



# Mémento de Bonne Gouvernance Open Source

Par l'OSPO Alliance et les participants au GGI

Version : v1.1

Date : 2023-05-10

## Table des matières

<b>1 Introduction</b>	<b>3</b>
1.1 Contexte	3
1.2 À propos de l'initiative « Good Governance »	4
1.3 À propos de l' « OSPO Alliance »	4
1.4 Traductions	5
1.5 Contributeurs	5
1.6 Licence	5
<b>2 Organisation</b>	<b>7</b>
2.1 Terminologie	7
2.2 Objectifs	7
2.3 Activités canoniques	7
2.4 Tableaux de bord d'activité personnalisés	8
<b>3 Méthodologie</b>	<b>9</b>
3.1 Préparer le terrain	9
3.2 Flux de travail	9
3.3 Configuration manuelle : utiliser les tableaux de bord personnalisés	10
3.4 Configuration automatique : utilisation de la fonctionnalité de déploiement de GGI	11
3.5 Profitez-en !	12
<b>4 Activités de l'objectif « Usage »</b>	<b>14</b>
4.1 Inventaire des ressources et compétences open source	14
4.2 Montée en compétences open source	15
4.3 Supervision de l'open source	17
4.4 Logiciel open source professionnel	18
4.5 Gérer les compétences et les ressources en matière de développement de logiciels open source	19
<b>5 Activités de l'objectif « Confiance »</b>	<b>21</b>
5.1 Gérer la conformité juridique	21
5.2 Gérer les vulnérabilités des logiciels	22
5.3 Gérer les dépendances logicielles	24
5.4 Gérer les indicateurs clés	26
5.5 Faire des revues de code	28
<b>6 Activités de l'objectif Culture</b>	<b>30</b>
6.1 Promouvoir les bonnes pratiques de développement open source	30
6.2 Contribuer aux projets open source	31
6.3 Appartenir à la communauté open source	32
6.4 Point de vue des RH	33
6.5 Regarder vers l'amont (« Upstream first »)	36
<b>7 Activités de l'objectif « Engagement »</b>	<b>37</b>
7.1 S'engager dans des projets open source	37
7.2 Soutenir les communautés open source	38
7.3 Afficher publiquement l'usage de l'open source	39
7.4 S'engager auprès des fournisseurs de logiciels libres	39
7.5 Politique d'achat de logiciels libres	40
<b>8 Activités de l'objectif « Stratégie »</b>	<b>43</b>
8.1 Vers une stratégie de gouvernance open source d'Entreprise	43
8.2 Sensibiliser les cadres dirigeants	44
8.3 Open source et souveraineté numérique	45
8.4 L'open source au service de l'innovation	46
8.5 L'open source au service de la transformation numérique	48
<b>9 Conclusion</b>	<b>50</b>
9.1 Contact	50
9.2 Annexe : Modèle de tableau de bord personnalisé d'activité	50

# 1 Introduction

Ce document introduit une méthodologie pour mettre en œuvre une gestion professionnelle de l'open source au sein d'une organisation. Il répond aux besoins d'utiliser les logiciels open source de manière adéquate et équitable et de protéger l'organisation des risques techniques ou juridiques, afin de maximiser les avantages de l'open source. Quel que soit votre degré d'implication dans ce domaine, ce document vous proposera conseils et idées vous permettant de progresser et faire de votre démarche un succès.

## 1.1 Contexte

La plupart des grands utilisateurs ou intégrateurs utilisent déjà du logiciel open source (FOSS = Free and Open Source Software), que ce soit dans leur propre système d'information, ou dans leur offre de produits ou services. La conformité (parfois nommée « compliance ») est une préoccupation croissante, et beaucoup de grandes entreprises ont nommé des responsables dédiés. Cependant, bien que l'assainissement de la chaîne de production open source d'une organisation soit fondamental (c'est généralement le but de la conformité), les utilisateurs *doivent* soutenir les communautés et contribuer à la soutenabilité de l'écosystème open source. Nous concevons la gouvernance de l'open source comme englobant l'écosystème dans son ensemble, s'engageant dans les communautés locales, cultivant une saine relation avec les fournisseurs et prestataires de service du logiciel open source. La notion de « conformité » change alors de dimension : voilà ce qu'est la *bonne* gouvernance de l'open source.

Cette initiative va au-delà de la conformité et de la responsabilité juridique. Il y est question de sensibiliser les utilisateurs (souvent eux-mêmes développeurs de logiciel) et intégrateurs, et de développer des relations mutuellement bénéfiques au sein de l'écosystème open source européen.

La Bonne Gouvernance Open Source permet à tous les types d'organisations (PME ou grandes entreprises, mairies, universités, associations, etc.) de maximiser les avantages tirés de l'open source en les aidant à aligner les processus, les gens, la technologie et la stratégie. Et dans ce domaine, optimiser les avantages de l'open source, particulièrement en Europe, nous sommes tous en phase d'apprentissage et d'innovation, et nul ne sait vraiment où il en est vis-à-vis de l'état de l'art.

