

2.2. Impacts sur la relation client

Publié sur Guide Open Source (<http://guideopensource.info>)

2.2. Impacts sur la relation client

Par le groupe de travail Syntec Informatique-FniLL

Créé le 10/04/2010 - 16:13

Soumis par [Benjamin Jean](#) [1] le lun, 10/04/2010 - 16:13

Les business model construits autour de Logiciels Libres, dans la mesure où ils comprennent l'accès au [code source](#) [2] et à l'exploitation du logiciel, modifient la relation client¹.

Le Logiciel Libre permet de répondre à certaines demandes du client, demandes relatives à la pérennité de son système d'information et à l'indépendance vis-à-vis d'un fournisseur. Néanmoins, un véritable travail d'explication et de sensibilisation est nécessaire pour tempérer certaines demandes inadaptées (tirées de l'expérience passée de certains clients), et rechercher la solution optimale pour tous.

L'enjeu est ici de permettre au client d'évaluer l'impact potentiel d'une utilisation de l'Open Source en lieu et place d'un [composant](#) [3] traditionnel. Même si l'impact peut s'avérer moindre que ce à quoi le client s'attend, il n'est pas dénué d'effet).

- ¹. À l'exception du [Saas](#) [4], ainsi que précisé supra.

2.2.1. L'expression des demandes

Les clients demandent souvent à disposer des « sources » (comprendre le [code source](#) [2]) ou bien encore à se voir attribuer l'ensemble des droits patrimoniaux attachés au logiciel.

Dans la logique propriétaire, l'éditeur titulaire des droits d'auteur ne donnera que très rarement accès au [code source](#) [2] et le compromis réside souvent dans la mise sous séquestre de ce [code source](#) [2]. La question est résolue d'office en matière de Logiciel libre puisque chaque utilisateur se voit transférer un logiciel, avec son [code source](#) [2] et la liberté (notamment) de le modifier.

Pour autant, il est impératif de connaître les motivations des clients afin de leur fournir la meilleure réponse et la meilleure solution technique. Ainsi, il ne suffira pas d'avoir accès au [code source](#) [2] ou d'être cessionnaire des droits pour assurer la pérennité du système si tel est l'objectif. Encore faudra-t-il être en mesure d'assurer la continuité.

[Back to Top](#)

2.2.2. Comprendre les motivations des demandes

Pourquoi le client demande-t-il à être propriétaire des « sources » ou au moins à en disposer ?

De façon classique, les demandes relatives au [code source](#) [2] ou à la propriété intellectuelle sont souvent animées par un souci d'ordre sécuritaire, d'assurer la pérennité des systèmes. Ces demandes seront d'autant plus insistantes que l'enjeu est lourd pour le client et que le risque de défaillance du projet est susceptible d'avoir des répercussions importantes sur le fonctionnement de l'entreprise. On ne peut blâmer un client préoccupé de savoir comment fonctionne (et fonctionnera) son logiciel. Le logiciel est un outil clé de l'entreprise, il fait tourner les ordinateurs, les téléphones et de nombreux appareils. Pour autant, disposer du [code source](#) [2] ou des droits de propriété intellectuelle permet-il d'atteindre l'objectif fixé ?

Il est nécessaire de rappeler au client que le transfert de propriété implique également transfert

2.2. Impacts sur la relation client

Publié sur Guide Open Source (<http://guideopensource.info>)

des droits et devoirs attachés à cette propriété. Ainsi, le client n'aura plus nécessairement intérêt à rester propriétaire de l'application : en adoptant un logiciel propriétaire, il bénéficiera d'une maintenance évolutive mutualisée par la [SSII](#) [5] pour tous les utilisateurs concernés ; en optant pour un Logiciel Libre, il aura accès, en plus de la maintenance évolutive mutualisée par une [SSLL](#) [6]¹, aux évolutions et améliorations apportées par la communauté (directement ou par le biais de son prestataire).

- ¹. À l'opposé, le choix de l'Open Source peut aussi permettre de maîtriser le cycle de vie du logiciel, sans support dégradé ou montée de version forcée

[Back to Top](#)

2.2.2.1. La préservation du savoir-faire

Autre explication : dans son cahier des charges, le client transfère beaucoup de son savoir-faire métier à la société de services. Son système d'information contribuera à le différencier de ses concurrents et à en tirer avantage. Par exemple, il peut s'agir de systèmes de fidélisation des clients, de méthodes originales de production, etc. Très logiquement, il ne souhaite pas que son savoir faire soit communiqué à ses concurrents – ce qui pourrait être le cas si le prestataire réutilisait des développements spécifiques pour compléter sa propre offre. Nombre de logiciels sont effectivement nés de la collaboration entre un éditeur et un client.

Notons que l'impact des licences Open Source, même copyleft, sera inexistant en l'espèce puisque :

- Dans le cas d'un développement spécifique, seul le client bénéficie de la cession « contrainte » des droits sur le logiciel : à sa charge ensuite de décider d'en limiter la diffusion ou de l'exploiter (les problèmes peuvent néanmoins ressurgir en cas de division de ses activités, acquisition partielle, etc.);
- Dans le cas d'une maintenance, d'un support ou d'une assistance réalisés sur les machines du client et sous son contrôle, alors le logiciel n'est pas distribué et le client n'est pas contraint de donner accès au [code source](#) [2]¹.

