

```
...iale  
)  
)  
[0]>0 a  
(n, pop)  
in range(  
crossover2  
) in range(  
mutation(n,  
assement(pop)  
if scores[0] <  
nb conflicts  
dames_best=  
dames_best[  
generation
```

Sommaire

I. Avant-propos.....	3
II. Prérequis.....	4
a) Logiciel libre.....	4
b) Code source.....	7
c) Licence libre.....	7
d) Licence copyleft.....	8
e) Logiciel propriétaire.....	8
f) Distribution ou distribution Linux.....	9
g) Compilation.....	9
h) Bug.....	10
i) Patch.....	10
j) Fourchette (fork).....	10
k) Paquetage.....	10
l) Système embarqué.....	10
III. Introduction.....	11
IV. Définition du modèle économique.....	12
V. Le mode de création de logiciels libres.....	12
VI. Base des modèles économiques du logiciel libre.....	14
VII. L'apparition du marché des logiciels libres et ses acteurs.....	16
VIII. Les modèles économiques.....	18
1. Le modèle de services.....	18
a) Présentation.....	18
b) Analyse.....	19
1) Modèle de services simples.....	19
2) Modèle de valorisation indirecte.....	21
c) Exemple.....	22
2. Modèle de distribution à valeur ajoutée.....	23
a) Présentation.....	23
b) Analyse.....	23

c) La valeur client.....	24
d) Exemple.....	25
3. Modèle de licences double.....	26
a) Présentation.....	26
b) Analyse des deux licences.....	26
c) Deux piliers forts.....	27
d) Exemple.....	28
e) Variante : la commercialisation d'un produit complémentaire.....	28
4. Le modèle de mutualisation.....	29
a) Présentation.....	29
b) Analyse.....	29
c) Création d'une communauté clients.....	30
d) Exemple.....	32
e) Variante : Mutualisation des modules des développeurs.....	32
5. Autres modèles.....	33
IX. Synthèse.....	34
X. Conclusion.....	35
XI. Annexe.....	36
1. Déclaration.....	36
2. Bibliographie.....	36
a) Livre.....	36
b) Entretiens.....	36
c) Sitographie.....	37

I. Avant-propos :

Ce travail a pour but de **présenter les modèles économiques** les plus importants qui sont sur le marché des logiciels libres. Il existe en effet d'autres modèles qui peuvent en découler, par ailleurs, différents modèles peuvent se combiner entre eux. Le choix d'un travail de maturité sur ces modèles économiques a été motivé par l'envie de lier une passion, l'informatique et mon option spécifique, économie et droit.

Le travail commence par **un prérequis**, qu'il est vivement conseillé de consulter, afin de comprendre correctement le document. Il vous y sera présenté des termes techniques qui reviendront fréquemment tout au long du dossier ainsi qu'un historique succinct des logiciels libres.

Veillez prendre note que ce travail est publié sous **licence Creative Commons** dans sa version 3.0, ce qui veut dire que vous êtes libres de reproduire, distribuer, communiquer cette création et de la modifier, en respectant une seule condition : **citez le nom de l'auteur original**. Pour plus d'informations consultez le site Internet de la licence *Creative Commons* : <http://creativecommons.org/licenses/by/3.0/deed.fr>



Illustration 1: Logo représentant la licence Creative Commons

II. Prérequis :

Les quelques définitions présentées dans cette section permettent une **meilleure compréhension** du document. Effectivement ce sont des termes clés que l'on retrouve tout au long du travail.

a) Logiciel libre :

Logiciel utilisant une **licence libre** (ce terme est expliqué au point II. c))

L'histoire du logiciel libre est évidemment liée à celle de l'informatique.

Il ne s'agit pas de faire un historique des dates qui ont marquées l'informatique, notons seulement que, jusque dans les années 70, elle est **réservé** aux universités et aux grandes entreprises. A cette époque, l'utilisateur est libre le plus souvent d'étudier, de modifier et de redistribuer le logiciel comme il le souhaite, ce qui est bénéfique pour les entreprises. Etant donné le peu de personnes capables de programmer, elles partagent ainsi leurs connaissances entre elles. Le logiciel s'améliore nettement afin de répondre aux besoins des clients. Au fil des années, les entreprises décident de changer de licence des logiciels qu'ils ont développés en **imposant** des nouvelles **licences plus restrictives**, dans le but de protéger leurs créations. C'est dans ce contexte que Richard Stallman, programmeur au *Massachusetts Institute of Technology (MIT)* constate qu'il n'arrive plus à modifier certains pilotes pour faire fonctionner son imprimante. Il décide alors de tout faire pour remédier à ce problème afin que d'autres utilisateurs ne soient plus confrontés à ce genre de situation par la suite. Il fonde en 1985 la **Free Software Foundation (FSF)** qui garantit les libertés liées aux logiciels libres. Auparavant, il crée **un système d'exploitation entièrement libre, GNU (GNU's Not Unix)**. Pour ce faire, il doit appliquer les principes qui lui sont chers : le droit d'**utiliser**, d'**étudier**, de **modifier** et de **redistribuer** un logiciel.

Pour être certain de mener à bien son projet, lui et ses collaborateurs développent le concept « d'interdire d'interdire » qui est représenté par le **copyleft**.

Le copyleft représente le « gauche d'auteur », donc contrairement au copyright, au lieu de limiter les droits de l'utilisateur, il lui en donne.

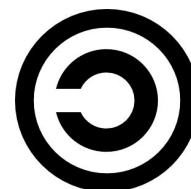


Illustration 2:

Logo du Copyleft

Le copyleft est un jeu de mot, car il oppose le terme *left* (gauche) à *right* (droit), à prendre dans le sens juridique. De plus, il s'oppose au droit de copier (*copyright*), en effet il peut se traduire en français par « copie laissée ».

Plus tard, des **licences copyleft** apparaissent pour rendre légal ce concept et pour permettre à ceux qui le souhaitent de l'utiliser.

Les programmeurs développent souvent des logiciels sous licences libres en réponse à un logiciel propriétaire similaire, par exemple, le célèbre *OpenOffice.org* qui concurrence la suite bureautique de *Microsoft*. On dénomme ce type de programme un « **clone** ».

Le grand public imagine souvent que les logiciels publiés sous licences libres sont forcément gratuits, **ce qui est faux**. Effectivement, il existe beaucoup de logiciels propriétaires (voir point II. e)) qui sont distribués gratuitement (*freeware* ou **gratuiciel** en français). Comme exemple de logiciel gratuit, on peut citer l'antivirus *Avast!* qui est distribué gratuitement mais qui n'est pas libre. Ce logiciel offre la gratuité et non pas la liberté de modifier le code source (voir point II. b)), comme le font les logiciels libres. Il faut encore faire la distinction avec les programmes qui sont distribués gratuitement mais qui ont une durée d'utilisation limitée (*shareware* ou **partagiciel**), cette durée passée l'utilisateur doit rétribuer le développeur s'il veut continuer à l'utiliser. *WinRar* est un utilitaire de ce type permettant de décompresser des archives. De ce fait, les logiciels gratuits ne sont pas obligatoirement libres. En effet, une licence libre autorise **la vente des logiciels**. Ces derniers une fois acquis, la licence garantit les libertés définies par celle-ci.

Tableau récapitulatif :

	Libre	Freeware	Shareware	Propriétaire
Utiliser	OUI	OUI	Limité	Limité (licence)
Copier	OUI	OUI	OUI	NON
Etudier	OUI	NON	NON	NON
Modifier	OUI	NON	NON	NON
Ordre de prix	0 ou \$	0	\$	\$\$\$

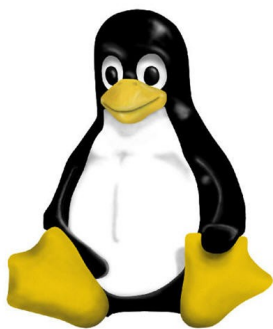


Illustration 3:

*Tux, la mascotte
de Linux*

Les logiciels libres ainsi que leur concept sont présents dans **tous les domaines informatiques**, tels que : système d'exploitation (*GNU/Linux*), langages de programmation (*Python, C, PHP*), serveurs (web, messagerie), logiciels de sécurité, environnements de travail (*KDE, Gnome*) et logiciels de bureautique (*OpenOffice.org, Mozilla Firefox*), évidemment, la liste n'est pas exhaustive. Il existe de nombreux sites proposant un catalogue de logiciels libres, comme <http://www.framasoft.net/>.

A noter que les formats ouverts (convention d'enregistrement de données dont les spécificités techniques sont publiques et sans restriction) dépendent fortement des logiciels qui les utilisent. Réciproquement, ces formats sont souvent implantés dans des logiciels libres. Ceci leur donne deux avantages : la **transparence d'accès** (code source fourni) ainsi qu'une **large diffusion**, par exemple le format ouvert d'*OpenOffice.org*, l'*Open Document (odt)* qui est devenu un standard, est diffusé avec le logiciel de bureautique.

Le format ouvert garantit la **pérennité des données**, car **il est le même quelle que soit la version du programme utilisé**.

Les **contributeurs** aux logiciels libres sont **variés** :

- Une **grande communauté de programmeurs** composée de bénévoles, chercheurs, professeurs, entreprises et administrations.
- **Traducteurs** : partout dans le monde des utilisateurs s'engagent à traduire un programme dans leur langue maternelle.
- **Rapporteurs de bugs**, très utiles aux programmeurs, effectivement ils repèrent les bugs rencontrés lors de l'utilisation des programmes.

