

LES LIVRETS BLEUS DU

LOGICIEL LIBRE

Modèles économiques

PATRICE BERTRAND

ÉTIENNE JULIOT

STÉFANE FERMIGIER

MARINE SOROKO

OLIVIER VINÉ

Préface de

STÉFANE FERMIGIER

LOGICIEL LIBRE

*Le présent document est inspiré d'une synthèse de l'atelier «**Les Business Modèles du Libre**» qui s'est tenu le 4 juillet 2013 à l'initiative du groupe thématique Logiciel Libre du pôle de compétitivité Systematic Paris-Region, grâce à l'hébergement gracieux de Smile.*

Sources :

Les modèles économiques du libre, Patrice Bertrand, président du CNLL

Modèle intégrateur open source, Olivier Viné, directeur ingénierie, Open Wide

L'open source professionnel : un business d'éditeur open source,

Stéphane Fermigier, fondateur et CEO, Abilian

Retour d'expérience sur le business model d'un intégrateur open source,

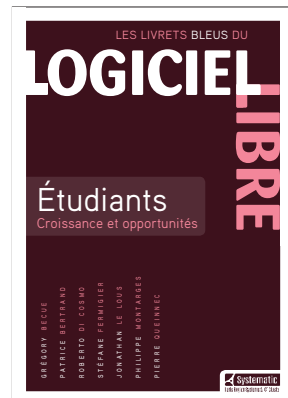
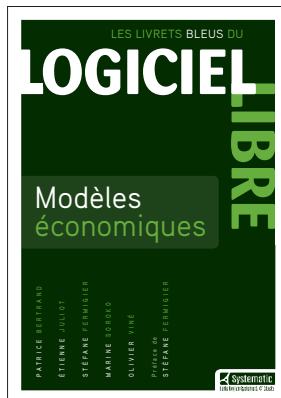
Marine Soroko, directrice associée, Core-Techs

GTLL : business editeur, Étienne Juliot, vice-président, Obeo

Présentations disponibles sur http://fr.slideshare.net/gtll_systematic/

Remerciements chaleureux à Anne de Forsan de l'agence StoriesOut, à Muriel Shan Sei Fan pour le travail d'édition, et à Didier Méresse (Nord Compo) pour la conception.

LES LIVRETS BLEUS DU GTLL : DES REPÈRES POUR COMPRENDRE



Sommaire

Chapitre 1

Les acteurs du logiciel libre? 7

Chapitre 2

Le modèle des fondations et communautés open source 9

Le modèle des communautés 9

Le modèle des fondations 10

Chapitre 3

Les modèles d'éditeurs open source 13

Les différents modèles d'éditeurs 13

Motivation économique de l'éditeur open source 15

Le modèle économique de l'éditeur open source 16

Les services professionnels de l'éditeur 16

L'enjeu de l'organisation du travail
entre les équipes 17

La mutualisation comme moyen de financement
d'un effort de R&D important **18**

Le modèle du revenu récurrent autour
du logiciel **18**

Les compétences clés de l'éditeur open source **22**

La vision **23**

Le rôle de gardien du temps **23**

La qualité **24**

Être au centre d'un système
d'innovation ouverte **25**

Avantages et faiblesses **27**

Avantages **27**

Inconvénients **27**

Dérives possibles **28**

La modélisation du revenu récurrent **30**

Chapitre 4

Le modèle des intégrateurs **34**

Le modèle économique de l'intégrateur open source **34**

Le modèle sans éditeur **35**

Le modèle avec éditeur **36**

Différences entre intégrateur open source et intégrateur propriétaire **36**

Les clés de l'intégrateur open source **38**

Le lien étroit avec les communautés
et les éditeurs **38**

Ressources humaines : le bon accueil
de profils open source **39**

Une expertise avérée sur les technologies
de pointe **39**

Un environnement technologique surveillé **40**

L'offre open source est technocentrée **40**

Risque inhérent au modèle économique
de l'intégrateur **41**

Chapitre 5

**Relation entre les acteurs :
enjeux et complexité 42**

Préface

Le logiciel libre représente à ce jour en France autour de 4 milliards d'euros de chiffre d'affaires annuel dans le secteur du logiciel et des services, et près de 30 000 emplois (source : PAC). La question de l'empreinte du logiciel libre sur l'économie de notre pays ne se pose donc plus depuis quelques années.

Ces chiffres mettent la France dans une position de leadership mondial, recoupant les analyses de nombreux experts depuis plusieurs années (sources : PAC, Georgia Tech), et constituent un atout pour notre compétitivité et notre capacité à gérer la transition numérique de notre économie.

<https://www.pac-online.com/france-leads-europe-open-source-adoption>

<http://www.redhat.com/f/pdf/ossi-index-ranks.pdf>

Un tel poids économique, pour un secteur perçu par certains observateurs non informés comme l'apanage du non-marchand, peut surprendre au premier abord. Pour éclairer ce paysage, de nombreux travaux de recherche ou de vulgarisation ont été publiés depuis une quinzaine d'années. Ils représentent un corpus de centaines d'articles et d'ouvrages qui mériteraient une analyse exhaustive. Mais pour tous ceux – entrepreneurs du logiciel libre, responsables de systèmes informatiques, décideurs des stratégies de soutien à notre économie – qui n'ont pas le temps nécessaire à consacrer à ce sujet, au demeurant passion-

nant, nous avons élaboré cette synthèse, issue d'une réflexion collective menée lors d'un atelier en juillet 2013.

L'objectif était d'offrir un panorama ancré dans la réflexion pragmatique d'acteurs du domaine – principalement éditeurs de logiciel libres et intégrateurs. Celle-ci a permis d'exposer les différents mécanismes économiques, marchands ou non, qui sous-tendent le développement des dizaines de milliers de logiciels libres utilisables en entreprise et des centaines d'entreprises en France qui en vivent exclusivement ou quasi-exclusivement – 400 à 500 selon le décompte du Conseil National du Logiciel libre, ou CNLL).

