouillant de rituels et d'approches différentes à partir duquel un système stable et cohérent ne pourrait apparemment émerger que par une succession de miracles. Le fait que ce style du bazar semblait fonctionner, et bien fonctionner, fut un choc supplémentaire. Alors que j'apprenais à m'y retrouver, je travaillais dur, non seulement sur des projets particuliers, mais encore à essayer de comprendre pourquoi le monde Linux, au lieu de se disloquer dans la confusion la plus totale, paraissait au contraire avancer à pas de géant, à une vitesse inimaginable pour les bâtisseurs de cathédrales. (Extrait de « la Cathédrale et le Bazar » Raymond, 1998)

Dans le courant de l'année 1996, Eric commence à comprendre ce mode de fonctionnement et décide de le mettre à l'épreuve dans le cadre d'un projet personnel. C'est l'histoire de ce projet qu'il décrit dans son article « The Cathedral and the Bazaar ».

Lors de cette expérience il met en évidence plusieurs « leçons » apprises au cours du développement de son projet en utilisant ce modèle de développement communautaire (qu'il nomme bazar). Ces leçons mettent en évidence les qualités de ce type de développement mais également certaines attitudes à avoir pour assurer le succès du projet dans ce cadre particulier. Le modèle Bazar se base sur la mise à contribution via

⁴⁴ GNU est un système d'exploitation libre lancé en 1984 par Richard Stallman et maintenu par le projet GNU. Son nom est un acronyme récursif qui signifie en anglais « GNU's Not UNIX » (littéralement, « GNU n'est pas UNIX »). Il reprend les concepts et le fonctionnement d'UNIX. (Wikipedia, GNU)

internet d'une communauté d'utilisateurs et de développeur mus par le même objectif : concrétiser le projet.

Tout bon logiciel commence par gratter un développeur là où ça le démange :

Les programmeurs actifs dans le monde Linux sont d'une manière ou d'une autre concernés par le programme qu'ils développent. Ils ont le choix de contribuer ou non au projet. À l'inverse des développeurs conventionnels payés pour produire des programmes dont ils n'ont pas besoin. Cet intérêt augmente leur motivation et permet un développement de qualité.

Les bons programmeurs savent quoi écrire. Les grands programmeurs savent quoi réécrire (et réutiliser) : Le fait d'opérer dans un environnement où le code source est partagé permet une grande source d'inspiration et la réutilisation de ce qui a déjà été fait.

Quand un programme ne vous intéresse plus, votre dernier devoir à son égard est de le confier à un successeur compétent : Un des principes implicite du développement communautaire est qu'il faut savoir lâcher un projet lorsqu'il ne nous intéresse plus. De sorte qu'il reste porté par des gens motivés et qu'il puisse continuer à évoluer en respectant le travail déjà effectué.

Traiter vos utilisateurs en tant que co-développeurs est le chemin le moins semé d'embûches vers une amélioration rapide du code et un débogage efficace : Le développement de type bazar fonctionne par la création d'une communauté d'utilisateurs et de développeurs concernés par un projet logiciel. Cette communauté autogérée portera le développement du projet. Plus la communauté est grande et motivée et plus le logiciel évolue vite et bien. Les optimisations se développent et les bugs⁴⁵ sont corrigés. Il est donc important de respecter et mettre en valeur le travail de chacun : un utilisateur chevronné qui répertorie les problèmes rencontrés aura tout autant d'importance qu'un co-développeur.

Distribuez tôt. Mettez à jour souvent (Linux Torvalds) : L'objectif est de sortir les nouvelles versions dès que possible afin d'impliquer au maximum la communauté. Dans le modèle « cathédrale » on attendra qu'une version soit stable avant de la sortir.

⁴⁵ En informatique, un bug (de l'anglais bug, « insecte ») ou bogue est un défaut de conception d'un programme informatique à l'origine d'un dysfonctionnement. Ce nom vient du tout premier incident informatique qui a été causé par un insecte. (Wikipedia, Bug (informatique))

Mettre à disposition les nouvelles versions dès que possible valorise les personnes impliquées, leur montre que le projet évolue et permet un recensement et une correction rapide des bugs.

Étant donné qu'il y a suffisamment d'observateurs, tous les bugs sautent aux yeux : Il n'est pas courant que ce soit la personne qui a identifié le problème qui le corrige. Le développeur sait comment utiliser son produit il est donc peu probable qu'il ait un comportement imprévu et ingérable pour l'application. En revanche l'utilisateur aura une approche plus « intuitive », plus personnelle et testera les fonctionnalités dans d'autres contextes sans se soucier du processus de fonctionnement du logiciel. Cette attitude place plus facilement l'application dans une situation « imprévue » et permet un recensement des problèmes plus efficace. En augmentant le nombre d'utilisateurs « testeurs » du logiciel on augmente considérablement la probabilité que l'un d'entre eux tombe sur un bug. Il en est de même pour les failles de sécurité.

Eric S. Raymond déclare même que, pour lui, c'est la différence fondamentale entre le style de développement de la cathédrale et celle du bazar :

« Dans la programmation du point de vue de la cathédrale, les bugs et les problèmes de développement représentent des phénomènes difficiles, ennuyeux, insidieux, profonds. Il faut à une poignée de passionnés des mois d'observations minutieuses avant de bien vouloir se laisser convaincre que tous les bogues ont été éliminés. D'où les longs intervalles séparant les mises à jour, et l'inévitable déception quand on se rend compte que la mise à jour tant attendue n'est pas parfaite.

Dans le point de vue bazar, d'un autre côté, vous supposez qu'en général, les bugs sont un phénomène de surface ou, en tout cas, qu'ils sautent rapidement aux yeux lorsqu'un millier de co-développeurs avides se précipitent sur toute nouvelle mise à jour. C'est pourquoi vous mettez à jour souvent afin de disposer de plus de corrections, et un effet de bord bénéfique est que vous avez moins à perdre si de temps en temps, un gros bogue vous échappe. » (Raymond, 1998)

Il est presque aussi important de savoir reconnaître les bonnes idées de vos utilisateurs que d'avoir de bonnes idées vous-même. C'est même préférable, parfois : Pour qu'un développement de type bazar fonctionne il faut savoir valoriser la communauté. Il est important de garder à l'esprit ce que le porteur du projet doit à celle-ci et garder une certaine humilité.

Tout outil doit être utile par rapport aux utilisations qu'il a été prévu d'en faire. Mais on reconnaît un outil vraiment excellent au fait qu'il se prête à des usages totalement insoupçonnés : Le développement communautaire offre de précieuses suggestions quand à de nouvelles fonctionnalités ou améliorations. Contrairement au développement de type « cathédrale » contraint à un certain rendement qui évitera toute fonctionnalité « non-rentable » ou incompatible avec la politique marketing.

Au travers de ses expériences, Eric S. Raymond nous montre comment le mode de construction « bazaar » fonctionne et comment l'utiliser au mieux en respectant et en entretenant une communauté d'utilisateurs-développeurs. Il présente de manière concrète les atouts de ce « mode de création communautaire » rendu possible grâce à l'émergence d'internet.

Les qualités de linux ne sont plus à démontrer et au-delà de sa gratuité il représente la preuve concrète qu'un logiciel de qualité peut être obtenu à partir d'un modèle de développement de type « bazar ». Aujourd'hui plusieurs entreprises se sont orientées vers le développement communautaire afin de garantir la pérennité, augmenter la popularité et réduire leur frais de développement de leur produit. Accentuant leurs revenus sur des services annexes générés par l'utilisation du logiciel (voir chapitre « les différents modèles d'affaire »).

Outre les aspects de performances, sécurités et robustesses, le développement collaboratif garanti une évolution du logiciel dans l'intérêt de l'utilisateur (puisque'il a la possibilité d'améliorer ou de suggérer des améliorations).

« En réalité, je pense que la bidouille la plus ingénieuse de Linus, et celle qui a eu le plus de conséquences, n'a pas été la construction du noyau de Linux en lui-même, mais plutôt son invention du modèle de développement de Linux. » (Raymond, 1998)

Nordine Benkeltoum⁴⁶ va plus loin en nous proposant une vision « historique » du développement collaboratif. Il sépare l'innovation collective de la production collective

⁴⁶ Benkeltoum, Nordine (2011), Gérer et comprendre l'open source

et met en évidence le fait que certaines avancées technologiques sont dues à un effort collectif. Par exemple certaines évolutions liées à l'industrie du fer, de par le partage d'information (conséquence d'une situation dans laquelle le partage d'informations était profitable à tous), sont imputables à la collectivité. On peut également relever certaines collaborations entre entreprises dans le cadre d'efforts de guerre (japonnais, américain, allemand etc.).

De même, concernant la production collective, il fait le lien entre ce type de communautés et la manufacture dispersée, utilisée depuis longtemps, qui consiste à faire produire à domicile le même type de produit à plusieurs personnes dispersées géographiquement. Le fruit de leur travail étant racheté par un intermédiaire. Ce type de production, permet de réduire les frais car il n'y a pas de grosse structure de production.

Malgré le fait que la construction d'un logiciel (écriture du code source) concerne l'innovation et la production. On peut constater qu'à la fois l'innovation et la production collective sont loin d'être des nouveautés mais des modèles utilisés depuis longtemps.

Les différents modèles d'affaire (Business model)

Les modèles d'affaires dans le monde du logiciel sont en pleine évolution. Alors que les logiciels libres et gratuits font partie du paysage. Des entreprises commerciales, de plus en plus nombreuses, se tournent vers eux comme moyen de faire connaître leur marque et de diffuser leurs produits. D'obscures start-ups deviennent mondialement reconnues du jour au lendemain en diffusant des logiciels Open Source. Ces entreprises n'entendent pas pour autant de renoncer à leurs ambitions commerciales, bien au contraire. L'enjeu devient alors de capitaliser sur la nouvelle notoriété de leur marque et de vendre les services associés ou le développement de composants spécifiquement adaptés aux demandes des clients, le portage sur des plates-formes exotiques, ou propriétaires ou l'intégration avec d'autres logiciels. Il existe également des modèles de diffusion à double licence : une licence Open Source de type GPL, et

une licence commerciale, nécessaire pour bénéficier des services supplémentaires.
(⁴⁷)

Dans ce chapitre sont présentés quatre types de modèles d'affaires généraux utilisés par différentes sociétés actives dans le domaine de l'Open Source.

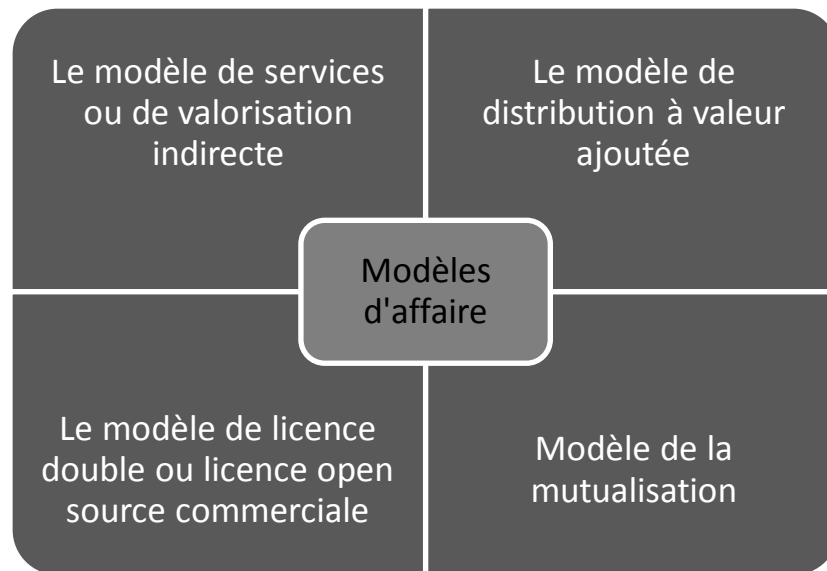


FIGURE 8 : LES DIFFÉRENTS MODÈLES D'AFFAIRE(SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007
[HTTP://FR.SLIDESHARE.NET/GILLESMU/BUSINESS-MODEL-OPEN-SOURCE](http://fr.slideshare.net/gillesmu/business-model-open-source))

⁴⁷ <http://www.cetic.be/L-Open-Source-comme-modele-d>

Le modèle de services ou de valorisation indirecte

Ce modèle repose sur les services complémentaires comme la maintenance, les développements spécifiques, le conseil et la formation. La société de service n'est pas nécessairement celle qui développe le logiciel. Ce modèle est le plus courant de l'open source. (LinuxTM, Spike source^{TM 48}, Typo3^{TM 49}, Compiere^{TM 50}).

Quelques exemples de services proposés :

Formation :

- Techniciens (certifications)
- Utilisateurs

Surveillance :

- Contrôle des algorithmes
- Détection des bugs
- Surveillance des problèmes de sécurité

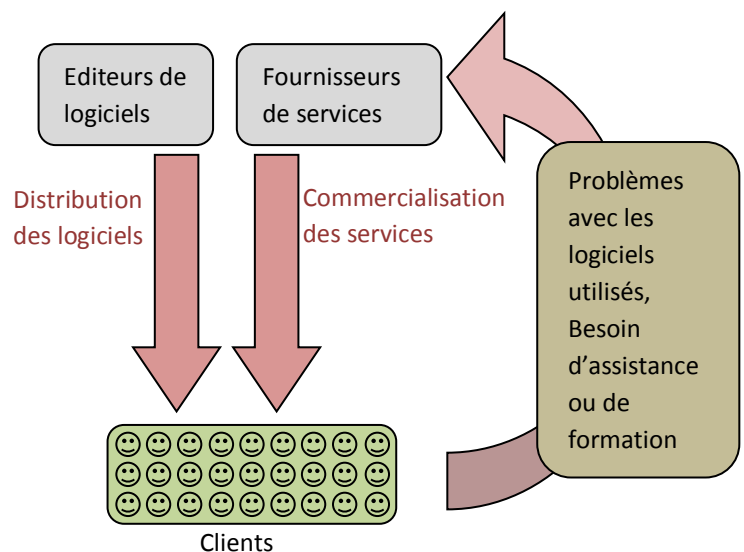


FIGURE 9 : MODÈLE DE SERVICES SIMPLE (SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007)

Tests et garantie :

- Aide à l'intégration de logiciels testés
- Garantie d'interopérabilité (avec de nouveaux systèmes ou avec le système actuel)

Assistance technique :

- Aide (support)
- Création de documentation d'aide ou de support
- Création de patches en cas de problèmes

La valorisation indirecte peut également se faire sous forme de partenariats publicitaires, dans ce cas ce n'est plus le client qui paie le service mais une autre

⁴⁸ SpikeSource est une société spécialisée dans le test logiciel automatisé. (Wikipedia, SpikeSource)

⁴⁹ TYPO3 est un système de gestion de contenu libre écrit en PHP. (Wikipedia, SpikeSource)

⁵⁰ Compiere est un progiciel de gestion intégré (PGI) et gestion de la relation client (GRC) open source pour les Petites et moyennes entreprises (PME) dans la distribution et le service. (Wikipedia, Compiere)

société partenaire qui, en échange, gagne une certaine visibilité grâce à la popularité du logiciel (ex : firefox^{TM 51} avec googleTM)(⁵²).

