

Livre blanc



Évaluation et valorisation des actifs immatériels ouverts

Comment évaluer et valoriser les actifs immatériels au sein de projets ouverts (Open Source, Open Data) ?

Le livre blanc « Évaluation et valorisation des actifs immatériels ouverts » a été présenté à l'occasion de l'Inno'vent organisé au sein des locaux du cabinet inno³ le 22 juin 2017.

À l'instar de tous nos travaux, ce livre blanc est ouvert et librement modifiable. Vous êtes chaleureusement invités à contribuer et à rejoindre le groupe de travail qui s'est formé autour de ce sujet.

Une nouvelle restitution de nos travaux est prévue en septembre 2017 ainsi que lors du Paris Open Source Summit en décembre 2017.

Préface

À l'heure où les modèles ouverts occupent une place de plus en plus importante au sein de notre société, la question de l'évaluation des actifs immatériels, sous-jacents à l'activité des entreprises de demain, devient de plus en plus centrale.

Il ne s'agit plus, en effet, de penser en termes d'exploitation de monopole, ni même en termes de coût lié à la production d'une ressource, mais en termes de posture et de « capacité d'engagement » d'une communauté à l'égard de ces ressources, ouvertes et de plus en plus construites collectivement.

À la fois opportunité et source d'imprévus, les acteurs économiques se retrouvent confrontés à l'équilibre complexe qui consiste à identifier la pertinence de s'appuyer sur des ressources libres et ouvertes – voire de partager leurs propres ressources sur les bases de ce modèle – tout en étant capables de justifier d'une véritable capacité à tirer profit de la place et du rôle dont ils bénéficient au titre de leur investissement à l'égard de cette ressource. Très fréquemment résultat d'un travail intuitif, ce premier exercice s'avère souvent insuffisant lorsqu'il est question de convaincre un financeur – voire seulement un sponsor interne – des bénéfices réels attendus.

Cette problématique, rencontrée au sein du cabinet inno³, a été présentée lors de la journée « IP LegalTech » organisée à l'INPI le 23 mai 2017, avant de donner lieu à la création d'un groupe de travail réunissant juristes experts en propriété intellectuelle, économistes et chercheurs en management et en innovation en France et en Europe.

Tous se sont retrouvés autour de ce besoin de créer une méthodologie spécifique et adaptée aux modèles ouverts, qu'il s'agisse d'Open Data, d'Open Source, ou plus globalement d'innovation ouverte.

Ce livre blanc a vocation à présenter les premiers travaux effectués autour de ce sujet afin de les enrichir durant l'année et de rapidement confronter le modèle aux cas concrets de projets faisant des ressources ouvertes un axe central.

Table des matières

Préface.....	2
Introduction.....	4
1 La notion de valeur en matière de projets ouverts.....	5
1.1 Définition de la notion de « projet ouvert ».....	5
1.1.1 La notion d'Open Source.....	5
1.1.2 La notion d'Open Data.....	6
1.1.3 Le succès grandissant de ces nouveaux modes de diffusion.....	6
1.2 L'ouverture n'implique pas l'absence de valeur.....	7
2 Le besoin d'une méthodologie spécifique.....	9
2.1 Les méthodes d'évaluation des actifs immatériels.....	9
2.1.1 Les méthodes classiques d'évaluation des actifs.....	9
2.1.2 La méthode globale développée par le Thésaurus-Bercy.....	11
2.2 Les méthodes d'évaluation de la qualité des projets Open Source.....	12
2.2.1 La méthode Open Business Readiness Rating (OpenBRR).....	12
2.2.2 L'Open Source Maturity Model (OSMM).....	12
2.2.3 L'Open-source Software Capability Assessment Radar (OSCAR).....	13
2.2.4 La méthode Qualification and Selection of Open Source software (QSOS).....	13
3 Présentation de la méthodologie développée.....	15
3.1 Calcul de la valeur globale du projet.....	15
3.1.1 Le capital technique.....	15
3.1.2 Le capital humain.....	16
3.1.3 Le capital organisationnel.....	17
3.1.4 Le marché.....	18
3.2 Calcul de l'emprise de l'entreprise sur le projet.....	18
3.2.1 La production du code source.....	19
3.2.2 Les droits de propriété intellectuelle.....	19
3.2.3 La gouvernance.....	19
4 Perspectives.....	20
5 Licence.....	21
6 À propos d'Inno ³	21
7 À propos des Auteurs.....	21
Annexe 1 – Évaluation du projet.....	23
Annexe 2 – Emprise de l'organisation sur le projet.....	30

Introduction

Le principe essentiel des modèles ouverts réside dans l'idée d'un partage non limité aux seuls acteurs d'un projet. L'intérêt de la ressource ouverte finira par être tel que les contributeurs externes souhaiteront eux aussi devenir moteurs du projet. L'enjeu est donc particulièrement complexe puisque le développement de cette ressource ouverte repose sur la capacité des auteurs à « laisser de la place » aux contributeurs externes, sachant que « l'incentive » de ces derniers – la motivation qui justifie leur implication – est généralement liée à l'autorité et à la légitimité qu'ils peuvent en tirer.

Destinée à accompagner l'usage et la diffusion de ressources ouvertes – qu'il s'agisse de logiciels ou de données –, cette méthodologie apporte une meilleure lecture des enjeux clés d'un projet ouvert ou d'une ressource ouverte. L'objectif est de permettre aux acteurs publics et privés de mieux s'engager sur cette voie et de confirmer, autant que nécessaire, que le recours à ce type de modèle ne représente pas une perte de valeur pour elles. Les acteurs n'ayant pas encore franchi ce pas auront, quant à eux, un certain nombre d'éléments rationnels susceptibles de leur donner confiance dans ces modèles, encore trop méconnus.

Cette méthodologie a ainsi pour vocation à servir aussi bien :

- Aux entreprises ou organisations qui décident de diffuser des ressources ouvertes, c'est-à-dire d'ouvrir des ressources développées en interne, qu'à celles qui décident d'utiliser des ressources ouvertes, c'est-à-dire d'utiliser au sein de l'entreprise des ressources développées en externe ;
- Aux entreprises ou organisations qui fonctionnent déjà dans des logiques d'ouverture et ont recours à l'Open Data et à l'Open Source, qu'à celles qui n'ont pas encore franchi ce pas.

Tout l'objectif de ce travail est donc de réunir un ensemble hétérogène d'acteurs, petits et grands, en provenance du secteur public comme privé, et aux compétences diverses, pour s'accorder à la fois sur les éléments importants à évaluer et sur une première méthode d'évaluation conforme au modèle de l'Open Source et de l'Open Data. Il s'appuie sur les travaux déjà existants et s'en écarte pour correspondre aux spécificités du modèle ouvert, beaucoup plus horizontal et inclusif que les modèles classiques.

Ce livre blanc définit ainsi les principales notions attachées au modèle ouvert (1), détaille la méthodologie spécifique proposée (3) au regard des méthodologies existantes (2), et expose enfin les perspectives ouvertes par ces premiers travaux (4).

1 La notion de valeur en matière de projets ouverts

Que faut-il entendre derrière le terme d'« actifs immatériels ouverts » ou de « projets ouverts » ? Hybrides, ces notions renvoient à la fois vers l'idée d'« actifs » valorisables à ce titre, et de ressources ouvertes et partagées, donc beaucoup plus complexes à évaluer.

1.1 Définition de la notion de « projet ouvert »

La notion de « projet ouvert » se réfère ici à des projets Open Source ou Open Data, suivant qu'il s'agisse de logiciels ou de jeux de données¹.

Ces deux logiques font partie de la même tendance et ont en commun une forte adoption. Ainsi, il y a fort à parier que l'ouverture d'un nombre croissant de données continuera d'alimenter un fonds croissant de données de référence de plus en plus utilisé, de la même façon que plus aucun projet informatique n'est développé sans s'appuyer sur des composants Open Source préexistants.

1.1.1 La notion d'Open Source

L'Open Source, formalisé en 1998 dans le domaine du logiciel, désigne un modèle de développement et de diffusion de logiciels qui pose comme principe fort l'ouverture et la non-discrimination. Concrètement, l'adoption de ce modèle consiste à co-développer un logiciel dont le code source est mis à disposition du grand public sous une licence dite « Open Source » qui organise un système de partage des droits patrimoniaux permettant à quiconque d'utiliser, modifier et redistribuer le logiciel.

Loin d'empêcher la commercialisation des produits ainsi dérivés, l'Open Source s'appuie sur le marché pour diffuser plus largement encore le projet et donner aux acteurs économiques un vrai retour économique valorisant leur propre valeur ajoutée – qu'il s'agisse de développements spécifiques, de services complémentaires, assurances, etc.

Cette notion rejoint la définition du logiciel libre qui qualifie de libre tout logiciel qui offre quatre libertés à ses utilisateurs² :

- La liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins (ce qui implique l'accès au code source) ;
- La liberté de redistribuer des copies du logiciel ;
- La liberté d'améliorer le programme et de publier ces améliorations (ce qui implique, là encore, l'accès au code source) ;
- La liberté d'exécuter le programme pour tous les usages.

1 Elle pourrait potentiellement être ouverte à toute autre application de ce modèle tel que l'Open Hardware ou l'Open Content.