Cette initiative vise à aider les organisations à atteindre ce but, avec :

- Un catalogue structuré d'**activités**, feuille de route pour la mise en œuvre d'une gestion professionnelle de l'open source.
- Un **outillage de gestion** pour définir, suivre, rapporter et communiquer l'état de la progression.
- Un **cheminement clair et pratique vers l'amélioration**, graduel et abordable, pour atténuer les risques, former les gens, adapter les processus, communiquer au sein et à l'extérieur de l'organisation.
- **Des conseils d'orientation** et un ensemble de **références sélectionnées** au sujet des licences open source, des bonnes pratiques, de la formation, et de l'engagement dans l'écosystème, pour tirer parti de la connaissance et la culture open source afin de consolider son savoir interne et d'étendre son leadership.

Ce guide a été développé dans le respect des exigences suivantes :

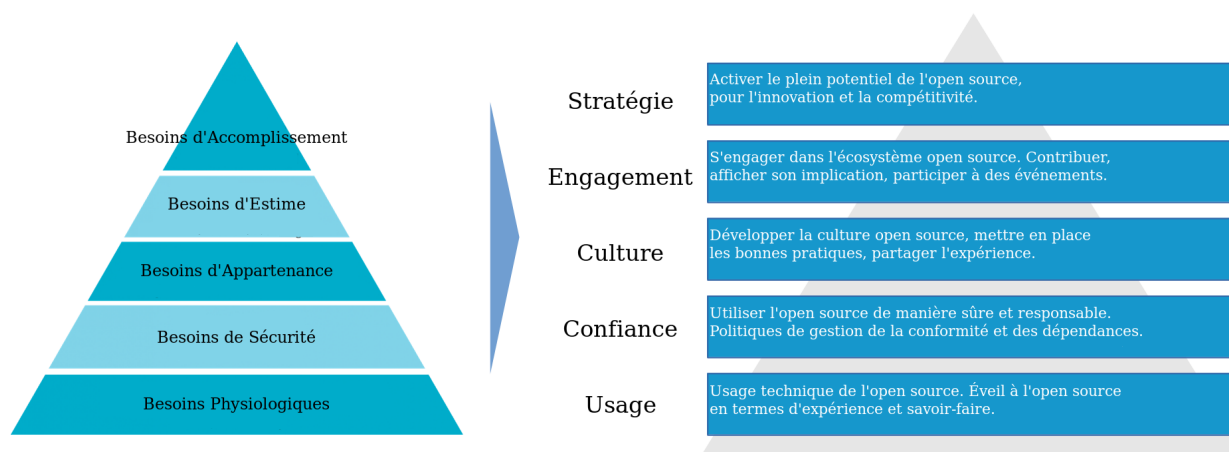
- Être applicable à tout type d'organisation : de la PME à la grande entreprise ou l'organisation sans but lucratif, des autorités locales (comme les mairies) aux grandes institutions (Européennes ou gouvernementales, par exemple). Le cadre fournit des briques pour bâtir une stratégie et des indications pour sa mise en œuvre, mais *comment* les activités sont exécutées dépend entièrement du contexte et reste à la discrétion du responsable du programme. Il peut être utile d'avoir recours à du conseil ou d'échanger avec ses pairs.
- Ne faire aucune hypothèse quant au niveau de connaissance technique interne à l'organisation ou au secteur d'activité. Par exemple, certaines organisations auront besoin de mettre en place un parcours complet de formation, alors que d'autres pourraient se contenter de proposer du contenu ad-hoc aux équipes.

Certaines activités ne seront pas applicables à toutes les situations, mais le cadre global fournit une feuille de route détaillée et ouvre la voie à des stratégies sur mesure.

## 1.2 À propos de l'initiative « Good Governance »

Chez OW2, une initiative est un effort commun pour satisfaire un besoin professionnel. L'initiative **OSS Good Governance** propose un cadre méthodologique pour mettre en place une gestion professionnelle de l'open source au sein d'une organisation.

L'initiative « Good Governance » est basée sur un modèle détaillé inspiré par la célèbre pyramide d'Abraham Maslow pour décrire les motivations humaines, comme illustré par l'image ci-dessous.



hiérarchie des motivations humaines par Abraham Maslow

Objectifs d'une bonne gouvernance open source, par OW

Par les idées, directives et activités qu'elle propose, l'initiative « Good Governance » est un schéma directeur de mise en place d'entités chargées de la gestion des logiciels open source, habituellement appelées OSPO (pour « Open Source Program Office »). La méthode est aussi un système de management pour définir des priorités, assurer le suivi et partager les progrès réalisés.

