

Le Portefeuille d'Etudes Yphise

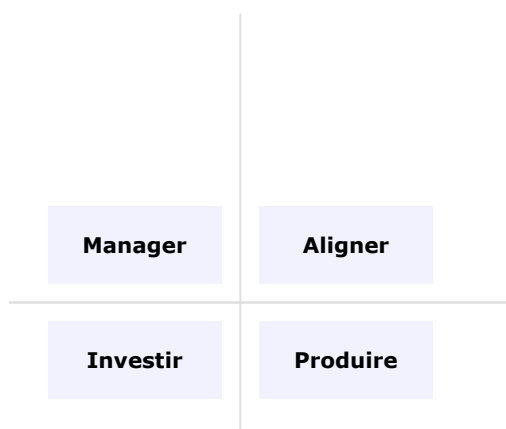
à l'attention des décideurs, managers et cadres pour réfléchir, agir et se former

Manager - Investir - Aligner - Produire

Recherche indépendante en informatique d'entreprise MOe et MOa depuis 1985

TIRER PROFIT DU LOGICIEL LIBRE OU OPEN SOURCE

L'industrie du logiciel à la recherche de modèles économiques



Nous avons le plaisir de mettre cette étude à votre disposition afin que vous ayez l'opportunité d'apprécier notre travail pour vous aider à réussir votre métier de cadre ou manager en informatique, MOa ou MOe.

Elle comporte en amont une brochure de présentation du **Portefeuille d'Etudes Yphise**.

Cette brochure inclut un **bon de commande**.

Table des matières	3
Résumé décideurs	5
Enjeu et plan	7
1. Les notions essentielles	8
2. Le besoin des éditeurs	14
3. La prise en compte par les entreprises	24

"On devrait tout rendre aussi simple que possible, mais pas plus" Albert Einstein

Votre réussite

Yphise apporte aux **cadres, managers et décideurs**, côté MOe ou MOa, le recul indépendant pour dynamiser la réflexion, sécuriser la décision et réussir l'action afin d'accroître l'efficacité de chaque activité informatique en entreprise.

Nous faisons ce métier avec passion depuis 1985 ; nous sommes fiers d'avoir contribué à la réussite d'équipes MOe ou MOa d'entreprises de tout secteur et de toute taille. C'est avec plaisir que nous vous aidons dans vos projets et vos défis d'aujourd'hui.

Utiliser l'expertise Yphise est simple

1. le Portefeuille d'Etudes Yphise

Notre originalité est de mettre notre expertise à votre disposition sous la forme d'un portefeuille d'études que vous pouvez utiliser librement pour réfléchir, agir et vous former. C'est simple, efficace et particulièrement bon marché.

Le Portefeuille d'Etudes Yphise

à l'attention des décideurs, managers et cadres pour réfléchir, agir et se former
Manager - Investir - Aligner - Produire

2. le Collège des cadres

Nous vous aidons à réussir la dynamique d'équipe, la conduite du changement et la montée en compétences de l'encadrement au moyen de séminaires de réflexion intra, à la carte ou prêts à l'emploi.

Le Collège des cadres Yphise

Séminaires intra à la carte de réflexion des cadres, MOe ou MOa
Dynamique d'équipe - Réussite du changement - Compétences d'encadrement

3. la Formation continue

Nous venons vous former, vous et vos équipes, aux compétences professionnelles en informatique d'entreprise. Sept filières : Maître d'ouvrage (MOa, aMOa) ; Manager informatique ; Qualité - Processus - Méthodes ; PMO - Responsable de SI ; Chef de projet MOe ; Architecte SI ; Service management.

4. les Missions d'accompagnement

Vous pouvez nous confier une mission d'accompagnement, côté MOe ou MOa, pour piloter, contrôler ou améliorer.

Présentation détaillée de chaque prestation : www.yphise.fr

Le Portefeuille d'Études Yphise

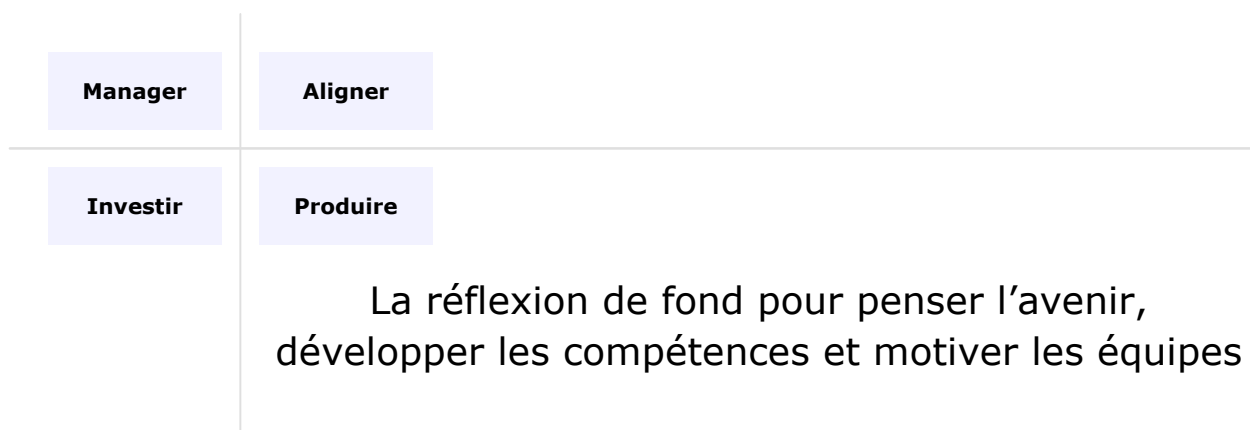
à l'attention des décideurs, managers et cadres pour réfléchir, agir et se former

Manager - Investir - Aligner - Produire

Recherche indépendante en informatique d'entreprise MOe et MOa depuis 1985

LE PORTEFEUILLE D'ÉTUDES YPHISE :

Présentation - Mode d'emploi - Bon de commande



Bon de commande	2
1. Simple, efficace et bon marché	3
1.1 La réflexion de fond dont vous avez besoin sans avoir le temps de la mener	
1.2 Le concret de chaque étude	
1.3 Votre réussite au prix de quelques heures d'un ingénieur.	
2. Mode d'emploi	5
2.1 Une veille d'ensemble en quelques minutes	
2.2 Agir au moment du besoin	
2.3 L'autoformation de chaque cadre, MOe ou MOa, pour une réussite collective	
3. Liste des études du Portefeuille	7
3.1 Manager	
3.2 Aligner	
3.3 Investir	
3.4 Produire	

Licence d'accès au Portefeuille sur www.yphise.fr

Bon de commande

Yphise - 72 rue Damrémont - 75018 PARIS - T 01 42 78 20 59 - yphise@yphise.com

Société	Nom
Tel	Fonction
Réf commande	Ident TVA ¹
Adresse	

(1) N° d'identification TVA de votre entreprise nécessaire pour toute commande (y compris France).

Paiement par chèque, virement bancaire ou carte bancaire sur www.yphise.fr.

Virement Code banque 30002, Code guichet 00404, Compte 0000007560Q, Clé RIB 51
Domiciliation LCL PARIS MONTORGUEIL

Prix

<input type="checkbox"/> Oui, je prends la licence d'accès au Portefeuille d'Etudes Yphise Toutes les études disponibles + les 6 à paraître sur les 12 mois à venir, à chaque collaborateur de mon choix dans mon entreprise 1200 Eur HT soit 1453.20 Eur TTC
<input type="checkbox"/> Oui, j'ai droit à la réduction DSI ou prestataire < 30 collaborateurs : -30% 840 Eur HT soit 1004.64 Eur TTC
<input type="checkbox"/> Oui, j'ai droit à la réduction consultant indépendant ou auto-entrepreneur : -60% 480 Eur HT soit 574.08 Eur TTC

Liste des destinataires

Nom	1 Email

(1) L'email constitue l'ID sur www.yphise.fr auquel nous attachons les droits.

Pas de souci si vous ne savez pas à ce jour qu'elle sera cette liste. Celle-ci est modifiable sans frais et à tout moment pendant les 12 mois de validité de votre licence. Il suffit de nous envoyer un e-mail.

Date - Signature

Le Portefeuille d'Etudes Yphise

1. Simple, efficace et bon marché

1.1 La réflexion de fond dont vous avez besoin sans avoir le temps de la mener

Il y a le temps de l'action et celui de la réflexion.