Cette diversité est d'une grande utilité.

b) Code source :

Le code source est **le code qui est à la base de tout programme**. Il est écrit par le programmeur dans **un langage particulier** pour permettre à l'ordinateur de transformer les instructions qu'il lui donne en tâches bien précises.

Pour mieux comprendre, on peut comparer le code source à une **recette de cuisine**. Effectivement, dans une recette de cuisine, il faut des ingrédients avec des quantités précises. Ils doivent être mélangés dans un certain ordre pour atteindre le but souhaité. Le code source est similaire, une suite d'instructions est écrite par le programmeur dans un ordre défini, ensuite l'ordinateur (= le cuisinier) exécute ce qu'il lit selon la hiérarchie qui lui est donnée.

La recette de cuisine est d'autant plus une bonne comparaison car il se peut qu'un cuisinier décide de distribuer sa recette pour que d'autres confrères puissent également reproduire son plat ou le modifier et le redistribuer. Le cuisinier peut aussi décider de conserver sa recette pour lui et de vendre son plat sans que le consommateur sache réellement ce qu'il y a dedans. Le client peut essayer de deviner les ingrédients mais n'arrivera jamais à connaître la recette entière.

Pour le programmeur c'est pareil, il peut décider de **distribuer son code** pour que d'autres l'utilisent, l'étudient, le modifient et le redistribuent (c'est le cas des programmeurs qui choisissent une licence libre). Comme pour le cuisiner, le programmeur peut décider de **distribuer uniquement le programme terminé** sans publier son code (c'est le cas des programmeurs qui choisissent une licence propriétaire).

c) Licence libre :

Elle offre **quatre libertés** :

- **utiliser le logiciel comme on le souhaite**
- **étudier le code source**
- **redistribuer** (gratuitement ou contre rémunération) des copies du logiciel
- **modifier** le logiciel et **publier** les modifications

Ce principe a été proposé par Richard Stallman dont le but était de rendre **indépendant les utilisateurs par rapport aux programmeurs**. La licence libre la plus connue et certainement la plus utilisée est la *GNU GPL (GNU Public Licence)*, mais depuis de nombreuses licences se sont créées, comme la *Mozilla Public License (MPL)*, dont *Mozilla Firefox* fait partie.

d) Licence copyleft : (GNU GPL, GFDL, ...)

Avec ce type de licence, l'auteur offre les **quatre libertés** citées plus haut. Ces licences obligent, en cas de modification, à redistribuer le nouveau programme selon **les mêmes critères** que ceux décidés par le créateur originel. Ce dernier ne permet pas que son logiciel puisse être utilisé sous licence restrictive. Ainsi, un utilisateur qui modifie le code source d'un programme n'aura pas le droit, si ce dernier est sous licence copyleft, d'en changer la licence.

La **GNU GPL (GNU General Public License)** est la licence la plus utilisée pour tous les logiciels publiés sous licence libre. Sa première version, en 1989, est publiée par Richard Stallman lui-même. Elle est basée sur le copyleft et permet la **revente**.



Illustration 4

La **GFDL (GNU Free Documentation License)** applique totalement ou partiellement les principes de la GNU GPL aux **documents écrits**, que ce soit des manuels, livres ou d'autres publications. Par contre, elle prévoit des **restrictions à propos de la modification de l'oeuvre**.

Certaines licences ont **un côté contagieux**, c'est-à-dire que si quelqu'un modifie le programme, il est obligé de le redistribuer sous **le même type de licence**, pour protéger la volonté des auteurs originaux afin de garder le travail sous licence libre. Cela permet également d'augmenter le nombre de logiciels libres disponibles.

e) Logiciel propriétaire :

Un logiciel propriétaire est le contraire du logiciel libre, ce qui implique qu'au moins une des actions garanties par les licences libres n'est pas respectée. Les logiciels propriétaires sont logiquement publiés sous licences propriétaires.

Il en découle que **seulement une ou un groupe de personnes** (physiques ou morales) **est autorisé à modifier, améliorer, redistribuer le logiciel**. Les utilisateurs sont donc **privés des quatre libertés**. L'utilisateur devient **dépendant du développeur**. Si un bug est découvert dans le programme, uniquement la ou les personnes qui est habilitée à lire le code source peuvent le résoudre. Logiquement une licence propriétaire crée **un monopole autour du logiciel distribué**.

f) Distribution ou distribution Linux :

Une distribution est **un ensemble de logiciels** (le plus souvent libres) fonctionnant grâce à un noyau Linux, le tout formant **un système d'exploitation** cohérent. Il existe **énormément de distributions** chacune ayant ses défauts et ses qualités. Le site <http://distrowatch.com/> en réunit une bonne partie.



Illustration 5:

*Ubuntu : Célèbre distribution Linux
grand public*

g) Compilation :

La transformation de **code lisible** pour un être humain en **fichiers binaires** interprétables par l'ordinateur. A noter que la compilation est à **sens unique**, on obtient, depuis le code source un exécutable qui sera le programme. Par contre, on ne peut pas obtenir le code source depuis un fichier exécutable.

h) Bug :

Un bug est **un défaut** dans un programme qui peut provoquer un calcul ou un traitement incorrect de l'ordinateur à l'origine de dysfonctionnements et de pannes. Le bug apparaît lorsqu'une erreur de programmation s'est glissée dans le logiciel.

Très souvent une plateforme de rapport de bugs est mise à disposition des utilisateurs, afin qu'ils envoient un descriptif du problème rencontré ainsi que **les circonstances** dans lesquelles il est survenu.

i) Patch :

Un patch est une partie de code que l'on rajoute à un programme pour corriger le(s) bug(s).

j) Fourchette (fork) :

Un fork fait référence à un objet (souvent un projet) ayant **une racine commune** avec un second. Ces deux objets jumeaux, au départ, après s'être séparés, suivent une évolution propre, différente, une partie de code des deux objets est similaire. Le terme fork se traduit par « fourchette » ce qui est une bonne image pour illustrer ce concept. Cette action peut aussi parfois consister à utiliser le code source du logiciel (tout à fait autorisé, vu que le code source est disponible) pour **développer un logiciel concurrent**.

k) Paquetage :

Un paquetage (ou paquet ou en anglais *package*) est une archive réunissant des fichiers informatiques, des informations et des procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation.

l) Système embarqué :

Un système embarqué est un système électronique et informatique qui a une tâche bien précise à réaliser. Il est autonome et ne possède que des ressources limitées. Il existe des systèmes embarqués dans tous les domaines : l'aéronautique, les télécommunications (routeur, téléphone portable), électroménager (four micro-onde), etc.

III. Introduction :

Le concept de Logiciel Libre est relativement **récent**. Il prend naissance au début des années 80 et connaît **un essor progressif** depuis plus de 10 ans.

Evidemment les logiciels libres dépendent du **développement récent des nouvelles technologies** et principalement d'**Internet** (qui a vu le jour avec l'aide du logiciel libre).

En considérant que le but d'une entreprise est de générer un certain **profit**, une question se pose alors : comment les entreprises distribuant des logiciels libres arrivent-elles à dégager un chiffre d'affaires ? La création d'un programme utilisé mondialement demande une équipe de programmeurs qui, évidemment reçoit un salaire pour leur travail fourni. Il se peut que le logiciel soit maintenu par **une communauté** qui n'a pas de but lucratif, elle souhaite simplement que son programme soit utilisé par le maximum de personnes. Les programmeurs qui publient de tels programmes, le font sur leur **temps libre et par passion**. Ils ont souvent une autre activité leur permettant de subvenir à leurs besoins. Ils mettent le plus souvent en place un système de **dons** par lequel les utilisateurs peuvent rémunérer le travail fourni.

Certaines entreprises distribuant des logiciels libres **vendent leurs produits**, ainsi un revenu peut se dégager. Le programme est habituellement nettement **moins onéreux** qu'un logiciel similaire avec une licence privée. Ce gain permet de couvrir uniquement certains frais.

Toutefois d'autres sources de devises sont habituellement nécessaires pour que l'entreprise soit viable. Comme exemple, on peut citer *OpenSUSE* (distribution GNU/Linux) qui propose un exemplaire gratuit, mais également des exemplaires payants (*Suse Entreprise*) qui peuvent être achetés en ligne ou dans certains commerces spécialisés.

Il existe différents modèles économiques qui sont **rentables à long terme** pour les entreprises désireuses de distribuer des logiciels libres. Ces modèles dépendent souvent des **clients** et leurs souhaits au niveau du logiciel et des **services** qui y sont liés.

Le but de cette étude est bel et bien de présenter les différents modèles économiques ou modèles d'affaires relatifs à ces logiciels.

IV. Définition du modèle économique :

L'objectif de toute entreprise est d'arriver à **générer un profit** ou au pire des cas d'éviter qu'il y ait un excédent de charges par rapport aux produits.

Le modèle économique permet **d'étudier** comment une entreprise arrive à dégager un revenu en comparant son bilan avec les entreprises concurrentes. Dans le cadre du logiciel libre, on peut **réunir** plusieurs entreprises sous **différents modèles économiques** bien distincts.

V. Le mode de création de logiciels libres :

Il s'appuie sur deux bases : **l'investissement des programmeurs et des utilisateurs** ainsi que **le partage des connaissances** entre les personnes travaillant sur le projet.

