

L'analyse stratégique : un outil pour les communautés libres

Ce document se propose d'étudier l'application de l'analyse stratégique à des communautés libres existantes ou à construire, et d'en montrer l'intérêt. Deux approches de l'analyse stratégique seront ici explorées : l'analyse de Porter et l'analyse du Boston Consulting Group.

L'objectif d'une communauté libre, outre le plaisir de programmer, est aussi d'offrir un produit utile et alternatif aux solutions propriétaires. Comme une entreprise commerciale, elle est confrontée à la difficulté de faire correspondre son offre (le logiciel libre) et une demande. Il s'agit donc d'un problème de marketing.

Que ce marketing ne concerne pas un produit commercial n'a pas d'importance. Kotler et Dubois (2000, p40) définissent en effet le marketing comme *le mécanisme économique et social par lequel par lequel individus et groupes satisfont leurs besoins et désirs au moyen de la création et de l'échange de produits et autres entités de valeur pour autrui*. Par groupes, nous pouvons entendre communautés libres et par entités de valeur, logiciels libres.

Le marketing exige une bonne connaissance de l'environnement. Il faut connaître pour agir (Dayan, 1985). L'analyse stratégique fournit des outils permettant d'atteindre cette connaissance.

L'analyse de Porter

L'analyse de Porter présente l'intérêt de permettre une vue plus large de la dynamique concurrentielle. Elle prend en compte cinq forces, appelées forces de Porter, qui sont : les concurrents, les fournisseurs, les clients, les barrières et les substituts (Porter & Dubois, p247). Appliquée à un projet de logiciel libre, l'analyse de Porter permet :

- d'apporter des éléments en vue d'affiner les contours du projet ;
- de maximiser les chances de réussite d'un projet, par une meilleure connaissance de son environnement.

Elle peut également conduire à l'arrêt d'un projet, lorsque son intérêt n'est pas confirmé par une étude de l'existant. Une fois le projet lancé, elle offre un cadre clair pour de la veille (technologique, concurrentielle, etc).

Les concurrents

Pour une communauté libre, la concurrence revêt une double forme :

- les autres projets libres ;
- les éditeurs de logiciels propriétaires.

La tendance à la concentration que l'on connaît dans le monde industriel se retrouve dans le logiciel libre. Selon Eric Raymond (1998), *certaines projets à succès deviennent des « tueurs de concurrence »*. *Personne ne voudra se tenir près d'eux car se mesurer à une base déjà établie sera une tâche trop dure. Les gens qui en revanche feront des efforts distincts finiront plutôt par contribuer à l'extension de ces projets à succès. L'exemple classique du « tueur de concurrence » est GNU Emacs. Ses variantes remplissent la niche écologique des éditeurs entièrement programmables, à tel point que personne n'a jamais vraiment essayé de créer un programme très différent dans ce domaine depuis les années 80. Au lieu de cela les gens écrivent des modules pour Emacs. Il convient donc d'identifier ce type de projet au plus vite afin, soit de s'y rattacher, soit de chercher de réelles voies de différenciation.*

Malgré la concurrence d'un projet *leader*, des communautés développant un produit de niche peuvent en effet subsister. Le serveur HTTP propriétaire Zeus possède une part de marché marginale par rapport au logiciel libre Apache : moins de 2% contre plus de 60% (Netcraft, 2003). Néanmoins, il s'impose dans le domaine

des serveurs HTTP à très haute performance, en autorisant de fortes montées en charge (Six, 2002). Il justifie par la même occasion son prix d'achat élevé.

Par ailleurs :

- L'ouverture du code garantit l'opérabilité d'autres logiciels avec le projet dominant. Les logiciels libres privilégient en outre les standards ouverts. Une domination d'Open Office, dont les fichiers de données se basent sur XML, n'aurait donc pas les mêmes répercussions que la domination de Microsoft Office.
- La contestation reste possible, et peut aboutir à la création de branches dissidentes (appelées « fork »). Par exemple, le projet BSD s'est progressivement scindé en trois projets distincts : FreeBSD (la plus répandue et la plus conviviale, qui privilégie les performances), OpenBSD (qui privilégie la sécurité) et la branche initiale, NetBSD (qui privilégie l'adaptation aux matériels).

