

**Concurrence et complémentarité des logiciels libres et propriétaires**

**SALON I-EXPO : ATELIER DU 28 MAI 2008 (9h30-12h30)**

*Implémentation de solutions libres, quel retour sur investissement ?*

Laurent Soual, consultant associé, doXulting

# Sommaire

## *Implémentation de solutions libres, quel retour sur investissement ?*

- *essai de typologie des projets open source*
- *relativité du critère budgétaire*
- *comparaison n'est pas raison ?*
- *où se cachent les coûts ?*
- *postes d'économie et de dépenses*
- *étude de cas*

# 1 – typologie des projets open source

## ***Développement collaboratif***

On développe les fonctions manquantes que l'on reverse à la communauté

**Type de produit** : applications métier immatures ou besoins spécifiques

**Ressources** : Architecture technique (matériel, réseau...), développeur interne (missionné officiellement, compétent et disponible) ou prestataire de service, communauté « intégrante » et dynamique.

**Coûts** : Forfait de développement. Economie sur le coût de licence

**Précautions** : en phase « amont » : étude très sérieuse des fonctionnalités, de la vitalité de la communauté et de ses règles de contribution. Savoir estimer le temps de développement et contrôler le respect de cette estimation

**Risque** de dérive coûts/délai si le volume de développement est sous estimé. Risque d'isolement si la communauté refuse d'intégrer les modifications (fork).

# 1 – typologie des projets open source

## « Sur étagère »

le logiciel est accepté tel que sans modifications.

**Type de produit** : Utilitaires, applications matures ou CMS ( si les besoins sont modérés et très classiques

**Ressources** : Architecture technique (matériel, réseau...), ressources internes + aide de la communauté pour l'installation.

**Coût** : très modéré, voire nul, sauf si l'on fait appel à une assistance externe pour la mise en oeuvre (paramétrages, formations...)

**Précaution** : en phase « amont » étude très sérieuse des fonctionnalités

**Risque** : si un manque fonctionnel apparaît en cours ou à l'issue du projet : dérive budgétaire ou retour arrière (temps perdu)

# 1 - typologie des projets open source

## Intégration

Plusieurs logiciels Open Source sont associés sous une même interface graphique/ergonomique/fonctionnelle

**Type projet** : Infrastructure, applications ou CMS avec besoin spécifique à couvrir

**Ressources** : développeur interne (missionné officiellement, compétent et disponible) ou prestataire de service très compétent en open source.

**Coût** : forfait de développement et d'intégration. économie du coût de licence

**Précaution** : Très bonne connaissance des logiciels de base, de leur interopérabilité, de la compatibilité de leurs licences. Savoir estimer le temps de développement et contrôler le respect de cette estimation.

**Risque** de dérive coût/délai si le volume de développement est mal estimé.

# 1 - typologie des projets open source

## **Un choix fondamental : s'éloigner ou non du standard**

Cas de développements additionnels reversés (et intégrés par la communauté) :

*Fonctionnement classique d'une communauté, les modifications sont intégrées au logiciel et seront disponibles dans les nouvelles versions.*

Cas de développements additionnels non reversés ou non intégrés par la communauté :

*Renoncement aux nouvelles versions. éventuellement création d'un fork.*

*Ou*

*Documentation rigoureuse des additifs et réintégration des ajouts à chaque chargement d'une nouvelle version. Coût récurrent*

# 1 - typologie des projets open source

## Spécificités d'un projet open source

- identification et évaluation des logiciels
- évaluation de la communauté
- choix du reversement ou non des modifications
- absence de relation client/fournisseur dans le cas d'un travail direct avec la communauté
- si prestataire (SSLL) il ne s'engage que sur ses prestations, et non sur l'évolution du logiciel

## Similitudes avec un projet "propriétaire"

- analyse des besoins
- gestion projet des développements
- relation client/fournisseur dans le cas d'un recours à une SSLL

## 2 – relativité du critère budgétaire

**Dans un projet open source les critères budgétaires ne sont pas toujours déterminants**

- coûts d'investissements moindres
  - mais dichotomie charges internes / charges externes
- coûts de licences nuls ou presque (cas des open source à distribution payante)
  - mais des droits attachés aux licences qui peuvent être complexes
- il faut se méfier des coûts cachés
  - investissement temps humain
  - coût interne ou externalisation (études et développements)
  - des mises en œuvre qui s'allongent dans le temps

### 3 – comparaison n'est pas raison ?

Type de coûts	Solutions propriétaires : en % du total des coûts	Solutions libres : en % du total des coûts	Comparaison des coûts bruts
Coûts d'investissement			
Matériels et système d'exploitation	10 à 20 %	20 à 30 %	Coûts équivalents
Hébergement externalisé de l'application	0 à 2 % par an	0 à 2 % par an	Coûts équivalents, peuvent être légèrement supérieurs pour les solutions propriétaires reposant sur des technologies ou plateformes non standard
Logiciels substrats (SGBDR, Middleware...)	0 à 15 %	0 %	Supérieurs pour les solutions propriétaires
Logiciel métier (SIGB, logiciel documentaire, logiciel d'archives, portail, outils multimédia...)	20 à 30 %	0 %	
Prestations externalisées (reprises, paramétrages, formations...)	50 à 70 %	10 à 80 %	Equivalents, peuvent être supérieurs pour les logiciels libres
Prestations assurées en interne et autres coûts cachés	10 à 50 %	20 à 100 %	Supérieurs pour les logiciels libres
Coûts récurrents			
Maintenance logicielle	2 à 5 % par an	0 à 30 % par an	Peuvent être supérieurs pour les logiciels libres
Veille sur le produit et autres coûts cachés	1 à 5 %	10 à 20 %	

