

Valorisation de l'open source en tant qu'éditeur

Matthieu BALONDRAGE

5 Février 2020

Table des matières

Glossaire	3
Acronymes	4
0.1 Contexte	8
0.2 Problématique	9
0.3 Hypothèses	9
0.4 Méthodologie de travail / périmètre	10
0.5 Présentation de l'open source	11
0.5.1 L'open source aujourd'hui	11
0.5.2 Les éditeurs open source	14
0.5.3 Le consommateur	15
0.5.4 Où trouver de l'open source ?	16
0.6 Optimisation des ressources	16
0.7 Etude du consommateur	16
0.8 Le marché de l'open source	16
0.9 Plateforme promotrice	17
0.10 Gestion des ressources	17
0.11 Chez le consommateur	17
0.12 Marketing de l'open source	17

Table des figures

Glossaire

Apache L'Apache Software Foundation (ASF) est une organisation à but non lucratif qui développe des logiciels open source sous la licence Apache, dont le renommé serveur web Apache HTTP Server. Elle a été créée en juin 1999 dans le Delaware aux États-Unis. . 12

Creative Commons <https://creativecommons.org/licenses/by-nd/2.0> . 8

Framework Un framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un "squelette" de programme, un canevas. Il est souvent fourni sous la forme d'une bibliothèque logicielle et accompagné du plan de l'architecture cible du framework. Un framework est conçu en vue d'aider les programmeurs dans leur travail. L'organisation du framework vise la productivité maximale du programmeur qui va l'utiliser. . 11

Open Source La désignation open source, ou « code source ouvert », s'applique aux logiciels (et s'étend maintenant aux œuvres de l'esprit) dont la licence respecte des critères précisément établis par l'Open Source Initiative (OSI), c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés . 8

Acronymes

BSD Berkeley Software Distribution. 12

FLOSS Free/Libre and Open Source Software. 8

FSF Free Software Foundation. 8, 11

GNU GPL GNU General Public Licence. 12

OSI Open Source Initiative. 3, 8, 11

SaaS Software as a Service. 13

Preface

Remerciement

FICHE DE CONFIDENTIALITE DES RAPPORTS, MEMOIRES, THESES ET SOUTENANCES PROFESSIONNELS
--

Formation/qualification préparée : Management des Systèmes d'information (MSI).....

Nom-Prénom du stagiaire : Balondrade Matthieu.....

Titre du dossier professionnel : Thèse MSI 2018 : Valorisation de l'open source en tant qu'éditeur ...

Date de la soutenance : 03 – 09 Février 2020

Nom de l'entreprise : Docdoku.....

Nom et qualité du représentant de l'entreprise : Garin Florent, Directeur innovation et technologie, Fondateur

Noms, entreprises et fonctions des membres de jury :

Nom-Prénom	Entreprise	Fonction

Mode de diffusion autorisé
(Cocher la case correspondante)

☐ **Diffusion libre**

Le dossier est conservé en archives au CESI, il peut être librement consulté et reproduit. Il peut être utilisé par les destinataires, les études peuvent faire l'objet de publication....

☐ **Diffusion limitée au CESI**

Les membres du jury rendent leur exemplaire au stagiaire à la fin de la soutenance. Le stagiaire est responsable de cette restitution. Un exemplaire est conservé en archives au CESI. Le dossier peut être consulté pour exemple ou illustration par les stagiaires des promotions suivantes mais il ne peut être ni sorti du CESI, ni reproduit, sauf autorisation expresse de l'auteur et de son entreprise. La mention « Diffusion limitée au CESI, reproduction interdite » doit figurer sur la page de garde.

☐ **Diffusion interdite**

Les membres du jury rendent leur exemplaire au stagiaire à la fin de la soutenance. Le stagiaire est responsable de cette restitution. Un exemplaire est conservé au CESI, à titre de preuve dans le dossier pédagogique du stagiaire. Le dossier ne peut être ni consulté, ni sorti du CESI, ni reproduit, sauf autorisation expresse de l'auteur et de son entreprise. La mention « Diffusion et reproduction interdites » doit figurer sur la page de garde.