2 Il s'agit de la « Free Software Definition » rédigée en 1986 par la Free Software Foundation.

1.1.2 La notion d'Open Data

Tel un cousin éloigné, l'Open Source a inspiré le mouvement de l'Open Data en matière de données. L'Open Data désigne la dynamique qui tend à favoriser l'accès aux jeux de données sous une licence permettant la libre reproduction, redistribution, modification et réutilisation (y compris à des fins commerciales)³.

Très souvent mise en avant pour les garanties qu'elle offre en termes de transparence et de démocratie, cette tendance est aussi vectrice d'une valeur qui profitera, à terme, à l'économie toute entière en permettant le développement de services innovants autour des données qui sont ainsi libérées⁴.

Les réformes législatives récentes – et en premier lieu la Loi pour une République Numérique – et celles à venir renforcent grandement ce mouvement en mettant à la charge de l'administration – vue de manière de plus en plus large – un nombre croissant d'obligations.

1.1.3 Le succès grandissant de ces nouveaux modes de diffusion

Ces nouveaux modèles de diffusion occupent une place croissante dans notre société, de plus en plus d'acteurs se tournant vers l'Open Source ou l'Open Data : dans le domaine informatique, mais également des secteurs stratégiques tels que la santé ou l'énergie⁵.

Qu'il s'agisse de l'Open Source ou de l'Open Data, ces deux mouvements font partie de la même dynamique d'ouverture et s'alimentent mutuellement. Plus que de technologies, il est question de modèles. L'Open Source, comme l'Open Data, impliquent de nouvelles logiques de création et pérennisation des actifs immatériels : tendance collaborative par un ensemble d'acteurs, et non plus par une seule organisation.

Ils s'inscrivent tous deux dans un schéma plus global de circulation des actifs immatériels et de « coopération » dans l'esprit de l'Open Innovation (ou innovation ouverte en français) théorisée par Henry Chesbrough en 2003. Poussant à l'extrême les logiques d'« ouverture des frontières des organisations », l'innovation ouverte permet de bénéficier des apports de contributeurs externes et internes. L'innovation ne se réduit désormais plus à une seule organisation : il s'agit de « *créer mieux avec et grâce aux autres* »⁶.

Même si les investisseurs outre-atlantique comprennent de mieux en mieux ces logiques de communautés, et sont prêts à financer des projets sur la seule idée d'une valorisation à venir au travers du développement d'une communauté d'utilisateurs, ce modèle repose souvent sur un nombre important d'aléas. On peut ici citer l'exemple de Twitter, qui a réussi à lever 200 millions de dollars en 2010 et n'a aujourd'hui toujours pas atteint sa rentabilité. Les modèles ouverts ajoutent à cela que, certains des actifs du projet étant coconstruits, il ne sera pas possible d'en faire une exploitation commerciale directe.

3 Se référer à l'Open Definition : <http://opendefinition.org/>

4 Voir : Livre blanc « *L'ouverture des données publiques : nouvelles obligations et nouveaux acteurs* », Benjamin Jean et Laure Kassem, juin 2017

5 Voir : Guide « *Empowering Open Innovation, du rôle et de la place des modèles ouverts* », sous la dir. de Benjamin Jean, janvier 2017

6 *Ibid*

Autant d'éléments qui renforcent l'intérêt d'expliquer ce choix, voire de le rationaliser, diffusant la connaissance du modèle dans un premier temps, mais également en proposant une méthode d'évaluation qui lui soit propre.

1.2 L'ouverture n'implique pas l'absence de valeur

Une compréhension trop superficielle des projets ouverts conduirait à penser que, du fait de l'ouverture des données ou des logiciels, ceux-ci seraient exempts de toute valeur. L'histoire dément heureusement cette idée et il suffit, pour s'en convaincre, de jeter un œil au prix auquel ont été valorisées de nombreuses entreprises développant des logiciels en Open Source ces dernières années :

Entreprise acquise	Acquéreur	Montant (en millions de dollars)	Date
Jboss	Redhat	420	2006
Zimbra	Yahoo	350	2007
MySQL	Sun	1,000	2008
Sun	Oracle	7,400	2009
SpringSource	VMWare	420	2009
Zimbra	VMWare	100	2010
SourceFire	Cisco	2,700	2013
Enovance	RedHat	70	2014
Zimbra ⁷	Synacor	24,5	2015
Ansible	RedHat	150	2015

Ces premières métriques sont autant d'éléments sur lesquels confronter un modèle d'évaluation spécifique à l'évaluation financière de ces entreprises ou organisations diffusant ou utilisant des ressources ouvertes.

La difficulté de l'exercice tient au fait que cette problématique est à la fois très proche et très éloignée de ce que l'on connaît classiquement : les actifs immatériels sont « juridiquement » les mêmes, mais les leviers de valorisation diffèrent. Ainsi, lorsqu'une ressource est diffusée sous une licence libre ou ouverte, cela n'implique pas qu'elle soit dépourvue de toute protection par la propriété intellectuelle et qu'il n'y ait plus d'opportunités économiques rattachées à cette ressource. L'idée est, tout au contraire, que l'exploitation sera différente, et donc peut être plus disruptive. Au lancement du projet Android en 2008, l'idée était claire : lancer un produit ouvert afin que les utilisateurs puissent se l'approprier, le modifier et l'améliorer, pour pénétrer un marché peu accessible⁸.

7 Il ne s'agit ici que de la partie messagerie, la partie collaboration ayant été vendue à Verint. Cela explique pourquoi la valorisation était alors moins élevée que lors de ses précédents rachats.

8 Voir : Guide « *Empowering Open Innovation, du rôle et de la place des modèles ouverts* », sous la dir. de Benjamin Jean, janvier 2017

Nous ne sommes donc plus dans une exploitation classique du monopole conféré par la propriété intellectuelle, mais dans une idée de partager les ressources le plus possible pour maximiser leur diffusion, afin de soi-même en tirer le meilleur. Les entreprises ou organisations qui diffusent ou utilisent des ressources ouvertes se situent donc plus dans une logique d'horizontalité que de verticalité, à l'inverse des projets classiques.

Comment alors, au vu des particularités que présentent les projets Open Source ou Open Data, réussir à évaluer financièrement les entreprises ou organisations utilisant ou diffusant des ressources ouvertes ?

2 Le besoin d'une méthodologie spécifique

Des méthodes spécifiquement conçues pour évaluer les actifs immatériels sont aujourd'hui développées pour évaluer des projets « classiques ». À l'inverse, certaines méthodes spécifiquement développées pour évaluer la qualité des projets Open Source permettent de pointer les éléments forts d'un projet ouvert.

Pourtant, aucune de ces méthodes ne permet d'aboutir à une évaluation financière des entreprises ou organisations utilisant ou diffusant des ressources ouvertes.

2.1 Les méthodes d'évaluation des actifs immatériels

Il n'existe pas une méthode unique d'évaluation des actifs immatériels. Plusieurs méthodes ont été développées, parfois complémentaires, parfois alternatives. Aux méthodes classiques d'évaluation s'intéressant à chaque actif de manière isolée (1) sont venues se greffer des méthodes plus globales, comme la méthode dite Thésaurus-Bercy (2).

2.1.1 Les méthodes classiques d'évaluation des actifs

Nous avons identifié quatre catégories de méthodes :

- **La méthode du coût historique** : elle repose sur l'idée que l'actif vaut ce qu'il a coûté. Pour calculer la valeur de l'actif, on va donc recenser toutes les dépenses qui ont été mises en œuvre pour sa création : main d'œuvre, frais de recherche et développement, frais lié au matériel, frais de marketing ou de communication...

Si cette méthode peut séduire par son apparente facilité à être mise en œuvre, elle est peu utilisée. Tout d'abord, car il n'est pas toujours simple de la mettre en œuvre, les entreprises ne gardant pas systématiquement une information complète des moyens ayant été engagés pour développer un actif donné. Ensuite – et c'est la raison principale – la somme des coûts engagés pour constituer un actif équivaut rarement à la valeur actuelle de cet actif.

- **La méthode du coût de renouvellement** : elle repose sur l'idée que l'actif vaut ce qu'il coûterait de le reconstituer dans son état actuel. La valeur de l'actif serait donc équivalente soit à la somme des dépenses qu'il faudrait engager pour acquérir un actif similaire sur le marché, soit à la somme des dépenses nécessaires pour développer un actif similaire en interne.

Cette méthode est cependant c – voire impossible – à mettre en œuvre pour certains actifs qui ne peuvent pas être facilement remplacés, comme certains brevets ou marques très connues.

- **La méthode transactionnelle** (ou « approche par les comparables ») : elle consiste à déterminer la valeur d'un actif en étudiant les transactions concernant les actifs du même type sur le marché. On suppose ici que la valeur de l'actif envisagé est équivalente au prix où un actif similaire s'échange sur le marché.

Cette méthode est considérée comme l'une des plus fiables, mais sa mise en œuvre est difficile. Il est, en effet, souvent complexe de trouver les données adéquates : il faut que l'actif en question se trouve sur un marché très dynamique. De plus, l'actif immatériel envisagé peut être trop particulier pour qu'il soit possible de trouver un actif équivalent sur le marché, et cette méthode est alors inapplicable.