Remarquons que si la diversité est un plus, la dispersion pénalise le mouvement dans son ensemble, car elle dilue les ressources (programmation, promotion, etc). La multiplication de projets de logiciels de gestion immatures (GNU Enterprise, ERP5, Compiere, etc) ou de navigateurs alternatifs (Mozilla, Galeon, Phoenix, Epiphany, Konqueror, etc) peut surprendre, même si une différenciation existe parfois bel et bien. Phoenix est par exemple une version allégée de Mozilla, complet mais lourd, dont elle partage le moteur de rendu, Gecko. Une concentration des efforts apporterait néanmoins un contrepois appréciable face aux grands éditeurs propriétaires (SAP, Microsoft, etc).

Les éditeurs de logiciels propriétaires peuvent être des concurrents, en particulier dans deux cas :

- le logiciel propriétaire est gratuit ;
- le logiciel propriétaire est payant, mais est soumis à une pression concurrentielle importante.

Dans le premier cas, il sera difficile de déloger le logiciel *freeware*, à moins d'apporter une innovation déterminante à l'utilisateur final.

Cette difficulté est notamment rencontrée par Mozilla / Netscape. Microsoft Internet Explorer bénéficie d'une supériorité écrasante sur le marché des navigateurs Internet, avec plus de 90% de parts de marché. Mozilla souffre de trois facteurs pénalisants :

- Microsoft Internet Explorer est fourni d'office avec Microsoft Windows.
- Malgré d'indéniables qualités (dont la navigation par onglet), Mozilla n'apporte pas suffisamment en terme de fonctionnalité ou d'ergonomie, que pour justifier l'effort de changer de logiciel.
- La première version de Netscape / Mozilla (Netscape 6) pour Windows souffrait d'instabilité et était gourmande en ressource. Elle a donc souffert d'une image négative.

Au début de l'année 2000 (Descombes, 2002), la part mondiale de marché du navigateur de Netscape était de 18 %. Au milieu de l'année 2001, elle descend vers les 12 %. Cette dernière baisse coïncide avec l'apparition de la version 6 du navigateur de Microsoft. La chute va alors s'accélérer de manière exponentielle. Une étude sur l'année 2001 crédite Navigator de 4,5% de parts de marché, contre 94,2% à Explorer. Selon OneStat (Delameilleure, 2002), en décembre 2002, Netscape occupait environ 3% du marché contre 1,1% à Mozilla. Les versions 6 et 7 comptent respectivement pour moins de 1% et 0,6% (Xiti, 2002).

Dans le second cas, la richesse de l'offre commerciale et la concurrence jouera son rôle de moteur de l'innovation et d'ajustement des prix (Smets, 1998), conduisant à diminuer :

- la motivation à contribuer,
- l'intérêt de passer à une solution libre, dans un premier temps moins riche et intégrée que son équivalent propriétaire.

C'est notamment la cas dans le domaine du multimédia (Smets, 1998).

Les fournisseurs

Pour une communauté libre, la notion de fournisseur revêt une double forme :

- les autres projets libres ;
- les éditeurs de logiciels libres et non libres.

Peuvent être considérées comme fournitures : l'infrastructure matérielle et logicielle de développement coopératif (comme SourceForge), les outils logiciels de développement (compilateur comme GCC, débogueur comme DDD, etc), les bibliothèques (comme Qt, GTK, etc), les outils d'édition (comme Open Office, LaTeX, etc), etc.

Une question concerne la présence ou non d'outils libres pour le développement de l'application libre. Plusieurs polémiques ont éclaté à ce sujet : utilisation du logiciel propriétaire Bitkeeper par les développeur du noyau Linux (LinuxFr, 2003a ; LinuxFr, 2003b), utilisation de la bibliothèque Qt sous licence propriétaire dans les première version de KDE, etc... Ce dernier exemple est à l'origine de la création du projet GNOME, concurrent de KDE purement libre (O'Reilly, 2003a).

Les clients

Une communauté libre possède deux types de clients :

- les contributeurs qui fixent les bogues, ajoutent des fonctionnalités, participent au support technique,...
- les simples utilisateurs.

Les enjeux sont différents :

- Les contributeurs sont nécessaires pour la fiabilisation et l'amélioration (innovation) du logiciel.
- Les utilisateurs font la popularité du logiciel. Leur nombre amplifie le bouche-à-oreille et est susceptible d'attirer de nouveaux contributeurs.

