

Logiciel libre, une introduction

Roberto Di Cosmo



Université Paris Diderot
UFR Informatique
Laboratoire Preuves, Programmes et Systèmes
roberto@dicosmo.org

March 15, 2012

Acteurs

Modèles économiques

Cadre général
Modèles hybrides

La commande publique comme moteur

Libre n'est pas gratuit, et gratuit n'est pas libre

non libre, gratuit :

Internet Explorer, MacTCP, Acrobat Reader, freeware, etc.

non libre, non gratuit :

le plus connu ...

libre, gratuit :

Firefox, Linux, Gimp, OpenOffice, SciLab, OCaml, R, GeoGebra, etc.

libre, non gratuit :

distributions GNU/Linux commerciales, développement de logiciel libre sur demande ...

Qui utilise ou écrit du logiciel libre? Pourquoi?

Logiciel libre: *les acteurs*

l'utilisateur actif :

- ▶ plus grande stabilité / pérennité / flexibilité
- ▶ mutualisation des coûts
- ▶ transfert du support vers développement
- ▶ sécurité
- ▶ maîtrise de son agenda technologique
- ▶ véritable mise en concurrence des fournisseurs

Parmi ces acteurs, on retrouve le cas particulier des *administrations publiques*, qu'on traitera à part plus bas.

Part VI

Modèles économiques

Economie du Logiciel Libre

- ▶ les acteurs en jeu, et leur motivations pour le Logiciel Libre
- ▶ *des* modèles économiques variés
- ▶ la commande publique

Quelques réflexions générales sur les motivations

il y a différent types d'acteurs en jeu

Logiciel libre: *les acteurs*

l'utilisateur passif :

- + logiciel presque gratuit
- + *pérennité* de la solution (mutualisation possible)
- + pas d'emprisonnement propriétaire
- coût de sortie parfois élevé (un hasard?)

Logiciel libre: *les acteurs*

Sociétés de service :

- ▶ demande de service (utilisateurs actifs et passifs)
- ▶ espace économique de *proximité*
- ▶ meilleure connaissance du produit
- ▶ *marge* et/ou *marché* plus importants
- si le client n'est pas démuné, on peut être mis en concurrence sur la base des *compétences*, et pas des accords d'exclusivité: on peut se retrouver dans un cadre de libre marché!

La question du modèle économique

la cle pour comprendre

Vaste "littérature" sur les modèles économiques du Libre. Repères.

- ▶ GNU Manifesto: La vision de Richard Stallman, circa 1985
- ▶ Chris Hecker: Setting up shop, 2000
- ▶ John Koenig: Seven open source business strategies for competitive advantage, 2004
- ▶ Gasperoni, Comar: Open Source in Dependable Systems
- ▶ Livre blanc Aful, 2007
- ▶ Livre blanc April, 2007

Livre blanc de l'April

Centré sur l'analyse de quelque dizaines de sociétés, segmentées selon leur chiffre d'affaires, et classées sur deux dimensions:

- ▶ service/édition
- ▶ commodity/métier

Quelques éléments qui ressortent:

- ▶ beaucoup de petits acteurs sur le créneau service-commodity
- ▶ les grands sont sur édition-métier

Logiciel libre: *les acteurs*

le développeur du logiciel:

non spécifiques:

- ▶ plus grande valeur du programmeur et de ses produits,
- ▶ marketing démultiplié, création de demande de services,
- ▶ pénétration des marchés par la base technologique
- ▶ popularisation d'un standard (TCP/IP)

spécifiques:

- ▶ recrutement de compétences de haut niveau (la "communauté")
- ▶ mutualisation des coûts off business core (gcc, systèmes embarqués, téléphonie, etc.)
- ▶ valorisation de l'individu (propre à l'informatique)
- ▶ développement à la demande adapté à petites software houses (plus sur les modèles plus avant)

Economie

Quelques repères

- ▶ en 2012, la filière Logiciel Libre française pèse plus de 2,5 milliards d'euros (PAC)
- ▶ la France a des belles "success stories":
 - ▶ Videolan : logiciel multimédia plebiscité
 - ▶ JBoss: racheté 400M\$ par RedHat en 2006
- ▶ RedHat, un milliard de dollars de chiffre d'affaires en 2010!

Question naturelle

Une économie du *gratuit*?

Livre blanc de l'Aful

- services* : récurrents (support, souscription, certification, prestations de services divers)
- mutualisations* : user-driven (développements au forfait, ecosystème, reverse bounties)
- segmentation verticale* : base libre avec couche propriétaire a valeur ajoutée (box, extensions, outils)
- licences* : segmentation temporelle (biodégradables); segmentation spatiale (multi-licensing)
- apports indirectes* : marchés dérivés autour des projets libres (merchandising, ads, donations, etc.)