<http://cnll.fr/news/resultats-etude-2014/>

Nous espérons que cette lecture dissipera certains malentendus, précisera certaines notions, et surtout encouragera la coopération entre tous les acteurs de l'écosystème, dans une optique de création de valeur et d'emplois sur nos territoires, en tirant parti d'un des avantages compétitifs majeurs que nous avons su construire, collectivement et avec le soutien des pouvoirs publics, depuis près de 15 ans.

Stéfane Fermigier,
président du groupe thématique Logiciel Libre

L'open source est un marché à la fois complexe et en constante évolution. Pour en comprendre les différents mécanismes économiques, il faut en identifier les acteurs, leurs offres respectives et positionnements, et analyser comment ils se partagent les revenus. C'est pourquoi le groupe thématique Logiciel Libre de Systematic a rassemblé éditeurs et intégrateurs afin de dresser un panorama des modèles économiques existants et des interactions entre les différents intervenants de ce marché.

Chapitre 1

Les acteurs du logiciel libre?

Le logiciel libre est caractérisé par sa liberté d'utilisation, un développement collaboratif et sa redistribution à une communauté de contributeurs et d'utilisateurs. Il est d'emblée nécessaire de rappeler que « libre » ne signifie pas gratuit. Bien que de nombreux projets open source soient accessibles gratuitement, gérés par des fondations ou des communautés, **de nombreux autres se positionnent clairement dans une logique marchande.**

Parmi les acteurs de l'open source figurent les éditeurs et les intégrateurs, entreprises à but lucratif dont l'un des objectifs est de créer de la valeur. Cet écosystème complexe, en constante évolution, nécessite pour le comprendre de cartographier les différents modèles économiques (business models) des parties prenantes et d'appréhender les problématiques auxquelles ces acteurs sont confrontés.

- Quels sont les modèles économiques qui font vivre le logiciel libre et open source ?
- Comment un modèle économique open source aidera-t-il au succès d'un éditeur de logiciel ?

- Les intégrateurs de solutions open source sont-ils des prestataires comme les autres ?

Dans l'écosystème de l'open source, on distingue les modèles suivants :

- le modèle des communautés,
- le modèle des fondations,
- le modèle des éditeurs,
- le modèle des intégrateurs et des prestataires de services (qui apparaît comme étant moins spécifique du logiciel libre).

Chapitre 2

Le modèle des fondations et communautés open source

Avant de détailler les modèles de deux des acteurs majeurs du modèle du libre, les éditeurs et les intégrateurs, on décrit ci-après les modèles des communautés et des fondations.

Le modèle des communautés

Les communautés se perçoivent comme étant à l'écart des mécaniques économiques, voire dans une relation antagoniste d'avec le monde des entreprises. Or, même dans ce domaine, des rouages économiques sont à l'œuvre. Dans les années 1990, des études universitaires ont été consacrées aux motivations des contributeurs bénévoles, pour certaines désintéressées, pour d'autres ayant des fondements économiques.

De façon désintéressée, les contributeurs espèrent gagner la reconnaissance de leurs pairs, être eux-mêmes

fiers du travail qu'ils ont réalisé, échanger avec des développeurs de haut niveau et produire des travaux utiles. **Mais les motivations économiques sont tout aussi présentes : certains espèrent recevoir des offres d'emploi, proposer des prestations payantes autour du logiciel, faire avancer des chantiers d'innovation et des fonctionnalités dans la perspective du développement d'une start-up.**

Le modèle des fondations

Le modèle économique des fondations open source, quant à lui, est caractérisé par la mutualisation des efforts de R&D. Dans ce modèle, les équipes d'ingénieurs de différentes entreprises collaborent à un même projet. Dans certains cas, la propriété intellectuelle issue de ces travaux communs peut être cédée à la fondation. À l'instar du modèle des communautés, celui des fondations comporte des variantes.

Les fondations portent une grande part du patrimoine du logiciel libre. Les acteurs de l'open source construisent ensemble un patrimoine logiciel mis à disposition de l'ensemble de l'humanité, sans condition, ce qui confère d'ailleurs à la démarche un caractère humaniste. Le plus souvent, les fondations prennent en charge le logiciel le plus générique, l'infrastructure, les outils de développement, les utilitaires, etc., en laissant des domaines

importants de recherche et d'innovation non couverts. Ce sont précisément ces domaines qui ont été conquis par les éditeurs open source.

Si la plupart des fondations sont des organisations à but non lucratif, ce n'est généralement pas le cas de leurs membres et de leurs contributeurs. Si les entreprises ne se positionnent ni dans une démarche philanthropique ni de mécénat, quel est alors leur intérêt ? L'intérêt des entreprises contribuant à des fondations peut être double.

Tout d'abord, **le logiciel peut être un « entrant » stratégique pour l'entreprise**, qu'elle intègre dans ses produits et ses activités, un composant sans lequel le modèle économique de l'entreprise serait en péril. Ici, l'entreprise a pour principaux objectifs :

- d'adapter des produits au logiciel et inversement,
- de disposer de logiciel pour une fraction du coût de R&D (en vertu du principe de la mutualisation de la R&D),
- de réduire et de maîtriser sa dépendance stratégique, notamment en participant à la gouvernance de ces logiciels.

Ensuite, **l'entreprise propose des services et des logiciels périphériques au logiciel porté par la fondation.** Ainsi, elle peut démontrer son expertise et se faire connaître à l'international (dimension marketing), mieux

intégrer son produit avec le logiciel de la fondation pour attirer et fidéliser des développeurs d'exception.

La fondation a un rôle déterminant dans l'écosystème du libre car elle rassemble les entreprises d'un même marché autour d'un projet commun. Elle permet à des entreprises concurrentes de dépasser leurs antagonismes pour se concentrer sur l'intérêt commun qu'elles auraient aux résultats d'un tel projet, et d'initier des projets qui n'auraient pu voir le jour autrement.

En conclusion, **le modèle des fondations est celui de la mutualisation de R&D, qui permet de répondre à des besoins génériques communs qu'aucune entreprise n'aurait les moyens de financer individuellement.**

Chapitre 3

Les modèles d'éditeurs open source

L'éditeur open source est une entreprise comme les autres, c'est-à-dire à but lucratif. Il développe et distribue ses produits sous une licence open source. **Le développement de ses logiciels représente un coût, que l'éditeur finance grâce à la perspective d'un retour sur investissement et de bénéfices futurs.**

L'éditeur est également au centre de l'écosystème du libre car il anime les communautés d'utilisateurs, de contributeurs et de prestataires.

