FABERNOVEL CONSULTING

Research paper – Septembre 2007

Modèles économiques des logiciels open source et logiciels libres : quelques points de repère

(cc) BY-NC-SA

faberNovel Consulting 2007
Document public

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique 3.0 disponible en ligne http://creativecommons.org/licenses/by-nc-sa/3.0/ ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA

(cc) BY-NC-SA

Ce qu'il faut retenir de ce document

- Le secteur du logiciel open source connaît une forte croissance qui devrait se prolonger dans les années à venir
- Les entreprises open source se sont structurées autour de quatre *business models* :
  - Le modèle de services
  - Le modèle de distribution à valeur ajoutée
  - Le modèle de licence double
  - Le modèle de mutualisation
- Ces *business models* sont rentables et viables sur la durée
- Au-delà de la diversité des business models, il existe plusieurs facteurs clés de succès communs à toutes les entreprises open source
- Plusieurs facteurs, comme l'intensification de la concurrence et une méfiance persistante vis-à-vis des solutions open source, pourraient conduire à une modification des stratégies à venir des entreprises

(cc) BY-NC-SA

Sommaire

- Contexte et objectifs de l'étude
- Les différentes typologies de *business models*
- Les facteurs clés de succès communs aux différents modèles
- Quelles stratégies à venir pour les logiciels open source ?

(cc) BY-NC-SA

Sommaire

- Contexte et objectifs de l'étude
- Les différentes typologies de *business models*
- Les facteurs clés de succès communs aux différents modèles
- Quelles stratégies à venir pour les logiciels open source ?

(cc) BY-NC-SA

A l'origine de l'open source : le mouvement du logiciel libre

- Le logiciel libre est apparu en 1985 avec la fondation de la *Free Sofware Foundation* (FSF) par Richard Stallman
- Selon la FSF, les logiciels libres doivent respecter quatre libertés :
  - Liberté d'exécuter le programme, pour tous les usages
  - Liberté d'étudier le fonctionnement du programme et donc liberté d'avoir accès au code source du logiciel
  - Liberté de redistribuer des copies et de les vendre
  - Liberté d'améliorer le programme et de publier ses améliorations
- La FSF attribue plusieurs licences, dont la plus répandue est la General Public License (GPL). En 2004, elle représentait 68,5%[1] des projets listés par SourceForge[2]
- Pour éviter la confusion libre/gratuit, l'*Open Source Initiative*, créée en 1998, a élaboré l'*Open Source Definition*

1) Sur un panel de 52.183 projets
2) Sourceforge est la plus grande plateforme internet de développement et de téléchargement des codes et applications open source

(cc) BY-NC-SA

Les licences open source répondent à dix critères

- La libre redistribution
- L'accès au code source
- Le droit de modifier le code source et de développer des applications dérivées
- Le respect de l'intégrité du code source de l'auteur : la licence peut exiger que les applications dérivées soient distribuées sous un autre nom, ou que la version originale préservée soit distribuée avec des patches
- L'interdiction de discriminer des personnes ou des groupes
- L'interdiction de discriminer des domaines d'applications
- L'universalité des droits attachés au programme. Ils doivent s'appliquer à tous ceux à qui il est redistribué sans obligation pour ces parties d'obtenir une licence supplémentaire (ceci permet d'éviter que des programmes bénéficiant initialement de la licence attribuée par l'*Open Source Definition* puissent ensuite être appropriés par des moyens indirects comme l'exigence d'une clause de dissimulation)
- La protection du programme et non pas du produit
- L'absence de prolifération* à d'autres produits intégrant un code source protégé
- La neutralité technologique. La licence ne peut discriminer aucune technologie ou style d'interface

* : On parle également très souvent de « contamination » de la licence
Source : Open Source Initiative

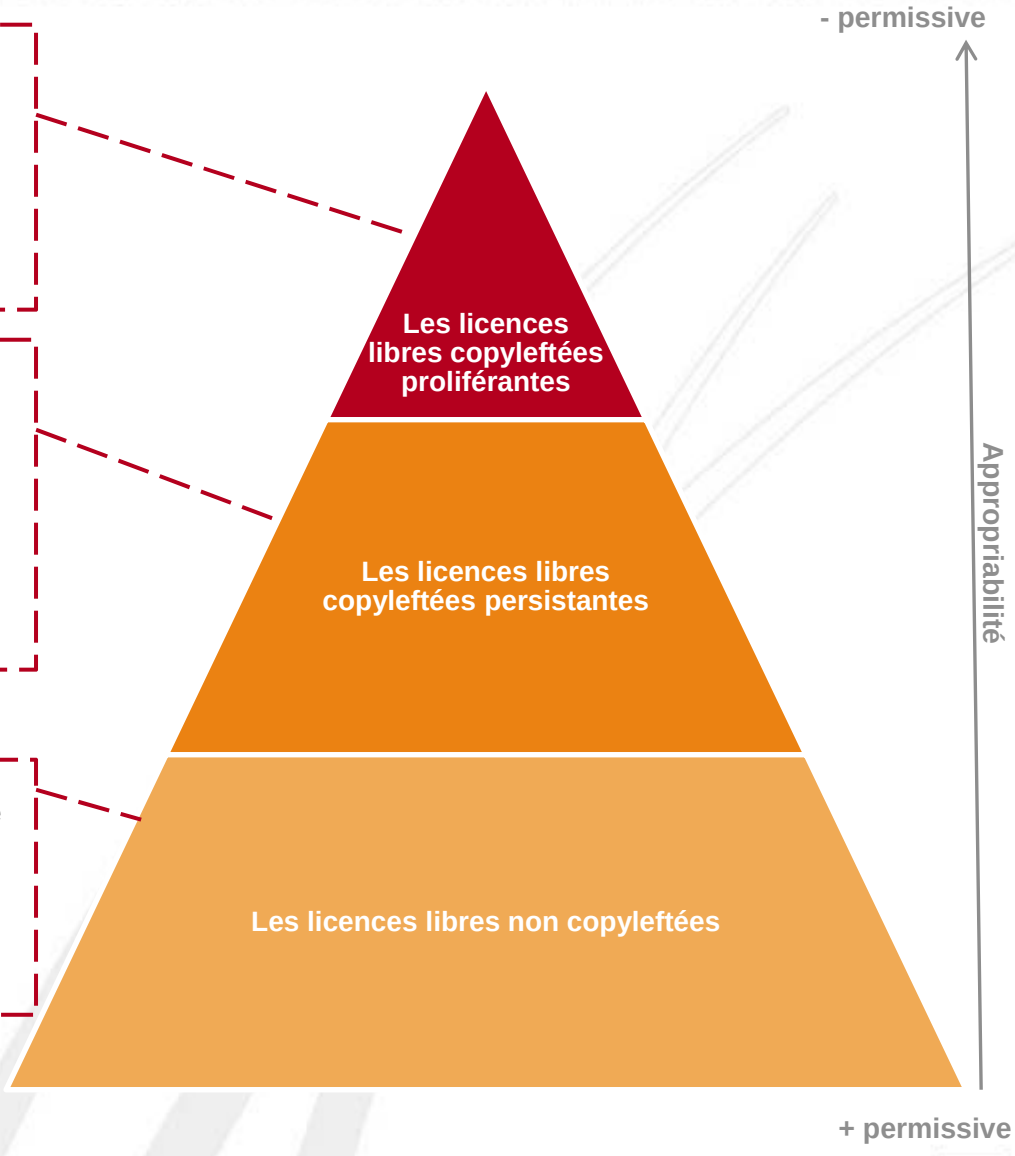
(cc) BY-NC-SA

Trois types de licences peuvent être identifiées en fonction de leur permissivité

- Elles imposent l'obligation de placer sous la même licence tout logiciel modifié et tout programme incluant ce logiciel dans un produit dérivé
- Exemples :
  - General Public License (GPL)
  - Mozilla Public License (MPL)

- Elles comportent une clause autorisant les utilisateurs à mélanger le logiciel à un logiciel propriétaire et à le placer sous licence propriétaire, à condition que le module libre utilisé reste sous licence libre
- Exemples :
  - Lesser General Public License (LGPL)
  - Berkeley Software Distribution (BSD)
  - License MIT

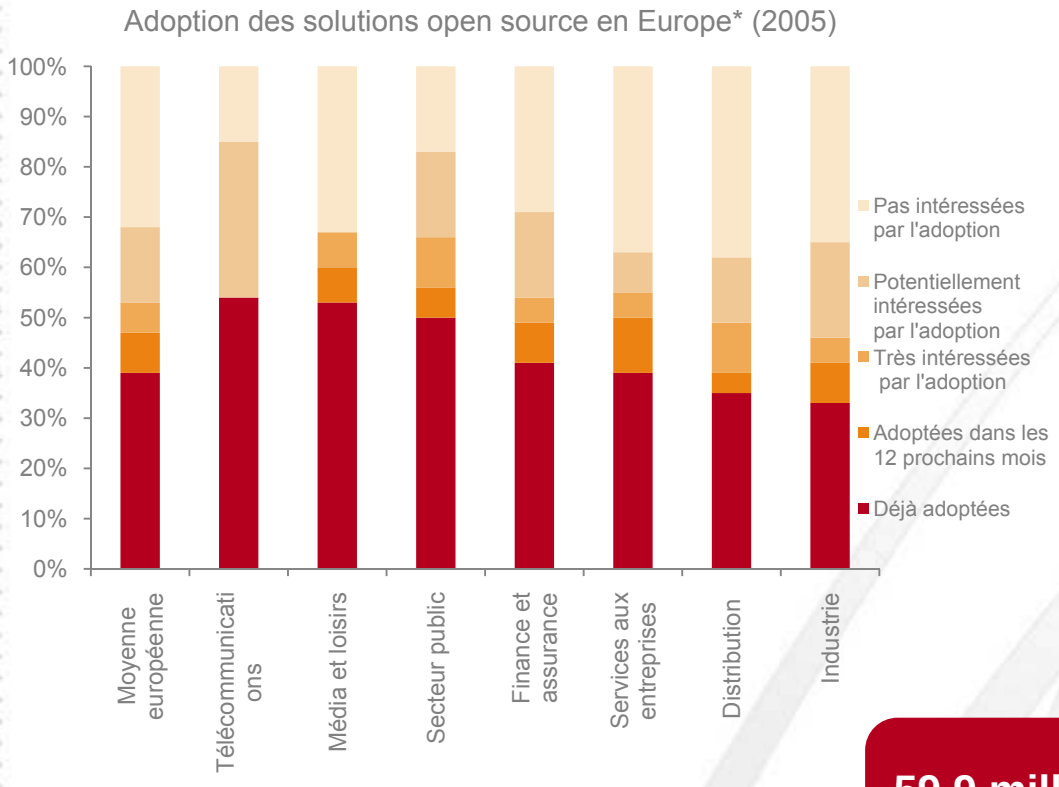
- N'importe qui peut s'approprier un code source sous cette licence
- Exemples :
  - Xfree86
  - X Consortium
  - License Apache

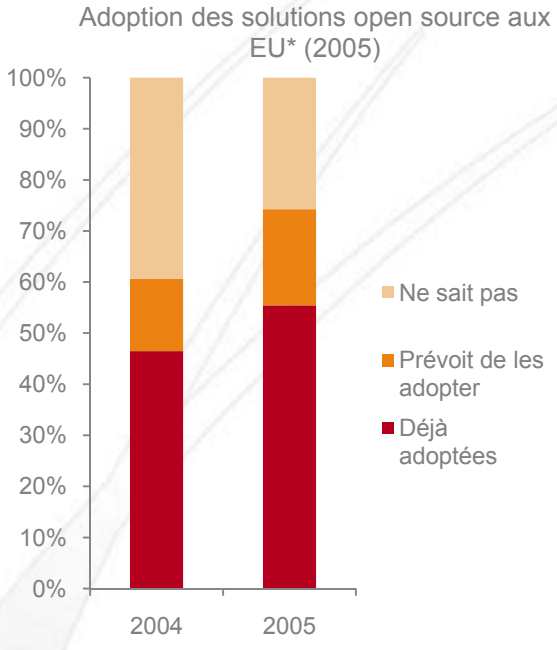


(cc) BY-NC-SA

La progression des logiciels open source devrait se maintenir dans les années à venir

- Les solutions open source conservent une marge de progression importante





59,9 millions de dollars :

Le résultat net de Red Hat en 2006

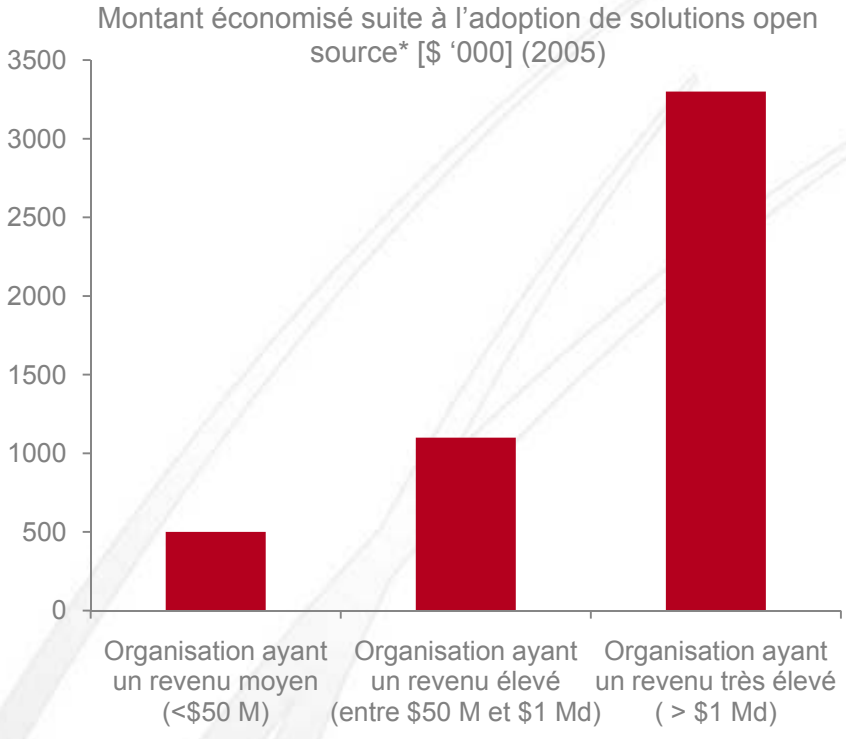
* : Sur un panel de 305 entreprises européennes et 104 entreprises américaines