Le problème est que l'action a tendance à tout remplir ; il est pourtant vital de préserver la place de la réflexion pour **penser l'avenir, développer les compétences et motiver les équipes**.

Le Portefeuille apporte à chaque cadre, manager ou décideur, côté MOe ou MOa, la **réflexion de fond** indispensable à sa réussite ; réflexion qu'il est si difficile de mener face aux pressions de l'opérationnel.

Réussir nécessite de trouver les moyens du « travail bien fait », de l'efficacité économique et de la motivation. Ce n'est jamais simple en raison des contraintes de budget, délai ou événements subis.

Les équipes qui réussissent sont plus que jamais celles qui agissent avec méthode, selon des savoir-faire qui vont à l'essentiel et sont au plus simple. L'enjeu est d'autant plus difficile en informatique qu'il faut en permanence réussir des collaborations entre expertises variées.

Le Portefeuille est **une base d'expertise constituée d'études**. Chacune traite en 30 pages un enjeu soigneusement sélectionné pour son importance. Elles sont classées en 4 domaines de compétences qui articulent les activités à réussir en informatique :

- **Manager**. Comment relever les défis de l'encadrement et de la gouvernance.
- **Aligner**. Comment faire pour que les systèmes d'information répondent bien au besoin de l'entreprise.
- **Investir**. Comment réussir l'investissement, c'est-à-dire les portefeuilles de projets et chaque projet.
- **Produire**. Comment réussir la production du service.

Le Portefeuille est ainsi l'outil de référence de chaque cadre, MOe ou MOa, pour **sécuriser l'opérationnel** : penser l'avenir, développer les compétences et motiver.

1.2 Le concret de chaque étude

Ce que nous essayons de faire à chaque étude tient en quatre points.

1. **Apporter la réflexion de fond** dont vous avez besoin, sans avoir le temps de la mener.
2. **Toujours viser des messages et solutions les plus simples**, sans être plus simples que la complexité du réel.
3. **Toujours se situer au niveau de l'action opérationnelle en entreprise**, sans débat théorique ou conceptuel inutile.

Le Portefeuille d'Etudes Yphise

4. **Favoriser une dynamique de compréhension et de collaboration** entre MOe et MOa, et entre les métiers d'une maîtrise d'oeuvre.

Une étude Yphise n'est pas un livre ; chacune est conçue pour permettre à la fois de mener une réflexion de fond et agir en opérationnel.

Une étude Yphise est ainsi un équilibre : du concret sans tomber dans la recette ; le recul sans tomber dans la généralité. Un équilibre est toujours délicat à trouver ; c'est notre savoir-faire particulier, fondé sur +25 années d'expérience, que d'y parvenir.

1.3 Votre réussite au prix de quelques heures d'un ingénieur

Le Portefeuille s'utilise librement via une licence annuelle qui donne accès à tout moment sur **www.yphise.fr** :

- **aux études disponibles et aux 6 nouvelles** publiées pendant les 12 mois à venir (+ des mises à jour),
- **à tous les collaborateurs souhaités** de votre entreprise.

La tarification du Portefeuille est faite pour que **chaque cadre de toute entreprise puisse en bénéficier.**

Le prix de la licence annuelle représente quelques heures d'un ingénieur par an : **son acquisition est ainsi une toute petite décision.**

L'utilisation d'une seule étude du Portefeuille pour en sortir ne serait ce qu'une seule idée, un seul contrôle ou une seule décision au moment du besoin rentabilise la licence annuelle.

En ces temps de stress et de complexité, le Portefeuille est ainsi votre fidèle compagnon pour vous assister. Notre indépendance depuis 1985, grâce à la fidélité des managers qui nous font confiance, est votre meilleure garantie.

Le Portefeuille d'Etudes Yphise

2. Mode d'emploi

Le Portefeuille est conçu pour des décideurs, managers ou cadres qui n'ont **ni le temps ni l'envie de lire** des rapports.

Il s'utilise de 3 manières complémentaires.

2.1 Une veille d'ensemble en quelques minutes

Notre recherche produit 6 nouvelles études par an en sélectionnant avec soin des enjeux dont le potentiel mérite aujourd'hui votre réflexion.

- A chaque parution, vous recevez un e-mail avec un lien pour y accéder.
- Chaque étude comporte un résumé décideur en 1 page.

La surveillance des enjeux sélectionnés par la recherche Yphise complétée de la lecture du résumé décideur constitue une veille efficace à votre intention, sans effort et au moindre coût.

2.2 Agir au moment du besoin

Chaque étude permet d'évaluer sa situation pour décider et agir si cela est utile.

La consultation du Portefeuille vous aide ainsi à diagnostiquer votre situation, avancer la réflexion, préparer la décision, planifier l'action et contrôler **au moment où vous en avez le besoin**.

Pour cela, vous avez un accès permanent sur l'espace client de **www.yphise.fr** à toutes les études du Portefeuille.

- Le Portefeuille vous permet d'agir rapidement sur un besoin, une demande ou une situation urgente en consultant l'étude sur le sujet.
- Chaque étude vous permet de dynamiser et sécuriser la réflexion pour que tous les intervenants se mettent plus facilement et plus rapidement d'accord sur un sujet.
- Chacune constitue une étude préalable prête à l'emploi pour cadrer, piloter et contrôler l'action en mettant en évidence le résultat, comment l'atteindre, les clés de succès et erreurs à éviter.

2.3 L'autoformation de chaque cadre, MOe ou MOa, pour une réussite collective

Le Portefeuille répond au besoin de **formation continue des cadres**, MOa ou MOe, pressés par le temps.

Chaque étude est rédigée pour s'adresser à tout cadre, manager ou décideur, indépendamment de sa spécialité ou de sa responsabilité du moment.

Le Portefeuille d'Etudes Yphise

La réussite professionnelle de chacun et celle collective, entre MOe et MOa et entre les métiers d'une maîtrise d'œuvre, nécessitent en effet une ouverture sur des sujets diversifiés.

Pour cette raison :

- la licence d'accès au Portefeuille est un forfait pour autant de collaborateurs de l'entreprise que souhaité,
- chacun a un accès permanent à toutes ses études avec des boites à cocher pour ne plus afficher celles qu'il a exploitées.

3. Liste des études disponibles

Voici la liste des études disponibles auxquelles la licence annuelle donne immédiatement accès. Elles sont classées sur l'espace client de www.yphise.fr en 4 domaines de compétences.

3.1 Manager

- **Difficultés du management en informatique : comprendre et agir** 11
- **Une stratégie de simplification de l'informatique** 11
- **Stress et démotivation : comprendre et agir** 10
- **Construire l'efficacité collective avec ISO 9001** 10
- **Maitriser l'amélioration continue d'un système de processus** 10
- **Responsabilité et collaboration** 09
- **Processus versus compétence** 09
- **Du système de processus au catalogue de services** 08
- **Utiliser COBIT** 07

3.2 Aligner

- **Gouverner un système d'information** 11
- **Gérer l'alignement d'un système d'information** 10
- **Tirer profit du logiciel libre ou open source** 09
- **Le Contrôle Interne d'un système d'information en production** 09
- **Gérer les risques d'exploitabilité** 07

3.3 Investir

- **Piloter la validation d'un projet applicatif** 11
- **Instruire un investissement informatique** 11
- **Du business case au projet informatique** 10
- **Préparer un projet informatique : réussir EB, CDC, PQP et PDV** 10
- **Aligner l'investissement informatique pour l'entreprise** 10
- **Les 'bons réflexes' pour maitriser la charge métier dans les projets** 10
- **Apprécier les coûts d'un projet en qualification** 09
- **Quels projets lancer ou au contraire supprimer en temps de crise ?** 09
- **Piloter un projet réalisé avec un prestataire** 06

Le Portefeuille d'Etudes Yphise

- **Maitriser le leadership dans les projets** 05
- **Mesurer et manager la productivité des développements** 05

3.4 Produire

- **Réussir le plan de reprise** 11
- **ITIL Lite : comment mettre en œuvre ITIL** 10
- **Le Contrôle Interne d'un système d'information en production** 09
- **Réussir le Service Management avec ISO 20000-1** 07
- **Organiser le SLM pour les applications en production** 06

Le Portefeuille d'Etudes Yphise

- **Maitriser le leadership dans les projets** 05
- **Mesurer et manager la productivité des développements** 05

3.4 Produire

- **Réussir le plan de reprise** 11
- **ITIL Lite : comment mettre en œuvre ITIL** 10
- **Le Contrôle Interne d'un système d'information en production** 09
- **Réussir le Service Management avec ISO 20000-1** 07
- **Organiser le SLM pour les applications en production** 06

Table des matières

Manager	Aligner	www.yphise.fr > Mon Yphise Ce rapport est dans la discipline suivante
Investir	Produire	Aligner les systèmes d'information

Août 2009

Résumé décideurs	5
Enjeu et plan	7
1. Les notions essentielles	
1.1 Le logiciel libre	9
1.2 Les licences	11
2. Le besoin des éditeurs	
2.1 Une industrie du logiciel en difficulté	15
2.2 Dysfonctionnements et espoirs	17
2.3 Les modèles économiques	20
3. La prise en compte par les entreprises	
3.1 Principe de prise en compte	25
3.2 Opportunités de gains financiers	28

Ce service anime la réflexion et l'action de l'équipe de management d'une informatique d'entreprise. Il couvre les enjeux du maître d'ouvrage (MOa) et du maître d'œuvre (MOe).