Les différents modèles d'éditeurs

On peut distinguer cinq grands modèles économiques d'éditeurs :

- Celui des *fondations*, vu plus haut.
- Celui des *doubles licences* : le logiciel développé est libre, sous une licence à copyleft fort (typiquement GPL ou Affero GPL) ce qui ne permet pas de l'intégrer, sous cette licence, dans un produit ou service propriétaire.

Une deuxième licence, propriétaire, est vendue aux utilisateurs qui souhaitent justement s'affranchir des contraintes du copyleft.

- Celui de l'*open core* et ses variantes : l'éditeur développe un noyau technologique open source, et un produit propriétaire au-dessus de ce noyau, ou bien développe deux produits au-dessus d'un noyau commun, l'un en open source et l'autre, avec des fonctions additionnelles, vendu sous une licence traditionnelle. Dans ce dernier cas, les deux variantes du logiciels sont souvent baptisées « community » et « entreprise ».
- Celui du *Cloud* : l'éditeur développe un produit open source et en tire des revenus en le proposant sous forme de service dans le cloud.
- Celui, enfin, de l'*open source professionnel* : il s'agit du modèle « dans lequel l'éditeur tire ses revenus de services professionnels, de la maintenance et du support associés au logiciel qu'il édite » (*source : Wikipédia*).

Nous allons principalement nous intéresser à ce dernier modèle économique, car c'est celui qui s'éloigne le plus des modèles propriétaires. Nous l'appellerons plus simplement, et par usage, « modèle de l'éditeur open source ».

Motivation économique de l'éditeur open source

La tendance naturelle, pour un éditeur de logiciel, qu'il soit open source ou non, est de rechercher à terme un revenu (en grande partie) passif et récurrent, de façon à recouvrer l'investissement initial en rémunérant par ailleurs la prise de risque liée à cet investissement.

Un autre de ses défis majeurs est de réussir à travailler avec des intégrateurs, notamment les convaincre de proposer sa technologie à leurs clients et leur fournir une solution complète. Dans cette optique, pour un partage pérenne de la chaîne de valeur, l'éditeur open source et les intégrateurs spécialisés doivent chercher à proposer des services professionnels complémentaires.

Enfin, la démarche qualité est au cœur de sa stratégie.

L'éditeur doit en effet assurer sur le court, moyen et long terme le bon fonctionnement de sa solution, pour maximiser la satisfaction de ses clients et minimiser le coût du support, et s'appuyer sur une architecture qui permette l'évolution de sa solution, au gré des besoins des clients et des contributeurs open source, sans remettre en question la qualité du logiciel développé.

Le modèle économique de l'éditeur open source

Dans ce modèle, l'éditeur open source assure ses revenus en développant et en commercialisant des logiciels mais également en proposant des services à ses clients.

Les services professionnels de l'éditeur

L'éditeur intègre dans son offre trois principaux types de services : le conseil, la customisation et le développement de solutions spécifiques, et la formation.

Le conseil : l'éditeur accompagne ses clients dans la mise en œuvre de sa solution afin de les adapter au mieux à leurs besoins métiers.

La customisation et les développements spécifiques : l'éditeur peut être amené à développer des modules spécifiques à un client, ou peut intervenir dans l'adaptation de sa solution aux besoins métier. Dans ce dernier cas, ces prestations peuvent se faire, selon la configuration des projets, soit seuls (auquel cas l'éditeur joue partiellement un rôle de SSII), soit conjointement à une SSII.

Les formations à destination de l'intégrateur partenaire ou du client : l'éditeur peut proposer de former les équipes de l'intégrateur ou directement celles du client sur sa solution logicielle. La formation est une source de revenu très intéressante pour l'éditeur car son budget

s'inscrit souvent dans l'enveloppe que les entreprises doivent légalement consacrer à la formation professionnelle continue. Cependant, cette prestation mobilise des équipes en interne chez l'éditeur et nécessite donc une organisation adaptée.

L'enjeu de l'organisation du travail entre les équipes

Comme dans toute entreprise, il faut veiller au bon équilibre entre demande et ressources disponibles. Du point de vue de l'organisation, l'équipe dédiée aux services professionnels peut être ou non séparée de l'équipe des développeurs.

Dans le cas où les équipes sont séparées, la problématique majeure consiste à **assurer une bonne circulation des compétences au sein des équipes** pour assurer aux clients un niveau de conseil et de compétences optimal sur la solution.

Dans le cas où les services sont assurés par l'équipe des développeurs, **le risque majeur est la mise en péril de la feuille de route du développement**, en allouant des ressources aux besoins clients, au détriment des développements prévus.

La mutualisation comme moyen de financement d'un effort de R&D important

Certains besoins génériques nécessitent de lourds efforts de R&D qu'il est difficile de consentir car les revenus qui en sont attendus ne pourront être collectés qu'à long terme. Regrouper ou mutualiser les besoins de plusieurs clients, et donc plusieurs financements, peut ainsi être une solution pertinente pour se lancer dans de tels développements. Cependant, la mutualisation présente un coût supérieur à celui d'une prestation de services spécifiques et requiert donc d'être face à un client compréhensif ou face à plusieurs clients voulant des fonctionnalités similaires.

Dans le cas où les développements réalisés sont réellement innovants, le crédit d'impôt recherche (CIR) peut être un moyen de financer le surcoût lié à la réalisation de développement génériques plutôt que spécifiques. Il présente cependant l'inconvénient d'engendrer un décalage de trésorerie significatif, avec une entrée en trésorerie en fin d'année $n + 1$ pour des développements réalisés plus tôt en année n .

