

La viabilité de la GPL

Par Philippe Frémy

La nouvelle du changement de licence de TuxRacer [3] m'a poussé à mettre par écrit quelques petites réflexions sur la GPL, le logiciel libre et l'économie, que je voudrais partager avec vous.

I - Barrière à l'entrée

Dans l'économie normale, si on prend un marché donné où il y a un certain nombre d'acteurs, tout nouvel entrant rencontre une barrière à l'entrée.

Les anciens possèdent une technologie mature, de l'expérience et une base de clients, ce qui leur permet de vendre respectivement un produit, du service et des partenariats.

C'est la difficulté à obtenir la même technologie, la même expérience ou les mêmes clients qui détermine la barrière à l'entrée.

La barrière à l'entrée est une chose importante car elle donne une stabilité à l'écosystème du marché. Pour être rentable une entreprise a en effet besoin de pouvoir affirmer qu'elle possèdera une part de marché sur une période de quelques années. Pour une société technologique de taille moyenne, 50 à 200 personnes, 6 ou 7 ans constituent une période de rentabilité raisonnable.

Si la barrière d'entrée est élevée, cela conduit à un monopole ou à un oligopole (très peu d'acteurs sur un marché). Les quelques acteurs dominants se partagent le marché, qui si il est large, peut-être très juteux.

Si la barrière à l'entrée est très élevée, cela devient difficile de s'installer, même pour les leaders. C'est le cas par exemple de l'industrie des consoles. Le coût de développement d'un jeu et d'une console est tellement élevé que même Sony et Nintendo ne sont jamais sûrs de leurs opérations. Sega a renoncé et Nintendo est en train de le faire.

Le marché de la téléphonie mobile 3^e génération (UMTS) est un autre exemple de marché nouveau avec une barrière d'entrée très élevée. Les coûts de licences et infrastructures sont tellement importants que la rentabilité n'est pas assurée, même pour un oligopole.

Notons que dans ces deux cas (UMTS et consoles), on vend à des particuliers. C'est un marché particulièrement difficile car le comportement des particuliers est difficile à prévoir. Achètera, achètera pas ?

Si la barrière à l'entrée est très faible, on obtient un marché volatile et instable. Il y a beaucoup d'acteurs qui essaient de rentrer en permanence sur le marché. Du coup, le marché devient extrêmement divisé et perturbé, la part de chaque acteur devient très petite et aucun des acteurs n'est rentable.

Beaucoup de sociétés disparaissent. Ceux qui arrivent à survivre aux phases initiales restent. Au fur à mesure que les autres tombent, ceux-là stabilisent leurs parts de marché et leur position dominante. Au final, tout nouvel entrant aura du mal à prendre une part de marché aux gros qui ont réussi à survivre. Il y a donc, après la stabilisation du marché, une barrière d'entrée qui s'est relevée.

C'est ce qu'on a vu pour les dot-com. Un marché en création, avec une barrière d'entrée extrêmement faible (un site web) et des bénéfiques potentiels élevés. Des centaines d'acteurs se sont rués naïvement dessus et ça a donné ce qu'on a vu. En fait, on s'aperçoit que la barrière d'entrée n'est pas si faible puisqu'il faut fournir un réel service,

puis que les revenus eux sont très faibles (marge sur des achats, publicité, ça ramène pas lourd) et que donc c'est un marché difficile. Un type intelligent disait qu'il y avait de la place pour 10 acteurs dans la publicité en ligne. C'est ce qu'on a aujourd'hui, maintenant que le marché s'est stabilisé: yahoo, voila, boursorama, spray, ... La barrière d'entrée est très élevée maintenant, parce que les acteurs sont établis et que les marges sont très faibles. On note encore que là aussi, c'est un business qui vend aux particuliers, donc difficile à prévoir.

Ce que je veux faire comprendre, c'est que la barrière à l'entrée sur un marché est quelque chose de capital pour le succès d'une entreprise. S'il n'y a pas de barrière à l'entrée suffisamment élevée, les entreprises n'arrivent pas à protéger leurs acquis et ne rentabilisent donc pas leurs investissements. Elles disparaissent, l'économie devient morose, on licencie, il y a des plans sociaux, bref, ça va mal.