Les organisations mettant en place la méthodologie « Good Governance » vont accroître leur savoir-faire dans plusieurs directions, parmi lesquelles :

- **utiliser** le logiciel open source à bon escient et de façon sûre pour améliorer la réutilisation et la maintenabilité du logiciel et accroître l'agilité de développement ;
- **atténuer** les risques juridiques et techniques associés au code de provenance externe et à la collaboration ;
- **identifier** les besoins de formation des équipes, des développeurs aux chefs d'équipe et managers, afin que tout le monde partage une même vision ;
- **prioriser** les objectifs et activités, pour développer une stratégie open source efficace ;
- **communiquer** efficacement en interne et vers l'extérieur pour tirer le meilleur de la stratégie open source ;
- **améliorer** la compétitivité et l'attractivité de l'organisation pour les meilleurs talents de l'open source.

## 1.3 À propos de l' « OSPO Alliance »

L'**OSPO Alliance** a été créée par une coalition d'organisations européennes sans but lucratif leaders dans l'open source, incluant OW2, la fondation Eclipse, OpenForum Europe, et la Foundation for Public Code, avec la mission d'accroître la sensibilisation à l'open source en Europe et dans le monde et promouvoir un management structuré et professionnel de l'open source par les entreprises et les administrations.

Alors que l'initiative « Good Governance » se focalise sur la mise au point d'une méthode de management, l' « OSPO Alliance » intègre l'objectif plus large d'aider les entreprises, particulièrement dans les secteurs non technologiques, et les institutions publiques à découvrir et comprendre l'open source, commencer à en bénéficier dans leurs activités, et évoluer jusqu'à accueillir leur propre OSPO.

L'« OSPO Alliance » a mis en place le site web **OSPO Alliance** accessible sur <https://ospo-alliance.org>. L'« OSPO Alliance » offre à la communauté un endroit privilégié pour discuter et échanger sur les sujets de l'OSPO, et fournit un référentiel pour un ensemble complet de ressources pour les entreprises, institutions publiques, et organismes académiques ou de recherche. L'« OSPO Alliance » permet de se connecter avec les OSPO à travers l'Europe et le monde, ainsi qu'avec les organisations communautaires soutenant la démarche, pour encourager les bonnes pratiques et les contributions à soutenir l'écosystème open source. Visitez le site web [OSPO Alliance](https://ospo-alliance.org) pour un aperçu rapide du cadre des meilleures pratiques de gestion de l'IT.

Le site web [OSPO Alliance](https://ospo-alliance.org) est également l'endroit où nous collectons les retours utilisateurs concernant l'initiative et son contenu (par exemple, les activités, le corpus des connaissances...), provenant de la communauté au sens large.

## 1.4 Traductions

Un travail collaboratif est en cours pour traduire le GGI Handbook dans différentes langues. Nous vous recommandons de consulter le site web officiel pour une liste à jour des traductions disponibles.

Voir <https://hosted.weblate.org/projects/ospo-zone-ggi/#languages>

Le handbook GGI est traduit à l'aide de [Weblate](https://www.weblate.org/), un projet et une plate forme open source qui propose de l'hébergement gratuit pour les projets open source. Nous tenons à les remercier chaleureusement, ainsi que tous les contributeurs ayant participé à la traduction. Vous assurez.

## 1.5 Contributeurs

Merci aux talentueux contributeurs de l'initiative « Good Governance » :

- Frédéric Aatz (Microsoft France)
- Boris Baldassari (Castalia Solutions, Eclipse Foundation)
- Philippe Bareille (Ville de Paris)
- Gaël Blondelle (Eclipse Foundation)
- Vicky Brasseur (Wipro)
- Philippe Carré (Nokia)
- Pierre-Yves Gibello (OW2)
- Michael Jaeger (Siemens)
- Sébastien Lejeune (Thales)
- Max Mehl (Free Software Foundation Europe)
- Hervé Pacault (Orange)
- Stefano Pampaloni (RIOS)
- Christian Paterson (OpenUp)
- Simon Phipps (Meshed Insights)
- Silvério Santos (Orange Business Services)
- Cédric Thomas (OW2)
- Nicolas Toussaint (Orange Business Services)

## 1.6 Licence

Cette œuvre est publiée sous licence [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) (CC-BY 4.0). Selon le site web Creative Commons :

Vous êtes autorisé(e) à :

- partager — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats
- Adapter — remixer, transformer et créer à partir du matériel

pour toute utilisation, y compris commerciale.

Vous devez créditer l'œuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'œuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'offrant vous soutient ou soutient la façon dont vous avez utilisé son œuvre.

Tout le contenu est sous Copyright : 2020-2022 OW2 & les participants au GGI.

