

Valorisation de l'open source en tant qu'éditeur

Matthieu BALONDRAGE

5 Février 2020

Table des matières

Glossaire	4
Acronymes	5
1 Introduction	9
1.1 Contexte	9
1.1.1 Introduction à l'open source	9
1.1.2 De l'importance de l'open source	10
1.2 Problématique	12
1.3 Hypothèses	12
1.4 Méthodologie de travail / périmètre	12
2 Etat de l'art	14
2.1 Présentation de l'open source	14
2.1.1 L'open source aujourd'hui	14
2.1.2 Les éditeurs open source	15
2.1.3 Le consommateur	17
2.1.4 Où trouver de l'open source ?	18
2.2 Optimisation des ressources	20
2.2.1 Le cerveau collectif	20
2.2.2 Rendre le libre populaire	22
2.2.3 Le modèle noyau-extension, tout un écosystème	23
2.2.4 Eveiller sa communauté	24
2.3 Etude du consommateur	25
2.3.1 le choix du consommateur	25
2.3.2 L'expérience des consommateurs	26
2.4 Le marché de l'open source	28
2.4.1 Les acteurs de l'open source	28
2.4.2 Chez les éditeurs	28
2.4.3 Communication par ses atouts	30
3 Etude terrain	32
3.1 Plateforme promotrice	32
3.2 Gestion des ressources	32
3.3 Chez le consommateur	32
3.4 Marketing de l'open source	32
4 Confrontation	33
5 Transposition	34
6 Conclusion	35
Appendices	

A Contexte de l'open source

37

Table des figures

1.1	Causes principales des réticences à l'open source	11
1.2	Évolution de l'open source dans les années à suivre	11
2.1	Interface de github	18
2.2	Interface de google open source	19
2.3	Interface de github explore	20
2.4	Interface de contributor ninja	21
2.5	Représentation du modèle noyau/extension	24
2.6	Problèmes rencontrés dans l'open source par les utilisateurs	26
2.7	Comportement néfaste dans l'open source	27
2.8	Ce que les utilisateurs d'open source recherchent dans les logiciels	27
A.1	Domaine d'application de l'open source dans l'IT	37
A.2	Secteur d'application de l'open source	38

Glossaire

Conteneurisation A l'image de la conteneurisation qui désigne le terme de transporter la marchandise par des conteneur, dans le domaine du logiciel, il s'agit d'un procédé permettant d'exécuter des applications dans une zone isolée appelée conteneur. L'application stocke les données dans le conteneur et celui-ci se connecte au noyau d'un système d'exploitation. Il n'est pas nécessaire d'installer un nouveau système d'exploitation car un seul est défini comme la base qui possède l'ensemble des conteneurs. . 11

Fork Un fork (terme anglais signifiant « fourche », « bifurcation », « embranchement ») est un nouveau logiciel créé à partir du code source d'un logiciel existant lorsque les droits accordés par les auteurs le permettent : ils doivent autoriser l'utilisation, la modification et la redistribution du code source. . 25

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. . 14

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 . 9

Scalable La faculté d'un logiciel à évoluer dans le temps pérennément en augmentant ses revenus sans avoir à augmenter les effectifs, et donc les coûts. . 30

Web 3.0 Le web 3.0 est considéré comme l'évolution du Web 2.0 qui correspondait à l'ère des réseaux sociaux. Même si la définition n'est pas fixée et encore débattue, on peut dire du web 3.0 qu'il appartient à l'ère de l'« internet des objets », ou Internet of Things (IOT). Il regroupe toutes les nouvelles technologies apparues et déployées largement aujourd'hui: cloud computing, intelligence artificielle, deep learning. . 11

Acronymes

CA chiffre d'affaire. 23, 29

CNLL Conseil National du Logiciel Libre. 23

FSF Free Software Foundation. 9, 14

IOT Internet of Things. 4

OSI Open Source Initiative. 4, 9, 14

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

Chapitre 1

Introduction

1.1 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 oeuvres de l'esprit dont il ne sera pas traité particulièrement mais désignera les logiciels dit "ouverts".

1.1.1 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

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.

Les deux appellations «open source» et «logiciel libre» sont presque équivalentes, mais correspondent à des écoles de pensées différentes. Libre ne signifie pas gratuit. Rien n'interdit de faire payer le logiciel bien que l'utilisateur en fonction de la licence pourra le redistribuer librement, on considère donc qu'un logiciel open source est généralement gratuit.

Pourquoi je vous parle de 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'entre 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.

Voici deux affirmations erronées autour de l'open source que je me suis faites

« 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.

Malgré le projet que développe notre entreprise sous open source, je n'ai pas réussi à obtenir des réponses concrètes sur ce qu'est l'open source, ce que cela implique, auprès de mes collègues de travail et de formation. A ceci, l'on peut ajouter que pour beaucoup, l'open source est signe de stabilité, fiabilité, sécurité.

En plus du manque de connaissance sur le sujet de moi-même et mon entourage, je constate ainsi qu'il existe encore beaucoup d'amalgames autour de l'open source, des défauts et abus de langage. Les institutions et entreprises que j'ai pu fréquenté 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 à 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 en me positionnant du côté des éditeurs.

1.1.2 De l'importance de l'open source

Une étude auprès de 950 entreprises dans l'informatique a été réalisé en 2019 par la fondation Redhat qui a permis de statuer sur l'évolution de l'open source. La moitié de ces entreprises sont situés aux États-Unis, le reste est répartie entre l'Asie Pacifique, le Royaume-Uni et l'Amérique Latine.

L'open source à travers le monde

L'open source est utilisé par beaucoup de personnes et ce à travers le monde selon une étude (voir Annexe 1) sur la diffusion du libre à travers les frontières. Son utilisation est versatile, la moitié des entreprises interrogé l'utilise pour différents secteurs

- Modernisation des infrastructures
- Transformation digitale
- Développement d'applications
- DevOps
- Intégration dans les applications
- Modernisation des applications

Les domaines informatique de l'open source dans l'entreprise

Les nouveaux domaines du numérique en vogue ainsi que certains réservés auparavant aux logiciels propriétaire s'ouvrent sur l'open source. Ceci inclut notamment les outils de gestion du cloud, security, la sécurité, les données analysées, et le stockage (Voir annexe 2)

Les réticences encore présente sur l'open source

Malgré l'impact bénéfique de l'open source pour les entreprises, il reste encore quelques inquiétudes planant au dessus de l'open source et des réticence à son utilisation généralisée.

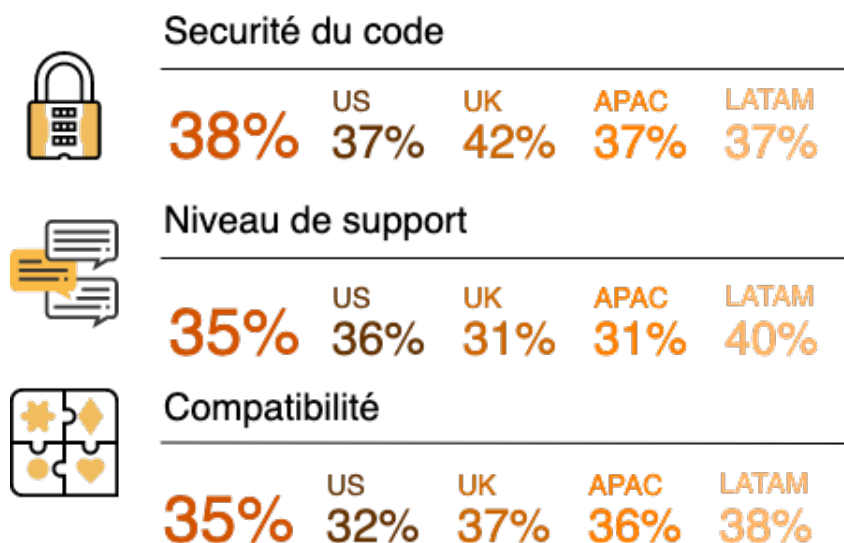


FIGURE 1.1 – Causes principales des réticences à l'open source

Source : redhat.com

J'en conclus donc qu'à travers le hack et le piratage de produit open source, les entreprise positionnent la sécurité de l'open source comme une inquiétude majeure.

L'évolution de l'utilisation de l'open source

Beaucoup d'entreprises continuent tout de même à utiliser des logiciels propriétaires, mais cette tendance tend à diminuer sur les 2 prochaines années. Ceci grâce aux nouvelles technologies provenant du Web 3.0, notamment la Conteneurisation, qui est considéré comme un brassage de produit collaboratif open sources. De nombreuses entreprise aujourd'hui se tourne vers des solutions de conteneurisation, due à l'open source.