Il en résulte **l'apparition de communautés** dans lesquelles se regroupent les utilisateurs, des industriels qui publient de nouveaux outils ou leurs recherches, des éditeurs et des contributeurs individuels.

Comme exemple, on peut citer l'*APRIL (Association Pour la Recherche en Informatique Libre)* dont le but principal est de **promouvoir et défendre** le logiciel libre. Elle réunit actuellement environ **4850 adhérents et 400 entreprises**. Basée en France, les actions de celle-ci se font essentiellement sur ce territoire, ses membres viennent de partout et forment une véritable communauté.

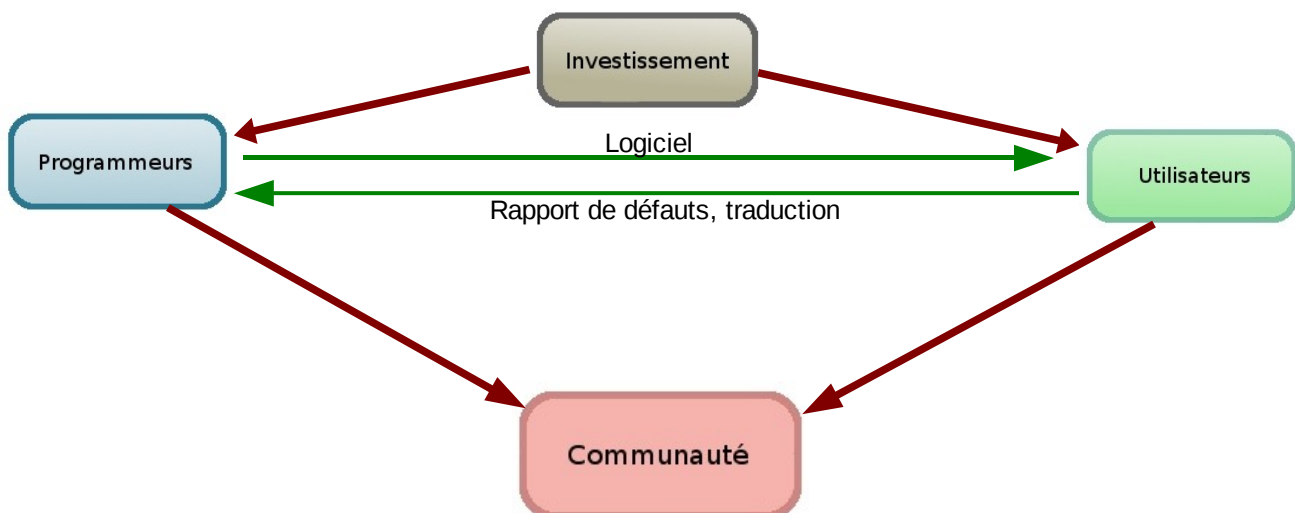
Ce modèle peut être efficace uniquement grâce au **développement récent d'Internet** et donc à une **délocalisation des agents** de la communauté, permettant ainsi **une évolution régulière** du projet. De ce fait, **l'innovation s'accélère** ce qui intéresse le grand public. Par exemple, les premiers bureaux virtuels « trois dimensions » ont été publiés sous licences libres.

Il est nécessaire de prendre en compte différents cas de figures pouvant être liés à la création de logiciels libres. En effet, il se peut qu'une entreprise décide de **publier le code d'un programme en perte de vitesse** pour augmenter sa popularité. Ainsi de nouveaux programmeurs pourront **le faire évoluer** à leur tour. Comme exemple, on peut citer *Gecko* qui est un moteur d'affichage de pages Internet très populaire. Il est notamment présent dans *Mozilla Firefox*.

Il est possible qu'un étudiant, un chercheur, un ingénieur ou n'importe qui décide de publier un logiciel qu'il a développé sous licence libre.

Ainsi de nouveaux logiciels libres apparaissent sur le marché. Certaines entreprises exigent également que les programmes qu'elles vont utiliser soient publiés sous licences libres mais elles payeront malgré tout le développement. Ce procédé est très peu répandu, pourtant c'est un bon moyen pour se faire de la publicité. Il se peut que plusieurs personnes d'une même profession se réunissent et décident de financer la création d'un logiciel bien spécifique à leurs besoins et par la suite de le distribuer sous licence libre.

Schéma 1 : le mode de création de logiciels libres



VI. Base des modèles économiques du logiciel libre :

Comme son nom l'indique les logiciels libres garantissent des **libertés**. L'**utilisation** est **libre** ainsi que la **diffusion** permettant l'apparition de différents marchés étudiés plus loin. Ces différentes libertés ne peuvent pas s'appliquer à un modèle traditionnel d'affaire. En outre, il faut prendre en compte des paramètres que les logiciels propriétaires ne possèdent pas :

- Les licences libres étant plus souples que les licences propriétaires, elles permettent donc **un meilleur accès au marché** pour les nouveaux arrivants. Leurs limites sont moindres ce qui laisse une grande place à l'imagination de chaque utilisateur.
- La libre utilisation et la modification par n'importe qui assurent aux logiciels libres la possibilité de répondre entièrement aux besoins de chacun.

A noter encore que ces modèles économiques obligent **une compétition entre les différents acteurs** du logiciel libre ce qui pousse les développeurs à mettre sur le marché **des solutions plus performantes** et **de meilleure qualité** dans le but de surpasser ses concurrents. Ainsi il règne une légère compétitivité entre certaines distributions *GNU/Linux*, chacune ayant ses propres caractéristiques. On peut constater qu'entre les distributions destinées au grand public comme *Ubuntu*, *Fedora* et *Mandriva Linux* c'est réellement la course à l'innovation, bien que les comparer est difficile tant chacune à ses qualités et ses défauts.



Illustration 6:
Logo de Fedora



Illustration 7:
Logo de Mandriva Linux

Grâce à l'expansion d'Internet, la publication de logiciels libres est vraiment facilitée, n'importe quel utilisateur a accès à une technologie évolutive et cela à bas prix. Le Web permet une très **large diffusion du logiciel au niveau mondial**. De plus, le concept garanti par le libre est né avec l'ère de l'informatique, car l'échange de données ne coûte rien. Le fait d'avoir un code source ouvert permet **une diffusion dans le temps** du logiciel et ainsi une certaine stabilité.

Si une entreprise développant un logiciel libre fait faillite, quelqu'un peut décider de reprendre le logiciel et de continuer son développement sans devoir acheter des droits et des licences. Tandis que si le logiciel est propriétaire, les utilisateurs ne pourront plus avoir de nouvelles versions et de mises à jour. Ils seront obligés de chercher un logiciel similaire.

Le profit envisageable pour l'entreprise ne se situe plus à la vente du projet (comparable à une rente), vu que l'entreprise créatrice essaie de le distribuer le plus possible. L'utilisateur n'achète pas (excepté dans certains cas) le logiciel de base, mais achète des **modules complémentaires** ou paye pour avoir une **formation** sur le logiciel ou encore pour avoir de l'**aide**. Le modèle ne se base sur les rentes générées par la vente du logiciel, mais bien sur **la création de valeur ajoutée au produit de base**.

Certains éditeurs ont remarqués que le marché du logiciel libre leur offrent de meilleures possibilités économiques, ils ont décidé de **changer le type de leurs licences** pour du libre. Ainsi certains logiciels qui étaient publiés sous licence propriétaire deviennent des logiciels libres par la **publication de leurs sources**. Ces entreprises ont dû changer profondément leur modèle économique. Comme exemple on peut citer la publication du code source de *Java* par *Sun Microsystems* dès 2006. Ce changement historique a permis à de nombreux développeurs d'intégrer cette technologie dans leurs programmes.

Le rôle de la **sécurité** dans le monde des logiciels libres n'est pas négligeable (70% des serveurs web utilisent des logiciels libres). En cas d'attaque, le développement d'un correctif est très rapide, grâce au code source disponible. Beaucoup de personnes peuvent se pencher simultanément sur le problème et la participation communautaire prend tout son sens.

Les systèmes libres garantissent également une meilleure sécurité à travers les **droits limités** accordés aux utilisateurs, seul les administrateurs auront un accès complet. Ainsi une erreur de manipulation est évitable.

Pour rappel l'essor du logiciel libre est un fait plutôt récent. Il est étroitement lié au fort développement technologique de ces dernières années. Il dépend de différents facteurs, eux aussi, récents :

- l'**expansion d'Internet**, déclencheur d'un si fort accroissement (communications facilitées)
- l'**amélioration des systèmes informatiques**
- le **besoin de logiciels** sûrs, fiables et stables

VII. L'apparition du marché des logiciels libres et ses acteurs :

La mise en ligne du noyau Linux en 1991 est l'élément déclencheur qui a permis la mise en place d'un marché. Les premiers acteurs de ces marchés sont évidemment les éditeurs de distributions GNU/Linux (*Red Hat, Debian, SuSE, ...*). Pour ces entreprises le marché se situe non pas sur la vente de distributions, mais bien sur l'achat de services ou de logiciels supplémentaires. Ces acteurs ont permis l'apparition de nouvelles firmes, certaines fois concurrentes, qui publient elles aussi des logiciels sous licences libres ou d'autres firmes qui proposent des services.

Pour information, *Red Hat* distributeur de *Red Hat Entreprise* et qui est à la tête du projet *Fedora* (distribution GNU/Linux) a, aujourd'hui, un chiffre d'affaire de 650 millions de dollars. Son modèle repose sur la vente d'abonnements d'assistance et sur la formation des clients. Cette société emploie 2800 personnes et est même cotée en bourse.