Sources : Forrester, 2005, analyses faberNovel

(cc) BY-NC-SA

Les qualités des solutions open source sont aujourd'hui largement reconnues

- Une utilisation plus souple et plus sûre :
  - Possibilité d'adapter le produit aux besoins exacts à partir du code source
  - Contraintes de transférabilité des logiciels moindre
  - Mises à jour fréquentes
  - Facilité d'émission des patches
- Un produit de qualité pour un prix inférieur aux logiciels propriétaires
- Un capital de sympathie important et une adhésion aux valeurs véhiculées par le mouvement open source



De nombreux cadres restent pourtant méfiants en raison d'une méconnaissance des modèles économiques des entreprises open source

* : Sur un panel de 502 entités (entreprises, agences gouvernementales et autres organisations) américaines entre août et septembre 2005
Sources : Optaros 2005, analyses faberNovel

(cc) BY-NC-SA

L'objectif de cette étude est d'analyser les différents *business models* des entreprises open source

- Schématiquement, un *business model* est constitué de deux éléments :

Le modèle de revenus	La structure de coûts

• Création de valeur : définition de l'offre générant la valeur d'utilité perçue par le client (VUPC) la plus élevée
• Capture de la valeur créée à travers :
  ▪ La vente de droits (vente de brevets, de licences, voire de fichiers clients)
  ▪ La vente de produits
  ▪ La vente de services
	• Définition en fonction des catégories de coûts (matières premières, marketing, R&D, administratif) et de leur type (fixe ou variable)
	• Identification des compétences propres à l'entreprise lui offrant un avantage comparatif sur la concurrence
	• Détermination des moyens d'accès aux sources de financements

- Les *business models* de l'open source ont la particularité de différer essentiellement par leurs modèles de revenus. Par souci de clarté, nous présenterons une typologie centrée sur ces modèles

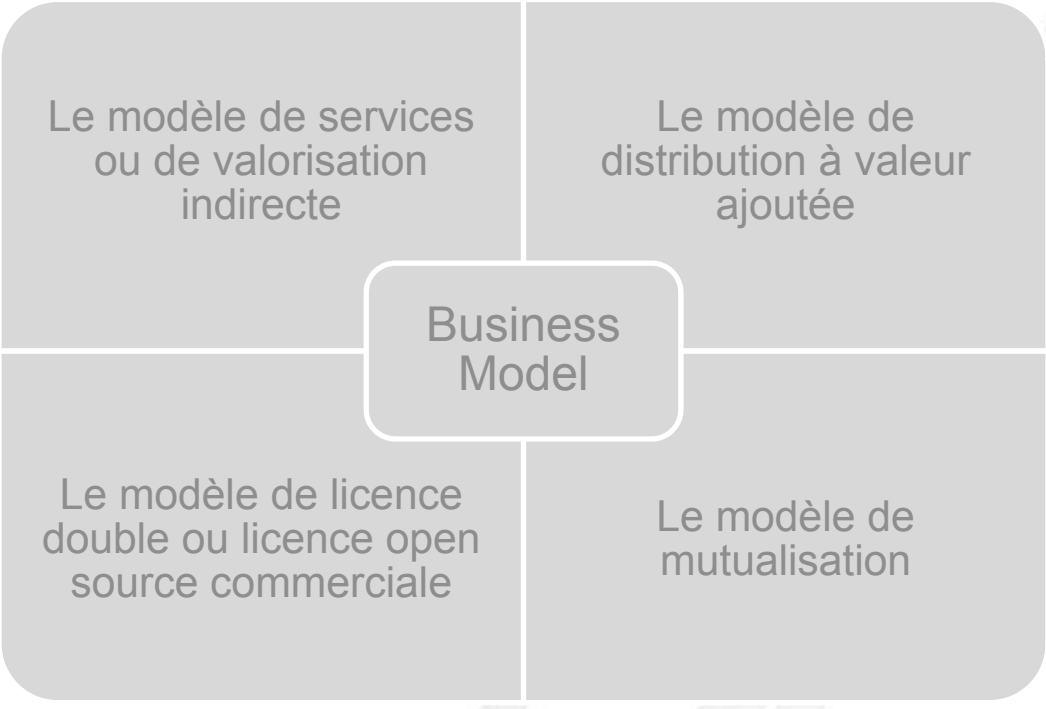
(cc) BY-NC-SA

Sommaire

- Contexte et objectifs de l'étude
- Les différentes typologies des *business models*
- Les facteurs clés de succès communs aux différents modèles
- Quelles stratégies à venir pour les logiciels open source ?

(cc) BY-NC-SA

Quatre *business models* peuvent être identifiés



Quel que soit le modèle adopté, toutes les entreprises proposent des services complémentaires à leurs produits pouvant représenter du quart à la moitié de leur chiffre d'affaires

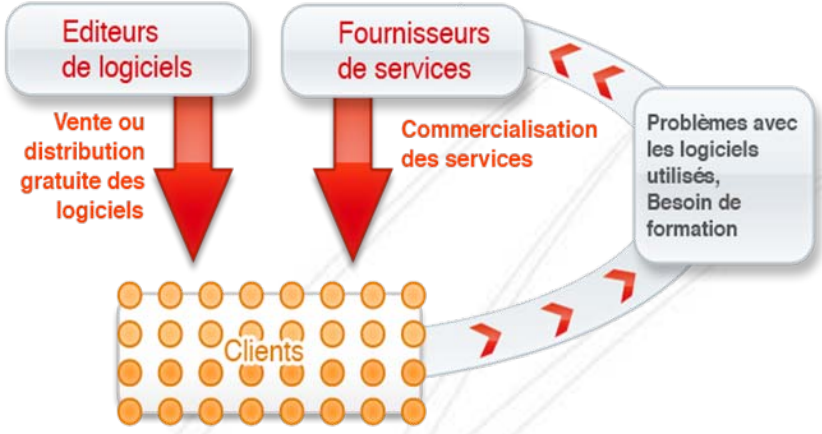
(cc) BY-NC-SA

Le modèle de services se présente sous deux formes

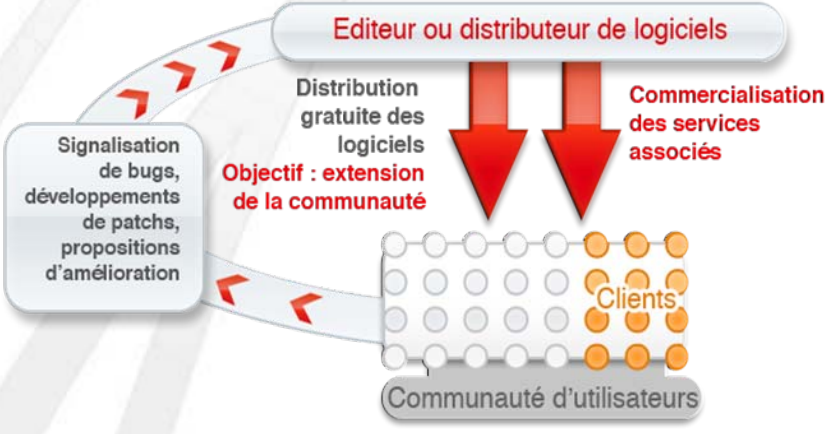
- **Modèle de services simples** : commercialisation de services sans lien avec un logiciel particulier

> **« Our job is to be the Switzerland of open source software components »**
> (M. Halsey, vice président des alliances et des ventes à l'international de Spikesource)

- Une variante de ce modèle consiste à fournir un service d'application sans lien direct avec les logiciels open source utilisés *via* le réseau internet en utilisant un protocole standard (modèle ASP)



- **Modèle de valorisation indirecte** : commercialisation de services associés à des logiciels développés ou packagés en interne



(cc) BY-NC-SA

Les services proposés sont de différentes natures

Surveillance

- Contrôle des algorithmes
- Détection de bugs
- Surveillance de problèmes de sécurité provenant d'autres logiciels open source

Formation

- Aide
- Publication de documents supports
- Création de patches en cas de problème

Tests et garantie

- Aide à l'intégration de logiciels testés
- Interopérabilité garantie

Assistance technique

(cc) BY-NC-SA

Le modèle de services simples repose sur deux leviers opposés

Leviers de croissance

Extension du nombre de services proposés

- Segments du marché dépendant du nombre d'entreprises commercialisant des solutions open source sans proposer de services complémentaires de qualité satisfaisante
- Capacité de l'entreprise à proposer des services sur un grand nombre de logiciels qu'elle n'a pas elle-même développés

Spécialisation des services proposés pour développer un avantage compétitif

- Degré d'intensité de la concurrence sur les services proposés : plus la concurrence est forte et plus l'entreprise open source a intérêt à développer des compétences particulières autour de quelques services
- Besoin des consommateurs et sensibilité au prix : si les clients potentiels ont des besoins spécifiques et sont peu sensibles au prix, l'entreprise open source aura intérêt à se spécialiser autour de quelques services très rentables et facilement valorisables

Facteurs d'évolution

(cc) BY-NC-SA

Le succès du modèle de valorisation indirecte repose sur deux leviers

Leviers de croissance

Augmentation de la taille du marché en favorisant une diffusion large des solutions

- Degré d'intensité de la concurrence sur les logiciels proposés, qui dépend du risque de « *forking*[1] »
- Choix de la licence :
  - Si les produits sont destinés à un usage direct, aucun autre logiciel ne sera développé à partir du code source développé par l'entreprise. Une licence de type copyleftée est donc adaptée car il n'y a pas de risque de prolifération
  - Si les produits sont plutôt des « modules », destinés à être insérés dans d'autres programmes, il est impératif pour l'entreprise d'utiliser une licence copyleftée persistante ou non copyleftée

Augmentation du taux de monétisation en proposant des services à un maximum d'utilisateurs

- Degré d'intensité de la concurrence sur les services complémentaires proposés
- Choix du degré de finition du produit :
  - Un produit trop élaboré nécessite peu de services complémentaires
  - Un produit trop éloigné du produit opérationnel risque de susciter le rejet de la part des utilisateurs et des développeurs

Facteurs d'évolution

1) : Utilisation du code source du logiciel (parfaitement possible puisque ce code est ouvert) pour créer un produit concurrent.

(cc) BY-NC-SA

Un exemple d'entreprise proposant un modèle de services : Spikesource

- La société Spikesource s'est spécialisée dans le test, la certification et l'intégration de logiciels open source du LAMP* et des différentes applications qui peuvent les utiliser. En 2005 elle affichait un chiffre d'affaire de 76.000 dollars, et a levé 21 millions de dollars pour poursuivre son développement

- L'entreprise propose deux offres :
  - *SpikeIgnite Platform* :
    - Un ensemble de logiciels et middlewares open source intégré et garanti
    - Des mises à jour développées à partir de la plateforme permettant de recenser 25.000 patches et mises à jour par semaine des différents logiciels pris en charge par Spikesource
    - Une offre de départ se situant aux alentours de 1.890$ par an.
  - *Spikenet* : un système d'assistance technique et de maintenance dont le prix de base est de 7.500$ par an

* : LAMP désigne un pack Linux, Apache, MySQL et PHP

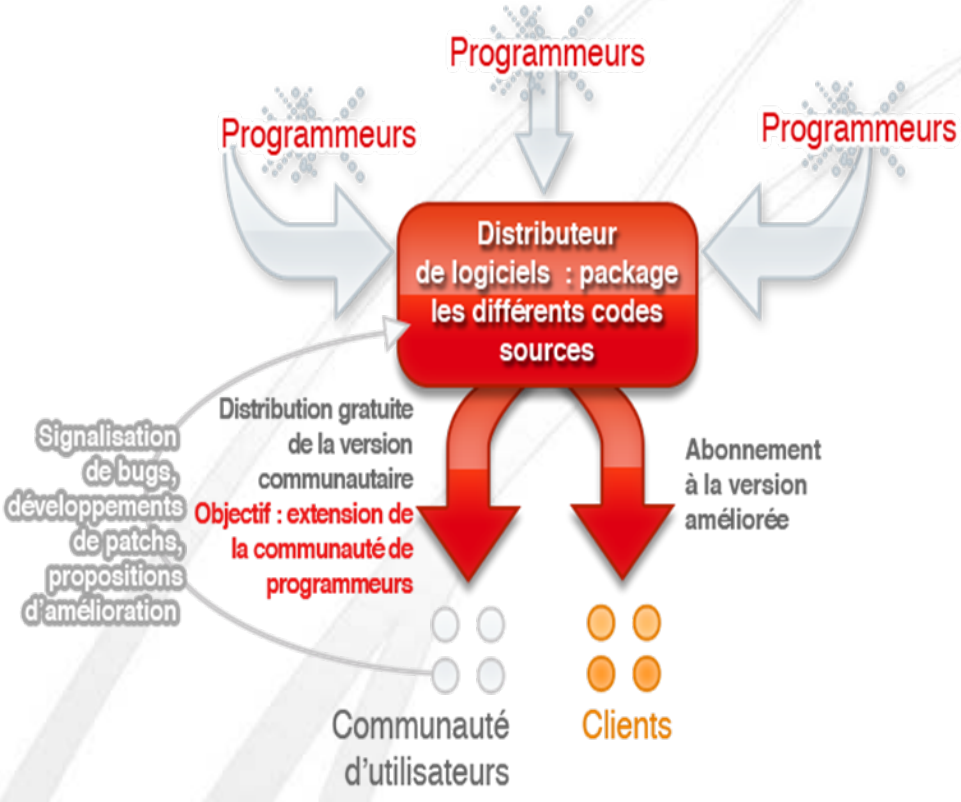
Sources : Spikesource, analyses faberNovel

(cc) BY-NC-SA

Le modèle de distribution à valeur ajoutée consiste à vendre une version standard d'un produit déjà existant

- 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, pré-installée sur les ordinateurs ou vendue sur support physique

- La « vente » s'effectue généralement sous la forme d'un abonnement à l'année au produit et à un ensemble de services annexes*



* : Les abonnements représentent ainsi 85% du chiffre d'affaires de Red Hat en 2006

Sources : Red Hat, analyses faberNovel

(cc) BY-NC-SA

Ce modèle propose une triple valeur client

Valeur client

Gain de temps : le client récupère directement une version packagée, testée et immédiatement compatible avec son ordinateur et ses logiciels

Transfert du risque lié à l'utilisation des solutions open source du client vers l'entreprise :

- Versions testées, certifiées et garanties
- Indemnisations en cas de problèmes graves
- Services d'assistance technique intégrés dans le packaging

Obtention régulière de nouveaux patches et mises à jour tout au long de la période d'abonnement

(cc) BY-NC-SA

Un exemple de distribution à valeur ajoutée : Red Hat

- Red Hat est spécialisée dans la distribution de Linux. Pour l'exercice 2006, elle affiche un chiffre d'affaires de 401 millions de dollars et un revenu net de 59,9 millions. En avril 2006, elle a racheté JBoss, entreprise open source spécialisée dans le middleware, pour un montant de 350 millions de dollars. En juillet 2007 la capitalisation boursière de Red Hat sur le NYSE était de 4,1 milliards de dollars et son nombre d'employés s'élevait à 1800 personnes.

