

# Revue d'économie industrielle

136 | 4ème trimestre 2011 L'économie du logiciel libre

# Le modèle économique du logiciel *Open Source* : viabilité et compétitivité

## **Dominique Torre**



#### Édition électronique

URL: https://journals.openedition.org/rei/5163

DOI: 10.4000/rei.5163 ISSN: 1773-0198

#### Éditeur

De Boeck Supérieur

#### Édition imprimée

Date de publication : 15 décembre 2011

Pagination: 11-16 ISSN: 0154-3229

### Référence électronique

Dominique Torre, « Le modèle économique du logiciel *Open Source* : viabilité et compétitivité », *Revue d'économie industrielle* [En ligne], 136 | 4ème trimestre 2011, mis en ligne le 17 février 2012, consulté le 02 juin 2022. URL : http://journals.openedition.org/rei/5163; DOI: https://doi.org/10.4000/rei.5163

© Revue d'économie industrielle

# LE MODÈLE ÉCONOMIQUE DU LOGICIEL *OPEN SOURCE* : VIABILITÉ ET COMPÉTITIVITÉ (\*)

es communautés du logiciel *Open Source* proposent depuis bientôt vingt ans un modèle d'innovation ouverte, concurrent du modèle dominant d'appropriation et de diffusion des innovations dans l'industrie. Quand des formes strictes de propriété intellectuelle protègent les innovations et incitent à l'effort dans le monde du copyright, la transmission des connaissances est privilégiée par le modèle « viral » du *copyleft*, au risque en apparence de décourager la créativité. Chaque développeur utilise alors librement l'apport des autres pour élaborer son propre programme et incrémenter la base de connaissances disponible. Il peut aussi tirer de cette connaissance accumulée d'éventuels avantages pécuniaires, à condition de mettre le résultat de son propre travail de développeur à la disposition d'autrui dans les mêmes conditions, de rendre contagieuse sa propre liberté.

Ce modèle en marge des comportements habituels de valorisation de la contribution individuelle dans la firme et sur le marché a longtemps suscité doute et scepticisme. Année après année, dans différents environnements économiques et politiques, il s'est pourtant développé. L'effectif des développeurs s'est renouvelé depuis 20 ans, certains défenseurs de la première heure de l'*Open Source* sont toujours là, d'autres ont disparu. À une première génération répondant à des motifs plutôt idéologiques ont succédé d'autres développeurs, plus pragmatiques et convaincus de la capacité de conjuguer réussite professionnelle et épanouissement personnel dans le choix de l'*Open Source*. Ces développeurs acceptent, pour une période plus ou moins longue de l'année, du mois, de la journée, de mettre à la disposition du collectif les innovations qu'ils réalisent grâce à ce même collectif. S'associent dans les communautés de développeurs, des professionnels de toutes nationalités, de tous âges et de tous niveaux de formation, des particuliers aussi dont certains contribuent à la base de connaissances laissée libre, grâce à l'ouverture du code-source qui permet une incrémentation collective de la base de connaissances disponible en ligne.

Enfin, gravitent autour de l'*Open Source* des tiers – pouvoirs publics, scientifiques, chercheurs – stimulés par cette forme de production des connaissances étonnamment durable dans un environnement qui semble la condamner. On

<sup>(\*)</sup> Cette introduction doit beaucoup aux remarques et suggestions d'Éric Darmon, Christian Longhi et Sylvie Rochhia. Que tous trois en soient remerciés.

parle du monde du « libre », de sa culture, de ses valeurs: loin d'ostraciser ceux qui s'en réclament, ces valeurs en marge du monde des affaires permettent à des milliers de développeurs souffrant d'un trop grand anonymat de trouver de nouveaux moyens d'exprimer leur différence, leur originalité, dans le cadre d'une valorisation sociale qui ne repose plus sur la reconnaissance de leurs pairs ou de leurs supérieurs dans l'entreprise mais sur l'estime de ceux qui, loin sur la toile et le plus souvent anonymes, partagent la même conception de la création technologique et de l'innovation numérique. De nombreux grands acteurs commerciaux de l'industrie du logiciel comme IBM, Sun, Hewlett Packard, ont choisi d'utiliser des bases de connaissances ouvertes et de laisser tout ou partie de leurs codes ouverts. Des centaines de milliers d'autres, plus petits, font aujourd'hui de l'*Open Source* un modèle d'affaire, dont la diversité et les multiples formes rendent toute description difficile, mais aussi actif dans le secteur du logiciel.