Illustration 8:

Linus Torvalds :
Créateur du noyau
Linux

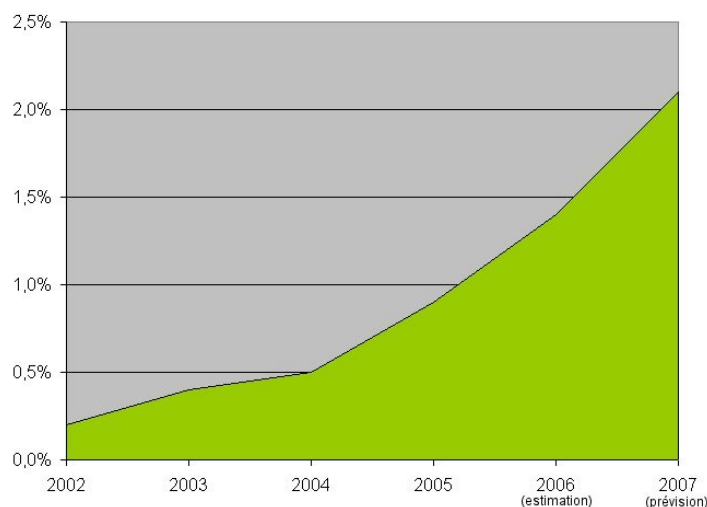


Illustration 9: Part du marché occupée par les logiciels libres. Source : Wikipédia

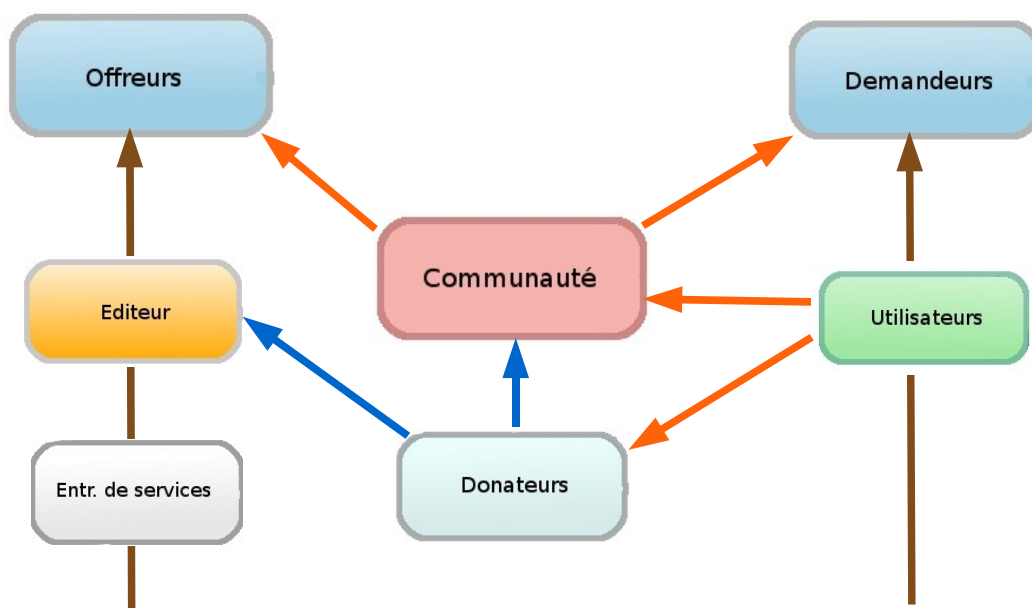
Comme sur chaque marché, plusieurs type d'acteurs se rencontrent, on peut les classer en différents groupes :

- les **utilisateurs**, le plus grand des groupes, réunit des entreprises, des privés et certaines institutions étatiques. Les utilisateurs sont comparables aux acheteurs sur un marché traditionnel. Ce sont donc les **demandeurs**.
- Les **éditeurs**, ils offrent les logiciels au public. Ils ont une équipe de programmeurs qui travaillent pour eux. Ils sont donc comparables aux **offreurs**.
- Les **entreprises commercialisant des services informatiques**, cette catégorie rassemble toutes les sociétés qui vendent des logiciels libres ou qui s'occupent de maintenance. Ce sont des **prestataires externes**.

Ci-dessus, ce sont les acteurs traditionnels du marché des logiciels. Ceux qui suivent appartiennent **exclusivement** au domaine du logiciel libre :

- Les **communautés** peuvent être placés du côté des demandeurs ou des offreurs. Celles-ci peuvent très bien demander à l'éditeur de rajouter une fonctionnalité aux programmes. Elles peuvent aussi la rajouter elle-même et redistribuer cette nouvelle version ou elle sera intégrée à la prochaine version de l'éditeur.
- Les **donateurs**, font des dons aux éditeurs ou aux communautés. Ce sont en principe des utilisateurs de la communauté

Schéma 2 : les différents acteurs du marché des logiciels libres



Légende :

- Appartenance
- Catégories
- Dons

VIII. Les modèles économiques :

Bien que chaque logiciel ait son propre modèle, ils ont été réunis ci-dessous d'après leurs caractéristiques communes.

1. *Le modèle de services :*

a) Présentation :

Les services que proposent ces entreprises sont très variés :

- le **conseil** : aide à la migration et recommandations afin de répondre au plus près aux attentes des utilisateurs
- l'**installation** : déploiement du logiciel

- la **formation** : l'utilisateur est formé sur la manière d'employer le logiciel de façon optimale et également d'écartier les craintes des nouveaux utilisateurs familiarisés depuis de longues années à la suite de logiciel propriétaire *Microsoft*.
- La **surveillance** : contrôle de compatibilité et détection de bugs lors de l'installation
- L'**assistance technique** : dépannage et documentation d'utilisation
- Les **tests** et la **garantie** : les logiciels sont pré-testés dans les conditions de l'entreprise, afin de garantir un bon fonctionnement
- Les **systèmes de sauvegarde** : conseils et développements d'outils de sauvegardes automatiques

b) Analyse :

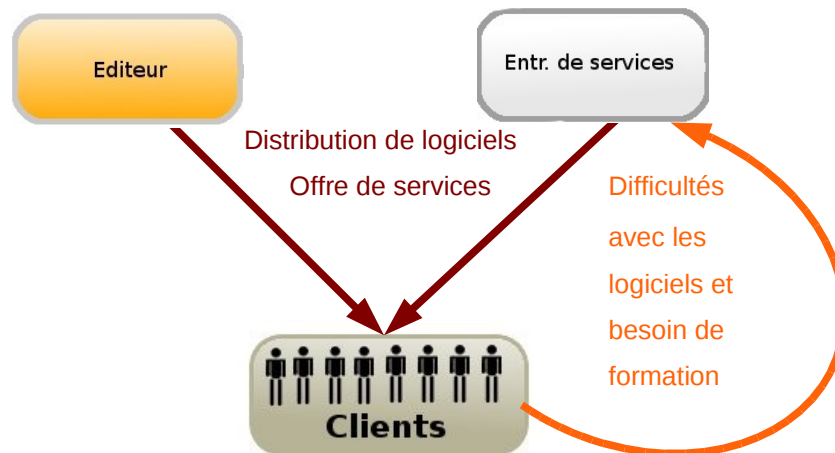
Deux types de modèles se différencient, les deux basés sur la commercialisation de services annexes :

1) Modèle de services simples :

Les éditeurs développent un logiciel et le vendent ou le distribuent gratuitement. Les clients ayant besoin de **formation** ou ayant des **problèmes** peuvent recourir à une entreprise proposant des services annexes sur des logiciels libres en général et non pas sur un en particulier. C'est, en principe, une société indépendante qui n'appartient pas à un éditeur.

Une application existante est le **modèle FAH** (Fournisseur d'Applications Hébergées) : l'entreprise fournit un service d'applications à ses clients à travers Internet. Ce modèle correspond à la demande des utilisateurs d'avoir accès aux services partout et rapidement. L'avantage est qu'il offre des applications complètes répondant aux besoins précis des utilisateurs qui s'abonnent ou qui louent le service.

Schéma 3 : Modèle de services simples

**Les facteurs évolutifs :**

La **demande régionale** est primordiale afin qu'une entreprise puisse proposer des services autour d'un logiciel qu'elle n'a pas créé, dans le but d'augmenter ses clients potentiels, elle peut proposer ses services sur de nouveaux logiciels.

Toutefois, si elle propose **un service similaire à celui offert par l'entreprise créatrice**, le client risque d'avoir **un attrait logique vers le développeur**. En effet, le support offert par ce dernier semble plus pertinent aux yeux des clients.

L'entreprise doit continuellement **innover** si elle souhaite rester concurrentielle. La **veille technologique** lui permet de proposer un service à valeur ajoutée par rapport au développeur et à ses concurrents.

Les références sont fondamentales dans la renommée de l'entreprise. Une entreprise peut avoir un grand panel de services si elle évolue sur un marché à faible concurrence.

Les **besoins des clients** et leur **sensibilité au prix** sont une part que les entreprises ne doivent pas négliger. Si le coût n'est pas déterminant, le prestataire va se concentrer exclusivement sur les **services importants**, ce qui **fidélisera** les clients. Dans le cas contraire, le prestataire proposera une plus **grande quantité** de service à un **maximum** de clients.