- L'offre Red Hat est proposée en deux versions :
  - La version entreprise, testée et dont l'interopérabilité est certifiée
    - Red Hat Enterprise Linux, ou RHEL, qui permet seulement à deux utilisateurs de posséder simultanément RHEL
    - RHEL Advanced Platform, qui permet à un nombre illimité d'utilisateurs d'en bénéficier
  - La version communautaire (Fedora)

- La version entreprise existe en 5 modules différents :
  - Pour la version RHEL
    - Offre basique : 349$, 2 réponses business par jour *via* email, illimitées pour les incidents
    - Offre standard : 799$ par an, aide téléphonique disponible 12x5, illimitée par internet
    - Offre premium :1.299$ par an, aide téléphonique disponible 24x7, illimitée sur internet
  - Pour la version RHEL Advanced Platform
    - Offre standard : 1.499$, aide téléphonique 12x5, illimitée sur internet
    - Offre premium : 2.499$, aide téléphonique 24x7, illimitée sur internet

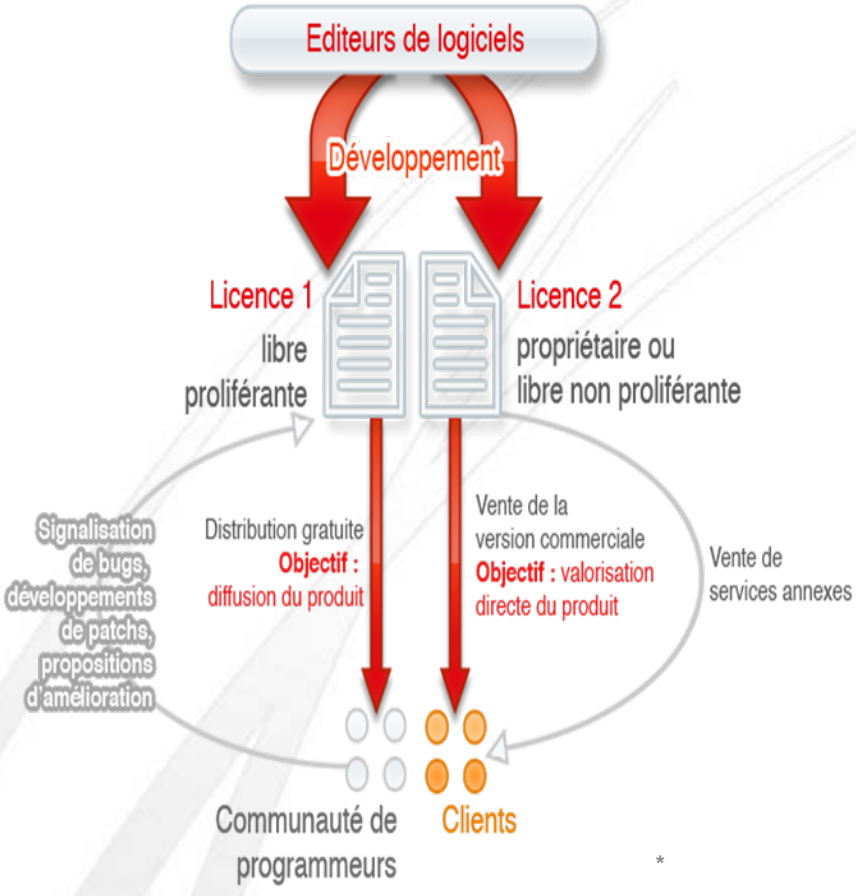
- Bien que ne lui assurant aucun revenu, Red Hat fait attention à ne pas se désintéresser de la version Fedora et participe activement à l'animation de sa communauté

Sources : Red Hat, analyses faberNovel

(cc) BY-NC-SA

Le modèle de licence double se fonde sur une discrimination des utilisateurs

- Le modèle repose sur un système de double licence :
  - Une licence libre open source pour le produit standard
  - Une licence plus protégée, garantie et généralement liée à un produit offrant plus de fonctionnalités
- La licence libre *doit* être de type copyleftée proliférante, car toute entreprise souhaitant intégrer le code source à un ensemble de produits plus vaste et le conserver sous licence propriétaire devra alors se tourner vers la version commerciale de la solution offerte
- Symétriquement, la version commerciale *doit* avoir une licence propriétaire pour éviter les risques de forking, ou libre non copyleftée ou persistante afin d'éviter les effets de contagion si l'entreprise cliente souhaite intégrer le code source dans un système plus vaste
- Cette solution permet de combiner les avantages de la licence libre (création d'une communauté de programmeurs, diffusion rapide permettant de bénéficier d'effets de réseaux) et ceux de la licence propriétaire (flux de revenu stable et connu, pas de risque de « prolifération » des licences open source)



(cc) BY-NC-SA

Les entreprises open source utilisant ce modèle de licence double doivent effectuer un double arbitrage

Type d'arbitrage

Pourcentage du code de la version commerciale contenu dans la version gratuite

- **Rôle joué par la communauté** : plus son rôle est important et plus la version gratuite doit détenir un pourcentage élevé du code de la version commerciale
- **Notoriété du produit** : plus le produit et le besoin auquel il répond sont connus et plus il sera facile pour l'utilisateur d'en percevoir son utilité. La société aura alors moins besoin de dévoiler sa version commerciale dans ce produit d'essai.
- **Notoriété de l'entreprise** :
  - Plus l'entreprise est connue et moins le risque de *forking* est important. La version gratuite open source peut donc contenir un pourcentage très important du code de la version commerciale, sans que le risque d'intensification de la concurrence ne soit trop élevé
  - Plus l'entreprise est connue et moins l'achat de sa version commerciale dépend de l'essai de sa version gratuite. Cette dernière n'a donc pas besoin d'être proche de la version commerciale

Degré de finition des produits commercialisés

- **Ressources et compétences internes de l'entreprise** : le modèle de licence double convient parfaitement aux entreprises développant des composants. Les entreprises choisissant de développer des produits finis doivent avoir les ressources internes pour animer une communauté de programmeurs, démarcher les grands comptes auxquels vendre leur solution commerciale, assurer l'assistance technique pour une base de clients beaucoup plus large, *etc.*

Facteurs de choix

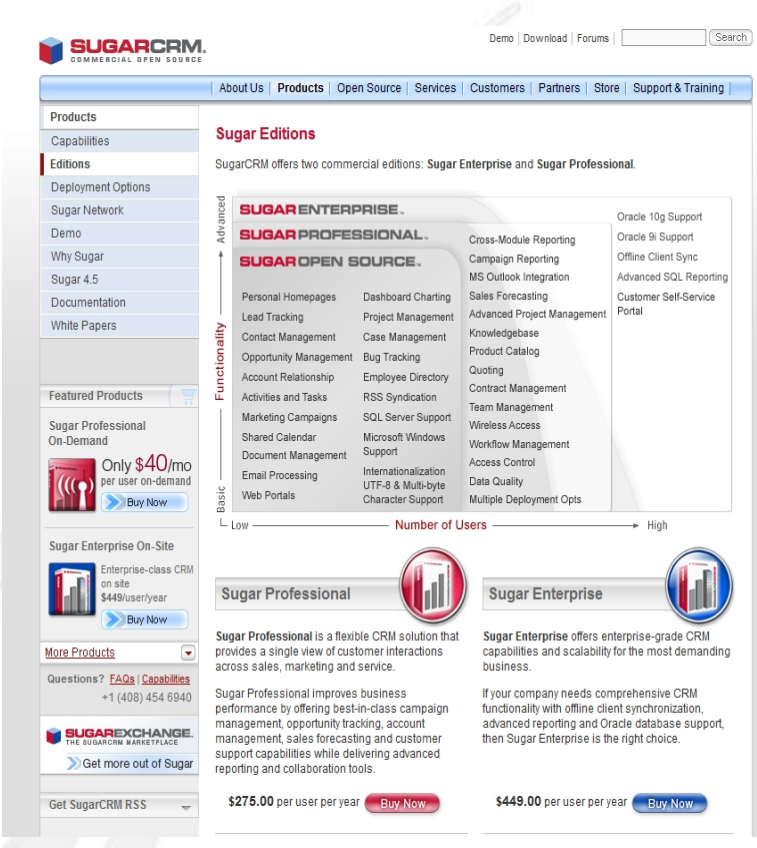
(cc) BY-NC-SA

Un exemple d'entreprise fondée sur le modèle de licence double : SugarCRM

- SugarCRM est une des entreprises leaders dans le domaine des outils open source de gestion des relations clients. Elle a affiché en 2006 un chiffre d'affaire de 6,60 millions de dollars et emploie plus de 100 personnes

- L'offre SugarCRM est divisée en deux versions :
  - La version open source totalement gratuite, qui contient 85% du code de la version professionnelle
  - La version commerciale
    - *Sugar Professional* : 275$ par an et par utilisateur
    - *Sugar Enterprise* : 449$ par an et par utilisateur, qui intègre des fonctionnalités plus développées (support pour les bases de données Oracle, synchronisation client hors ligne, *etc.*)

- SugarCRM propose également
  - Un ensemble de services (assistance technique, formation ligne, envoie de patches, *etc.*) accessibles depuis sa plateforme Sugar Network (119$ par an et par utilisateur)
  - Le *Sugar Sales Professional Service*, qui permet de définir une offre personnalisée pouvant comprendre une aide d'installation, une assistance plus importante et une offre de conseil pour optimiser le logiciel SugarCRM et l'adapter au besoin spécifique du client. Le coût de ces services peut fortement varier (entre 239$ et 4,995$)

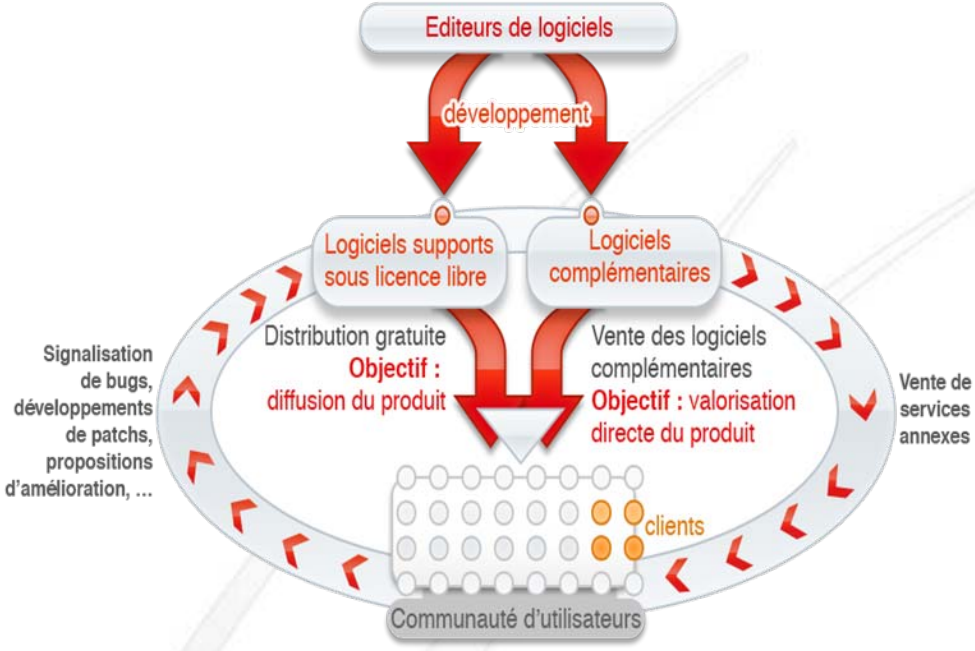


Sources : SugarCRM, analyses faberNovel

(cc) BY-NC-SA

Une variante de ce modèle : la commercialisation d'un produit complémentaire

- Cette variante consiste à commercialiser des logiciels complémentaires, plutôt que de vendre un produit quasiment identique sous deux licences différentes
- La valorisation complémentaire repose sur la conversion des *utilisateurs* en *clients* :
  - Les utilisateurs sont une base qui ne constitue pas une source de revenu
  - Cette base doit ensuite être *monétisée* via l'adoption du module complémentaire



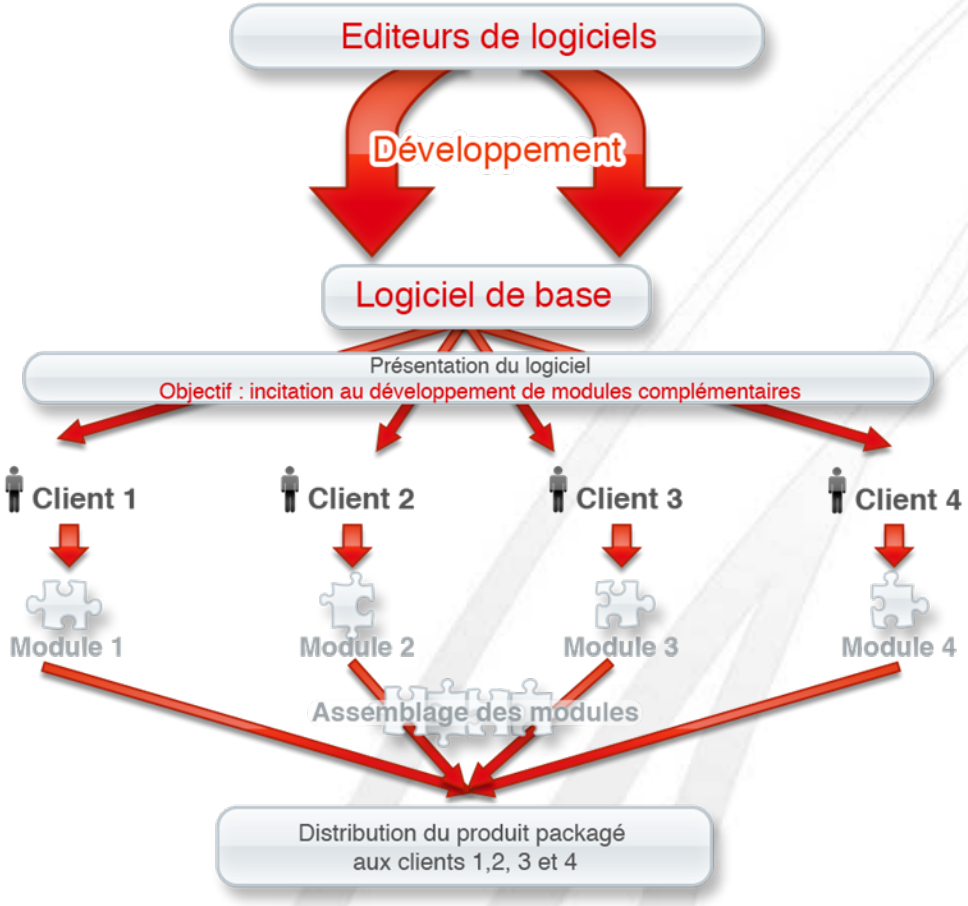
- Jusqu'en 1999, la société Roxen offrait un excellent exemple de ce principe
  - Centrée sur le développement et l'amélioration de son serveur web, la seule mesure de succès était alors le nombre de téléchargements quotidiens et le nombre total d'utilisateurs
  - Les choses ont changé avec le renouvellement de l'équipe de direction et aujourd'hui, comme l'explique un des managers : "*We have constantly moved away from the OSS concept towards a more traditional approach of selling proprietary software. We felt that something had to be done in order to survive. The original approach which was strongly influenced by ideas within the free and open source software movement was impossible to combine with profits in our case.*"