Pour réussir, la communauté doit prendre en compte les divergences d'intérêt entre contributeurs et utilisateurs, ainsi que la propension des clients à contribuer.

Les contributeurs sont intéressés par le développement et ont une approche technique du logiciel. Le simple utilisateur cherche avant tout un lien de téléchargement intuitif vers un programme d'installation.

Les sites des communautés libres sont souvent compliqués : surcharge d'information sur la page de démarrage, multiplication des versions du logiciel à télécharger, ... L'utilisateur ne vice ne peut qu'être découragé par une telle complexité. Stephane Mariel (2003), consultant indépendant, affirme d'ailleurs : *Comment prétendre en effet être une alternative sérieuse au logiciel propriétaire, comment vouloir toucher le plus grand nombre quand dans le même temps aucun effort n'est fait pour comprendre l'utilisateur potentiel (osons l'appeler client). Ce mépris de l'individu lambda est un frein considérable à l'adoption du logiciel libre sorti de sa sphère de prédilection naturelle : nombre de projets ne communiquent pas hors de la communauté, et quand ils communiquent les informations indisponibles sont incompréhensibles par le non initié. Un exemple : combien d'utilisateurs potentiels d'OpenOffice ont-ils renoncé à télécharger cet outil devant la numérotation invraisemblable qui a prévalu avant la version 1.0 ? Ceci alors même que ces versions étaient tout à fait fonctionnelles...*

La différence se ressent d'autant mieux lorsque l'on regarde l'interface de sites commerciaux comme ceux de Windows ou mieux encore de Lycoris :

- L'interface est plus légère et aérée.
- L'identité d'entreprise est affirmée avec élégance et efficacité (Lycoris – *Linux is for everyone*). Le positionnement est clair.
- Le lien pour l'achat de la distribution Linux est facile à trouver.

Les barrières

Différents types de barrières peuvent être distinguées :

- les brevets logiciels,
- les licences des logiciels,
- les autres formes de propriété intellectuelle, comme la marque ;
- les spécificités de matériel,
- ...

Le logiciel Quicktime (Vande Walle, 2003) de Apple fait appel au brevet Sorenson. Aussi, il n'est pas possible à ce jour de concevoir librement une implémentation de Quicktime pour Linux. Des brevets d'Unisys et IBM (GNU, 2003) couvrent l'algorithme de compression LZW qui est utilisé dans la création des fichiers GIF. Ces brevets rendent impossible l'utilisation des logiciels libres pour générer un format GIF.

La marque peut également porter atteinte aux communautés libres. Il y a quelques temps, les Éditions Albert René ont ainsi porté plainte contre le site Mobilix (MOBILe unIX) qui d'après eux pouvait porter à confusion avec les personnages Asterix et Obélix. La communauté a réagit de manière efficace, en produisant une étude d'antériorité remarquablement complète.

Remarquons que la marque est aussi un outil au service de la communauté. Linux est ainsi une marque déposée par Linus Torvald, le créateur de Linux ; Debian, de Debian (Debian, 2003) ; etc. L'objectif est d'éviter une utilisation abusive d'un nom populaire. Le terme « Open Source » a ainsi été présenté comme marque, mais son dépôt a échoué (FSF, 2003a). Une certification *Open Source Software* a par contre été mise en place par l'Open Source Initiative (OSI).

Remarquons que la marque s'intègre dans un concept plus large, celui d'identité. L'identité d'une communauté (ou d'un logiciel) libre permet son identification et la consolidation d'une image autour de la marque. Open Office –*OpenOffice.org 1.0, The Open Source Office Suite*-, avec sa marque, sa signature, son logo et sa palette de couleur caractéristique bénéficie d'une

bonne identité, malheureusement déforcée par un site compliqué pour le novice.

La propriété intellectuelle au sens large soutient d'ailleurs le logiciel libre. La licence GPL inverse par exemple le droit le copyright pour garantir qu'un logiciel libre reste libre : c'est le principe du *copyleft* (FSF, 2003b).

Les substituts

Parmi les causes de substitution, nous pouvons citer :

- l'introduction de logiciels conviviaux et efficacement promotionnés ;
- l'introduction de nouvelles technologies (format de données comme XML, architecture comme J2EE ou .NET, etc) ;
- les changements de licences.

Comme expliqué *supra*, le navigateur Netscape Navigator, dans un premier temps, a été progressivement éliminé par Microsoft Internet Explorer, gratuit, intégré d'office dans Windows et offrant des prestations comparables.