Le modèle de distribution à valeur ajoutée

Dans ce modèle, les logiciels open source ne sont pas créés par les entreprises qui facturent leurs services : ils existent déjà et sont rassemblés dans une version standard qui peut être téléchargée, préinstallée sur un ordinateur ou vendue sur un support physique. La valeur ajoutée correspond à l'assemblage cohérent d'un ensemble de logiciels libres autour du système d'exploitation. (RedHat Entreprise Linux^{TM 53}, Ubuntu^{TM 54}, SUSE^{TM 55})

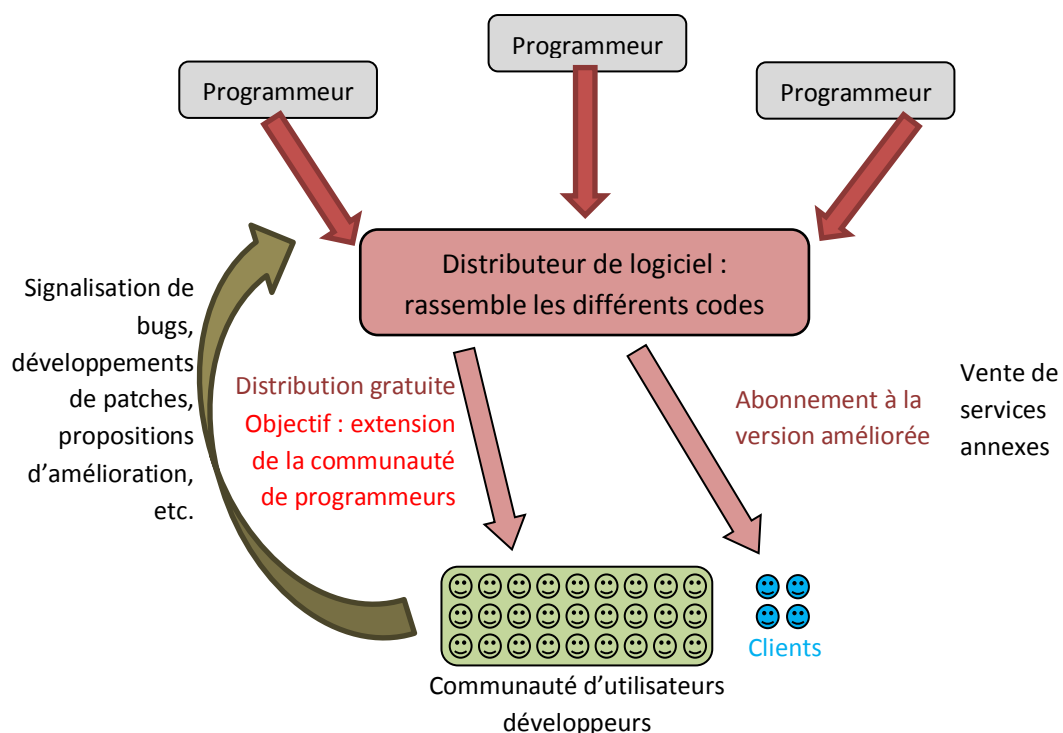


FIGURE 10 : MODÈLE DE DISTRIBUTION À VALEUR AJOUTÉE
(SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007)

⁵¹ Mozilla Firefox est un navigateur Web libre et gratuit, développé et distribué par la Mozilla Foundation avec l'aide de milliers de bénévoles grâce aux méthodes de développement du logiciel libre/open source. (Wikipedia, Mozilla Firefox)

⁵² <http://www.pcinpact.com/news/67789-firefox-google-mozilla-partenariat.htm>

⁵³ Red Hat Enterprise Linux (souvent abrégé RHEL) est une distribution Linux produite par Red Hat et orientée vers le marché commercial et les serveurs d'entreprise. (Wikipedia, Red Hat Enterprise Linux)

⁵⁴ Ubuntu est un système d'exploitation libre fondé sur Linux et commandité par la société Canonical et une marque déposée par cette même société. (Wikipedia, Ubuntu)

⁵⁵ SUSE est une distribution Linux majeure, d'origine allemande et principalement développée en Europe. (Wikipedia, SUSE)

Le modèle de licence double ou licence open source commerciale

Consiste à proposer une seule version du logiciel dont l'utilisateur choisira la licence en fonction de ses besoins. S'il ne veut pas donner le code source de son produit (qu'il aura constitué à l'aide du composant) il devra payer une licence. S'il souhaite intégrer le composant dans une solution open source, alors il utilisera la version gratuite. Les entreprises ne souhaitant pas donner le code source de leurs produits devront opter pour une licence commerciale payante. (Qt library^{TM 56}, MySQL^{TM 57})

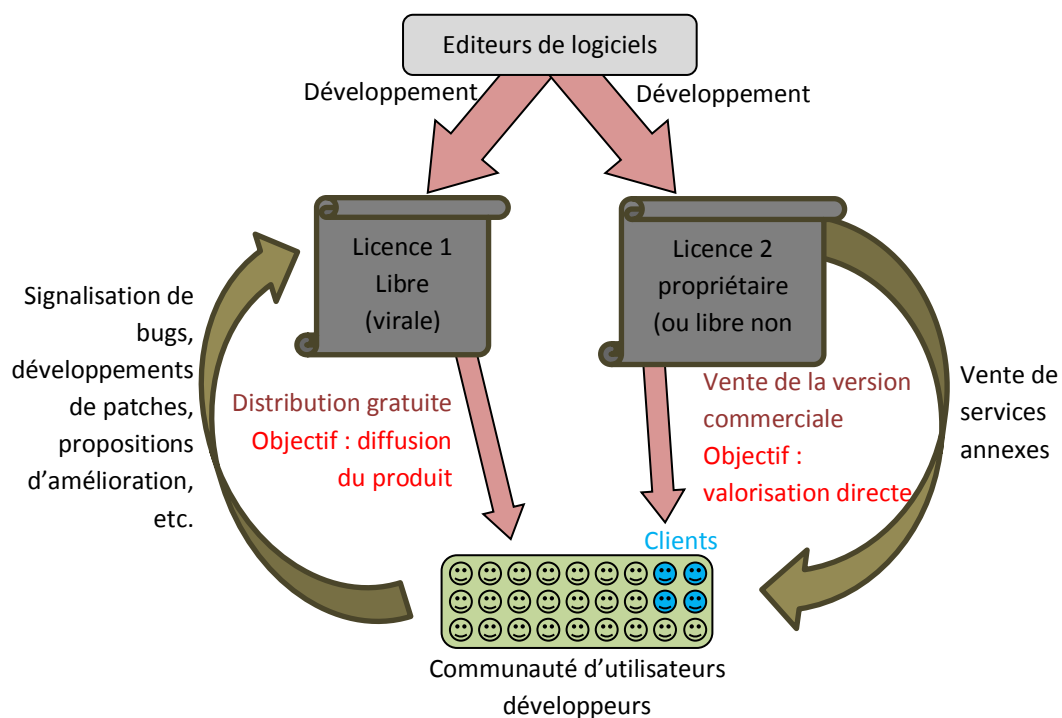


FIGURE 11 : MODÈLE DE LICENCE DOUBLE (SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007)

⁵⁶ Qt est une API orientée objet. Qt offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. (Wikipedia, Qt)

⁵⁷ MySQL est un système de gestion de base de données (SGBD). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde. (Wikipedia, MySQL)

Une variante de ce modèle consiste à commercialiser un produit complémentaire : Le produit est vendu en deux versions sous deux licences différentes. La version non-libre se destine à des besoins spécifiques et intègre des éléments additionnels. La version libre vise à satisfaire des besoins non-évolus et est doté de fonctionnalités basiques. Il peut s'agir également de la commercialisation des versions les plus récentes, les anciennes étant distribuées sous licence libre. (Novell Evolution^{TM 58}, SugarCRM^{TM 59})

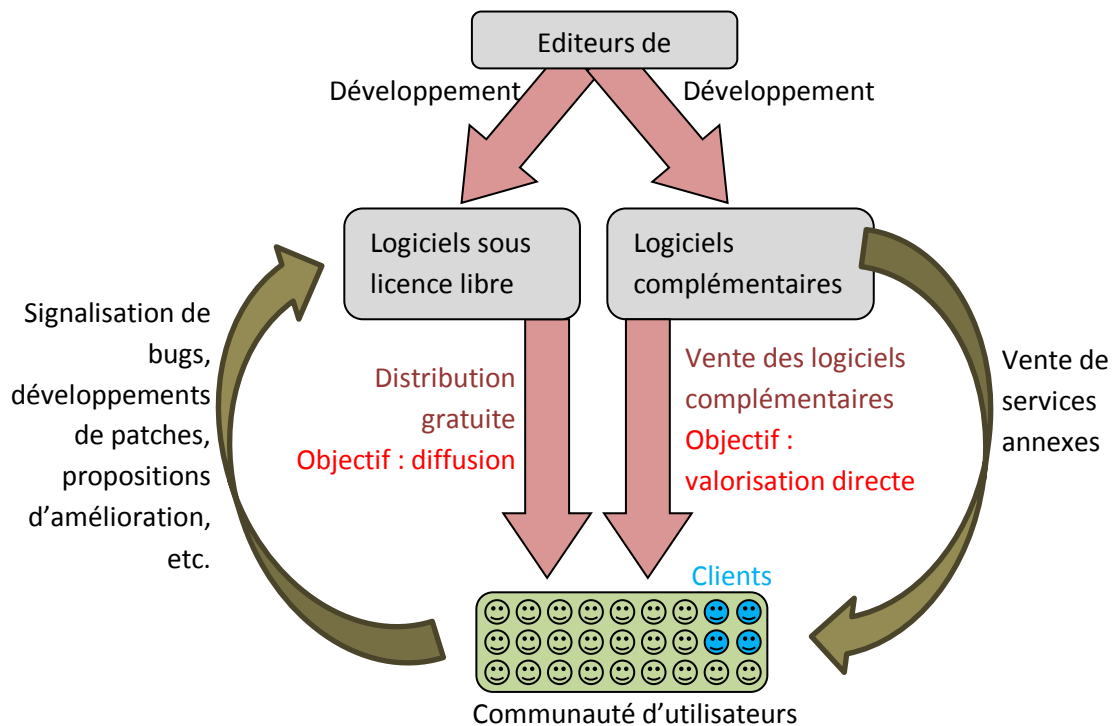


FIGURE 12 : MODÈLE DE LICENCE DOUBLE (SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007)

⁵⁸ Evolution (ou Novell Evolution) est un logiciel de groupe de travail libre. Il sert de client de messagerie, carnet d'adresses, agenda, portail d'information. (Wikipedia, Evolution (logiciel))

⁵⁹ SugarCRM est une solution Open Source de gestion de la relation client éditée par la société américaine SugarCRM. (Wikipedia, SugarCRM)

Le modèle de la mutualisation

Le modèle de la mutualisation consiste à développer une version « simple » du produit pour ensuite développer des extensions à la demande des clients. Les clients payent pour un module spécifique et bénéficient, en plus de ce module, de ceux développés pour les autres clients. Ainsi les clients deviennent investisseurs. Pour faciliter le développement de modules coûteux, l'entreprise peut créer une communauté de clients, mettant en commun leurs ressources pour financer le développement du module. Il est fréquent qu'une entreprise open source développe des modules à la demande d'un intégrateur⁶⁰ ou qu'un consortium d'entreprises ayant besoin d'un même produit s'unisse afin de le développer de manière mutualisée et partager les frais.

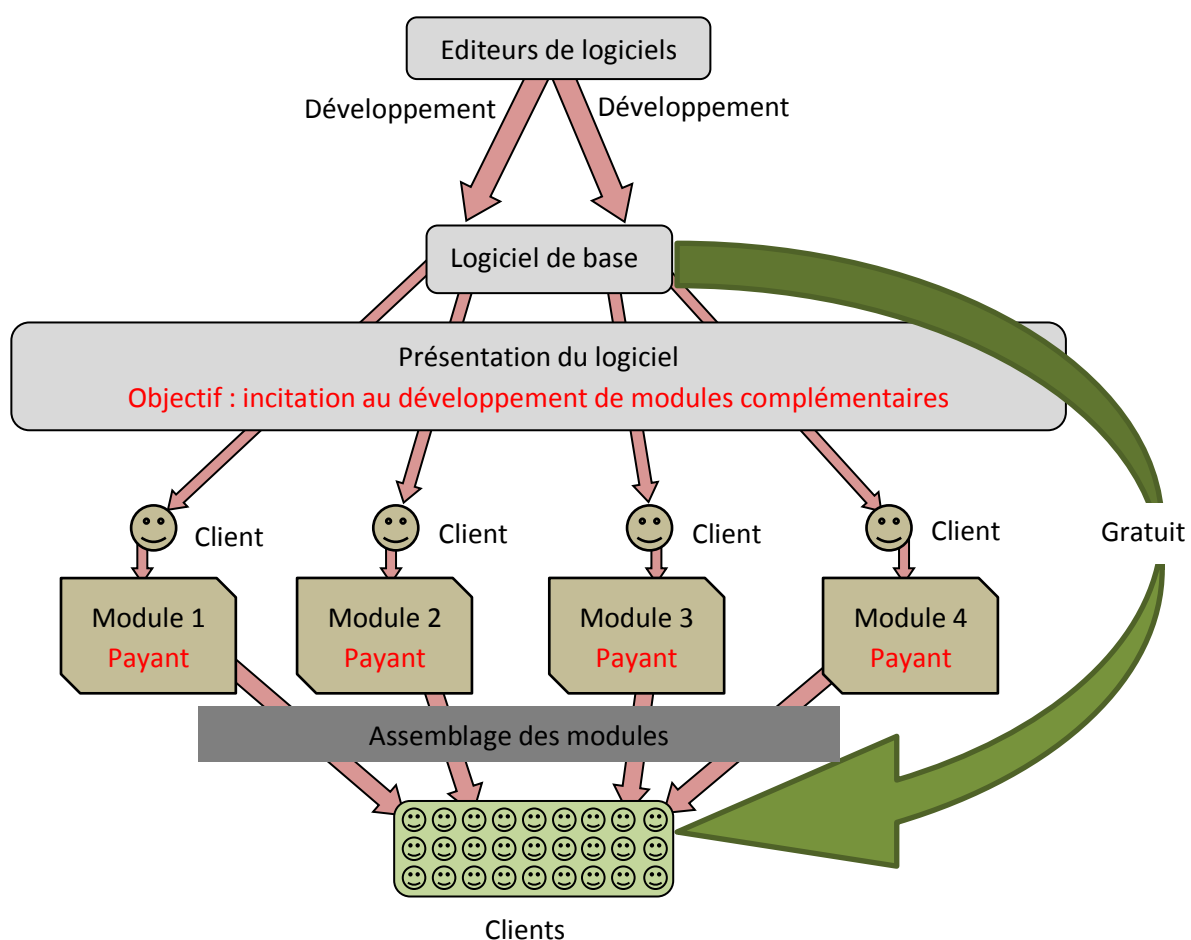


FIGURE 13 : MODÈLE DE LA MUTUALISATION (SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007)

⁶⁰ Un intégrateur (dans ce contexte) est une société chargée d'assembler différents logiciels afin de réaliser une application

Le modèle de la mutualisation ne s'applique que dans des conditions spécifiques :

- Lorsqu'il s'agit de développer des solutions pour des besoins très ciblés permettant de préempter rapidement le marché et de limiter la concurrence.
- Lorsqu'il s'agit d'un produit complexe pouvant donner lieu au développement de nombreux modules annexes.
- Lorsque le rythme d'obsolescence des solutions développées est élevé.
- Lorsqu'un consortium d'entreprise partage un intérêt commun pour un même produit logiciel.

Nous avons passé en revue quatre grandes catégories de modèles d'affaires de l'open source. Cependant ces modèles évoluent et nous pouvons trouver presque autant de variantes qu'il y a de sociétés sur ce type de marché. Chaque produit ayant sa popularité, son marché et sa manière d'être diffusé. Par exemple : Une société qui édite un logiciel très populaire comme un navigateur internet ou un système d'exploitation grand public peut aujourd'hui difficilement espérer vendre son produit. Il faut alors, pour ces sociétés, s'orienter vers un modèle de services ou de valorisation indirecte. En revanche un éditeur de logiciels professionnels ou d'outils de développements pourra se permettre une valorisation directe sur la vente d'une double licence.

Le tableau ci-dessous résume les différents modèles d'affaires :

Modèle	Offre de produits	Proposition de valeur	Principales entreprises
Offre de service	Support pour tout type de logiciel open source ou logiciel développé en interne.	Tests, certifications, intégrations de logiciels (LAMP), offre de patches, mises à jour et assistance technique modulable	Spikesource, SourceLabs, Optaros, Bearstech, Openlogic ; Open Cascade, Altic, Nuxeo, Core-Techs, Pilot Systems
Distribution à valeur ajoutée	Intégration de différents code sources, offre de patches et mises à jour uniquement pour le ou les logiciel(s) distribué(s)	Tests et garantie, offre de patches et assistance technique modulable	Red Hat, Novell SUSE, Mandriva, Mostick
Licence double	Développement d'un logiciel ou d'une « brique logicielle » d'entreprise (CRM, ERP, CMS, ...) de ses patches et de ses mises à jour	Version commerciale du logiciel contenant tout ou partie du code de la version gratuite, assistance technique modulable, envoi de patches et mises à jour et/ou personnalisation du logiciel aux besoins de l'entreprise	MySQL, JBoss, SugarCRM, Optaros, Alfresco, Sun, Wengo, ExoPlatform
Mutualisation	Développement d'un logiciel d'entreprise et de ses compléments pour un « pool » de clients	Services spécifiques, totalement adaptés aux besoins de l'entreprise, modulables et pour un coût très inférieur aux prix de marché	OpenTrust, AF83, Emencia

TABEAU 2 : SYNTHÈSE DES DIFFÉRENTS BUSINESS MODELS (SOURCE : FABERNOVEL CONSULTING – SEPTEMBRE 2007)

Quelques chiffres sur l'état actuel de l'open source

Selon la dernière étude du CNLL⁶¹ (janvier 2013)⁶² sur l'état actuel de l'Open Source :

L'industrie de l'open source en France représente

- 2.5 Milliards d'Euros (en 2011) soit 6% du marché total des logiciels et services.
- Connait une croissance de 30% par an.
- Concerne 30'000 emplois et 300 entreprises.