D'où la nécessité, d'identifier en amont d'une part les objectifs du client et d'autre part les composants logiciels.

Plusieurs réponses sont alors possibles, dont une relative aux licences Open Source :

- Commencer par identifier clairement les composants et/ou développements correspondants aux demandes spécifiques du client. Ils seront alors distingués et un éventuel accord de transfert pourra les viser. Le fournisseur pourra ainsi écarter de la discussion tout ce qui concerne les Logiciels Libres, modifiés ou non.
- Ensuite, proposer des solutions alternatives au transfert de propriété comme une clause d'exclusivité sur le marché considéré, pour une durée et une étendue géographique à déterminer avec le client, par exemple, assurant au client qu'il ne retrouvera pas son savoir-faire chez un concurrent. C'est ici le fournisseur/l'éditeur qui s'oblige le plus, et donne plus de droits à son client, ce qui est tout à fait conforme au texte des licences.
- Transférer le [code source](#) [2] sous licence Open Source pour garantir au client non seulement la disponibilité des sources correspondantes, mais encore la possibilité de les utiliser, modifier, développer à sa convenance et sans contrainte². Cette solution permet aussi d'assurer la continuité en cas de disparition du prestataire.
- Transférer les composants spécifiques en pleine propriété (c'est-à-dire céder la titularité des droits). En pareil cas, il convient d'identifier avec le client quels sont les droits patrimoniaux à transférer pour répondre à son attente. Par exemple, droits de commercialisation sur certains marchés ou zones géographiques, droits d'utilisation dans certains contextes, etc. Cela permettra également au client de bien identifier son « vrai » besoin et, au fournisseur, d'y

2.2. Impacts sur la relation client

Publié sur Guide Open Source (<http://guideopensource.info>)

apporter des réponses appropriées en appauvrissant aussi peu que possible son propre potentiel professionnel.

- [1.](#) Certaines licences viennent d'ailleurs préciser ce point et l'étendre : voir notamment l'article 2 de la [GNU GPL](#) [7] v3 « You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you. »
- [2.](#) Au risque de la création d'un [fork](#) [8] si le projet est ensuite maintenu par une autre société et dans un autre sens.

[Back to Top](#)

2.2.2.2. Autonomie vis-à-vis du fournisseur

Il s'agit pour l'utilisateur d'assurer la pérennité globale de son système et donc de la problématique de dépendance envers ses fournisseurs, de leurs disponibilités pour la maintenance et/ou les développements ultérieurs, de la cohérence du système contractuel et technique (avec notamment des enjeux de cohérence entre les divers choix en matière de licences Open Source).

Par exemple, il arrive fréquemment qu'un Grand Compte détecte une solution logicielle qui lui convient parfaitement, mais dont le porteur lui semble être une société fragile financièrement. Il aura tendance à se tourner vers un Intégrateur Système qui prendra en charge la cohérence de l'ensemble et qui en sera responsable en maintenance et en suivi.

Le client veut pouvoir s'affranchir de la dépendance d'un fournisseur en cas de défaillance de celui-ci et éviter tout litige commercial, désaccord sur le prix des prestations, etc. Il souhaite donc disposer de l'ensemble des sources et ressources lui permettant d'assumer, par lui-même ou avec l'aide d'un sous-traitant, la continuité de service et les développements. Comme toute médaille à son revers, il perdra alors beaucoup de poids vis-à-vis de son fournisseur qui se sentira libre de s'affranchir, au moins partiellement ou temporairement d'obligations de maintenance et de suivi, au motif qu'il n'est plus indispensable à son client, celui-ci disposant de toute la « matière première » nécessaire. Le fournisseur peut être tenté de mettre ses moyens en priorité vers les clients qui dépendent totalement de lui.

Pourtant, si le client externalise ses développements, leur maintenance et même leur hébergement ; c'est bien pour ne pas avoir à le faire par lui-même. Il ne dispose généralement pas des ressources ou des compétences et connaissances nécessaires. Il restera donc, au moins à court terme, dépendant de son prestataire.

Il est possible au client de demander un transfert de l'ensemble des composants (composants matériels, composants plates-formes et composants spécifiques), dans le cadre d'une licence Open Source. Il disposera ainsi de l'ensemble des sources - y compris des plates-formes de développement et de déploiement Open Source - nécessaires pour assurer par lui-même, ou avec l'aide d'un autre prestataire, la pérennité de son système. Les prestataires qui fournissent ce genre de solutions facturent essentiellement leur expertise en matière d'architecture, de choix de composants, d'assemblage et de validation desdits composants, leur déploiement, leur paramétrage¹. Le fait pour le client de disposer de l'ensemble des sources représente non seulement un confort « psychologique », mais une réelle possibilité d'indépendance vis-à-vis de la fourniture de services : il peut ainsi changer de prestataire sans coût disproportionné. L'usage de Logiciels Libres est ainsi parfaitement adapté à ce type de demande.