Chaque étude est conçue pour être accessible à tout cadre, manager et décideur, MOe ou MOa, quel que soit son domaine de responsabilité. Ce service permet une autoformation de chacun par un accès permanent à la totalité du Portefeuille sur **www.yphise.fr > Mon Yphise**.

Le Collège des cadres Yphise

Séminaires intra à la carte de réflexion des cadres, MOe ou MOa

Dynamique d'équipe - Réussite du changement - Compétences d'encadrement

www.yphise.fr > le Collège des cadres > Brochure

Résumé décideurs

Nous sommes convaincus de l'importance du logiciel libre ou free software.

Mais attention, à cette notion est encore souvent attachée une vision naïve de gratuité. Apparu dans les années 80 teinté d'un idéal de dépassement de la société marchande, le logiciel libre est maintenant au cœur de l'économie de l'industrie du logiciel.

L'industrie du logiciel fonctionne mal. Elle considère le logiciel libre comme l'une des clés pour accroître revenus et profits.

Le logiciel libre est un sujet passionnant et complexe, sur lequel il y a encore beaucoup à inventer. Tout manager en informatique d'entreprise doit avoir les idées claires sur ce qu'est et n'est pas le logiciel libre.

En particulier la notion de logiciel libre, qui est affaire de liberté, est distincte et plus riche que celle d'open source, c'est-à-dire d'accès au code.

La liberté d'un logiciel s'apprécie par la licence d'utilisation qui lui est attachée. La licence de référence est la GPL (General Public Licence). Mais il existe une grande diversité de licences qui ne sont pas équivalentes.

La distinction fondamentale entre ces licences est d'être ou non copyleft. Une licence dite copyleft est nécessaire pour garantir que le logiciel ne devienne pas propriétaire dans certaines versions.

A situation fonctionnelle suffisante pour un besoin, un logiciel libre selon une licence GPL est a priori préférable à un logiciel propriétaire. Cela est bien sûr vrai pour tout ce qui est outillage de l'informatique. Mais c'est également vrai sur les logiciels critiques, c'est-à-dire pour supporter les systèmes d'information en production.

Ceci étant, beaucoup de domaines indispensables à la compétitivité des systèmes d'information d'une entreprise ne sont pas couverts par du logiciel libre. Il ne faut alors évidemment pas différer l'investissement au prétexte de ce manque.

Attention, libre ne veut pas dire gratuit, ni même moins cher qu'une solution propriétaire. Libre ne signifie pas non plus s'affranchir des contraintes d'un éditeur ou des conséquences des rachats d'éditeurs (ex MySQL racheté par Sun puis Oracle).

Nous attirons l'attention sur les coûts cachés. Une raison fondamentale de l'intérêt de l'industrie du logiciel est de passer outre les processus mis en place dans les entreprises pour rationaliser l'achat. Il faut donc veiller à disposer des contrôles utiles.

Enfin, il n'y a pas de loi naturelle d'évolution de l'industrie du logiciel vers du logiciel libre sous licence GPL. Les éditeurs ont souvent des stratégies qui permettent un retour au propriétaire. Les donneurs d'ordre doivent mettre tout leur poids pour exiger lorsque possible les garanties nécessaires.

Nous souhaitons qu'à la lecture de ce rapport vous soyez convaincus que le logiciel libre n'est pas une philosophie mais un levier puissant et complexe des affaires à savoir utiliser avec efficacité.

Yphise
Août 2009

Enjeu et plan

- Enjeu
- Résultat
- Plan

Enjeu

Le logiciel libre est maintenant un phénomène majeur de l'industrie du logiciel.

A l'heure où les entreprises cherchent à réduire leurs coûts, alors que les éditeurs profitent d'une position dominante pour augmenter leurs tarifs, tout manager a besoin d'une compréhension de fond des opportunités et limites du logiciel libre.

Résultat

Ce rapport montre tout d'abord que l'industrie du logiciel a besoin du logiciel libre. Elle est amenée à se réinventer autour de modèles économiques qui prennent en compte, à des degrés différents, le logiciel libre ou les idées dans cette mouvance.

Ce rapport montre que le logiciel libre est au cœur des affaires ; il faut définitivement sortir d'une vision simpliste qui en ferait une option philosophique en dehors des enjeux de revenus et de profits d'une industrie qui recherche la croissance.

La notion de logiciel libre est à bien des égards géniale ; elle permet une dynamique sur laquelle il y a encore beaucoup à inventer. Sa bonne prise en compte par les DSI reste en partie à découvrir. On peut néanmoins amener d'ores et déjà différents résultats importants que ce rapport présente.

Plan

1. La première partie présente les notions essentielles dont tout manager en informatique d'entreprise à besoin (**1. Les notions essentielles**).
2. La notion de logiciel libre est née en opposition avec le logiciel propriétaire prôné par les éditeurs. Aujourd'hui, l'industrie du logiciel s'approprie cette notion. La seconde partie vise à faire comprendre ce basculement (**2. Le besoin des éditeurs**).
3. La troisième partie discute les opportunités de prise en compte par les DSI (**3. La prise en compte par les entreprises**).

1. Les notions essentielles

1.1 Le logiciel libre

1.2 Les licences

Ce chapitre présente les notions essentielles dont tout manager en informatique d'entreprise a besoin.

Nous attirons l'attention sur l'importance de ces différentes notions. Elles ne sont pas triviales. Des éditeurs ont tendance à introduire ou à laisser faire la confusion.

1.1 Le logiciel libre

- La liberté
- Logiciel libre et open source
- Logiciel libre et gratuité

La liberté

La notion de logiciel libre (free software) est apparue au début des années 80 à l'occasion du développement de Linux.

La définition d'un logiciel libre est gérée par la Free Software Foundation (www.fsf.org).

Un logiciel est libre lorsque sont garanties quatre **libertés** (freedom).

- La liberté d'exécuter le logiciel, pour n'importe quel besoin (freedom 0).
- La liberté d'étudier le fonctionnement du logiciel et de le modifier pour l'adapter à ses besoins (freedom 1).
- La liberté de redistribuer des copies (freedom 2).
- La liberté d'améliorer le logiciel et de publier ces améliorations (freedom 3).

La seconde et la quatrième libertés impliquent l'accès au code source.

Un logiciel libre garantit ces libertés pour l'utilisateur.

Un logiciel qui ne garantit pas ces quatre libertés est dit **propriétaire**.

La notion de logiciel libre est fondamentalement à l'encontre des réflexes d'un éditeur. Elle n'est pas naturelle pour un éditeur qui a besoin de protéger sa création. Nous verrons dans la suite de ce rapport les raisons de l'intérêt des éditeurs pour le logiciel libre.

Logiciel libre et open source

La notion d'open source n'est pas aussi rigoureusement définie que celle de logiciel libre.

Un logiciel open source signifie que l'utilisateur a accès au code.

Cet accès est une condition du logiciel libre. Mais un open source n'est pas forcément un logiciel libre, c'est-à-dire auquel sont attachées les quatre libertés.

Il faut faire très attention lorsqu'un éditeur parle d'open source. Le terme open source est parfois utilisé comme synonyme de logiciel libre et parfois dans son sens littéral de code source ouvert. Ce n'est pas du tout la même chose. Certains éditeurs n'hésitent pas à entretenir la confusion, notamment sur leur web.

Les libertés attachées à un logiciel libre sont définies dans une licence de logiciel libre (chapitre suivant). En cas de doute sur l'utilisation du terme open source, il faut regarder si ce logiciel fait l'objet d'une licence de logiciel libre. Si ce n'est pas le cas, ce n'est pas un logiciel libre.