- **La méthode par les revenus futurs nets actualisés** (ou « approche par les flux de trésorerie ») : elle consiste à évaluer la valeur de l'actif en fonction des revenus futurs qu'il va générer.

Si cette méthode est préconisée par un certain nombre de cabinets spécialisés, elle est toutefois difficile à mettre en œuvre : comment déterminer avec certitude à quel actif les profits futurs de l'entreprise peuvent être attribués ?

Chacune de ces méthodes présente à la fois des avantages et des inconvénients, que nous avons regroupés dans ce tableau :

Méthode	Description	Avantages	Inconvénients
Méthode du coût historique	Reconstitution des coûts qui ont été nécessaires au développement (ou à l'acquisition) de l'actif.	Méthode facile à mettre en œuvre.	Ne prend pas en compte l'évolution du contexte. Ne tient compte que du passé et ne tient pas compte de la capacité de l'actif à générer des revenus. → Méthode peu fiable.
Méthode du coût de renouvellement	Calcul des coûts qui seraient nécessaires pour créer l'actif <i>ex nihilo</i> .	Méthode facile à mettre en œuvre.	Impossible à mettre en œuvre pour certains actifs ne pouvant pas être facilement remplacés.
Méthode transactionnelle	Valeur de l'actif calculée à partir de l'observation de transactions récentes dans le même secteur.	Méthode considérée comme l'une des plus fiables.	Difficulté pour trouver des données fiables et adéquates : peu de références sur certains marchés spécifiques.
Méthode par les revenus nets actualisés	Consiste à évaluer la valeur de l'actif en fonction des revenus futurs qu'il va générer.	Méthode préconisée par un grand nombre de cabinets spécialisés.	Difficile de déterminer à quel actif les revenus sont dus.

Toutes ces méthodes ont, en outre, un inconvénient commun : en étudiant la valeur de chaque actif envisagé de manière isolée, elles ne permettent pas d'apprécier le « goodwill ».

Le goodwill vise la différence entre la valeur économique de l'entreprise (c'est-à-dire la valeur de son actif et de son passif) et sa valeur sur le marché. C'est pour cela qu'on parle également d'écart d'acquisition ou de survaleur. Le goodwill est positif lorsque la valeur sur le marché de l'entreprise est plus importante que sa valeur économique, mais il peut également être négatif : on parle alors de badwill.

2.1.2 La méthode globale développée par le Thésaurus-Bercy

La méthodologie Thésaurus-Bercy, initiée par Alan Fustec au sein du cabinet Goodwill Management puis développée par l'Observatoire de l'Immatériel et le ministère de l'économie, a pour ambition d'évaluer le goodwill des entreprises.

Elle a fait l'objet d'une publication en deux volets : le volet 1 a été publié le 7 octobre 2011 et le volet 2 le 13 octobre 2015.

Le volet 1 est consacré à l'évaluation extra-financière des actifs immatériels de l'entreprise. La valeur de chaque actif est modélisée comme une fonction de plusieurs critères, chacun de ces critères étant ensuite doté d'une note allant de 0 à 20. Cette évaluation, bien qu'étant limitée à une dimension extra-financière, permettra néanmoins de pondérer l'évaluation financière qui sera effectuée de l'entreprise⁹.

Le volet 2 a pour objectif de compléter cette méthodologie d'évaluation par de bonnes pratiques de valorisation (évaluation financière) du capital immatériel de l'entreprise.

La méthodologie globale proposée consiste donc à :

- Calculer le coût de remplacement brut de l'actif selon la méthode classique du coût de renouvellement : calcul des coûts qui seraient nécessaires pour recréer l'actif *ex nihilo*.
- Appliquer deux coefficients à cette valeur brute :
 - Un coefficient de pérennité, qui a pour but de tenir compte du temps restant pendant lequel l'actif pourra être utilisé en l'état.
 - Un coefficient d'état, déterminé grâce à l'évaluation extra-financière de l'actif détaillée dans le volet 1.

La valeur de remplacement nette d'un actif est donc égale à la valeur de remplacement brute de l'actif, pondérée par le coefficient de pérennité et le coefficient d'état.

La valeur globale de l'entreprise est ensuite égale à la somme des valeurs des actifs qui la composent. La valeur de chaque actif peut également faire l'objet d'une pondération. Le volet 2 de la méthodologie Thésaurus-Bercy formule l'hypothèse selon laquelle le poids de chaque actif dans la valeur globale de l'entreprise n'est pas le même en fonction du secteur considéré et de la taille de l'entreprise.

La méthodologie Thésaurus-Bercy, si elle permet de prendre en compte le goodwill de l'entreprise grâce aux différentes pondérations, n'est toutefois pas utilisable en l'état pour évaluer des projets Open Source ou Open Data, et ce pour deux raisons. La première tient au fait que cette méthode a été créée pour évaluer des entreprises, et non pas des projets. La seconde est liée au caractère ouvert des actifs utilisés ou développés par les projets ouverts étudiés.

9 La méthodologie Thésaurus-Bercy volet 1 fait cette analogie : « *C'est une question de bon sens : il ne viendrait pas à l'idée à l'acquéreur potentiel d'un appartement d'acheter celui-ci au prix moyen du m² d'un quartier (méthode des comparables) sans procéder en premier lieu à une évaluation extra-financière : la rue est-elle bruyante ? L'appartement est-il bien insonorisé ? Est-il bien isolé thermiquement ? L'ascenseur est-il assez spacieux pour monter des objets encombrants jusqu'à l'étage ? La vue depuis la fenêtre du salon est-elle dégagée ? ... L'établissement de la valeur du bien dépendra de tous ces éléments.* »

En définitive, ni les méthodes classiques d'évaluation des actifs immatériels, ni la méthodologie Thésaurus-Bercy ne semblent adaptées à l'évaluation des projets ouverts.

2.2 Les méthodes d'évaluation de la qualité des projets Open Source

Plusieurs méthodes ont été développées pour fournir des grilles d'évaluation des logiciels libres. Nous faisons ici le choix de présenter celles qui semblent être les plus abouties.

2.2.1 La méthode Open Business Readiness Rating (OpenBRR)

Développée en 2005, la méthode OpenBRR s'intéresse aux projets formés autour d'une communauté pour fournir une évaluation de leur capacité à être déployés au sein d'un environnement commercial.

La note finale, allant de 1 (inacceptable) à 5 (excellent), résulte de l'étude de douze catégories : le caractère fonctionnel du logiciel, son ergonomie, sa qualité, sa sécurité, ses performances, son évolutivité, son architecture, son support, la documentation qui l'accompagne, son adoption, sa communauté, son professionnalisme.

Pour chaque catégorie, différents critères sont évalués pour aboutir à une note entre 1 et 5 pour chacune de ces catégories. Ensuite, suivant l'usage final du logiciel, il est possible de pondérer chaque catégorie pour obtenir la note finale du projet.

Bien que faisant partie des projets d'évaluation des projets Open Source les plus aboutis, il semble que le projet OpenBRR n'ait pas trouvé son public. Le projet semble être aujourd'hui à l'abandon : le site internet est inactif et le livre blanc n'est plus téléchargeable. Par ailleurs, le projet OpenBRR visait non pas à évaluer la valeur du projet mais à donner un indice de confiance dans un projet.

2.2.2 L'Open Source Maturity Model (OSMM)

D'abord développée par Capgemini en 2003 puis reprise par Navica en 2004, l'OSMM est une méthodologie destinée à évaluer la maturité des projets Open Source.

Comment la méthodologie OpenBRR, elle se base sur un certain nombre d'indicateurs, classés en quatre catégories :

- **Le produit** : il s'agit ici de s'intéresser aux caractéristiques intrinsèques du produit, à travers plusieurs indicateurs (l'âge du projet, la licence sous laquelle il est distribué, la hiérarchie de la gouvernance, la communauté des développeurs...) ;
- **L'intégration** : cette catégorie évalue la possibilité de lier le produit avec d'autres produits ou infrastructures, en se fondant sur la modularité du produit et les éventuelles collaborations avec d'autres produits.
- **L'utilisation** : ce critère est évalué à travers deux indicateurs, la création d'un standard et le support aux utilisateurs.

- **L'adoption par la communauté** : il s'agit ici de s'intéresser à la manière dont est reçu le produit par la communauté. Cette catégorie est évaluée à travers : la facilité d'installer le logiciel, la communauté des utilisateurs et la pénétration du marché.

Le projet OSMM, bien qu'étant probablement le projet le plus connu, n'est pas le plus abouti : le nombre d'indicateurs utilisés reste relativement faible. Par ailleurs, comme le projet OpenBRR, le projet OSMM n'a pas vocation à évaluer la valeur d'un projet Open Source, mais sa maturité.

2.2.3 L'Open-source Software Capability Assessment Radar (OSCAR)

Le projet OSCAR a été lancé par OW2, un consortium dédié au développement d'une base de logiciels d'infrastructure en Open Source. Le projet, destiné à améliorer la qualité des logiciels développés en Open Source, est organisé autour de deux piliers : la gouvernance et la technique.

Dans le cadre du projet OSCAR, OW2 propose un formulaire basé sur l'Open Source Maturity Model. OW2 va plus loin que le projet OSMM lancé par Capgemini en 2003 : les catégories étudiées sont plus nombreuses et plus détaillées, avec plusieurs questions à poser pour chacune d'entre elles.