Sources : L. Dahlander [2005], analyses faberNovel

(cc) BY-NC-SA

Le modèle de « mutualisation » repose sur le développement successif de plusieurs modules…

- Le modèle de mutualisation consiste à éditer une version relativement simple de son produit pour ensuite développer des modules à la demande



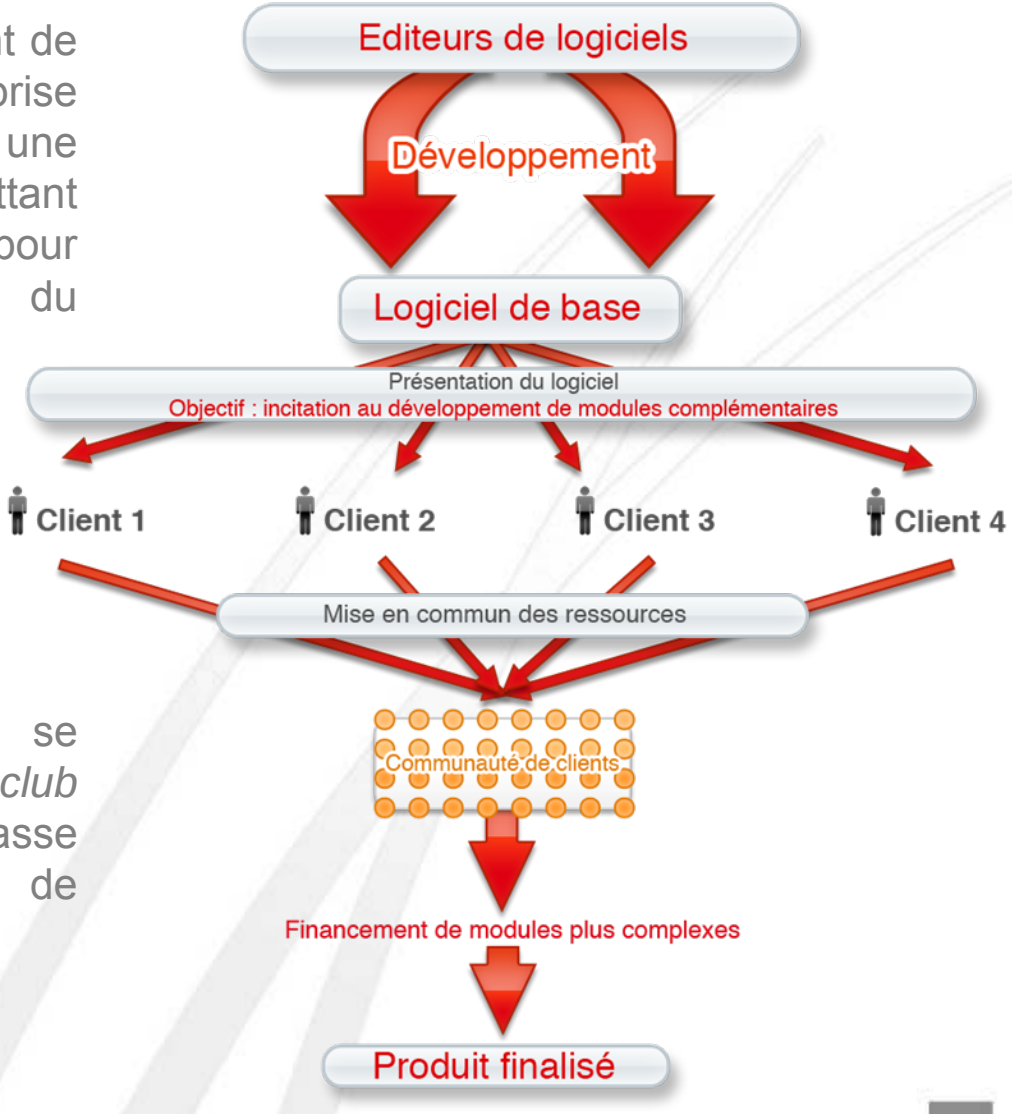
Sources : Muselli [2007], analyses faberNovel

(cc) BY-NC-SA

… et débouche généralement sur la création d'une communauté de clients

- Pour faciliter le développement de modules coûteux, l'entreprise open source peut créer une communauté de clients, mettant en commun leurs ressources pour financer le développement du module

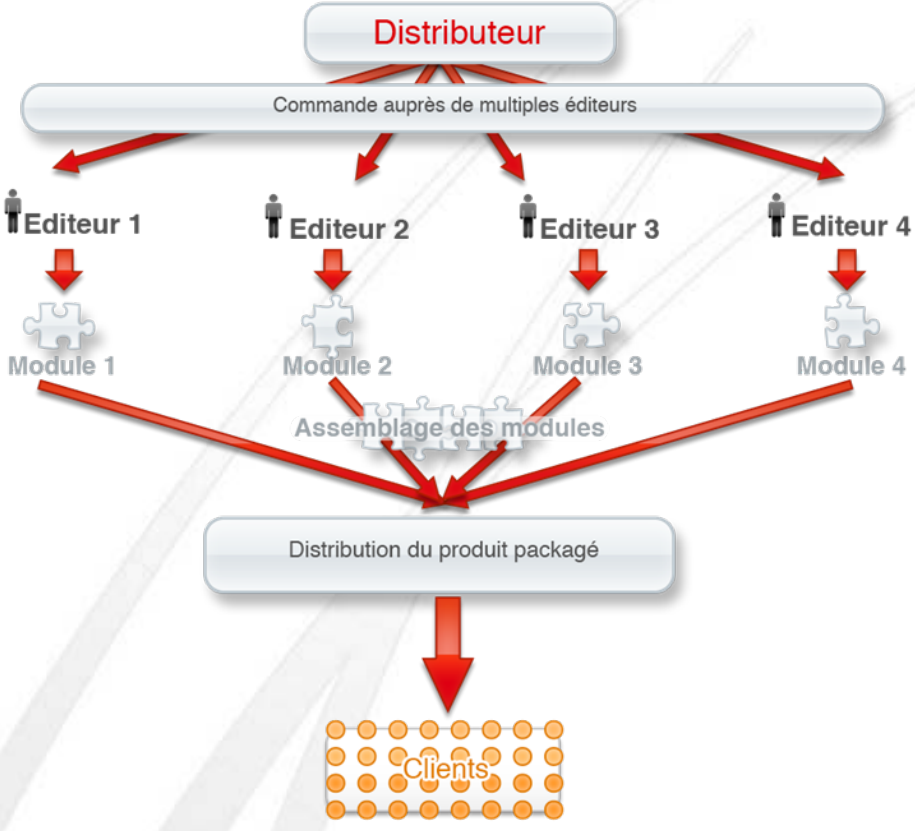
- Cette communauté peut se pérenniser et devenir un *club d'investisseurs* qui passe régulièrement commande de nouveaux modules



(cc) BY-NC-SA

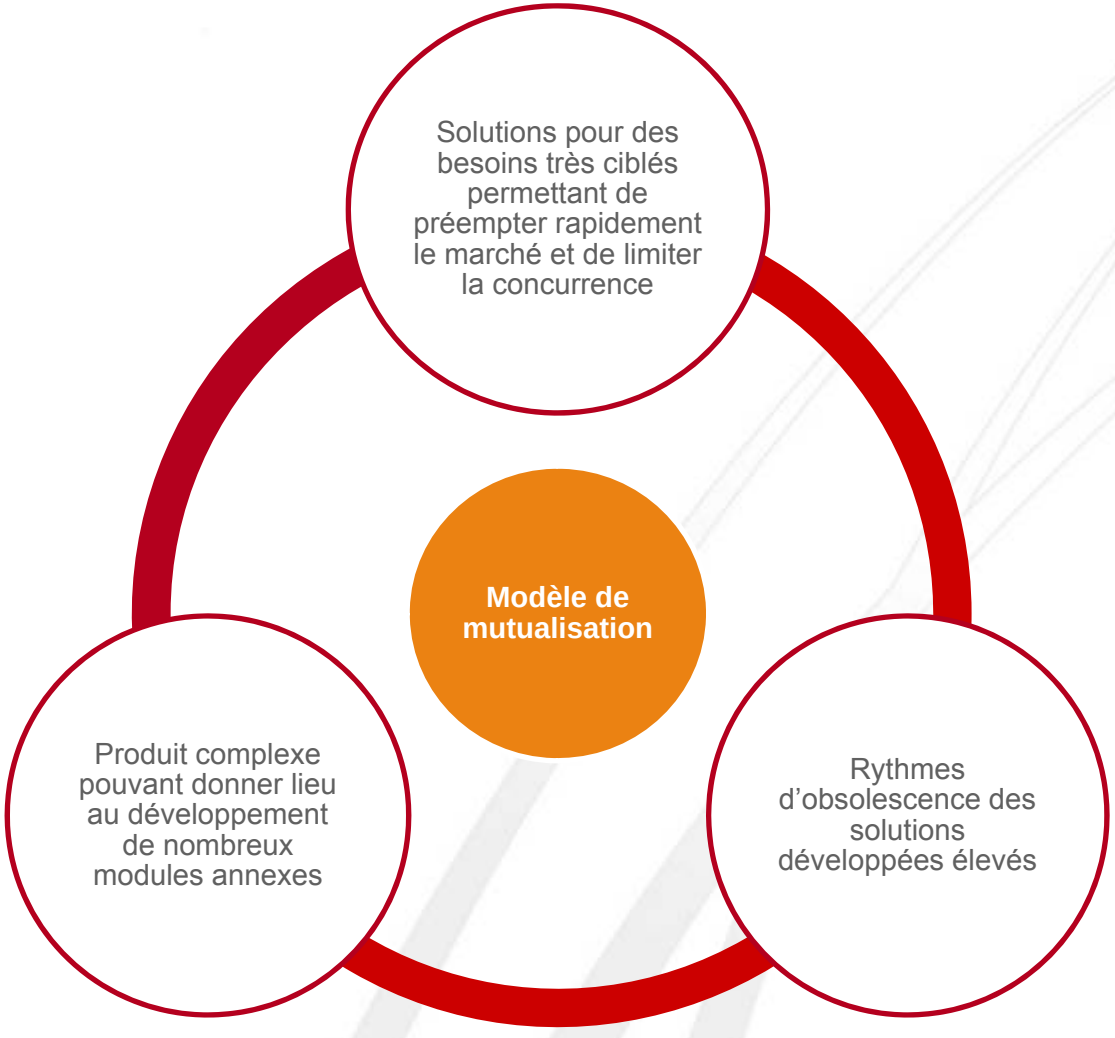
Une variante de ce modèle consiste pour des développeurs à mutualiser leurs modules

- Un modèle fréquent pour les entreprises open source est de développer des modules à la demande d'un distributeur-intégrateur
- Plusieurs éditeurs vont ainsi contribuer à l'élaboration du produit fini, qui, une fois packagé est distribué aux clients de l'intégrateur



(cc) BY-NC-SA

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



(cc) BY-NC-SA

La configuration des revenus des entreprises suivant ce modèle est variable

		Coûts pour le client 1 (1)	Coûts pour le client 2 (2)	Coûts pour le client 3 (3)	Revenu de l'entreprise open source (1)+(2)+(3)
Le module de base est développé en interne et n'est pas facturé	**Module de base (gratuit)**	0	0	0	0
Le client 1 paie le développement du module 1, les autres ne paient que son intégration	**Module 1**	Développement + intégration	Intégration	Intégration	1 développement + 3 intégrations
Les trois clients mutualisent leurs ressources pour payer le développement du module 2, chacun paie un tiers et son intégration	**Module 2**	1/3 du développement + intégration	1/3 du développement + intégration	1/3 du développement + intégration	1 développement + 3 intégrations
Les clients se réunissent dans un club d'investisseurs	**Module 3**	Frais d'inscription au club + cotisation annuelle + intégration du module 3	Frais d'inscription au club + cotisation annuelle + intégration du module 3	Frais d'inscription au club + cotisation annuelle + intégration du module 3	Frais d'inscription + cotisation annuelle des clients 1,2 et 3 + 3 intégrations

Sources : Muselli [2007], analyses faberNovel

(cc) BY-NC-SA

Un exemple de modèle de mutualisation : OpenTrust

- Open Trust est une société open source spécialisée dans les logiciels de sécurité des informations, qui comptait 60 employés pour un chiffre d'affaires de 6,7 million d'euros en 2004

OPENTRUST™ Building Trust in the Electronic World

- Elle développe en interne un module de base d'ICP (Infrastructure à Clefs Publiques) puis :
  - Le module est présenté aux clients potentiels
  - La communauté de clients se constitue
    - Chaque client va payer le développement d'un nouveau module correspondant spécifiquement à ses besoins
    - Les modules ainsi financés sont packagés en un produit unique
    - Ce produit est distribué à tous les membres de la communauté de clients, sans que les autres clients aient à repayer pour les modules qu'ils n'ont pas financés
  - Cette version améliorée est par la suite mise à la disposition de tous en téléchargement gratuit, après une période allant de 6 à 18 mois

- Open Trust a également créé un « club des contributeurs » avec un ticket d'entrée variant de 50.000 à 100.000€, auquel s'ajoute un abonnement annuel variant de 15.000€ à 25.000€