Role des acteurs

- ▶ auteur
- ▶ éditeur
- ▶ formation
- ▶ support
- ▶ conseil
- ▶ SSII
- ▶ intégrateur

Un point commun

Toutes ces analyses partent, explicitement ou implicitement, de l'observation que la voie de la facturation à l'unité (licences) est à priori barrée

Il faut donc trouver *autre chose*... mais cet autre chose peut prendre bien de formes différentes, et si on veut comprendre, il faut une cle de lecture.

Logiciel Libre : les bases des modèles économiques

Pour comprendre, il faut revenir aux définitions de base.

Rappel:

économie : l'étude de comment une société choisit d'utiliser des ressources *limitées* pour produire, échanger et consommer des biens et services. Ruffin, Gregory, "Principles of Economics", 1990

Sans rareté de ressources, pas d'économie.

logiciel propriétaire: logique d'offre

se fonde sur la "rarété" artificielle des copies d'un logiciel ...

logiciel libre: logique de demande

supprime la "rarété" des copies, et révèle les autres ressources "rares"

Economie du logiciel: licences vs. service

modèle centré sur les licences : profit non proportionnel au travail, peu ou pas d'emplois

modèle centré sur les services : tendance naturelle des grandes sociétés (IBM, Oracle etc.), profit proportionnel au travail, à la limite, le Logiciel Libre

	Income	Profit	%	Employees	P/E ⁵⁴	E/P ⁵⁵
IBM	81,667 M\$	6,328 M\$	7	290.000	21820 \$	45
Oracle	7,143 M\$	955 M\$	13	40.000	23875 \$	41
Microsoft	20,000 M\$	8,000 M\$	40	29.000	275000 \$	3

⁵⁴"taxe" collecté par employé

⁵⁵employés à *votre* service

Ressources rares : l'infrastructure

Exemple: les forges, comme SourceForge ou GitHub, qui fournissent un environnement de développement complet. Elles peuvent vendre des services à valeur ajoutée en valorisant une infrastructure largement connue.

Un cadre général

la cle pour comprendre

Ressources rares : les compétences

C'est la base du modèle du *service* proné par Stallman depuis 1985. Beaucoup de consultants individuels font du service. Plusieurs sociétés de service en Logiciel Libre (SS2L) en France se basent sur ce modèle: Smile (la plus grande), Linagora, Alterway, en sont des exemples.

Evolution de la situation entre 2002 et 2006

	Données 2002					
	Income	Profit	%	Employees	P/E	E/P
IBM	81,667 M\$	6,328 M\$	7	290.000	21820 \$	45
Oracle	7,143 M\$	955 M\$	13	40.000	23875 \$	41
Microsoft	20,000 M\$	8,000 M\$	40	29.000	275000 \$	3

	Données 2006					
	Income	Profit	%	Employees	P/E	E/P
IBM	91.000 M\$	8.000 M\$	9	366.500	21830 \$	46
Oracle	14.000 M\$	3.300 M\$	24	56.000	58930 \$	17
Microsoft	44.282 M\$	12.600 M\$	28	71.000	177460 \$	6

Ressources rares : l'industrialisation

Un exemple essentiel: les distributions GNU/Linux, comme RedHat (1Md de dollars en 2010).
Un autre exemple: AdaCore.

Ressources rares : la communauté

Un logiciel libre sans communauté a rarement beaucoup de valeur. L'accès à une communauté (decision, interaction, vision sur l'évolution technologique) est une ressource rare. Des développeurs phare sont recrutés par des entreprises en raison de leur compétences. Des marchés sont gagnés par des entreprises grace au fait qu'elles ont des développeurs phare.

Le cas des *éditeurs* de logiciels libres

Une typologie particulière d'acteurs se développe beaucoup dans le monde du logiciel libre, celle des éditeurs de logiciel libre. On parle d'éditeurs quand une partie significative des prérogatives de la communautés (droits de commit, développeurs phare, decision sur l'évolution du logiciel) se retrouve concentrée essentiellement dans les mains d'un seul acteur. Dans ce cas, l'éditeur devient un référent incontournable des grands utilisateurs du logiciel. Exemples: RedHat (qui a racheté JBoss), MySQL (avant rachat par Oracle), Nuxeo, Nexedi, Talend, Qualixo, Engineering (semaine prochaine), AdaCore, etc. etc.

Les stratégies mixtes

jouer sur les deux tableaux

Licences différents pour clients différents

On peut, tout en gardant la licence libre, séparer les groupes des utilisateurs en réservant une deuxième licence différente à ceux qui payent.

OCaml : les membres du consortium peuvent utiliser une licence différente à la QPL

Qt/Trolltech : qui achète la licence commerciale peut utiliser les librairies sans GPL

Mysql : qui achète la licence commerciale peut embarquer les pilotes et la BD sans GPL

Ressources rares: la communauté, bis

Tout le monde peut faire une copie des sources d'un projet libre. Maintenir une version séparée d'un logiciel peut couter très cher (il faut maintenir des centaines de patch!) Très peu de personnes ont le droit de commiter dans la branche principale d'un logiciel libre!