Les catégories étudiées sont les suivantes : la documentation du projet, l'utilisation de standards, la qualité des processus de tests, la gestion des droits de propriété intellectuelle (licences, droits d'auteur), l'environnement, la maintenance des bugs, commits et autres contributions, la maintenance des propriétés non-fonctionnelles, la configuration, le déploiement sur le cloud, les activités projetées, la gestion des pré-requis, la roadmap du projet, et enfin les porteurs du projet – c'est-à-dire la gouvernance.

Le formulaire développé par OW2, bien qu'étant plus détaillé que le projet OSMM initial, poursuit néanmoins le même but : il s'agit plus d'évaluer la qualité et la maturité d'un projet que sa valeur intrinsèque.

2.2.4 La méthode Qualification and Selection of Open Source software (QSOS)

La méthodologie QSOS, développée par ATOS en 2004, décrit le processus permettant d'évaluer des projets Open Source. Le processus comporte quatre étapes :

- La définition des données de référentiel ;
- L'évaluation en tant que telle, selon trois axes :
 - La couverture fonctionnelle ;
 - Les risques du point de vue de l'entreprise utilisatrice ;
 - Les risques du point de vue du fournisseur de services (expertises, formation, support).
- La qualification du contexte spécifique de l'entreprise (ou de l'utilisateur) en effectuant une pondération des critères précédents ;
- La sélection et la comparaison des logiciels répondant aux besoins.

Ce processus génère des fiches d'évaluation de logiciels ainsi que des grilles de comparaison et de choix.

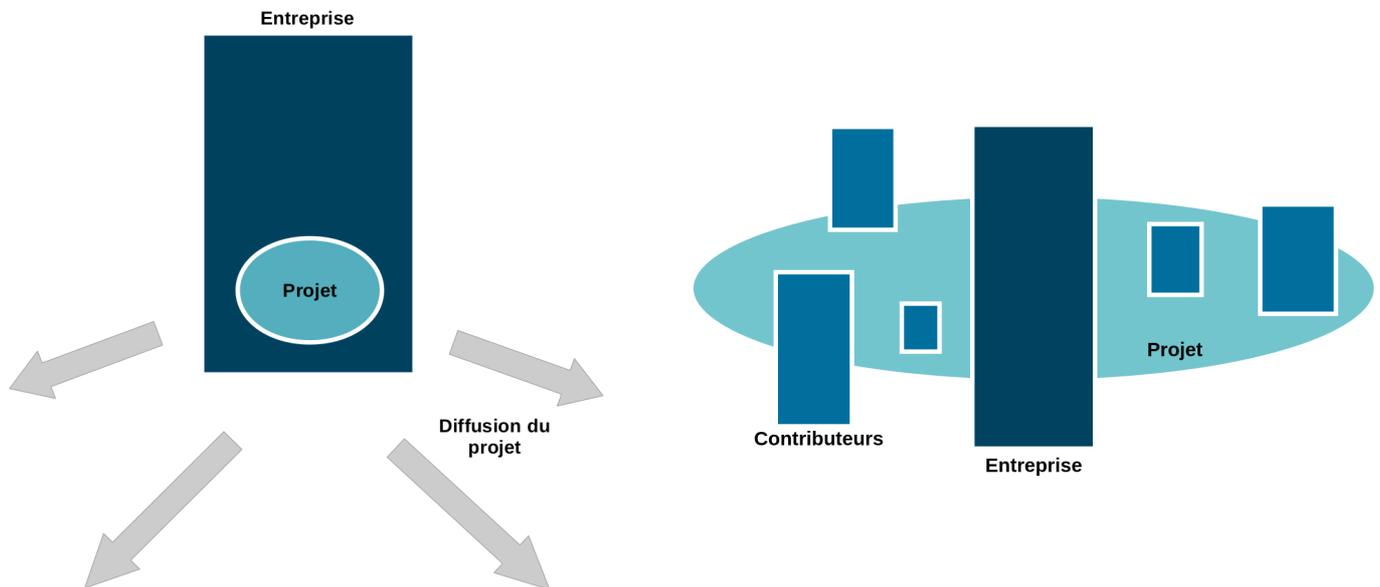
Le projet QSOS, bien que prometteur, ne semble plus développé aujourd'hui. Par ailleurs, le projet QSOS a vocation à sélectionner le projet Open Source le plus adapté aux besoins d'une entreprise ou d'un utilisateur, et pas d'évaluer la valeur financière d'un projet.

En définitive, aucune des méthodes d'évaluation des projets Open Source ne s'intéresse à l'évaluation de leur valeur financière.

Par conséquent, ni les méthodes d'évaluation des actifs immatériels, ni les méthodes d'évaluation des projets Open Source ne permettent de répondre à notre problématique, à savoir l'évaluation financière des projets ouverts.

3 Présentation de la méthodologie développée

Le constat sur lequel repose la méthodologie est le suivant : le projet possède une valeur globale susceptible d'être « partagée » entre tous les acteurs y participant.



On passe ainsi d'un schéma classique dans lequel le projet est développé en interne, au sein de l'entreprise, et est ensuite diffusé vers l'extérieur, à un schéma où le projet est développé de manière collaborative à la fois par l'entreprise mais également par tous les contributeurs externes qui vont se greffer au projet.

Il nous revient ainsi de réfléchir aux critères permettant dans un premier temps de calculer la valeur globale du projet pour, dans un second temps, déterminer l'« emprise » (aussi dite « capacité d'engagement ») de l'organisation sur ledit projet.

3.1 Calcul de la valeur globale du projet

L'étude des projets Open Source nous a poussés à diviser chaque projet en quatre axes : le capital technique (1) ; le capital humain (2) ; le capital organisationnel (3) et enfin le marché sur lequel se situe le projet (4).

3.1.1 Le capital technique

Il s'agit ici d'analyser le projet sous son angle technique. Dans le cas d'un projet Open Source, il s'agira donc de s'intéresser au logiciel qui est développé, c'est-à-dire à son code source et à la documentation qui l'accompagne :

- S'agissant de l'étude du code source, différents critères sont à étudier :
 - L'ampleur du logiciel développé, c'est-à-dire la **quantité de code** produit par les développeurs. Nous sommes conscients que le nombre de lignes de code ne

représente en rien l'investissement des développeurs dans un projet, et qu'un bon développeur se reconnaîtra à sa capacité à développer une fonctionnalité en un nombre très restreint de lignes, cependant il s'agit d'une métrique simple à analyser.

- La technicité du logiciel, mesurée à travers la **technicité du langage de développement utilisé**. Il s'agit ici d'évaluer le caractère courant et accessible du langage utilisé pour développer le logiciel. Certains langages de programmation sont, en effet, bien plus courants que d'autres. Si le caractère courant du langage de programmation peut être un atout pour agréger une communauté autour du logiciel, cela rend cependant le logiciel plus facile à s'approprier, et donc à « *forker*¹⁰ ».
- La **qualité juridique**, c'est-à-dire le respect des licences des composants utilisés.
- **L'accessibilité** aux développeurs : il s'agit ici de se demander si des dispositions ont été prises pour faciliter aux développeurs l'accès à l'écriture du code (*easyhacks*, système de mentors...).
- La **qualité de la rédaction** du code : le code est-il rédigé de manière claire, permettant les modifications ? Il serait envisageable de se fonder sur l'utilisation d'un logiciel comme SonarQube¹¹, bien que la pertinence des résultats obtenus fasse parfois l'objet
- S'agissant de l'étude de la documentation, les critères sont :
 - La **qualité** de la documentation, celle-ci étant considérée comme bonne lorsque les détails d'architecture sont bien documentés et que la documentation des instructions permettant aux développeurs de contribuer.
 - **L'accessibilité** de la documentation, celle-ci étant considérée comme bien accessible lorsqu'elle comporte une FAQ, un fichier README et qu'elle est rédigée en anglais.

3.1.2 Le capital humain

Il s'agit ici d'analyser le projet sous son angle humain, c'est-à-dire sous celui de la communauté que le projet a agrégée autour de lui. Une communauté peut être définie comme un regroupement, majoritairement informel, de personnes partageant un objectif commun¹². Dans le cas des projets Open Source, cet objectif commun est généralement le développement d'un logiciel.

Au sein de la communauté, on distingue :

- **Les utilisateurs** : dans le cadre d'un projet Open Source, les utilisateurs sont ceux qui vont télécharger et utiliser le logiciel développé. Ce sont ceux qui font vivre le produit. Il s'agira de répondre aux questions suivantes :

10 La notion de « fork » renvoie à la possibilité, offerte par la licence, d'initier une version dérivée du projet en parallèle du projet lui-même. Souvent décriée compte tenu de la division des forces qui suit généralement les forks, elles jouent souvent la fonction de « soupape » de sécurité à l'encontre de gouvernance trop strictes et peu participatives. Des exemples de forks réputés peuvent être cités tels que MariaDB (fork de MySQL) ou encore Cyanogen (fork d'Android).

