

Open Source

# Comment promouvoir son usage dans les entreprises ?

## Introduction

L'adoption de l'open source est en nette hausse dans les entreprises. Selon une étude publiée par Black Duck software début 2015, qui repose sur l'interrogation de plus de 1.300 personnes (professionnels, DSI, responsables métiers...), 78% des répondants déclaraient utiliser des logiciels open source dans leur activité. Cinq ans auparavant, en 2010, seuls 42% des répondants s'appuyaient sur des solutions open source.

Le constat est là. Qui ignore aujourd'hui des noms comme Linux (parmi les distributions les plus connues Ubuntu, Debian, SUSE, Red Hat, etc...) Firefox, Apache, Android, les langages PHP, Python et beaucoup d'autres ?

Or, malgré un succès évident, le déploiement de l'open source dans les entreprises reste assez limité, à l'exception de quelques entreprises comme Google, Amazon et Facebook qui, étant donné le dimensionnement de leur infrastructure et les contraintes économiques, n'ont guère d'autre choix que de déployer de l'open source pour leurs propres besoins.

Pour la plupart des entreprises, en revanche, **le déploiement de l'open source reste freiné par certaines idées reçues ou bien par des questions pertinentes auxquelles on n'a pas encore apporté de réponses appropriées.**

**C'est l'objectif de cette Fiche Pratique de tenter d'y répondre.**



## Définitions

L'open source, un ensemble de logiciels « développés par » et qui « doivent leur évolution à » une communauté d'individus, à des contributions individuelles, à des sociétés éditeurs de logiciels, à des sociétés de services ou bien à des sociétés qui contribuent sous la forme de don de code ou de participation au développement.

On parlera aussi de « logiciel libre », un logiciel open source mais ayant une licence « libre » donnant accès à quatre libertés (définition Free Software Foundation) :

- Liberté d'utiliser le logiciel
- Liberté de le modifier
- Liberté de redistribuer des copies
- Liberté de distribuer des copies des versions modifiées

Dans cette Fiche Pratique, nous parlerons indifféremment de logiciels open source ou libres, le but étant de fournir quelques idées et actions permettant leur promotion.



## Les idées reçues ont la vie dure...

**N°1 - La première des idées reçues est que « libre = gratuit ».**

Elle trouve probablement son origine dans le fait que « free » signifie à la fois libre et gratuit en anglais. Hélas non, la licence logicielle est gratuite mais la mise en œuvre ne l'est pas et le support non plus. Dès lors, à première vue, pourquoi ne pas prendre une solution propriétaire ?

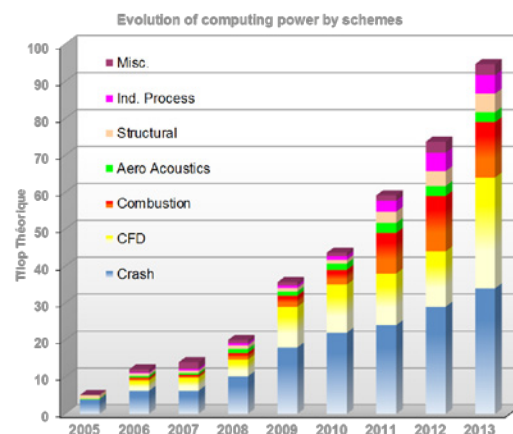
Il s'avère que l'analyse comparative des coûts logiciels libres versus logiciels propriétaires reste - dans bien des cas - en faveur du libre. Le comparatif mérite d'être effectué en toute objectivité et la balance penche en faveur de la solution libre (voir le paragraphe « Aspects coûts »).