Signatures :

Pour l'entreprise

Le stagiaire

Le CESI

Abstract

Introduction

0.1 Contexte

Développement logiciel

Le contexte de cette thèse se focalise sur le monde et l'environnement de développement logiciel au coeur de l'informatique. L'Open Source aujourd'hui, peut s'étendre aux oeuvre de l'esprit dont il ne sera pas traité particulièrement mais désignera les logiciels dit "ouverts".

Sous licence

Une partie du contenu de cette thèse a été inspiré d'un ouvrage sous licence Creative Commons "Paternité-pas de Modifications" 2.0. Les modifications apportés à l'oeuvre originale rendent donc ce document non distribuable.

Introduction à l'open source

Afin de lever l'ambiguïté entre les logiciels "libres" (en anglais "free", pouvant désigner un logiciel gratuit, ou libre) et les logiciels "ouverts", l'expression "Open Source" est apparue en 1998 par Christine PETERSON du Foresight Institute. Richard MATTHEW STALLMAN défend le terme de *free software* à travers son organisme la Free Software Foundation (FSF) Eric RAYMOND et Bruce PERENS créent en 1998, l'Open Source Initiative (OSI), qui délivre le label "OSI approved" aux licences qui satisfont aux critères définis dans l'Open Source Definition.

« L'Open Source permet une méthode de développement de logiciels qui exploite la puissance de l'évaluation par les pairs distribuée et la transparence des processus. La promesse de l'open source est une meilleure qualité, une meilleure fiabilité, une plus grande flexibilité, des coûts moins élevés et la fin de l'immobilisme prédateur des fournisseurs. » - OSI

Le code source est la version d'un programme qui est lisible et intelligible pour l'homme. C'est le code source qui est écrit par l'informaticien, le programmeur, et qui pourra être relu et modifié par d'autres. Les programmes peuvent ensuite être compilés, ce qui produit le code *objet*, ou *binnaire*, ou encore *exécutable*, qui lui n'est pas compréhensible.

Un logiciel *libre*, ou logiciel *open source*, est un programme dont le code source est distribué et peut être utilisé, copié, étudié, modifié et redistribué sans restriction.

Un logiciel dit open source ou logiciel libre ne signifie pas gratuit. Rien n'interdit de faire payer la distribution d'un logiciel bien que celui à qui il sera vendu pourra le redistribuer gratuitement. Dans la pratique il est donc considéré qu'un logiciel open source est généralement gratuit. Nous verrons ultérieurement, les différents moyens de générer de l'argent avec de l'open source par le biais d'intégration, support, formation, développement supplémentaire.

Les deux appellations «open source» et «logiciel libre» sont presque équivalentes, mais correspondent à des écoles de pensées différentes.

La commission européenne adopte le terme Free/Libre and Open Source Software (FLOSS). J'ai pris le parti d'utiliser le terme *open source* pour simplifier la lecture.

Pourquoi l'open source ?

En 2019, une étude de la société RedHat, leader dans le monde de l'open source, menée sur 950 entreprises rapporte que 89% d'entres elles considèrent l'open source comme important si ce n'est plus.

En tant que développeur j'ai découvert l'open source au sein de mon entreprise actuelle, Docdoku. Docdoku développe un logiciel open source aidant de nombreuses entreprises. Je constate également qu'il existe encore beaucoup d'amalgames autour de l'open source, des défauts et abus de langage. Les institutions et entreprises dans le monde de l'informatique ne sensibilisent pas assez les développeurs actuels et en devenir sur l'open source. Les avantages qu'offrent celui-ci, l'aspect juridique, les conditions d'utilisation et bien d'autres caractéristiques propres de l'open source ne sont pas enseignés.