11 SonarQube est un logiciel libre permettant de mesurer la qualité du code source en continu. Le logiciel SonarQube est distribué sous licence GNU GPL version 3. Lien vers le site officiel : <https://www.sonarqube.org/>

12 François Pellegrini & Sébastien Canevet, « Droit des logiciels », Presses Universitaires de France, 2013

- Combien d'utilisateurs le projet compte-t-il aujourd'hui ? Quelle est l'évolution attendue : sont-ils de plus en plus nombreux ?
- Sont-ils fidèles au projet ? Existe-t-il des coûts de changement (*switching costs*) qui augmenteraient, de fait, leur fidélité ?
- Sont-ils actifs ? Sont-ils inscrits aux *mailing lists* associées au projet et celles-ci sont-elles actives ?
- **Les contributeurs** : ce sont ceux qui vont participer concrètement au développement du projet en y apportant leur contribution. Par ce terme, on vise en premier lieu les développeurs, mais également les contributeurs à l'assurance qualité, à la documentation ou à la localisation¹³ du logiciel. L'étude s'attache donc ici à répondre aux questions suivantes :
 - Combien d'utilisateurs le projet compte-t-il aujourd'hui ? Quelle est l'évolution attendue : sont-ils de plus en plus nombreux ?
 - Quel est leur niveau d'expertise ? Observe-t-on une grande diversité dans leurs profils (développeurs, assurance qualité, traducteurs...) ? Sont-ils tous issus d'une même entreprise ?
 - Sont-ils fidèles au projet ?
 - Combien de personnes représentent la moitié du total des *commits* ? Cette question a pour but de mesurer le *bus factor*, c'est-à-dire le nombre de personnes en-deçà duquel le projet ne peut pas se poursuivre de manière viable.
 - Les contributeurs sont-ils fidèles au projet ?

3.1.3 Le capital organisationnel

Le capital organisationnel s'intéresse à la structuration qui a été mise en place au sein du projet et à son efficacité. Il s'agira ainsi d'étudier :

- **La structuration de la communauté**, et notamment sa hiérarchie, son mode de gouvernance, son mode de financement, sa transparence, le suivi qui est fait de la communauté, l'efficacité de sa gestion, son accessibilité, son dynamisme – tant en matière de développement que s'agissant des équipes – et sa gestion des outils de communication.
- **La structuration technique du projet**, au sein de laquelle on peut distinguer entre :
 - Le processus de production du code source, comprenant la résolution des bugs, la réalisation de tests, la documentation des modifications apportées au code source, l'utilisation d'outils d'intégration continue et le respect des conventions de chaque langage de programmation.

13 Dans le contexte d'un logiciel, le terme de « localisation » vise la traduction de l'interface utilisateur et de l'aide, mais également son adaptation au public visé : adoption des monnaies locales, utilisation de formats spécifiques pour les dates, adresses ou numéros de téléphone, etc.

- Le processus de production de la documentation, comprenant notamment l'utilisation d'outils spécifiques tels que Doxygen ou sa mise à jour régulière.
- **Le degré d'ouverture du projet** : celui-ci s'apprécie à travers le dépôt du code source sur des plateformes comme GitHub ou Gerrit, mais également à travers la licence sous laquelle est diffusé le projet ou encore la fréquence des *releases*.

3.1.4 Le marché

À la différence des parties précédentes, cette partie ne s'intéresse pas aux caractéristiques intrinsèques du projet mais à l'environnement dans lequel il se situe. L'étude du marché sur lequel se trouve le projet passe par l'étude :

- De la **concurrence** présente sur ce marché, à la fois directe et indirecte. Il s'agit également de se demander si le logiciel développé est remplaçable ou non par d'autres logiciels sur le marché.
- De la **réputation** du projet et de la marque sous laquelle il est diffusé.
- Du nombre et du type de **clients** du projet : les clients sont-ils plutôt des entreprises ou des particuliers ?
- De la **stratégie industrielle** qu'il est possible de mettre en place autour du projet. Cela dépend :
 - Du niveau de généralité de la solution : s'agit-il d'une application métier ou d'une plateforme générique ? Dans le premier cas, le niveau de généralité de la solution est moins grand.
 - Du niveau de servicisation de la solution : quel est le niveau d'adaptation du logiciel nécessaire à son utilisation ? Plus le niveau de customisation exigé est important, plus il sera possible de mettre en place un *business model* basé autour de ce service.
 - De la criticité de la solution : le dysfonctionnement du logiciel aurait-il un impact important sur ses utilisateurs ? Si la réponse est oui, il est possible de mettre en place un modèle basé sur l'assurance du logiciel développé.
 - Enfin, du caractère de « bien public industriel » que peut revêtir le logiciel : si le logiciel est perçu comme tel, il est probable que des acteurs seront prêts à investir dans le projet. C'est généralement le cas lorsqu'on s'intéresse à des systèmes d'exploitation.

3.2 Calcul de l'emprise de l'entreprise sur le projet

Une fois la valeur globale du projet déterminée, la question est celle de l'emprise de l'entreprise ou de l'organisation sur le projet : quelle est la valeur du projet qui est « attribuable » à celle-ci ?

Joel West et Siobhan O'Mahony ont étudié douze projets Open Source créés par des entreprises¹⁴ afin d'identifier trois leviers :

¹⁴ Joel West et Siobhan O'Mahony, « *The Role of Participation Architecture in Growing Sponsored Open Source Communities* », 2008

1. La production du code source ;
2. Les droits de propriété intellectuelle sur le code source ;
3. La gouvernance du code source.

Nous avons décidé de reprendre ces trois axes afin de déterminer l'emprise de l'entreprise ou de l'organisation sur le projet.

3.2.1 La production du code source

Ce levier est fonction de deux critères :

- L'**expertise** de l'organisation : l'entreprise possède-t-elle une forte expertise sur le logiciel développé par le projet au sein de ses équipes ?
- Les **moyens** que l'organisation alloue au projet, tant en termes de moyens humains (développeurs, mais également juristes, *brand managers*, open source evangelists...) que de moyens financiers ou autres.

3.2.2 Les droits de propriété intellectuelle

On peut diviser les droits de propriété intellectuelle en trois catégories :

- **Le droit des marques** : l'emprise que possède l'organisation sur le projet à travers sa marque va dépendre à la fois de la force du titre (c'est-à-dire de son étendue géographique et de sa notoriété) et du contrôle qu'elle exerce sur ce titre (c'est-à-dire la sévérité de sa politique de marque).
- **Le droit d'auteur** : le premier point à examiner ici est la titularité du droit d'auteur. L'organisation possède-t-elle effectivement les droits d'auteur sur le code source et la documentation qui sont produits au sein du projet ? Si tel est le cas, il faut ensuite analyser le degré de contrôle que l'organisation exerce sur ce titre.
- **Le droit des brevets** : l'emprise que possède l'organisation sur le projet à travers ses brevets va dépendre à la fois de la force du titre (c'est-à-dire de son étendue géographique et de son potentiel d'innovation) et du contrôle qu'elle exerce sur ce titre (c'est-à-dire la sévérité de sa politique en matière de brevets).

3.2.3 La gouvernance

Étudier la gouvernance revient à se pencher sur :

- Le **contrôle de la roadmap** : de quelle façon sont prises les décisions à court et à long terme ?
- La **hiérarchie** : l'organisation contrôle-t-elle les responsables au sein du projet ?
- Le **contrôle du marché** : l'entreprise possède-t-elle une forte notoriété dans son domaine ainsi qu'une grande légitimité ?

4 Perspectives

Ce livre blanc n'est qu'une présentation de l'état de nos travaux à l'heure actuelle, il reste encore beaucoup à faire, et notamment :

- En premier lieu, à adapter cette méthode à l'Open Data : la grille d'évaluation proposée s'intéresse pour l'heure uniquement aux projets Open Source, c'est-à-dire diffusant des logiciels sous licence libre. Il sera intéressant de voir quels sont les critères à adapter pour que la grille puisse être transposée aux données ouvertes, puis éventuellement à d'autres types de ressources ouvertes.
- À établir le barème, c'est-à-dire le poids que l'on accorde à chaque critère et l'agencement des arbres (déterminer, pour chaque critère, s'il est évalué par l'attribution d'une note /20 ou d'un multiplicateur).
- À vérifier la pertinence du modèle en l'appliquant à des valorisations d'entreprises qui ont été réalisées. Ce travail permettra à la fois de vérifier la viabilité du projet et de l'ajuster au besoin.
- Il sera également intéressant d'étudier un possible rapprochement avec des outils générant des mesures automatiques de certaines métriques des projets Open Source, comme Bitergia par exemple.

Enfin, l'idée de ce travail est de collaborer avec tous les acteurs concernés (professions du chiffre, experts en propriété intellectuelle, porteurs de projets ouverts...) pour faire une vraie place à ce modèle dans l'évaluation des actifs immatériels ouverts.

5 Licence

Ce livre blanc a été écrit par Benjamin Jean et Ségolène Delmas pour la société Inno³ sous licence Creative Common By SA 4.0. Pour obtenir des informations complémentaires sur cette licence, veuillez consulter le site : <http://creativecommons.fr/>

6 À propos d'Inno³



Inno³ est un cabinet français indépendant accompagnant les organisations souhaitant tirer pleinement profit de l'Open Source/ au sein de leur IT et de leur R&D.