Un exemple de substitution possible par une nouvelle architecture est donné par les concepts de noyau monolithique et du micro-noyau. Pour information, Linux est un système d'exploitation à noyau monolithique, tandis que HURD, développé dans le cadre du projet GNU, est un système d'exploitation à micro-noyau Mach (O'Reilly, 2003b). Les débats sur les avantages comparés vont bon train, mais il en ressort que l'utilisation de noyau monolithique correspond à une conception traditionnelle d'un système d'exploitation (O'Reilly, 2003c). Pour l'instant, les systèmes d'exploitation à architecture plus classique comme Linux ou NetBSD rencontrent néanmoins davantage de succès.

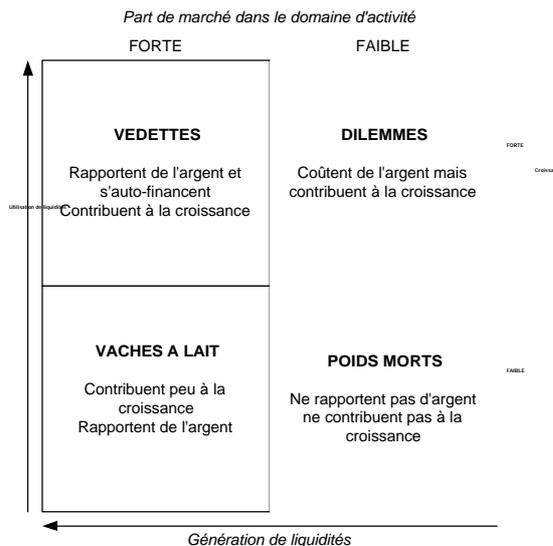
Les changements de licence peuvent entraîner des revirements importants. Citons le cas de Qt, une bibliothèque graphique utilisée notamment dans KDE. Les conditions de la licence de Qt en interdisaient la modification ou l'utilisation en relation avec tout logiciel d'affichage

graphique autre que le vieillissant X Window System. Pour ces autres utilisations, il fallait acheter une licence de développeur qui s'élevait à 1500 USD (O'Reilly, 2003a). En réaction, un projet alternatif libre, baptisé Harmony, vit le jour. Par la suite, Trolltech édita Qt sous licence libre (GPL), faisant perdre tout intérêt au projet Harmony.

Le Boston Consulting Group

Le Boston Consulting Group (Dayan, p12) propose pour sa part une approche permettant de gérer un portefeuille d'activités. Elle se concentre sur deux séries de critères :

- l'attrait du marché, mesuré par son taux de croissance ;
- la position de l'entreprise sur ce marché, mesurée par la part de marché relative.



Le portefeuille d'activité selon BCG (Dayan, 1985, p12)

Elle conduit également à discriminer les activités en fonction, d'une part, de la génération de liquidités et, d'autre part, de l'utilisation des liquidités. Elle distingue quatre types d'activités : les vedettes, les dilemmes, les vaches à lait et les poids morts.

L'argent est ici un facteur discriminant, de peu d'intérêt pour une communauté libre, où domine la culture du don. Par contre, il est possible de modifier cette

grille en en modifiant la variable « argent » :

- la popularité du projet (notoriété, nombre de téléchargement, apport de nouveaux contributeurs, etc), à mettre en parallèle avec l'apport de liquidités ;
- l'exigence en terme de maintien (fixation de bogues, ajout de nouvelles fonctionnalités, rédaction des documentations, etc), à mettre en parallèle avec la consommation de liquidités.

Cette grille conduit à distinguer :

- les vedettes, à savoir des logiciels nécessitant un maintien important mais bénéficiant d'une réelle popularité et d'un grand nombre de contributeurs ;
- les vaches à lait, à savoir les logiciels ne nécessitant plus un maintien important mais toujours populaires ;
- les dilemmes, populaires mais nécessitant un maintien important sans apporter beaucoup de contributeurs ;
- les poids morts, peu populaires et n'apportant pas de nouveaux contributeurs.

	Apache	GNU
Vaches à lait	Web Server	Gcc, Emacs
Vedettes	Tomcat	GNU Enterprise
Dilemmes	?	?
Poids morts	?	Hurd

Proposition de classification (à valider à l'aide d'une métrique cohérente).