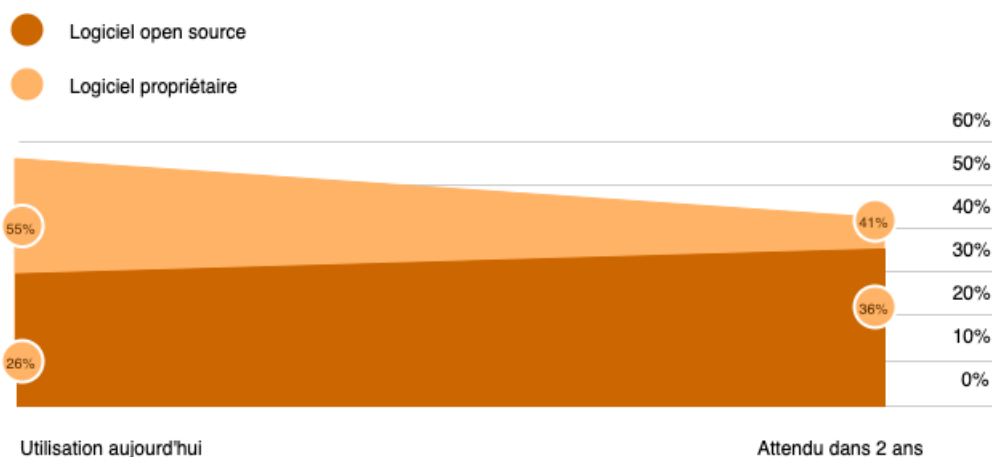


FIGURE 1.2 – Évolution de l'open source dans les années à suivre

Source : redhat.com

1.2 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.

1.3 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 les projets open source, par une démarche plus marketing et publicitaire, utilisation d'images, présentation du projet par une vidéo, 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.

1.4 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.

Les éditeurs open source

Je me positionne du côté des éditeurs, pour diverses raisons 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.

Chapitre 2

Etat de l'art

2.1 Présentation de l'open source

2.1.1 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. Selon Smile, la société leader en Europe de l'open source, 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

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 de changer de modèle même (GPL, Affero, AGPL), 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.

2.1.2 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 position acquise d'oligopoles sur le marché du logiciel. Il s'agit donc majoritairement de petites entreprises éditrice qui font du support et du développement du produit leur credos. Développer un programme open source coute (un peu) moins cher pour ces entreprises car :

1. Elles peuvent s'appuyer sur autant de brique que la licence de son logiciel lui permette.
2. Elles bénéficient de contributions communautaire, que je détaille ultérieurement.
3. Elles possèdent généralement plus de développeurs passionnés qui participent 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 dite 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 oeuvre.

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 autorise leurs employés à travailler une partie de leurs temps d'activité sur des projets open sources.

Parmi ces contributeurs, on retrouve beaucoup de salariés d'entreprise IT et ce pour plusieurs raisons comme :

- Le marketing : statuer que l'on a un développeur qui travaille sur un « Grand projet » pour dorer son image et promouvoir son entreprise.
- La gouvernance : car cela permet d'avoir son mot à dire sur les orientations stratégiques d'un produit
- Le socle technique : Plus il y a de contributions à un socle de produit open source dont l'entreprise est utilisatrice, meilleur sera leur business.
- La maîtrise du produit : Monter en compétence et proposer du support sur ce produit.

Les contributions communautaires

De manière générale, les éditeurs open source ne sont pas trop pour l'apport par le biais de la communauté, du moins sur le coeur de leur produit. Ils les acceptent car c'est dans la logique de l'open source, mais ne les encouragent guère ce qui m'apparaît comme un frein à l'utilisation du plein potentiel de l'open source

Lorsque le code du contributeur est accepté, il passe sous un accord spécifique signé avec l'éditeur qui dispose librement du code. Cela empêche chaque contributeurs de spécifier ses clauses de licence conduisant à un sac de noeud de licences.

Afin de conserver la maîtrise de leur noyau mais d'apporter l'aspect communautaire, les éditeurs utilisent un dispositif d'extensions, qui enrichit le produit indépendamment du noyau . Bien que rapidement évoqué, je détaille plus ce modèle noyau-extension dans la partie sur l'optimisation des ressources.

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.

2.1.3 Le consommateur

Qui est le consommateur ?

Je classe les consommateurs en trois catégories distinctes :

Les end-users : Ce sont les clients finaux du produit open source. Si le logiciel open source a pour vocation d'être utilisé comme tel par le grand public ou bien parmi les entreprises, on considère ces personnes comme des utilisateurs finaux ou « *end-users* ». Ils utilisent le logiciel, remontent leurs besoins d'amélioration, de correctifs et de support.

Les contributeurs et la communauté : consomme l'open source car ils y contribuent, en adaptant certains besoins du logiciel par la création d'extension, par l'aide au développement, et tout soutien qui implique l'utilisation du produit de l'éditeur.

Les autres éditeurs et prestataires : Comme nous l'avons aperçu précédemment, une licence open source permet de bénéficier de toutes les ressources sous la même licence. Les développeurs chez les éditeurs et prestataires réalisent des agrégats de différentes solutions open sources à laquelle ils intègrent la leur. Nous le verrons plus en détail dans la partie concernant l'intégration de solution.

Il est important de considérer que nous sommes tous des consommateurs appartenant à la catégorie « end-user » car aujourd'hui, nous avons tous utilisé au moins une application, une partie de logiciel qui utilise de l'open source. Même sans en avoir conscience.

Favoriser l'open source pour ces consommateurs c'est leur permettre une expérience plus libre dans leurs besoins. Voyons à présent quels en sont leurs bénéfices.

Les bénéfices de l'open source pour le client

Même si les solutions open source ne sont pas toutes gratuites, elles sont en générale moins coûteuses, ce qui en fait un critère de choix essentiel aux yeux des consommateurs.

Le produit étant ouvert, la diffusion des sources tend à réduire le coût des prestations associées car la communauté est de plus en plus compétente et permet de se passer de prestations telles que le support.

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**, par rapport à un éditeur. Comme changer d'outil est souvent cher pour les entreprises, il peut être intéressant de se détacher d'un propriétaire qui souhaite garder sa « vache à lait ».
- **L'ouverture** des solutions open source les rend plus respectueuses des standards, et plus ouvertes vers l'ajout de modules d'extension.
- **La pérennité**, du fait de sa diffusion, le logiciel devient un bien commun que l'on souhaite pérenniser.

- **Et la qualité** due au grand nombre de déploiements et donc de retours d'expérience, mais aussi le modèle de développement et l'intégration de composants de haut niveau, permet de dépasser les logiciels propriétaires.

A ceci s'ajoute le plaisir d'utiliser des programmes dont on peut acquérir la totale maîtrise, sans barrière ni technique ni juridique.

2.1.4 Où trouver de l'open source ?

Que l'on souhaite démarrer son projet open source ou contribuer au patrimoine du Logiciel Libre, de nombreuses plateforme et espace web permettent le stockage et la redistribution de ces projet open sources.

Plateforme de dépôt de code

Github

La plateforme «open source» par excellence centralisant le code source des plus grandes entreprises du monde : Google, Microsoft, Netflix, Facebook, Apple entre autres.

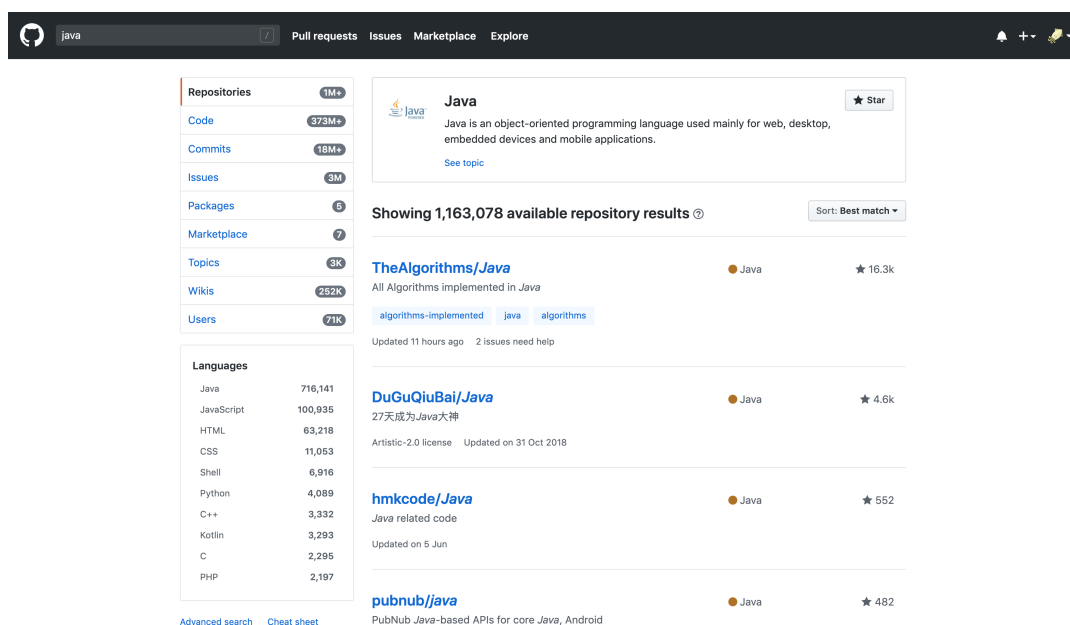


FIGURE 2.1 – Interface de github

Bitbucket

Très similaire à GitHub, Bitbucket est un hébergeur de code source libre qui permet à chacun de s'intéresser à des solutions libres.