**N°2 - « Les solutions open source ne sont pas capacitaires et non adaptées aux applications critiques exigeant de la haute disponibilité ».**

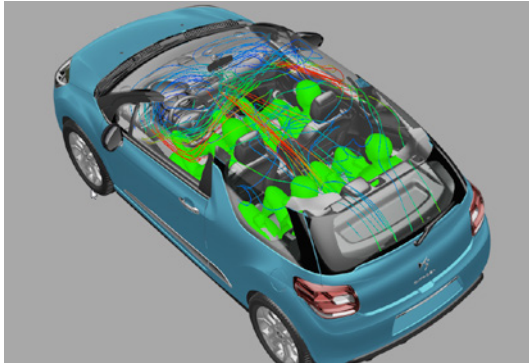
En réalité, des solutions open source proposent des architectures techniques qui répondent aux questions du capacitaire et de la haute disponibilité, tout comme pour les logiciels propriétaires (voir le paragraphe « Aspects techniques »).

A titre d'exemple, en 2013, les applications de calcul scientifique chez PSA Peugeot Citroën tournaient sur

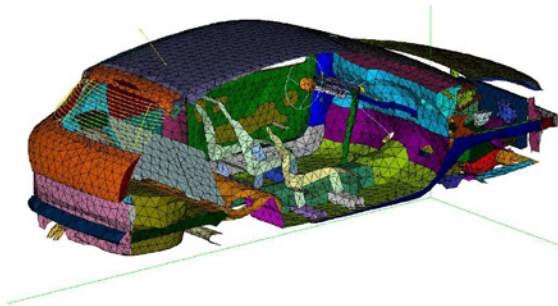
une infrastructure open source de 800 serveurs Linux avec 9.000 cœurs en production et une capacité d'environ 100 Teraflops. Ces applications se situent au cœur du métier de la conception des véhicules chez PSA (CFAO, simulation de crash, etc...). A souligner également que ces infrastructures de calcul basées sur des serveurs standard x86 ont remplacé les supercalculateurs de la génération précédente, entraînant une baisse de coûts très significative alors même que les applications requièrent des besoins en puissance de calcul exponentiels, comme en témoigne le schéma ci-dessous.



Evolution des besoins de la puissance de calcul par type d'utilisation - Copyright PSA Peugeot Citroën



Exemple d'utilisation de l'infra open source dans la phase de conception : schéma de câblage du véhicule  
Copyright PSA Peugeot Citroën



Exemple d'utilisation de l'infra open source dans la phase de conception : calcul de structure d'un véhicule  
Copyright PSA Peugeot Citroën

**N°3 - Variante de l'idée précédente, « on réserve l'open source pour le développement et non pour la production ».** Le paragraphe sur les « périmètres » montrera que les domaines d'utilisation de l'open source vont bien au-delà.

**N°4 - « Le risque est trop élevé et je n'ai pas de support en cas d'incident ».**

La crainte est, qu'en cas de dommage, aucun cadre contractuel ne permette d'identifier les responsabilités voire de fixer des pénalités. Or, depuis longtemps déjà, des sociétés fournissent des services d'implémentation et une prestation de support dans un cadre contractuel (voir le paragraphe « Risque et responsabilité »).

**N°5 - On pense parfois que « comme il n'y a pas de redevance, il n'y a pas de licence ».** Faux, les logiciels libres sont soumis à des licences libres qui ont comme but - notamment - de préserver le caractère libre du logiciel. De ce fait, elles déterminent les droits et devoirs des utilisateurs du libre.

## Une adoption différenciée selon les périmètres concernés

Même si dans une stratégie open source, on ne peut exclure un domaine en particulier, les retours d'expérience montrent que certains périmètres rendent plus facile l'adoption de l'open source. Pour quelles raisons et quels sont ces domaines ?

### L'infrastructure, domaine de prédilection des logiciels open source

Dans le domaine de l'infrastructure, l'offre est importante, les solutions répandues, à la fois des solutions en production et ayant une large diffusion. Il s'agit de solutions éprouvées. En outre, la mise en œuvre de ces solutions n'entraîne pas de difficultés d'accompagnement au changement. On introduit un produit dans un paysage d'infrastructure existant, en vérifiant sa bonne intégration dans l'environnement de l'entreprise comme avec n'importe quel autre produit.