Sources : Open Trust, analyses faberNovel

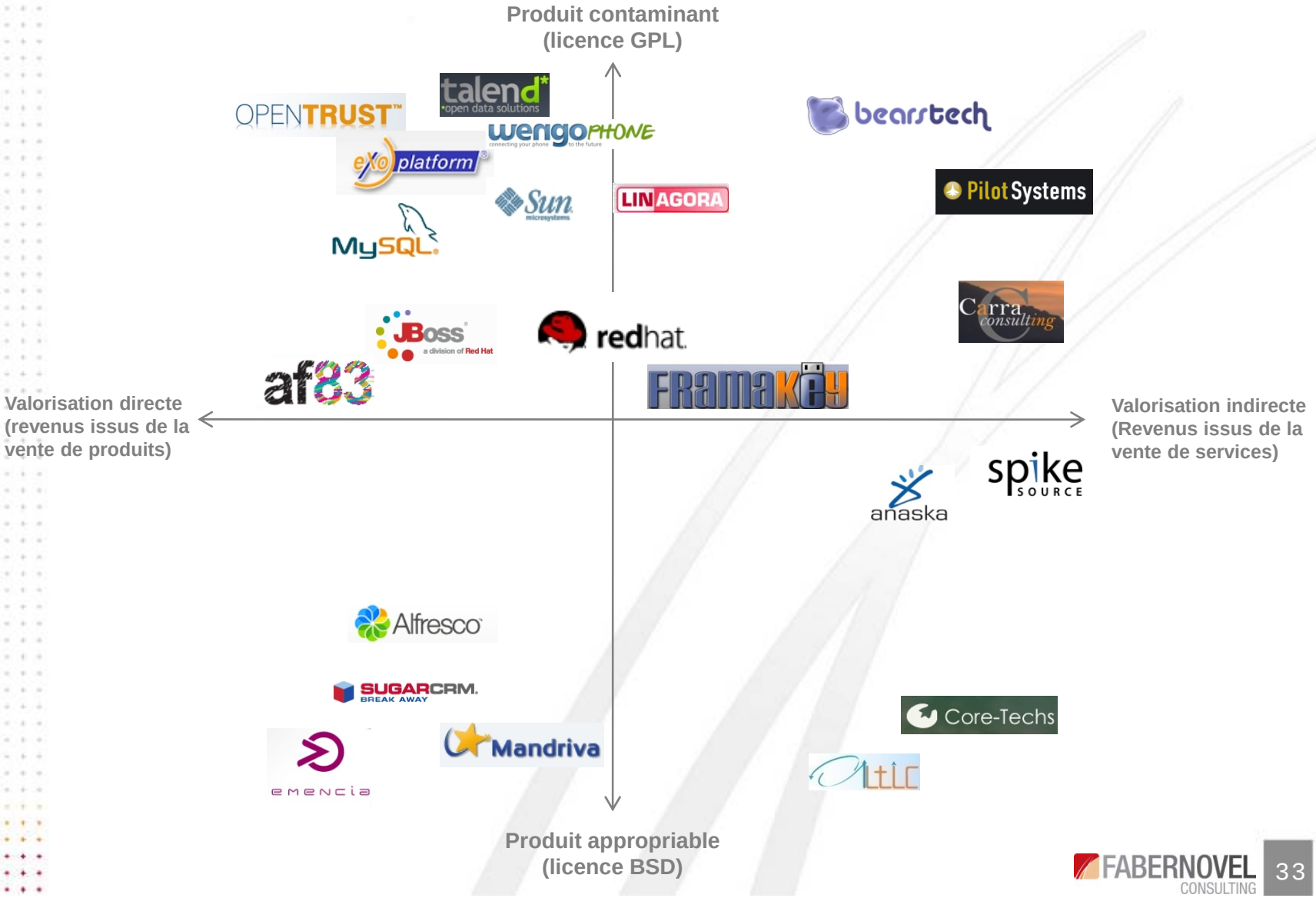
(cc) BY-NC-SA

Synthèse des différents *business models*

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, Core-Techs, Pilot Systems,
Distribution à valeur ajoutée	Intégration de différents code sources, offre de patches et mises à jour uniquement pour le logiciel distribué	Tests et garantie, offre patches et assistance technique modulable	Red Hat, Novell's SUSE, Mandriva, Framakey,
Licence double	Développement d'un logiciel 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, ExoPlateform,,
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,

(cc) BY-NC-SA

Les *business models se* distinguent par le niveau d'appropriabilité de leur licence et le mode de valorisation



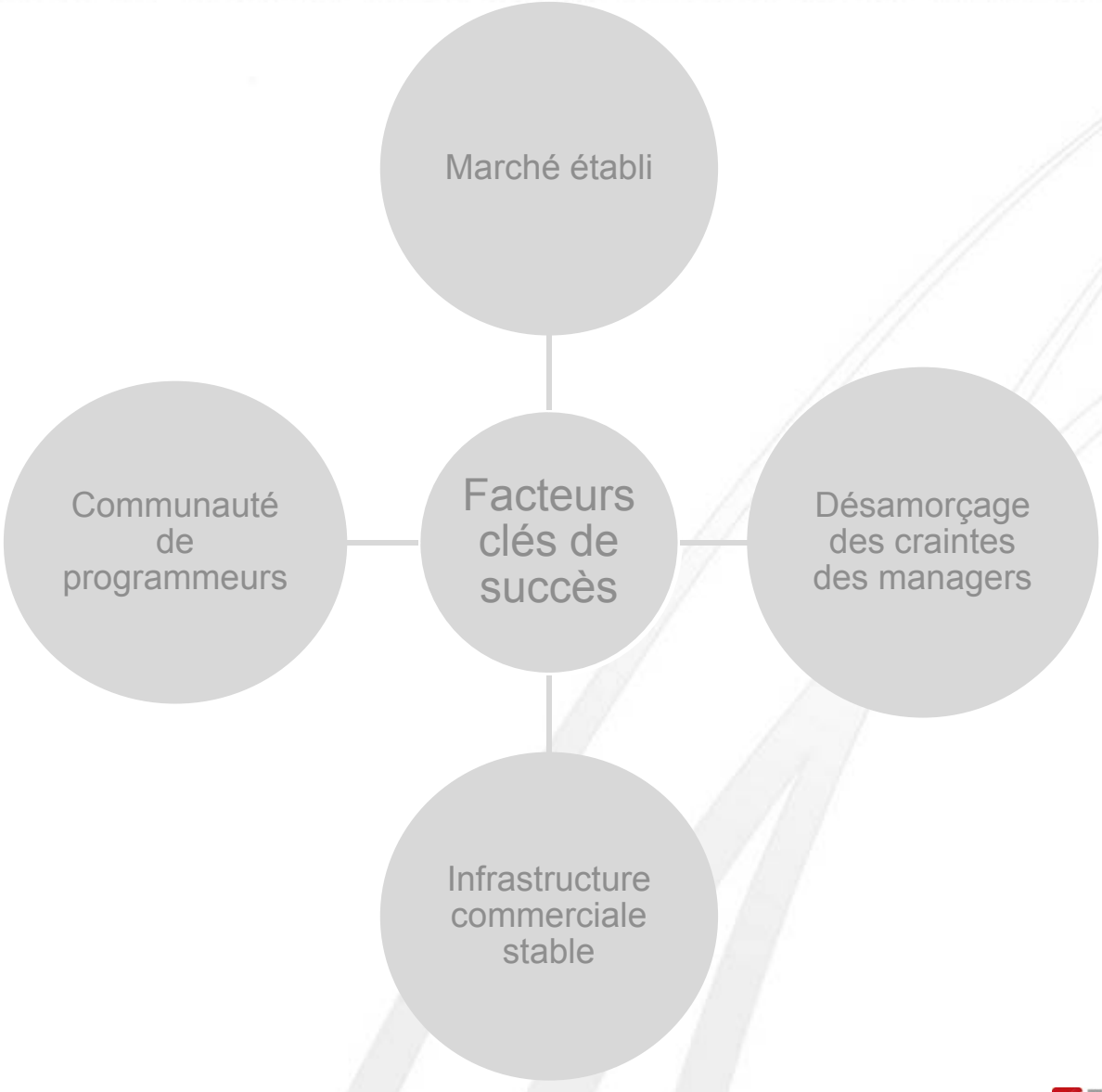
(cc) BY-NC-SA

Sommaire

- Contexte et objectifs de l'étude
- Les différentes typologies des *business models*
- Les facteurs clés de succès communs aux différents modèles
- Quelles stratégies à venir pour les logiciels open source ?

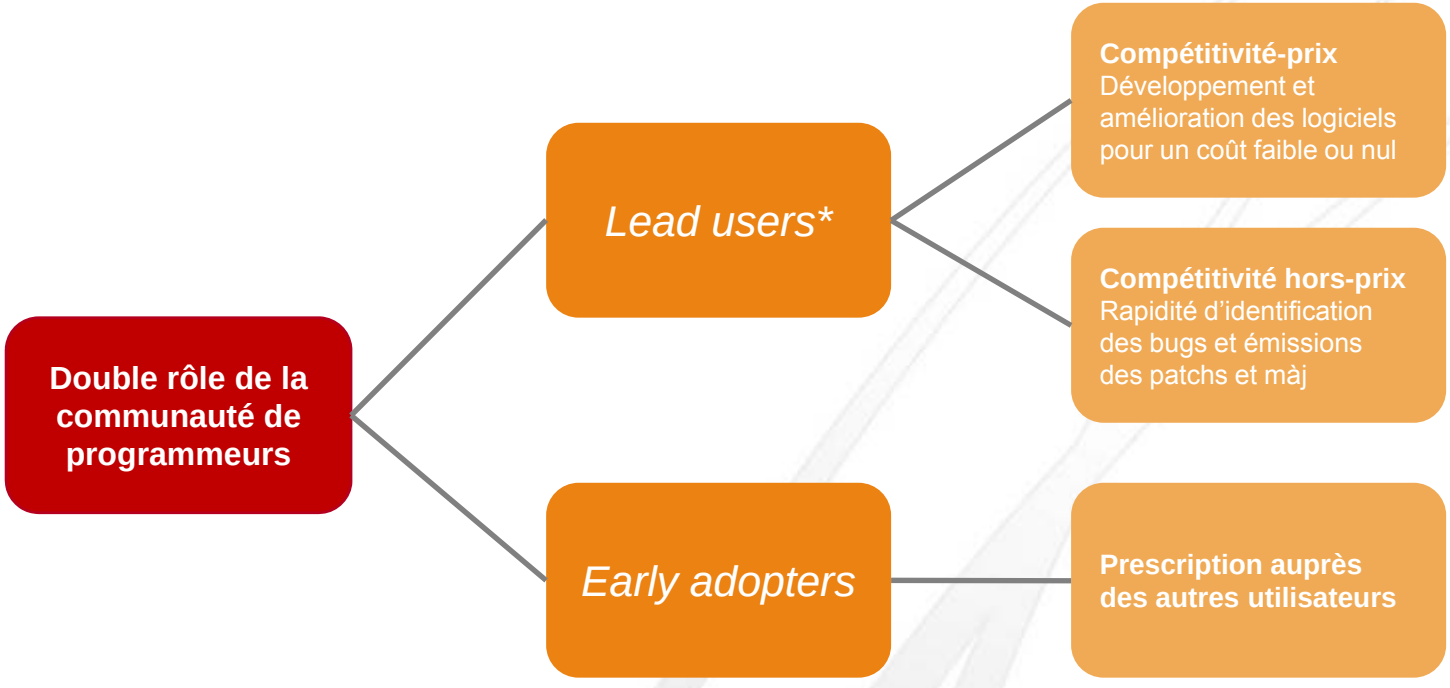
(cc) BY-NC-SA

Quatre facteurs clés de succès sont à prendre en compte pour la réussite d'une entreprise open source



(cc) BY-NC-SA

La communauté de programmeurs d'une entreprise open source constitue une ressource essentielle



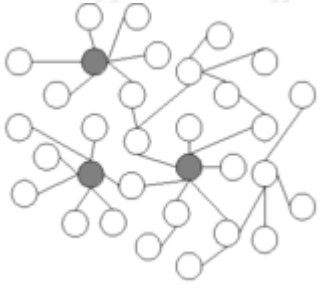
* Les *lead users* se distinguent des autres utilisateurs par leurs besoins en avance sur certaines tendances du marché. Les réponses à ces besoins précoces présentent une forte valeur pour ces consommateurs particuliers. Ces derniers sont alors incités à développer eux-mêmes des solutions

(cc) BY-NC-SA

La mise en œuvre d'un projet open source ne garantit pas la création d'une communauté

- Les communautés de développeurs sont des réseaux de type « *scale free network* », c'est-à-dire des réseaux organisés autour de quelques nœuds centraux et se développent selon le principe du « *preferencial attachment* », selon lequel plus un nœud a de connexions et plus il a de chance d'en ajouter de nouvelles

Exemple de random network



Exemple de scale free network

- Dans ces conditions, si certaines communautés peuvent croître très rapidement, la grande majorité des projets est condamnée à stagner si elle n'a pas atteint une certaine masse critique se situant aux alentours de la centaine de personnes

Composition de la communauté de programmeurs des projets recensés sur Sourceforge en 2003

Taille de la communauté	Nombre de projets	Administrateurs	Développeurs réguliers	Développeurs occasionnels	Utilisateurs actifs*
≤ 88	64847	80329 (47,8%)	34659 (20,6%)	33275 (19,8%)	19941 (11,8%)
88< ≤ 279	193	590 (2,1%)	1703 (5,7%)	17334 (60,3%)	9124 (31,7%)
>279	70	798 (0,9%)	2576 (2,7%)	53030 (55,8%)	38593 (40,6%)

* : Les utilisateurs actifs signalent des bugs, proposent des améliorations mais ne modifient pas le code source

Sources : Jin Xu, Scott Christley et Gregory Madey [2005], analyses faberNovel

(cc) BY-NC-SA

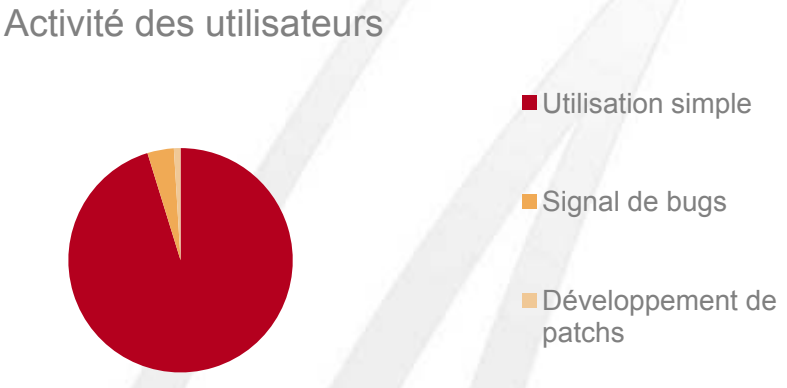
Trois leviers existent pour fédérer une communauté de programmeurs