Un temps d'apprentissage

Cette rareté n'est pas toujours perçue clairement: jouer le *free rider* est une stratégie qui peut rapporter à court terme! Mais cela comporte des risques lourds à moyen terme: on a un exemple culte de ce fait en Ile de France (on en parle off the records). Une fois le message compri, les flux financiers entre revendeurs/intégrateurs, et les vrais développeurs des logiciels libres s'établissent naturellement.

Une autre ressource rare: le droit de changer la licence

Nous avons vu que qui detient le droit sur un logiciel peut l'offrir sous des licences différentes à des acteurs différents. Cela fait de cet acteur un partenaire incontournable pour toute une classe de clients, et ouvre la voie à des modèles économiques qui génèrent des revenus à partir de qui veut des versions "moins libres". Voyons quelques exemples.

Versions et licences différentes: séparation temporelle

On peut séparer les clients *temporellement*:

- ▶ par des releases à distance de temps (AdaCore/GNAT)
- ▶ par des licences "biodégradables" Ex: Precision Insight, Ghostscript

Exemple: Aladdin Ghostscript

From Ghostscript's README

What about commercial use?

GNU Ghostscript may not be incorporated into commercial products which forbid copying or for which customers cannot obtain source code for no more than the cost of reproduction, although it may be distributed ("aggregated") with commercial products; Aladdin Ghostscript may not be incorporated into commercial products at all, and may be distributed commercially only under extremely limited circumstances.

However, Ghostscript is also available for commercial licensing, which in addition to the right to incorporate Ghostscript into commercial products includes support, a limited warranty, high-quality fonts, and other benefits.

For more information about commercial licensing of Ghostscript, please contact Aladdin Enterprises' commercial distribution partner, the only entity legally authorized to distribute Ghostscript per se on any terms other than the GNU or Aladdin free licenses [...]

Le cas du *code drop*

Un mot à part mérite la pratique du *code drop*, qui est pratiqué régulièrement par Google:

- ▶ on release le code de temps à autre, et pas forcément à tous en même temps
- ▶ on ne rend publique aucune partie du processus de développement

Cela montre bien que la licence à elle seule ne garantit pas l'ouverture

2006: Freemium est né

Un VC qui parle de Skype, Flickr, etc...

« Donnez votre produit ou service gratuitement, éventuellement avec une offre publicitaire, acquérez une base de clientèle avec du bouche-à-oreille, des réseaux de référence, du search marketing organique, etc., ensuite offrez des offres Premium à cette même base de clients. » Fred Wilson, 2006

Il cherche un nom pour ce modèle économique, et un des commentateurs lui propose Freemium = Free+Premium.

L'état est un client spécial

Licences différentes: séparation spatiale

On peut choisir une licence libre restrictive, qui permet de séparer clairement les groupes des utilisateurs, pour se garantir une exclusivité d'exploitation commerciale.

complication du fork : Trolltech v 2 , OCam!

exclusion d'usage commercial : Trolltech v 1 [Free Edition/Professional Edition] (non accepté par FSF/OSI)

Les stratégies cyniques

sur la ligne jaune ou au delà

Freemium et Open Core

Cette idée se repand dans plusieurs logiciels Open Source (en particulier américains).

On sépare la base de code en plusieurs parties:

- ▶ un noyau libre (Open Core), qui donne des fonctionnalités de base, dont le rôle est d'attirer des utilisateurs
- ▶ des modules propriétaires, qui sont indispensables à un usage professionnel, et vont engendrer des revenus

Exemples: Zimbra, SugarCRM

L'administration publique est un utilisateur privilégié

L'État *n'est pas* une "entreprise" comme les autres!

Quelques exigences *spécifiques* de l'administration

- ▶ **archivage à très long terme, intégrité des données** (état civil, impôts...)
- ▶ **sécurité** (respect de la vie privée, confidentialité, protection des informations sensibles, défense, ...)
- ▶ **coût modéré**
- ▶ **identification sûre du citoyen**
- ▶ **devoir de transparence**
tout cela doit se faire dans le cadre d'un
- ▶ oecuménisme technologique

Interopérabilité, standards et qualité

Des grandes administrations demandent des solutions interopérables, et donc les veulent

- ▶ conformes à des standards normalisés (ISO, RFC, etc.)
- ▶ garanties contre la non conformité sur tout le spectre
- ▶ avec livraison du code source. . .

Il est très difficile de répondre à ces demandes avec des offres propriétaires. . .

Le même type d'exigences commence à se généraliser pour le logiciel critique et embarqué.

Quelques exemples

Off the record.

Vision à long terme et logiciels à écrire

L'administration publique est un client qui a des besoins spécifiques (par exemple, la gestion des cimetières) et qui a (ou devrait avoir) une vision à long terme.

Il est donc une grande source d'activité économique pour les logiciels libres *non encore écrits*.

Un point de vue intéressant, inspiré par les besoins de l'administration publique, se trouve exposé dans le livre *Economie du Logiciel Libre* de François Elie (qui a dit « Un logiciel libre est gratuit une fois qu'il a été payé »).