Plateforme de promotion de projet

Google Open Source

Avec plus de 2000 projets libres sous sa tutelle, Google Open Source est un incubateur des nouveaux projets libres sur le marché. La plateforme est simple, rechercher un projet comme une recherche google, ou attendre la présentation de différents projets et laisser la magie opérer



FIGURE 2.2 – Interface de google open source

Github Explore

L'interface simplifiée pour démarrer et trouver un projet open source auquel contribuer.

Open source friday

Github propose comme date de rendez-vous tout les vendredi pour contribuer chaque semaines à enrichir le logiciel libre pour les curieux et les plus passionnés.

24 pull request

Chaque année pour la période de Noël, 24 Pull Request organise un moment de développement logiciel sur de l'open source avec comme objectif de remercier les éditeurs et leurs projets qui nous aident tant. En 2018, **1364** contributeurs ont participés sur **3321** différents projets open sources.

Code triage

La plateforme CodeTriage est non seulement utile pour trouver son projet et démarrer rapidement dessus mais nous envoie également un mail quotidien pour nous soutenir et nous rappeler nos engagements.

Contributor ninja

Ici plus qu'un événement c'est une plateforme avec une interface rapide, pour contribuer très rapidement (tel un ninja!) à n'importe quel projets selon le langage de prédilection du contributeur potentiel.

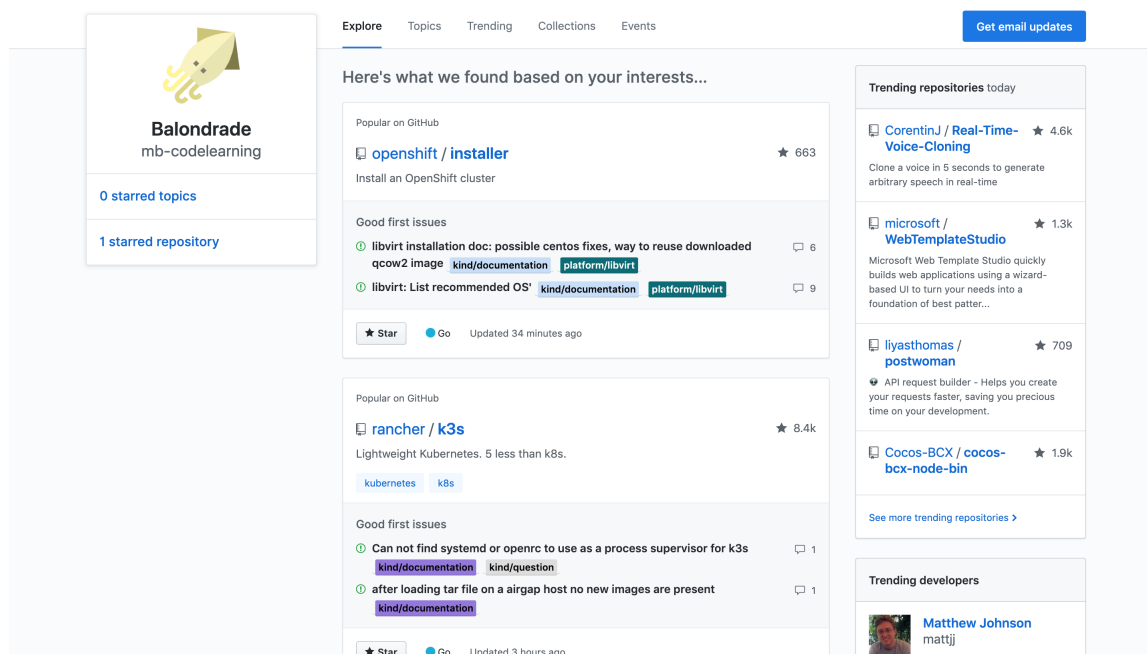


FIGURE 2.3 – Interface de github explore

Up for grabs Pour vaincre toute forme de procrastination et la peur de faire le premier pas (qui handicape si souvent l'homme), up for grabs permet de rapidement trouver la bonne chaussure à son pied et se lancer dans la contribution à l'open source. Le processus est simple :

- Lire les quelques ligne du guide de contribution
- Installer le projet localement
- Laisser un message sur la tâche que l'on s'apprête à faire
- Se lancer dans le travail !

First Timers Only On peut être pris au dépourvu lorsque l'on s'aperçoit que l'open source est un monde bien vaste que l'on ignorait. Pour cela, First Timers Only nous donne les clés pour débiter l'aventure dans l'open source en tant que contributeur.

Pour conclure

L'open source est donc un mouvement important. Il apporte des valeurs de liberté, solidarité mais aussi des bénéfices tant pour les citoyens que pour les entreprises.

2.2 Optimisation des ressources

Concept de l'optimisation

Le succès d'un projet open source dépend pour beaucoup de leur code, mais pas uniquement. De leur concept, de leur capacité à faire connaître le projet, à faire naître une communauté afin d'assurer la pérennité de leur aventure.

2.2.1 Le cerveau collectif

Introduction au cerveau collectif

A travers l'ouvrage de Napoleon HILL, intitulé « *Réfléchissez et devenez riche* », on retrouve plusieurs fois la notion de cerveau collectif.

L'élaboration d'un cerveau collectif constitue à regrouper des membres partageant les mêmes intérêt et passions ou du moins complémentaire. Chaque personnes veillent à maintenir un climat de confiance au sein du cerveau collectif. Au sein du cerveau collectif, l'intérêt est que chaque membre possède sa

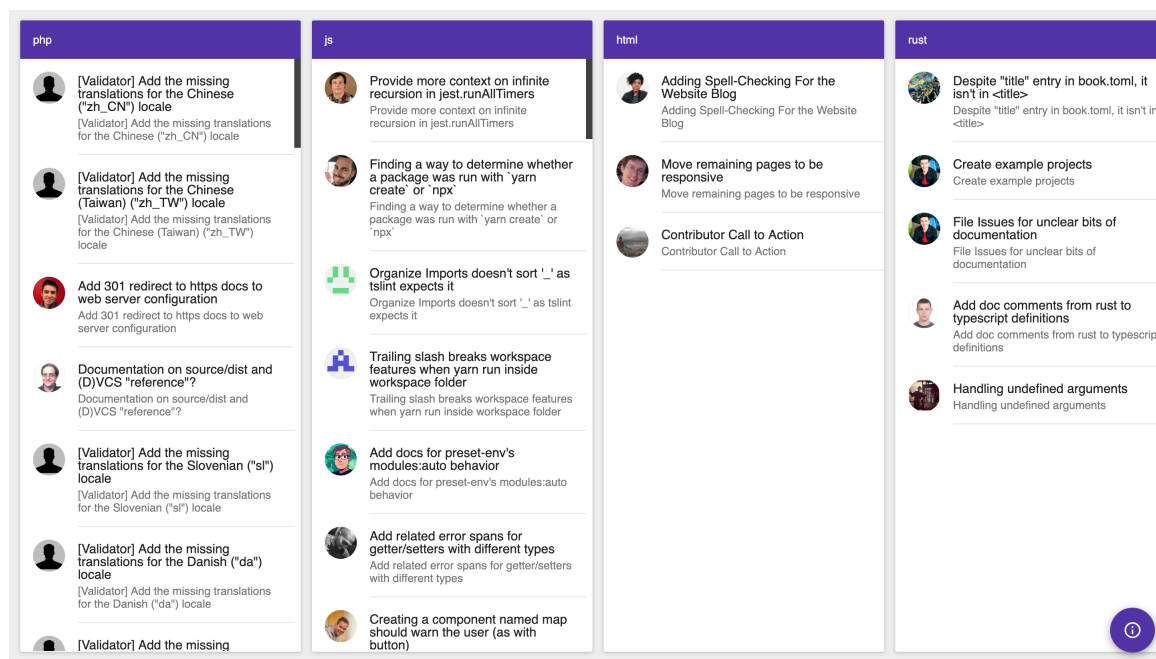


FIGURE 2.4 – Interface de contributor ninja

connaissance, son domaine particulier. Napoleon HILL dans son oeuvre traite de personne instruite. Il définit une personne instruite, tout personne qui sans léser les intérêt d'autrui sait obtenir ce qu'elle veut.