Leviers	Actions
Adhésion	•Publier régulièrement des documents d'utilisation et être aussi transparent que possible (valeur cardinale de la communauté open source) •Développer les différents outils permettant d'interagir avec la communauté (mailing listes, forum, assemblées avec les membres de la communauté)
Animation	•Envoyer une newsletter de façon régulière aux membres de sa communauté (Pentaho envoie une newsletter mensuelle avec les évènements à venir ou différents conseils techniques) •Encourager les membres de sa communauté à partager les patches qu'ils ont développés (ex. : Sourcefire) •Participer à des projets émanant de sa communauté (les développeurs employés chez Sun participent à de nombreux projets et influencent indirectement l'orientation des développements)
Valorisation	•Impliquer sa communauté en la faisant participer à la définition des caractéristiques nécessaires aux nouvelles versions (Sun utilise sa communauté pour faire des *focus group* et des recherches marketing) •Inciter les employés de l'entreprise open source à participer et à intervenir lors de conférences ou d'évènements (ex. : MySQL ou Alfresco)

(cc) BY-NC-SA

La contribution de la communauté au développement du code source ne doit cependant pas être surestimée

- La communauté de programmeurs participe généralement activement à *l'amélioration* du produit, mais elle ne développe que rarement le *corps du programme*
- Parmi les 50 programmeurs ayant le plus contribué au développement de SugarCRM, 95% d'entre eux sont des associés de l'entreprise, alors que cette dernière revendique 5000 collaborateurs sur plus de 220 extensions. D'une façon générale, moins de 15 programmeurs développent plus de 85% du programme de base des logiciels distribués par les entreprises open source
- La participation des utilisateurs obéit à la règle du 1000/10/1 : 1000 utilisent le logiciel, 10 signalent des bugs et 1 développe des patches



Sources : Matt Asay, 2006, analyses faberNovel

(cc) BY-NC-SA

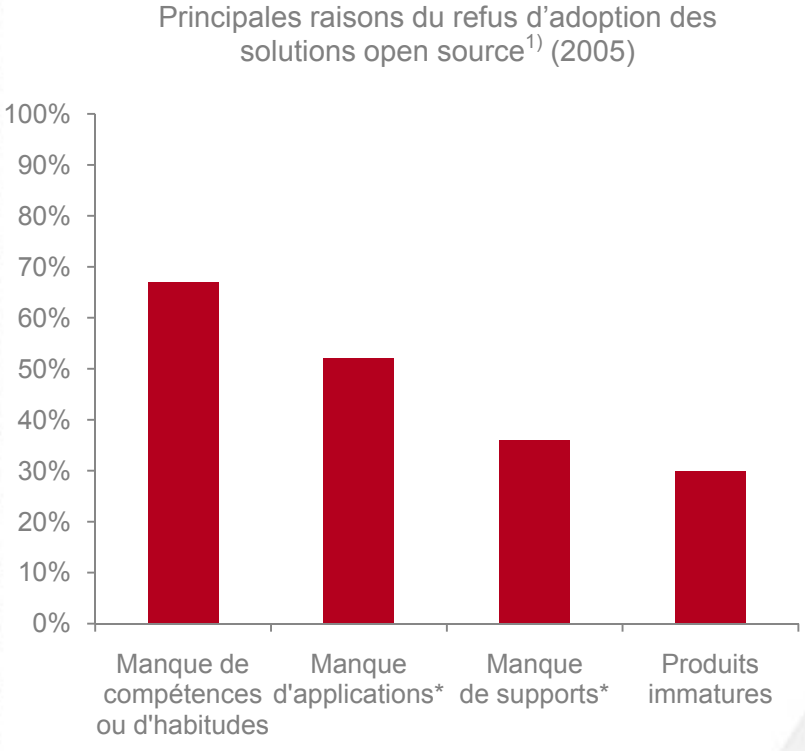
Les entreprises open source ont intérêt à opérer sur un marché déjà établi

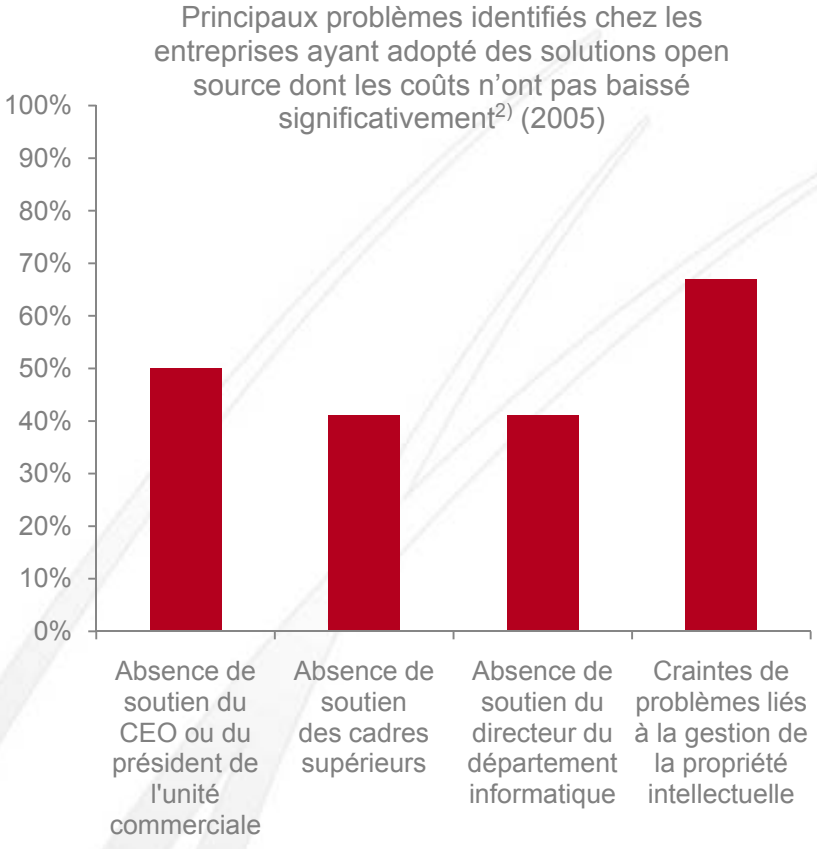
- Opérer sur un marché établi garantit :
  - **Une éducation des consommateurs**. Les clients potentiels ont précisément identifié leurs besoins, facilitant ainsi la valorisation du service vendu par l'entreprise open source
  - **L'existence d'une offre de référence**. Les entreprises open source font face à un problème de confiance de la part des utilisateurs qui tend à diminuer si un ou plusieurs logiciels propriétaires ont déjà fait leurs preuves et démontré leur efficacité. La présence d'une offre de référence permet par ailleurs de mettre plus facilement en valeur les principaux atouts d'un produit open source (prix, qualité de service, etc.)

- La plupart des grands succès de l'open source sont apparus sur un marché dominé par un vendeur de logiciels propriétaires :
  - Bases de données : MySQL et PostgreSQL vs. Oracle, IBM et Microsoft
  - ERP : Compiere vs. Oracle et SAP
  - CRM : SugarCRM et Compiere, vs. Siebel ou Oracle
  - OS : Red Hat vs. Microsoft

(cc) BY-NC-SA

Les managers manquent de confiance dans les entreprises open source





* : La crainte de manque d'applications et de manque de supports pour la mise en place des solutions open source est sans doute autant due à une situation réelle qu'aux préjugés et à la défiance naturelle des managers vis-à-vis des solutions open source. Cette crainte reflète donc largement la méconnaissance des solutions open source de la part des managers

1) Sur un panel de 140 entreprises, plusieurs réponses possibles
2) Sur un panel de 350 entreprises, plusieurs réponses possibles

Sources : Optaros 2005, Forrester 2005, analyses faberNovel

(cc) BY-NC-SA

Les entreprises open source doivent adopter une politique de communication axée sur la levée de la méfiance des managers

Rassurer sur le prestataire

- Actions ciblées auprès des managers et directeurs d'unités commerciales
- Objectif : démentir les préjugés sur la qualité du produit et rassurer quant au sérieux de l'entreprise
  - Mettre en avant la sécurité des solutions et les garanties proposées par l'entreprise open source
  - Démontrer la viabilité et la pérennité du prestataire, présenter le business model

Rassurer sur la propriété intellectuelle

- Actions ciblées auprès des managers et directeurs d'unités commerciales
- Objectif : démentir les préjugés habituels sur les risques liés à la gestion de la propriété intellectuelle
  - Mettre en avant les faibles risques légaux qui sont en réalité associés à l'utilisation des solutions open source
  - Insister sur le fait que les entreprises ont en réalité plus de risques de se faire poursuivre par une entreprise de logiciels propriétaires pour des problèmes dans leur gestion des comptes des licences, que par des entreprises open source pour tout autre type de problématique liée à la propriété intellectuelle

Sources : Forrester 2005, analyses faberNovel

(cc) BY-NC-SA

Les entreprises open source doivent offrir à leurs clients une infrastructure commerciale stable

- La plupart des utilisateurs non spécialistes et des entreprises clientes comprennent encore mal le fonctionnement du monde open source et craignent de ne pas trouver d'interlocuteur en cas de problème

- Le seul moyen de limiter cette crainte des utilisateurs est l'existence d'une entité commerciale claire, s'engageant à garantir les produits commercialisés et capable d'assurer un SAV

- La nécessaire création d'un SAV représente un centre de coûts important pour les entreprises open source

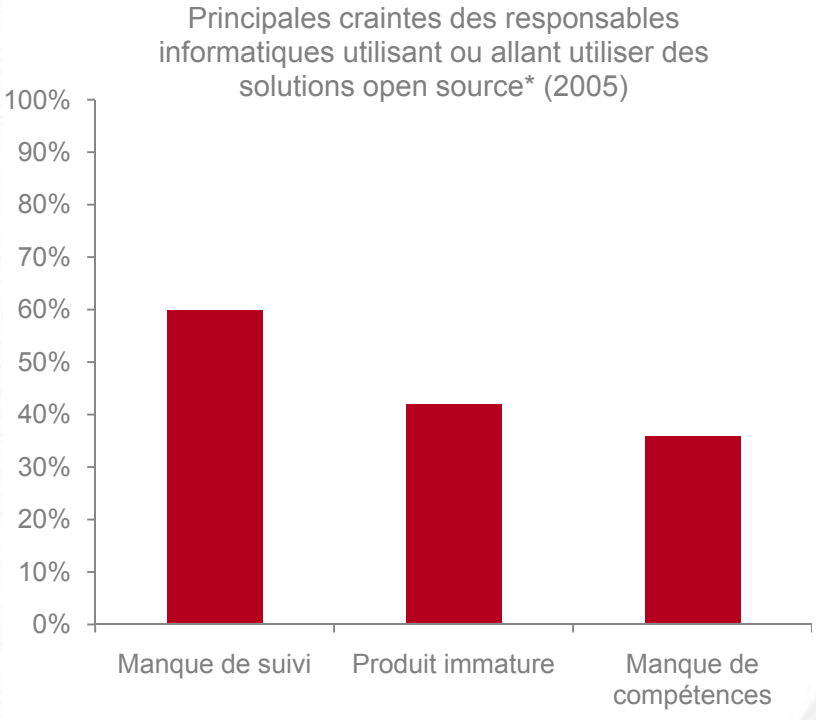
47%

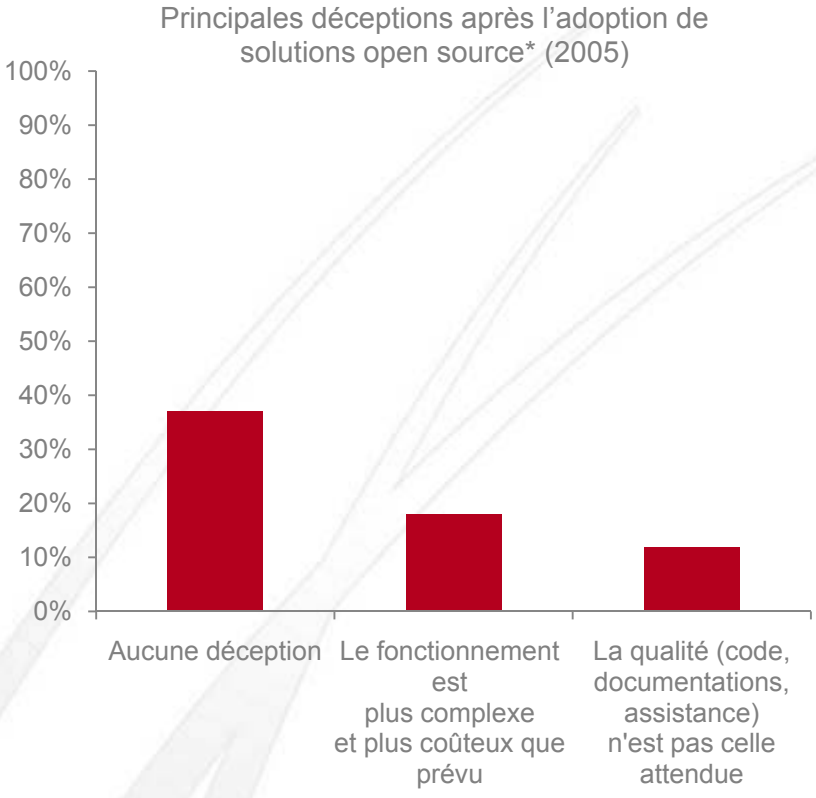
Pourcentage d'organisations américaines n'ayant pu trouver des ressources internes ou externes pour maintenir et améliorer leur système open source

Sources : Optaros 2005, analyses faberNovel

(cc) BY-NC-SA

Les entreprises déjà utilisatrices de solutions open source font face à un manque de suivi et de soutien





- Les entreprises open source doivent réduire la complexité de l'utilisation de leurs solutions. Ceci passe nécessairement par une amélioration de leur gestion de la relation client

* : Sur un panel de 95 directeurs informatiques, plusieurs réponses possibles

Sources : Forrester 2005, analyses faberNovel

(cc) BY-NC-SA

Sommaire

- Contexte et objectifs de l'étude
- Les différentes typologies de *business models*
- Les facteurs clés de succès communs aux différents modèles
- Quelles stratégies à venir pour les logiciels open source ?

(cc) BY-NC-SA

Les stratégies des entreprises open source pourraient connaître d'importantes inflexions dans les prochaines années

Les stratégies marketing

- Quel modèle marketing adopter ?
- Quelle place accorder au marketing et à la distribution ?

La concurrence accrue sur le marché des solutions open source

- Comment faire face à la concurrence des géants des solutions propriétaires se lançant dans l'open source ?
- Comment éviter l'intensification de la concurrence des autres entreprises open source ?