Dans ces 300 entreprises

- 68% étaient en croissance en 2012 et
- 83% prévoient la croissance en 2013

Une étude réalisée en 2010 (en France) par le même institut met en évidence :

Que les entreprises françaises du Logiciel Libre sont majoritairement de petites tailles, puisque la moitié d'entre elles comptent moins de 5 salariés, 84% comptent moins de 20 salariés et 78% réalisent un chiffre d'affaire de moins d'un million d'Euros annuel.

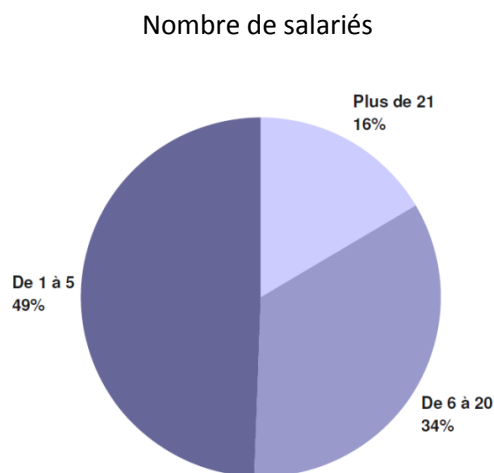


FIGURE 15 : RÉPARTITION DES ENTREPRISES PAR NOMBRE DE SALARIÉS (SOURCE CNLL)

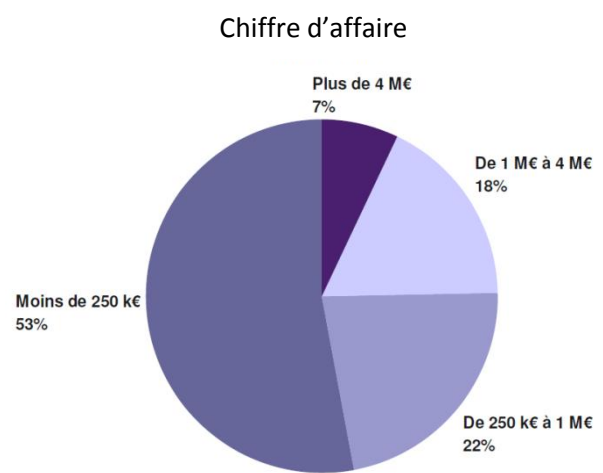


FIGURE 15 : RÉPARTITION DES ENTREPRISES PAR CHIFFRE D'AFFAIRE (SOURCE CNLL)

⁶¹ Conseil national du logiciel libre (France) - <http://cnll.fr/>

⁶² <http://www.cnll.fr/static/pdf/cnll-carto-07.pdf>

La répartition moyenne de l'activité des entreprises de l'Open Source se représente comme suit :

Répartition moyenne par typologie de clients

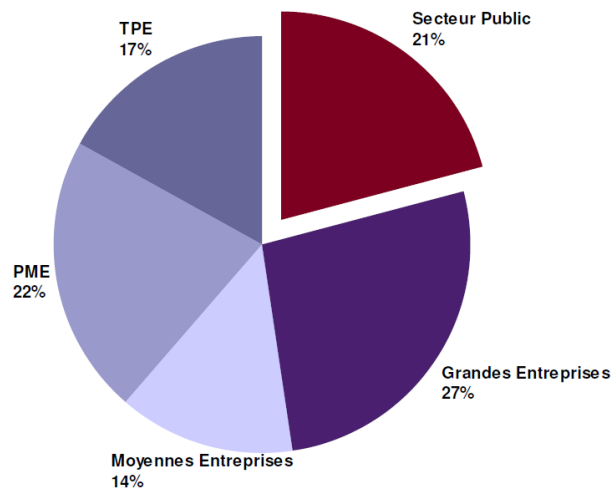


FIGURE 16 : RÉPARTITION MOYENNE DE L'ACTIVITÉ D'UNE ENTREPRISE, SELON LA TYPOLOGIE DE CLIENTS (SOURCE CNLL)

On relève que les plus gros clients sont majoritairement des grandes entreprises (ici une entreprise est considérée comme grande dès lors qu'elle emploie plus de 2000 personnes) et que ce type de client est même plus conséquent que les institutions publiques.

Et pourquoi ces clients se tournent vers des solutions libres ?

Bénéfices attendus par les clients

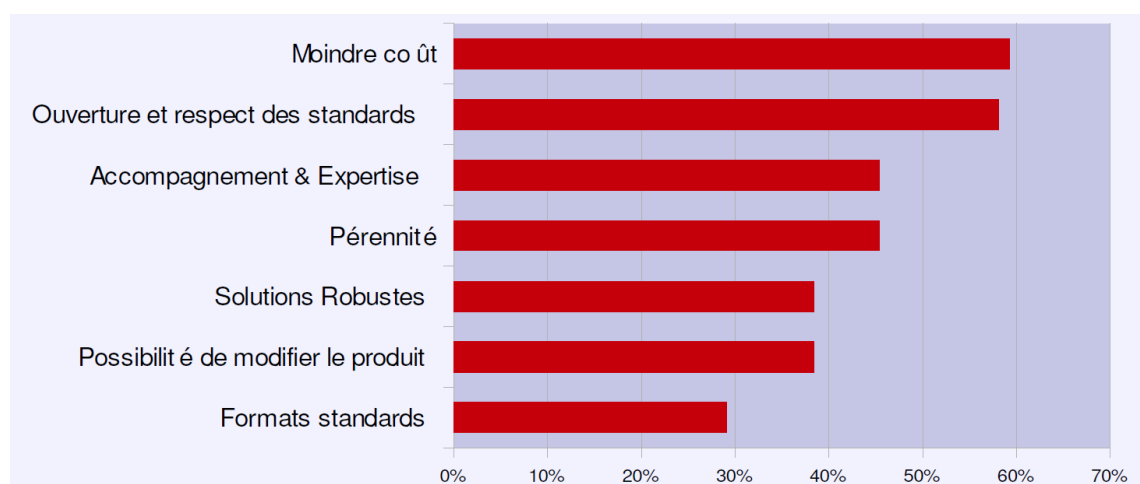


FIGURE 17 : LES PRINCIPAUX BÉNÉFICES ATTENDUS DES SOLUTIONS LIBRES, PAR LEURS CLIENTS – (SOURCE CNLL)

On remarque ici que le coût est l'argument prépondérant, puis vient l'ouverture et la compatibilité et ensuite les arguments autour de la qualité du logiciel (pérennité, robustesse, modifiabilité etc.).

Un des éléments en faveur de ce type de marché est que la plupart de ces entreprises n'ont majoritairement pas subi la crise :

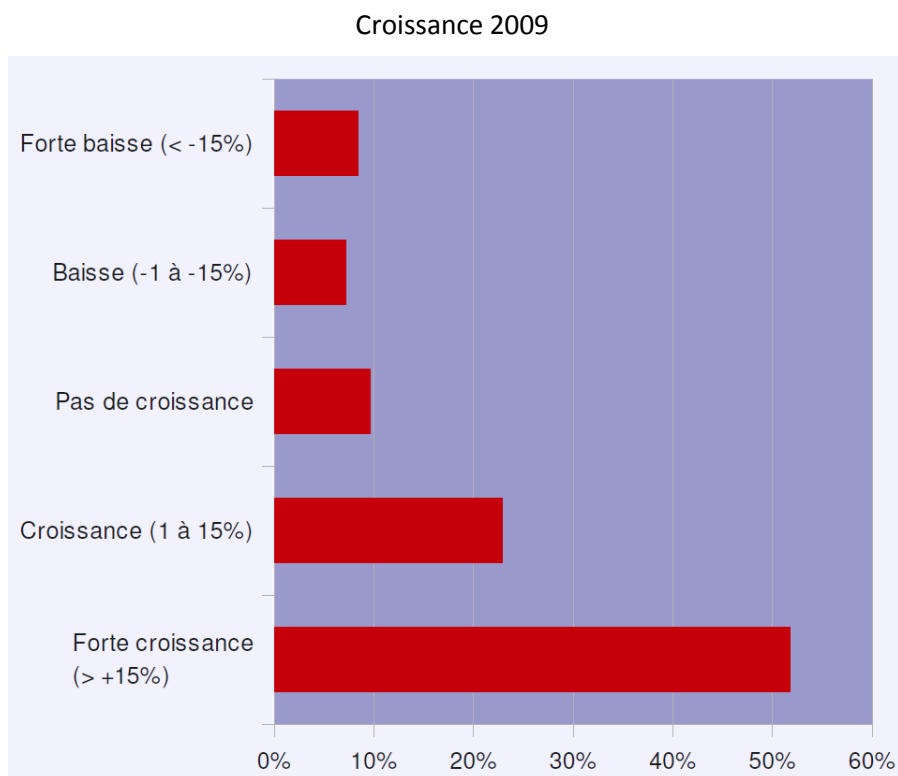


FIGURE 18 : LA CROISSANCE OBSERVÉE EN 2009 – (SOURCE CNLL)

De plus, toujours en 2010, la majorité d'entre elles prévoyaient une forte croissance (ce qui a été confirmé par les études, malheureusement pas encore publiées totalement).

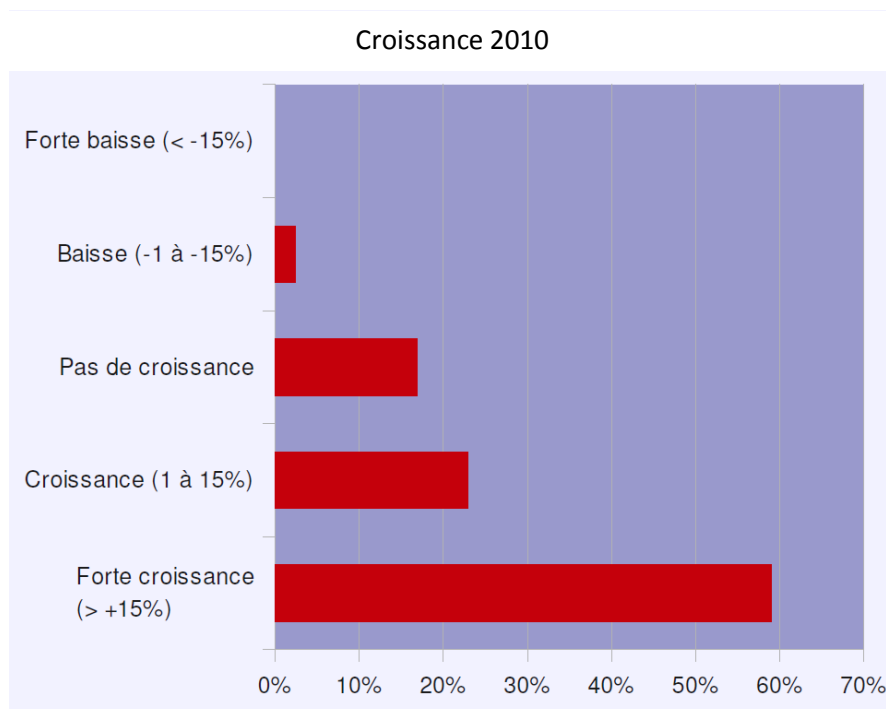


FIGURE 19: LA CROISSANCE ATTENDUE EN 2010 – (SOURCE : CNLL)

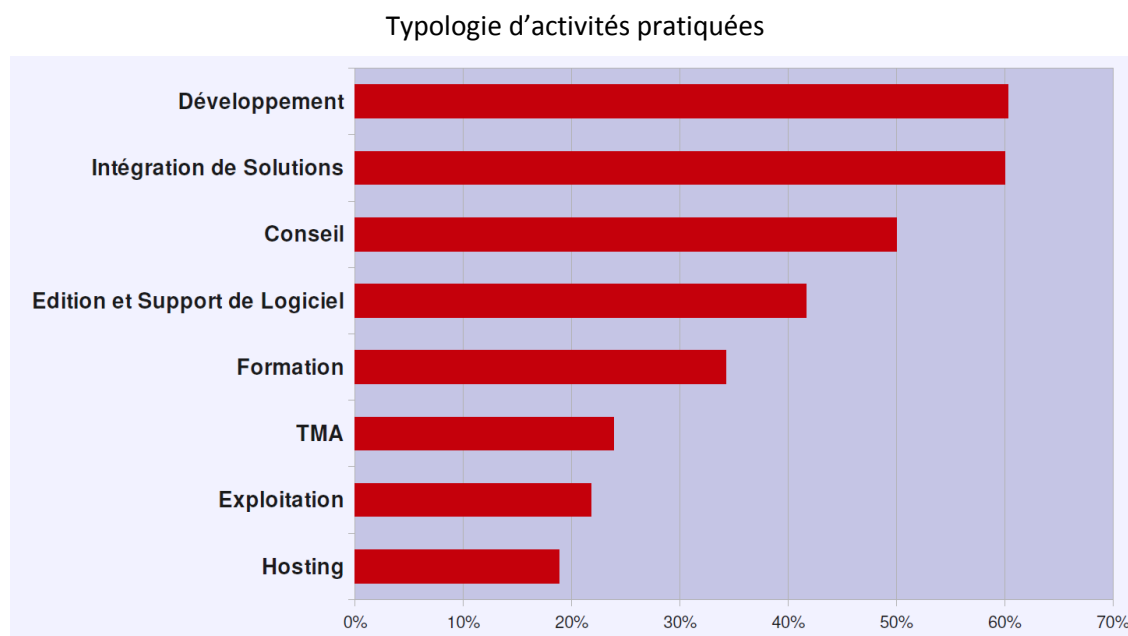


FIGURE 20 : TYPOLOGIES D'ACTIVITÉ PRATIQUÉES – (SOURCE CNLL)

Dans chacun des cas, il s'agit d'activités dans lesquelles l'entreprise a estimé réaliser plus de 15% de son activité. On remarque que le développement, l'intégration de solutions et le conseil occupent sont les activités principales de ce type d'entreprises.

Contrairement aux idées reçues pas toutes ces entreprises participent à des projets de logiciel libre.

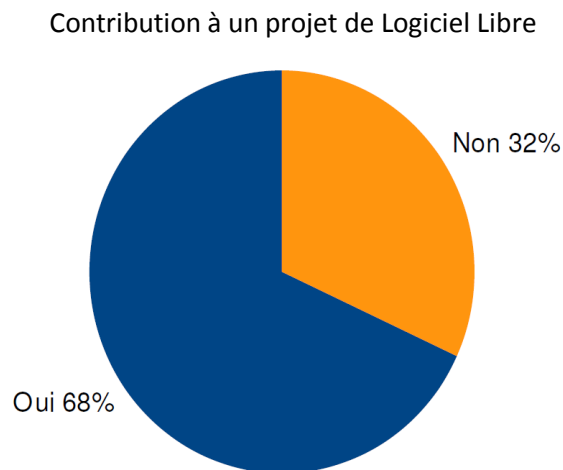


FIGURE 21: ENTREPRISES CONTRIBUANT À UN PROJET LIBRE – (SOURCE CNLL)

L'open source s'est aujourd'hui largement implanté dans l'industrie du logiciel, il fait partie intégrante de notre quotidien. Au-delà des statistiques, qui placent l'argument financier en tête de liste, il existe également d'autres éléments prépondérants dans le choix d'une solution libre :

- **La légèreté** : plusieurs constructeurs s'orientent vers de petits noyaux pour leurs systèmes embarqués⁶³. À l'inverse du logiciel privatif où « l'alourdissement » des nouvelles solutions pousse à la consommation (en imposant au client de renouveler son matériel pour pouvoir utiliser le nouveau produit). Beaucoup de versions légères⁶⁴ d'outils ou de systèmes d'exploitation libres continuent à être entretenus afin de consommer le moins de ressources possible et de fonctionner sur un maximum d'installations.

⁶³ Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé. (Wikipedia, Système embarqué)

⁶⁴ Dont l'exécution nécessite peu de ressources

- **La fiabilité et la sécurité** : La plupart des serveurs web d'aujourd'hui fonctionnent sur Linux et l'émergence des services de Cloud⁶⁵ accroît la demande de serveurs, fiables, sécurisés et performants. Certaines idées reçues voudraient qu'un logiciel dont le code source est ouvert serait forcément moins sécurisé qu'un logiciel de type propriétaire. Ces préjugés sont compréhensibles, animent d'intenses débats sur la toile et les tests concernant le sujet sont souvent ininterprétables car peu d'entre eux sont objectifs. Cela dit il serait faux de considérer évident le fait que le partage du code source rend nécessairement une application plus vulnérable. En effet il est dans l'intérêt de tous d'avoir des solutions fiables et les failles de sécurité sont également corrigées le plus rapidement possible sur les logiciels libres. De plus, comme mentionné précédemment, la popularité de certains systèmes les rend nécessairement plus exposés à des attaques. Il est donc important d'étudier les solutions au cas par cas.
- **l'indépendance technologique** : certains états ou ONG ne voulant pas d'une dépendance technologique liée à des éditeurs issus de grandes puissances (principalement américains) choisissent des solutions open source pour traiter leurs données sensibles (militaires ou étatiques).