En conclusion on distingue deux assises opposées :

L'une se basant sur la **spécialisation de l'entreprise** et l'autre sur **l'extension des services proposés**.

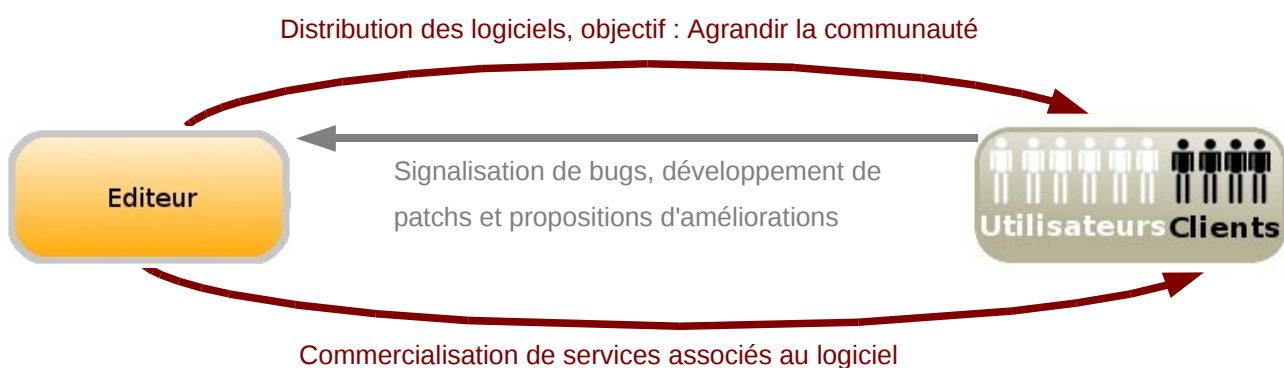
2) Modèle de valorisation indirecte :

Les éditeurs développent un logiciel, le vendent ou le distribuent gratuitement dans le but d'agrandir la communauté d'utilisateurs. Par contre, ils **commercialisent des services** tel que la formation, l'assistance, l'installation, la sécurité. Ces services sont **facultatifs** car le logiciel n'en requiert pas spécialement un en particulier.

La **communauté d'utilisateurs**, dans le but d'avoir un logiciel en constante évolution, est invitée à participer activement à l'amélioration du produit afin de lui faire bénéficier de :

- **rappports de bugs**
- **propositions d'améliorations**
- **développements de patches ou d'applications additionnelles**

Schéma 4 : Modèle de valorisation indirecte



Les facteurs évolutifs :

Sur le marché du logiciel libre il existe **un risque de forking**, lié à la concurrence. Si une entreprise évolue dans un marché non-concurrentiel, le risque de voir son code utilisé par une autre société sera plus élevé. A l'inverse sur un marché concurrentiel le code source a moins de chance d'être utilisé pour créer un logiciel similaire.

Si le programme a atteint une **maturité** et qu'il a fait ses preuves, le risque de *forking* est plus faible. Car les concurrents seront plus attirés par le **développement de modules complémentaires**, au lieu de redistribuer l'ensemble du logiciel.

En fonction d'une analyse concurrentielle, l'entreprise choisira soit de proposer **une multiplicité de services** soit de **resserrer ses activités** autour de quelques spécialités.

Un logiciel trop parfait ou trop élaboré ne laisse que peu de place aux services complémentaires. À l'inverse un produit immature risque de provoquer une crainte liée au coût de développement voire un rejet de la part des utilisateurs et des développeurs.

En conclusion, le succès de ce modèle se base sur deux leviers :

- **augmentation du marché** en élargissant la gamme de produits (développement de logiciels)
- **proposer des services** dans le but d'augmenter son capital

c) Exemple d'entreprise proposant un modèle de service :

L'entreprise *CRIL* basée à Cernier (NE) développe des systèmes embarqués et commercialise des services liés aux logiciels libres, comme par exemple :

- conseils
- développement
- formation
- sécurité
- gestion de documents
- systèmes de liaison

Le choix de l'entrepreneur de travailler dans les logiciels libres résulte de plusieurs motivations : l'accès au code source, la stabilité de fonctionnement par rapport aux homologues propriétaires ainsi que les aspects philosophiques et politiques.

Les coûts liés aux licences propriétaires sont alors reportés aux services tels que la formation et la maintenance du système des solutions libres. Selon l'entrepreneur, entre 15-20% du chiffre d'affaires est généré par la formation des clients.

La clientèle est majoritairement constituée de PME sur un marché local qui impose une spécialisation et une innovation de la part de l'entreprise *CRIL*. Ces PME se font habituellement conseiller des solutions propriétaires et il n'est donc pas toujours aisé pour cette société neuchâteloise de convaincre des réels avantages d'une infrastructure libre.

2. *Modèle de distribution à valeur ajoutée :*

Ce modèle est basé sur la création de deux versions du programme, la première qui sera distribuée gratuitement aux utilisateurs et la seconde qui est commercialisée.

a) **Présentation :**

L'entreprise rassemble des logiciels (distribution) dans une version standard. Cette version est distribuée en deux vecteurs bien distincts :

1. Distribution gratuite sans support
2. Distribution payante avec des services liés sous forme d'abonnements et pas sous forme de licences

b) **Analyse :**

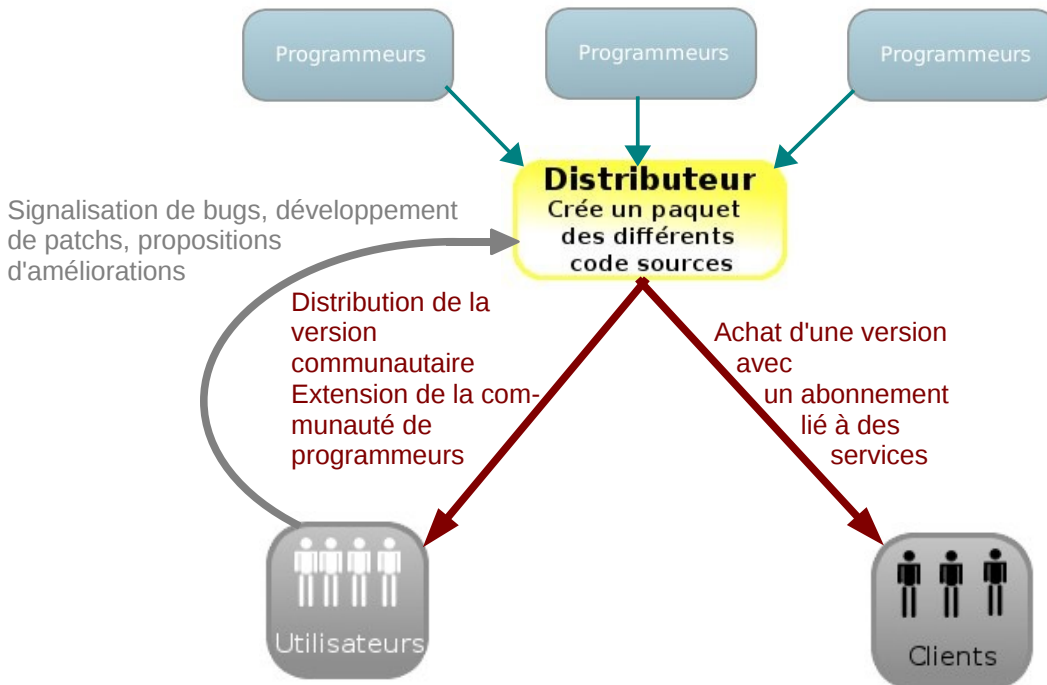
Le modèle est formé de différents acteurs qui interagissent entre eux :

- Les **programmeurs** créent différents logiciels libres, ils travaillent de manière indépendante. Leurs programmes proposés sur Internet peuvent être téléchargés et utilisés indépendamment les uns des autres.
- Le **distributeur** réunit plusieurs logiciels ensembles pour former un paquetage. Il arrive qu'il participe au développement et à l'amélioration des programmes qu'il intègre.

Le mode de diffusion du produit se divise en deux secteurs :

- D'un côté une distribution gratuite destinée à la communauté d'utilisateurs dont le but premier est d'agrandir l'ensemble des programmeurs.
- De l'autre côté une distribution payante et légèrement améliorée proposant un abonnement aux services de l'entreprise distributrice.
- La **communauté d'utilisateurs** est précieuse pour le distributeur car elle permet d'avoir un retour des bugs, le développement de patches ainsi que des propositions d'améliorations.

Schéma 5 : Modèle de distribution à valeur ajoutée



c) La valeur client :

Ce modèle est axé autour de la satisfaction du client, selon trois piliers :

1. Gain de temps :

Du fait que l'entreprise propose un packaging de logiciels, le client n'a pas à réinstaller tous les programmes ce qui est un **gain de temps** considérable.

2. La réduction des risques et garanties :

- Le packaging est certifié et testé par l'entreprise distributrice, ce qui implique que le client n'a pas à mettre les mains dans le code.
- Indemnisation en cas de problèmes et de défauts.