## 2 Organisation

### 2.1 Terminologie

Le schéma directeur de la méthodologie « Good Governance » est structuré autour de quatre concepts : les objectifs, les activités, les tableaux de bord d'activité, et les itérations.

- **Objectifs** : un objectif est un ensemble d'activités liées à un sujet commun. On définit cinq objectifs : l'Usage, la Confiance, la Culture, l'Engagement et la Stratégie. Les objectifs peuvent être atteints indépendamment, en parallèle, et complétés de manière itérative, à travers les activités.
- **Activités** : dans le cadre d'un objectif, une activité concerne une préoccupation ou un sujet particulier - par exemple, la prise en compte de la conformité juridique - que l'on peut considérer comme une étape vers les objectifs du programme. L'ensemble des activités défini par la méthode GGI est appelé « activités canoniques ».
- **Tableaux de bord d'activité** : pour mettre en œuvre la GGI dans une organisation, les activités canoniques doivent être adaptées au contexte, ce qui donne naissance à un ensemble de « tableaux de bord d'activité » personnalisés. Un tableau de bord d'activité personnalisé décrit comment l'activité sera mise en œuvre dans le contexte d'une organisation, et comment son avancement sera contrôlé.
- **Itérations** : la méthodologie GGI est un système de management, qui nécessite donc des évaluations, examens et révisions périodiques. Prenons pour exemple la comptabilité dans une organisation : c'est un processus continu avec au moins un arrêté annuel, le compte de résultats ; de même, le processus GGI nécessite au moins une revue annuelle, quand bien même les revues peuvent être plus ou moins partielles ou fréquentes selon les activités.

### 2.2 Objectifs

Les activités définies par GGI sont regroupées par « objectifs ». Chaque objectif concerne un domaine spécifique d'avancement du processus. De l'Usage à la Stratégie, les objectifs couvrent des problématiques liées à toutes les parties prenantes, des équipes de développement aux cadres dirigeants.

- **Usage** : cet objectif recouvre les étapes initiales de l'utilisation de logiciel open source. Les activités liées à cet objectif sont les premières étapes d'une programmation de l'open source, visant à identifier avec quelle efficacité l'open source est utilisé et ce qu'il apporte à l'organisation. Ceci inclut la formation et la gestion du savoir, l'inventaire de l'open source déjà utilisé en interne, et la mise en évidence de certains concepts de l'open source qui peuvent être mis en œuvre lors du processus.
- **Confiance** : cet objectif cible un usage sûr de l'open source. La confiance est question de conformité juridique, de gestion des dépendances et des vulnérabilités, il s'agit de confiance quant à l'usage et la gestion de l'open source au sein de l'organisation.
- **Culture** : l'objectif culturel se compose d'activités destinées à rendre les équipes à l'aise avec l'open source, la participation individuelle à des activités collaboratives, la compréhension et la mise en œuvre des bonnes pratiques de l'open source. Cet objectif favorise le sentiment d'appartenance des individus à la communauté de l'open source.
- **Engagement** : cet objectif vise à engager une organisation dans l'écosystème open source. Des ressources humaines et financières sont allouées pour contribuer à des projets open source. Ici, l'organisation assume son rôle citoyen dans l'open source et sa responsabilité à garantir la soutenabilité de l'écosystème.
- **Stratégie** : cet objectif consiste à rendre l'open source visible et acceptable au plus haut niveau du management de l'organisation. Il s'agit de reconnaître que l'open source est un catalyseur stratégique de souveraineté numérique, d'innovation, d'attractivité et d'image positive.

### 2.3 Activités canoniques

Les « activités canoniques » sont au centre du schéma GGI. Dans la version actuelle, la méthodologie GGI propose cinq activités canoniques par objectif, soit vingt-cinq au total. Les activités canoniques sont décrites en utilisant le modèle suivant de division en sections :

- *Description* : résumé du thème abordé par l'activité et des étapes de sa mise en œuvre.

- *Évaluation des opportunités* : décrit quand et pourquoi il est pertinent d'entreprendre l'activité.
- *Suivi de l'avancement* : décrit comment mesurer la progression de l'activité et évaluer son succès.
- *Outils* : liste de technologies et d'outils qui peuvent aider à accomplir l'activité.
- *Recommandations* : conseils et bonnes pratiques proposés par les contributeurs à GGI.
- *Ressources* : liens et références pour en savoir plus quant à la thématique liée à l'activité.

### **Description**

Cette section propose une description générale de l'activité, un résumé thématique pour positionner l'activité et l'approche de l'open source dans le contexte d'un objectif.

### **Évaluation de l'opportunité**

Pour aider à structurer une approche itérative, chaque activité propose une section « Évaluation des opportunités », avec une ou plusieurs question(s) associée(s). L'évaluation se concentre sur ce qui est pertinent pour entreprendre cette activité, et les besoins auxquels elle répond. Évaluer l'opportunité aidera à définir les efforts et les ressources nécessaires, ainsi qu'à évaluer les coûts et le retour sur investissement attendu.