La Free Software Foundation et la plupart des personnes engagées dans le logiciel libre n'utilisent pas le terme open source.

Logiciel libre et gratuité

Un logiciel libre ne signifie pas qu'il est gratuit.

Ce point est très important. C'est ce qui fait que peut se développer une économie s'appuyant sur la notion de logiciel libre.

Un éditeur peut faire payer l'utilisation d'un logiciel libre.

Il peut par exemple constituer ce qu'on appelle dans le monde Linux une distribution, c'est-à-dire un package prêt à l'emploi d'un ou plusieurs logiciels libres adapté à un type de besoin. Il peut s'engager sur une mise à jour périodique de cette distribution avec une garantie de compatibilité ascendante et un support professionnel. Il est légitime que tout cela se paie.

Par contre, il ne peut pas dans son contrat avec son client mettre des clauses qui restreignent les quatre libertés.

Le logiciel libre est affaire de liberté et non pas de gratuité. Un logiciel libre peut être un **produit commercialisé** (commercial software).

Ce point, maintenant clairement établi, a fait débat pendant de nombreuses années. Ce qui explique qu'il y ait encore des confusions.

A l'origine en effet, les promoteurs du logiciel libre (en particulier R Stallman) s'inscrivaient dans un idéal de dépassement de la société marchande ; la dynamique de collaboration au sein d'une communauté de personnes permettant d'atteindre la gratuité.

Cet idéal n'a pas disparu, comme le montre notamment le mouvement qui promeut Internet en tant que vecteur libre et gratuit de création et de partage du savoir. Il y a toujours des aficionados du logiciel libre qui ne voient pas d'un bon œil sa récupération par le business. Mais la situation est aujourd'hui claire et solide.

1.2 Les licences

- Logiciel libre et domaine public
- Une licence unique ?
- Le garant de la liberté

Logiciel libre et domaine public

Un logiciel est une création intellectuelle. Il est la propriété intellectuelle d'auteurs.

Il y a donc un droit d'auteur (**copyright**) qui protège cette création.

Une création intellectuelle peut être mise volontairement dans le domaine public. Ceci en donne la liberté d'utilisation. Mais, le domaine public est insuffisant pour garantir les quatre libertés du logiciel libre. Il est en effet possible d'utiliser un logiciel dans le domaine public pour en faire un logiciel propriétaire.

Pour cette raison, un logiciel libre n'est pas du domaine public. C'est une création intellectuelle protégée par copyright ; elle a un propriétaire.

De nombreux logiciels libres ont comme propriétaire la Free Software Foundation à la fois en raison de son importance historique (Linux) et parce que cet organisme propose à tout auteur d'un logiciel qui souhaite en faire un logiciel libre d'y déposer cette propriété.

D'autres organismes à but non lucratif ont vu le jour, notamment à l'initiative d'éditeurs, pour jouer ce même rôle (ex Apache).

Un logiciel libre peut aussi avoir pour propriétaire un éditeur. Cette situation est de plus en plus fréquente.

L'utilisation de cette création intellectuelle, protégée par un copyright, nécessite donc une licence. C'est la licence qui garantit les libertés du logiciel libre.

Un logiciel libre est donc un logiciel auquel est attachée **une licence de logiciel libre**. Ce sujet mérite la plus grande attention.

Une licence unique ?

La licence de référence pour un logiciel libre a évidemment été élaborée par la Free Software Foundation.

Il s'agit de la GNU General Public Licence, couramment appelée **GPL** (v1 1999, v3 2007, disponible sur www.fsf.org). La GPL est un texte juridique qui définit les droits d'utilisation d'un logiciel en des termes qui garantissent les quatre libertés. C'est évidemment un texte un peu technique. Mais il comporte un préambule synthétique bien conçu pour une compréhension aisée de l'essentiel.

Un logiciel auquel est attaché une licence GPL est libre ; c'est simple, clair et solide.

A un éditeur qui prétend commercialiser un logiciel libre, il faut lui demander sous quelle licence il le fait. Si la réponse est la GPL, alors c'est bon. Un éditeur qui souhaite vraiment inscrire son offre dans une logique de logiciel libre n'a absolument aucune raison de ne pas appliquer la GPL.

Mais la GPL n'est pas la seule licence de logiciel libre. Nous avons vu précédemment que la notion de logiciel libre s'entendait par la garantie de quatre libertés. On peut formuler de différentes manières une licence conforme à ces libertés.

Néanmoins, lorsqu'un éditeur n'utilise pas la GPL au profit d'une autre, c'est qu'elle contient des clauses restrictives par rapport à la GPL, sans pour autant remettre en cause la notion de liberté.

Pour s'y retrouver, la Free Software Foundation publie la liste des licences qu'elle considère ou non comme acceptable pour du logiciel libre (www.fsf.org). Elle gère également un répertoire de logiciels libres (id).

Le garant de la liberté

Concrètement, la différence entre une licence GPL et une autre concerne essentiellement ce qui est appelé le **copyleft**.

La GPL est copyleft ; de nombreuses licences de logiciel libre ne sont pas copyleft (ex Apache, Eclipse). Ce sujet n'est pas trivial. Il n'avait pas été vu au départ par les inventeurs de la GPL. Les versions antérieures de la GPL n'étaient pas, ou pas clairement, copyleft.

La question de fond est de savoir si un logiciel libre doit ou non rester libre dans toutes ses versions à venir.

Une licence copyleft répond par l'affirmative. Elle exige que toute nouvelle version du logiciel soit libre.

Une licence non copyleft répond par la négative. Elle permet à l'auteur d'améliorations d'en garder le copyright et de mettre une licence spécifique. Le logiciel de départ reste libre. Mais toute version qui en découle ne l'est pas obligatoirement.

Ce type de licence est d'un grand intérêt pour l'industrie du logiciel. Il permet à un éditeur de prendre un logiciel libre pour, sur cette base, construire des offres propriétaires. Il peut ainsi par exemple développer une version propriétaire pour un environnement donné. Il peut également gérer une gamme de solutions qui va du logiciel libre à des versions propriétaires plus sophistiquées. Un exemple est le développement par IBM de la gamme de serveurs d'application Websphere à partir d'Apache.

L'existence de licences non copyleft est incontestablement un facteur important de prise en compte du logiciel libre par l'industrie du logiciel. En effet, la GPL ne permet pas un éditeur de protéger un avantage concurrentiel lié à l'originalité de sa contribution.

Une autre manière pour un éditeur de limiter la GPL est d'utiliser la **LGPL** pour Lesser GPL conçue également par la Free Software Foundation.

La LGPL est conçue pour lier un logiciel libre à des bibliothèques propriétaires. L'éditeur peut alors concentrer sa valeur ajoutée sur ces bibliothèques. Ainsi par exemple si Red Hat publie Linux sous GPL, il a ressenti le besoin de gérer JBoss sous LGPL afin de garantir sa viabilité économique.

2. Le besoin des éditeurs

- 2.1 Une industrie du logiciel en difficulté
- 2.2 Dysfonctionnements et espoirs
- 2.3 Les modèles économiques

On a vu précédemment que la notion de logiciel libre était née en opposition avec le logiciel propriétaire prôné par les éditeurs. Aujourd'hui, l'industrie du logiciel s'approprie cette notion. Elle est au cœur des stratégies. Il est important de comprendre ce basculement.

2.1 Une industrie du logiciel en difficulté

- Le logiciel au cœur de l'industrie informatique
- L'exigence de liberté a-t-elle obligé les éditeurs à s'aligner ?
- Une industrie du logiciel qui fonctionne mal

Le logiciel au cœur de l'industrie informatique

D'abord, il est utile de noter que le logiciel est considéré par l'industrie informatique comme étant le meilleur réservoir de croissance des revenus et de la profitabilité.

Le hardware est marginalisé. Le service, qui a été le grand vecteur de croissance de la décennie 90, est soumis à une rude concurrence. Les grands de l'informatique, sans évidemment freiner sur les services, souhaitent clairement pouvoir revenir sur une activité capitalistique.

L'exemple d'IBM est significatif. Pendant la décennie 90, l'activité du logiciel était largement perçue comme fournisseur de technologies à l'attention des services ; le discours marketing faisait un grand usage des concepts de suite ou de framework. Depuis le milieu de la décennie 00, cette activité logiciel s'est mise à retravailler, avec succès, l'excellence de chacun de ses produits.