Aujourd'hui beaucoup de sociétés s'appuient sur des technologies libres pour développer leurs produits. Par exemple Free^{TM 66} utilise linux pour sa Freebox^{TM 67}, Synology^{TM 68} également pour ses produits de stockage et la popularité de ces sociétés réside dans le fait de proposer des produits innovants. Quelles innovations ces entreprises auraient pu proposer si elles n'avaient pas bénéficié de la liberté d'action que permet le logiciel libre ? À quels prix seraient vendus leurs produits s'ils avaient dû payer différentes licences ?

L'open source est plus qu'une simple histoire de gratuité. Au-delà de l'aspect militant ou engagé il s'agit de considérer un produit comme un bien d'intérêt général qu'il faut entretenir et faire évoluer. De constituer des « briques » technologiques utilisables par tous afin de favoriser l'innovation et le développement de nouvelles activités (commerciales ou non).

⁶⁵ Le Cloud computing est une nouvelle manière de fournir et d'utiliser les aptitudes des systèmes informatiques basée sur les nuages (cloud en anglais). Un nuage est un parc de machines, d'équipement de réseau et de logiciels maintenu par un fournisseur, que les consommateurs peuvent utiliser en libre-service via Internet. (Wikipedia, Cloud computing)

⁶⁶ Free, entreprise filiale du groupe Iliad, est l'un des principaux fournisseurs d'accès à Internet en France.

⁶⁷ La Freebox est un appareil électronique fourni par le fournisseur d'accès à Internet français Free à ses abonnés haut débit.

⁶⁸ Synology est une société spécialisée dans les serveurs de stockage réseau (NAS).

Le concept de l'open source en dehors du monde logiciel

Il y a deux grandes idées dans l'open source : Celle du partage de l'information dans un but d'amélioration de la connaissance et de la compétence « commune » et celle du développement collaboratif (celle-ci dépend de la première) dans le but de répartir le travail, utiliser au mieux les différentes compétences de chacun et disposer de différentes approches/points de vue. Ces deux grandes idées ont commencé avec le développement commun de produits logiciels grâce à internet (permettant une communication rapide et un échange de données « gratuit » entre les acteurs d'un projet).

Le produit logiciel constitue le « produit pilote » parfait pour ce genre d'expérimentation car il est particulier dans le sens où il est dématérialisé. Son stockage et sa production de manière industrielle (copie) nécessite très peu de frais. Il suffit de compiler le code et le mettre à disposition sur un support physique ou internet pour obtenir un produit utilisable. Il est donc « facile » de mener un projet à bout avec très peu de moyens sans prendre de risque.

Aujourd'hui les produits Open Source font partie de notre quotidien, que ce soit dans nos maisons ou dans les entreprises. Ces produits ont fait leurs preuves et ils démontrent que la méthode utilisée pour les développer fonctionne.

Ce modèle de développement, ses règles, ses licences devraient donc être applicables à des projets dans d'autres domaines que l'informatique. Ici quelques exemples de mouvements ou projets inspirés des principes et du modèle Open Source :



FIGURE 22 : UN DES LOGOS DE L'OPEN DATA -
[HTTP://USGIN.ORG](http://usgin.org)

L'**Open Data** est un « mouvement » ou une

philosophie visant à rendre les données numériques publiques accessibles à tous en s'affranchissant des

restrictions sur le droit d'accès et de réutilisation. Il s'agit là de maximiser le partage d'information et d'augmenter la transparence.



FIGURE 23 : LOGO DE CREATIVE COMMONS -
[HTTP://CREATIVECOMMONS.ORG](http://creativecommons.org)

En s'inspirant des licences de logiciels libres et du mouvement Open Source la fondation **Creative**

Commons⁶⁹ propose une solution alternative légale aux personnes souhaitant libérer leurs œuvres des droits de propriété intellectuelle standards, jugés trop restrictifs. Ces licences s'appliquent à tout contenu soumis au droit d'auteur.

L'**Open Educational Resources** (en français : ressources éducatives libres (REL)) désigne un mouvement mondial lancé par des fondations, universités, enseignants et pédagogues visant à créer et distribuer via internet des ressources éducatives (cours, manuels, logiciels éducatifs, etc.) libres et gratuites. Elles sont notamment mises en œuvre dans des cours en ligne ouverts et massifs (MOOC ou Massive Open Online Course). Le concept a été inspiré par les concepts connexes comme le Creative Commons, Open Source, données ouvertes et Accès ouvert, et les étend à inclure des exposés et autres didacticiels.⁽⁷⁰⁾



FIGURE 24 : LOGO DES RESSOURCES ÉDUCATIVES LIBRES - [HTTP://WWW.WIKIMEDIA.ORG](http://www.wikimedia.org)



Open Access (en français : Le libre accès) est la mise à disposition en ligne de contenus numériques, qui peuvent eux-mêmes être soit libres (Creative commons, etc.), soit sous un des régimes de propriété intellectuelle. L'open access est principalement utilisé pour les articles de revues de recherche universitaires, sélectionnés par des pairs. ⁽⁷¹⁾

FIGURE 25 : LOGO OPEN ACCESS - [HTTP://WWW.LIBRARY.VCU.EDU](http://www.library.vcu.edu)

Open Document OpenDocument est un format ouvert de données pour les applications bureautiques : traitements de



FIGURE 26 : UN DES LOGOS DE L'OPEN DOCUMENT (ODT) - [HTTP://OPENCLIPART.ORG](http://openclipart.org)

⁶⁹ Le Creative Commons (CC) est une organisation à but non lucratif dont le but est de proposer une solution alternative légale aux personnes souhaitant libérer leurs œuvres des droits de propriété intellectuelle standards de leur pays, jugés trop restrictifs. L'organisation a créé plusieurs licences, connues sous le nom de licences Creative Commons. Ces licences, selon leur choix, restreignent seulement quelques droits (ou aucun) des travaux. (Wikipedia, Creative Commons)

⁷⁰ Wikipedia, Ressources éducatives libres

⁷¹ Wikipedia, Libre accès

texte, tableurs, présentations, diagrammes, dessins et base de données bureautique. La spécification OpenDocument s'est très largement inspirée du format créé pour les premières versions de la suite bureautique libre OpenOffice.org. Les deux formats sont en effet fondamentalement similaires, tant par leur structure générale que par la liste des objets documentaires qu'ils permettent de représenter. ⁽⁷²⁾



FIGURE 27 : LOGO OPEN SOCIETY FOUNDATIONS - [HTTP://WWW.OPENSOCIETYFOUNDATIONS.ORG/](http://www.opensocietyfoundations.org/)

L'Open society Foundations (OSF) est un réseau de fondations créé en 1993 par le milliardaire américain George Soros, et dont l'objectif est de promouvoir la gouvernance démocratique, les droits de l'homme et des réformes économiques, sociales et légales. A l'échelon local, l'OSF met en œuvre une gamme d'initiatives visant à appuyer la primauté du droit, l'éducation, la santé publique et l'indépendance des médias. ⁽⁷³⁾

L'Open Hardware Open source hardware regroupe les conceptions "Hardware" réalisées publiquement et disponibles de manière à ce que n'importe qui puisse étudier, modifier, distribuer, créer et vendre un "design" ou un produit basé sur ce design. La source du produit hardware, le design duquel le produit est issu, est disponible sous un format choisi pour permettre de faire des modifications. Idéalement, open source hardware utilisera des composants et matériaux facilement



FIGURE 28 : LOGO OPEN SOURCE HARDWARE - [HTTP://OSHWLOGO.COM](http://oshwlogo.com)

approvisionnables, des procédés de fabrication standard, des infrastructures libres, des contenus libres de droit et des outils de design "Open-source" pour maximiser la possibilité donnée à d'autres de concevoir ou utiliser un produit hardware. Open source hardware permet à quiconque d'avoir le contrôle sur leur technologie du moment qu'elles partagent leur savoir et encourage le commerce au travers de l'échange de design libre. ⁽⁷⁴⁾

⁷² Wikipedia, OpenDocument

⁷³ Wikipedia, Open Society Foundations

⁷⁴ OSHW - <http://www.oshw.org/>

L'open source appliqué aux objets

« L'internet a démocratisé la publication, la diffusion et la communication, ce qui a eu pour conséquence d'accroître de façon massive le degré de participation et de participants dans le monde digital : la longue traîne des bits. La même chose est en train de se produire en termes de fabrication : la longue traîne des objets. Le web n'était qu'une démonstration de faisabilité (proof of concept). Maintenant, la révolution va aussi toucher le monde réel. » (Chris Anderson)

Introduction

L'open hardware (appelé également Open Source Hardware ou matériel libre) est un terme qui regroupe des artefacts tangibles machines, dispositifs ou toutes choses physiques dont les plans ont été rendus publics, de façon que quiconque puisse les fabriquer, modifier, distribuer et les utiliser. ⁽⁷⁵⁾.

S'agissant d'un concept relativement novateur (à ma connaissance les premiers projets datent de 2006). Il existe encore peu de documentation sur le sujet c'est pour cela que cette partie de mon travail se focalisera plus sur une interprétation personnelle.

Aujourd'hui le matériel que nous achetons, que ce soit un ordinateur, un appareil électroménager ou encore une voiture souffre d'un certain verrouillage technologique dans le but de programmer son obsolescence et pousser le consommateur à racheter du neuf.

Ce mouvement, comparable à celui du logiciel libre, chamboule le monde de la conception du matériel. Là où les sociétés avaient l'habitude de garder secret les plans électroniques, les programmes, les nomenclatures de pièces et tous les brevets, les défenseurs du matériel ouvert prônent le partage, la transparence et une simplification de la conception des produits. Certains projets ont également le but de rendre accessible certaines technologies, à priori complexe, en fournissant des outils

⁷⁵ Wikipedia, Matériel libre

simplifiant leur utilisation. À l'instar de la société ArduinoTM qui développe des produits rendant accessibles la robotique et l'automatisation aux non-initiés.

Les avantages sont multiples : contrôle de la technologie par les utilisateurs et les concepteurs, réparation facilitée, meilleure vision sur l'approvisionnement des pièces, amélioration continue du matériel par la communauté... ⁽⁷⁶⁾

En détail

Le produit logiciel est particulier dans le sens où il est dématérialisé et est donc considéré comme une œuvre de l'esprit (voir plus haut). Sa conception est étroitement liée à sa production car il suffit de compiler le code et le mettre à disposition sur un support physique ou internet pour obtenir un produit utilisable. Le matériel est quant à lui soumis à des contraintes de construction ce qui peut parfois s'avérer onéreux et nécessiter des compétences pointues ou des machines industrielles.

Pour cette raison, l'exploitation commerciale de l'Open Hardware diffère de celle possible avec les solutions Open Source logicielles. S'il est possible de proposer le même type de services que ceux associés à une application Open Source. Il est également possible de commercialiser la fabrication du matériel ainsi que, contrairement au logiciel qui n'est pas sujet à l'usure, des services d'entretien et de réparation.

Avec le partage des plans, la simplification de la conception d'appareils électronique, la démocratisation de l'impression d'objets 3d et la réalisation de circuits imprimés il est possible de passer de la conception d'un prototype d'appareil à une production industrielle sans trop de contraintes. Le développement d'appareils électronique devient alors accessible à toute personne motivée sans nécessiter de compétences particulières.

Règles et licences

Étant donné l'aspect novateur de ce concept où les « règles » de l'open source logiciel sont utilisées dans un contexte qui n'est pas nécessairement adapté. Il convient d'en définir les règles. C'est pour cela que certaines personnes ont adaptés les 4 règles du logiciel libre au matériel libre :

⁷⁶ Greenit

- Liberté 1: La liberté d'utiliser le produit, pour tous les usages.
- Liberté 2: La liberté d'étudier le fonctionnement du produit (ainsi que des parties logicielles), de réaliser des copies du produit, et de l'adapter à ses besoins (Les schémas de fonctionnement du produit, la liste des pièces, ainsi que le plan de montage doivent donc être disponibles)
- Liberté 3: La liberté de réaliser et de redistribuer des copies du dit produit (ainsi que des parties logicielles).
- Liberté 4: La liberté d'améliorer le produit (ainsi que ses parties logicielles) et de publier ses améliorations, pour en faire profiter toute la communauté

Au même titre que l'Open Source logiciel, l'Open Hardware a lui aussi ses licences. Fortement inspiré de celles déjà présentes dans l'Open Source les différences entre la conception logicielle et la conception d'un produit physique nécessitent quelques adaptations. Dans les licences principales actuelles on peut citer :

- TAPR : rédigé par l'avocat John Ackermann, examiné par les dirigeants de la communauté des logiciels libres Bruce Perens et Eric S. Raymond (auteur de La cathédrale et le bazar), et discuté par des centaines de volontaires dans une communauté ouverte de discussion. ⁽⁷⁷⁾
- CERN Open Hardware Licence (OHL) : soutenue par le CERN et principalement utilisée sur l'Open Hardware Repository ⁷⁸ ⁽⁷⁹⁾



FIGURE 29 : LOGO DE LA TAPR
OPEN HARDWARE LICENCE -
[HTTP://WWW.TAPR.ORG/OHL.HTML](http://www.tapr.org/ohl.html)



FIGURE 30 : LOGO DE LA LICENCE
OPEN HARDWARE DU CERN -
[HTTP://WWW.OHWR.ORG/](http://www.ohwr.org/)

⁷⁷ <http://www.tapr.org/ohl.html>

⁷⁸ L'Open Hardware Repository (OHWR) est un projet lancé à l'initiative de concepteurs en électronique travaillant dans des laboratoires de physique expérimentale qui ont estimé nécessaire de faciliter les échanges à très grande échelle, pour aller dans le sens d'une science « ouverte », encouragée par des organisations telles que le CERN. (Cern)

⁷⁹ <http://www.ohwr.org/documents/88>

Ces licences se différencient également sur la portée du Copyleft appliqué en fonction du contexte.

Ici figure les embryons des principales licences, comme dans l'Open Source Logiciel il est fréquent qu'un projet se protège en créant sa propre licence spécifique et que d'autres s'en inspirent par la suite.

La communauté

Les principaux acteurs de l'Open Hardware se réunissent tous les ans afin de faire le point sur les nouveaux projets et les problèmes de lois qui s'y réfèrent lors des Open Hardware Summit organisé par l'Open Source Hardware Association (OSHWa)

L'Open Hardware est donc, au-delà du concept et au même titre que les communautés Open Source, une communauté d'entraide persuadé de l'importance de s'orienter vers une économie qui favorise le partage et la transparence, la durabilité des produits et l'intérêt du consommateur et pour qui l'innovation peut venir de n'importe qui.

Au même titre que les communautés d'utilisateurs-développeurs ces communautés proche de la communauté des hackers⁸⁰ s'organisent à travers internet, différentes manifestations sous forme de workshop ou de « club de bricole » communément appelés FabLab.

Les FabLab sont des endroits où du matériel permettant le prototypage d'objets physiques est mis à disposition librement. Il s'agit (traduit littéralement) de « laboratoire de fabrication ». Concrètement une FabLab regroupe un ensemble de machines dédiées au prototypage de produits où leurs membres, tous partisans de l'open source,



FIGURE 31 : LOGO COMMUNÉMENT
CHOISI PAR BON NOMBRE DE FABLAB -
SOURCE DE L'IMAGE :
[HTTP://MYGLOBALBORDEAUX.COM](http://myglobalbordeaux.com)

⁸⁰ Le hacker est comme une personne qui se délecte de la compréhension approfondie du fonctionnement interne d'un système. Cette dernière définition fait écho à celle utilisée depuis les années 1950 par les radioamateurs pour qui le hacking est un bricolage créatif visant à améliorer le fonctionnement d'un système.

partagent leurs connaissances afin de pouvoir réaliser les projets de chacun.

C'est la « réalisation concrète » des sites internet communautaire habituellement utilisés dans le monde de l'Open Source (la réalisation « physique » d'un produit nécessitant des ressources « physique »).

Quelques projets populaires

De multiples projets de matériel libre sont en cours, leur nombre étant élevé je présente ici que quelques produits phares :

Les imprimantes 3-D

Elles sont importantes dans le monde du matériel libre car elles permettent la construction à l'unité ou en petite série d'objets spécifiques soit issu de plans ouvert soit issus de créations personnelles pour des objets de décoration ou des boîtiers nécessaires à la finalisation de prototypes. Pour l'instant ces modèles d'imprimante ne reproduisent principalement que des pièces en plastique mais certains modèles récents permettent l'impression de pièce dans de multiples matériaux tels que le ciment, le silicone, l'acier inoxydable, le bois ou encore la nourriture (sucre ou pâte). Ce n'est donc pas étonnant que divers projets s'orientent vers ce type de produit. Dans les projets phares on peut recenser :

RepRap : est une imprimante 3D développée par une communauté capable d'imprimer des pièces de plastique à partir d'un fichier informatique. Elle est capable d'imprimer le principal de ses composants et est donc quasi autoreproductible (à au moins 70%). Le restant des composants étant volontairement génériques, bon marché et constitué de pièces que RepRap n'est pas capable d'imprimer (pièces complexes ou constitué de métal, de bois ou de verre). Les plans de cette imprimante sont disponibles gratuitement il est donc possible d'acheter les composants séparément et les monter soi-même. Cependant il existe plusieurs sites de vente en ligne proposant des kits comprenant toutes les pièces ou uniquement celles non-imprimables par une autre RepRap ou imprimante 3D.