Leader du domaine, Inno³ dispose de toutes les compétences nécessaires pour adresser les trois piliers sur lesquels repose cette mutation : la dimension outil (par la gestion des progiciels Open Source), la dimension juridique (au travers de la gestion des licences Open Source et des contrats associés) et la dimension humaine (dans l'organisation des rapports avec les communautés dont sont issus les projets). Les offres de services proposées vont de la simple expertise ponctuelle (notamment audit et analyse d'opportunité) à l'accompagnement au changement (définition de stratégie, mise en place de politique et gouvernance dédiées) et au transfert de compétences (au travers de cursus préconstitués ou de modules « à la carte »).

La société anime enfin un réseau d'experts métiers qui permettent de porter les transformations liées à l'Open Source dans tous les secteurs industriels ou publics.

7 À propos des Auteurs



Fondateur du cabinet Inno³, **Benjamin Jean** travaille essentiellement sur les nouveaux usages et collaborations qui se développent autour de la création et de l'innovation (Open Source, Open Data ou plus généralement toute la mouvance en faveur d'une innovation ouverte). Actif depuis près de dix ans dans ce domaine, il a notamment œuvré pour leur prise en compte au sein des gouvernances de multiples acteurs privés et publics, de tailles et de domaines variés, et formé de nombreuses équipes internes aux aspects juridiques afférents.

Il enseigne par ailleurs la propriété intellectuelle au sein de plusieurs Masters spécialisés, intervient comme consultant au sein du cabinet Gilles Vercken et maître de conférences à Sciences Po.

Il est administrateur de PLOSS Paris Région, membre du Directoire de Systematic et Président de l'association *Open Law, le droit ouvert*. Il est également très présent dans les communautés du Logiciel Libre, administrateur de Framasoft et cofondateur de Veni Vidi Libri.



Juriste spécialisée en propriété intellectuelle et droit des nouvelles technologies, **Ségolène Delmas** est consultante junior au sein du cabinet Inno³. Ségolène est principalement en charge des questions d'innovation ouverte et des problématiques liées à la gestion de l'Open Source. Elle possède une expérience dans la gestion des actifs immatériels, et plus précisément des portefeuilles de marques et noms de domaines.

Diplômée du Master 2 Droit de l'Économie numérique de l'Université de Strasbourg et du Master 2 Droit de l'innovation technique de l'Université Paris XI, elle est également titulaire du CRFPA et couple donc cette dimension stratégique à de solides bases juridiques.

Annexe 1 – Évaluation du projet

Capital technique

Critère	Sous-critère	Question	Réponses
Code source	Quantité	Combien de lignes de code représente le logiciel développé au sein du projet ?	<ul style="list-style-type: none"> - Moins de 10,000 lignes - Entre 10,000 et 100,000 lignes - Entre 100,000 et 1 million de lignes - Entre 1 et 5 millions - Plus de 5 millions
	Difficulté technique	Quel est le(s) langage(s) de développement utilisé(s) ?	<ul style="list-style-type: none"> - Les langages utilisés sont des langages courants et accessibles à un grand nombre de personnes (ex : Python) - Les langages utilisés sont des langages courants mais ils sont utilisés dans une version ancienne - Les langages utilisés sont des langages peu fréquemment utilisés - Les langages utilisés sont des langages rares, que peu de personnes maîtrisent
	Qualité juridique	Les licences des composants utilisés ont-elles bien été respectées ?	<ul style="list-style-type: none"> - Aucune attention n'a été portée aux licences Open Source sous lesquelles étaient distribués les composants utilisés - Les licences ont bien été prises en compte, mais aucun outil spécifique n'a été utilisé pour assurer leur respect - Les licences ont bien été prises en compte et des outils spécifiques ont été utilisés pour assurer leur respect - Les licences ont bien été prises en compte, des outils spécifiques ont été utilisés et un audit du code source a été réalisé
	Accessibilité	Des dispositions ont-elles été prises pour faciliter l'entrée des développeurs sur l'écriture du code ? (easyhacks, mentors...)	<ul style="list-style-type: none"> - Oui - Non
	Qualité de la rédaction	Le code est-il rédigé de manière claire, permettant les modifications ?	<ul style="list-style-type: none"> - Le code développé est parfaitement clair, bien rédigé et bien documenté, rendant les modifications ultérieures faciles - Le code développé est clair et bien rédigé mais n'est pas suffisamment documenté, compliquant les modifications ultérieures - Le code développé manque de clarté et n'est pas très bien rédigé mais est bien documenté - Le code développé manque de clarté et de lisibilité, rendant les modifications ultérieures compliquées
Documentation	Qualité	Les détails de conception / d'architecture du logiciel sont-ils documentés ?	<ul style="list-style-type: none"> - Les détails de conception et d'architecture du logiciel sont bien documentés - Les détails de conception et d'architecture du logiciel sont partiellement documentés - Les détails de conception et d'architecture du logiciel ne sont pas documentés
		La documentation comporte-t-elle des instructions permettant aux développeurs de contribuer ?	<ul style="list-style-type: none"> - Le projet comporte des instructions claires et complètes pour permettre aux développeurs de coder, tant en termes d'écriture que de style - Le projet comporte bien des instructions, mais celles-ci ne sont pas claires et complètes

			- Le projet ne comporte aucune instruction à destination des développeurs
		La documentation comporte-t-elle des instructions permettant aux développeurs de contribuer ?	- Le projet comporte des instructions claires et complètes pour permettre aux développeurs de coder, tant en termes d'écriture que de style - Le projet comporte bien des instructions, mais celles-ci ne sont pas claires et complètes - Le projet ne comporte aucune instruction à destination des développeurs
	Accessibilité	La documentation comporte-t-elle une FAQ ?	- Le projet comporte une FAQ claire et complète, régulièrement mise à jour - Le projet comporte une FAQ claire et complète, mais celle-ci n'est pas toujours mise à jour - Le projet comporte une FAQ mais celle-ci n'est pas complète ou pas claire - Le projet ne comporte pas de FAQ
		Un fichier README est-il mis à disposition à la racine du projet ?	- Oui - Non
		La documentation est-elle rédigée en anglais ?	- Oui - Partiellement - Non

Capital humain

Critère	Sous-critère	Question	Réponses
Utilisateurs	Nombre d'utilisateurs	Combien de fois le produit a-t-il été téléchargé ? (dans sa dernière version stable)	- Le logiciel a été téléchargé plus de 10 millions de fois - Le logiciel a été téléchargé plus d'1 million de fois - Le logiciel a été téléchargé plus de 100,000 fois - Le logiciel a été téléchargé plus de 10,000 fois - Le logiciel a été téléchargé moins de 10,000 fois
		Quelle est l'évolution attendue ? (croissance...)	- Le nombre de téléchargements augmente rapidement - Le nombre de téléchargements tend à augmenter - Le nombre de téléchargements est stable - Le nombre de téléchargements tend à diminuer - Le nombre de téléchargements décroît rapidement
	Fidélité	Les utilisateurs sont-ils fidèles au projet ?	- Les utilisateurs sont très fidèles au projet - Les utilisateurs sont plutôt fidèles au projet - Les utilisateurs ne sont pas vraiment fidèles au projet - Les utilisateurs ne sont pas du tout fidèles au projet
		Les coûts de changement (switching costs) sont-ils importants ?	- Les coûts de changement sont très importants - Les coûts de changement sont importants mais pas rhédictoires - Il existe des coûts de changement, mais ils ne sont pas très importants - Il n'y a pas de coûts de changement
	Nombre d'utilisateurs actifs	Combien de personnes sont-elles inscrites aux outils de communication du projet (mailings lists, forums, asks...) ?	- Plus de 25,000 personnes - Plus de 10,000 personnes - Plus de 1,000 personnes - Moins de 500 personnes
		Combien de personnes sont-elles	- Plus de 500 personnes

		inscrites aux outils de chat (IRC / Telegram...)?	<ul style="list-style-type: none"> - Plus de 200 personnes - Plus de 100 personnes - Moins de 40 personnes
Contributeurs	Nombre de contributeurs	Combien de contributeurs le projet compte-t-il aujourd'hui? (submitters)	<ul style="list-style-type: none"> - Le projet compte plus de 1,500 submitters - Le projet compte plus de 500 submitters - Le projet compte plus de 100 submitters - Le projet compte moins de 10 submitters
		Combien de "sous-groupes" le projet compte-t-il? (groupe QA, groupe localisation, groupe marketing, groupe UX design, groupe documentation...)	<ul style="list-style-type: none"> - Le projet compte des sous-groupes dans plus de 5 catégories - Le projet compte des sous-groupes dans plus de 2 catégories - Le projet compte au moins un sous-groupe - Le projet ne compte aucun sous-groupe
		Quelle est l'évolution attendue? (croissance...)	<ul style="list-style-type: none"> - Le nombre de contributeurs augmente rapidement - Le nombre de contributeurs tend à augmenter - Le nombre de contributeurs est stable - Le nombre de contributeurs tend à diminuer - Le nombre de contributeurs décroît rapidement
	Profil	Quel est le niveau d'expertise des contributeurs?	<ul style="list-style-type: none"> - Les contributeurs ont un haut niveau d'expertise - Les contributeurs ont un niveau d'expertise relativement bon - Les contributeurs ne sont pas très expérimentés
		Observe-t-on une grande diversité dans leurs profils? (développeurs, assurance qualité, UX / UI design...)	<ul style="list-style-type: none"> - Les contributeurs ont des profils très variés - Les contributeurs ont des profils plutôt variés - Les contributeurs ont tous plus ou moins le même profil
		Les contributeurs sont-ils tous issus d'une même entreprise?	<ul style="list-style-type: none"> - 100% des contributeurs sont issus de la même entreprise - Plus de 50% des contributeurs sont issus de la même entreprise - Moins de 50% des contributeurs sont issus de la même entreprise - Moins de 20% des contributeurs sont issus de la même entreprise
	"Bus factor"	Combien de personnes représentent 50% du nombre total de commits?	<ul style="list-style-type: none"> - 1 à 2 - 3 à 4 - 5 à 10 - 10 à 20
	Fidélité	Les contributeurs sont-ils fidèles au projet?	<ul style="list-style-type: none"> - Les contributeurs sont très fidèles au projet - Les contributeurs sont plutôt fidèles au projet - Les contributeurs ne sont pas vraiment fidèles au projet - Les contributeurs ne sont pas du tout fidèles au projet