Dans les comptes d'IBM ou HP, l'activité la plus rentable est aujourd'hui le logiciel. Un enjeu clé pour HP est d'ailleurs de rattraper son retard dans la taille de cette activité dans ses revenus.

L'exigence de liberté a-t-elle obligé les éditeurs à s'aligner ?

Nous avons vu que la notion de logiciel libre est aujourd'hui ancienne, puisqu'elle a un quart de siècle ce qui est beaucoup dans une industrie particulièrement remuante.

Or on ne peut pas dire que jusqu'à présent elle ait fondamentalement pressé les éditeurs aux succès les plus éclatants. MySQL ou Ingres n'ont pas empêché le développement d'une situation hégémonique d'Oracle. Le logiciel libre n'a pas réussi à mettre à terre la bureautique Microsoft. SAP s'est développé sur l'ERP sans être inquiété par du logiciel libre. Les grands acteurs du Service Management n'ont pas de concurrence libre.

La situation est en train de bouger.

Le logiciel libre a déjà changé la donne sur le système d'exploitation, une partie du middleware et l'environnement de développement. Mais sur ces sujets, ce sont largement des stratégies d'acteurs, plus que des pressions du marché, qui ont fait basculer la donne.

L'environnement de développement est un excellent exemple. Il est en deux temps. D'abord une stratégie de logiciel libre par Sun pour redéfinir un marché sur lequel il était

absent, en accélérant la rupture technologique à partir de Java. Puis, en réaction, l'accélération du mouvement par IBM en contribuant de manière décisive au développement d'Eclipse. Les dégâts collatéraux de ce combat de géants ont été les Borland, Natstar ou Powersoft. On assiste aujourd'hui à des tentatives ciblées de retour du propriétaire.

Une industrie du logiciel qui fonctionne mal

Aujourd'hui, un grand nombre d'éditeurs ont de façon plus ou moins claire et ambitieuse des stratégies de logiciel libre ou d'open source.

Comme nous venons de l'expliquer, ce n'est pas par philanthropie.

L'industrie informatique compte bien sur le logiciel pour générer du profit, encore plus de profit. La diversité des expérimentations et stratégies des éditeurs en terme de logiciel libre montre bien que l'industrie informatique considère qu'il s'agit de l'une des clés de succès.

L'industrie du logiciel a aujourd'hui besoin du logiciel libre, parce qu'elle est en crise.

Le succès de quelques très grands noms du logiciel cache une industrie en difficulté. La plupart des éditeurs ont une rentabilité insuffisante pour assurer un développement sain de leur offre. On assiste en particulier à une vague sans précédent de concentration par quelques gros acheteurs, en particulier IBM, HP ou Oracle.

Cette concentration se traduit par une augmentation, plus ou moins brutale, des coûts de maintenance, sans rapport avec l'effort de développement des offres achetées. Elle induit des abus de position dominante dans les tarifications.

Cette concentration amène en outre des instabilités ou des disparitions de produits, pour forcer les clients à des évolutions ou des migrations non voulues et sans rapport avec l'intérêt des entreprises clientes. La complexité et les coûts de mise en œuvre et de maintien des solutions grimpent sans bénéfice pour l'entreprise.

Le développement insuffisant des offres par des éditeurs innovants mais qui n'ont pas les ressources et les dérives de la concentration ont pour effet une **crise de confiance** aiguë des entreprises. L'investissement est en berne. Il est frappant de constater l'extrême décalage aujourd'hui atteint entre les propositions de valeur des éditeurs de logiciels et la réalité dans les DSI.

2.2 Dysfonctionnements et espoirs

- Dysfonctionnements
- Innovation sur segments matures
- Prise de décision des entreprises
- Immobilisation de capitaux

Dysfonctionnements

Trois challenges se posent à l'industrie du logiciel pour redresser la barre.

Son intérêt pour le logiciel libre réside dans l'espoir qu'il suscite pour avancer sur ces challenges.

- Maintenir une dynamique d'innovation sur des segments de marché matures.
- Simplifier et accélérer la prise de décision dans les entreprises.
- Accélérer le retour sur investissement.

Ces challenges ont en commun l'enjeu de booster une industrie menacée de léthargie. Ils prennent le sujet sous différentes facettes qui interagissent.

Innovation sur segments matures

L'industrie du logiciel a un besoin permanent d'innovation sur les solutions déjà vendues. C'est vital.

Ses revenus sont aujourd'hui aux trois quarts liés à la facturation de maintenance ou d'extensions. Il faut donc amener régulièrement de l'innovation dans les solutions existantes.

Cet enjeu est très difficile pour les éditeurs car l'expérience montre que les équipes en place se sclérosent.

Les éditeurs ont tout essayé en terme de gestion de ces équipes et de leur renouvellement. Cela ne suffit pas. L'histoire de l'industrie du logiciel a maintes fois démontré ce phénomène, avec des éditeurs qui a un moment donné ont apporté une véritable innovation sur le marché, pour ensuite se conter de vivre grassement.

Cet enjeu est d'autant plus difficile que sur beaucoup de domaines, on a maintenant de bons outils.

Une idée fondamentale du logiciel libre est celle de communauté de contributeurs, d'origine et d'intérêts diversifiés, confrontés à des problèmes opérationnels à résoudre. Cette **communauté est le remue méninges** permanent pour éviter l'endormissement

d'une solution et garantir son adéquation à de vrais besoins opérationnels versus des grandes idées de la R&D.

Cet enjeu du logiciel libre, qui peut paraître un peu secondaire vu d'une DSI, est fondamental pour un éditeur.

Prise de décision des entreprises

L'industrie du logiciel est maintenant confrontée à la rigueur des processus d'achat, à la rationalisation du portefeuille de fournisseurs et à la lourdeur des prises de décision des entreprises.

Cette situation est particulièrement exacerbée avec l'organisation des grandes entreprises en processus souvent trop compliqués. Ils font en particulier fréquemment faire le « tour de la planète » à des dossiers qui auraient été traités localement, avec efficacité et en quelques réunions, il y a encore trois ou quatre ans.

L'enjeu est donc de baisser de manière drastique le coût et l'effort d'entrée d'un logiciel dans une entreprise. La solution la meilleure est de donner un accès gratuit, au moins dans un premier temps et sur une version de base.

Ce type de stratégie n'exige pas forcément du logiciel libre.

Auparavant l'industrie du logiciel avait utilisé la facilité de piratage dans cet objectif. En particulier, Microsoft a été l'un des premiers à comprendre l'importance de **remplir au plus vite un marché du logiciel**. Un moyen pour ce faire a été de ne pas entraver le piratage par des utilisateurs pas encore matures pour payer.

Pour un éditeur, une autre manière de répondre à cet objectif est de donner un accès gratuit à un logiciel ou à un de ses modules. C'est une stratégie qui fonctionne (ex Tibco), mais « la ficelle est peu grosse » pour réellement faire tomber les barrières de protection qu'une entreprise met via ses processus.

Le logiciel libre amène une efficacité nouvelle dans ce type de stratégie.

Sa bonne utilisation par un éditeur permet de gérer avec beaucoup de finesse la progressivité des prix, de l'engagement contractuel et de la dépendance du client. Ceci répond en outre à un vrai besoin d'expérimentation et de réactivité dans des entreprises qui se sont laissées étrangler par leurs processus.

Immobilisation de capitaux

L'industrie du logiciel est financée par du capital développement qui cherche un retour en quelques années.

Un nouveau segment de marché ou une innovation majeure sur un segment existant est identifié. Une levée ou une affectation de capitaux est réalisée de sorte à porter cette innovation et à pouvoir la généraliser le plus vite possible.

La frilosité et la lourdeur de fonctionnement des entreprises font que le cycle de vente, de la compréhension de l'innovation à l'acte d'achat, est devenu économiquement trop long pour cette industrie. L'immobilisation de capitaux est trop élevée et trop longue.

En outre, un marché qui se développe trop lentement court un risque élevé de ne jamais atteindre la taille voulue parce qu'entre temps le contexte ou la technologie a changé.

Il est maintenant avéré que le logiciel libre permet d'amener de nouveaux éditeurs dont le business plan sera plus favorable pour un investisseur que celui d'un éditeur de logiciel propriétaire (voir Red Hat /Tibco au chapitre suivant).