FIGURE 32 : LOGO DE
REPRAP -
[HTTP://REPRAP.ORG](http://reprap.org)

Makerbot™ : Est une société qui commercialise et développe des imprimantes 3d basées sur le projet RepRap. Son modèle d'affaire a créé la polémique au sein du mouvement du matériel libre car une partie des développements de ses produits provient de la communauté (RepRap). Cependant ses produits sont plus élaborés que la version gratuite et de qualité professionnelle. Elle représente donc un parfait exemple de création de valeur à partir de produit ouvert.



FIGURE 33 : LOGO DE MAKERBOT
- [HTTP://WWW.MAKERBOT.COM](http://www.makerbot.com)

L'automobile

L'industrie de l'automobile s'oriente vers des produits rentables et le cloisonnement de leurs technologies, il n'est aujourd'hui pas étonnant de voir certains modèles difficilement démontable et dont l'entretien est réservé aux succursales du constructeur. De plus les lobbies pétroliers ont considérablement ralenti la recherche vers de nouveaux types de motorisations. C'est pour cela que certains ont jugé intéressant de développer des plans de voitures ouverts destinés à tous mécaniciens ayant les capacités de produire ces machines.

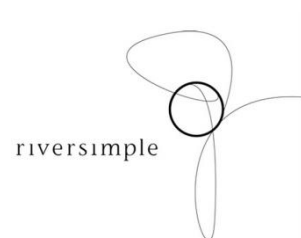


FIGURE 34 : LOGO DE RIVERSIMPLE
- [HTTP://WWW.RIVERSIMPLE.COM](http://www.riversimple.com)

Riversimple Urban Car est un projet de véhicule urbain fonctionnant à l'hydrogène, sa mise en production est planifiée pour 2013.

Wikispeed est un constructeur automobile qui fabrique des voitures modulables et open source. La voiture a fait ses débuts lors du Salon international de l'automobile d'Amérique du Nord en janvier 2011 à Detroit. ⁽⁸¹⁾



FIGURE 35 : LOGO DE WIKISPEED -
[HTTP://WWW.WIKISPEED.COM](http://www.wikispeed.com)

⁸¹ Wikipedia, Wikispeed

Matériel électronique grand public

Il existe une multitude de projets de matériel électronique libre, du baladeur mp3 à la téléphonie mobile en passant par la console de jeux, la machine à laver ou encore la caméra HD. Ces projets étant trop nombreux je ne m'attarderais pas sur le listing de ceux-ci.

Matériel informatique

L'architecture ouverte pour les composants informatiques est un peu compliquée car elle nécessite l'utilisation de puces électroniques dont la production est réservée qu'à quelques grosses sociétés (fondeurs⁸²) ayant les ressources et la technologie nécessaire à leur fabrication. Cependant, malgré l'utilisation de composants complexes non libres, de plus en plus de projets voient le jour afin de proposer des architectures génériques et modifiables à faible prix. Ces projets se focalisent sur des architectures simples contenant, de préférence, des composants facilement accessible et bon marché.

Arduino™ : est une société concevant des circuits imprimés « ouverts » (dont les plans des cartes sont publiés en licence libre) sur lesquels se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques. Les cartes Arduino sont très présentes dans les projets open source car elles sont bon marché et permettent l'automatisation de



FIGURE 36 : LOGO DE LA SOCIÉTÉ ARDUINO
- [HTTP://WWW.ARDUINO.CC](http://www.arduino.cc)

systèmes électroniques ou électriques par une programmation simple de la puce. Il est alors plus nécessaire d'avoir des connaissances poussées en électronique pour développer un produit électronique. Elles sont donc fréquemment utilisées comme contrôleur dans des projets open hardware par exemple c'est une carte Arduino qui contrôle les imprimantes RepRap. L'Arduino constitue donc une « brique technologique » (au même titre que MySQL pour les logiciels) utilisable pour la conception ou le prototypage de produits électroniques complexes. (⁸³)

⁸² Une société fabricant des puces électroniques

⁸³ <http://www.arduino.cc/fr/>

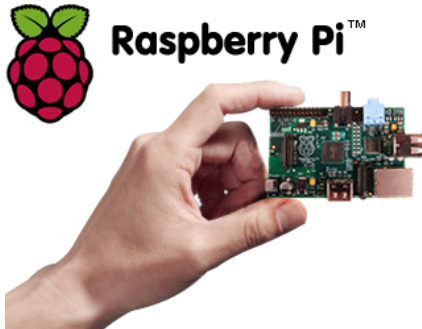


FIGURE 37 : LOGO DE RASPBERRY PI AVEC UNE IMAGE DU PRODUIT - [HTTP://WWW.RASPBERRYPI.ORG](http://www.raspberrypi.org)

Raspberry Pi™ : est une société concevant des ordinateurs de la taille d'une carte de crédit qui sont vendus aux alentours des 25 USD. Doté des différents ports usuels (ethernet, audio, vidéo, usb) il permet l'exécution de plusieurs versions du système d'exploitation Linux ou Android. Il est également possible d'y connecter un Arduino ou

différents modules électroniques annexe. Il est souvent utilisé pour des systèmes intégrés nécessitant un ordinateur (par exemple pour du traitement d'image) mais également comme mini-serveur. ⁽⁸⁴⁾

Matériel médical libre

Il existe une multitude de projets médicaux où l'objectif est de partager de manière plus fluide des développements universitaires, de mettre en place des standards ouverts et de créer des versions génériques bon marché d'appareils dont le prix est souvent exorbitant.

Open Prosthetics Project : Est un projet de design de prothèses libres dans le but de créer une communauté d'utilisateurs, de designer et de donateurs mettant au point différents types de prothèses dont le design est disponible gratuitement afin de stimuler et accélérer l'impact de leurs innovations dans l'industrie. Ils sont également à l'origine du projet Open Myoelectric Signal



FIGURE 38 : LOGO CRÉÉ PAR OPEN PROSTHETICS PROJECT - [HTTP://OPENPROSTHETICS.ORG](http://openprosthetics.org)

Processor destiné à créer une interface matériel libre pour diriger des prothèses à partir de signaux du corps ⁽⁸⁵⁾.

⁸⁴ <http://www.raspberrypi.org/>

⁸⁵ <http://openprosthetics.org/>



FIGURE 39 : LOGO DE WHIRLWIND WHEELCHAIR - [HTTP://WWW.WHIRLWINDWHEELCHAIR.ORG](http://www.whirlwindwheelchair.org)

Whirlwind Wheelchair International est une

entreprise sociale à but non lucratif dont l'objectif est d'améliorer la vie des personnes handicapées dans les pays en voie de développement tout en favorisant l'économie local. Travaillant à fournir à toute personne le nécessitant un fauteuil roulant de haute qualité adapté à ses besoins via un réseau de fabricants régionaux. ⁽⁸⁶⁾

Open EEG est un projet de développement d'appareils EEG (Électro-encéphalographie) et de logiciels liés en Open Source. Les EEG actuels étant onéreux l'objectif est de développer des plans libres d'accès afin de rendre ces appareils accessibles aux amateurs et favoriser le développement de cette technologie très prometteuse.



FIGURE 40 : LOGO DU PROJET OPENEEG, SYMBOLISE LE GNU (ANIMAL) AVEC UN EEG CONNECTÉ À UN ORDINATEUR, LE GNU EST L'EMBLÈME DU PROJET GNU À L'ORIGINE DE LA LICENCE GPL - [HTTP://OPENEEG.SOURCEFORGE.NET](http://openeeg.sourceforge.net)

Agriculture

Il existe de multiples projets agricoles destinés au partage de plans de matériel ou de technique de culture. Voici probablement le plus populaire d'entre eux :

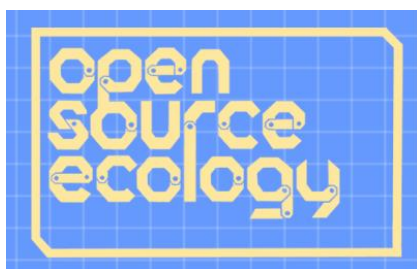


FIGURE 41 : LOGO DU PROJET OPEN SOURCE ECOLOGY - [HTTP://OPENSOURCEECOLOGY.ORG](http://opensourceecology.org)

Open Source Ecology est un réseau de fermiers, d'ingénieurs et de supporters dont l'objectif est la production du Global Village Construction Set (GVCS). Tel que décrit par Open Source Ecology « le GVCS est une plateforme technologique ouverte qui permet la production aisée des 50 machines industrielles nécessaires pour construire une petite civilisation avec tout le confort moderne ». Des groupes à Oberlin, Ohio,

⁸⁶ <http://www.whirlwindwheelchair.org/>

Pennsylvania, New York et en Californie développent des plans et construisent des prototypes qui sont ensuite envoyés dans le Missouri. Les machines sont construites et testées sur la Factor e-Farm dans le Missouri ⁽⁸⁷⁾. ⁽⁸⁸⁾

Il existe également une multitude de projets écologiques (traitement de l'eau, production d'énergie verte), de communication libre (satellites, réseau GSM ouvert, ballons internet), de médicaments libres, de nourriture libre et même de plans d'armes pour imprimantes 3d libre. Les domaines d'application de l'Open Hardware n'ont donc pas de limites. Il s'agit simplement de rendre « générique » un produit physique existant en le reconstruisant ou de créer un nouveau produit d'intérêt général de manière collective et accessible à tous.

Avantages et opportunités

L'Open Hardware n'a rien de nouveau en soit il s'agit simplement de développer un produit en commun dont on partagera les plans. Cependant les différentes licences, la communication simplifiée grâce à internet et le phénomène de mode permettent un certain engouement pour ces « nouveaux » projets collectif. À l'heure où la crise financière et l'austérité occupe une bonne partie du paysage médiatique. L'entraide et le « fais-le-toi-même » (DIY⁸⁹) semblent de bonne augure.

Les différents modèles d'affaire de l'open source sont applicables pour l'open hardware. Que ce soit dans le service lié à l'entretien de produits libres, la distribution à valeur ajoutée, le modèle de licence double ou la mutualisation le fait de travailler avec une base « commune » reste la même pour le monde du matériel. À la différence près qu'au-delà de la conception, à l'inverse du logiciel libre (dont il suffit de compiler le code pour obtenir un produit utilisable), les plans de matériel libre permettent la valorisation de la fabrication de la réparation et de l'entretien du produit fini.

⁸⁷ Wikipedia, Open Source Ecology

⁸⁸ <http://opensourceecology.org/>

⁸⁹ Do it yourself (DIY) est une appellation, dont une traduction littérale en français serait « Faites-le vous-même ».

Plusieurs entreprises se sont d'ailleurs focalisées sur ce marché. Par exemple Sparkfun^{TM 90} ou SeedStudio^{TM 91} qui vendent différents produits open hardware en kit ou partiellement montés.

D'autres projets se destinent à une simplification ou une simplification de l'accessibilité à certaines technologies. Comme la société ArduinoTM qui développe une « brique technologique » très intéressante pour toutes personnes désirant développer un produit électronique en petite série ou créer un prototype. Le Raspberry PI qui permet l'intégration d'un mini-ordinateur bon marché dans un projet. Et paradoxalement ces sociétés ne cessent de croître en vendant leurs produits malgré le fait que les plans de leurs produits soient publics. D'autres projets, comme Open EEG se sont développés avec le but de rendre « générique » certaines technologies afin d'améliorer leur accessibilité à toute personne intéressée.

Les différents projets Open Hardware, au même titre que les projets Open Source, peuvent donc constituer de précieuses « briques » technologiques pour développer la créativité de nouvelles entreprises ou conquérir de nouveaux marchés à l'instar de MakerBotTM qui vend des imprimantes 3d de qualité professionnelle développées à partir de projet open source.

Ces différentes technologies « libérées » représentent donc une opportunité très intéressante pour toutes personnes désirant développer certains produits technologiquement pointus sans avoir nécessairement de gros moyens ou de grandes connaissances. Cela représente également une opportunité énorme pour toutes sociétés ou organisations humanitaires présentes dans des pays en voie de développement n'ayant pas nécessairement les fonds ou les connaissances pour développer des produits hautement sophistiqués ou payer différents brevets.

Pour l'utilisateur lambda cela représente l'opportunité d'être maître des technologies qu'il utilise et la standardisation de celles-ci améliore indirectement l'offre de gros industriels grâce à la concurrence (ce qui favorise le développement).

⁹⁰ SparkFun Electronics (parfois connu sous son abréviation, SFE) est un détaillant de composants électroniques basé à Boulder, dans le Colorado, aux États-Unis. Il confectionne et vend des cartes de développement de microcontrôleur et des circuits imprimés. (Wikipedia, SparkFun Electronics)

⁹¹ SeedStudio, au même titre que SparkFun vend est un détaillant de composant électronique mais il entretient également une communauté d'utilisateur-développeur orientés autour de projets Open Hardware.

L'accessibilité que permettent des appareils comme l'Arduino pousse de grosses entreprises à proposer des produits similaires bon marché. À l'instar de Microsoft avec son projet Gadgeteer⁹² (⁹³) ou encore Texas Instrument avec son LaunchPad⁹⁴.

Plusieurs grandes entreprises commencent également à ouvrir leurs produits au monde de l'open source comme SonyTM avec une montre SmartWatch⁹⁵ qui permet l'utilisation de firmware⁹⁶ personnalisés⁹⁷ et MicrosoftTM qui a également ouvert son produit KinectTM⁹⁸ en fournissant un kit de développement ce qui a créé le projet open kinect⁹⁹. Google qui a ouvert son système d'exploitation Android (qui est Open Source) et son kit de développement ADK (Android Development Kit).

Avenir et complications

Si le domaine de l'Open Hardware semble glorieux il présente tout de même certains problèmes liés au design des produits ou au contrôle de la production de certains produits considérés comme dangereux.

En effet, il est aujourd'hui facile de numériser le boîtier (ou l'esthétique) d'un produit. Ce problème, de même nature que la copie numérique d'œuvre artistique (piratage de musique, livres, films etc..), commence à devenir conséquent pour tous les designers de produits à cause des nouvelles technologies d'impression 3d permettant le « piratage » d'objets numérisés.

⁹² Microsoft .NET Gadgeteer est une plateforme de prototypage rapide Open Source programmable avec le langage .NET (propriété de Microsoft)

⁹³ <http://www.extremetech.com/computing/91990-gadgeteer-microsofts-open-source-net-alternative-to-arduino>

⁹⁴ http://www.ti.com/ww/en/launchpad/home_head.html?DCMP=Value_Line&HQS=launchpad

⁹⁵ Une smartwatch, littéralement « montre intelligente », est une montre bracelet informatisée avec des fonctionnalités allant au-delà du simple chronométrage, présentant des caractéristiques comparables à celles d'un PDA. Alors que les premiers modèles, apparus dans les années 1980, étaient capables d'effectuer des tâches de base comme les calculs, l'agenda-planning puis les traductions ou les jeux, les smartwatches modernes sont devenues des ordinateurs à part entière. Certaines smartwatches peuvent exécuter des applications mobiles. (Wikipedia, SmartWatch)

⁹⁶ Un firmware est un ensemble d'instructions et de structures de données qui sont intégrées dans un appareil électronique pour qu'il puisse fonctionner

⁹⁷ <http://www.futura-sciences.com/magazines/high-tech/infos/actu/d/technologie-open-hardware-sony-ouvre-montre-smartwatch-hacking-47160/>

⁹⁸ Kinect est un périphérique destiné à la console de jeux vidéo Xbox 360 permettant de contrôler des jeux vidéo sans utiliser de manette. (Wikipedia, Kinect)

⁹⁹ http://openkinect.org/wiki/Main_Page

Un autre problème est le contrôle des produits sensibles, en effet si il est facile de contrôler la production et la distribution d'un produit par le peu de constructeurs qui existent, il en sera d'autant plus difficile à partir du moment où n'importe qui aura la possibilité de le produire à domicile. En effet les imprimantes 3d sont capables aujourd'hui d'imprimer des objets en plastique mais bientôt en bois, métal ou d'autres matériaux. Ce qui permettra à tout un chacun d'imprimer une arme blanche ou à feu à domicile, ce qui est déjà le cas avec des projets comme Defcad¹⁰⁰ qui propose les plans de différentes pièces d'armes à feu en plastique imprimables par des imprimantes 3d d'aujourd'hui ce qui pose problème non seulement au niveau des permis nécessaires à l'obtention de telles armes mais également dans les systèmes de contrôles (scanners dans les douanes, aéroport etc.) conçus pour détecter des objets métalliques.