II - Business model

Le business model, qui doit bien s'appeler quelque chose comme "modèle d'affaire" en français bien de chez nous, c'est le truc qui décrit comment vous gagnez de l'argent. En général, 4 paramètres interviennent dans un business model:

1. Ce que vous vendez: produit ou service
2. Combien ça coûte à faire: un logiciel à la con, ça prend 6 mois de dev à deux. Un gros logiciel, c'est 3 ans, voire plus, pour une équipe de 10 personnes. C'est ce qu'on retrouve pour de très gros logiciels d'entreprise ou pour les jeux modernes.
3. Combien vous en vendez: évidemment, c'est important à savoir.
4. A qui vous le vendez :
 - Entreprises: c'est un marché prévisible: elles achètent si ça leur fait gagner du temps ou de l'argent (ce qui revient au même). Elles ont beaucoup d'argent donc on peut éventuellement vendre très cher à un petit nombre de clients privilégiés.
 - Particuliers: c'est un marché relativement imprévisible. Le particulier achète si ça lui plaît et si ça ne coûte pas trop cher.

Si un outil est indispensable à une entreprise, elle est prête à le payer cher. Donc on peut mettre une grosse équipe pendant 3 ans sur un logiciel qu'on est sûr ensuite de vendre plusieurs centaines de milliers de francs ou à plusieurs centaines de milliers d'exemplaires. Pour être sûr de le vendre, il suffit de prouver, à l'entreprise qui l'achète, qu'elle va gagner de l'argent avec.

L'avantage du marché des particuliers, c'est qu'il y en a beaucoup. On peut donc vendre un logiciel pas cher à beaucoup d'unités pour rentabiliser un gros et long développement (typiquement, un gros jeu). Mais le marché des particuliers est assorti d'une incertitude, on n'est jamais sûr qu'ils vont acheter.

Dans les deux cas que je viens de citer (gros développement), la valeur ajoutée est sur la technologie. C'est une grosse techno qui coûte cher qu'on vend. Celle-ci constitue la barrière avec le concurrent qui devra passer beaucoup de temps pour en avoir une similaire.

On peut aussi vendre des logiciels à faible technologie, mais dans ce cas, il n'y a pas de barrière technologique. Si on veut empêcher un concurrent de prendre toutes les parts du marché, il faut trouver un autre type de barrière: un très bon contact avec le client, une relation privilégiée, un partenariat, des services spécifiques, une image, un bon réseau de distribution.

III - Logiciel libre

Maintenant, regardons quels Business Model sont viables pour le logiciel libre.

Pour gagner de l'argent, une boîte doit, d'une part, se protéger de ses concurrents, et d'autre part, s'en différencier.

Le gros problème lié à la commercialisation du logiciel libre (sous GPL), c'est que quand vous livrez toutes vos sources, rien n'interdit à votre concurrent de reprendre votre outil à son compte et de le vendre aussi. Votre technologie n'est pas protégée. C'est agréable idéologiquement d'avoir un logiciel en GPL, mais ça peut vous faire perdre de l'argent si vous essayez de le vendre.

La barrière à l'entrée que constitue l'accès à la technologie est éliminée. Voire même dans certains cas, elle est négative. Imaginons un gros développement technologique qui est livré sous GPL. Celui-ci coûte cher à son auteur. Un nouvel entrant arrive. Tout ce qu'il a à faire, c'est récupérer le développement fait gentiment par son concurrent, et le commercialiser. Comme il économise sur les coûts de développement, il peut se permettre de dépenser plus sur la commercialisation. Si le développeur original n'arrive pas à capitaliser assez sur son expérience, il se sera fait prendre injustement le fruit de ses efforts.

On a vu ça avec la naissance de Mandrake. Au départ, Mandrake, ce n'était qu'une Redhat pour laquelle on avait package KDE. Par la suite, Mandrake a ajouté sa propre couche par dessus la Redhat, mais il n'en reste pas moins que Mandrake a récupéré tout le travail de Redhat gratuitement.

Etant donné que les trois distributions se partagent aujourd'hui grosso modo le marché à parts égales, on peut à peu près estimer de façon très simpliste que Redhat a perdu 50% de son marché.

Je vois souvent des gens critiquer le modèle du logiciel commercial en prônant le tout-GPL. Mais il faut bien se rendre compte que dans bien des cas, le modèle du tout-GPL n'est pas viable économiquement.

Ce qu'il faut voir, c'est d'où provient le revenu d'une société :

1. la vente d'un produit logiciel
2. la vente d'un service basé sur un logiciel
3. la vente d'une bibliothèque logicielle

Avec des développements propriétaires, les 3 modèles peuvent être viables.

Attention, le modèle 1, c'est uniquement la vente d'un produit fini sans service. Sinon, c'est le modèle 2.