Je pars donc du principe que l'open source ne fait pas assez de "bruit" dans le monde, ainsi je souhaite développer à travers cette thèse, l'intrigation, la passion et l'envie de nous tourner vers l'open source, d'y contribuer et d'en faire la solution privilégiée des développeurs et des consommateurs occasionnels.

Voici deux affirmations que beaucoup d'entre nous ont faites ou peuvent faire.

« L'open source c'est moins bien »

L'open source est souvent associé à de mauvais termes. On peut entendre logiciel gratuit, libre, non supporté par une entreprise, développé par le particulier ou des communautés en tant que passe-temps. Un manque de rigueur peut donc s'en dégager et dévaloriser l'image de celui-ci auprès des éditeurs et consommateurs.

« L'open source n'a pas besoin de moi »

L'open source ne se résume pas seulement au développement de code informatique et est un vaste domaine où chaque individu, quelque soit ses compétences, peut y participer afin de rendre le monde et l'informatique meilleure qu'elle n'est.

0.2 Problématique

Je vais donc vous présenter la problématique suivante :

Comment valoriser, en tant qu'éditeur, l'open source et en faire la solution privilégiée des consommateurs ?

Afin de répondre à cette problématique, je vais m'appuyer sur des hypothèses fondées sur mon vécu en entreprise mais également issues de ma réflexion personnelle dans mon métier de développeur logiciel.

0.3 Hypothèses

Les plateformes promotrices

Selon moi si le nombre de plateforme hébergeant du contenu open source et offrant la possibilité d'y contribuer mettaient plus en avant le contenu et les projets, l'open source serait plus prisé par les consommateurs et l'on constaterait plus de contributions à ces projets.

L'optimisation

Si nous améliorons l'aspect communautaire dans son organisation, donnons accès à des formations, inclure et impliquer un peu mieux les personnes sans pour autant compliquer la participation à celle-ci, nous valoriserons l'open source.

Les envies et besoin des consommateurs

Si les consommateurs possédait des moyens efficaces afin d'exprimer leur souhaits et besoins en terme d'open source, ils en seraient plus satisfait et cela en ferait la solution idéale à leurs yeux.

0.4 Méthodologie de travail / périmètre

Etude de l'open source

Présentation de l'open source

Afin d'avoir un retour sur mes hypothèses, il m'est nécessaire d'avoir une meilleure vision d'ensemble de l'open source et un périmètre défini sur son impact, les conditions, les chiffres et son implication dans le monde.

Les éditeurs open source

Je me positionne du côté des éditeurs, pour diverse raison évoqués précédemment, je souhaite donc en savoir plus sur l'éditeur, ses rôles, son périmètre d'action.

L'optimisation des ressources

Après avoir fait la lecture et l'analyse personnelle de plusieurs ouvrages dont 2 principaux sur lesquels je m'appuierai :

« Réfléchissez et devenez riche » - Napoleon HILL - 1937

« Oser la confiance » - Bertrand MARTIN (†), Vincent LENHARDT, Bruno JARROSSON - 1996

Je souhaite effectuer des recherches sur l'optimisation (et ses moyens) des ressources qui définissent l'open source.

Le consommateur

En étudiant le consommateur, nous verrons les besoins et activités relatives à l'open source.

Le marché de l'open source

Enfin nous verrons l'aspect économique associé à l'open source en étudiant son marché.

Sur le terrain

Lors de mon étude terrain je consoliderai mes recherches dans les domaines cités précédemment pour pouvoir les confronter.

L'analyse et les résultats

Je transposerai alors mes études et relevés afin de valider ou réfuter mes hypothèses et apporter une réponse acceptable à ma problématique en guise de conclusion.

Etat de l'art

0.5 Présentation de l'open source

0.5.1 L'open source aujourd'hui

Faire de l'open source c'est ...

Un mouvement de pensée

Le logiciel open source est une affaire de liberté. N'importe quel programme devrait être utilisable, modifiable et redistribuable. Les logiciels non libres, "propriétaire", porte atteinte à cette philosophie. Le logiciel libre n'est pas une alternative ou bien un autre business model mais une lutte pour la liberté.