Parmi les logiciels grand public, la contribution de l'*Open Source* est particulièrement importante bien que souvent sous-estimée. Ces logiciels démentent l'opinion selon laquelle un développement communautaire, centré sur les seuls besoins de développeurs-utilisateurs (von Hippel, 2005), ne pourrait répondre à une demande finalisée, s'avérerait peu à même d'engager des dépenses de marketing et ne pourrait proposer de solutions adaptées à une consommation de masse. Quand manque parfois la touche de personnalisation qui garantit le succès immédiat de certains produits propriétaires, la qualité des solutions *Open Source* s'apprécie à l'usage, les effets de réputation faisant plus pour leur adoption que les campagnes de publicité accompagnant le lancement des solutions propriétaires : c'est ainsi que le système d'exploitation Linux a pu acquérir chez les utilisateurs exigeants une notoriété qui impose durablement ses qualités sous des dehors pourtant traditionnellement austères. Par leur effort d'adoption, voire par leur insertion aux communautés, les organisations (firmes industrielles, services publics, collectivités locales) ont contribué de façon décisive au succès de l'*Open Source* (Dahlander et Magnusson, 2008; Le Texier et Versailles, 2009; Hauge, Ayala et Conradi, 2010).

Pour les développeurs, les obstacles liés à la personnalisation insuffisante n'en sont plus: les utilisateurs professionnels profitent de la richesse des bases *Open Source*, de l'étendue des solutions qu'elles proposent, de la possibilité de communiquer avec d'autres développeurs, mais aussi de la sécurité que présente un code non crypté par un éditeur propriétaire. Enfin, comme l'utilisateur final, ils sont sensibles au prix, l'accès libre au code-source s'accompagnant souvent de la gratuité des lignes de codes, logiciels ou modules (l'*Open Source software* est souvent *freeware*).

L'émergence du mouvement est décrite en 2002 par un article de J. Lerner et J. Tirole (2002) qui reste aussi une référence à peine datée pour la plupart des thèmes qui touchent à son développement. Trois vagues initiales (cf. Le Texier, 2008) caractérisent cette phase importante. D'abord, les balbutiements du logiciel dans les années 1960 et 1970 qui associent le privé (les laboratoires ATT Bell) et le public (l'université de Berkeley) dans l'élaboration de la base Unix, sorte d'ancêtre de l'*Open Source* dans sa configuration initiale. Puis, en 1983, la création par Richard Stallman de la *Free Software Foundation*, avec l'émergence de la fameuse licence GPL qui permet l'usage du code sous licence mais oblige l'utilisateur à laisser libre disposition du résultat de son usage. Enfin, viennent les années 1990 et le

développement des communautés *Open Source*, autour des projets Linux et Debian notamment. Une discussion débute alors autour du caractère excessif de la licence GPL et met en évidence divers degrés d'acceptation du principe de non-appropriation des connaissances en matière d'innovation logicielle. À partir de 1997, les économistes et les sociologues prennent conscience du caractère novateur du mouvement. Paul Raymond en popularise le modèle avec une série de conférences pendant les années d'euphorie de la « Nouvelle Économie » puis en publiant son ouvrage *La Cathédrale et le Bazar* (Raymond, 1999): deux modèles coexisteraient en matière d'innovation logicielle, d'une part celui de la rationalisation des tâches et des projets, symbolisé par la démarche de Microsoft et, d'autre part, celui des initiatives décentralisées et à peine coordonnées par les communautés, celui de l'*Open Source*. Le modèle *Open Source* acquiert une rationalité économique (Rossi, 2006) et les deux modèles propriétaire et ouvert confirment leur emprise sur l'industrie du logiciel, dans une sorte de contestation permanente par l'un des deux mondes des avantages momentanément acquis par l'autre (Bonaccorsi et Rossi, 2003).

Les deux sujets abordés ici contribuent à la réflexion relative aux enjeux industriels et économiques actuels de l'*Open Source*. Le premier aborde la question du statut temporel du modèle du logiciel *Open Source*, comme forme d'organisation de l'innovation. Le second examine la question de sa coexistence dans le temps avec le modèle propriétaire.