## 4 – où se cachent les coûts ?

### **Dans un projet open source le critère organisationnel est crucial**

- un projet open source ne peut être mené sans équipe
- il n'est plus possible de dériver la responsabilité sur un éditeur (externalisation de la responsabilité)
- même si le retour en arrière est toujours possible : une maquette ne devient pas un prototype si ce dernier n'est pas porté par une équipe convaincue et opérationnelle
- **Conclusion** : le temps passé par les équipes est plus important. On y gagne en adhésion et motivation, on y perd en ressources mobilisées

## 4 – où se cachent les coûts ?

### **Dans un projet open source on ne peut pas faire l'économie d'une analyse préalable**

- › même si un projet open source n'entre pas dans les "canons projets" (budget prévisionnel, cahier des charges, mise en concurrence, validation du choix par une commission d'AO)

Cette analyse préalable doit répondre aux questions suivantes :

- l'outil pressenti répond-il aux besoins minimaux ?
- les contraintes ont-elles été analysées ?
- le contexte organisationnel a-t-il été pris en compte ?
- › si cette étude est externalisée, elle constitue un coût à prendre en compte

# 5 – postes d'économie et de dépense : maturité du projet et retour sur investissement.

## Tout dépend du degré de maturité du projet

- les projets pionniers ne sont pas forcément plus coûteux en investissement, l'objectif est le retour sur investissement **à long terme, à condition**
  - de drainer de nouveaux entrants sur la solution (pas de fork)
  - de partager les développements (et leurs coûts)
- l'investissement humain est toujours très important, quel que soit le projet et il est rarement chiffré
  - vacataires
  - gens du métier versés dans l'informatique (bibliothécaires, documentalistes)
  - informaticiens de métier
- à long terme, l'absence de contrat de maintenance, qui est le propre des logiciels propriétaires, permet de consacrer, le cas échéant, des moyens à une véritable maintenance évolutive
- la présence d'une communauté active permet de disposer de modules qui seraient chiffrés auprès d'éditeurs propriétaires
- en cas de réinformatisation après un épisode libre, pas de « chantage » de l'éditeur sur la reprise de l'existant

## 6 – 1ère étude de cas : l'École des Mines de Paris.

### - Contexte

- Environnement technique hétérogène
- Logiciel non maintenu
- Enveloppe budgétaire limitée

### - Décisions affirmées

- Analyse préalable (l'essentiel, le superflu)
- L'harmonisation des données
- L'observation de Koha

### - La démarche

- Installation du logiciel
- Maquettage itératif, tests, pédagogie
- Développements, réflexion sur le support
- Mise en exploitation définitive au bout de 15 mois

### - Analyse et bilan

- Une équipe impliquée
- Un projet assez coûteux en assistance et en développement mais plus économique que s'il y avait eu appel d'offres
- Un effet d'entraînement notable
- Un retour sur un investissement qui se profile

## 6 – 2e étude de cas : Conseil Général du Jura

### - Contexte

- Dès le départ le périmètre fonctionnel est cadré : permettre aux jurassiens de localiser un ouvrage dans le réseau de lecture publique et d'effectuer des demandes de prêts entre bibliothèques en ligne
- Après un tour d'horizon, le choix se porte sur un produit open source (MoCCAM)
- mais nécessité de développer les fonctions manquantes (PEB)

### - La démarche

- Appel d'offre pour le choix d'un prestataire pour la mise en œuvre (installation, paramétrage + développement de fonctions manquantes ),
- Projet long (difficultés sur l'architecture technique notamment) : 2 ans
- Fort investissement de l'équipe de la BDP (Conseil Général)
- Participation active à la communauté : les développements payés par le Jura sont ou seront reversés à la communauté

### - Analyse et bilan financier

- Coût global : 100 000 € (non compris charges internes)
- Coûts récurrents : 1 800 € HT / an
- Principal problème : le logiciel choisi n'était pas encore vraiment utilisé et la communauté était très réduite et peu active en échanges ; tâtonnements et pertes de temps
- Au final, un coût équivalent à celui d'une solution propriétaire

## 6 – 3e étude de cas : SAN Ouest Provence

### - Contexte

- Réinformatisation d'un réseau inter-communal : 7 sites, 250 000 doc, 600 000 prêts annuels, 160 professionnels

### - La démarche

- Étude préalable : choix entre le connu (solution propriétaire estimée à 250 000 HT, 15 % de maintenance annuelle) et le moins connu (Koha, solution open source, avec des fonctions manquantes à développer, estimée à 200 000 € en comptant le salaire des informaticiens internes chargés du développement)
- Lancement du projet en juin 2005 ; mise en place d'une structure de projet importante (commissions techniques)
- mutualisation avec la communauté Koha : choix d'une solution mixte
- Répétition générale : décembre 2006 ; mise en service en janvier 2007
- Mise en place d'une maintenance en interne

### - Analyse et bilan

- Globalement positif ; appropriation par les professionnels ; dynamique collaborative
- Une complexité à gérer : rester en phase avec les évolutions du logiciel

## 7 – conclusion

- dans une période pionnière, le coût d'un projet open source n'est pas nécessairement déterminant par rapport à une solution « propriétaire »
- le retour sur investissement n'est pas immédiat
- solution non viable si non portée par une équipe
- l'avenir du projet open source (retour sur investissement rapide) réside donc dans la mutualisation (consortium, groupements d'achat..)
- ...ce qui est dans la logique communautaire de l'open source

**Merci de votre attention**

**Avez-vous des questions?**

**[Isoual@doxulting.fr](mailto:Isoual@doxulting.fr)**