Assistance comprise dans l'abonnement

3. Mises à jour et patches :

Les mises à jour et les **patches** sont disponibles automatiquement durant la période d'abonnement. Pour le client, cela implique qu'il n'est pas obligé d'avoir une maintenance informatique à l'intérieur de sa société.

d) Exemple d'entreprise proposant un modèle de distribution à valeur ajoutée :

Red Hat est une entreprise développant principalement des distributions Linux, son modèle est basé sur les services et principalement l'assistance client qui représente plus de la moitié du chiffre d'affaires. Actuellement (décembre 2009) la capitalisation de Red Hat sur le *NYSE* (*New York Stock Exchange*, bourse de New York) s'élève à plus de 6 milliards de dollars et son chiffre d'affaires dépasse les 650 millions de dollars. Cette entreprise emploie plus de 2800 personnes à travers le monde, principalement aux Etats-Unis. L'offre de Red Hat se divise en plusieurs secteurs bien distincts :

- Une version communautaire destinée au grand public : **Fedora**
- Une version entreprise, testée et garantie fonctionnelle couplée à un abonnement (*Red Hat Enterprise Linux*)

La version entreprise propose plusieurs offres allant du basique au premium avec des services supplémentaires.

En conclusion, ce modèle bénéficie de plusieurs avantages :

- **Gain de temps pour le client** (installation regroupée, stabilité et solution « clé en main » testée)
- **Retour des expériences** des utilisateurs sur le produit lié à la distribution gratuite vers la communauté.

En revanche on remarque deux inconvénients :

- Un logiciel intégré dans la distribution peut **disparaître** et rendre la distribution **obsolète**
- L'abonnement aux services de maintenance peut être réalisé par **une autre entreprise** (modèle 1)

3. *Modèle de licences double :*

Il est basé sur deux licences, une **libre** et une **propriétaire**. Il s'appuie sur deux piliers forts qui sont la **communauté** et la **finition du produit**.

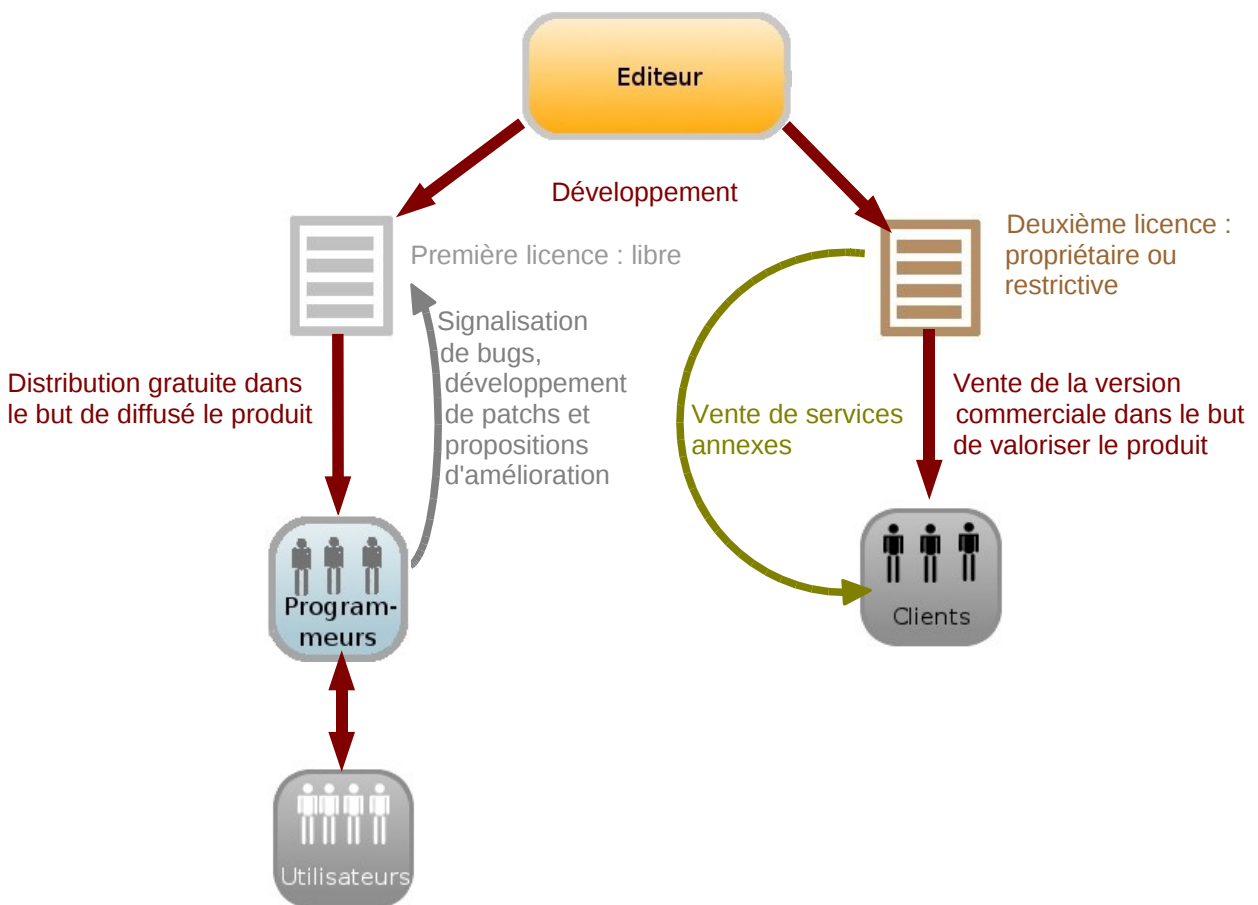
a) **Présentation :**

Ce modèle se fonde sur une **diffusion** du **code source limité** du programme, afin de **sensibiliser** la communauté sur la qualité et la finition du **produit propriétaire non-limité** dans l'utilisation.

b) **Analyse des deux licences :**

1. **Libre** : Diffusion de la version limitée du code source (utilisation, étude, modification et rediffusion) bénéficiant des **avantages** d'une communauté d'utilisateurs (signalisation de bugs, développement de patches)
2. La deuxième licence est similaire à la **vente de logiciel propriétaire** dans laquelle l'éditeur bénéficie immédiatement de revenu sur la vente, l'utilisation et des services liés.

Schéma 6 : Modèle de licences double



c) Deux piliers forts :

Deux piliers sur lesquels va s'appuyer toute la politique de l'entreprise éditrice :

1. Le premier pilier est **les limitations de fonctionnalités** par rapport à la version payante :
 - La **communauté** joue un rôle décisif. Plus la communauté est grande, plus la version libre devra contenir des fonctionnalités par rapport à la version commerciale.
 - L'**efficacité du produit** : la version libre doit donner un aperçu exhaustif des fonctionnalités payantes.
 - Le **degré de finition** des produits qui seront commercialisés.

2. La **notoriété du produit** : La satisfaction du client véhiculée dans la communauté envers ce produit fait bénéficier aussi bien l'entreprise éditrice et le logiciel d'une forte notoriété.

d) Exemple d'entreprise proposant un modèle de licence double :

SugarCRM (USA) évolue sur un marché international dans le domaine des relations clients, elle développe des outils de management et de gestion. Elle bénéficie d'une grande notoriété de part un portefeuille clients important : la compagnie d'assurance BDO, Coca-Cola et l'Etat américain de l'Oregon. *SugarCRM* a affiché en 2006, un chiffre d'affaires de 6,60 millions de dollars et emploie plus de 150 personnes.

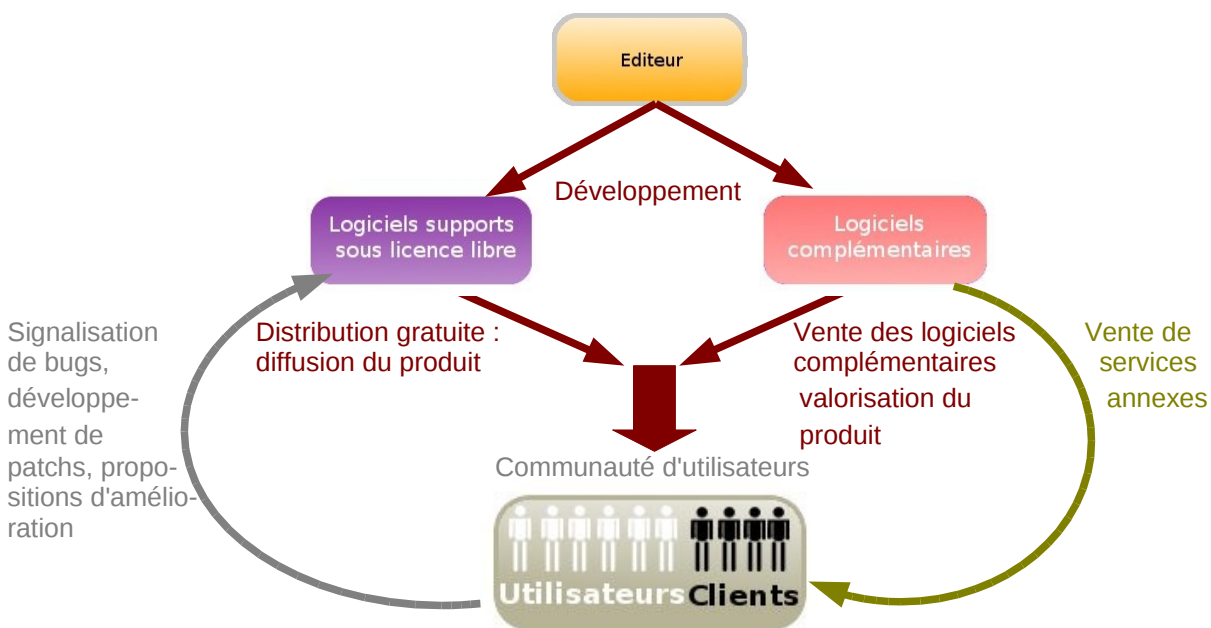
Son offre est divisée en deux versions :

- Une **version libre** (gratuite). Elle contient 85% des fonctionnalités de la version payante
- Plusieurs **versions commerciales** avec chacune des améliorations et accompagnées d'un support technique.

e) Variante : la commercialisation d'un produit complémentaire

Dans cette variante, l'éditeur de la version libre propose l'**ajout de fonctionnalités par module payant**.