FIGURE 42 : LOGO DE DEFENCE DISTRIBUTED, À L'ORIGINE DE DEFCAD - [HTTP://DEFDIST.ORG](http://defdist.org)

Différents projets sont également à l'étude pour la conception de médicaments à partir d'imprimantes 3d. Ce qui risque de compliquer considérablement le contrôle de certaines substances illicites ou de médicaments illégaux.

Le domaine de l'open hardware et de la « libéralisation » du matériel face à l'industrie ne se fera donc pas sans conséquences. Cependant il représente une réelle opportunité pour de nouvelles entreprises et de nouveaux marchés. Et comme toute révolution il nécessitera une adaptation autant au niveau juridique qu'organisationnel.

¹⁰⁰ Defense Distributed est une association en ligne de fabrication d'armes à feu open-source, appelé Wiki Weapon. L'un des objectifs de l'organisation est de développer et de publier librement les schémas de conception d'armes à feu qui peuvent être téléchargés et reproduits par n'importe qui avec une imprimante 3D. (Wikipedia, Defense Distributed)

Réalisation concrète d'un projet basé sur du matériel Libre

Descriptif du projet

En résumé

Ce projet a pour but l'automatisation du système multimédia d'une salle de classe. Il a été développé pour la société Lumens 8 SA afin de pouvoir contrôler du matériel installé dans des classes d'école.

L'objectif est de pouvoir allumer ou éteindre un projecteur, sélectionner les sources vidéo et son à partir d'un commutateur d'entrées audio/vidéo, allumer ou éteindre un amplificateur son et faire monter ou descendre un écran à partir d'un périphérique Android.

Technologies choisies

L'amplificateur son n'étant pas contrôlable via un système quelconque il s'agit uniquement d'allumer ou d'éteindre son alimentation (220v). L'écran est contrôlé via un moteur alimenté par 2 alimentations standards (220v) soit une pour le monter et l'autre pour le descendre. Le projecteur et le commutateur audio/vidéo sont contrôlables via un connecteur RJ45¹⁰¹ en utilisant le protocole Telnet¹⁰². Tout cela piloté via un périphérique Android¹⁰³ connecté en Bluetooth¹⁰⁴.

Il fallait donc un appareil « intelligent » doté d'un port Ethernet (RJ45), de trois relais¹⁰⁵ 220v et d'une connexion Bluetooth.

J'ai donc choisi comme base une carte Arduino Ethernet, ce qui me permettait d'avoir à la fois un port Ethernet et l'accès à un périphérique de stockage (carte microSD¹⁰⁶)

¹⁰¹ Connecteur utilisé couramment pour les connexions Ethernet ou réseau filaires entre ordinateurs et périphériques compatible (imprimantes, webcam, etc.)

¹⁰² Telnet est un protocole réseau, son but est de fournir un moyen de communication très généraliste souvent utilisé pour le contrôle d'appareils via le réseau.

¹⁰³ Android est un système d'exploitation spécialisé dans les périphériques mobiles. Utilisant le noyau Linux et conçus par Android, une startup rachetée par Google.

¹⁰⁴ Bluetooth est un standard de communication utilisant les ondes radios afin de connecter entre eux différents appareils électroniques.

¹⁰⁵ Un relai électromécanique est un composant permettant la commutation de liaisons électriques. Il s'agit d'un conducteur magnétique chargé d'établir le contact ou non d'une liaison électrique (de plus grande puissance). Il est souvent piloté par du courant faible et permet le contact de liaisons de courant fort.

¹⁰⁶ Le format MicroSD est une unité de stockage qui utilise de la mémoire flash. Ce format est souvent utilisé dans les périphériques mobiles.

afin de pouvoir y stocker la configuration de l'appareil. Cette carte étant dépourvue de connexion Bluetooth j'y ai ajouté une carte compatible Arduino (BlueSMIRF Gold¹⁰⁷) et ajouté trois relais (220v)¹⁰⁸ nécessaires au contrôle de l'amplificateur audio et de l'écran.

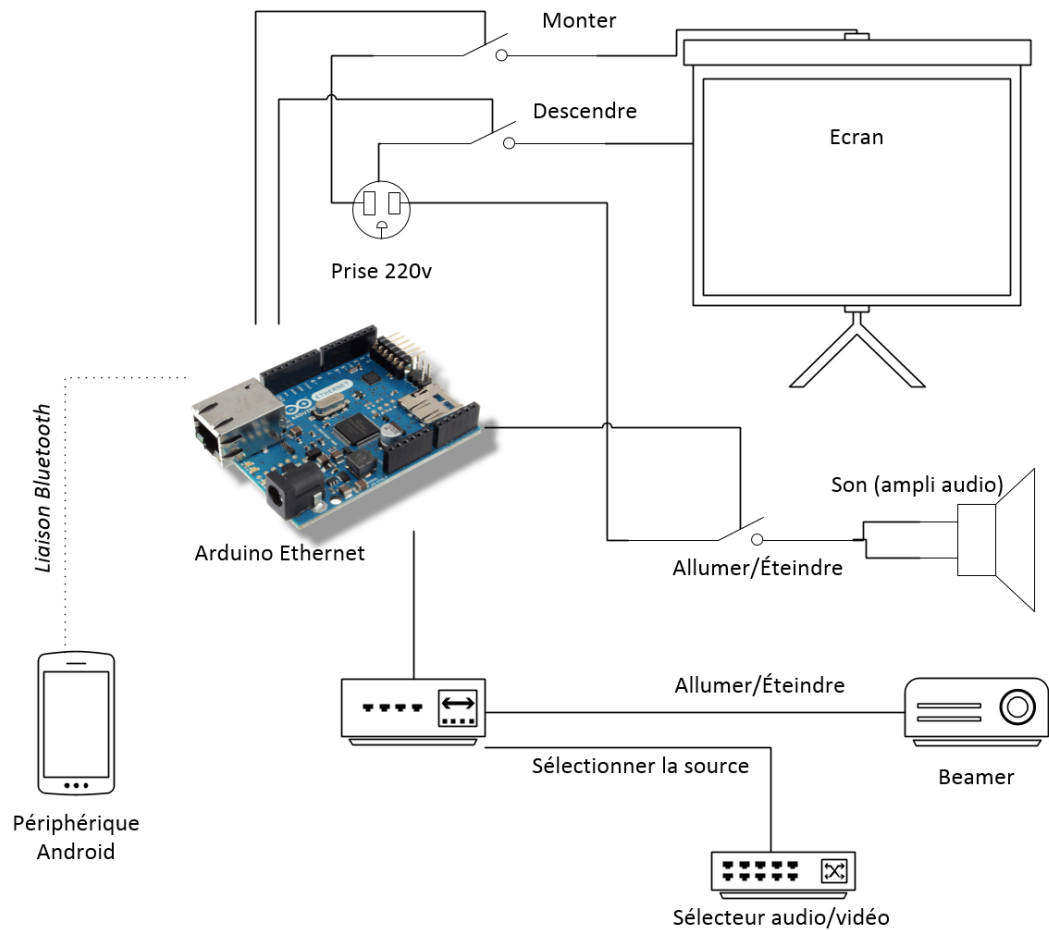


FIGURE 43 : SCHÉMA DU PROJET DE TÉLÉCOMMANDE MULTIMÉDIA POUR LUMENS 8

¹⁰⁷ <https://www.sparkfun.com/products/10268>

¹⁰⁸ http://www.evola.fr/product_info.php/platine-relais-p-102

Journal du projet

1^{er} jour : Analyse des contraintes, sélection et commande du matériel adéquat, soit :

- 1 x Arduino Ethernet = 65chf
- 3 x Platine relais = 9.80chf
- 1 x Module Bluetooth BlueSMIRF Gold = 50.80chf
- 1 x Boitier pour Arduino = 12.80chf

Ce jour-là j'ai travaillé environ 6h.

2^{ème} jour : N'étant pas du tout familiarisé avec ce type de technologies je décide de tester les composants afin de comprendre leur fonctionnement. Je les branche et m'inspire d'expériences sur internet. En fin de journée j'arrive à piloter un appareil compatible Telnet via mon téléphone portable connecté en Bluetooth.

Ce jour-là j'ai travaillé environ 6h.

3^{ème} jour : Je teste l'envoi de requêtes Telnet complètes stockées dans l'Arduino et ça marche ! Cependant je fais face à un problème : mon contrôleur doit se connecter sur deux « serveurs » Telnet différents, je dois donc soit établir deux connexions soit fermer et rouvrir la connexion à chaque requêtes Telnet.

Ce jour-là j'ai travaillé environ 6h.

4^{ème} jour : Je constate, après de multiples tests, que ma carte Arduino ne gère pas le fait d'être client de plusieurs serveurs (ici Telnet) je décide donc de réinitialiser la connexion à chaque requêtes. De plus mes requêtes commencent à saturer la mémoire de l'Arduino qui est relativement limitée.

Ce jour-là j'ai travaillé environ 6h.

5^{ème} jour : Je décide d'utiliser la carte SD pour stocker les paramètres de connexions (adresse ip et ports des serveurs Telnet ainsi que les requêtes). Je test et adopte également le fait de lire directement sur la carte les requêtes Telnet ce qui me permet de régler mes soucis de mémoire (j'évite ainsi de charger les requêtes dans la mémoire vive). Je finis la programmation de l'Arduino en lui ajoutant une fonction de test qui renvoie l'état de l'appareil.

Ce jour-là j'ai travaillé environ 6h.

6^{ème} jour : Je réfléchis à un boitier et vais acheter les fournitures nécessaires à l'élaboration finale de mon produit :

- 3 x fiches T 12 = 2.85CHF
- 3 x prises T 13 = 5.25CHF
- 2 x boîtiers de dérivation = 21.80CHF
- 1 x reglette D.bornes = 1.25CHF
- 1 x Câble 220v 10m = 12.90CHF

De plus je commande les connecteurs et le câble nécessaire à une finalisation correcte du produit :

- 8 x Connecteur jack 3,5mm mâle mono = 4.95CHF
- 1 x Câble haut-parleur 2 x 0,50mm 10m = 4.45CHF
- 8 x Connecteur jack châssis 3,5mm femelle mono = 3.95CHF

Je commence également l'application Android en testant différents exemples de code permettant la connexion Bluetooth.

Ce jour-là j'ai travaillé environ 8h.

7^{ème} jour : Je code l'application Android et finalise le développement applicatif (Android et Arduino) du produit.

Ce jour-là j'ai travaillé environ 8h.

8^{ème} jour : Je reçois les connecteurs, conçois et câble les boîtiers.

9^{ème} jour : Je me rends chez le client et test mon système sur son matériel.

Présentation du résultat

Ce projet, dans l'état actuel, m'a coûté environ 200 CHF de matériel et environ 60 heures de main d'œuvre. Si l'on considère, dans ce domaine, que le tarif facturé à une entreprise est d'environ 80 CHF le projet a coûté :

- 200 CHF de matériel
- 4800 CHF de main d'œuvre

Soit un total de 5000 CHF.

Ces chiffres ne tiennent pas compte du fait que je suis un débutant dans ce domaine et que par conséquent j'ai passé beaucoup d'heures à comprendre l'utilisation des différentes technologies utilisées.

Les problèmes rencontrés :

- **La connaissance des différentes technologies à utiliser** : Étant un novice dans ce domaine il m'a fallu un certain temps afin de maîtriser les différentes technologies utilisées.
- **Les ressources limitées de l'Arduino** : il faut prévoir de bien adapter la carte choisie en fonction des spécifications. La programmation facilitée de ces cartes permet un résultat rapide mais il est primordial de tenir compte des limites de l'appareil (spécialement au niveau de la mémoire) et d'adapter son code en conséquence. À partir d'une certaine complexité, il sera préférable d'opter pour un appareil plus puissant ou disposant de plus de ressources par exemple un Arduino Mega ou, si du traitement de signal est nécessaire, un raspberry pi. Il est aussi important d'adapter le produit choisi aux connaissances connues par exemple Microsoft™ (Gadgeteer) ou Texas Instrument (LaunchPad) ont développé leur propre plateforme de prototypage basé sur d'autres langages de programmation ce qui permet un gain de temps d'apprentissage conséquent pour les personnes maîtrisant ce type de langage.
- **Le packaging et le boîtier** : S'il est « facile » de développer un appareil fonctionnel il n'est pas nécessairement évident qu'il soit intuitif ou esthétique. Dans le cadre de ce projet l'appareil fait partie d'un équipement complet qui sera caché dans une armoire ce qui permet l'utilisation de boîtiers génériques bon marchés mais peu esthétiques. Cependant si le produit devait être vendu et installé par le client il serait intéressant de soigner son apparence et de concevoir un joli boîtier de finition professionnelle. Pour cela l'impression 3d est un outil précieux car elle permet l'impression de prototypes de pièces de plastique via un ordinateur et un logiciel de conception 3d. Il est donc possible de dessiner son boîtier puis l'imprimer pour le tester. Une fois le design du boîtier fini les fichiers peuvent être directement envoyés à une entreprise spécialisée pour une production en série.

Ce projet a été concrétisé avec succès en respectant les contraintes fournies. Cependant devant le produit fini il s'est avéré intéressant de pousser le développement afin d'améliorer l'interopérabilité de l'appareil en le rendant compatible avec un plus grand nombre de projecteurs (n'étant pas tous compatible Telnet). Ainsi que de développer des fonctionnalités supplémentaires comme le contrôle du volume ou des stores de la classe.

Opportunités de production en série du prototype développé

Pour produire un appareil en série il faut faire construire son boîtier et le circuit intégré qui le constitue. Pour cela il existe une multitude de fournisseurs de services accessibles via internet qui, de par ce nouvel engouement pour le prototypage facile, proposent des interfaces simplifiées.

Afin de pouvoir créer son propre boîtier il convient de sélectionner la technologie la plus appropriée :

- **La Stéréolithographie** (procédé similaire à l'impression 3d) où l'objet est obtenu par ajout de couches successives de résine photosensible. Ce procédé est intéressant pour de petites à moyennes productions.
- **L'usinage** : où l'on enlève les couches plastiques d'un bloc afin d'obtenir l'objet désiré. Cela peut être intéressant pour la production en petites et moyennes séries de certaines formes.
- **L'injection** : qui consiste à injecter la matière dans un moule. Ce procédé nécessite la création d'un moule mais permet de réduire les frais dans le cadre d'une production en grande quantité.

Design de PCB (circuits imprimés) :

Devant l'engouement de ces nouvelles cartes destinées au prototypage, de multiples sociétés d'impression de PCB se sont adaptées pour proposer des interfaces simplifiées et compatibles avec les formats de design open source permettant des devis en ligne simplifiés (à l'instar de oshpark¹⁰⁹ qui propose de poser le fichier du design du projet afin d'obtenir un devis immédiat). La pose des composants est également un service supplémentaire fourni par beaucoup de ces entreprises. Cependant le prix de l'impression de ces cartes et la pose des composants devient intéressant (en moyenne) à partir d'une centaine de pièces.

Ce projet utilisant tous les composants et 86% des ressources de l'Arduino Ethernet et étant destiné à une production de moins de 100 exemplaires il ne m'a pas semblé intéressant d'en concevoir le schéma et d'envisager une production en série.

Je n'ai donc pas de devis pour la production en série de cet appareil à présenter.

¹⁰⁹ <http://oshpark.com/>

Bilan

Conclusion du projet

Ce projet s'est avéré relativement facile à concrétiser. Il a prouvé la faisabilité de l'appareil, nous a permis de se faire une idée de son prix et de son potentiel d'évolution. Il est maintenant intéressant d'en développer une version vendable et durable en faisant attention à la fois au design de l'application qu'à l'esthétique du produit.

Comparatif avec le milieu professionnel

Afin de compléter mon analyse j'ai cherché à connaître le prix d'un tel projet effectué par un professionnel. Malheureusement, après avoir identifié une entreprise sur Genève proposant ce genre de services, je n'ai pas pu me permettre de payer un devis. Cependant la personne de contact a gentiment répondu à mes questions concernant le rôle de l'Open Source dans son activité. Et les différences qu'il peut y avoir en utilisant des technologies de ce type pour ce genre de projet.

Il s'agit de Mr. Théo Reichel de Complex IT¹¹⁰ voici les réponses à mes questions :

Moi : J'essaie de trouver une idée de tarif pour développer un produit de ce type sans utiliser de technologies libres (ici l'arduino) auriez-vous une idée ?

TR : A développement équivalent il n'y a généralement pas de grande différence sur le prix d'une solution open source ou propriétaire.