Le modèle du revenu récurrent autour du logiciel

Des revenus récurrents assurent la pérennité de l'entreprise : c'est souvent en priorité ce type de revenus que

l'éditeur va chercher à développer. On peut d'ailleurs noter que **la taille des clients à qui se destine le logiciel est structurante pour la capacité à le monétiser : utilisateurs grands comptes ? Entreprises ? Particuliers ?**

L'éditeur open source pourra principalement développer cinq sources de revenus récurrents :

- le support forfaitaire,
- la maintenance (accès à des correctifs),
- les garanties (de fonctionnement, juridiques, etc.),
- les services complémentaires (en SaaS notamment),
- les partenariats (payants) avec les intégrateurs.

Le support

L'éditeur propose un service de support de ses produits : réponses garanties aux questions que peuvent se poser les équipes de développement et d'exploitation pendant les phases de réalisation puis de production du projet. Généralement, plusieurs niveaux de support sont proposés, les plus élevés impliquant une disponibilité 24h/24 et 7j/7 des équipes de support, ce qui a un impact sur l'organisation du travail dans la société (astreintes, équipes distribuées sur plusieurs fuseaux horaires).

La maintenance corrective

La maintenance corrective d'un logiciel générique consiste en la correction des bogues du logiciel et la mise

à disposition de correctifs (*patches*), notamment par rapport à des versions officielles (*releases*). L'éditeur espère prévenir tous problèmes et bogues car ce service mobilise des équipes pour résoudre les problèmes techniques.

Dans ce contexte, il est important de **définir précisément la politique de maintenance et de releases du logiciel**, notamment la durée pendant laquelle une version est maintenue. Une durée de 3 ans pour les applications d'entreprises paraît un minimum. Pour certaines applications industrielles, des durées de plus de 10 ans sont courantes.

Le fait d'assurer la maintenance (et le support, d'ailleurs) de versions plus anciennes que la version de développement d'un logiciel (autrement dit de toutes les versions déployées en clientèle !) a un coût en ressources de développement, assez élevé. Là encore, c'est une discipline importante de l'éditeur que de savoir gérer ces contraintes de manière harmonieuse.

La garantie

La garantie que propose l'éditeur représente pour les clients une assurance (dans une certaine mesure au même titre que le support et la maintenance) du bon fonctionnement, voire de la conformité de sa solution. Elle peut servir de socle légal en cas de problème logiciel majeur.

Dans certains contextes où une certification par un organisme tiers est exigée, c'est une façon de différencier une version « communautaire », non certifiée, d'une version « entreprise » certifiée. Il va de soi que cette certification a un coût, en général élevé, et représente là encore un investissement initial qui ne peut être envisagé que si les perspectives de retour sur investissement le permettent.

Les services complémentaires

Ces services peuvent être fournis notamment sous forme de SaaS. Il peut s'agir par exemple pour l'éditeur de mettre à disposition une assistance aux montées de version (conserver le système du client en production tout en bénéficiant des améliorations), de faire bénéficier à ses clients d'une place de marché proposant des extensions, ou encore d'outils de configuration et de gestion du logiciel dans le cloud.

Les partenariats payants avec les intégrateurs

L'éditeur peut proposer des partenariats payants aux intégrateurs qui leur donnent accès à un ensemble de ressources dont ils peuvent avoir besoin : formation, certification, une plus grande souplesse dans la maintenance des différents environnements qu'ils gèrent pour les clients finaux, etc.

Le marketing et la vente jouent aussi un rôle primordial dans ces partenariats : les intégrateurs partenaires, en fonction de leur niveau, peuvent bénéficier d'un accès à des listes de prospects générés par l'éditeur, s'afficher comme partenaires officiels et donc remporter des appels d'offres contre des concurrents qui ne bénéficient pas de ce soutien, pour peu que le client final ait été éclairé sur l'intérêt de tels partenariats.

Ces partenariats sont donc à la fois une source de revenus pour l'éditeur, mais aussi voire surtout **un instrument essentiel de sa politique de développement commercial.**

Les compétences clés de l'éditeur open source

Une des raisons d'être de l'éditeur open source est d'assurer le lien entre le marché et la communauté qui se constitue autour de son produit. L'éditeur produit un logiciel autour duquel évoluent une communauté de développeurs (équipe de développement interne à l'éditeur et contributeurs externes) et une communauté d'utilisateurs (qui, pour ajouter à la complexité du système, sont parfois eux-même des contributeurs), qui ont **des attentes différentes en termes de rythme de release, d'accompagnement, d'outils, de garanties (SLA) et de technicité.**

L'éditeur doit donc créer et partager une vision, être le gardien du temps, être garant de la qualité et être au centre d'un écosystème d'innovation ouverte.

La vision

La vision d'un produit ou d'une architecture innovante est rarement produite par un comité. En s'appuyant sur sa connaissance des enjeux technologiques et des besoins de son marché cible, **l'éditeur développe une vision qu'il doit ensuite partager avec tous les intervenants de son écosystème** pour la transformer en réalité.

Le rôle de gardien du temps

L'un des rôles fondamentaux d'un éditeur de logiciel – celui qui lui permet de réellement créer de la valeur pour son marché – est d'être le gardien du temps, à plusieurs niveaux :

- temps de réponse aux besoins du support ou de la maintenance, en fonction des niveaux de service (SLA) sélectionnés : établir des priorités entre les demandes clients, veiller à la réactivité des réponses ;
- temps des développements prévus dans la feuille de route fonctionnelle et technique : rythme de publication des versions, timing de la publication de certaines fonctionnalités et gestion des dépréciations et de la dette technique (voir ci-après) ;

- cycles de vie des versions publiées et maintenues, des mises à jour d'une version à l'autre (incluant la migration des données), en fonction de la politique de maintenance et de support.

La qualité

La qualité du logiciel open source est assurée notamment par la réalisation de tests réguliers (vérification du bon fonctionnement des logiciels sur les versions relasées comme en intégration continue), la traçabilité, la gestion de la dette technique, la création de documentations.