### **Suivi de l'avancement**

Cette étape se concentre sur la définition d'objectifs, de KPI, et la fourniture de *points de contrôle* aidant à évaluer la progression de l'activité. Les points de contrôle sont proposés, ils peuvent aider à définir un planning pour le processus GGI, ses priorités, et la manière d'en mesurer la progression.

### **Outils**

Cette section liste des outils pouvant aider à mettre en œuvre l'activité ou en instrumenter une étape particulière. Il ne s'agit pas de recommandations impératives, et la liste d'outils n'est pas exhaustive : il s'agit de suggestions sur lesquelles on peut se baser avant de les adapter au contexte.

### **Recommandations**

Cette section sera régulièrement mise à jour avec le retour d'expérience d'utilisateurs, entre autres recommandations pouvant aider à mener à bien l'activité.

### **Ressources**

Les ressources se composent d'études de cas, documents de référence, événements ou contenu en ligne, dont la thématique peut permettre d'enrichir et développer l'activité. Les ressources ne sont pas exhaustives, elles servent de point de départ ou de suggestions pour étendre la sémantique de l'activité à un contexte particulier.

## **2.4 Tableaux de bord d'activité personnalisés**

Les tableaux de bord d'activité personnalisés sont nettement plus détaillés que les « activités canoniques ». Un tableau de bord inclut des détails spécifiques à l'organisation qui met en œuvre GGI. L'usage des tableaux de bord est décrit dans la section « Méthodologie ».



## 3 Méthodologie

Mettre en place la méthodologie GGI est une initiative à impact, ayant des conséquences. Cela implique plusieurs services, processus et catégories de personnel, des développeurs aux cadres dirigeants et du travail quotidien à la gestion des ressources humaines. Il n'y a pas de solution miracle pour établir une bonne gouvernance de l'open source. Différents types de culture d'entreprise et de situations impliqueront différentes approches de gouvernance de l'open source. Pour chaque organisation, contraintes et attentes seront différentes, conduisant à différentes manières de mettre en œuvre le programme.

Dans ces conditions, la méthodologie GGI fournit un schéma directeur générique pouvant être adapté au domaine, à la culture et aux besoins particuliers d'une organisation. Alors que le schéma se veut complet, la méthodologie peut être mise en œuvre progressivement. Il est possible de démarrer le programme en sélectionnant les objectifs et activités jugés pertinents dans un contexte précis : l'idée étant de construire une première ébauche de feuille de route, aidant à démarrer l'initiative.

Au-delà de ce cadre, nous recommandons aussi d'entrer en contact avec ses homologues ou confrères via un réseau établi, comme l'initiative [OSPO Alliance](#) en Europe, ou d'autres initiatives comparables telles que TODO Group ou OSPO++. L'important étant de pouvoir échanger avec des gens engagés dans une initiative similaire, et partager les problèmes rencontrés et les solutions appliquées.

### 3.1 Préparer le terrain

Vu l'ambition de la méthodologie GGI et son impact potentiel, il est important de communiquer avec diverses personnes au sein de l'organisation. Il serait adéquat de les impliquer afin de définir un ensemble d'attentes et de besoins réalistes pour partir sur de bonnes bases, et susciter l'intérêt et l'adhésion. Il paraît opportun de publier les tableaux de bord personnalisés dans la plateforme collaborative de l'organisation, afin qu'ils puissent servir à communiquer avec les parties prenantes. Quelques conseils :

- Identifier les parties prenantes, les faire s'entendre sur un ensemble d'objectifs primaires. Les impliquer dans le succès de l'initiative dans le cadre de leur activité quotidienne.
- Obtenir l'adhésion nécessaire à démarrer, un accord sur les étapes et le rythme, et prévoir des points réguliers pour informer les acteurs de l'avancement.
- S'assurer que les acteurs comprennent les avantages de ce qui sera réalisé, et ce que ça implique : les améliorations attendues doivent être clairement formulées, et les résultats visibles.
- Établir un diagnostic de l'état de l'art quant à l'open source au sein de l'organisation concernée. Résultat : un document décrivant ce que le programme va réaliser, où en est l'organisation, et où elle souhaite aller.

### 3.2 Flux de travail

En tant que praticiens modernes du logiciel, nous apprécions les méthodes agiles qui s'appuient sur des incréments progressifs et sûrs, considérant de bon aloi de réévaluer la situation régulièrement et de fournir un minimum de résultats intermédiaires significatifs.