Un grand mouvement est initié par Richard STALLMAN, le père fondateur de la FSF qui soulève un patrimoine extraordinaire de logiciel mis à disposition de tous. Il s'agit d'une véritable révolution pour ce mouvement de pensée profond ou la solidarité sociale et l'entraide en sont les piliers.

Un modèle de développement

L'utilisation d'un modèle de développement open source communautaire est pour Eric RAYMOND un moyen de démontrer la *supériorité* des logiciels réalisés. Plus que des valeurs éthiques, c'est par la création de ce mouvement avec la fondation de l'OSI qu'Eric RAYMOND espère imposer l'open source. Pour certain, ceci apparaît comme une *opération marketing*, pour d'autre comme Richard STALLMAN, il n'est pas permis de jeter les valeurs fondatrices notamment la *liberté*.

Une dimension humaniste et un patrimoine

L'open source permet avant tout d'offrir une chance pour les informaticiens futur de ne pas repartir de zéro, ne pas ré-inventer la roue. Chaque participation de l'Humain apporte sa pierre à l'édifice. Seulement 10% d'un code source est issue de notre création pour 90% de réutilisation de code issus de système d'exploitation, Framework, et autres composants.

C'est ici la valeur ajoutée de l'open source. L'informatique progresse essentiellement car le socle de code qui constitue notre patrimoine s'agrandit.

Respecter des droits

Je ne peux pas vous parler d'open source sans mentionner les licences et les différents droits inhérent à cette mouvement.

Les programmes open source ne sont pas des programmes « sans licences » comme on l'entend parfois. C'est au contraire leur licence qui les fait open source. Ils ne sont pas non plus dans le domaine public, c'est à dire n'appartenant à personne en particulier, ou du moins exempts de droits patrimoniaux.

Lorsqu'un développeur écrit un programme, il en détient les droits d'auteur, le « copyright ». Dans certains cas, ce peut être l'entreprise qui l'emploie qui en détient les droits. Et ce copyright peut être vendu, comme bien immatériel, d'une entreprise à une autre.

Le détenteur du copyright est libre de définir l'utilisation qui peut être faite de son programme :

- Il peut le garder pour lui, en interdire l'utilisation à qui que ce soit.
- Il peut vendre ses droits à un tiers, personne physique ou morale.

- Il peut utiliser son droit d'auteur pour préciser les conditions qu'il pose à l'utilisation de son programme. Il écrit ces conditions dans les termes de la licence d'utilisation.

Il est donc important de bien assimiler la logique suivante : à la base de l'open source il y a la licence, et la licence n'existe qu'à partir du droit d'auteur.

Ainsi tous les logiciels open source ont un propriétaire, ils ne sont pas « à personne », ni même « à tout le monde ». Dans certains cas, ce propriétaire peut être une fondation à but non lucratif, ou bien ce peut être une entreprise commerciale ordinaire. Il peut s'agir aussi de plusieurs coauteurs, en particulier à la suite de contributions successives.

Le détenteur des droits est libre de fixer les conditions de licence, il est libre d'en changer même, et il est libre d'y faire des aménagements ou exceptions, ou de diffuser à certains selon une licence, à d'autres selon une autre licence.

Celui qui reçoit le programme, en revanche, n'est pas libre. Il est lié par les termes de la licence. Certes il n'a pas signé de contrat, mais la licence lui a été bien énoncée, et elle stipule qu'il n'a le droit d'utiliser le programme que sous telles et telles conditions. S'il refuse ces conditions, il n'a pas le droit d'utiliser le programme.

Je vais détailler légèrement cette partie car elle est également une explication des différents modes de pensées et communauté qui ont un impact sur l'évolution de l'open source et mes hypothèses.