Une raison essentielle de l'intérêt des éditeurs pour le logiciel libre est qu'il s'agit d'**une exigence des investisseurs.**

Le modèle économique de l'éditeur est construit en tenant compte de cette variable. La manière dont elle prise en compte est d'ailleurs aujourd'hui un levier d'innovation important. Elle permet en effet de développer des éditeurs selon différents modèles économiques. Beaucoup de choses sont encore à inventer en la matière.

2.3 Les modèles économiques avec du logiciel libre

- L'innovation en terme de modèles économiques
- Le modèle des distributions
- Le modèle de mise en libre de son logiciel
- Le modèle des stratégies mixtes
- Le modèle du low cost

L'innovation en terme de modèles économiques

On cœur d'une entreprise on trouve son modèle économique (business model). Si celui-ci n'est pas intrinsèquement bon, le capital alloué et le travail réalisé ne donneront rien, même en présence d'équipes compétentes et motivées.

L'industrie du logiciel s'est développée en inventant des modèles économiques adaptés aux caractéristiques particulières de ce secteur, en particulier le fait d'être « une industrie sans usine ».

L'innovation en terme de modèles économiques est plus fondamentale que celle en terme de produits. Le leadership américain dans le domaine du logiciel est lié à une capacité d'innovation plus forte en terme de modèles économiques qu'en Europe, alors qu'il y avait en Europe les ingénieurs pour développer les produits.

Face à ses challenges, l'industrie du logiciel tente aujourd'hui de se renouveler en expérimentant de nouveaux modèles économiques qui ont comme point commun de prendre en compte, à des degrés différents, le logiciel libre ou les idées dans cette mouvance.

Certains de ces modèles économiques ont maintenant fait leur preuve. D'autres sont en expérimentation. Il y a **encore beaucoup à inventer** dans les années à venir. Les prochains grands succès du logiciel seront bâtis sur des idées innovantes en la matière.

La suite de ce chapitre passe en revue ces modèles. Les décideurs en informatique d'entreprise ont en effet besoin de les appréhender correctement pour de bonnes décisions.

Le modèle des distributions

Ce modèle est apparu dans les années 90 (ex création de Red Hat en 95).

Il consiste à fournir un **package prêt à l'emploi** d'un ou plusieurs logiciels libres adapté à un type de besoin. Il est apparu avec le développement de Linux.

Le logiciel libre amène potentiellement la possibilité d'une diversité de contributions qui aboutit à un ensemble de technologies plutôt qu'à une solution clé en main pour une entreprise.

C'est ce qui s'est passé avec Linux. Sont alors apparus des acteurs réalisant des packages, appelés distributions (ex Slackware 93, Red Hat 95). On trouve toute sorte d'acteurs : personnes physiques, associations ou éditeurs.

Red Hat est l'éditeur de la distribution du même nom. La rapidité de son développement et sa rentabilité ont convaincu les investisseurs de la pertinence de ce modèle économique.

Une comparaison entre Red Hat et Tibco est à ce titre intéressante. Ce sont deux éditeurs de taille comparable, nés dans la mouvance Unix et avec comme origine la fourniture de logiciel système. Chacun assure un leadership dans son domaine. Ils ont choisi des voies complètement différentes.

- Red Hat est resté sur la couche système en développant et supportant une distribution Linux sous licence GPL.
- Tibco est monté progressivement dans les couches d'un système d'information avec comme point de départ un MoM (Message oriented Middleware). Tibco a un réel engagement en terme d'innovation de percée pour développer et exploiter des systèmes d'information d'entreprise (MOM > ESB > SOA > BPM > CEP). Ses solutions sont propriétaires.

Le développement de Red Hat (création 95) a été plus rapide que celui de Tibco (85). Sur le début 09, période de crise particulièrement dure, Tibco régresse alors que Red Hat continue à progresser. Il n'en faut pas plus pour convaincre les investisseurs en capital développement d'exiger des éditeurs une stratégie de logiciel libre.

Ce modèle s'applique aux technologies ou frameworks qui justifient des spécialisations : système d'exploitation, serveurs d'application, environnement de développement (Eclipse).

Le modèle de mise en libre de son logiciel

Ce modèle est apparu un peu plus tard que le précédent, au début des années 2000.

Il s'agit d'éditeurs qui choisissent de **développer leur offre logicielle sous une forme libre**, souvent sous licence GPL.

Au départ, la création d'un éditeur de ce type se faisait plutôt après un premier développement du logiciel afin d'en gérer la commercialisation lorsqu'il devenait mature (ex JBoss 00, MyQSL 95).

Les investisseurs ont considéré ce modèle comme suffisamment prometteur pour investir dans des éditeurs partant d'une feuille blanche ou quasiment ; par exemple Collabnet pour la gestion des versions Subversion (licence Apache), Colosa (création 00) pour le BPM Processmaker ou Nexedi (société française création 01) pour l'ERP Erp5.

Ceux qui ont réussi ont donné lieu à des éditeurs viables, qui visent un revenu de la commercialisation du logiciel, de son support et de ses évolutions (ex Red Hat qui a racheté JBoss en 06, licence LGPL).

Mais beaucoup ne se sont pas développés de la manière voulue. Ils visent alors un revenu de services ou de conseil. Ces prestations peuvent être centrées sur leur logiciel ; mais elles sont souvent plus larges pour considérer l'ensemble du domaine adressé. On n'a donc plus affaire à des éditeurs de logiciel au sens strict, mais plutôt à des prestataires de services qui développent leur notoriété et leur expertise en assurant le leadership d'un logiciel libre.

Les investisseurs continuent à croire en ce modèle. Le SGBD Ingres est un bon exemple. Après avoir été propriété de CA jusque en 05, sans succès face à Oracle, le produit a été racheté par des investisseurs et est commercialisé en logiciel libre. Leur objectif est de réitérer le succès d'un Red Hat.

Ce modèle a donc donné des résultats hétérogènes. Il n'en reste pas moins que les succès rencontrés font que l'industrie du logiciel va continuer à le développer.

Le modèle des stratégies mixtes

Ce modèle adresse l'utilisation de logiciel libre, sous licence non copyleft, afin de distinguer précisément ce qui est libre de ce qui est propriétaire.

Le logiciel libre sous licence non copyleft fournit aux éditeurs un moyen très efficace de **maitrise dans la durée de la ligne de partage** entre valeur ajoutée propriétaire et standard de marché banalisé.

L'utilisation intelligente de logiciel libre sous licence non copyleft permet ainsi des stratégies produit sophistiquées pour pénétrer un marché, accroître un revenu ou éliminer un concurrent.

Rappelons les deux exemples cités plus haut concernant IBM : le développement de la gamme Websphere avec un point d'entrée Apache ; la réaction d'IBM contre Sun en contribuant de manière décisive à Eclipse, et au passage supprimer Borland, Natstar ou Powersoft.

Le logiciel libre développé à cet effet est le plus souvent mis dans des consortiums destinés à gérer des boîtes à outils de logiciels (ex Apache, Eclipse). Depuis le début des consortiums (X/Open) et des approches pragmatiques de développement par des communautés (TCP/IP) dans les années 80, l'industrie du logiciel a acquis une grande maturité en matière de mutualisation de composants.

Ce type d'utilisation est le fait de gros acteurs. Lorsqu'ils disent leur engagement sur le logiciel libre, c'est vrai. Mais à leur manière, c'est-à-dire pour établir ou maintenir une domination. L'intérêt des entreprises clientes n'est en effet pas la considération immédiate.

C'est la même situation que des constructeurs automobiles qui s'allient pour développer une nouvelle génération de moteurs ou autres organes. Cela permet une diminution des coûts. Mais cette diminution ne profite pas nécessairement de facto au client. C'est l'existence d'une concurrence et la pression du marché qui font que ce gain ne parte pas dans la poche des actionnaires.

Le modèle du low cost

Ce dernier modèle est le plus récent.

L'idée pour un éditeur est d'entrer sur un marché mature en proposant une approche low cost, c'est-à-dire en centrant l'effort sur la valeur a minima attendu d'un client : un produit, simple et robuste, au meilleur prix.

Citons par exemple Talend sur l'ETL ou Alfresco sur la gestion documentaire.

L'éditeur escompte deux gains principaux. Le premier est la facilité de pénétration d'un compte. Le second est l'excellence du produit par l'engagement de la communauté des clients.

Attention, à la différence du modèle de mise en libre de son logiciel, ces éditeurs ne choisissent a priori pas une licence GPL. Ils se situent dans **une logique plus propriétaire que libre**. Ils se contentent d'ailleurs souvent d'un simple open source. L'idée est d'inventer une relation avec le client qui soit différente, pas de promouvoir les libertés du logiciel libre.