Être instruit

Quand l'on sait que Henry FORD n'est allé à l'école que jusqu'à ses 6 ans. Malgré cela on peut qualifier cet homme d'instruit. Il fût traité de pacifiste ignorant pendant la première guerre mondiale par un journaliste. Afin de prouver son esprit inculte, les avocats des journalistes lui posèrent des questions innatendues sur des sujets variés. Face à tant de colles, celui-ci répondit :

« Permettez moi de vous rappeler que j'ai dans mon bureau une rangée de boutons électriques. Il me suffit d'appuyer sur l'un d'eux pour appeler un homme capable de répondre à n'importe quelle question relative à l'affaire dont je m'occupe personnellement(...), pour-quoi je devrais avoir la cervelle farcie de culture générale alors que je suis entouré de collaborateur qui suppléent à toute lacune ou défaillance de ma part.»

L'intérêt de vous partager cela est de vous faire réagir sur ce concept que l'on essaye tant d'inculquer à notre époque. La multi-compétence. A quoi bon souhaiter être bon dans une multitude de domaine, quand l'on peut être excellent voire une référence dans l'un des domaines qui nous siet tant.

C'est sur ce principe que se base le fondement du cerveau collectif. **Utiliser le plein pouvoir de la connaissance de personnes** si compétentes dans un domaine qu'il serait une perte de temps de chercher à les égaier. Egalement, je rajouterai que le cerveau collectif se veut complémentaire et que l'interaction entre les divers membres permet d'aller encore plus loin ensemble mais surtout indépendamment. Ainsi, on peut définir un cerveau collectif par les concepts suivants :

- Avoir un même idéal, but commun et être sur la même longueur d'onde.
- Un climat de confiance règne au sein de l'équipage.
- Chacun sa compétence qu'il améliore au fil du temps.
- Utiliser en collaboration les compétences de chaque membre dans leur domaine afin d'en résulter un plein pouvoir permettant de mener un projet à l'excellence.

« Si deux esprits travaillent ensemble, ils libèrent une troisième force invisible et intangible qui est semblable à un troisième esprit » - Napoleon HILL - *Réfléchissez et devenez riche*

Imaginez si ce concept s'applique à l'ensemble de la communauté. C'est avec ce principe que naît mon idée d'optimiser la communauté d'un projet open source et le rendre incontournable.

Le cerveau collectif a déjà fait ses preuves pour nombres de personnes. Napoleon Hill décrit le succès du célèbre industriel Andrew CARNEGIE dont je vous invite à lire sa bibliographie. Il a utilisé le cerveau collectif avec son équipe d'une cinquantaine de personnes dans l'industrie de l'acier.

Application du cerveau collectif au sein de la communauté Bien évidemment avant de pouvoir appliquer ce concept il faut le faire accepter par les membres, qu'ils aient le même esprit de travail, ambition ou qu'ils puissent comprendre et se tourner vers ce mode de pensée.

Choisir les membres du cerveau collectif

Bien choisir son équipe est l'élément prédominant dans tout projet. Je pars du principe que l'on souhaite monter une communauté qui nous aidera à travailler sur notre projet. Il me semble alors essentiel avant de fonder le projet, de rechercher les membres essentiels qui composeront le cerveau collectif qui mènera à bien le projet.

L'idée de démarrer dès que possible son projet est certainement une direction tentante pour très vite faire contribuer des membres mais penser dès la conception du projet à monter le cœur du cerveau collectif permet de gagner un temps d'avance sur les étapes futures. Également cela renforcera les décisions prises afin que le projet se stabilise le plus rapidement possible.

Alors avant de démarrer sur les chapeaux de roue, prenons le temps dès que l'on souhaite se lancer dans un projet, de trouver et d'impliquer les ressources humaines essentielles nécessaires à la construction de celui-ci.

Choisir un support de communication adapté

L'outil de communication avec le cerveau collectif sera la principale source d'inspiration, de compte rendu, de retour et d'évolution du projet. Lorsque l'on souhaite se lancer en tant qu'éditeur d'un logiciel open source, il est important de veiller à mettre en place un outil de communication propre et clair sur lequel les décisions seront inscrites durablement. Savoir prendre des décisions précises sur lesquelles nous ne revenons pas fait partie des qualités essentielles d'un dirigeant.

Chaque membre du cerveau collectif est donc responsable des décisions qu'il souhaite faire prendre au projet, à l'équipe, à la communauté et l'on attendra de lui qu'il ne devienne pas une girouette car il représente une partie importante du projet.

Gestion du cerveau collectif

Le cerveau collectif est alors composé des membres du cœur du produit, le « noyau » que je détaille dans la partie sur le modèle noyau-extension et de la communauté qui eux constituent les équipes qui gravitent autour du projet et réalisent les extensions.

Gérer la communauté par domaine de connaissance ou métier permet de catégoriser et focaliser sur les contributions à apporter. Un cerveau collectif central et de multiples cerveaux collectifs qui gravitent autour du noyau permettront de maintenir le cadre d'évolution nécessaire à ce mode de pensée.

La communication au cœur de la stratégie

Et maintenant quoi ? On a un cerveau collectif, on a choisi les membres, on a respecté leur domaine de prédilection, mis en relation ... Mais où est la stratégie pour développer et déployer notre projet ?

La gestion de projet inhérente à l'édification d'un logiciel est un standard que je ne développe pas ici (méthodologie agile, gestion d'entreprise) mais l'on pourra compléter ceci avec l'ouvrage « Oser la confiance » comme nous le verrons dans la partie « Éveiller sa communauté » .

2.2.2 Rendre le libre populaire

Pareto revisité

Une étude sur la participation de développeur et le code total rédigé rapporte que si l'on prend un projet où 200 programmeurs participent, seulement, 10 d'entre eux ont écrit 50% du code. C'est donc

la preuve d'un investissement mal répartis généralement dans la programmation logicielle. Il faut donc veiller à **mettre en place une implication des participant au code** et plus largement au projet.

L'enseignement du logiciel libre

Comme mentionné dans l'introduction de ce document, le logiciel libre me semble pas suffisamment enseigné.

Qu'est-ce que j'entend par enseigner le logiciel libre ?

Susciter l'intérêt du libre

Enseigner un logiciel libre par la mise en place d'efforts spécifiques à celui-ci.

Je retrouve dans le livre blanc «Point de vue sur l'open source » de Smile mon hypothèse concernant le manque d'enseignement de l'open source. Ce n'est pas en utilisant sans le savoir de l'open source ou en cherchant spécifiquement à remplacer tout logiciel propriétaire par du libre que l'on enseigne l'open source. Il est nécessaire d'**expliquer les mécanismes** liés à l'open source, de prendre le temps d'**informer sur toute l'importance** de ce qui gravite autour de l'open source.

Ceci afin de permettre à des centaines de programmeur éparpillé sur la planète à coopérer de façon cohérente sur la réalisation des millions de lignes de code.

Cela passe également, selon moi, par la **mise en relation des étudiants avec les communautés de développeur**.

Améliorer la recherche sur l'open source

Il est nécessaire d'encourager la recherche, comme le souligne Smile dans son livre blanc «Comprendre l'open source», qui se développe autour des logiciels libres et **fournir des outils nouveaux** pour accompagner leur essor.

Les plateformes distribuant l'open source sont à première vue trop réservée à la communauté de développeurs, des « Nerds » qui développent la nuit.

Des spécificité métier, tant dans le juridique et l'architecture logicielle pourrait se dégager afin de veiller au bon respect du libre.

Un gisement d'emplois futur

Tant par ses valeurs humanistes que par la contribution au patrimoine de l'humanité, l'open source doit être vue comme un bien commun qu'il faut cultiver en commun.

Tout autant que l'art qui est exposé dans de nombreux musée que l'on rends accessibles de temps en temps gratuitement pour contribuer à la culture de l'homme, **l'open source devrait être considéré de même**.

Au coeur de l'activité industrielle encore méconnu, l'open source c'est 4,46 Md d' € de CA révèle l'enquête du Conseil National du Logiciel Libre (CNLL), en 2017. 4000 emplois nets ont été estimée d'ici 2020. La France est le leader Européen de l'open source. Et pourtant, le système éducatif actuel ne perçoit pas ce gisement.

2.2.3 Le modèle noyau-extension, tout un écosystème

Dans ce schéma de développement, connu pour déployer son activité open source, on distingue le noyau du produit, sous la responsabilité de l'éditeur, et les extensions, réalisées par la communauté.