Il y a deux grandes familles de licences open source : la famille BSD et la famille GNU GPL. le terme de licence copyleft apparaît pour les premières. « Copyleft » est bien sûr un jeu de mot en référence au « copyright ». Il ne signifie pas pour autant un abandon de droits.

Les licences BSD

La licence Berkeley Software Distribution (BSD) autorise n'importe quelle utilisation du programme, de son code source et de travaux dérivés. Le code sous licence BSD peut en particulier être utilisé, intégré à des logiciels sous licence non open source. Microsoft a repris du code sous licence BSD dans Windows, et que MacOSX est basé sur FreeBSD, une distribution unix BSD.

La seule contrainte spécifique à cette licence est l'interdiction de chercher à tirer avantage de la dénomination de l'auteur, ici l'Université de Berkeley.

Moins de contrainte, plus de liberté : les programmes sous licence BSD sont quasiment dans le domaine public.

Dans la famille BSD, on trouve aussi la licence MIT, et la licence Apache. Les différences entre ces différentes licences sont de l'ordre du détail.

Les licences GNU/GPL

La licence GNU General Public Licence (GNU GPL) est utilisée par 70% des programmes open source.

La licence GNU GPL, se caractérise principalement par son article 2, qui énonce le droit de modifier le programme et de redistribuer ces modifications, qui constituent des œuvres dérivées, à la condition que ce soit sous la même licence GPL.

C'est ce que certains appellent le caractère viral de la licence : elle se communique aux travaux dérivés. Mais je parlerai ici de donnant-donnant.

Qu'est-ce exactement qu'une œuvre dérivée et qu'entend-on par distribuer ?

Vous verrez que la définition est vaste et pas toujours évidente à percevoir.

Est considéré comme une oeuvre dérivée :

- Si l'on prend un programme A, que l'on modifie des lignes de codes pour obtenir un programme B.
- Egalement, utiliser un programme A depuis un programme B, en faisant appel à certaines fonctions (on parlera de «*link*») est considéré comme oeuvre dérivée.
- Comme il existe beaucoup de façon d'appeler un programme A depuis un programme B, on considère donc que si un programme B ne peut pas fonctionner de manière utile sans A alors il est une oeuvre dérivée.

Lorsque l'on « Distribue » un logiciel open source, on livre l'ensemble du code source aux personnes concernées. Commercialiser son programme c'est le distribuer.

A l'inverse il n'est pas considéré comme distribué une oeuvre dérivée qui est utilisé au sein de son entreprise constructrice. Proposer un oeuvre dérivée à travers son service en ligne, comme les Software as a Service (SaaS), n'est pas considéré comme *distribué*.

Je vous détaille cela plus précisément dans la licence AGPL

En synthétisant, l'idée de la licence GPL est que, en tant qu'auteur ou propriétaire d'un programme, je vous donne le droit de l'utiliser et d'utiliser ses sources à condition que vous en fassiez autant.

Ceci à pour effet de diviser le monde en deux « camps ». Si vous êtes du côté GPL, alors tout le patrimoine open source sous GPL vous est accessible sans restriction.

C'est donc ce que j'appellerai ici du **donnant-donnant**.

La licence AGPL (Afferro)

Comme dit précédemment il est possible de prendre un programme sous licence GPL, le modifier et l'utiliser au sein de son organisation sans en livrer les sources. Rien n'empêche non plus d'offrir un service en ligne construit avec une oeuvre dérivée sans diffuser les sources.

A l'ère des services hébergés de type SaaS, ce type d'usage est régulier or il devrait être considéré comme prohibé étant donné qu'il ne respecte pas vraiment le dogme de la FSF. En effet si le programme est accessible via internet, commercialisé généralement mais que les sources ne sont pas disponibles, on devrait parler de distribution.

Pour répondre à cette controverse, une licence AGPL ou Affero GPL a été créée par la société Affero avec la FSF. Elle ajoute un article qui dit que si le programme initial permettait un accès par le réseau et diffusait ses sources par le réseau, alors le programme dérivé doit en faire de même.