Ce modèle n'a pas encore fait ses preuves. Mais il a incontestablement une place, dès lors que la famille de logiciel considérée est mature (ex ETL).

3. La prise en compte par les entreprises

3.1 Principe de prise en compte

3.2 Opportunités de gains financiers

On aura compris à la lecture précédente que l'intérêt de l'industrie du logiciel pour la notion de logiciel libre constitue un mouvement complexe et en cours. Sa bonne prise en compte par les DSI est donc encore en partie à découvrir. On peut néanmoins amener d'ores et déjà différents résultats importants.

3.1 Principe de prise en compte

- L'innovation de percée
- Le principe de préférence
- Les limites au principe de préférence
- La liberté par le donneur d'ordre

L'innovation de percée

Premier point, on reste loin de couvrir l'ensemble des besoins pour développer et exploiter les systèmes d'information nécessaires à la compétitivité des entreprises.

Plus précisément, le logiciel libre n'a pas encore été capable de générer des innovations de percée en terme de système d'information d'entreprise, c'est-à-dire des technologies ou des concepts qui font faire un saut créatif à l'état de l'art¹.

Ces innovations de percée sont le fait d'éditeurs qui font du logiciel propriétaire. Les exemples sont nombreux, BMC sur le Service management, Dataflux sur la qualité des données, SAS sur le décisionnel, SPSS sur l'analytique, Tibco sur le BPM /CEP, etc.

Le modèle économique d'un éditeur engagé sur des innovations de percée a nécessité jusqu'à maintenant du logiciel propriétaire. La réussite d'une innovation de percée est lente. Il faut plusieurs années pour réussir. L'ERP est un bon exemple : aujourd'hui évident, le concept d'ERP a été lent à s'imposer dans les années 90.

La réussite d'une innovation de percée exige une stratégie dans le temps. Il faut un acteur ayant un engagement fort sur la durée. Il faut la confiance de clients qui vont accepter de financer cet effort d'innovation en surpayant la maintenance des produits existants.

L'industrie pharmaceutique est dans une situation comparable avec un prix du médicament qui incorpore pour une part conséquente l'effort de R&D des suivants. Le logiciel libre qui a réussi en informatique d'entreprise est proche du générique en médicament. La R&D est amortie.

Le logiciel libre fournit ainsi pour une grande entreprise le système d'exploitation, le SGBD, le serveur d'application et autres middleware de base, et l'environnement de développement. Mais il n'apporte aujourd'hui pas l'ERP, le BPM, le CEP, les systèmes décisionnels ou le Service Management.

Le point important est qu'il ne faut pas différer un investissement dans un système d'information d'entreprise sous prétexte qu'il n'y a pas de solution libre. D'autant plus

¹ Notons que les innovations du logiciel libre du Web 2.0 ont encore peu pénétré les systèmes d'information d'entreprise.

que la maturation d'un logiciel libre est lente. Enfin, il n'est pas avéré que le logiciel libre arrive, à échéance raisonnable, à faire de l'innovation de percée.

Cela paraît être une évidence, mais nous voyons des entreprises où se développe un engouement peu rationnel pour l'idée de logiciel libre. Il fait passer au second plan **l'enjeu d'alignement** des systèmes d'information pour l'entreprise, ce qui est évidemment une erreur.

Le principe de préférence

A situation fonctionnelle suffisante pour un besoin, toujours préférer un logiciel libre selon une licence GPL ou équivalent à un outil propriétaire.

Une situation fonctionnelle suffisante peut éventuellement être sensiblement plus rustique qu'un équivalent propriétaire.

Les solutions propriétaires ont souvent comme défaut de se disperser pour justifier la facturation en maintenance ou pour attraper le plus grand nombre possible d'appels d'offre.

Il ne faut évidemment pas de compromis sur les caractéristiques fonctionnelles essentielles aux gains attendus. Dans ce cadre, la solution la plus simple se révèle souvent être le meilleur investissement dans la durée.

Privilégier le logiciel libre est évident pour tout ce qui est outillage de l'informatique.

Mais cela est aussi vrai sur les logiciels critiques. Les éditeurs qui ont réussi sur les distributions Linux, comme Red Hat, ont montré que l'on peut avoir du logiciel libre au niveau requis pour une production. MySQL ou Ingres en sont la preuve pour la base de données ; JBoss pour le serveur d'application.

Attention. Ce principe de préférence **ne dispense pas d'une comparaison des coûts**. Donc d'une mise en concurrence avec les solutions propriétaires éventuelles. Un logiciel libre n'est pas forcément moins cher qu'un logiciel propriétaire dès lors qu'un éditeur est nécessaire (voir chapitre suivant).

Inversement, un logiciel libre selon une licence non copyleft ne constitue pas en tant que tel un avantage. Il est certain en effet que l'éditeur va avoir une stratégie d'upgrades ou de versions propriétaires. Un tel logiciel est à préférer a priori uniquement pour de l'outillage avec besoin d'évolution faible (exemple Jmeter sous licence Apache pour des tests de performance).

Un open source ne constitue pas non plus en tant que tel un avantage. D'autant que la bonne pratique de mise en œuvre d'un logiciel acquis est de ne pas aller le modifier.

Les limites au principe de préférence

Il y a cependant deux limites à cette préférence pour du logiciel libre.

La première limite est la nécessité d'une **expertise en interne** rédhibitoire.

De nombreux logiciels libres sont des technologies plutôt que des solutions d'entreprise. En faire une solution d'entreprise a un coût et nécessite des compétences. On ne trouve pas toujours un éditeur qui ait fait ce travail ou à des conditions de prix jugées acceptables. Il faut alors le faire soit même, ce qui est rédhibitoire en deçà d'un seuil de taille.

Ainsi par exemple, on peut aujourd'hui monter un environnement de développement d'applications pour une entreprise en allant chercher du logiciel libre sur Internet. Mais la gestion de cet environnement, avec le choix des bonnes librairies et versions et avec la réalisation des personnalisations indispensables, nécessite un minimum d'une à deux personnes hautement qualifiées à temps plein.

Ceci explique qu'il existe toujours une place pour des éditeurs d'environnements de développement propriétaires et que cela puisse rester une excellente décision, notamment en PME. Chez Yphise nous avons par exemple fait le choix de 4D en gestion il y a vingt ans. Ce choix s'est révélé pérenne sur deux décennies, avec une bonne compatibilité ascendante malgré plusieurs changements d'architecture, et un prix bas.

La seconde limite est l'incapacité à garantir la parfaite **compatibilité d'échange** ou de reprise d'un existant.

La difficulté d'une parfaite compatibilité avec Office est une raison de la solidité de Microsoft en bureautique malgré les assauts de la concurrence depuis vingt ans. Partout où ils le pourront, les éditeurs de logiciel propriétaire utiliseront ce moyen pour freiner l'avancée du logiciel libre.

La liberté par le donneur d'ordre

Il n'y pas de loi naturelle d'évolution de l'industrie du logiciel vers du logiciel libre sous licence GPL.

Les éditeurs ont la plupart du temps des stratégies de retour au propriétaire au moyen d'une licence sans copyleft ou en vendant du logiciel open source et non pas libre. Le mouvement vers du logiciel libre n'est pas irréversible.

Le succès de l'évolution vers du logiciel libre sous licence GPL dépend de vous, donneur d'ordre en entreprise.

Point essentiel : pour les grands donneurs d'ordre, qui ont le poids nécessaire, exiger lorsque possible une licence avec copyleft (donc a priori GPL).

3.2 Opportunités de gains financiers

- Libre n'est pas forcément moins cher
- Distinguer selon la criticité opérationnelle
- « Business as usual » pour le logiciel critique
- Face aux éditeurs sur le logiciel critique
- Eviter les coûts cachés

Libre n'est pas forcément moins cher

Ce qui fait baisser un prix de vente est la concurrence. Ce qui fait l'exagération du prix est la position dominante.

Une solution propriétaire qui dérive d'un logiciel libre n'a aucune raison intrinsèque d'être moins chère. L'éditeur a probablement diminué son coût d'investissement en mutualisant l'effort de développement. Il va chercher à conserver la marge plutôt que de la donner à son client.

L'éditeur d'un produit commercial libre sous GPL a le même objectif de profit qu'un éditeur de logiciel propriétaire.