Avec du logiciel GPL :

- 1 n'est pas viable. Votre concurrent n'aura qu'à récupérer tout votre développement pour le vendre à votre place. Vous ne pouvez pas capitaliser sur votre expérience puisqu'elle est déjà incluse dans votre produit. Vous pouvez uniquement jouer sur votre image et votre relation client, ce qui est très risqué comme unique moyen de sécuriser une position sur un marché. On est dans le cas d'une barrière à l'entrée négative, c'est à dire qu'un nouvel entrant intelligent est en meilleure position que vous, puisqu'il récupère vos développements sans les payer. Comme exemple de produit, on peut citer les petits shareware, comme WinZip. Si les sources des WinZip sont disponibles sous GPL,

personne n'achètera la licence commerciale qui procure les revenus à son auteur.

- 2 est le modèle souvent prôné dans le logiciel libre. Ce sont ceux qui ont développé le logiciel libre qui vendent du service dessus. Ils peuvent capitaliser sur leur expérience et n'ont qu'à développer leur relation client. En général, ça marche bien si on trouve les clients. Cependant, un concurrent peut toujours arriver, acquérir de l'expérience et revendre le même service. C'est ce qu'il semble se passer dans l'affaire MySQL. Si le marché est trop petit, le simple fait de le segmenter peut suffire à tuer les deux acteurs.

- 3 est en général viable. Avec une bibliothèque logicielle, toute entreprise raisonnable achète du support. Comme votre bibliothèque est en GPL, l'entreprise utilisatrice devra mettre son produit en GPL et si c'est un produit qu'elle vend, elle ne voudra probablement pas (parce qu'elle ne veut pas se retrouver dans le cas 1). Donc elle achètera une licence commerciale qui coûte cher et vous gagnerez de l'argent. C'est le modèle choisi par Trolltech pour Qt et par l'ex-Cygnus pour Cygwin [1]. Le risque est que votre bibliothèque soit utilisée pour un produit interne, auquel cas la GPL peut convenir et vos clients ne vous achètent rien.

Si vous vendez votre bibliothèque en LGPL, vous vous retrouvez dans le cas 1 (donc très risqué), avec le seul espoir de vendre du service dessus. Si vous vous retrouvez à vendre du service sur votre bibliothèque, c'est d'une part qu'elle est difficile à utiliser donc pas géniale, d'autre part que vous êtes dans le cas 2.

Une petite remarque tout de même. Je parle de modèle viable et non viable. On va sûrement m'exhiber des sociétés vivant sur un modèle 1, alors que je viens de le décrire comme non viable. Le terme viable ici est à prendre dans le sens "viable à long terme", c'est-à-dire "qui ne risque pas de disparaître". On peut faire de l'argent en étant dans le modèle 1 pendant des années sans avoir de soucis. Mais le jour où quelqu'un se rend compte qu'il peut se faire de l'argent facilement avec votre technologie, vous êtes grillés. Le risque est important. Notamment, si vous commencez à gagner beaucoup d'argent, des petits malins vont vouloir vous en prendre et vous devriez éviter de leur donner votre technologie.

Il y a un cas qui n'est pas traité dans cette brève présentation, c'est quand une entreprise récupère un logiciel GPL développé bénévolement, et le commercialise. Certes, elle peut s'accommoder du modèle GPL, mais si le logiciel est un produit, elle se retrouve dans la situation 1. Comme toute entreprise n'aime pas prendre de risques, même quand elle n'a pas payé le développement, elle va plutôt essayer de récupérer les développeurs sous son aile de façon à pouvoir changer la licence. C'est ce qui s'est passé pour TuxRacer. C'est ce qui est en train de se passer pour Quanta (et ça ne fait pas la joie de tout le monde, cf <http://dot.kde.org/995743381/995776534/>) et pour deux ou trois autres produits de TheKompany.

Du point de vue d'un développeur libre, on est rarement content quand un "fork" se produit, c'est-à-dire qu'un autre projet démarre à partir de votre projet. Mais quand on est une société et qu'un concurrent récupère votre produit principal, c'est un risque de faillite donc c'est très grave. Tout ce qui permet de s'en prémunir est bon.

IV - Conclusions

C'est pour ça que je pense que si une société veut vendre TuxRacer sans prendre de risques inutiles, elle ne doit pas le relancer sous GPL. Certains ont imaginé qu'on pourrait garder le moteur en GPL et ne vendre que les graphiques. Mais le risque reste encore trop important. Une société concurrente peut très bien payer une équipe de 3 personnes pendant 3 mois pour refaire un jeu complet à partir du moteur et vous griller

sur votre marché.

Donc il faut tous passer une licence commerciale.

Si ID Software avait mis son moteur 3D sous GPL, ils seraient beaucoup moins riches à l'heure actuelle. Peut-être même qu'ils auraient coulé, à cause du manque à gagner.