La **dette technique** est une métaphore du développement logiciel inventée par Ward Cunningham. Elle assimile à une *dette* un écart entre les développements techniques réalisés effectivement, sous contraintes des compétences de l'équipe, des connaissances disponibles et des ressources en général limitées disponibles, et une version idéale du même logiciel qui aurait pu être réalisée si toutes ces contraintes avaient disparu. **Les intérêts de cette dette sont les coûts liés à la maintenance, dans le temps, de versions « désynchronisées » d'un même logiciel.** La gestion de la dette technique pour un éditeur est donc bien une problématique d'arbitrage des ressources et du temps pour maintenir le rythme fixé de progression du logiciel et, par là même, sa qualité.

Être au centre d'un système d'innovation ouverte

L'éditeur a ce rôle crucial d'animation pour fédérer l'écosystème dans son ensemble, entre les académiques fournisseurs de fondamentaux technologiques, les développeurs et contributeurs, prestataires, et les utilisateurs.

Innovation happens elsewhere est le titre d'un livre de Richard Gabriel de 2005 sur l'open source en tant que stratégie business. Ce titre sonne comme un constat, celui que **l'innovation, tant vantée comme source d'avantage compétitif, n'est plus l'apanage d'organisations isolées mais le résultat de la confrontation des points de vues**, de l'enrichissement des idées et de la mise en œuvre d'une intelligence collective par des ensembles d'acteurs, partageant l'impulsion commune de faire émerger de nouvelles idées, technologies, processus, produits, etc.

L'innovation ouverte ou open *innovation* est une démarche qui formalise ces idées. Un prérequis fort, bien sûr, est que tous les acteurs qui y participent y trouvent leur compte. De ce point de vue, **l'open source est sans doute l'incarnation la plus fidèle de ces idées, en raison de la très grande transparence qui le caractérise** : les idées, en l'occurrence réalisées dans le code source des logiciels et dans les différentes activités sociales qui gravitent autour de la création de code (forums, listes

de discussion, outils de suivi de bugs, etc.) circulent librement entre les acteurs, la participation de tous est encouragée.

Une autre forme d'innovation ouverte est constituée par les projets collaboratifs, notamment ceux des pôles de compétitivité tel Systematic, qui associent en général PME innovantes (éditeurs de logiciels libres, intégrateurs spécialisés), grands comptes utilisateurs et institutions académiques, autour de projets spécifiques financés par l'État, les collectivités territoriales, ou l'Europe. Il est bien sûr recommandé d'articuler ces projets avec les écosystèmes d'innovation plus larges mais plus informels évoqués plus haut.

Compte-tenu de l'avantage concurrentiel que représente l'existence d'un écosystème dynamique d'innovation ouverte autour de ses produits, **l'un des cœurs de métier de l'éditeur open source est de savoir animer ces communautés, en équilibrant l'attention à donner aux idées et aux besoins de chacun et la nécessité d'avancer ensemble dans la même direction**, conformément à la vision, aux contraintes de gestion du temps et de qualité évoquées précédemment.

Avantages et faiblesses

Avantages

Les principaux avantages du modèle d'éditeur open source par rapport à un modèle propriétaire sont :

- un développement plus efficace ;
- un marketing et une évangelisation plus simple (puisque la solution est librement « testable ») ;
- des cycles de vente plus courts (car les équipes commerciales peuvent commencer à intervenir une fois que les clients ont commencé à tester la solution en interne) ;
- la constitution d'un écosystème.

Inconvénients

L'un des principaux inconvénients du modèle réside dans **la difficulté du financement par les investisseurs (*venture capitalists*), qui attendent souvent des barrières fortes à l'entrée et à la sortie pour justifier un retour sur investissement.**

En effet, les barrières à l'entrée pour les concurrents sont plus basses : concrètement, on peut utiliser la technologie librement et se la réapproprier, pour peu qu'on en ait les compétences ; ainsi les intégrateurs peuvent décider de se passer d'éditeurs (ce qui représente une menace à terme sur la pérennité de l'ensemble de l'écosystème).

De manière similaire, les barrières à la sortie sont plus basses pour les utilisateurs : c'est à la fois un argument commercial des éditeurs open source face à leurs concurrents propriétaires, mais aussi pour eux un risque plus élevé de voir des revenus récurrents se tarir à moyen terme.

De manière plus positive, on peut voir ces derniers points comme une incitation plus forte pour les éditeurs open source à jouer sur la qualité des produits et des services associés, plutôt que de se contenter d'en extraire des rentes de situation comme le font certains éditeurs propriétaires.

Enfin, on peut noter que la valeur créée par un éditeur open source ne réside pas seulement dans son produit, mais également dans sa capacité à le mettre en œuvre par des services associés et, surtout, par l'écosystème qu'il est capable de constituer autour de son produit, à la fois par sa taille, sa qualité, et bien sûr, dans une optique commerciale, par les flux de trésorerie qu'il peut engendrer vers l'éditeur.

Dérives possibles

Confronté à la difficulté de s'assurer un revenu récurrent dans ce modèle, l'éditeur open source peut être tenté par les dérives suivantes, qui l'écartent du modèle et freinent la constitution d'un écosystème :

- la migration d'une offre d'open source professionnel à une offre d'*open core* ;
- la négligence du soutien à la communauté (ex. : documentation insuffisante, peu d'interventions sur les listes de discussion, etc.) ;
- la négligence de la qualité de l'offre pour justifier la maintenance ;
- le souhait de travailler seul plutôt qu'en partenariat, notamment avec les intégrateurs ;
- le manque de transparence (sur la vision, la feuille de route, etc.).

Ces dérives peuvent se manifester, au niveau organisationnel, par des défauts d'alignement entre les services de l'entreprise (marketing, commercial, développement, support...).

Il est donc important pour le CEO d'un éditeur open source de **piloter au plus fin les compromis entre croissance du revenu à court terme et croissance de l'écosystème, générateur de revenus à moyen terme**, et de mettre en place les mécanismes d'incitation et de contrôle qui évitent ces dérives.

La modélisation du revenu récurrent

Les revenus récurrents sont un élément majeur du revenu mix d'un éditeur. Ils ont de plus l'avantage de pouvoir être modélisés de manière assez simple, par rapport aux revenus du service pur, par nature conjoncturels et liés aux performances de l'équipe commerciale.

Comment se modélise le revenu récurrent et quel est son impact sur l'écosystème ?