(cc) BY-NC-SA

Deux modèles marketing s'opposent

Adoption par l'usage

- Des versions d'essais relativement proches de la version commerciale sont mises en téléchargement gratuit
  - La version gratuite de SugarCRM contient 85% du code de Sugar Professional
  - La version gratuite d'Alfresco contient 100% du code de sa version commerciale
- Après une période d'essai, des besoins de support et de nouvelles fonctionnalités émergent chez l'utilisateur
- Les dépenses marketing sont faibles, les ressources de l'entreprise sont concentrées sur le développement des produits

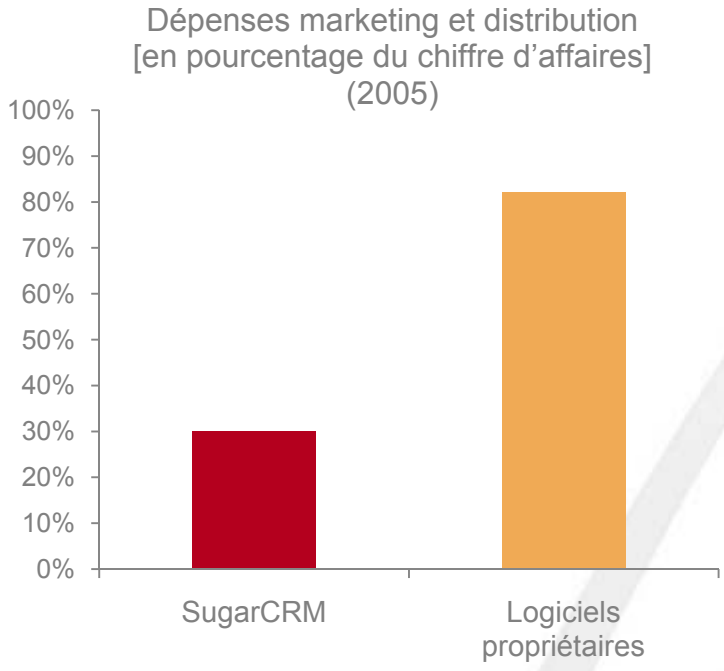
Campagnes marketing

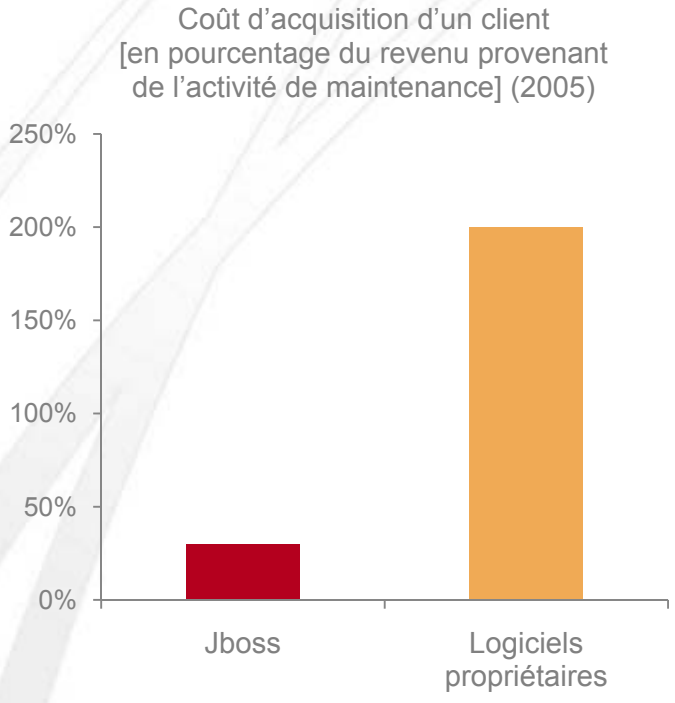
- Les entreprises consacrent une part importante de leurs ressources au département marketing
  - MySQL et SugarCRM renforcent leur département marketing
  - SugarCRM et JBoss étudient la possibilité de recourir à Google Adwords et de lancer des campagnes publicitaires à partir de bannières internet
  - Firefox a demandé en 2005 à sa communauté d'utilisateurs de participer à l'élaboration de publicité vidéos, destinées à de futures campagnes télévisées et internet
- Les entreprises ne comptent pas uniquement sur une adoption virale et développent une compétence en communication
- Cette stratégie permet de mieux cibler les clients et d'avoir un meilleur contrôle de son image

(cc) BY-NC-SA

La plupart des dirigeants d'entreprises open source mettent en avant les faibles coûts marketing et de distribution

- Le modèle marketing des entreprises open source serait de type « *pull* », dont l'objectif est d'obtenir un maximum de téléchargement de la version gratuite et de « monétiser » ensuite cette base
- Les entreprises open source n'ont pas à « pousser » l'adoption de leurs produits par de vastes campagnes marketing



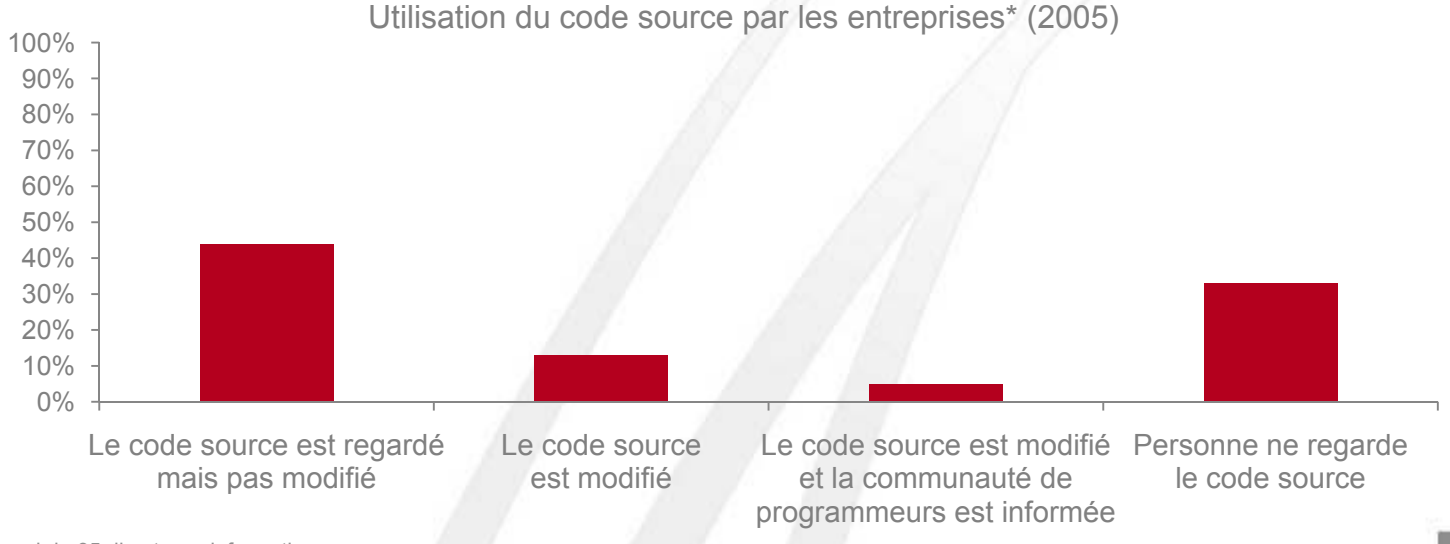


Sources : Goldman Sachs, SugarCRM, Jboss, analyses faberNovel

(cc) BY-NC-SA

Les dépenses marketing des entreprises open source augmentent cependant avec leur développement

- Les coûts de marketing et de distribution sont un poste de dépenses relativement faible au démarrage de l'entreprise, mais sont appelés à croître avec son développement
- Le processus « d'adoption par l'usage » fonctionne peu pour les transactions de montants élevés, mais est néanmoins vérifié en ce qui concerne les transactions de faibles montants
- Le *business model* d'une entreprise open source peut donc :
  - Ne pas intégrer *au départ* des coûts de marketing et de distribution élevés, libérant des ressources pour que l'entreprise se consacre au développement de son produit qui doit permettre à lui seul de lui apporter ses premiers clients (constitution d'une communauté d'utilisateurs)
  - Prévoir *à terme*, des coûts de marketing importants pour contrôler son image de marque
  - Adjoindre des coûts d'action commerciale, concentrée sur les « grands comptes »
- Une politique de communication accrue est d'autant plus importante que l'accès au code source n'est pas *en soi* une raison suffisante pour adopter des solutions open source



* : Sur un panel de 95 directeurs informatiques

Sources : Forrester 2005, analyses faberNovel

(cc) BY-NC-SA

Les stratégies de communication pourraient devenir plus ciblées

- Le coût des outils informatiques est rarement supporté par les départements qui les utilisent, mais est attribué* :
  - Soit au département informatique
  - Soit directement au niveau général de l'entreprise
- Les membres des différents départements d'une entreprise préfèrent commander un logiciel propriétaire, car :
  - Ils le croient plus « sûr » et ont l'habitude de travailler avec
  - Ils n'ont pas à supporter le surcoût que représente l'utilisation d'un logiciel propriétaire, puisque celui-ci est reporté sur un autre département
- Les entreprises open source doivent donc avoir une politique de communication active auprès des directions générales des entreprises en insistant sur la baisse globale des coûts si les solutions open source sont étendues à *l'ensemble* des départements de l'entreprise

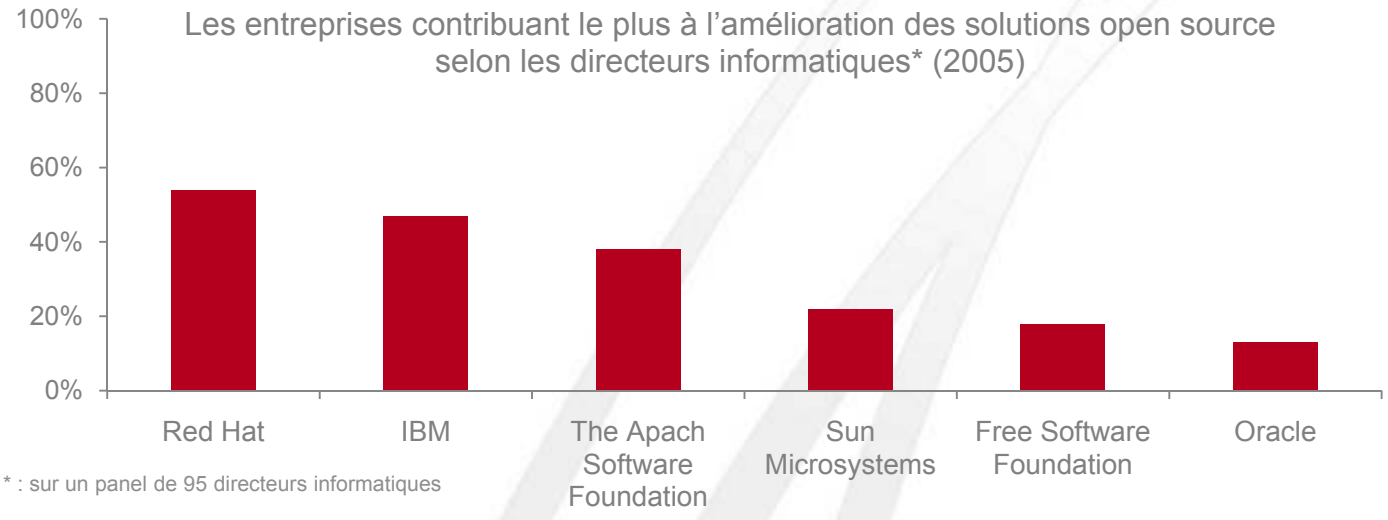
* : L'achat d'une licence pour un logiciel de GRC par le département marketing ne sera pas un coût directement imputé au fonctionnement de ce département, mais aura tendance à être reporté au niveau général de l'entreprise ou au niveau du budget du département informatique

(cc) BY-NC-SA

Les entreprises open source affrontent une concurrence accrue des entreprises de logiciels propriétaires

- Récemment, les entreprises de logiciels propriétaires ont investi massivement dans le domaine de l'open source.
  - IBM 2001 : annonce d'un programme d'investissement d'un un milliard de dollars dans Linux
  - IBM 2005 4,5 milliard de dollars de revenus estimés provenant de l'open source

- Cette orientation correspond à une modification dans les stratégies des entreprises de logiciels propriétaires, plutôt qu'une simple tentative pour améliorer leur image :
  - Sur un échantillon de 50 projets open source ayant une véritable activité en 2005[1)], 18 ont reçu 99,99% des investissements réalisés entre 1995 et 2005.
  - Cette importante concentration révèle un processus de sélection interne des investissements et témoigne du caractère stratégique de ces activités pour les entreprises propriétaires



* : sur un panel de 95 directeurs informatiques

1) : C'est-à-dire les projets dont la participation de la communauté équivaut en nombre d'heures à 7 personnes à temps plein

Sources : Marco Iansiti et Gregory L. Richards [2006], IBM, Forrester 2005, analyses faberNovel

(cc) BY-NC-SA

Les entreprises de logiciels propriétaires utilisent désormais l'open source pour valoriser leurs solutions propriétaires

- Les entreprises de logiciels propriétaires ont recours au modèle de valorisation complémentaire et à une politque de verrouillage des utilisateurs :
  - La distribution gratuite ou à prix très bas de solutions open source permet d'influencer les standards utilisés par les communautés d'utilisateurs et de les conduire vers les solutions propriétaires sur des segments différents
  - Une fois cette influence opérée, les utilisateurs sont plus enclin à payer un prix élevé pour les solutions propriétaires proposées

- IBM et Oracle offrent une parfaite illustration de cette stratégie :
  - Elles sont en compétition directe avec des solutions open source sur leur cœur d'activité, notamment les solutions middleware et les applications d'entreprises
  - Elles développent des projets open source sur les segments de marché qui ne correspondent pas à leur cœur d'activité. Ainsi Oracle contribue au développement d'Apache, Eclipse ou PHP sur le segment du middleware développement d'outils