Parmi ces logiciels d'infrastructure les plus emblématiques, citons :

- l'OS Linux
- Nagios pour la supervision
- PostgreSQL et MySQL pour les bases de données

- Tomcat, JBoss et JOnAS pour les serveurs d'application
- Les messageries asynchrones ou inter-applicatives (MOM) comme RabbitMQ
- OpenStack comme suite de mise en œuvre de gestion de cloud
- Etc...

Une **grande compagnie aérienne française** cite, comme raisons du choix de **Nagios** :

- La réduction des coûts,
- L'indépendance vis-à-vis des solutions propriétaires,
- L'agilité.





Un grand **opérateur télécom français** déclare avoir choisi de construire son infrastructure cloud avec **OpenStack** pour les raisons suivantes :

- « La solution de cloud Open Stack a vocation à devenir le Linux du cloud »
- « Elle est tirée par les grands acteurs du marché (Rackspace, HP, Red Hat, IBM, ...) »
- Elle a une communauté de développeurs significative (exemple : environ 1.500 développeurs actifs pour la dernière version)
- Elle offre un socle d'infrastructure proposé maintenant par la plupart des fournisseurs d'équipements réseaux virtualisés

Cet opérateur télécom est d'ailleurs devenu Corporate sponsor de la communauté OpenStack, à laquelle il contribue principalement sur les modules de gestion du réseau.



**Conclusion :** l'infrastructure est un domaine favorable au déploiement open source.

### Le poste de travail spécialisé

C'est un poste léger qui permet exclusivement d'accéder à une (ou plusieurs) applications métier.

Par exemple, **le terminal industriel PSA Peugeot Citroën** : plus de 4.000 postes sous Linux Firefox, des terminaux fixes installés en usine ou embarqués sur des chariots et qui sont utilisés pour la manutention de pièces sur les lignes de montage.

On est sur un périmètre métier bien maîtrisé et la présence de logiciels open source est transparente pour l'utilisateur.

Autre facteur facilitant l'adoption, le fait qu'en cas de dysfonctionnement, le poste léger soit remplacé en quelques secondes par un autre poste en état de marche.



**Conclusion :** le poste de travail spécialisé est un domaine favorable au déploiement open source.

### Les logiciels métier

L'offre open source est plus réduite et la plupart du temps moins riche fonctionnellement (exemple : un outil de gestion commerciale, CRM, ERP, etc...).

Les utilisateurs vont être plus difficiles à convaincre pour franchir le pas open source.

Dans l'évaluation des plus et des moins, il faut mettre en balance la complexité des fonctions métier ciblées, le poids des contraintes budgétaires, l'enrichissement fonctionnel éventuellement rendu possible soit par la société elle-même ou bien par d'autres membres de la communauté.



**Conclusion :** A évaluer en fonction des contraintes fonctionnelles et budgétaires de la société.

### Le poste de travail bureautique

Sur ce domaine, l'adoption de l'open source est pénalisée de par la prééminence de l'environnement Microsoft qui entraîne 2 phénomènes :

- La nécessité d'un vrai effort de conduite du changement auprès des utilisateurs, qui, à titre privé, maîtrisent déjà l'environnement Microsoft, dans la plupart des cas
- Les problèmes de compatibilité de format entre les suites Microsoft Office et les formats Open Office, Libre Office, etc...

L'expérience montre que, même si théoriquement la conduite du changement reste possible, en pratique ce choix ne fonctionne que sous forte contrainte budgétaire et managériale.

Plus le poste est générique, plus l'adoption open source est difficile pour des raisons de compatibilité de document (entre Open Office et Windows), pour des raisons de confort et d'ergonomie (l'open source étant souvent plus « rustique » que les solutions propriétaires).