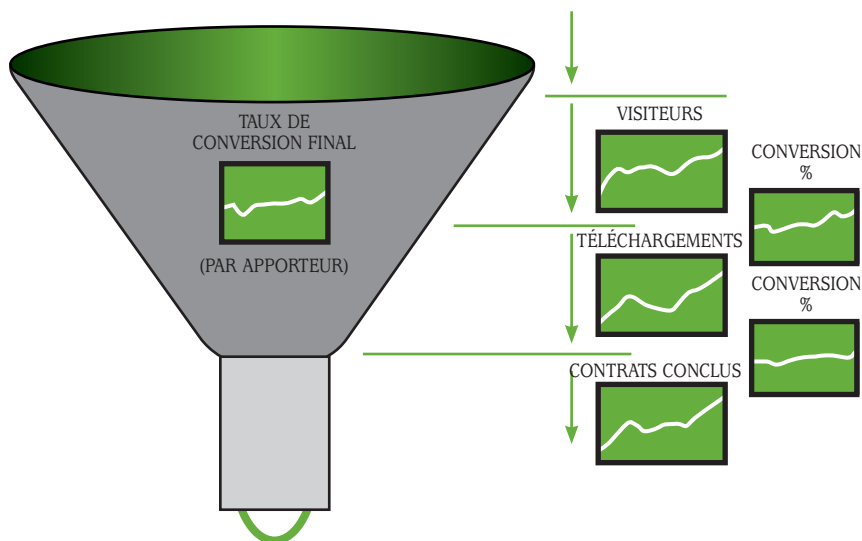
Le modèle classique est celui de l'entonnoir, que l'on peut résumer, dans sa forme la plus simple, de la manière suivante :

1. Les équipes de web marketing effectuent le travail nécessaire pour attirer des visiteurs sur le site internet et les inciter à télécharger une version d'évaluation ou effectuer une évaluation en ligne.
2. Une équipe commerciale entre ensuite en action pour faire signer les clients.
3. À chaque étape du processus de conversion, des visiteurs sortent de l'entonnoir et d'autres y entrent.

Au bout de l'entonnoir, il reste les visiteurs convertis en clients finaux.

Le coût total d'acquisition d'un nouveau client peut être facilement calculé en tenant compte des dépenses de

marketing et du temps passé par l'équipe commerciale sur chaque prospect, ainsi que des différents taux de conversion.



Source : d'après The Open Source Business Model: Key Metrics and Levers, David Skok

Par ailleurs, une fois les clients acquis, ceux-ci peuvent décider de ne pas renouveler leur abonnement à un moment donné. On parle de *taux d'attrition* pour désigner la proportion d'utilisateurs qui arrête leur abonnement à la fin d'une période donnée, par exemple mensuelle. Ce taux, qu'on voudra le plus bas possible (qui peut même « s'inverser » si on prend en compte la possibilité d'*upsell* sur des clients existants), dépend principalement de la qualité des produits et des services fournis par l'éditeur, ainsi que du contexte concurrentiel.

Les variables clés qui permettent de modéliser l'acquisition des utilisateurs/clients sont donc les suivantes :

- le CAC (*customer acquisition cost*), coût d'acquisition moyen d'un client ;
- le MRR (*monthly recurring revenue*), revenu mensuel récurrent moyen ;
- le MCC (*monthly customer churn*), taux d'attrition mensuel moyen (taux de perte des clients).

À partir de ces trois variables, on définit :

- la LTV (*lifetime customer value*), qui représente le revenu total moyen par client sur toute la durée pendant laquelle il reste client, et qui vaut précisément : MRR/MCC .

Par exemple, supposons que le coût d'acquisition d'un nouveau client est de 5 000 € :

CAC	5 000 €	Coût d'acquisition d'un client
MRR	500 €	Montant de l'abonnement mensuel payé par le client
MCR	2,5 %	Taux d'attrition (perte en % de clients par mois)
LTV = MRR / MCR	20 000 €	Lifetime customer value (abstraction faite des coûts)

Dans cet exemple, on paie donc 5 000 € pour un retour total attendu de 20 000 € par client (*lifetime customer value*), ratio a priori intéressant. Mais en termes de trésorerie, il reste une difficulté : avec cet exemple à 500 € d'abonnement par mois, l'éditeur ne récupère son inves-

tissement qu'après 10 mois, ce qui nécessite d'assurer un fond de roulement non négligeable.

L'une des solutions possible à ce dernier problème est d'obtenir des clients qu'ils paient en avance, notamment par l'incitation de remises. L'autre est de faire appel à des mécanismes de préfinancement par des établissements de crédit.

Chapitre 4

Le modèle des intégrateurs

Au modèle économique de l'éditeur open source répond celui de l'intégrateur open source. Ce dernier collabore avec toutes les parties prenantes de l'environnement : communautés, éditeurs et clients ; il joue un rôle essentiel dans l'écosystème.

Le modèle économique de l'intégrateur open source

L'intégrateur open source tire ses revenus de l'intégration et du support de solutions open source qu'il réalise pour des entreprises et des administrations. **Son métier consiste à proposer des services permettant d'exploiter les logiciels libres sur la totalité du cycle de vie.** Il évolue dans un écosystème composé de deux grands types de partenaires : les clients (administrations, tertiaire, milieu associatif, grands comptes industriels et PME) et les fournisseurs de technologies (éditeurs, communautés, milieu académique).

Il existe deux modèles d'intégrateur : le modèle avec éditeur et le modèle sans éditeur.

Le modèle sans éditeur

L'intégrateur gère pour le client l'intégralité du projet open source, maintenance et support inclus. **L'intégrateur se tourne ainsi vers la communauté pour récupérer le code source, réalise les développements nécessaires au client, puis redistribue tout ou partie de ses développements à la communauté.**

L'intégrateur assure trois types de fonctions auprès du client : le conseil, l'intégration et la tierce maintenance applicative (TMA).