Dans le contexte vivant d'un programme d'OSPO, ceci est notoirement pertinent, car de nombreux aspects secondaires vont évoluer au fil du temps, depuis la stratégie et la réaction de l'organisation vis-à-vis de l'open source jusqu'à l'implication et la disponibilité des personnes impliquées. L'itération et la réévaluation périodique permettent également de s'adapter à l'acceptation du programme, de mieux suivre les tendances et opportunités, et de laisser l'organisation et les parties prenantes prendre acte des améliorations progressives.

Idéalement, la méthodologie peut être mise en œuvre en cinq phases, comme suit :

1. **Découverte** Comprendre les concepts clés, s'approprier la méthodologie, aligner les objectifs et les attentes.
2. **Personnalisation** Adapter la description des activités et l'évaluation des opportunités aux spécificités de l'organisation.

3. **Priorisation** Identifier les résultats et objectifs principaux, les tâches et outils, planifier les étapes et établir un premier échéancier.
4. **Activation** Finaliser les tableaux de bord, le budget, les affectations, documenter les tâches et les intégrer à son outillage de gestion projet.
5. **Itération** Évaluer et mesurer les résultats, mettre en évidence les problèmes, améliorer, ajuster. Itérer chaque trimestre ou semestre.

Se préparer à la première itération du programme :

- Identifier un premier ensemble de tâches à réaliser, et les prioriser selon les besoins (écarts à la situation cible) et l'échéancier. Résultat : liste des tâches sur lesquelles travailler durant l'itération.
- Définir un ensemble d'exigences et de domaines d'amélioration, les communiquer aux parties prenantes et utilisateurs, et obtenir leur approbation ou engagement.
- Remplir les tableaux de bord pour suivre la progression. Un modèle de tableau de bord peut être téléchargé sur le [dépôt GGI](#).

À la fin de chaque itération, faire une rétrospective et se préparer à l'itération suivante :

- Communiquer sur les derniers progrès.
- Évaluer où vous en êtes, si les tâches prévues ont été finalisées, et redéfinir le planning en fonction.
- Vérifier les points bloquants et les problèmes, demander assistance aux autres acteurs ou services si besoin.
- Re-prioriser les tâches selon le contexte mis à jour.
- Définir un nouveau sous-ensemble de tâches à effectuer.

### 3.3 Configuration manuelle : utiliser les tableaux de bord personnalisés

Un tableau de bord personnalisé est un formulaire décrivant une « activité canonique » adaptée aux besoins spécifiques d'une organisation. L'ensemble des tableaux de bord personnalisés constitue la feuille de route pour la gestion de l'open source.

Pour information, de par l'expérience acquise, compter environ une heure pour adapter une activité canonique et l'intégrer à un tableau de bord personnalisé pour votre organisation.

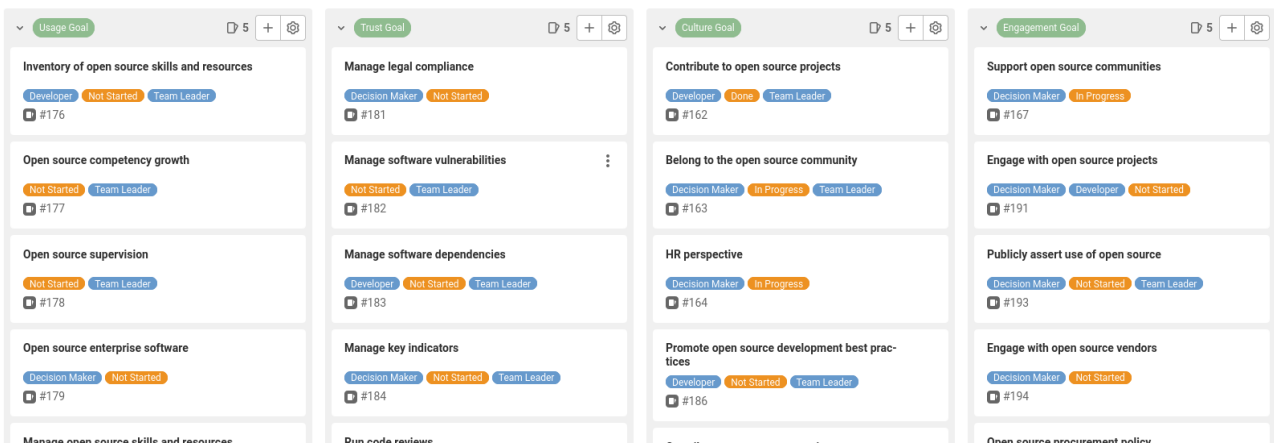
Le tableau de bord personnalisé se compose des sections suivantes :