Les principes de séparation sont les suivants :

- Le noyau doit être d'une grande robustesse, il est certifié par l'éditeur, les contributions externes y sont rares

- L'interface entre le noyau et les extensions est bien documentée et stable, c'est à dire qu'un changement de version du noyau n'implique pas, du moins le plus souvent, un changement de version des extensions.
- L'éditeur stimule la réalisation d'extensions, car elles donnent de la valeur à son produit et témoignent aussi de l'existence d'une communauté, en soi un gage de pérennité.

Ce **modèle noyau-extensions** est celui qui réalise le **meilleur point d'équilibre** entre les rôles respectifs de l'éditeur et de la communauté, réunissant la garantie et l'engagement de l'éditeur avec le dynamisme et l'énorme capacité de développement de la communauté.

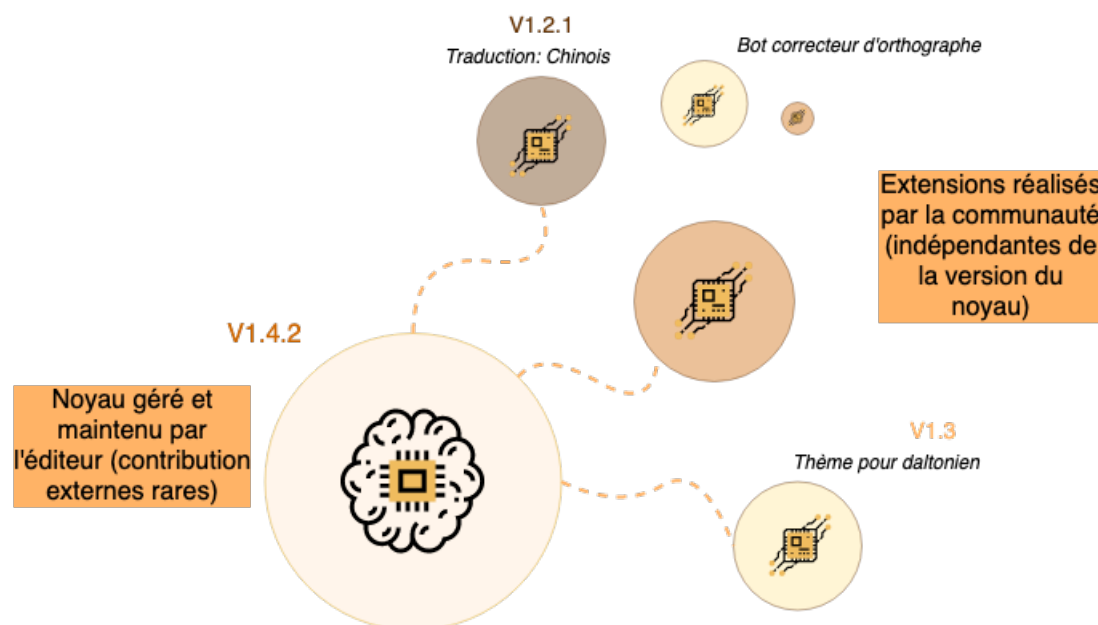


FIGURE 2.5 – Représentation du modèle noyau/extension

Ce modèle d'architecture présente plusieurs avantages :

- Ne pas faire grossir inutilement un programme en se concentrant sur l'essentiel du business et de sa fonctionnalité.
- Jouer sur l'aspect modulable et simple du logiciel et concurrencer les logiciels propriétaires qui, sous pressions, rajoutent de plus en plus de fonctionnalités sur leur logiciel jusqu'à le rendre trop lourd .
- Déjouer la concurrence sur le marché, car la communauté se concentre sur la création et l'amélioration perpétuelle des extensions. Donnant un coup de neuf tout les jours.
- Eviter de destabiliser son logiciel car les fonctionnalités peuvent être ajouté sous forme de modules.

Le modèle noyau-extension permet de tracer cette frontière entre l'éditeur et la communauté tout en s'assurant que la communauté trouve sa place au sein du produit et que chacun puisse répondre à son besoin.

Pour favoriser la mise en place et le maintient du modèle noyau-extension, l'éditeur peut **mettre généralement en place une plateforme pour accueillir ces extensions** afin d'avoir une meilleur visibilité, classer par popularité, trier selon les besoins et de rendre gloire aux auteurs de ces extensions.

2.2.4 Eveiller sa communauté

Afin de faire naître et grandir sa propre communauté autour de son produit open source, il existe déjà quelques points clés souligné dans le livre blanc « comprendre l'open source » de Smile :

- Semer la graine en mettant en ligne son code source. Être patient vis à vis de l'épanouissement de la communauté qui va prendre racine. Celà ne prendra pas quelques jours mais est un travail de longue halaine.

- Être transparent tant dans son projet et ses orientations que dans la gouvernance du projet.
- Choisir et décréter son support d'échange avec la communauté. Afin de centraliser la communauté sur ses outils.
- Instaurer sa politique de l'open source en proposant un modèle noyau-extension, celui qui fonctionne pour le mieux. Bien préciser quels sont les aspects que l'on veut retrouver dans notre vision open source.
- Inspirer des valeurs : le logiciel que l'on va construire n'a pas de but lucratif ou du moins il est moindre. Ceci aide à construire la communauté plus facilement car elle n'aura pas l'idée qu'on génère de l'argent sur son dos. On aspire à aider, changer le monde. Il faudra trouver l'idée d'une mission que les gens adoptent.

Le respect de ces standards permet d'instaurer la communauté au projet. Pour compléter ces bonnes pratiques, j'y ajoute la notion de gestion de la communauté appuyée par les propos tirés du livre « Oser la confiance » qui sont les suivants :

Les valeurs La mission que doit relever l'éditeur à travers son projet open source n'est pas anodine. Les valeurs qu'elle représente peuvent apporter le comportement idéal attendu des contributeurs. Je souhaite donc **mettre en avant lors de la présentation de son projet, les valeurs représentées**.

Dans l'ouvrage « Osez la confiance », une entreprise de construction navale est sur la faillite, le nouveau PDG, Bertrand MARTIN leur indique à défaut de solution qui marche avec un grand succès pour autant qu'elle n'était pas vraiment salvatrice et il témoigne :

« C'est parce que le personnel le voulait, qu'il avait pu s'approprier le projet, qu'il en comprenait les enjeux, que nous avons opéré avec le maximum d'efficacité »

Mettre en confiance Vincent LENHARDT dans son livre nous confie le vrai pouvoir de la confiance, il emploie le mot *empowered* qui définit clairement l'acteur d'un projet à qui l'on attribue toute confiance et dont on apporte toute ressource et aide nécessaire donne alors le meilleur de lui-même.

Il y a donc une attitude à rechercher chez l'éditeur c'est la confiance en ses contributeurs. **Apprendre à faire confiance en sa communauté** en leur confiant des tâches essentielles, avec des enjeux dont ils ont pleinement conscience permet de dépasser ses limites pour atteindre les sommets.

C'est ce qui a permis de relever l'entreprise de chantier naval de Bertrand MARTIN.

Donner l'exemple L'éditeur se doit de faire partie des éléments contributeurs, se mettre non au dessus mais dans le bocal permet à chacun de se responsabiliser mais également de comprendre que l'on est une force selon le dicton « L'union fait la force ». Dans oser la confiance, ce principe est bien détaillé. Vincent LENHARDT, analyse cela en concluant que le pouvoir et l'initiative sera entre les mains des acteurs ce qui les poussera à mettre en œuvre les décisions. On ne recherche plus forcément la compétence de l'éditeur, ni de donner les solutions mais de **déléguer et faire partie du groupe** afin d'améliorer la responsabilité des acteurs.

2.3 Etude du consommateur

2.3.1 le choix du consommateur

L'activité d'un projet Afin d'établir un choix de solution pérenne, le consommateur va s'interroger sur l'historique du projet et les orientations de celui-ci si il provient d'un « Fork » ou si un fork est fortement probable comme le souligne la société OpenDsi dans son article de blog intitulé « Comment choisir un logiciel libre ou open source ». Il pourra alors s'orienter vers toutes sources d'informations disponibles : blog, article, forum de discussion et liste de diffusion des activités liées à ce projet. Il peut être également intéressant de faire attention au nombre de contributeurs au projet car il peut y avoir certaines incohérences autour d'un projet massivement adopté et le nombre de développeurs qui le portent.

La nature des contributions Dans certain domaine, notamment dans celui de l'industrie, un projet open source peut fortement dépendre des contributions d'une entreprise et non d'une communauté. Ceci peut prendre différentes tournures, le projet peut partir vers un modèle propriétaire au bout d'un certain temps. Prendre des décisions qui arrangent l'entreprise en question et qui sera pénalisant pour l'utilisateur final.