L'article est le suivant :

« Si le programme tel que vous l'avez reçu est prévu pour interagir avec les utilisateurs au travers d'un réseau, et si, dans la version que vous avez reçue, un utilisateur interagissant avec le programme avait la possibilité de demander la transmission du code source intégral du programme, vous ne devez pas retirer cette possibilité pour la version modifiée du programme ou une oeuvre dérivée du programme (...) »

Droits d'auteurs

Qu'en est-il des droits d'auteurs lors de la réalisation d'une oeuvre dérivée ?

Le droit d'auteur ou copyright sur un programme est une notion quand à elle assez claire contrairement à la notion d'oeuvre dérivée, source d'ambiguïté.

Si un programmeur écrit du code que son esprit conçoit, il n'est pas en train de violer un quelconque copyright. Lui-même, ou son employeur, est titulaire des droits d'auteur **sur son code**. Plus simplement, on sait quand on enfreint un droit d'auteur.

Statistiques

0.5.2 Les éditeurs open source

Zoom sur l'éditeur

L'éditeur, c'est celui qui détient les droits du produit, en assure le développement, la promotion, la diffusion et le support.

La seule différence avec l'éditeur de logiciels pour l'éditeur open source est qu'il publie son produit sous licence open source. Sinon l'investissement dans le développement du produit et son marketing est le même qu'un produit propriétaire.

Ce modèle a été élu pour permettre de briser les positions acquises d'oligopoles sur le marché du logiciel.

Il s'agit donc majoritairement de petites entreprises éditrices qui font du support et du développement du produit leur credo.

Développer un programme open source coûte (un peu) moins cher pour ces entreprises car :

1. Il peut s'appuyer sur autant de briques que la licence de son logiciel lui permette.
2. Il bénéficie de contribution communautaire, que je détaille ultérieurement.
3. Il possède généralement plus de développeurs passionnés participant à son travail.

Nous parlerons plus en détail du modèle économique des éditeurs dans la partie sur le marché de l'open source.

Généralement, l'éditeur fait le choix de partir sur une licence GPL car elle présente pour eux deux avantages considérables :

1. Du fait de sa popularité, elle est parfaitement lisible et compréhensible ce qui la rend gage de transparence.
2. Elle empêche les autres de se faire de l'argent sur son dos car elle interdit l'intégration du produit dans un développement propriétaire.

L'aspect communautaire

L'éditeur open source a à sa disposition une communauté qui pourra l'aider non seulement dans le support sur les ressources open source qu'il utilise mais également au développement et au support de son œuvre.

Au sein de la communauté il est possible de distinguer deux types d'acteurs :

les développeurs indépendants : Qu'il s'agisse de gloire, de montée en compétence sur un domaine ou d'altruisme, il existe des développeurs qui soutiennent le développement de produit et participent au support.

Les contributeurs et entreprises contributrices : Certaines entreprises favorisent l'aide et autorisent leurs employés à travailler une partie de leurs temps d'activité sur des projets open sources.

Les supports de l'open source

Le support dans le monde du logiciel, c'est la capacité à apporter de l'aide dans l'utilisation du programme et à corriger le programme le cas échéant.

Le support peut s'adresser aux utilisateurs finaux, comme aux exploitants du programme, ou encore aux programmeurs travaillant sur le programme.

Le déploiement de programmes pour des tâches critiques, en particulier dans des entreprises, requiert absolument un support, car le risque d'une situation de blocage est trop important, cela que ce blocage soit dû à une anomalie ou à un mauvais usage, mauvaise configuration, incompatibilité, etc.

Le support de l'éditeur

Du côté des éditeurs open source (MySQL, eZ Publish, OpenERP, ...), la question est différente : l'éditeur est une société commerciale et son business model est essentiellement basé sur son offre de support. Ici donc, le dispositif de support est très proche de celui des produits propriétaires. Pas identique toutefois car *en parallèle, en complément* au support payant de l'éditeur, il existe souvent un support communautaire, plus ou moins vivace selon les produits. Mais le plus souvent, les corrections touchant au code ne sont assurées que par l'éditeur.