Conclusion

Cette approche offre donc de multiples avantages aux communautés libres. Remarquons néanmoins une limite. La motivation du hacker à programmer provient plutôt de l'exercice de sa créativité, de l'occupation de ses loisirs et de l'amélioration de son expertise (BCG, 20??). L'approche « programmation pour le fun » présente donc un décalage avec la méthode rigoureuse présentée supra. L'efficacité des communautés à contribuer à ce type d'étude est néanmoins réelle. Et l'analyse stratégique apporte une assistance appréciable dans le choix des

orientations pour le noyau dur du projet libre. ♦

© Robert VISEUR, 2003
Robert.Viseur@ecocentric.be

Sources

-> Dayan (1985), *Le marketing*, Que-Sais-je , PUF, Paris.

-> Debian (2003), *Annonce concernant la marque déposée Debian*, <http://www.debian.org/News/1998/19980306a.fr.html> (consulté le 21-02-2003).

-> Delameilleure J., *Internet Explorer domine le marché*, <http://www.vninet.be/detalle.asp?ids=/News/Ebusiness//20021219014>, 19-12-2002 (consulté le 21-02-2003).

-> Descombes S. (2002), http://solutions.journaldunet.com/0208/020823_net_scrosoft_shtml (consulté le 19-02-2003).

-> FSF (2003a), *Pourquoi nous parlons de Logiciel Libre*, <http://www.fsfurope.org/documents/whyfs.fr.html> (consulté le 19-02-2003).

-> FSF (2003b), *What Is Copyleft?*, <http://www.fsf.org/copyleft/copyleft.html> (consulté le 19-02-2003).

-> GNU (2003), *Pourquoi il n'y a pas de fichiers GIF sur le site de GNU*, <http://www.gnu.org/philosophy/gif.fr.html> (consulté le 19-03-2002).

-> Kotler & Dubois (2000), *Marketing Management*, Publi-Union.

-> Lycoris, <http://www.lycoris.com> (consulté le 20-02-2003).

-> Lindows, <http://www.lindows.com> (consulté le 20-02-2003).

-> LinuxFr (2003a), *Bitkeeper, RMS et PLONK*, <http://linuxfr.org/2002/10/23/10070.html> (consulté le 19-02-2003).

-> LinuxFr (2003b), *BitKeeper et logiciel libre*, <http://linuxfr.org/2002/10/18/10019.html> (consulté le 19-02-2003).

-> LinuxFr (2003c), *Asterix & Obelix vs. Mobilix.org*, <http://linuxfr.org/2001/08/08/4499.html> (consulté le 21-02-2003).

-> LinuxFr (2003d), *Obelix vs. Mobilix : Obelix*

vainqueur ?, <http://linuxfr.org/2003/01/31/11212.html> (consulté le 21-02-2003).

-> Mariel S. (2003), *Marketing du libre et de l'open source*, http://www.stephanemariel.com/rubrique.php?id_rubrique=17 (consulté le 19-02-2003).

-> Martinez M., *Entrevue avec Linus Torvalds*, <http://www.linuxfocus.org/Francais/March1998/article27.html> (consulté le 19-02-2003).

-> Mobilix (2003), *Trademark Trouble: Obelix versus Mobilix*, http://mobilix.org/mobilix_asterix.html (consulté le 19-02-2003).

-> Open Source Initiative, <http://www.opensource.org> (consulté le 21-02-2003).

-> O'Reilly (2003a), *KDE, Qt, et Troll Tech*, <http://www.oreilly.fr/divers/tribune-libre/fr-x2536.html> (consulté le 19-02-2003).

-> O'Reilly (2003b), *Le GNU Hurd*, <http://www.oreilly.fr/divers/tribune-libre/fr-x982.html> (consulté le 20-02-2003).

-> Raymond E. (1998), http://www.firstmonday.dk/issues/issue3_10/raymond/ (consulté le 19-03-2002).

-> Smets-Solanes J.-P. (1998), 'L'économie du logiciel libre', <http://www.smets.com>, 29 juin 1998 (consulté le 10 janvier 2002).

-> Vande Walle P. (2003), *La brevetabilité du logiciel en questions*, <http://www.isoc.lu/chroniques/brevet-2002.html> (consulté le 19-03-2003).

-> Xiti (2002), *INTERNET EXPLORER : 95% DU MARCHE !*, <http://www.xiti.com/etudes.asp> (consulté le 21-02-2003).