Ainsi le consommateur sera davantage **rassuré par le nombre de contributeurs et la diversité de ceux-ci**, signe d'indépendance et de pérennité du projet.

Les droits inhérents de la licence En fonction du besoin du consommateur comme je vous l'ai présenté à plusieurs reprise, les licences vont permettre la ré-utilisation, la commercialisation et bien d'autres actions au projet. Le choix d'un logiciel open source se fait donc également entre la **cohérence du besoin et des idées du consommateur et les permissions accordées** à celui-ci.

2.3.2 L'expérience des consommateurs

Github a lancé une étude auprès de nombreux dépôt de code et de la communauté qui gravite autour. C'est plus de 3800 projets open source et plus de 500 réponses qu'ils ont réussis à obtenir les conclusions suivantes :

Le retour des consommateurs

Beaucoup de critiques font rages auprès des consommateurs et « pilleurs » de l'open source car ils laisseraient sans retour les éditeurs. C'est ce que critique l'article « Les logiciels libres meurent lentement sans contributions » de Framasoft, une société éditrice de solution libre. Malgré la liberté prônée depuis le début, les éditeurs du logiciel libre s'attendent à des retours de la part des consommateurs et utilisateurs finaux. Egalement, les propositions de correction ou d'amélioration sont des retours appréciable du consommateur. Cette contribution permet aux éditeur de progresser et de s'adapter au marché et à la demande. Ainsi il s'agit d'une notion à prendre en compte dans l'optimisation de la communication avec les consommateurs. Il est possible **d'intégrer un module sur la plateforme de promotion qui facilitera les retours d'informations et critiques à l'éditeur.**

Les problèmes rencontrés par les consommateurs

Les consommateurs recensent de nombreux problèmes que l'on retrouve dans l'open source et qui peuvent être un frein à son utilisation. Lors d'une étude réalisé par github, il ressort que la documentation et le support sont essentiel auprès des consommateurs auquel je suggère de faire principalement attention.

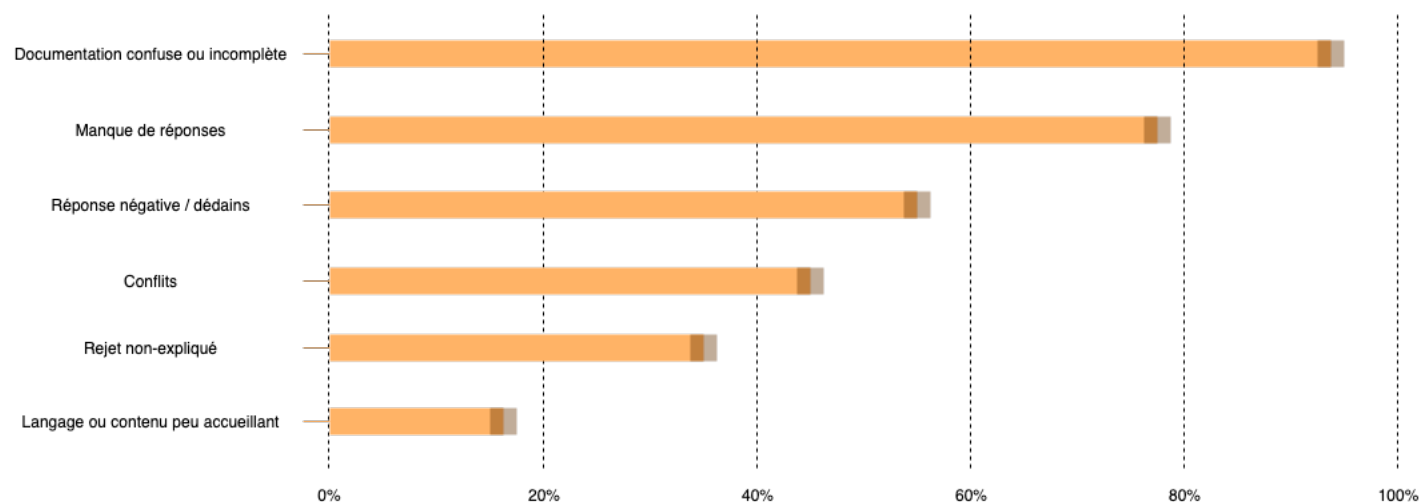


FIGURE 2.6 – Problèmes rencontrés dans l'open source par les utilisateurs

Source : opensource.surveymonkey.com

Un climat néfaste ?

Au sein de projet open source du au fait d'une communauté décentralisé, virtuelle et des enjeux, nous pouvons retrouver de nombreuses violences dans les communication entres consommateur et éditeurs. Le **respect dans la communication et la bienveillance** sont des facteurs essentiels pour la satisfaction client et donc croître le nombre d'utilisateurs.

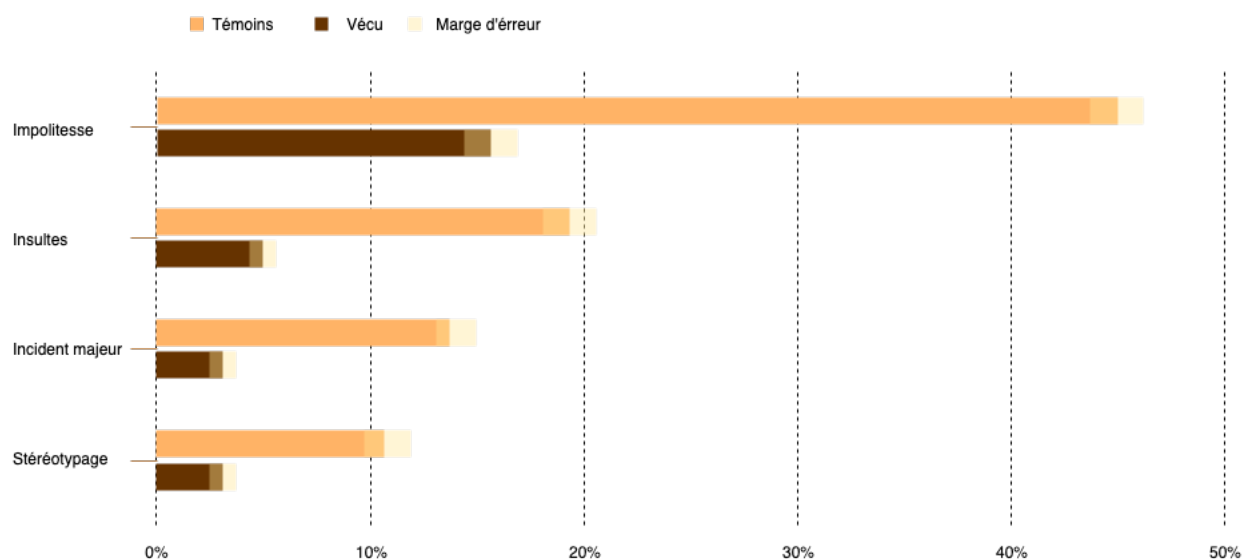


FIGURE 2.7 – Comportement néfaste dans l'open source

Source : opensourcesurvey.org

Ce qui est regardé dans le choix d'un logiciel open source

Sur les utilisateurs interrogés, un classement des critères de sélection d'une application a été réalisé. Les premiers regards des consommateurs sont porté sur la stabilité et la sécurité du projet open source. Je préconise ainsi **l'utilisation d'outils spécialisés pour vérifier, certifier le projet** open source pour rassurer les consommateurs.

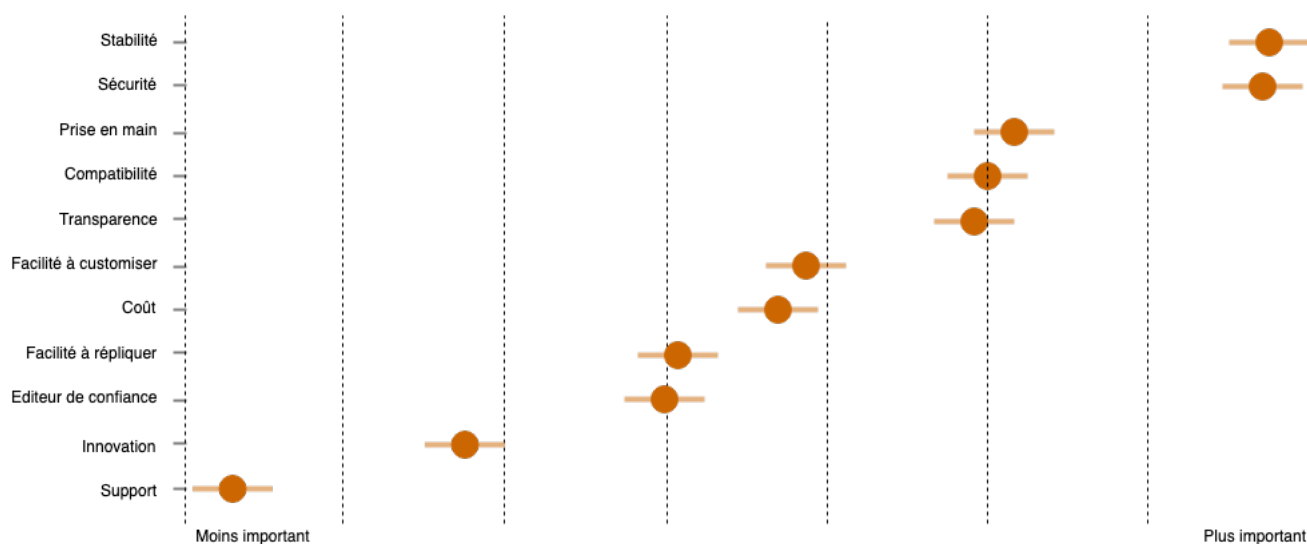


FIGURE 2.8 – Ce que les utilisateurs d'open source recherchent dans les logiciels