Capital organisationnel

Critère	Sous-critère	Questions	Réponses
Structuration de la communauté	Hiérarchie	La communauté est-elle structurée de manière horizontale ou verticale?	<ul style="list-style-type: none"> - La communauté est structurée de manière très verticale (grand nombre de niveaux hiérarchiques) - La communauté est structurée de manière plutôt verticale - La communauté est structurée de manière plutôt horizontale - La communauté est structurée de manière très

		horizontale (très faible nombre de niveaux hiérarchiques)
	Existe-t-il une forte hiérarchie ?	<ul style="list-style-type: none"> - Chaque sous-projet compte en moyenne moins de 5 personnes (forte hiérarchie) - Chaque sous-projet compte en moyenne moins de 10 personnes - Chaque sous-projet compte en moyenne entre 10 et 50 personnes - Chaque sous-projet compte en moyenne plus de 50 personnes (faible hiérarchie)
Mode de gouvernance	La communauté est-elle gouvernée par une fondation ?	<ul style="list-style-type: none"> - Oui - Non
Financement	Comment le projet est-il financé ? (dons des utilisateurs, subventions par la commission européenne, financement par les entreprises membres de la gouvernance...)	<ul style="list-style-type: none"> - Le projet est financé par des sources d'origines diverses - Le projet est financé par une source unique de financement - Le projet ne bénéficie d'aucune source de financement
Transparence	Le code source est-il accessible à tous, en ligne et dans sa dernière version ?	<ul style="list-style-type: none"> - Le code source est disponible en ligne, accessible à tous, dans sa dernière version - Le code source est disponible en ligne, accessible à tous mais pas toujours mis à jour - Le code source n'est pas disponible en ligne ou pas mis à jour
	Les décisions sont-elles rendues publiques ?	<ul style="list-style-type: none"> - Toutes les décisions prises au sein du projet sont rendues publiques - Une partie seulement des décisions sont rendues publiques - Les décisions ne sont jamais rendues publiques
Suivi de la communauté	Des outils spécifiques de suivi de la communauté sont-ils utilisés ? (Bitergia, Redmine...)	<ul style="list-style-type: none"> - Oui - Non
Efficacité de la gestion de la communauté	Le projet est-il présent sur les réseaux sociaux ?	<ul style="list-style-type: none"> - Oui, le projet est présent sur tous les réseaux sociaux et les comptes sont actifs - Oui, mais les comptes ne sont pas très actifs - Non
	Les retours utilisateurs sont-ils bien pris en compte ? (collecte des besoins, remontée des bugs...)	<ul style="list-style-type: none"> - Les retours utilisateurs sont parfaitement pris en compte, tant en matière de collecte des besoins que de remontée des bugs - Les retours utilisateurs sont partiellement pris en compte - Les retours utilisateurs ne sont pas pris en compte
Accessibilité	Tous les outils permettant aux développeurs de contribuer sont-ils bien mis à leur disposition ? (clés SSH, clés GPG, comment écrire un patch...)	<ul style="list-style-type: none"> - Tous les outils sont bien mis à la disposition des développeurs et sont facilement accessibles - Une partie des outils est manquante et/ou les outils ne sont pas très accessibles - Aucun outil n'est mis à la disposition des développeurs à cet effet
	Le process pour devenir committer est-il clairement expliqué ?	<ul style="list-style-type: none"> - Oui - Non
	Le process pour devenir committer est-il légitime ?	<ul style="list-style-type: none"> - Oui - Non
Dynamisme	Le code est-il fréquemment réécrit pour	- Plus de 80% du code est réécrit d'une année sur

	(développement)	ajouter de nouvelles fonctionnalités ?	l'autre - Plus de 50% du code est réécrit d'une année sur l'autre - Plus de 20% du code est réécrit d'une année sur l'autre - Plus de 10% du code est réécrit d'une année sur l'autre - Moins de 10% du code est réécrit d'une année sur l'autre
		Combien de commits sont faits chaque année ?	- Plus de 5,000 commits / an en moyenne - Plus de 2,000 commits / an en moyenne - Plus de 500 commits / an en moyenne - Moins de 500 commits / an en moyenne
		En combien de temps les demandes de commits sont-elles traitées ?	- 1 à 2 jours - 3 à 4 jours - Plus d'une semaine - Plus d'un mois
	Dynamisme (équipe)	L'équipe de développement est-elle stable ?	- Oui, tout à fait - Oui, plutôt - Non
		L'équipe de développement intègre-t-elle régulièrement de nouveaux développeurs ?	- Oui, régulièrement - De temps en temps - Rarement - Jamais
		Les modules sont-ils générables par une équipe réduite ? (15 développeurs max)	- Oui - Non
	Outils de communication	Les outils dédiés à la communication au sein de la communauté sont-ils facilement accessibles ? (listes de diffusion, chats, forums, ask...)	- Oui, tout à fait - Oui, plutôt - Non
Structuration technique	Processus de production du code source	Des processus ont-ils été mis en place pour la résolution des bugs ?	- Oui - Non
		Des outils spécifiques de suivi des bugs sont-ils utilisés ? (Bugzilla, Mantis...)	- Oui - Non
		Des processus ont-ils été mis en place pour la réalisation de tests ?	- Oui - Non
		Des outils spécifiques pour la réalisation de tests sont-ils utilisés ? (Tinderbox, Code coverage, CPPCheck...)	- Oui - Non
		Les modifications apportées au code source sont-elles bien documentées ?	- Les modifications sont toutes documentées, dans la langue du projet - Seules certaines modifications sont documentées et/ou elles sont faites dans une langue autre que celle du projet - Aucune des modifications apportées au code source n'est documentée
		Des outils d'intégration continue sont-ils utilisés ? (Jenkins, Travis CI...)	- Oui - Non
	Les conventions de chaque langage de code sont-elles bien respectées ?	- Oui - Non	
Processus de production de la documentation	Une liste de tâches à destination des contributeurs est-elle mise à leur disposition ?	- Oui - Non	

		La documentation est-elle régulièrement mise à jour ?	<ul style="list-style-type: none"> - La documentation est très fréquemment mise à jour - La documentation est rarement mise à jour - La documentation n'est jamais mise à jour
		Des outils spécifiques sont-ils utilisés pour mettre à jour automatiquement la documentation ? (Doxygen...)	<ul style="list-style-type: none"> - Oui - Non
	Structure du code	L'architecture du logiciel est-elle modulaire ?	<ul style="list-style-type: none"> - Oui - Non
Analyse de l'ouverture "technique" du projet	Dépôt du code source	Des plateformes spécifiques sont-elles utilisées pour le dépôt du code source ? (GitHub, GitLab, Gerrit...)	<ul style="list-style-type: none"> - Oui - Non
	Licence	La licence sous laquelle le projet est distribué est-elle clairement indiquée ?	<ul style="list-style-type: none"> - La licence est clairement indiquée - La licence est indiquée mais n'est pas située à un endroit très visible, ou la version de la licence n'est pas indiquée - La licence n'est pas indiquée
		Le projet est-il diffusé sous une licence plutôt permissive ou restrictive ?	<ul style="list-style-type: none"> - Le projet est diffusé sous une licence permissive - Le projet est diffusé sous une licence restrictive
	Releases	Le projet utilise-t-il le versionnage sémantique (main version, features change, bug fixes...)?	<ul style="list-style-type: none"> - Oui - Non
		Un changelog est-il publié avec chaque version ?	<ul style="list-style-type: none"> - Oui - Non
		Quelle est la fréquence de release ? ("release often, release early")	<ul style="list-style-type: none"> - Modèle basé sur des release à date fixe (ex : toutes les trois semaines) - Modèle basé sur des releases effectuées en fonction de l'avancement des développements