# LE MODÈLE D'ORGANISATION DU LOGICIEL *OPEN SOURCE* EST-IL DURABLE?

La pérennisation de la forme d'organisation des communautés *Open Source* tient en partie aux motivations de leurs membres. Les motivations des développeurs demeureront-elles intactes au cours du temps? Ces motivations ont été analysées dès l'émergence de l'*Open Source* (Ghosh, Glott, Krieger et Robles, 2002; Hars et Ou, 2002; Lerner et Tirole, 2003; Lakhani et Wolf, 2005). On a identifié des motivations extrinsèques, assises sur la réalité d'un gain individuel souvent différé quand il est monétaire (les avantages sur le marché du travail lié à la réputation), mais assez immédiat quand il relève d'autres formes de rétributions (la reconnaissance par les pairs...) moins naturellement quantifiables. À côté, on trouve des motivations intrinsèques également observées dans le monde de l'entreprise, telles que la satisfaction procurée par l'atteinte d'un objectif. D'autres motivations intrinsèques sont cependant spécifiques à l'*Open Source*. L'appartenance à une communauté donne ainsi un poids plus important aux phénomènes de réciprocité, aux dons mutuels, souvent mieux analysés par les sociologues et les anthropologues que par les économistes. Ces motivations peuvent-elles s'émousser avec le développement de l'industrie du logiciel? De branche de la « Nouvelle Économie », cette industrie s'est en effet muée progressivement en partenaire de toutes les autres, pourvoyeuse de modules, de composantes, de commandes, incorporées dans tous nos produits technologiques. En pénétrant d'autres secteurs, le monde du logiciel a perdu de son originalité, un peu de son élitisme aussi. Cette banalisation du produit n'est-elle pas à même de diluer l'effort d'innovation de ceux qui s'attachent à en faire évoluer les formes? L'auto-organisation n'est-elle pas susceptible de ramifier les communautés en mille segments peu favorables aux interactions?

Le texte de Stefan Koch analyse l'organisation des communautés de développeurs pour en examiner les performances, mesurées à l'effort consenti et à l'efficacité de cet effort. Au-delà des motivations individuelles, de la nature du code auquel s'applique l'effort collaboratif (Den Besten, Dalle et Galia, 2008), c'est l'organisation de la communauté qui compte aussi : améliorer cette organisation pourrait alors pallier le risque d'affaiblissement des motivations extrinsèques en permettant à la réciprocité de s'exprimer dans de meilleures conditions. Au-delà des strictes communautés, Nicolas Jullien et Jean-Benoît Zimmermann appréhendent l'écosystème du logiciel dans sa globalité. Cet ensemble est composé des éditeurs propriétaires, des universités, des développeurs indépendants, des utilisateurs professionnels du libre et les conditions de son fonctionnement efficace tiennent à la coordination de composantes organisationnelles différenciées mais complémentaires, les formes adéquates de protection de la propriété intellectuelle/de la diffusion des innovations, devant tenir compte de cette complexité.

Si l'efficacité du modèle ouvert rapidement prédite (Kogut et Meciu, 2001) est bien admise aujourd'hui, doit-on envisager son exportation vers d'autres branches de l'activité productive, elles aussi dépendantes d'un processus d'innovation efficace? La réponse de Julien Pénin est circonstanciée: des conditions nécessaires doivent être respectées, essentiellement quant aux modalités d'utilisation et de valorisation de la connaissance dans l'industrie concernée. La communauté est efficace quand l'innovation peut puiser dans un stock de connaissances mais elle devient inefficace quand la connaissance doit être ordonnée *a priori*, organisée dans un objectif extrêmement précis et peu compatible avec une logique d'évolution naturelle du savoir. Une fois établi qu'une forme d'organisation de l'innovation est efficace, il est du rôle des pouvoirs publics d'en favoriser le développement. Stefano Comino, Fabio Manenti et Alessandro Rossi analysent empiriquement le rôle de l'État en matière de soutien des initiatives Open Source en Europe. Leurs conclusions attestent d'une grande hétérogénéité de l'initiative publique des différents États et d'un manque de la coordination communautaire quand le versant propriétaire peut se prévaloir de stratégies de firmes qui ne connaissent pas les frontières.