Source : opensourcesurvey.org

2.4 Le marché de l'open source

Libre n'est pas gratuit

C'est bien l'un des principes de l'open source. Même si l'on y a pour vocation de s'étendre et de distribuer son produit, il est nécessaire, si ce n'est vital pour certains éditeurs de trouver une source de revenus pour leur logiciel.

Il faut savoir qu'il existe une différence entre faire payer un logiciel propriétaire et un logiciel open source.

Payer un logiciel propriétaire : permet non seulement d'apporter des revenus à une entreprise mais d'obtenir un « droit de possession », d'acquisition du logiciel.

Payer un logiciel open source : N'est pas un prix d'acquisition ni un droit d'utilisation mais une source de revenus à l'éditeur pour permettre au logiciel de prospérer.

2.4.1 Les acteurs de l'open source

Il existe dans l'open source 4 grands acteurs :

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.

Les distributeurs : 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

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

Les 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 de l'open source qui devront mettre en place des solutions pérennisant financièrement leur activité.

Et de la prospérité les éditeurs en ont bien besoin car pour les autres acteurs, la taille, la mission et le produit présenté ne pose plus aucune difficulté de revenu. Doit-on encore se soucier du bon développement de Linux et de sa communauté ? Ces géants de l'open source, association à but non lucratif, ont des moyens marketings (plus de 60% de leur revenus) et financier bien supérieurs aux éditeurs et prestataires qui restent pour la majorité des petites et moyennes entreprises.

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

2.4.2 Chez les éditeurs

Même s'ils bénéficient de l'open source pour réduire leur coût de ressources, celle-ci ne comble pas les besoins de financement de la partie développement en interne, hébergement, marketing. Il est donc nécessaire de trouver des sources de revenus pour les éditeurs.

Parmi celles-ci, on distingue 3 principales sources pour l'éditeur :

- Vendre des licences
- Vendre du support
- Vendre de l'intégration

Vendre des licences

Même s'il est interdit de faire payer l'utilisation d'un logiciel open source, les éditeurs ont bien compris comment faire bon usage de la « double licence ».

Double licence commerciale

Il est possible de distribuer une œuvre dérivée utilisant le programme sans diffuser ses sources à l'aide d'une licence commerciale.

Le programme officiel open source est gratuit mais sa version dérivée elle est payante à travers l'achat de licence commerciale.

Pour l'entreprise MySQL par exemple, la vente de licence représente plus de la moitié du chiffre d'affaire (CA).

Les extensions payantes

L'éditeur peut proposer des extensions aux fonctionnalités présente dans le logiciel open source. Le logiciel initial est suffisamment de qualité et donne envie de payer quelques extension *optionnelles* supplémentaire pour le confort et le besoin de l'utilisateur.

Un support uniquement sur licence commerciale

Ici aussi, le logiciel est sous licence open source mais si l'on désire avoir le moindre support dessus, il faudra se tourner vers son confrère et sa licence commerciale.

Vendre du support

La prestation de support est une source principale de revenu pour l'éditeur, même s'il doit pour cela faire face à la concurrence potentielle de prestataires tiers que je précise juste après.

Le support d'un logiciel inclut généralement :

- L'accès privilégié aux correctif et ressources spécifique
- Une prise en charge des problèmes (anomalies, utilisation, mise en oeuvre ...)
- Des prestation d'audit, de certification, ou de prise de contrôle à distance, ainsi que la surveillance proactive et les corrections.

On peut distinguer différent modèles de financement concernant la vente de support :

Uniquement le support

Certains éditeurs misent uniquement sur la vente de support. Le logiciel est gratuit mais le support lui est payant. Ce modèle de financement fonctionne pour certaines entreprises comme Tiny(OpenERP), Nuxeo.

Le problème est qu'en cas ou le support n'a pas été utile l'année souscrite pour cause de stabilité du logiciel, le client voudra surement le résilier pour la suite.

Plus le produit est de qualité, moins le support est facile à vendre car le client rencontre moins de difficultés.

L'avantage est que plus on avance dans la technologie et plus la concurrence est présente on doit donc sortir des nouveautés constamment ce qui fragilise le produit et le rend instable. Le support est donc précieux dans le cadre professionnel.

Faire payer la stabilité

Pour d'autre éditeurs, la stabilité d'un logiciel peut devenir source de profit. L'idée est de proposer deux logiciel :

- Celui en licence open source sera classifié de « community edition ». Il sera présenté comme instable, pas entièrement testé, à ne pas déployer en production
- Tandis que le logiciel sous licence non-libre sera la licence sécurisée, une version « enterprise-ready », « fully tested ».

Au final, la version community, c'est celle en cours de développement donc en avance de phase alors que la version entreprise c'est celle qui a été *gelée* dans un état dit « *stable* ».

L'éditeur peut alors diffuser et promouvoir son produit à travers la version community, et apâter les entreprises dans la version payante ou généralement le support est packagé avec.

L'éditeur jongle sur la stabilité de la version community en la rendant suffisamment stable pour donner envie de l'utiliser mais inciter fortement les entreprises à prendre la version sous licence commerciale qui sera accompagnée du support.

Fonctionnalités avancées

Pour l'éditeur, un autre business model et celui de la fonctionnalité avancée. La version community et entreprise n'est pas différente d'un point de vue stabilité mais ce sont les fonctionnalités présente qui sont réduites sur la version community.

Ceci permet de se dégager du paradigme « libre = instable » totalement infondé.

La difficulté pour l'éditeur va être d'avoir suffisamment de fonctionnalités pour rendre la version libre intéressante mais d'avoir une forte valeur ajoutée sur les fonctionnalités dans la version payante.

Vendre de l'intégration

Pour les éditeurs qui ne sont pas mondialement connu, il est possible de trouver la rentabilité en proposant l'intégration de son produit open source. C'est un moyen de démarrer dans le milieu sans trop de risque mais qui n'est pas « Scalable ». On ne peut s'étendre à l'étranger si l'on est le concurrent direct de ses intégrateurs partenaires. On reste donc sur un marché réduit.

Autres sources de revenus

Les campagnes de crowdfunding

En Mai 2019, la plateforme d'hébergement de logiciel Github, propose un programme de sponsoring et de crowdfunding.

Vendre des extensions

Selon le modèle noyau-extension, nombre d'éditeur open source on trouvé leur part de marché dans la vente des extensions. Bien que réalisées généralement par la communauté, l'éditeur peut considérer la vente des extensions réalisées afin de rémunérer les contributeurs mais prélève un montant sur la vente d'extensions. De cette manière, l'éditeur de la solution open source Magento pour le e-commerce, prélève 30% du prix de vente des extensions à la manière de l'Apple Store.

2.4.3 Communication par ses atouts

En guise de communication, l'open source n'a pas tant besoin de budget marketing. En effet il puise sa force là où il a pris racine, c'est à dire dans sa communauté. Nul besoin de mettre de faux posts et avis sur tous les blogs du net, la vérité et la promotion de l'open source se fait principalement par la communauté.

Le caractère open source permet en général de diffuser bien plus rapidement son produit.

La communauté utilise donc tout support moderne de communication pour diffuser, twitter, poster ces informations

Ainsi malgré le marketing ordinaire (campagne publicitaires, affiches, buzz-marketing ...) que ne peuvent se permettre bon nombre d'éditeur, l'éditeur se doit d'**utiliser le marketing moderne 2.0**, qui est la force et le fondement de l'open source, pour prendre la relève.