**Conclusion :** l'adoption par les utilisateurs est souvent problématique.

## Aspects techniques

### Capacité, haute disponibilité, fiabilité...

Les solutions d'infrastructure open source (OS et middleware) disposent de **caractéristiques techniques qui permettent d'adresser les besoins capacitaires, de haute disponibilité pour des applications qui requièrent un excellent niveau de fiabilité.**

Ceci est permis par :

- La capacité à **construire des architectures scalables de type « fermes de serveurs »**, physiques ou virtuels, lesquelles garantissent la continuité de fonctionnement, la haute disponibilité et le traitement de gros volumes.

On pense notamment aux fermes de serveurs Tomcat sous Linux ou à l'exemple de la ferme de calcul scientifique chez PSA vue précédemment.

- Des solutions de **bases de données distribuées** (cluster) qui assurent le traitement de gros volumes sans interruption de service (par exemple : MySQL cluster, PostgreSQL, etc...).



- Enfin, l'open source –avec OpenStack- permet aujourd'hui de **construire des environnements cloud complets** (fonctions de provisioning et d'administration de cloud incluses) qui vont supporter le S.I. de l'entreprise.

Comme l'indique un membre du Groupe de Travail, « **on peut choisir de travailler sur un périmètre évolutif afin de crédibiliser le choix du produit** ».

C'est ainsi que cette société a implémenté Nagios :

- Dans une première étape, sur un périmètre réduit mais complexe avec 20 serveurs
- Puis sur un grand périmètre non critique (2.600 serveurs Windows)
- Enfin sur un périmètre large et très critique : la supervision de 5.000 équipements réseau.

Autre **exemple illustrant le traitement de gros volumes et l'aspect fiabilité** :

**l'application Chorus** qui traite la comptabilité de l'Etat français, avec un système central propriétaire

et de nombreuses applications frontales développées en open source.

Pour donner un ordre de grandeur, cette application est accédée par 55.000 utilisateurs environ.

### Pérennité de la solution

Comme pour les logiciels commerciaux, c'est la **taille de la base installée** qui est le meilleur garant de la pérennité d'une solution.

En outre, la taille et le **dynamisme de la communauté** est un point à mesurer lors du choix d'une solution open source.

A la différence d'une solution propriétaire, la pérennité de la solution open source n'est **pas menacée par la logique commerciale d'éditeurs** qui, soit rachètent des solutions pour mieux les contrôler voire les supprimer, soit -après fusion de sociétés- choisissent d'abandonner une solution pour raison de rentabilité.

Et dans le cas où une solution open source est rachetée par un éditeur (cas de GlassFish suite au rachat de Sun par Oracle), la communauté développe généralement **un fork**, version alternative du produit, qui, lui, reste totalement open source.

### Rythme de changement de version

L'utilisateur est moins soumis au rythme de changement des versions imposé par les éditeurs, à travers les dates de fin de support annoncées par les éditeurs. Le nombre de versions actives supportées par la communauté est en général supérieur à ce que l'on constate dans le domaine des logiciels commerciaux. Néanmoins, pour bénéficier d'un support optimum, délivré par la communauté, il ne faut quand même pas rester trop en retard quant aux évolutions de produits.

### Roadmap produit

Autre facteur spécifique de l'open source : **la possibilité pour une société** de répondre à un besoin important pour elle, **en développant une fonction manquante** qui va intégrer le produit et être prise en charge - en termes de support et d'évolution - par la communauté. Ceci, au lieu d'attendre une éventuelle prise en compte par l'éditeur dans sa feuille de route « roadmap » de développement.

### Accélérateur de développement

L'open source est un **facteur important de réutilisation et d'accélérateur du développement interne** de par le nombre de composants, bibliothèques, etc... disponibles dans la communauté et réutilisables. C'est une pratique largement répandue chez les développeurs d'aller sur internet pour trouver des « bouts de code » et de les intégrer en tout ou partie, d'où un bon gain de productivité et de qualité.



Toutefois, cette pratique présente un **risque juridique** : la violation de licence GPL causée par la présence de code open source dans une application destinée à être diffusée en externe. Un conseil : mettre en place des moyens de contrôle et d'administration de ces composants, dans le cadre d'une gouvernance IT.