- **Le conseil** : l'intégrateur conseille l'entreprise dans le choix des solutions open source et des modules. Il peut également assurer les activités d'audit, d'expertise technique, d'assistance à maîtrise d'ouvrage (AMOA) et de formation.
- **L'intégration** : l'intégrateur intègre la solution open source dans l'environnement de l'entreprise, recourt à des extensions déjà développées par la communauté et/ou développe des extensions complémentaires qui pourront être reversées à la marketplace de la communauté.
- **La tierce maintenance applicative** : l'intégrateur réalise la maintenance et le support de la solution complète qu'il a développée pour son client.

Sans lien avec les éditeurs, il peut prendre la posture d'un arbitre neutre sur le choix des technologies.

Le modèle avec éditeur

L'intégrateur va redistribuer une partie des revenus à l'éditeur (qui développe le logiciel et en assure la maintenance) ; l'éditeur aussi devient prestataire du service pour le client. Ici, l'intégrateur n'assure qu'une partie des fonctions de conseil et de TMA. S'agissant du conseil, il n'a plus les rôles d'audit, d'expertise technique et de formation, qui reviennent à l'éditeur.

Pour la TMA, il assure essentiellement la maintenance du logiciel, le support étant fourni par l'éditeur.

Différences entre intégrateur open source et intégrateur propriétaire

Le métier d'intégrateur, qu'il soit open source ou propriétaire, reprend les bases du métier de service :

- conseil et expertise,
- ingénierie et projets forfaitaires,
- support, tierce maintenance applicative (TMA),
- assistance technique,
- formation.

L'intégrateur propose également ses compétences en développement de logiciel : la méthodologie de développement et le respect des standards de qualité.

Il assure également la gestion des technologies de base classiques : langages (C, C + + , Java, PHP) et technologies (OS, base de données, SOA, etc.).

La différence entre intégrateurs open source et intégrateurs propriétaires est d'une part de nature commerciale, d'autre part dans la nature du projet (voir tableaux ci-après).

Tableau 1. Les différences commerciales

Intégrateur open source [avec éditeur]	Intégrateur propriétaire
<ul style="list-style-type: none"> • Des coûts d'entrée nuls [sauf si partenariat payant] • Absence de [Faible] barrière à l'entrée et à la sortie • Un prix moins élevé ? • Pas de [partage de revenu] • [Le marché « gris »] • Efficacité du discours commercial • Cycle de vente plus rapide • [Avant-ventes possibles avec l'éditeur] • [Communication/marketing portée par l'éditeur] • Diversité possible de l'offre • [Choix de licence suivant l'éditeur] 	<ul style="list-style-type: none"> • Un pipe tiré par l'éditeur • Des démarches d'avant-vente communes • Démarche communication/marketing portée par l'éditeur

Tableau 2. Les différences de projet

Intégrateur open source [avec éditeur]	Intégrateur propriétaire
<ul style="list-style-type: none">• Richesse fonctionnelle• Qualité des contributions hétérogène [homogénéisée par l'éditeur]• Support communautaire non garanti [support éditeur garanti]• Maîtrise de l'environnement• [Roadmap influençable]• [Visibilité de la roadmap]	<ul style="list-style-type: none">• Support garanti• Roadmap influençable• Visibilité de la roadmap

Les clés de l'intégrateur open source

Le modèle économique de l'intégrateur sur le marché de l'open source s'appuie sur cinq fondamentaux.

Le lien étroit avec les communautés et les éditeurs

En open source, la technologie est fournie par des communautés et des éditeurs avec lesquels l'intégrateur a tout intérêt à créer et entretenir des liens. En effet, les éditeurs réalisent un travail de valorisation de la technologie open source (communication, évènements, démonstrations etc.) à laquelle est associée l'activité de l'intégrateur. Ils peuvent également recommander à leurs

clients des intégrateurs avec lesquels ils ont collaboré et pu créer des relations professionnelles.

Ces relations de partenariat permettent de générer des opportunités de marché. De la même manière, créer des liens étroits avec les communautés représente pour l'intégrateur une garantie d'efficacité.

Ressources humaines : le bon accueil de profils open source

Les intégrateurs peuvent accéder à des profils d'ingénieurs attractifs (passionnés et souvent militants) dans le vivier des contributeurs de la communauté. **Cela leur confère un avantage en termes d'offre, dans la mesure où ils peuvent mettre en avant le savoir-faire et l'expertise open source de leurs collaborateurs.**

Une expertise avérée sur les technologies de pointe

Les technologies spécifiques sont un enjeu de taille pour l'open source : très en pointe mais encore mal connues, elles nécessitent de fortes expertises et des soutiens financiers importants. À titre d'exemple, on peut citer le temps réel sous Linux, l'informatique scientifique, le traitement d'images ou l'émulation de plates-formes embarquées.

Présenter une expertise avérée sur l'une de ces technologies constitue un avantage concurrentiel décisif et différenciant pour les intégrateurs. Cependant, ces technologies doivent être soutenues par une R&D spécifique et réclament des ressources financières importantes pour arriver à maturité, ce qui en constitue un frein à l'adoption.

Un environnement technologique surveillé

L'open source présente une très grande variété de projets, de frameworks, etc. Pour l'intégrateur open source, il ne s'agit pas de convaincre que sa technologie de prédilection est la meilleure, mais d'identifier les composants qui répondent au mieux à un besoin spécifique. **La veille technologique, à la fois celle formelle et structurée alimentée par les projets des clients et la R&D, mais aussi celle plus spontanée réalisée par les collaborateurs, représente un réel atout** pour l'intégrateur car une très bonne connaissance de l'environnement open source lui confère crédibilité et différenciation.

L'offre open source est technocentrée

Les logiciels open source ne sont pas développés pour répondre à un besoin client, l'activité de R&D est donc nécessaire pour structurer un ensemble de composants technologiques et créer des solutions verticales. **Dans cet**

écosystème, l'intégrateur doit faire le lien entre une technologie et un métier, et entre une méthodologie industrielle et des outils techniques. C'est en développant des compétences métier que l'intégrateur répond directement aux besoins de ses clients. C'est pourquoi la maîtrise technologique représente un réel facteur de différenciation.