Par ailleurs les kits de développement open source/propriétaire sont d'un prix relativement égal.

On trouve plusieurs explications logiques à ce phénomène. D'une part le prix de la production du matériel open source, même si la question des royalties ou d'une valeur ajoutée liée au droit d'auteur est écartée, reste le même que pour un produit propriétaire. Si le coût d'une solution propriétaire (comme le launchpad de TI) rejoint celle du design open source c'est notamment à cause de la concurrence : des appareils open source ont tiré le prix des kits de développement vers le bas ces 5 dernières années !

¹¹⁰ <http://www.complex.ch/>

Quant à la question du développement, il s'agit d'un travail incompressible qui doit être effectué une fois par un ingénieur, lequel ne sera plus rétribué une fois le design ouvert. Il a donc généralement la responsabilité de produire un objet durable, car son design sera non seulement consulté mais en plus dérivé (dans le meilleurs des cas, comme c'est le cas de l'Arduino).

Moi : Quels sont pour vous les avantages à utiliser des technologies de ce type ?

TR : Il n'y a pas de réelle marge à dégager lors de la commande d'un produit open source sur mesure (qui engage un développement personnalisé), si ce ne sont tous les avantages qu'offre l'open source, à savoir :

1. la durabilité et l'écologie

- l'adaptation plus facile*
- la transmission à d'autres ingénieurs*
- la possibilité de réparer le produit*
- etc.*

2. l'économie

- la reproduction facile de la solution*
- la mise en commun de technologie parfois onéreuse à développer, donc un meilleur rapport qualité/prix.*

3. la formation

- transmission des compétences*
- la sauvegarde du savoir*

Moi : Qu'est-ce que l'Open Hardware apporte à votre activité ?

TR : L'initiative open source nous permet de produire plus rapidement des produits mieux adaptés pour nos clients, à coût restant concurrentiel.

Et ceci nous pouvons le faire en suivant nos critères de qualités, lesquels prennent en considération les paramètres écologiques et sociaux.

Il ne s'agit donc pas d'éléments purement économiques, mais aussi d'engagement moral.

Conclusion

Open, un terme bien à la mode !

Au-delà de l'Open Source c'est l'aspect participatif, dans une période de récession, qui a la cote. À l'heure où le pouvoir d'achat diminue, où l'éducation et le savoir souffrent d'une certaine privatisation et que la crise menace de plus en plus d'entreprises, internet semble un merveilleux moyen de communication et d'organisation pour mettre en commun les ressources de chacun et se passer d'intermédiaires jusqu'alors indispensables.

Des sites d'achats groupés voient le jour afin de faire diminuer les prix.⁽¹¹¹⁾ Des organisations de production participative permettent à n'importe qui de contribuer à la production d'un album, d'un livre, d'un film, d'un jeu vidéo ou d'une entreprise, évitant ainsi le monopole des grosses compagnies de l'industrie du divertissement. ⁽¹¹²⁾ Les forums d'entraide permettent à tout un chacun d'acquérir les connaissances nécessaires à la réalisation d'un projet qu'il soit culinaire, technique ou artistique. Prouvant à tous que tout est réalisable à condition d'y consacrer l'énergie nécessaire et que l'innovation peut venir de n'importe qui.

L'espoir d'une économie plus saine

En ces temps de crises nos régions cherchent de nouvelles alternatives au consumérisme et à la surproduction. S'orienter vers une économie durable tournée vers le service et une production locale de qualité avec un impact écologique faible.

Devant les différents scandales liés principalement à des systèmes de production massif (alimentaire, énergétique, technologique) ou à des organisations trop lourdes pour se permettre de conserver une production locale (délocalisation). Le consommateur présente un intérêt grandissant pour la provenance et la qualité des produits qu'il consomme et tend à favoriser, dans la mesure où il a le choix, des produits durables et une production locale. Cette démarche diminue la consommation inutile et rapproche le consommateur de son produit.

¹¹¹ <http://www.topdeal.ch/>

¹¹² <http://www.mymajorcompany.com/>

L'open Source qu'il soit logiciel ou matériel favorise ce type d'économie en pérennisant les produits. Des produits fiables et évolutifs et dont la valorisation réside plus sur les services annexes que sur la production.

Exemple : aujourd'hui on conseillera à un utilisateur de changer d'ordinateur lorsqu'il a une version de Windows obsolète car une nouvelle licence lui coûtera le quart d'une nouvelle machine. Ce rachat à neuf en permanence exclut toute réparation ou entretien ce qui tue l'économie locale spécialisée dans ce type de services au profit de grandes sociétés de production de matériel. L'Open Source, en maintenant des versions de systèmes d'exploitation gratuits et peu gourmands, permet à l'utilisateur de conserver sa machine.

Impact collatéral

L'Open Hardware, au même titre que l'Open Source, crée une concurrence qui pousse les entreprises à proposer des produits innovants, à s'adapter à des standards et à baisser leurs prix (exemple cité plus haut avec Gadgeteer ou LaunchPad). Devant cet engouement et la démonstration de l'efficacité des différents modèles d'affaires de l'Open Source, ces entreprises décident même d'ouvrir leurs produits afin de les rendre plus populaires et miser sur le rendement de services annexes. Parce qu'elles comprennent que la valeur de leurs technologies est proportionnelle à leur popularité et qu'aujourd'hui, devant la qualité des solutions Open Source présentent sur le marché, il faut que leurs produits restent performants et accessibles au risque de ne pas perdurer.

Perspectives

L'Open Hardware a donc de beaux jours devant lui et si l'Open Source logiciel a fait ses preuves et s'est aujourd'hui implanté un peu partout il semble que le matériel libre suive la même direction. Pour l'instant les projets commercialisés ont comme clients principalement des communautés de bricoleurs. Cependant certains projets de plus grande envergure commencent à voir le jour. Par exemple différents véhicules sont en cours de développement et si l'on se profile sur l'exemple de l'évolution de l'Open Source logiciel et que ceux-ci sont commercialisés un jour, l'industrie automobile devra certainement prendre des mesures afin de s'adapter à cette nouvelle concurrence.

Interrogations

Ces mouvements sont actuellement dépendants de leurs contributeurs et si le « fais-le-toi-même » (DIY) fait aujourd'hui la popularité de ce type de produits qu'en sera-t-il dans les années à venir ?

Est-ce que ce modèle n'est pas la simple démonstration que la collaboration peut être profitable à tous y compris à de grandes multinationales ?

Est-ce le signe de saturation d'une politique trop centrée sur le rendement, le profit ou de manière plus générale : l'égoïsme ?

Au final

Dans tous les cas L'Open Source est bénéfique au consommateur que ce soit en l'utilisant directement ou par le biais de la concurrence qu'il crée. Incitant ainsi les entreprises à ne pas franchir une certaine « limite » éthique dans la conservation, la commercialisation ou le développement de leurs produits, au risque de voir un modèle similaire développé par une communauté lui faire de l'ombre.

Lexique

Tutoriel	Le tutoriel désigne un guide d'apprentissage permettant à un débutant de se former de manière autonome
Android	Android est un système d'exploitation spécialisé dans les périphériques mobiles. Utilisant le noyau Linux et conçu par Android, une startup rachetée par Google.
ARPANET	ARPANET ou Arpanet (acronyme anglais de « Advanced Research Projects Agency Network », souvent typographié « ARPAnet ») est le premier réseau à transfert de paquets développé aux États-Unis par la DARPA. Le projet fut lancé en 1969 et la première démonstration officielle date d'octobre 1972. (Wikipedia, ARPANET)
BIND	BIND (Berkeley Internet Name Domain) est un système d'adressage des sites internet permettant de ne pas avoir à entrer l'adresse au format numérique mais sous forme de noms de domaines. (Benkeltoum, 2011)
Bluetooth	Bluetooth est un standard de communication utilisant les ondes radios afin de connecter entre eux différents appareils électroniques.
Bug	En informatique, un bug (de l'anglais bug, « insecte ») ou bogue est un défaut de conception d'un programme informatique à l'origine d'un dysfonctionnement. Ce nom vient du tout premier incident informatique qui a été causé par un insecte. (Wikipedia, Bug (informatique))
Cloud Computing	Le Cloud computing est une nouvelle manière de fournir et d'utiliser les aptitudes des systèmes informatiques basée sur les nuages (cloud en anglais). Un nuage est un parc de machines, d'équipement de réseau et de logiciels maintenu par un fournisseur, que les consommateurs peuvent utiliser en libre-service via Internet. (Wikipedia, Cloud computing)
CLUF	Le terme Contrat de Licence Utilisateur Final (CLUF) est une traduction du terme anglais EULA, End User License Agreement. Il s'agit d'un contrat liant une personne installant un logiciel affecté par ce type de licence sur un/son ordinateur et l'éditeur du logiciel. La plupart d'entre elles, que l'on dénomme aussi licences de logiciels propriétaires, limitent le nombre de machines sur lesquelles on peut installer le logiciel, le nombre d'utilisateurs qui peuvent utiliser le logiciel, et contiennent d'autres limitations qui ne sont pas inhérentes à la technologie. (Wikipedia, License de logiciel)
CNLL	Conseil national du logiciel libre (France) - http://cnll.fr/
Code source	Le code source est un texte qui représente les instructions qui doivent être exécutées par un microprocesseur. Le code source est généralement écrit dans un langage de programmation permettant ainsi une meilleure compréhension par des humains. Une fois le code source écrit, il permet de générer une représentation binaire d'une séquence d'instructions code binaire (ou langage machine) exécutables par un microprocesseur. (Wikipedia, Code source)
Compiere	Compiere est un progiciel de gestion intégré (PGI) et gestion de la relation client (GRC) open source pour les Petites et moyennes entreprises (PME) dans la distribution et le service. (Wikipedia, Compiere)

Copyleft	Le copyleft est l'autorisation donnée par l'auteur d'un travail soumis au droit d'auteur (œuvre d'art, texte, programme informatique ou autre) d'utiliser, d'étudier, de modifier et de copier son œuvre, dans la mesure où cette autorisation reste préservée. L'auteur refuse donc que son travail puisse évoluer avec une restriction du droit à la copie. De ce fait, le contributeur apportant une modification (correction, ajout, réutilisation, etc.) est contraint de redistribuer ses propres contributions avec les mêmes conditions d'utilisation que l'original. Autrement dit, les créations réalisées à partir d'éléments sous copyleft héritent de facto ce copyleft. (Wikipedia, Copyleft)
Creative Commons	Le Creative Commons (CC) est une organisation à but non lucratif dont le but est de proposer une solution alternative légale aux personnes souhaitant libérer leurs œuvres des droits de propriété intellectuelle standards de leur pays, jugés trop restrictifs. L'organisation a créé plusieurs licences, connues sous le nom de licences Creative Commons. Ces licences, selon leur choix, restreignent seulement quelques droits (ou aucun) des travaux. (Wikipedia, Creative Commons)
CUD ou Communauté d'utilisateurs développeurs	CUD ou Communauté d'utilisateurs développeurs désignent une communauté internet d'utilisateurs et/ou de développeurs d'un logiciel libre. Chaque projet de logiciel libre a sa propre communauté d'utilisateurs développeurs. (Benkeltoum, 2011)
Defense Distributed	Defense Distributed est une association en ligne de fabrication d'armes à feu open-source, appelé Wiki Weapon. L'un des objectifs de l'organisation est de développer et de publier librement les schémas de conception d'armes à feu qui peuvent être téléchargés et reproduits par n'importe qui avec une imprimante 3D. (Wikipedia, Defense Distributed)
DIY	Do it yourself (DIY) est une appellation, dont une traduction littérale en français serait « Faites-le vous-même ».
Evolution	Evolution (ou Novell Evolution) est un logiciel de groupe de travail libre. Il sert de client de messagerie, carnet d'adresses, agenda, portail d'information. (Wikipedia, Evolution (logiciel))
Firefox	Mozilla Firefox est un navigateur Web libre et gratuit, développé et distribué par la Mozilla Foundation avec l'aide de milliers de bénévoles grâce aux méthodes de développement du logiciel libre/open source. (Wikipedia, Mozilla Firefox)
Firmware	Un firmware est un ensemble d'instructions et de structures de données qui sont intégrées dans un appareil électronique pour qu'il puisse fonctionner
Fondeur	Une société fabricant des puces électroniques
Free	Free, entreprise filiale du groupe Iliad, est l'un des principaux fournisseurs d'accès à Internet en France.
Freebox	La Freebox est un appareil électronique fourni par le fournisseur d'accès à Internet français Free à ses abonnés haut débit.
Freeware	Un freeware, graticiel ou gratuiciel, est un logiciel propriétaire distribué gratuitement sans toutefois conférer à l'utilisateur certaines libertés d'usage associées au logiciel libre. (Wikipedia, Freeware)
FSF (Free Software Foundation)	La Free Software Foundation (FSF) (littéralement « Fondation pour le logiciel libre »), est une organisation américaine à but non lucratif fondée par Richard Stallman le 4 octobre 1985, dont la mission mondiale est la promotion du logiciel libre et la défense des utilisateurs. (Wikipedia, Free Software Foundation)
Gadgeteer	Microsoft .NET Gadgeteer est une plateforme de prototypage rapide Open Source programmable avec le langage .NET (propriété de Microsoft)

GNU	GNU est un système d'exploitation libre lancé en 1984 par Richard Stallman et maintenu par le projet GNU. Son nom est un acronyme récuratif qui signifie en anglais « GNU's Not UNIX » (littéralement, « GNU n'est pas UNIX »). Il reprend les concepts et le fonctionnement d'UNIX. (Wikipedia, GNU)
Hacker	Le hacker est comme une personne qui se délecte de la compréhension approfondie du fonctionnement interne d'un systèmes. Cette dernière définition fait écho à celle utilisée depuis les années 1950 par les radioamateurs pour qui le hacking est un bricolage créatif visant à améliorer le fonctionnement d'un système.
Hardware	Hardware signifie matériel
HTML	L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. (Wikipedia, HTML)
Intégrateur	Un intégrateur (dans ce contexte) est une société chargée d'assembler différents logiciels afin de réaliser une application
Internet	Internet est un système d'interconnexion de machines et constitue un réseau informatique mondial, utilisant un ensemble standardisé de protocoles de transfert de données. (Wikipedia, Internet)
Internet Explorer	Logiciel de navigation internet (navigateur) développé par la société Microsoft™
Kinect	Kinect est un périphérique destiné à la console de jeux vidéo Xbox 360 permettant de contrôler des jeux vidéo sans utiliser de manette. (Wikipedia, Kinect)
langage machine	Le langage machine, ou code machine, est la suite de bits qui est interprétée par le processeur d'un ordinateur exécutant un programme informatique. C'est le langage natif d'un processeur, c'est-à-dire le seul qu'il puisse traiter. Il est composé d'instructions et de données à traiter codées en binaire. (Wikipedia, Language machine)
LED	LED ou diode électroluminescente est un composant opto-électronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. (Wikipedia, Diode électroluminescente)
Linux	Linux est un système d'exploitation libre créé en 1991, sur l'initiative de Linus Torvalds.
MicroSD	Le format microSD est une unité de stockage qui utilise de la mémoire flash. Ce format est souvent utilisé dans les périphériques mobiles.
MySQL	MySQL est un système de gestion de base de données (SGBD). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde. (Wikipedia, MySQL)
Netscape	Société éditeur de logiciels fondée en 1994, rachetée en 1998 par AOLTM puis dissolue en 2003.
NTIC	Nouvelles Technologies de l'Information et de la Communication : regroupent les techniques utilisées dans le traitement et la transmission des informations, principalement de l'informatique, de l'Internet et des télécommunications.
Open Source	La désignation open source s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire la possibilité de libre redistribution, d'accès au code source et aux travaux dérivés. (Wikipedia, Open Source)
Open Source Initiative	L'Open Source Initiative est une organisation dédiée à la promotion des logiciels open source. (Wikipedia, Open Source Initiative)
OSHW	Open Source HardWare (Matériel libre sous licence Open Source)

OSHW	L'Open Hardware Repository (OHWR) est un projet lancé à l'initiative de concepteurs en électronique travaillant dans des laboratoires de physique expérimentale qui ont estimé nécessaire de faciliter les échanges à très grande échelle, pour aller dans le sens d'une science « ouverte », encouragée par des organisations telles que le CERN. (Cern)
Qt	Qt est une API orientée objet. Qt offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. (Wikipedia, Qt)
Red Hat Enterprise Linux	Red Hat Enterprise Linux (souvent abrégé RHEL) est une distribution Linux produite par Red Hat et orientée vers le marché commercial et les serveurs d'entreprise. (Wikipedia, Red Hat Enterprise Linux)
Relai (électromécanique)	Un relai électromécanique est un composant permettant la commutation de liaisons électriques. Il s'agit d'un conducteur magnétique chargé d'établir le contact ou non d'une liaison électrique (de plus grande puissance). Il est souvent piloté par du courant faible et permet le contact de liaisons de courant fort.
RJ45	Connecteur utilisé couramment pour les connexions Ethernet ou réseau filaires entre ordinateurs et périphériques compatible (imprimantes, webcam, etc.)
SeedStudio	SeedStudio, au même titre que SparkFun vend est un détaillant de composant électronique mais il entretient également une communauté d'utilisateur-développeur orientés autour de projets Open Hardware.
SmartWatch	Une smartwatch, littéralement « montre intelligente », est une montre bracelet informatisée avec des fonctionnalités allant au-delà du simple chronométrage, présentant des caractéristiques comparables à celles d'un PDA. Alors que les premiers modèles, apparus dans les années 1980, étaient capables d'effectuer des tâches de base comme les calculs, l'agenda-planning puis les traductions ou les jeux, les smartwatches modernes sont devenues des ordinateurs à part entière. Certaines smartwatches peuvent exécuter des applications mobiles. (Wikipedia, SmartWatch)
SparkFun Electronics	SparkFun Electronics (parfois connu sous son abréviation, SFE) est un détaillant de composants électroniques basé à Boulder, dans le Colorado, aux États-Unis. Il confectionne et vend des cartes de développement de microcontrôleur et des circuits imprimés. (Wikipedia, SparkFun Electronics)
SpikeSource	SpikeSource est une société spécialisée dans le test logiciel automatisé. (Wikipedia, SpikeSource)
SugarCRM	SugarCRM est une solution Open Source de gestion de la relation client éditée par la société américaine SugarCRM. (Wikipedia, SugarCRM)
SUSE	SUSE est une distribution Linux majeure, d'origine allemande et principalement développée en Europe. (Wikipedia, SUSE)
Synology	Synology est une société spécialisée dans les serveurs de stockage réseau (NAS).
Système d'exploitation	Un système d'exploitation (OS pour Operating System en anglais) désigne un logiciel qui joue le rôle d'intermédiaire entre un utilisateur, ses programmes et le matériel sur lequel il est installé (exemple : Windows, Mac OS, Linux)
Système embarqué	Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé. (Wikipedia, Système embarqué)
TCP/IP	Le TCP/IP (Transmission Control Protocol/Internet Protocol) est un protocole de communication conçu en 1974 afin de contourner la disparité des systèmes.