Schéma 7 : Commercialisation d'un produit complémentaire



En conclusion, l'éditeur aura intérêt à être **connu sur le marché**, afin de développer une version libre **éloignée** de la version commerciale. Tandis qu'une entreprise avec une faible notoriété devra proposer une version standard avec **beaucoup de fonctionnalités**. Ce modèle est donc conseillé aux entreprises ayant déjà **une bonne assise** sur le marché.

4. Le modèle de mutualisation :

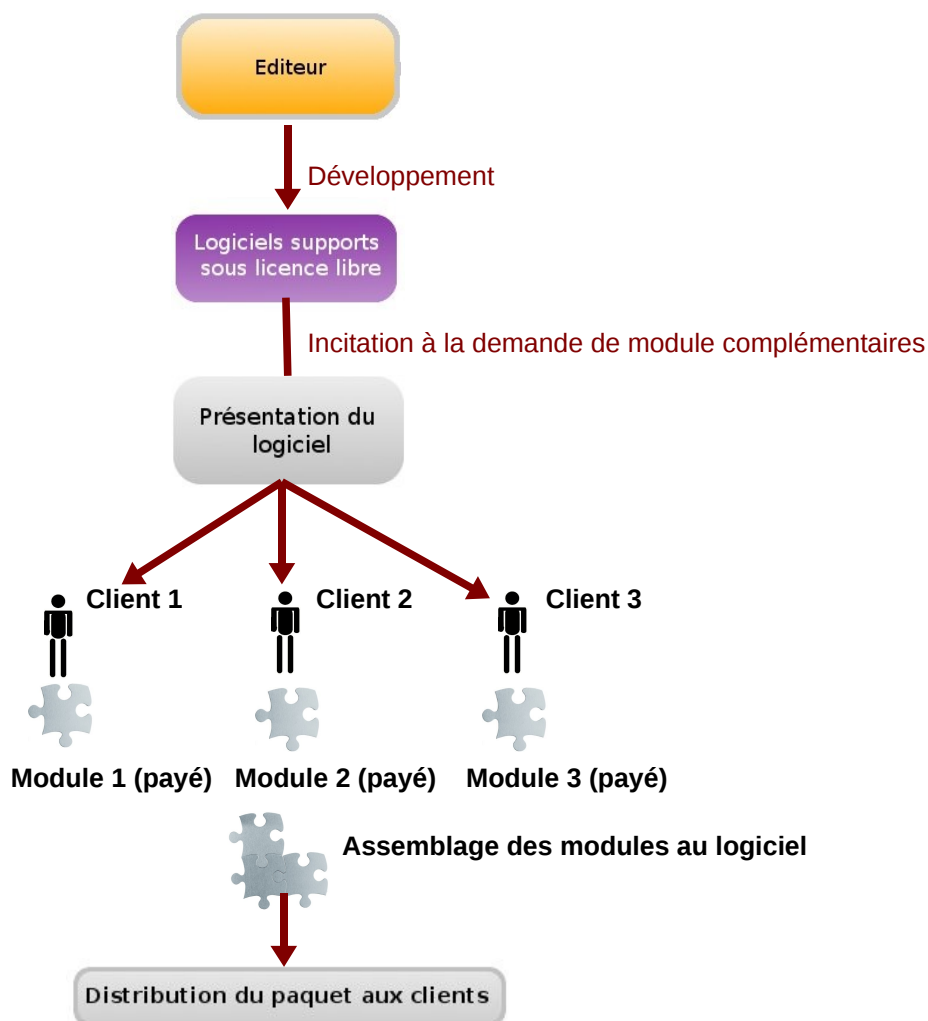
a) Présentation :

Ce modèle est basé sur la **répartition des frais de développement** appliqués à des **modules complémentaires**.

b) Analyse :

L'assise de ce modèle consiste à développer **une version basique** du logiciel et de promouvoir cette version à des clients potentiels en leur proposant **le financement du développement de modules complémentaires**. Les modules financés sont développés par l'éditeur et tous distribués dans une version finale (paquetage). Ce modèle n'est pas propre au domaine du logiciel libre.

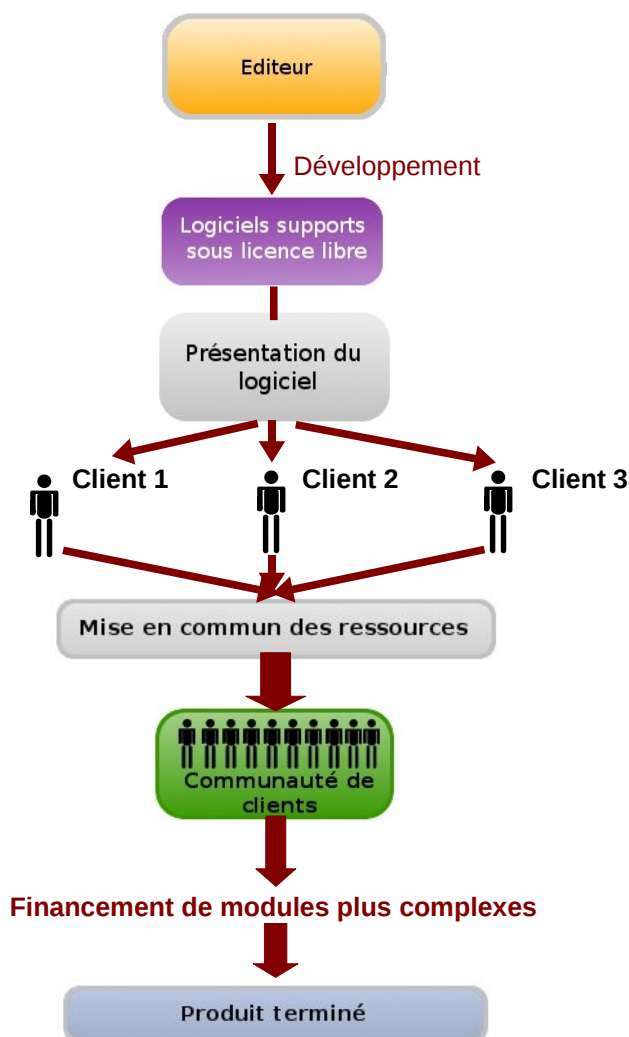
Schéma 8 : Modèle de mutualisation



c) Création d'une communauté clients :

Ce modèle amène souvent la création d'une **communauté de clients**. Cette communauté permet à l'entreprise de retirer des avantages intéressants surtout lors du développement de modules coûteux. En effet, les membres de cette communauté peuvent **mettre en commun** leurs ressources financières pour permettre la création du module. La communauté de clients peut après un certain temps devenir un « club d'investisseurs » qui passe régulièrement commande pour la création de nouveaux modules.

Schéma 9 : Modèle de mutualisation avec la création d'une communauté de clients



L'entreprise éditrice peut espérer dégager des revenus de manière variable :

- Par le **développement de modules complémentaires**
- Par l'**intégration de nouveaux modules** chez d'autres clients
- La **mutualisation des frais** générés par le développement d'un module conséquent
- Par la facturation de l'intégration de modules payés par le club d'investisseurs

Ce modèle exige que les méthodes de financement mises en place couvrent les coûts de développement du logiciel de base.

d) Exemple :

La société *OpenTrust* (F) spécialisée dans les logiciels de sécurité des informations. Créée en 2001, elle possède actuellement plusieurs centaines de clients internationaux. Parmi ses clients, on trouve plusieurs grandes entreprises. Ses domaines de compétences sont : chiffrer la messagerie pour un ministère, protéger les données d'une grande entreprise, crypter les numéros de cartes de crédit d'un grand acteur de la distribution, etc. Elle a dégagé un chiffre d'affaires de 6,7 millions d'euros en 2004.

Elle crée, tout d'abord, un module de base *ICP (Infrastructure à Clés Publiques)*, qu'elle présente aux clients potentiels. Une communauté dans laquelle, chaque client va financer le développement d'un nouveau module plus spécifique à ses besoins, est constituée. Les modules nouvellement créés sont paquetés et distribués à tous les membres de la communauté clients.

OpenTrust a également mis en place un club d'investisseurs.

e) Variante : Mutualisation des modules des développeurs

Un distributeur commande auprès de **multiples éditeurs** différents modules. Ces derniers sont ensuite assemblés par le distributeur dans un paquetage qui est ensuite distribué aux clients.