## Risque et responsabilité

### Support et risque opérationnel

La communauté assure le support des produits open source. Il est généralement suffisant pour les produits simples et pour des usages non critiques, d'autant plus que la taille de la communauté est grande.

Si les sociétés ne jugent pas cette formule suffisante pour faire face aux risques opérationnels, elles peuvent acheter une prestation de support :

- Soit à travers **l'acquisition d'une distribution du logiciel libre** par des sociétés spécialisées telles que SUSE pour Linux et OpenStack ou Red Hat pour OpenStack, etc...
- Soit en faisant **appel à des sociétés de services**. Celles-ci fournissent du support pour un portefeuille négocié de produits, dans une approche flexible de facturation à l'usage réel.

Ces sociétés fournissent des correctifs à court terme, font l'interface avec la Communauté pour permettre de développer la correction, tout cela dans un cadre contractuel et de manière similaire à un éditeur.

### Qui est responsable ?

Sur le **plan légal**, l'absence de cadre contractuel avec des responsabilités identifiées et une assurance en cas de dommages causés par le logiciel peut constituer un frein à l'adoption. **L'acquisition auprès de distributeurs** adresse ce point, au moins en partie.

### Consulter un juriste

Dans le cas où une application est développée en utilisant du code open source publié sous « licence contaminante » (licence GNU GPL), cette application est soumise aux mêmes obligations que le code open source utilisé. Ceci peut s'avérer problématique dans le cas où ladite application est destinée à être diffusée. Mieux vaut, alors, prendre un avis juridique.

### Sécurité

Dans le domaine de la sécurité, les risques en terme de failles de sécurité sont a priori moindres.

Le code est ouvert, lisible et contrôlable par toute la communauté, donc la probabilité pour qu'il contienne du code malveillant (malware), du code « espion » étatique (sous couvert de Patriot Act ou autre raison d'Etat) ou du code espion à visée industrielle est plus faible.

Plus difficile, par exemple, d'insérer un bout de code espion dans du Linux.

A contrario, pour un hacker, plus le logiciel open source va être répandu, plus il devient attractif donc ciblé.



## Aspects coûts

### Côté gain

Le gain incontestable, par rapport aux logiciels propriétaires, porte sur **l'absence de redevance logicielle et de coût de maintenance**.

Ceci est d'autant plus vrai que la volumétrie concernée est élevée, même si, dans certains cas, les éditeurs concèdent des droits d'usage illimités.

L'open source peut être utilisé également pour limiter la croissance du nombre de licences installées. L'évolution des besoins business peut entraîner une utilisation croissante de licences propriétaires. Afin de rester dans les limites du nombre de licences prévu au contrat, on va orienter une partie des besoins nouveaux vers des produits open source.

**La plupart des produits open source simples se satisfont du support communautaire, donc à coût zéro.**

Si le produit est complexe ou utilisé sur une application critique, la société peut choisir d'acheter cette prestation à une société externe. Mais le modèle de facturation à l'usage réel s'avère malgré tout largement moins onéreux qu'une prestation de maintenance chez l'éditeur.

### Côté coût

En revanche, **les coûts d'intégration de produits open source ne doivent pas être oubliés.**

En effet, les suites propriétaires sont généralement mieux intégrées que les suites communautaires qui nécessitent un travail d'intégration d'autant plus important que la solution est riche et complexe.

Si la solution considérée est simple, le travail d'intégration est faible et pris en charge par la société utilisatrice elle-même, de même que le support.

Pour les solutions open source communautaires complexes, par exemple la gestion du cloud par OpenStack, deux cas de figure sont possibles :

- Les utilisateurs assurent directement ce travail d'intégration.
- D'autres sociétés n'ont ni la compétence ni les effectifs pour l'assumer. Elles font appel à des distributeurs qui fournissent la solution intégrée et vendent le service et le support afférents.

En termes de coûts, les aspects formation et conduite du changement de produits open source versus propriétaires sont équilibrés.