Les éléments clés à retenir Afin de valoriser l'open source, nous pouvons utiliser différents leviers. Autour de la communication, l'importance d'avoir un support clair de présentation, de documentation et de communication à l'aide d'outils efficaces. Dans l'organisation et la structuration du projet, il est important de veiller à ce que les parties prenantes constructrices du projet s'épanouissent dans leur domaine afin d'apporter la meilleure pierre à l'édifice. Côté management dans l'open source, l'intérêt est de se rapprocher de la communauté, lui faire confiance pour décider ensemble de l'avenir du projet par des méthodes de management horizontale en se mettant au même niveau que le groupe. Auprès du consommateur il faut noter l'importance d'écoute du besoin et de l'adaptation du projet à celui-ci, c'est le consommateur qui sollicite ensuite les principales sources de revenus comme le support logiciel.

Chapitre 3

Etude terrain

- 3.1 Plateforme promotrice
- 3.2 Gestion des ressources
- 3.3 Chez le consommateur
- 3.4 Marketing de l'open source

Chapitre 4

Confrontation

Chapitre 5

Transposition

Chapitre 6

Conclusion

Appendices

Annexe A

Contexte de l'open source

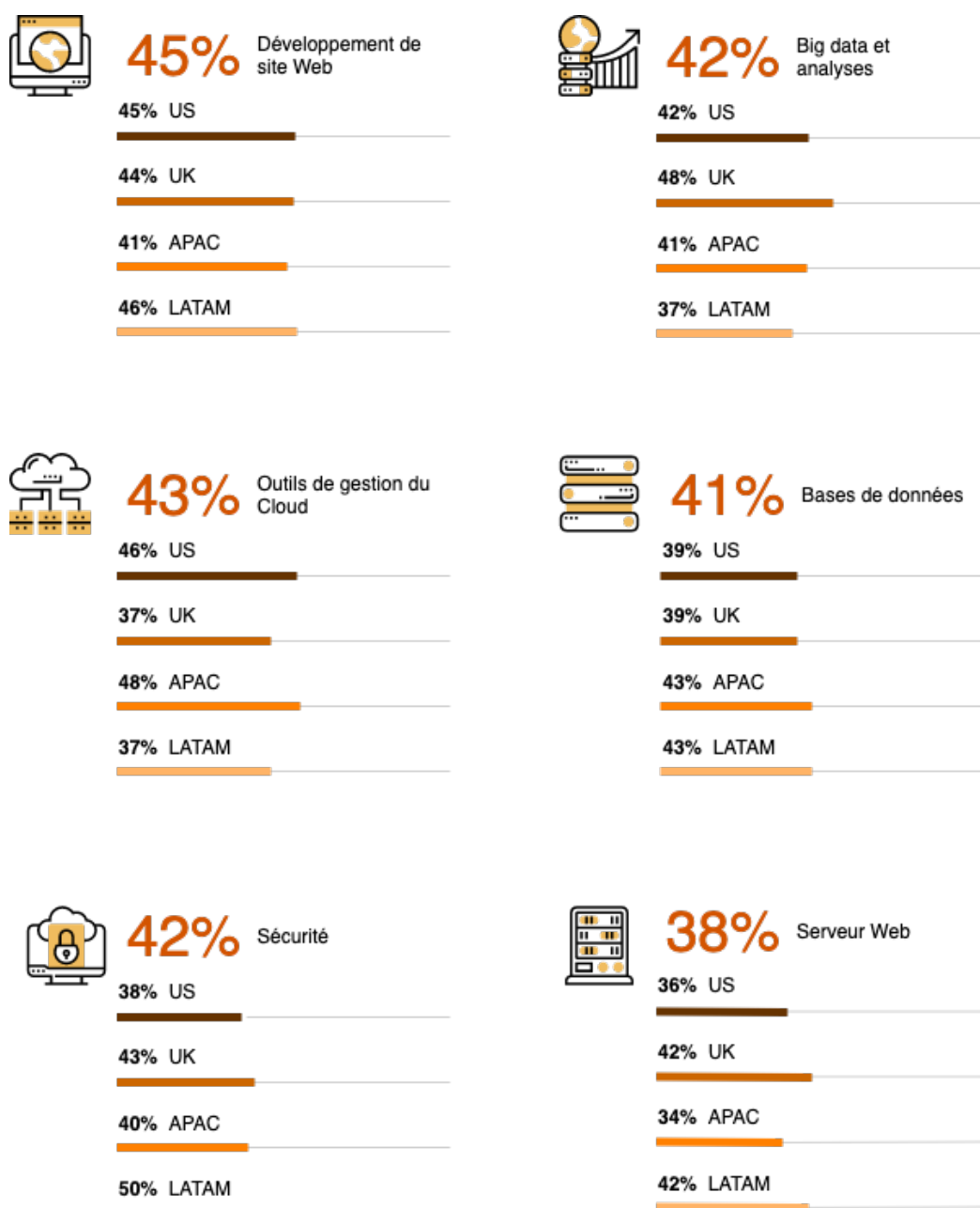


FIGURE A.1 – Domaine d'application de l'open source dans l'IT

Source : redhat.com

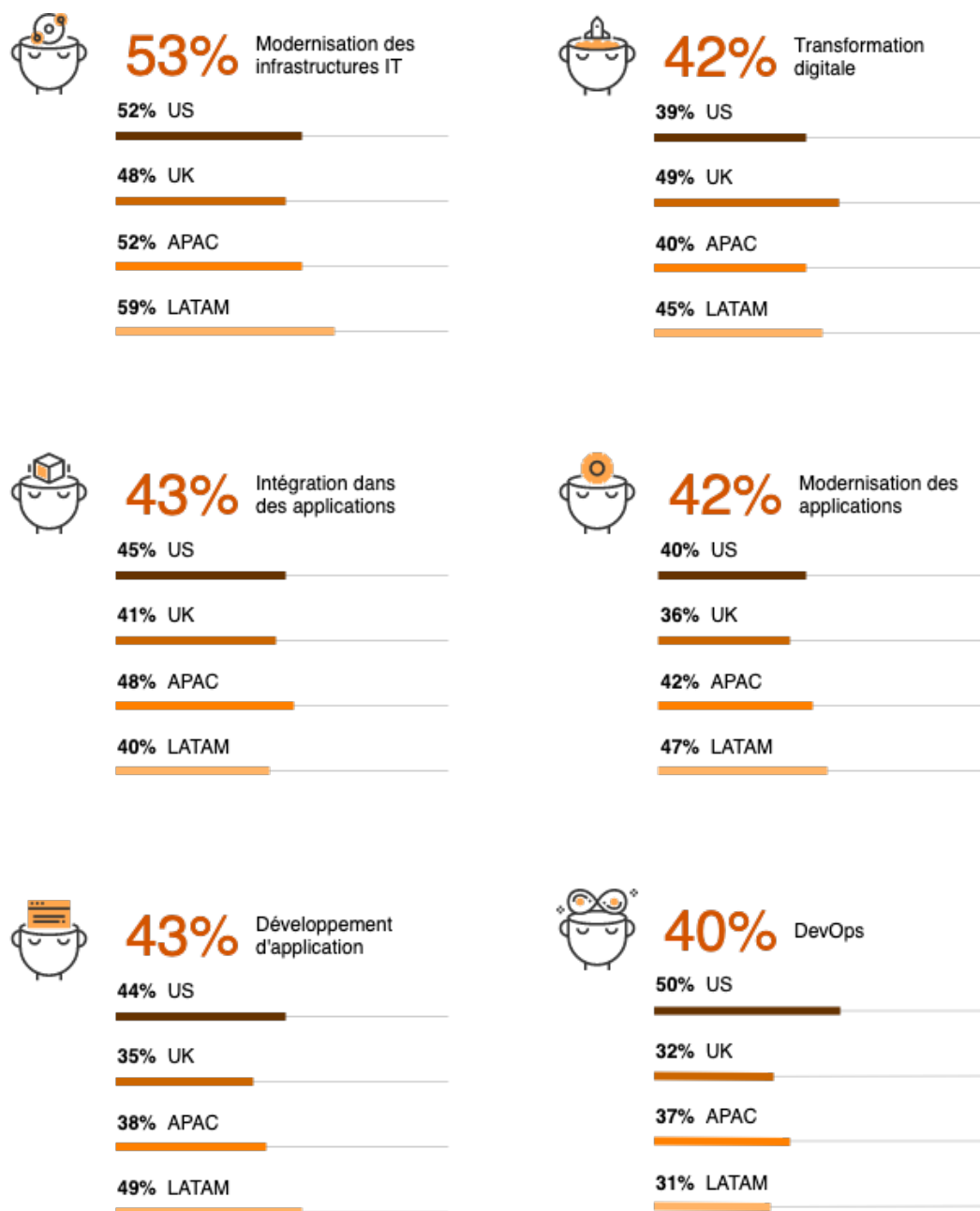


FIGURE A.2 – Secteur d'application de l'open source

Source : redhat.com