Marché

Critère	Sous-critère	Question	Réponses
Concurrence sur le marché	Concurrence directe	Le projet est-il concurrencé de manière directe ?	<ul style="list-style-type: none"> - Oui - Non
	Concurrence indirecte	Le projet est-il concurrencé de manière indirecte ?	<ul style="list-style-type: none"> - Oui - Non
	Caractère remplaçable	Le logiciel développé peut-il être remplacé par une autre solution ?	<ul style="list-style-type: none"> - Oui - Non
Réputation	Marque	La marque sous laquelle est diffusée le projet bénéficie-t-elle d'une grande notoriété ?	<ul style="list-style-type: none"> - La marque bénéficie d'une très grande notoriété - La marque bénéficie d'une certaine renommée - La marque bénéficie d'une petite renommée - La marque ne bénéficie d'aucune renommée
		La marque sous laquelle est diffusée le projet bénéficie-t-elle d'une bonne image auprès du public visé ?	<ul style="list-style-type: none"> - La marque bénéficie d'une très bonne image - La marque bénéficie d'une bonne image - La marque a une image plutôt moyenne - La marque souffre d'une mauvaise image
	Labellisation / Certification	Le projet est-il labellisé ? Bénéficie-t-il d'une certification ?	<ul style="list-style-type: none"> - Le projet bénéficie d'un label ou d'une certification

			- Le projet ne bénéficie d'aucun label ou certification
Clients	Nombre de clients	Combien de clients le projet compte-t-il aujourd'hui ?	- 1 à 2 - 3 à 4 - 5 à 10 - Plus de 10
		Quelle est l'évolution attendue ? (croissance...)	- Le nombre de clients augmente rapidement - Le nombre de clients tend à augmenter - Le nombre de clients est stable - Le nombre de clients tend à diminuer - Le nombre de clients décroît rapidement
	Type de clients	Les clients sont-ils des particuliers ou des entreprises ?	- Les clients sont exclusivement des entreprises - Les clients sont principalement des entreprises - Le projet compte autant d'entreprises que de particuliers en tant que clients - Les clients sont principalement des particuliers - Les clients sont exclusivement des particuliers
		S'il s'agit de professionnels, s'agit-il de TPE, PME, grands comptes ?	- Les clients sont principalement des grands comptes - Le projet compte à la fois des grands comptes et des petites structures comme clients - Les clients sont principalement des TPE ou PME
	Partenaires	Combien d'entreprises proposent-elles des solutions autour du projet ?	- 1 à 2 - 3 à 4 - 5 à 10 - Plus de 10
Stratégie industrielle	Niveau de généricité	Le logiciel développé est-il une application métier ou une plateforme générique ?	- Le logiciel est une application métier - Le logiciel est une plateforme générique
	Niveau de servicisation	Quel est le niveau de customisation de la solution nécessaire pour son utilisation ?	- La solution nécessite un fort niveau de customisation - La solution nécessite une petite customisation - La solution ne nécessite aucune customisation
		Des modules génériques / spécifiques doivent-ils être développés ? (développement à façon)	- Oui - Non
	Criticité (assurance)	Le logiciel développé a-t-il une forte criticité ? Son dysfonctionnement aurait-il un impact important ?	- Oui - Non
	Bien public industriel	Le logiciel développé est-il nécessaire pour le bon fonctionnement de l'industrie ? Exemples : - Les systèmes d'exploitation pour les constructeurs informatiques ; - Middleware type PHP/MySQL pour l'industrie du service et des applications métiers.	- Oui - Non
Existe-t-il des acteurs prêts à investir dans le projet pour cette raison ?		- Oui - Non	

Annexe 2 – Emprise de l'organisation sur le projet

Gouvernance

Critère	Sous-critère	Question	Barème
Roadmap	Décisions à long terme	De quelle façon sont-elles prises les décisions à long terme ?	<ul style="list-style-type: none"> - Les décisions sont uniquement prises par l'entreprise - Les décisions sont réservées à un certain nombre de personnes (dont l'entreprise) - Les décisions sont ouvertes à l'ensemble de la communauté
	Décisions à court terme	De quelle façon sont-elles prises les décisions à court terme ?	<ul style="list-style-type: none"> - Les décisions sont uniquement prises par l'entreprise - Les décisions sont réservées à un certain nombre de personnes (dont l'entreprise) - Les décisions sont ouvertes à l'ensemble de la communauté
Hierarchie	Contrôle des responsables	L'entreprise contrôle-t-elle les responsables du projet ?	<ul style="list-style-type: none"> - Oui - Non
		L'entreprise contrôle-t-elle les responsables des sous-projets / des modules?	<ul style="list-style-type: none"> - Oui - Non
Contrôle du marché	Notoriété de l'entreprise	L'entreprise bénéficie-t-elle d'une forte notoriété dans son domaine ?	<ul style="list-style-type: none"> - L'entreprise bénéficie d'une très grande notoriété dans son domaine - L'entreprise bénéficie d'une certaine notoriété dans son domaine - L'entreprise a une faible notoriété dans son domaine - L'entreprise n'a aucune notoriété dans son domaine
	Légitimité de l'entreprise	L'entreprise bénéficie-t-elle d'une certaine légitimité dans son domaine ?	<ul style="list-style-type: none"> - L'entreprise bénéficie d'une très grande légitimité dans son domaine - L'entreprise bénéficie d'une certaine légitimité dans son domaine - L'entreprise a une faible légitimité dans son domaine - L'entreprise n'a aucune légitimité dans son domaine

Titularité des droits

Critère	Sous-critère	Question	Réponses
Marque	Force du titre	Quelle est la force du titre (étendue géographique, notoriété, etc) ?	<ul style="list-style-type: none"> - La marque est très forte : grande notoriété, grande étendue géographique, grand nombre de produits et services protégés - La marque a une certaine force :
	Contrôle par l'organisation	Quelle est la part de contrôle de l'entreprise sur ce titre (titularité vs politique de marque du projet) ?	<ul style="list-style-type: none"> - L'entreprise a une politique de marque très restrictive - L'entreprise a une politique de marque plutôt restrictive - L'entreprise a une politique de marque ni très restrictive, ni très permissive - L'entreprise a une politique de marque plutôt permissive - L'entreprise a une politique de marque très permissive
Droit d'auteur	Droits sur le code source	L'entreprise possède-t-elle des droits qui lui donneraient un avantage sur le projet ?	<ul style="list-style-type: none"> - L'entreprise est titulaire de tous les droits d'auteur sur le code source - L'entreprise est titulaire de la majeure partie des droits d'auteur sur le code, le code restant pouvant être redéveloppé sans grande difficulté - L'entreprise n'est pas titulaire des droits d'auteur sur le code, ou pas de la majeure partie du code
	Droits sur la documentation	L'entreprise possède-t-elle les droits sur la documentation du logiciel développé ?	<ul style="list-style-type: none"> - Oui - Non
	Contrôle par l'organisation	Quelle est la part de contrôle de l'entreprise sur les droits d'auteur ? La licence est-elle permissive ou restrictive ?	<ul style="list-style-type: none"> - L'entreprise a une politique de licence des droits d'auteur plutôt stricte, la licence est restrictive - L'entreprise a une politique de licence des droits d'auteur plutôt souple, la licence est permissive
Brevet	Force du titre	Quelle est la force du titre (étendue géographique, potentiel d'innovation, etc.) ?	<ul style="list-style-type: none"> - Le brevet est très fort : grand nombre de pays couverts, fort potentiel d'innovation - Le brevet est plutôt fort : pays importants couverts, certain potentiel d'innovation - Le brevet est plutôt faible : peu de pays couverts, faible potentiel d'innovation
	Contrôle par l'organisation	Quelle est la part de contrôle de l'entreprise sur l'innovation brevetée ? La licence est-elle permissive ou restrictive ?	<ul style="list-style-type: none"> - L'entreprise a une politique de licence de brevet plutôt stricte, la licence est restrictive - L'entreprise a une politique de licence de brevet plutôt souple, la licence est permissive

Production du code

Critère	Sous-critère	Question	Réponses
Expertise	Niveau d'expertise au sein des équipes	L'entreprise possède-t-elle une forte expertise au sein de ses équipes ?	- Oui - Non
Moyens alloués	Moyens humains (ingénieurs)	Combien de développeurs salariés de l'entreprise travaillent sur le projet ?	- 1 à 2 - 3 à 4 - 5 à 10 - Plus de 10
		Quel est le pourcentage de committers employés par l'entreprise ?	- 100% des committers sont des salariés de l'entreprise - 80% des committers sont des salariés de l'entreprise - 60% des committers sont des salariés de l'entreprise - 40% des committers sont des salariés de l'entreprise - 20% des committers sont des salariés de l'entreprise - 0% des committers sont des salariés de l'entreprise
		Quel est le pourcentage de commits réalisés par des salariés de l'entreprise ?	- 100% des commits sont réalisés par des salariés de l'entreprise - 80% des commits sont réalisés par des salariés de l'entreprise - 60% des commits sont réalisés par des salariés de l'entreprise - 40% des commits sont réalisés par des salariés de l'entreprise - 20% des commits sont réalisés par des salariés de l'entreprise - 0% des commits sont réalisés par des salariés de l'entreprise
	Autres moyens humains	D'autres salariés de l'entreprise travaillent-ils sur le projet ? (juristes, brand managers, open source evangelists...)	- De nombreux autres salariés de l'entreprise travaillent sur le projet, à temps plein - Quelques salariés de l'entreprise travaillent sur le projet, à temps partiel - Aucun autre salarié de l'entreprise ne travaille sur le projet
	Autres moyens (financiers, locaux...)	L'entreprise alloue-t-elle d'autres moyens au projet ? (financiers, locaux...)	- Oui - Non