- **Intitulé mis au clair** Avant tout, prenez quelques minutes pour bien comprendre à quoi s'applique l'Activité et évaluer sa pertinence, dans le cadre de votre cheminement global vers la gestion de l'open source.
- **Description personnalisée** Adaptez l'activité aux spécificités de votre organisation : définissez son champ d'application, le cas d'usage particulier auquel l'appliquer.
- **Évaluation de l'opportunité** Expliquez en quoi il est pertinent d'entreprendre cette activité, à quels besoins elle répond. Quels sont nos points faibles ? Nos opportunités de progrès ? Que peut-on gagner ?
- **Objectifs** Définissez quelques objectifs centraux pour l'activité. Points à renforcer, opportunités de progrès, souhaits. Identifiez les tâches essentielles. Ce que l'on souhaite réaliser dans cette itération.
- **Outils** Technologies, outils et produits utiles à cette activité.
- **Notes opérationnelles** Indications sur l'approche, la méthode, la stratégie pour avancer dans cette activité.
- **Résultats clés** Définissez des résultats mesurables et vérifiables. Choisissez des résultats attestant de progrès en phase avec les objectifs. Listez les KPI correspondants.
- **Progression et score** La progression est, en %, le pourcentage d'achèvement du résultat ; Le score est la note de réussite personnelle.
- **Évaluation personnelle** Pour chaque résultat, vous pouvez fournir une courte explication et exprimer votre taux de satisfaction personnelle via le score.
- **Chronologie** Précisez les dates de début et de fin, un phasage des tâches, les étapes critiques, les jalons.
- **Efforts** Évaluez les besoins en temps et ressources matérielles, internes et externes. Quels sont les efforts attendus ? Quel en sera le coût ? De quelles ressources a-t-on besoin ?

- **Acteurs** Listez les participants. Définissez qui est responsable ou en charge des tâches et activités.
- **Problèmes et enjeux** Identifiez les principaux enjeux, les difficultés attendues, les risques, points de blocage, incertitudes, points d'attention, dépendances critiques.
- **État** Rédigez une évaluation synthétique expliquant comment se porte l'activité : situation saine ? retards ? etc.
- **Évaluation globale des progrès** Votre propre évaluation synthétique du progrès de l'activité, axée sur la gestion et envisagée globalement.

### 3.4 Configuration automatique : utilisation de la fonctionnalité de déploiement de GGI

À partir de la version 1.1 du manuel, GGI propose [My GGI Board](#), un outil automatisé permettant de déployer votre propre instance de la GGI en tant que projet GitLab. Le processus d'installation prend moins de 10 minutes, est entièrement documenté et offre un moyen simple et fiable de personnaliser les activités, de suivre leur exécution au fur et à mesure de leur progression et de communiquer les résultats à vos partenaires. Voici un exemple concret du déploiement : [GitLab de l'initiative](#), avec le site Web généré automatiquement disponible sur [ses pages GitLab](#).



Voici un workflow standard pour utiliser la fonctionnalité de déploiement :

1. Forkez My GGI Board vers votre propre instance ou projet GitLab et configurez-le en suivant les instructions du fichier README : <https://gitlab.ow2.org/ggi/my-ggi-board> du projet. Cela permettra :
  - Créez toutes les activités en tant que tickets dans le projet.
  - Créez un joli dashboard pour vous aider à visualiser et à gérer les activités.
  - Créez un site web statique, hébergé sur les pages de votre instance GitLab, avec les informations extraites des activités.
  - Mettez à jour la description du projet avec les liens appropriés vers le dashboard d'activités et votre site web statique.
1. À partir de là, vous pouvez commencer à examiner les différentes activités et à remplir la section du tableau de bord.
  - La section des tableaux de bord est l'équivalent électronique (et simplifié) des tableaux de bord ODT mentionnés ci-dessus. Elles permettent d'adapter l'activité à votre contexte, en répertoriant les ressources locales, les risques et les opportunités, et en définissant des objectifs personnalisés nécessaires à la réalisation de l'activité.
  - Si une activité ne s'applique pas à votre contexte, il suffit de la marquer comme "Non Sélectionnée" ou de la fermer.
  - Il s'agit d'un processus assez long, mais essentiel car il vous aidera, étape par étape, à définir votre propre feuille de route et votre propre plan.
1. Une fois les activités définies, vous pouvez commencer à mettre en œuvre votre propre OSPO. Sélectionnez quelques activités qui vous semblent pertinentes pour commencer, et changez leur label de progression de l'état "Pas encore commencé" à l'état "En cours". Vous pouvez utiliser les fonctionnalités de GitLab pour vous aider à organiser le travail

(commentaires, attributaires, etc.) ou tout autre outil. Il est facile de créer des liens avec les activités, et il existe de nombreuses intégrations de qualité.

- Régulièrement (chaque semaine, chaque mois, selon votre emploi du temps), évaluez et passez en revue les activités en cours et lorsqu'elles sont terminées, changez le label de «En cours» à «Terminé». Sélectionnez-en quelques autres et recommencez à l'étape 3 jusqu'à ce qu'ils soient tous terminés.