L'industrie du logiciel aboutit toujours à une situation dominante (Oracle, Microsoft, SAP, etc.). Le logiciel le mieux conçu pour un besoin, supporté par une commercialisation efficace, finit toujours par dominer.

La raison essentielle est que l'industrie du logiciel est « sans usine ». Entre deux voitures, vous pouvez préférer l'une, mieux conçue, et acheter l'autre parce qu'elle est moins chère en raison d'une fabrication plus efficace. La compétitivité du produit se fait à deux niveaux : la R&D et la fabrication. Dans l'industrie du logiciel, il n'y a pas de différenciation par la fabrication (il suffit de recopier des fichiers). Seule reste la différenciation par la conception du produit. Un bon logiciel est rapidement reconnu comme tel. Il finit toujours par dominer, sauf stratégie commerciale défaillante ce qui arrive mais est peu fréquent.

Il est important de bien comprendre que **le prix est d'abord affaire de concurrence.**

Un logiciel libre sous GPL favorise clairement la concurrence. C'est un facteur de pression sur les prix. Mais il ne supprime pas la tendance structurelle de l'industrie du logiciel à évoluer vers des positions dominantes.

L'exemple de Red Hat est éloquent. Sa distribution Linux devenant incontournable, il a pu se permettre des augmentations dignes d'éditeurs propriétaires.

Distinguer selon la criticité opérationnelle

La première distinction à faire en terme d'opportunités de gains concerne l'outillage interne de l'informatique du logiciel destiné à supporter les systèmes d'information en production.

Un logiciel critique à la production nécessite un support en conséquence et des nouvelles releases, validées et avec compatibilité ascendante, selon une périodicité adaptée. Il faut donc passer par un éditeur. Cela a un prix.

Un **outil sans criticité particulière** peut s'utiliser sans prestation de support de production. Bien souvent, le besoin d'évolution est faible. On peut alors utiliser tout logiciel libre mis à disposition gratuitement. Il faut mettre en face de cette gratuité l'effort de mise en œuvre afin d'apprécier s'il y a réellement ou non un gain. Le plus souvent, il est significatif.

La situation du **logiciel critique** nécessite une discussion plus précise. On est dans la réalité économique d'un éditeur qui fournit une prestation pour en vivre et en retirer du profit. Il n'y a pas de miracle ou de gain type. Le gain n'est pas automatique ; on retombe dans le « business as usual ».

« Business as usual » pour le logiciel critique

Comme pour le logiciel propriétaire, on se trouve dépendant d'un éditeur.

Le constat important est que le logiciel libre **ne modifie pas significativement la structure de coût** d'un éditeur.

Reprenons notre comparaison Red Hat /Tibco. Le tableau montre tout d'abord qu'il s'agit de deux entreprises efficaces (Income of operations) et bien gérées (General and administrative). On constate ensuite que les coûts de R&D d'une part et de marketing et commercial d'autre part sont de même ordre.

Mars Avril Mai 09	Red Hat	Tibco
Cost of revenue	15%	28%
R&D	20%	18%
Sales and marketing	37%	33%
General and administrative	14%	8%
Amortization of intangible asset	0%	3%
Income of operations	14%	10%
Total	100%	100%

La construction du prix est d'abord affaire de situation concurrentielle et non de distinction entre logiciel libre et propriétaire. On peut avoir une situation concurrentielle favorable au client sur un marché d'offres propriétaires.

Ceci étant, il est clair qu'un marché avec une offre mature de logiciel libre sous GPL permet de limiter considérablement les situations dominantes. Le système d'exploitation est l'exemple le plus convaincant. Le plus gros acteur Linux est Red Hat. Sa capacité à imposer sa volonté au marché n'a rien de comparable avec celle de Microsoft ou d'IBM sur leurs systèmes d'exploitation propriétaires.

Plus le logiciel est critique et plus il est aisé pour un éditeur de remonter sa tarification en jouant sur les besoins de support et d'évolution. Ainsi par exemple Ingres met en avant son coût par rapport à Oracle. Ingres, qui est minuscule à côté d'Oracle sur un marché mature, n'a pas d'autre solution que de facturer nettement moins cher. Mais quitter Ingres après l'avoir mis en œuvre pour des applications critiques nécessiterait un effort de migration. Ingres développera donc un potentiel d'augmentation de ses tarifs au fur et à mesure de sa réussite chez chaque client. Le fait qu'Ingres soit moins cher qu'Oracle est davantage lié à la situation concurrentielle qu'au logiciel libre.

Le logiciel libre ne fait pas non plus sortir des logiques d'acquisition des éditeurs. Un bon exemple est MySQL qui s'est développé en concurrence d'Oracle. Après avoir été acheté par Sun en 05, il se trouve dans le giron d'Oracle par le rachat de Sun en 09.

Enfin, la plupart des éditeurs savent maintenant utiliser des logiciels libres sous licences non copyleft ou LGPL pour gérer astucieusement des différenciations propriétaires.

Face aux éditeurs sur du logiciel critique

Dès lors que l'on est face à des éditeurs, il ne faut pas a priori selon que la solution est dite libre ou propriétaire. L'important reste :

- la négociation contractuelle,
- l'organisation de clubs utilisateurs puissants,
- l'utilisation simultanée d'offres concurrentes lorsque pertinent.

Nous insistons sur ce dernier point qui constitue une opportunité du logiciel libre à bien utiliser. Il accroît en effet sur différents domaines la possibilité d'utiliser deux solutions, celle du leader et une autre, afin d'avoir le recul sur la valeur ajoutée et les coûts du leader (ex Ingres versus Oracle). On peut alors jouer sur l'évolution de l'équilibre entre les deux.

Nous insistons d'ailleurs sur l'importance à choisir un éditeur dont la logique de croissance et la surface sont cohérents avec le besoin de son entreprise, sans a priori (ex 4D voir chapitre précédent).

En particulier, plus le domaine est mature d'un point de vue fonctionnel, et plus l'important concerne la satisfaction par rapport au support et à la prise en compte des demandes d'évolution. On n'a alors pas forcément besoin d'un leader mondial avec une prise de décision à plusieurs milliers de kilomètres. Un acteur local dont on fait partie des grands clients peut être préférable (ex réussite d'Absyss sur l'ordonnanceur).

Ce sujet est important pour les années à venir. De nombreux segments du marché du logiciel sont maintenant fonctionnellement à maturité. On ne doit alors pas se limiter à la question du logiciel libre, mais se poser celle du choix de l'éditeur à même de satisfaire au mieux l'entreprise. On peut arriver au choix d'un éditeur de logiciel propriétaire qui permette une nette diminution des coûts, effort de migration compris.

Nous pensons que l'objectif de diminution des coûts sur le logiciel critique doit être vu avec **un regard plus large que le strict logiciel libre**. Il faut prendre le sujet par le choix de l'éditeur.

Eviter les coûts cachés

Nous avons vu qu'une motivation clé des éditeurs de logiciel pour le logiciel libre ou l'open source est de passer outre les processus mis en place dans les entreprises pour rationaliser l'achat.

La gratuité dans un premier temps ou sur une version de base amène le risque de commencer à utiliser des logiciels qui deviendront par la suite incontournables et coûteux.

Nous voyons clairement ce phénomène se développer dans les DSI.

C'est notamment le cas lorsqu'il est fait appel à des prestataires de services qui ont un intérêt évident à privilégier du logiciel libre gratuit. Ils évitent la relation avec les éditeurs et le coût des licences. La DSI se retrouve ainsi à utiliser des logiciels non explicitement choisis.

Il est important que la décision d'utilisation d'un logiciel libre ou open source, même gratuit, fasse l'objet d'une décision explicite. Les processus de la DSI doivent garantir ce contrôle.

A l'inverse, il faut aussi prendre garde à ne pas fermer la porte à toute innovation par des réflexes de standardisation d'architecture appliqués sans discernement. Pour sortir d'une domination CA, IBM, Microsoft, Oracle, SAP ou autres, il faut bien à un moment donné accepter l'utilisation de plusieurs solutions à peu près équivalentes.

Pour garantir cette capacité d'innovation, il faut une réflexion qui ne soit pas uniquement de rationalisation d'une architecture de système d'information. Comme l'a montré l'ensemble de ce rapport, à enjeu économique, il faut des raisonnements économiques qui dépassent le périmètre d'un processus d'architecture habituel. Concrètement, le cadre de référence est défini par le **Service Strategy ITIL v3** dont la mise en œuvre reste souvent à faire.