Telnet	Telnet est un protocole réseau, son but est de fournir un moyen de communication très généraliste souvent utilisé pour le contrôle d'appareils via le réseau.
TYPO3	TYPO3 est un système de gestion de contenu libre écrit en PHP. (Wikipedia, SpikeSource)
Ubuntu	Ubuntu est un système d'exploitation libre fondé sur Linux et commandité par la société Canonical et une marque déposée par cette même société. (Wikipedia, Ubuntu)
Unix	Unix, est un système d'exploitation multitâche et multi-utilisateur créé en 1969, conceptuellement ouvert et fondé sur une approche par laquelle il offre de nombreux petits outils, chacun doté d'une mission spécifique. (Wikipedia, Unix)

Bibliographie

- Arduino. (n.d.). *OSHW summit speech 2011*. Retrieved 06 10, 2013, from slideshare.net:
<http://fr.slideshare.net/arduinoteam/open-source-hardware-summit-speech-2011>
- Arduino. (n.d.). *Page d'accueil de la référence Arduino en français*. Retrieved 06 10, 2013, from arduino: <http://www.arduino.cc/fr/>
- Benkeltoum, N. (2011). *Gérer et comprendre l'open source*.
- celtic.be. (n.d.). *L'open source comme modele d'affaire dans les TIC*. Retrieved 06 22, 2013, from celtic.be: <http://www.cetic.be/L-Open-Source-comme-modele-d>
- CERN. (n.d.). *Cern Open Hardware Licence*. Retrieved 06 10, 2013, from ohwr.org: <http://www.ohwr.org/documents/88>
- Cern. (n.d.). *Le CERN lance l'initiative sur le matériel libre*. Retrieved 06 24, 2013, from CERN: <http://press.web.cern.ch/fr/press-releases/2011/07/le-cern-lance-linitiative-sur-le-materiel-libre>
- CNLL. (2010). *Cartographie des entreprises du logiciel libre du CNLL*. CNLL.
- CNLL. (2010, 06). *Cartographie des entreprises du logiciel libre du CNLL*. Retrieved 06 10, 2013, from CNLL: <http://www.cnll.fr/static/pdf/cnll-carto-07.pdf>
- complex.ch. (n.d.). *index*. Retrieved 07 10, 2013, from complex.ch: <http://www.complex.ch/>
- culture.gouv.fr. (n.d.). *La propriété littéraire et artistique*. Retrieved 06 02, 2013, from culture.gouv.fr: <http://www.culture.gouv.fr/culture/infos-pratiques/droits/oeuvres.htm>
- Ecology, O. S. (n.d.). *index*. Retrieved 07 5, 2013, from opensourceecology.org: <http://opensourceecology.org/>
- Evola.fr. (n.d.). *Platine relais*. Retrieved 07 10, 2013, from evola.fr: http://www.evola.fr/product_info.php/platine-relais-p-102
- extremetech.com. (n.d.). *Gadgeteer: Microsoft's open-source .NET alternative to Arduino*. Retrieved 07 10, 2013, from extremetech.com: <http://www.extremetech.com/computing/91990-gadgeteer-microsofts-open-source-net-alternative-to-arduino>
- Fabernovel, C. (2007). *BUSINESS MODEL OPEN SOURCE*. Retrieved 06 20, 2013, from fr.slideshare.net: <http://fr.slideshare.net/gillesmu/business-model-open-source>
- freebsd. (n.d.). *license*. Retrieved 06 10, 2013, from freebsd.org: <http://www.freebsd.org/copyright/license.html>
- futura-science.com. (n.d.). *Open hardware : Sony ouvre sa montre SmartWatch au hacking*. Retrieved 07 10, 2013, from futura-science.com: <http://www.futura->

sciences.com/magazines/high-tech/infos/actu/d/technologie-open-hardware-sony-ouvre-montre-smartwatch-hacking-47160/

GNU. (n.d.). *GPL*. Retrieved 06 10, 2013, from gnu.org: <http://www.gnu.org/licenses/gpl.html>

GNUPG. (n.d.). *gnupg*. Retrieved 06 10, 2013, from gnupg.org:
<http://www.gnupg.org/index.fr.html>

Greenit. (n.d.). *Après le logiciel libre*. Retrieved 06 23, 2013, from Greenit:
<http://www.greenit.fr/article/materiel/apres-le-logiciel-libre-le-materiel-ouvert-et-libre-4191>

Instrument, T. (n.d.). *Get started with the LaunchPad Evaluation Platform from Texas Instruments*. Retrieved 07 10, 2013, from ti.com:
http://www.ti.com/ww/en/launchpad/home_head.html?DCMP=Value_Line&HQS=launchpad

journaldunet.com. (n.d.). *open Hardware définition*. Retrieved 06 24, 2013, from journaldunet.com: <http://www.journaldunet.com/solutions/magazine/open-hardware-et-materiel-libre-definition.shtml>

linternaute. (n.d.). *Intégrateur*. Retrieved 06 21, 2013, from linternaute:
<http://www.linternaute.com/dictionnaire/fr/definition/integrateur/>

Metycea. (s.d.). *blog de metycea*. Consulté le 06 02, 2013, sur blog de metycea - qu'est-ce que l'open source: <http://blog.metycea.com/actus-internet/definition-qu-est-ce-que-l-open-source-2-4.html>

mymajorcompany.com. (n.d.). *MyMajorCompany, Tout le financement participatif*. Retrieved 07 10, 2013, from mymajorcompany.com: <http://www.mymajorcompany.com/>

OpenKinect.org. (n.d.). *Main Page*. Retrieved 07 10, 2013, from openkinect.org:
http://openkinect.org/wiki/Main_Page

oshpark.com. (n.d.). *OSH Park An electric ecosystem*. Retrieved 07 10, 2013, from oshpark.com: <http://oshpark.com/>

OSHW. (n.d.). *Hardware Libre*. Retrieved 06 27, 2013, from freedomdefined.org:
<http://freedomdefined.org/OSHW/translations/fr>

OSHWA. (n.d.). *oshwa.org*. Retrieved 06 10, 2013, from oshwa.org: <http://www.oshwa.org/>

pcinpact. (n.d.). *firefox-google-mozilla-partenariat*. Retrieved 06 10, 2013, from pcinpact.com:
<http://www.pcinpact.com/news/67789-firefox-google-mozilla-partenariat.htm>

pgp, O. (n.d.). *openpgp.org*. Retrieved 06 10, 2013, from openpgp.org: <http://openpgp.org>

Project, O. P. (n.d.). *index*. Retrieved 06 10, 2013, from openprosthetics.org/:
<http://openprosthetics.org/>

Raspberrypi.org. (n.d.). *raspberrypi*. Retrieved 06 10, 2013, from raspberrypi.org:
<http://www.raspberrypi.org/>

Raymond, E. S. (1998). *The Cathedral and the Bazaar*.

sparkfun.com. (n.d.). *Bluetooth Modem - BlueSMiRF Gold*. Retrieved 07 10, 2013, from
 sparkfun.com: <https://www.sparkfun.com/products/10268>

TAPR. (n.d.). *Open Hardware Licence*. Retrieved 06 10, 2013, from tapr.org:
<http://www.tapr.org/ohl.html>

topdeal.ch. (n.d.). *index*. Retrieved 07 10, 2013, from topdeal.ch: <http://www.topdeal.ch/>

whirlwindwheelchair. (n.d.). *index*. Retrieved 06 10, 2013, from whirlwindwheelchair.org:
<http://www.whirlwindwheelchair.org/>

Wikipedia. (n.d.). *ARPANET*. Retrieved 05 19, 2011, from Wikipedia:
<http://fr.wikipedia.org/wiki/Arpanet>

Wikipedia. (s.d.). *BIND*. Consulté le 05 19, 2013, sur Wikipedia:
<http://fr.wikipedia.org/wiki/BIND>

Wikipedia. (n.d.). *Bruce Perens*. Retrieved 06 28, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Bruce_Perens

Wikipedia. (n.d.). *Bug (informatique)*. Retrieved 06 18, 2013, from Wikipedia:
[http://fr.wikipedia.org/wiki/Bug_\(informatique\)](http://fr.wikipedia.org/wiki/Bug_(informatique))

Wikipedia. (n.d.). *Cloud computing*. Retrieved 06 22, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Cloud_computing

Wikipedia. (n.d.). *Code source*. Retrieved 05 19, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Code_source

Wikipedia. (n.d.). *Compiere*. Retrieved 06 22, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Compiere>

Wikipedia. (n.d.). *Composant logiciel*. Retrieved 06 20, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Composant_logiciel

Wikipedia. (n.d.). *Copyleft*. Retrieved 06 02, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Copyleft>

Wikipedia. (n.d.). *Creative Commons*. Retrieved 06 23, 2013, from Wikipedia:
https://fr.wikipedia.org/wiki/Creative_Commons

Wikipedia. (n.d.). *Defense Distributed*. Retrieved 06 28, 2013, from Wikipedia:
https://fr.wikipedia.org/wiki/Defense_Distributed

Wikipedia. (n.d.). *Diode électroluminescente*. Retrieved 06 27, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Diode_%C3%A9lectroluminescente

Wikipedia. (n.d.). *Eric Raymond*. Retrieved 6 18, 2013, from Wikipedia.

Wikipedia. (n.d.). *Evolution (logiciel)*. Retrieved 06 22, 2013, from Wikipedia:
[http://fr.wikipedia.org/wiki/Evolution_\(logiciel\)](http://fr.wikipedia.org/wiki/Evolution_(logiciel))

Wikipedia. (n.d.). *Free Software Foundation*. Retrieved 05 19, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Free_Software_Foundation

Wikipedia. (s.d.). *Freeware*. Consulté le 05 19, 2013, sur Wikipedia:
<http://fr.wikipedia.org/wiki/Freeware>

Wikipedia. (n.d.). *GNU*. Retrieved 05 19, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/GNU>

Wikipedia. (n.d.). *Hacker*. Retrieved 06 24, 2013, from Wikipedia:
[http://fr.wikipedia.org/wiki/Hacker_\(universit%C3%A9\)](http://fr.wikipedia.org/wiki/Hacker_(universit%C3%A9))

Wikipedia. (n.d.). *HTML*. Retrieved 05 19, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Hypertext_Markup_Language

Wikipedia. (n.d.). *Internet*. Retrieved 05 19, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Internet>

Wikipedia. (n.d.). *Kinect*. Retrieved 06 24, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Kinect>

Wikipedia. (n.d.). *Language machine*. Retrieved 05 19, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Code_machine

Wikipedia. (n.d.). *Libre accès*. Retrieved 06 27, 2013, from Wikipedia:
[http://fr.wikipedia.org/wiki/Libre_acc%C3%A8s_\(%C3%A9dition_scientifique\)](http://fr.wikipedia.org/wiki/Libre_acc%C3%A8s_(%C3%A9dition_scientifique))

Wikipedia. (n.d.). *License de logiciel*. Retrieved 06 02, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Licence_de_logiciel

Wikipedia. (n.d.). *Linus Torvalds*. Retrieved 06 18, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Linus_Torvalds

Wikipedia. (n.d.). *Linux*. Retrieved 05 19, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Linux>

Wikipedia. (n.d.). *Logiciel libre*. Retrieved 05 19, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Logiciel_libre

Wikipedia. (n.d.). *logiciel propriétaire*. Retrieved 06 02, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Logiciel_propri%C3%A9taire

Wikipedia. (n.d.). *Matériel libre*. Retrieved 06 22, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Mat%C3%A9riel_libre

Wikipedia. (n.d.). *Mozilla Firefox*. Retrieved 06 22, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Firefox>

Wikipedia. (n.d.). *MySQL*. Retrieved 06 20, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/MySQL>

Wikipedia. (n.d.). *Open Society Foundations*. Retrieved 06 27, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Open_Society_Foundations

Wikipedia. (s.d.). *Open Source*. Consulté le 05 19, 2013, sur Wikipedia:
http://fr.wikipedia.org/wiki/Open_source

Wikipedia. (n.d.). *Open Source Ecology*. Retrieved 06 24, 2013, from Wikipedia:
https://fr.wikipedia.org/wiki/Open_Source_Ecology

Wikipedia. (n.d.). *Open Source Initiative*. Retrieved 05 19, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Open_Source_Initiative

Wikipedia. (n.d.). *OpenDocument*. Retrieved 06 27, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Open_document

Wikipedia. (n.d.). *OS X*. Retrieved 06 20, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/OS_X

Wikipedia. (n.d.). *Qt*. Retrieved 06 20, 2013, from Wikipedia: <http://fr.wikipedia.org/wiki/Qt>

Wikipedia. (n.d.). *Red Hat Enterprise Linux*. Retrieved 06 22, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Red_Hat_Enterprise_Linux

Wikipedia. (n.d.). *Ressources éducatives libres*. Retrieved 06 27, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/%C3%89ducation_ouverte

Wikipedia. (n.d.). *Richard Stallman*. Retrieved 06 28, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Richard_Stallman

Wikipedia. (n.d.). *SmartWatch*. Retrieved 06 27, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Smartwatch>

Wikipedia. (n.d.). *SparkFun Electronics*. Retrieved 06 28, 2013, from Wikipedia:
https://fr.wikipedia.org/wiki/SparkFun_Electronics

Wikipedia. (n.d.). *SpikeSource*. Retrieved 06 22, 2013, from Wikipedia:
<http://en.wikipedia.org/wiki/SpikeSource>

Wikipedia. (n.d.). *SugarCRM*. Retrieved 06 22, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/SugarCRM>

Wikipedia. (n.d.). *Suse*. Retrieved 06 22, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/SUSE>

Wikipedia. (n.d.). *SUSE*. Retrieved 06 22, 2013, from wikipedia:
<http://fr.wikipedia.org/wiki/SUSE>

Wikipedia. (n.d.). *Système embarqué*. Retrieved 06 27, 2013, from Wikipedia:
http://fr.wikipedia.org/wiki/Syst%C3%A8me_embarqu%C3%A9

Wikipedia. (s.d.). *TCP-IP*. Consulté le 05 19, 2013, sur Wikipedia:
<http://fr.wikipedia.org/wiki/TCP/IP>

Wikipedia. (n.d.). *TYPO3*. Retrieved 06 22, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/TYPO3>

Wikipedia. (n.d.). *Ubuntu*. Retrieved 06 22, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Ubuntu>

Wikipedia. (n.d.). *Unix*. Retrieved 06 18, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Unix>

Wikipedia. (n.d.). *Wikispeed*. Retrieved 06 24, 2013, from Wikipedia:
<http://fr.wikipedia.org/wiki/Wikispeed>