## Quelle gouvernance mettre en place ?

Pour piloter et suivre le déploiement des solutions Open Source dans l'entreprise, il est recommandé de mettre en place une gouvernance dont la mission va s'organiser autour de six objectifs principaux :

### **1/ Constituer et mettre à disposition « sur étagère » une offre de produits disponibles pour les équipes IT.**

Pour chaque fonction concernée, l'entité de gouvernance identifie les produits candidats, les évalue, les certifie et les met à disposition. Afin de limiter la diversité, qui est coûteuse, et de bénéficier des effets de volumes, le nombre de logiciels pour une fonction donnée sera limité. L'entité de gouvernance gère les entrées mais également les sorties de logiciels dans le « catalogue » des produits disponibles.

### **2/ Piloter la stratégie Open Source de l'entreprise, faire la promotion des solutions Open Source et s'assurer de leur mise en œuvre.**

L'entité de gouvernance est responsable de faire connaître aux équipes IT la stratégie Open Source. De même, elle communique sur l'offre interne de logiciels Open Source pour chaque fonction IT. Elle s'assure de la mise en œuvre de la stratégie Open Source à l'occasion des comités techniques des projets : cela consiste, par exemple, à vérifier que la solution Open Source est bien retenue, quand on a décidé qu'elle est la solution par défaut pour une fonction donnée.

### **3/ Piloter et encadrer le développement pour sensibiliser les équipes sur les bonnes pratiques d'utilisation des morceaux de code Open Source dans les développements internes.**

Facteur de productivité grâce à la réutilisation, cette pratique présente néanmoins un risque juridique de violation de licence GPL. L'entité de gouvernance devra mettre en place des moyens permettant, au minimum, de savoir quels sont les composants utilisés et où ils sont utilisés.

### **4/ Identifier et mettre à disposition les moyens de support des logiciels Open Source utilisés quand cela est nécessaire.**

L'entité de gouvernance identifie la liste des logiciels devant faire l'objet d'un support par une société prestataire. Elle proposera la ou les sociétés qui devront fournir le support. Elle établira, en support de la Direction des Achats, les contrats de prestation et en suivra l'utilisation (consommation des tickets par exemple).

## 5/ Piloter la contribution de l'entreprise à la communauté.

Dans le cas où l'entreprise souhaite contribuer en termes de don de code à la communauté, l'entité de gouvernance participera au choix de la communauté à laquelle on décide de contribuer. Elle pilotera la contribution à la communauté dont elle sera l'interface pour l'entreprise.

## 6/ Faire appel aux juristes.

L'entité de gouvernance sollicite les juristes internes, et éventuellement externes, dans les cas où la question juridique se pose : application basée sur des logiciels Open Source et destinée à être commercialisée, mise dans le domaine communautaire de code de l'entreprise, etc...

## Conclusion

S'il est évident que le choix des solutions Open Source par la plupart des entreprises est motivé en premier lieu par des raisons économiques, il est tout aussi clair que ce n'est plus aujourd'hui le seul critère, même s'il restera vraisemblablement le premier.

Devant une offre Open Source riche, mature et éprouvée - tout particulièrement dans le domaine des produits d'infrastructure - les entreprises peuvent envisager de bâtir une stratégie technique qui repose sur des briques Open Source. Le développement des solutions de Cloud privé ou public sur une base Open Source (OpenStack, Docker, CloudFoundry) en est une illustration majeure.

La question n'est plus de savoir si on peut utiliser ou pas des solutions Open Source mais sur quels périmètres, avec quels avantages et avec quel dispositif d'accompagnement. La présente fiche fournit quelques éléments issus de l'expérience des entreprises du CRiP pour aider cette réflexion et favoriser une mise en œuvre réussie des solutions Open Source.



**Club des Responsables d'Infrastructures et de Production**

24 rue Erlanger 75016 Paris - [contact@crip-asso.fr](mailto:contact@crip-asso.fr)

[www.crip-asso.fr](http://www.crip-asso.fr)