## L'OPEN SOURCE RESTERA-T-IL COMPÉTITIF?

Dans les phases de croissance explosive de l'activité logicielle, la dynamique industrielle a pris des formes particulières, guerre de standards en termes de compatibilité avec le hardware, mises en place de compatibilités ascendantes ou descendantes. Ces formes de concurrence révèlent rétrospectivement trois de leurs caractéristiques essentielles: elles concernent un marché de consommateurs finals, l'initiative des actions revient à un éditeur propriétaire, enfin cet éditeur propriétaire a une stratégie appropriée à son ou ses concurrent(s) également propriétaire(s). L'*Open Source* a survécu sans grand mal à ces phases de constitution, puis de consolidation de l'industrie logicielle: il proposait un exutoire aux frustrations de développeurs enrôlés professionnellement au monde du propriétaire, offrait aux éditeurs une base de connaissances parfois « recyclable », garantissait enfin la gratuité d'un grand nombre de solutions professionnelles. Par moments, le choc entre ces deux mondes s'est cependant fait plus frontal, parce que les utilisateurs finals des solutions proposées étaient les mêmes ou presque,

ou parce qu'il était tentant pour l'éditeur de profiter au mieux d'externalités liées à la taille de son marché. Le second bloc d'articles de ce numéro à thème envisage les formes de concurrence qui s'exercent alors entre éditeurs et communautés. Différents *scénarii* concurrentiels sont étudiés, depuis les plus classiques, jusqu'à des formes de compétition plus récentes, où la confrontation des partenaires n'exclut pas entre eux certaines formes de coexistence et de coopération.

Éric Darmon, Thomas Le Texier et Dominique Torre analysent un scenario d'adoption mettant en regard un éditeur de logiciel propriétaire doté d'actions rationnelles et une communauté de développeurs *Open Source* stratégiquement passive malgré son potentiel créatif et l'adaptabilité de ses solutions logicielles. Face à des utilisateurs potentiels hétérogènes quant à leur propension à adopter une solution libre d'accès moins immédiat, cette configuration engendre très souvent des équilibres stables dépendant de l'état des anticipations initiales et tels que l'une des bases exclut l'autre du marché. Même quand elles ne s'imposent pas durablement, les solutions Open Source incitent cependant l'éditeur à baisser ses prix et à améliorer sa qualité, ce qui rend la présence du libre, même « latente », bénéfique du point de vue du bien-être. La concurrence entre libre et propriétaire est aussi étudiée par Riccardo Leoncini, Francesco Rentocchini et Giuseppe Vittucci Marzetti. À l'aide d'une méthodologie originale, celle d'un modèle épidémiologique intégrant les motivations des développeurs à côté de celle de l'éditeur, ils vérifient que les équilibres stationnaires stables de ce modèle sont aussi des configurations de type winner-takes-all.

Cette confrontation entre deux mondes irréconciliables n'est cependant pas la seule perspective offerte à l'industrie du logiciel. Les logiques propriétaire d'une part et *Open Source* d'autre part peuvent aussi interférer, alterner, s'hybrider (Bonaccorsi, Giannangeli et Rossi, 2006). L'ouverture partielle du code est une stratégie déjà adoptée par certains éditeurs : ceux-ci mettent alors dans le domaine public une partie de leurs incréments et, en contrepartie, intègrent à leurs lignes de code des inputs provenant du libre (Darmon et Torre, 2009). Le duopole asymétrique qu'étudie Mourad Zeroukhi explore les possibilités ouvertes par cette stratégie quand deux éditeurs propriétaires se font face. L'ouverture du code de deux concurrents permet une fertilisation mutuelle de leurs bases respectives et engendre des effets à la fois positifs et négatifs sur leur capacité à s'imposer au marché. Seule une symétrie suffisante entre les compétiteurs rend cependant avantageuses ces interactions du point de vue du bien-être.