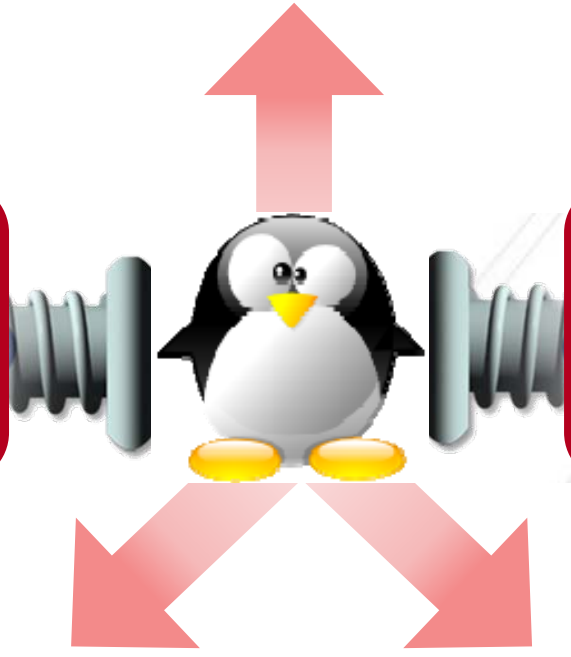
(cc) BY-NC-SA

Le monde de l'open source pourrait être amené à évoluer selon trois axes

Concentration du secteur

Intensification de la concurrence

Arrivée des entreprises de logiciels propriétaires



Effet de seuil

Des compétences coûteuses (marketing, commercial, distribution, etc.) sont nécessaires au développement

Stratégie de recherche de niche

Coopération renforcée avec des entreprises propriétaires

(cc) BY-NC-SA

Une tendance à la concentration des entreprises open source émerge

- La concentration d'un secteur est une conséquence classique de l'intensification de la concurrence

- Dans l'open source, ces rapprochements suivent deux schémas différents :
  - **Une concentration verticale** : les entreprises d'un même secteur d'activité (serveur d'application, gestion des bases de données) intègrent l'ensemble des éléments de la chaîne de valeur, de la production à la distribution en passant par la vente de services complémentaires. Ex. : Novell Bull et Open Trust signent un partenariat en septembre 2004 sur l'assistance technique des solutions Novell

  - **Une concentration horizontale** : les entreprises conservent la spécificité de leur *business model* et cherchent à atteindre une masse critique suffisante. Ex. : Alliance Open Trust HP et Atos Origin en décembre 2004 pour créer une plateforme unique et centralisée d'assistance

(cc) BY-NC-SA

La recherche de niche ou la coopération sont des solutions *a priori* plus risquées

Stratégie de niche	Coopération
Les coûts de développement relativement bas des produits open source - Permettent d'adopter une stratégie de niche (amortissement de l'investissement sur une base réduite d'utilisateurs) - Limitent les barrières à l'entrée du secteur (menace permanente de nouveaux entrants)	La coopération avec les entreprises de logiciels propriétaires - Permet aux firmes open source de profiter de compétences précieuses (ex. : Marketing et commercial) - Peut aboutir à une relation type prestataire/client (un catalogue de fournisseurs de taille réduite pour un client de taille mondiale)

(cc) BY-NC-SA

Et maintenant, permettez-nous de nous présenter…

(cc) BY-NC-SA

L'activité de faberNovel se décompose en trois métiers

Consulting

Stratégie et organisation pour la croissance et l'innovation

- Méthodes, analyses et aide à la décision pour les grandes entreprises
- Mission de conseil sur le thème de l'innovation
  - Stratégie et innovation
  - Organisation et innovation
  - Gestion du changement
  - Gestion des connaissances
  - Gestion de portefeuille R&D

Stimuler les gènes innovants

XP

Conduite d'expérimentations et de lancements de projets

- Expérimentation stratégique
  - Acquisition rapide de compétences clés
  - Déploiement de solutions à risques limités
- Conception et développement d'offre innovante
  - Conception fonctionnelle
  - AMO et conduite de projet
  - Conception et validation économique
  - Recherche de partenaires
  - Lancement de pilotes et REX

Mettre en œuvre rapidement

Venturi

Développement de projets internes et investissements

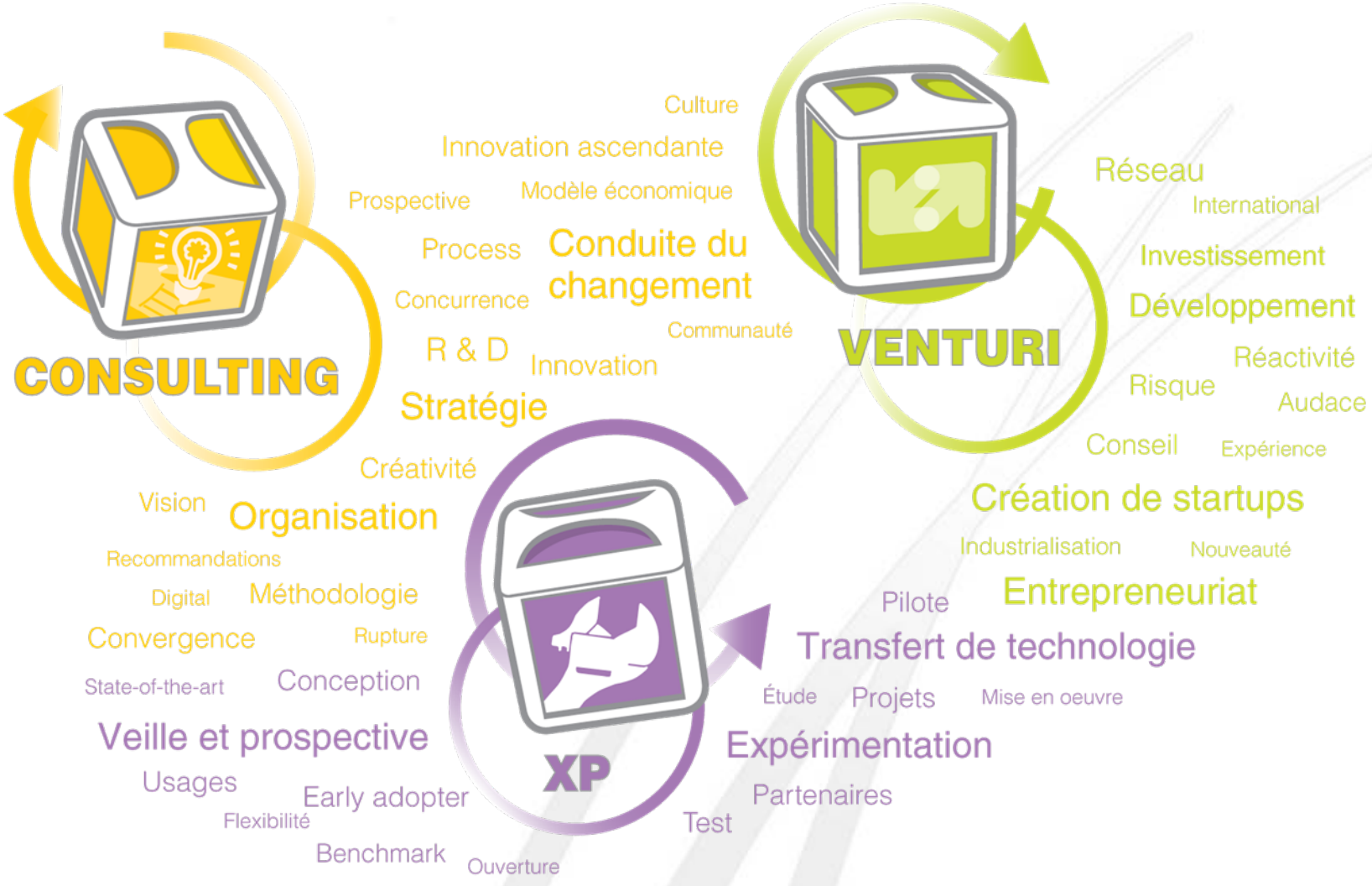
- Activité de capital amorçage, « excubateur »
  - Investissement et concrétisation de projets portés en interne
  - Aide à la création d'entreprise
  - Prise de participation dans le capital offrant de véritables leviers d'action

af83 digitick C4Mprod Timuzo

Rester entrepreneur

(cc) BY-NC-SA

faberNovel accompagne les projets depuis leur positionnement et jusqu'à leur industrialisation

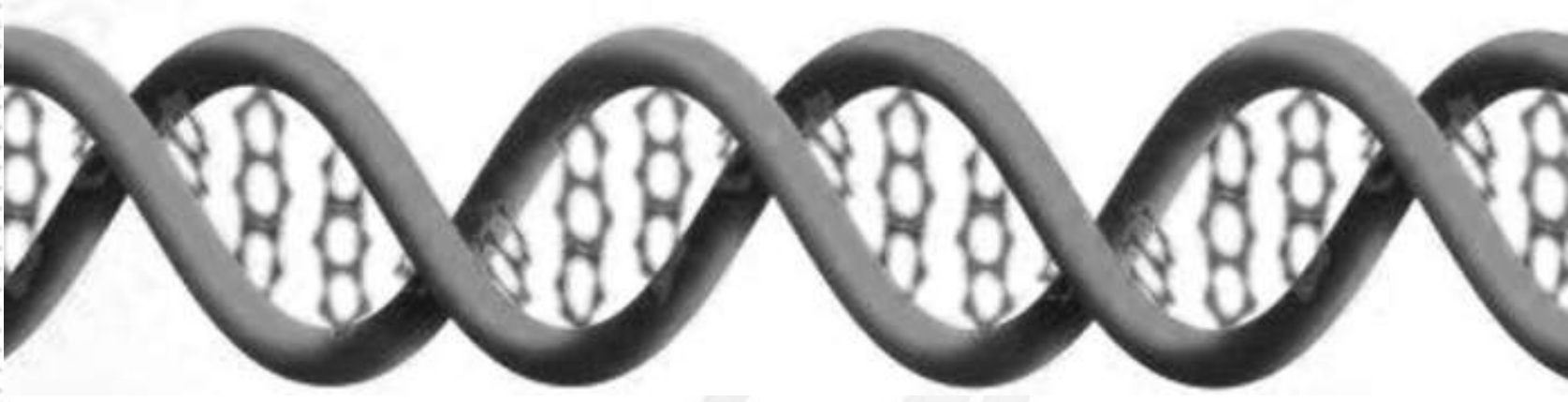


(cc) BY-NC-SA

faberNovel Consulting rassemble les activités de conseil du groupe faberNovel

La mission de faberNovel Consulting : stimuler les gênes innovants des entreprises

- **Veille prospective**
  - Technologies
  - Marchés
  - Usages
- **Stratégie**
  - Stratégies de croissance
  - Plate-forme d'innovations
  - Gestion de portefeuille projets
  - Pilotage de l'innovation
- **Mise en œuvre**
  - Benchmark concurrentiel
  - Cahier des charges fonctionnel
  - Partenariats / valorisation



- **Organisation**
  - Innovation participative (SMI)
  - Innovation collaborative (IRC ®)
  - Développement de l'intrapreneuriat
- **Gestion du changement**
  - Partage de Bonnes Pratiques
  - Animation de communautés
  - Design des connaissances
  - Transfert de technologie

(cc) BY-NC-SA

Pour en savoir plus sur ce sujet, n'hésitez pas à nous contacter…

42, boulevard de Sébastopol I 75003 Paris I France
Tél. : +33 1 42 72 2004 I Fax : + 33 1 42 72 2003
Web : www.fabernovel.com
Email : stephane.distinguin@fabernovel.com
pierre-yves.platini@fabernovel.com

(cc) BY-NC-SA

FABER NOVEL [fabɛʀn>vɛl] cabinet de conseil - 2003 : lat. *faber*, qui fabrique et *novel*, dimin. de novus, nouveau. 1◆ STRATEGIE Politique d'entreprise dans les secteurs impliquant des technologies de pointe ; innovation de rupture. 2◆ ORGANISATION Services d'optimisation pour la performance de l'innovation 3◆ MISE EN ŒUVRE Déploiement de dispositifs d'impulsion et de gestion de l'innovation, conduite du changement et formation pour l'innovation. 4◆ ASSISTANCE A MAITRISE D'OUVRAGE Etude de faisabilité, pilotage et accompagnement de projets innovants. 5◆ PROSPECTIVE Anticipation de tendances et d'usages technologiques, études veille et méthodologies pour l'innovation.

(cc) BY-NC-SA

Bilbiographie

- Dahlander L., “Appropriation and approbility in Open Source Software”, *International Journal of Innovation Management* Vol. 9 No.3 pp. 259-285, sept. 2005
- Goulde, M. et Mulligan, J. A. (2007), “How to Turn an Open Source Product into a Commercial Business”, *Forrester Research*, 23 Janvier 2007
- Goulde, M. (2005), “Open Source Usage is up, but Concerns Linger”, Forrester Research Paper, 23 Juin 2005
- Iansiti Marco et Richards Gregory L. (2006), “The Business of Free Software: Enterprise Incentives, Investment, and Motivation in the Open Source Community”, Working paper, Harvard Business School, http://www.hbs.edu/research/pdf/07-028.pdf
- Jullien, N. Clément-Fontaine, M. et Dalle J.M. (sous la direction de), “Nouveaux Modèles économiques, nouvelle économie du logiciel”, rapport RNTL Janvier 2000
- Krishnamurhty Sandeep (2003), « An Analysis of Open Source Business Models », Working paper, University of Washington, Bothell
- Lerner, J. et Tirole, J. (2000), “ The Simple Economics of Open Source”, *NBER Working Paper*, No. 7600
- Lerner, J. et Tirole, J. (2001), “The Open Source Movement : Key Research Questions”, *European Economic Review*, 45:819-826
- Muselli L. (2004), « Licences informatiques et modèles d’affaires open source », *13ème conférence de l’AIMS*, 2004
- Muselli L. (2007), “Business models and the payment of open-source software publishers. Mutualisation : an original business model” Conférence “*The diffusion of FLOSS and the Organisation of the Software Industry: From Social Networks to Economic and Legal Models*”, Nice-Sophia Antipolis, 31 mai et 1er juin 2007
- Pal, N. et Madanmohan, T. (2002), “Competing on Open Source : Strategies and Practise”, MIT Working Paper,
- Rogers, *Diffusion of Innovations*, the Free Press, 1995
- Raymond, E. (1999), « La Cathédrale et le bazar », http://www.linux-france.org/article/these/cathedrale-bazar/
- Schiff Aaron (2002), « The Economics of Open Source Software : A Survey of the Early Literature », *Review of Network Economics*, Vol. 1, Isssue 1-March 2002
- Stürmer, M. (2005), « Open Source Community Building », Working Paper, Open Source Community, MIT, http://opensource.mit.edu/papers/sturmer.pdf
- Välimäki, M. (2003), « Dual Licensing in Open Source Software Industry », *Systèmes d’Information et Management*, 8(1), 63-75
- Walli, S., Gynn D. et von Rotz B., (2005) :”The Growth of Open Source Software in Organizations”, *Optaros White paper,* http://www.optaros.com/en/publications/white_papers_reports
- XU, J., Gao Y., Christley, S.et Madey G. (2005), “A Topological Analysis of the Open Source Software Development Community”, *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005

(cc) BY-NC-SA