Risque inhérent au modèle économique de l'intégrateur

Le modèle d'affaires de l'intégrateur est fondé sur un pari : la diffusion de la technologie open source. Il lui est donc nécessaire de contribuer à cette diffusion, avec le risque inévitable de la banalisation de cette diffusion, qui signe la fin du critère différenciant d'un intégrateur donné.

Chapitre 5

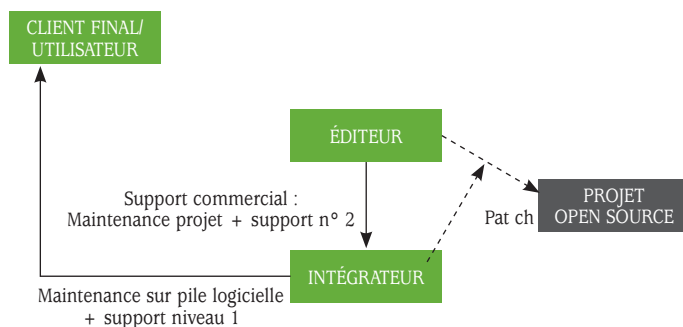
Relation entre les acteurs : enjeux et complexité

L'éditeur et l'intégrateur peuvent choisir de coopérer – ou pas. Sachant leurs apports respectifs à la chaîne de valeur, **il est indispensable d'instaurer une coopération en bonne intelligence, source de synergies pérennes et créatrices de valeur.**

Plus précisément, ils doivent s'assurer d'un partage équitable des budgets des clients sur le marché. Or, il est parfois difficile pour un client de ne pas les percevoir comme des concurrents. L'intégrateur peut être tenté de développer sa propre offre de logiciel pour assurer des revenus complémentaires, empiétant ainsi sur le marché de l'éditeur. Ils sont rares cependant à opter pour cette option, qui peut brouiller leur positionnement et causer une perte de crédibilité auprès des clients, l'intégrateur quittant sa position de neutralité au détriment des autres solutions du marché. De la même manière, on constate parfois une ambiguïté autour des services proposés par les éditeurs et les intégrateurs (le support, la maintenance, la formation...).

La bonne entente entre ces acteurs de l'open source résiderait donc dans leur capacité à s'en tenir à leurs domaines de compétences respectifs et ainsi garantir une dynamique aussi bien technique que commerciale, telle que dans le modèle de la *maintenance globalisée vertueuse* qui propose une complémentarité saine de l'offre de l'éditeur et de l'intégrateur.

Dans ce modèle, l'éditeur et l'intégrateur open source se partagent l'activité de maintenance proposée au client. L'éditeur réalise le support commercial qui est composé de la maintenance du projet et du support de niveau 2. L'intégrateur, lui, assure la maintenance sur le logiciel et le support de niveau 1.



Plus généralement, résumons comment se répartissent les apports, les prestations et les efforts et bénéfices de commercialisation de logiciels libres.

Tableau 3 Apports respectifs des acteurs principaux : fondation, éditeur, intégreur

	La fondation...	L'éditeur...	L'intégrateur...
Apports	Crée les sources du logiciel...	En crée une solution « industrialisée »...	Accompagne le client et réalise l'intégration de la solution logicielle dans l'entreprise.
Prestations	Réalise un travail de R&D de projets open source génériques (<i>working groups</i>)...	L'adapte aux besoins métiers des entreprises (rôle d'expertise) ; produit des logiciels, les développe et assure le suivi et la formation des intégrateurs.	Intègre ensuite le logiciel dans l'entreprise.
Version commerciale	Héberge et promeut les projets open source...	Utilise les travaux issus de ces projets pour produire et vendre des logiciels adaptés aux besoins des entreprises ; fournit une solution prête-à-l'emploi à l'intégrateur et en assure la garantie.	Vend comme prestation le travail d'intégration de la solution dans l'entreprise ainsi que la maintenance (mise à disposition de patches de résolution de bugs).

À PROPOS DU GROUPE THÉMATIQUE LOGICIEL LIBRE DE SYSTEMATIC (GTLL)

Créé en 2007, le groupe thématique Logiciel Libre du pôle Systematic Paris-Region forme l'un des principaux viviers de l'Open Source en France.

Avec pour mission de «développer l'écosystème du libre en Île-de-France», le GTLL regroupe plus d'une centaine d'acteurs de l'innovation ouverte (PME, ETI, grands groupes et académiques). Il vise à favoriser la coopération, l'innovation et l'emploi, autour de projets de R&D collaborative et grâce à des actions de soutien au développement des entreprises innovantes (promotion, marketing, stratégie, aide à la recherche de financements...), dans le cadre des principes et des valeurs de l'open source. Il est à ce jour le plus important cluster au monde à focaliser ses activités de R&D collaborative sur les logiciels libres et les défis spécifiques à l'open source, comme l'after PC (l'ère informatique du Cloud, des mobiles et des objets connectés), la qualité logicielle, et le déluge des données. Après 7 ans d'existence, 40 projets de R&D collaborative consacrés au logiciel libre représentant un effort de R&D de près de 160 M€ ont déjà été financés grâce à l'aide du GTLL.

Le logiciel libre n'échappe pas à la mécanique économique, bien au contraire. Les acteurs de l'écosystème open source nourrissent un intérêt économique, quoique de nature diverse : perception de revenus, ressources humaines, mutualisation de R&D.

Nous proposons ici une vue globale et microéconomique du logiciel du libre, l'éditeur et l'intégrateur reflétant l'essentiel des modèles économiques de l'open source.

Bien que mature, le marché est soumis à des problématiques de captation et de création de richesse, et repose sur des rouages complexes de partage des revenus. Autant de défis à relever pour développer durablement un marché et un écosystème, mouvement que le **GT Logiciel Libre de Systematic** est décidé à soutenir.

Ce guide sera lu avec profit par tous ceux qui souhaitent comprendre les mécanismes économiques du logiciel libre et de l'open source.

Le **groupe thématique Logiciel Libre** forme l'un des principaux viviers de l'open source en France. Il rassemble startups, PME, grands groupes, universités et centres de recherche autour d'une même vision des défis technologiques de demain et d'un engagement profond pour la compétitivité de notre économie.