La mise sous séquestre du patrimoine, accessible au client sous condition, est une alternative disponible tant à l'égard de logiciels libres que propriétaires, elle concernera les composants

2.2. Impacts sur la relation client

Publié sur Guide Open Source (<http://guideopensource.info>)

plates-formes, Framework métier et composants spécifiques client. Le véritable enjeu pour le client n'est pas patrimonial, mais bien opérationnel : il souhaite pérenniser son système d'Information ou de production. Naturellement, le contenu de ce séquestre doit être soigneusement décrit (cf [9]. infra l'annexe technique de description d'une livraison, en cas de séquestre) et surtout testé. Il doit également faire l'objet d'un suivi pour limiter de façon raisonnable la différence entre la dernière version séquestrée et la version en production².

En matière d'Outsourcing et de mutualisation des plates-formes de production (ASP [10] SaaS [4]), les enjeux concernent principalement les services (en matière de Contrat de Niveau de Service³ et de Plan de Reprise d'Activité⁴). Les problématiques sont généralement les mêmes en matière de logiciel propriétaire et de Logiciel Libre - à l'exception des acteurs qui étendent les libertés des utilisateurs à ces usages⁵.

Dans bien des cas, les demandes clients peuvent être traitées par une clause de cession sous licence Open Source (si le prestataire assure le pilotage du projet Open Source), de transfert des sources (s'il peut assurer, ou faire assurer la maintenance), selon le type de **composant** [3] concerné. Toutefois et dans la mesure où l'utilisateur n'a pas précisément de compétence en matière de développement et de maintenance de ladite application, disposer des « sources » peut poser plus de problèmes que cela n'apporte de solutions crédibles à ses questions sur la pérennité de son investissement, et donc de tout ou partie de son système d'information⁶. Le client doit mesurer l'intérêt de disposer des « sources » à l'aune de ses objectifs.

- ¹. S'ajoute le cas échéant la maintenance évolutive.
- ². Option complémentaire : Il est possible de proposer au client que la version initiale comme toute version majeure qui lui sera livrée soit entièrement produite à par tir des seules sources contenues dans le séquestre. Il aura alors l'assurance de la parfaite symétrie entre les « sources séquestrées » et la version déployée. Et ceci est valable aussi bien pour du Logiciel Libre que pour du code propriétaire dans la mesure où il n'y a pas de transfert de propriété. En ce qui concerne le Logiciel Libre, cela permet également d'identifier clairement l'état technique des Logiciels Libres que le fournisseur aura intégrés dans sa livraison et qu'il aura testés et validés en conséquence.

Cette solution permet également d'identifier clairement l'état du patrimoine concerné en cas de litige et notamment en cas d'intervention simultanée sur le fonctionnement du logiciel de la part du client comme du fournisseur. Par exemple, état des structures des bases de données, paramétrages...

Ne disposant pas des sources, l'utilisateur ne peut être mis en cause dans ce type de cas. De même, le prestataire sera assuré qu'aucune intervention intempestive ne sera effectuée par l'utilisateur, mettant en cause le bon fonctionnement de l'application.

Le séquestre pourra également recevoir un certain nombre de documents, procédures, composants qui ne sont généralement pas fournis au client, mais dont la disponibilité est importante en matière de reprise de maîtrise de l'application : règles internes de développement et de nommage, copie des bases de source, dossiers d'architectures, procédures automatisées de mise à jour et de maintenance, etc.

- ³. Aussi appelé **SLA** [11] pour Service Level Agreement.
- ⁴. **DRP** en anglais, pour Disaster Recovery Plan.
- ⁵. Voir supra : Open Source et Cloud Computing.
- ⁶. Et pour être efficace, le client devra former en permanence ses équipes, tant pour la livraison initiale que pour les mises à jour et développements additionnels, ce qui est très coûteux. Il perdrait ainsi les avantages financiers d'une externalisation et d'une mutualisation des coûts, son fournisseur pouvant répartir ces coûts sur tous ses clients, à l'exception des composants spécifiques qui pourront faire l'objet d'un traitement contractuel particulier.

[Back to Top](#)

2.2. Impacts sur la relation client

Publié sur Guide Open Source (<http://guideopensource.info>)

URL source: <http://guideopensource.info/guide/impacts-relation-client>

Liens:

- [1] <http://guideopensource.info/user/1>
- [2] <http://guideopensource.info/glossary/term/4>
- [3] <http://guideopensource.info/glossary/term/3>
- [4] <http://guideopensource.info/glossary/term/62>
- [5] <http://guideopensource.info/glossary/term/65>
- [6] <http://guideopensource.info/glossary/term/66>
- [7] <http://guideopensource.info/glossary/term/33>
- [8] <http://guideopensource.info/glossary/term/8>
- [9] <http://guideopensource.info/glossary/term/26>
- [10] <http://guideopensource.info/glossary/term/16>
- [11] <http://guideopensource.info/glossary/term/64>