Avec ces formes d'ouverture partielle, l'hybridation des lignes de code libres et propriétaires se développe aujourd'hui. Ces associations tendent vers un continuum de stratégies de la part des éditeurs, le même centre de décision pouvant jouer à la fois des possibilités d'une licence traditionnelle pour protéger une partie de son code, de licences ouvertes mais permissives (non *copyleft*) de type BSD pour une autre partie de ses apports, enfin de licences GPL de type *copyleft* pour les modules laissés au soin d'un développement communautaire. Une telle diversification revient pour les éditeurs à reconnaître explicitement l'intérêt de l'*Open Source* tandis que le succès de ces croisements met un peu plus de distance entre les communautés de développeurs des années quatrevingt-dix et leurs héritières pragmatiques des années 2010. Qui pourrait cepen-

dant nier que l'idéalisme, voire le sectarisme des premières a permis au modèle de grandir, et de proposer aujourd'hui des solutions pragmatiques cherchant à conjuguer la libre disposition des répertoires de connaissance et les incitations individuelles et collectives nécessaires à la réussite d'un projet industriel?

#### RÉFÉRENCES

- BONACCORSI A., GIANNANGELI S., ROSSI C. (2006), « Entry strategies under competing standards: hybrid business models in the Open Source software industry », *Management Science*, 52 (7), 1085-1098.
- BONACCORSI A., ROSSI C. (2003), « Why Open Source software can succeed », Research Policy, 32 (7), 1243-1258.
- COGUT B., MECIU A., (2001), « Open Source Software Development and Distributed Innovation », Oxford Review of Economic Policy, 17 (2), 248-264.
- DAHLANDER L., MAGNUSSON M.G. (2005), « Relationships between Open Source software companies and communities: observations from Nordic firms », Research Policy, 34, 481-493.
- DAHLANDER L., MAGNUSSON M.G. (2008), « How do firms make use of Open Source communities? », *Long Range Planning*, 41 (6), 629-649.
- DARMON E., TORRE D. (2009), « Open Source and Commercial Software Platforms: is Coexistence a temporary or sustainable Outcome? », *International Journal of Open Source Software and Processes*, 1, (2), 67-80.
- DEN BESTEN M., DALLE J.-M., GALIA F. (2008), « The allocation of collaborative efforts in open-source software », *Information Economics and Policy*, 20, (4), 316-322.
- GHOSH R.A., GLOTT R., KRIEGER B., ROBLES G. (2002), Free/Libre and Open Source Software: Survey and Study, International Institute of Infonomics, University of Maastricht.
- HARS A., OU S. (2002), « Working for Free? Motivations of Participating in Open Source Projects », *International Journal of Electronic Commerce*, 6, (3), 25-39.
- HAUGE Ø., AYALA C., CONRADI R. (2010), « Adoption of Open Source software in software-intensive organizations. A systematic literature review », *Information and Software Technology*, 52, 1133-1154.
- HERTEL G., NIEDNER S., HERRMANN S. (2003), « Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel », Research Policy, 32 (7), 1159-1177
- LAKHANI K. R., WOLF R. G. (2005), « Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects », in Perspectives on Free and Open Source Software, Cambridge and London: MIT Press, 3-21.
- LERNER J., TIROLE J. (2002), « Some simple economics of Open Source », *Journal of Industrial Economics*, 52, 197-234.
- LE TEXIER T. (2008), Le rôle des innovations ouvertes dans la reconstruction des activités industrielles, thèse de doctorat en Sciences économiques, université Nice-Sophia Antipolis, décembre 2008.
- LE TEXIER T., VERSAILLES D. (2009), « Open Source Software Governance Serving Technological Agility: The Case of Open Source Software within the DoD », *International Journal of Open Source Software and Processes*, 1, (2), 14-27.
- RAYMOND E., (1999), The Cathedral and the Bazaar: Musings on Linux and Open Source from an Accidental Revolutionary. O'Reilly and Associates, Sebastopol, CA.
- ROLANDSSON B., BERGQUIST M., LJUNGBERG J. (2011), « Open Source in the firm: Opening up professional practices of software development », *Research Policy*, 40, 576-587.
- ROSSI M.A. (2006), « Decoding the Free/Open Source Software Puzzle: A Survey of Theoretical and Empirical Contributions », in Blitzer, J. and Schröder, J.H., eds. *The Economics of Open Source Software Development*, Elsevier, Amsterdam.
- von HIPPEL E. (2005), Democratizing Innovation. The MIT Press, Cambridge, Mass.
- von KROGH G., von HIPPEL E. (2006), « The promise of research on Open Source software », *Management Science*, 52 (7), 975-983.
- YU L., (2008), « Self-organization process in open-source software: an empirical study », *Information and Software Technology*, 50 (5), 361-374.