Pour conclure, les avantages du modèle sont :

- **bénéficiaire de l'intégration de modules non-financés**
- la **mutualisation des frais liés** à la création de modules complexes

En revanche, deux désavantages se distinguent :

- Les modules complémentaires doivent **financer le programme**
- **Coûts élevés** pour la création de nouveaux modules

5. *Autres modèles :*

Un autre modèle qui ne sera pas présenté dans ce document, consiste à avoir un **partenariat** avec une autre entreprise et **intégrer un module** (propriétaire ou non) dans un logiciel qui sera publié finalement sous licence libre. L'exemple que l'on peut citer n'est autre que le très célèbre navigateur Internet *Mozilla Firefox* qui a un partenariat avec *Google*. A chaque démarrage du navigateur, une page par défaut avec le moteur de recherche de *Google* s'affiche. Ce modèle a permis à la *Mozilla Corporation* (filiale qui s'occupe du développement et du marketing lié à *Mozilla Firefox*) de dégager un revenu combiné constitué à plus de 90% de ce partenariat.



Illustration 10: Logo de Mozilla Firefox

Un modèle unique a été développé par l'entreprise *Linbox FAS*, elle publie un logiciel sous licence propriétaire mais une fois que **le cumul des ventes est égal au frais** engendrés par la création du logiciel, la société libère son code source.

Il existe également un modèle, non-exclusif aux logiciels libres, qui est basé sur **la publicité**. Ce modèle touche principalement des logiciels destinés à une **très large diffusion**, il est très peu utilisé dans le domaine du libre. En effet, peu d'entreprises liées à ce dernier ont les fonds nécessaires pour mener une politique marketing suffisante.

On peut encore nommer le projet de **mécénat général** qui vise à rémunérer les artistes et créateurs de programmes à travers une « **donation** » au FAI (Fournisseur d'Accès à Internet), qui redistribua par la suite à ces derniers les fonds récoltés. Cela permettrait peut-être de résoudre les problèmes liés au téléchargement illégal.

IX. Synthèse :

Les modèles présentés ont tous trois points communs principaux : ils dépendent de l'**innovation des créateurs**, de la **communauté d'utilisateurs** (pour les signalisations de bugs et les propositions d'améliorations) ainsi que celle des **programmeurs** (pour le développement de patches) et surtout de l'**intensité de la concurrence** qui règne sur le marché.

Le modèle de services est plus spécifiquement axé sur **une offre de services annexes** au logiciel. Ces services sont très variés et peuvent très bien être fournis par une entreprise tierce ou par l'entreprise éditrice.

Le modèle de création de valeur ajoutée repose sur **deux versions** du programme, une distribuée gratuitement et l'autre commercialisée qui est très fréquemment liée à un ou plusieurs services.

Le troisième modèle est axé autour de **deux licences**. Une première licence garantit **les libertés de l'utilisateur** (liberté d'utiliser, d'étudier, de modifier et de redistribuer), tandis que la seconde est nettement plus **restrictive**. Cette dernière est souvent liée à un programme possédant plus de fonctionnalités et/ou services.

Le dernier modèle repose sur le développement d'une **version simplifiée** du programme qui sera par la suite complétée selon le souhait du client. L'ensemble des modules développés sont réunis en un paquet qui sera distribué à tous les clients (excepté le financement du module désiré).

Les autres modèles qui régissent le domaine des logiciels libres peuvent être un **partenariat** avec une entreprise tierce, la **libération** du code source après avoir remboursé les frais de développement, ou encore être basé sur la **publicité**.

X. Conclusion :

Les modèles présentés précédemment ont tous leurs particularités. Mais on peut constater que tous ces modèles ont plusieurs limites communes.

Il existe de nombreux **préjugés** autour du logiciel libre, comme quoi il ne rapporte pas d'argent aux éditeurs, qu'il n'y a aucune innovation et que le libre reprend les principes du communisme. Ces idées fausses ont été démantelées dans le présent travail.

Les entreprises commercialisant ce type de logiciel devraient faire passer **une autre vision** dans l'opinion public en développant une **meilleure politique marketing**.

La publicité est un manque majeur dans l'industrie du libre. Il n'existe aucune publicité destinée au grand public, afin de le sensibiliser à la question des libertés logicielles. Les entreprises se basent sur leur **notoriété** et sur le « **bouche-à-oreilles** » pour faire connaître leurs produits. Elles devraient également être plus présentes dans la **publicité traditionnelle** (journaux, affiches, spots télévisés, ...) afin de lutter contre les préjugés qui entourent les logiciels libres.

Le **grand panel** de logiciels libres proposés déroute fréquemment les nouveaux arrivants. De plus les différences entre le monde propriétaire et libre rendent le changement encore plus difficile. En augmentant la publicité, les arrivants sauront immédiatement à quoi correspond tel ou tel logiciel.

Les entreprises doivent faire face **au risque d'obsolescence** du programme.

L'**innovation** est le moteur des logiciels libres, elles doivent donc toujours présenter un logiciel répondant aux dernières avancées. Un logiciel bien maintenu et à jour peut être utilisé comme argument de vente et gage de qualité.

Une entreprise désireuses de publier des logiciels sous licences libres devra **se faire connaître** dans le milieu mais également chercher à **attirer de nouveaux clients** à travers le soutien à des associations d'utilisateurs de logiciels libres, des journées de présentations, de l'aide (financière ou matérielle) à des projets liés au domaine du libre. Elle peut également proposer une « **installation teste** » à une société, par exemple un temps d'essai pour qu'elle se rende compte des possibilités qu'offre telle ou telle solution libre.

De mon point de vue, les modèles présentés doivent encore être perfectionnés pour répondre à une demande toujours plus grande et variée d'utilisateurs. Le marché ne va pas cesser de s'élargir et de nouveaux modèles vont faire leur apparition.

XI. Annexe :

1. Déclaration :

Je déclare par la présente que j'ai réalisé ce travail de manière autonome et que je n'ai utilisé aucun autre moyen que ceux indiqués dans le texte. Tous les passages inspirés ou cités d'autres auteur-es sont dûment mentionnés comme tels. Je suis conscient que de fausses déclarations peuvent conduire le Lycée cantonal à déclarer le travail non recevable et m'exclure de ce fait à la session d'examens à laquelle je suis inscrit.

Ce travail reflète mes opinions et n'engage que moi-même, non pas le professeur responsable de mon travail ou l'expert qui m'a accompagné dans cette recherche.

Porrentruy, le 12 février 2010

David Caillet

2. Bibliographie :

a) Livre :

- F. Elie, *Economie du logiciel libre*, Paris, Eyrolles, 2009

b) Entretiens :

- Au sujet des modèles économiques et de l'expérience personnelle de M. Marc Schaefer, entrepreneur, le 12 août 2009
- A propos des modèles économiques et des logiciels libres avec Richard Stallman, créateur du principe de logiciel libre, le 11 février 2010

c) Sitographie :

- <http://www.jcheritier.net/memoire/>, Mémoire de fin d'étude, Université de Strasbourg, 8 février 2009
- <http://www.april.org/articles/livres-blancs/modeles-economiques-logiciel-libre>, Livre blanc sur les modèles économiques du logiciel libre réalisé par l'APRIL, 8 février 2009
- <http://www.aful.org/professionnels/modeles-economiques-logiciels-libres>, Livre sur les modèles économiques du logiciel libre réalisé par l'AFUL, 8 février 2009
- <http://www.fabernovel.com>, Site d'une société qui a étudié les différents modèles économiques des logiciels libres, 9 février 2009
- <http://www.fsf.org> Site de la *Free Software Foundation*, 12 mars 2009
- <http://www.logiciellibre.net/article.php> Site traitant des logiciels libres et leurs implications économiques, 9 avril 2009
- http://fr.wikipedia.org/wiki/Licence_libre Article de Wikipédia sur les licences libres, 9 avril 2009
- http://fr.wikipedia.org/wiki/Logiciel_libre Article de Wikipédia sur les logiciels libres, 9 avril 2009
- <http://www.april.org/> Communauté oeuvrant pour la promotion du logiciel libre, 22 juin 2009
- <http://www.libroscope.org/Faut-il-un-modele-economique-du> Article au sujet du modèle des logiciel libre, 13 juillet 2010
- <http://www.cril.ch/> Site Internet de l'entreprise CRIL à Cernier, 12 août 2009
- <http://bydorian.com/un-modele-economique-du-logiciel-libre/> Article au sujet d'un modèle, 15 septembre 2009
- http://fr.wikipedia.org/wiki/Red_Hat Article Wikipédia sur Red Hat, 24 décembre 2009
- http://www.votreargent.fr/bourse/fiches-valeurs/actualite_dep.asp?id=119343 Article au sujet de Red Hat et sa relation boursière, 24 décembre 2009
- <http://forum.ubuntu-fr.org/viewtopic.php?pid=3235833> Post sur le forum Ubuntu au sujet du document, 11 janvier 2010
- <http://www.sugarcrm.com/> Site Internet de la société SugarCRM, 1 février 2010

- <http://www.opentrust.com/> Site Internet de la société OpenTrust, 5 février 2010
- <http://www.presence-pc.com/tests/open-source-developpement-23244/> Article expliquant l'implication des différentes licences, 6 février 2010
- <http://linuxfr.org/2005/04/09/18693.html> Un autre modèle envisageable, 7 février 2010
- http://www.dailymotion.com/video/x6pgjy_le-modele-economique-du-logiciel-li_news Vidéo de présentation des modèles économiques du logiciel libre, 7 février 2010
- http://www.vd.ch/fileadmin/user_upload/organisation/dinf/dsi/fichiers_pdf/OT_Seminaire_ABC_2009-09-17_SOLL.pdf Présentation au sujet des logiciels libres, 7 février 2010
- http://www.afjv.com/press0402/040211_anvar.htm Explication de l'émergence des logiciels libres, 8 février 2010