Pour les nouveaux éditeurs de l'open source commercial, le support produit est le fondement du business model, il est leur raison de vivre, leur unique source de revenus. On peut donc s'attendre à un support de grande qualité.

Le support de la communauté

Les produits communautaires bénéficient avant tout d'un support communautaire. C'est à dire basé sur le volontariat de développeurs impliqués, qui répondent aux questions des utilisateurs sur les mailing-lists et forums. Et basé également sur le suivi et la prise en charge des anomalies sur les plateformes de développement communautaires.

Lorsque la communauté est active, comme c'est le cas autour des grands produits, ce support communautaire peut être d'une très grande efficacité, d'une très grande réactivité, très supérieur à un support commercial.

Nous verrons plus tard comment améliorer cette gestion de la communauté dans la partie optimisation des ressources.

0.5.3 Le consommateur

Les bénéfices de l'open source pour le client

Bien sûr, les bénéfices économiques sont parmi les premières raisons dans le choix de solutions open source. Même si « libre ne signifie pas gratuit », ces solutions ont toujours un coût de possession sensiblement moins élevé que leurs équivalents propriétaires.

D'autant que les prix de prestations tendent aussi à être moins élevés, car l'ouverture du produit facilite la diffusion de la connaissance.

Mais au fur et à mesure que ces solutions arrivent à maturité, le moindre coût n'est plus le premier critère de choix.

Les principaux arguments sont alors :

- **La non-dépendance**, ou moindre dépendance, par rapport à un éditeur. On sait que changer d'outil peut coûter très cher, et les éditeurs peuvent être tentés de profiter de la vache à lait que constituent ces clients devenus captifs. En anglais, on parle de vendor lock-in, le verrouillage par le fournisseur.
- **L'ouverture** est également un argument de poids. Les solutions open source sont en général plus respectueuses des standards, et plus ouvertes vers l'ajout de modules d'extension.
- **La pérennité** est un autre critère de choix fort, nous y revenons plus loin.
- **Et la qualité** finalement, car dans beaucoup de domaines les solutions open source sont réellement, objectivement, supérieures. Le très grand nombre de déploiements et donc de retours

d'expérience, mais aussi leur modèle de développement et leur intégration de composants de haut niveau, permet à beaucoup de surclasser les produits propriétaires souvent vieillissants.

A quoi on peut ajouter le plaisir, pour les informaticiens, d'utiliser des programmes dont ils peuvent acquérir une totale maîtrise, sans barrière ni technique ni juridique.

0.5.4 Où trouver de l'open source ?

0.6 Optimisation des ressources

0.7 Etude du consommateur

0.8 Le marché de l'open source

Parmi les 4 grands acteurs de l'open source :

Fondation : Les fondations telles que Apache ou Eclipse, sont des organismes à but non lucratif qui stimulent et pilotent le développement de grands produits open source.

Distributeur : Redhat, Canonical (Ubuntu) ou Mandriva sont des distributeurs (très souvent éditeurs par la même occasion). Ils sélectionnent des outils et composants autour d'un noyau Linux, en assurent le packaging, la distribution et le support

Éditeurs : Diffusent des logiciels sous licence open source, ils réalisent la promotion de leurs produits et proposent du support

Prestataires : Ils vendent des services sur l'open source. Il peut s'agir de conseil, d'intégration, de support, de la formation, des solutions d'hébergement, etc.

Je m'intéresse particulièrement aux éditeurs et prestataire de l'open source qui devront mettre en place des solutions pour péreniser financièrement leurs activités.

Comment fonctionne le modèle économique ou « business model » de ces entreprises ?

Etude terrain

0.9 Plateforme promotrice

0.10 Gestion des ressources

0.11 Chez le consommateur

0.12 Marketing de l'open source

Confrontation

Transposition

Conclusion