Il y a à Paris une boîte dont j'ai oublié le nom, qui est en train de développer un jeu dont le moteur sera sous GPL mais pas les graphismes ou les clients (je ne me souviens plus très bien des détails). Je leur envoie tous mes voeux de réussite, mais je crois qu'ils se placent dans une position très dangereuse d'un point de vue de rentabilité de leur société.

Les implications de cette petite démonstration vont loin. Il n'y aura jamais, à mon avis, de jeux sous Linux, développés par une société, en GPL. Un jeu, ça coûte très très cher à développer (une très bonne équipe de 10 personnes pendant 3 ans, à plein temps) et une société ne peut se permettre de risquer un tel investissement. Il n'y a que deux types de jeu sous Linux pour l'instant. Ceux qui sont sous une licence commerciale payante et ceux, purement GPL, qui sont développés par des bénévoles. Ces derniers étant réalisés par des personnes non payées, sur leur temps libre, ils ont beaucoup de mal à atteindre, en qualité et en diversité, les logiciels commerciaux. Le travail d'une équipe de 10 personnes pendant 3 ans, c'est dur à récupérer sur ses week-end et ses soirées [2].

Personnellement, je rêve depuis longtemps d'être payé pour développer un logiciel sous GPL. La découverte de ce que je viens de vous exposer, m'a fichu un coup. Alors, le GPL (à part pour mettre dans les voitures), ça ne sert à rien ? En tout cas, ça ne peut pas rapporter d'argent?

Ce qui est sûr, c'est que vendre un produit et le développer sous GPL est incompatible. Ce qui reste possible, c'est vendre une bibliothèque sous GPL (bravo Trolltech!) ou vendre des services bases sur des logiciels GPL (vive Zope!).

Pour une société, il est également très intéressant d'utiliser des logiciels GPL si ils fournissent les mêmes fonctionnalités que des logiciels payants, ça coûte moins cher. Et si les logiciels ne sont pas tout à fait à niveau, ça peut même être super rentable de payer les développeurs pour qu'ils rajoutent ce qu'il faut, plutôt que d'acheter ce qui manque. En fait, c'est un des modèles les plus intéressants pour une société. Si elle a un besoin qu'elle pourrait couvrir par un développement, qui n'a rien à voir avec ce dont elle tire son revenu, elle a tout intérêt à s'associer avec d'autres qui ont le même besoin et à payer une petite équipe pour faire ça en GPL. C'est cela qui donne naissance à Apache et à d'autres logiciels serveurs très utilisés.

Et puis si une société montée par des fans de logiciel libre, vend un produit sous licence commerciale qui marche bien, ça peut lui permettre de faire du logiciel libre -par ailleurs-.

Je vous invite tous à réfléchir à la démarche de Suse. Oui, leur outil d'installation n'est pas sous GPL. Oui, il est interdit de copier une Suse. Oui, elle n'est pas disponible gratos sur le net pour l'install. Mais du coup, il n'est pas possible de faire à Suse ce que Mandrake a fait à Redhat. Suse possède réellement sa distribution. Et certes, elle ne donne pas ses outils de config. Mais elle paye un nombre impressionnant de développeurs de logiciels libres : deux développeurs de XFree, tout le projet alsa, 4 ou 5 développeurs sur le noyau Linux, 5 développeurs pour KDE, 1 développeur pour Gnome, des développeurs sur la libc. Donc quand on dit que Suse ne redonne pas à la communauté, je ne suis pas du tout d'accord. Payer des développeurs, ça me paraît autrement plus productif et utile que de mettre HardDrake sous GPL.

V - Notes

1: Et oui, peu de gens le savent mais il est écrit je ne sais plus où, que comme la libc patchée et utilisée par cygwin est en GPL, tout programme produit avec cygwin gcc est en GPL . Pour faire du close-source avec, vous devez payez une licence de 10000\$!

2: Cela dit, on voit des jeux magnifiques arriver en GPL. Comme d'une part, l'industrie du jeu peine à apporter de nouveaux concepts, et d'autre part, les jeux en GPL peuvent se réutiliser les uns et les autres, on va peut-être finir par avoir un jeu de qualité commerciale sous GPL. Peut-être. Quand on aura des scénaristes notamment.

3: Tux Racer 1.0 ne sera pas libre, LinuxFr.Org, 06 août 2001.

VI - Sommaire

I - Barrière à l'entrée	1
II - Business Model	2
III - Logiciel libre	3
IV - Conclusions	4
V - Notes	6
VI - Sommaire	6

Cet article a fait l'objet d'une publication initiale et d'un débat sur le site LinuxFr.Org le 14 août 2001 (URL : <http://linuxfr.org/2001/08/14/4558.html>).

Auteur: Philippe Fremy (2001)

Publication: Robert Viseur @ LogicielLibre.Net (2004)