Le site web propose une vue d'ensemble rapide des activités actuelles et passées, et extrait la section tableau de bord des tickets pour n'afficher que les informations pertinentes au niveau local. Lorsque des modifications sont apportées aux tickets (activités), elles sont automatiquement mises à jour dans le site web généré. A noter que les pipelines CI pour la génération automatique du site web sont exécutés automatiquement chaque nuit, mais vous pouvez facilement les lancer à partir de la section CI/CD du projet GitLab. L'image suivante montre l'interface du site web généré automatiquement.

The screenshot shows a web interface for 'My Good Governance Initiative'. At the top, there are navigation links for 'Dashboard' and 'My Board'. Below this is a 'Welcome' section with a sub-header 'Welcome' and a description: 'This is the website of your own good governance Initiative.' It then states 'There are currently:' followed by a list of activity counts: '17 activities not\_started', '4 activities in\_progress', and '4 activities done'. Below this is a section for 'Current activities' with a '[ details ]' link. It defines current activities as those with the 'in\_progress' label and lists two activities with progress bars: 'Open source enabling digital transformation (GGI-A-37)' at 50% and 'Support open source communities (GGI-A-30)' at 66%.

Vous pouvez poser des questions ou obtenir de l'aide pour la fonction de déploiement sur notre page d'accueil GitLab, les commentaires sont les bienvenus.

Page d'accueil de GGI Deploy : <https://gitlab.ow2.org/ggi/my-ggi-board>

### 3.5 Profitez-en !

Communiquez sur votre réussite et appréciez la tranquillité d'esprit que procure une stratégie open source à l'état de l'art !

GGI est une méthodologie permettant de déployer un programme d'amélioration continue, et à ce titre, n'a pas de fin. Quoi qu'il en soit, il est important de mettre en valeur ses étapes intermédiaires et apprécier les évolutions qu'elles engendrent, pour rendre le progrès perceptible et partager les résultats.

- Communiquez avec les parties prenantes et les utilisateurs pour leur faire connaître les avantages et bénéfices apportés par les efforts qu'impose l'initiative.
- Favorisez la durabilité du programme. Assurez-vous que les bonnes pratiques et les leçons acquises via le programme continuent à être appliquées et mises à jour.

- Partagez votre expérience avec vos pairs : fournissez un retour d'information au groupe de travail GGI et à votre communauté OSPO d'adoption, et partagez votre approche.

## 4 Activités de l'objectif « Usage »

### 4.1 Inventaire des ressources et compétences open source

Identifiant de l'activité : [GGI-A-20](#).

#### Description

À chaque étape, dans une perspective de management, il est utile de disposer d'une cartographie, d'un inventaire des ressources, actifs et usages open source, de leur statut, ainsi que des besoins potentiels et solutions disponibles. Il s'agit également d'évaluer les efforts et compétences nécessaires à combler les manques.

Cette activité vise à prendre un cliché de la situation interne de l'open source, de l'état de l'art du marché, et d'évaluer l'écart entre eux.

- Inventorier l'usage de l'OSS dans la chaîne de développement logiciel comme dans les produits et composants logiciels utilisés en production.
- Identifier les technologies open source (solutions, frameworks, fonctionnalités innovantes) pouvant correspondre à vos besoins et aider à améliorer vos processus.

Ceci n'inclut pas

- L'identification et qualification des écosystèmes et communautés OSS liées (Objectif « Culture »).
- L'identification des dépendances vis-à-vis de librairies et composants OSS (Objectif « Confiance »).
- La détermination des compétences techniques (langages, frameworks...) et relationnelles (collaboration, communication...) nécessaires (faisant partie d'autres Activités : « Montée en compétence OSS » et « Compétences en développement logiciel open source »).

#### Évaluation de l'opportunité

Inventaire des ressources open source disponibles pouvant aider à optimiser l'investissement et prioriser le développement des compétences.

Cette activité crée les conditions utiles à améliorer la productivité du développement, eu égard à l'efficacité et la popularité des composants, outils et principes de développement OSS pour produire des applications et infrastructures modernes.

- Ceci peut impliquer de simplifier le portefeuille de ressources OSS.
- Ceci peut nécessiter de former du personnel.
- Ceci permet d'identifier des besoins et alimenter votre feuille de route IT.

#### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Il existe une liste exploitable de ressources OSS que l'on « utilise », « intègre », « produit », « héberge », et les compétences associées.
- Nous sommes en voie d'amélioration de l'efficacité en utilisant des méthodes et outils à l'état de l'art.
- Nous avons identifié des ressources OSS non répertoriées à ce jour (qui peuvent avoir émergé en interne : est-il possible de définir une politique dans ce domaine ?)
- Nous demandons aux nouveaux projets d'approuver ou réutiliser des ressources OSS existantes (Objectif « Culture » ?).
- Nous avons une perception et compréhension raisonnablement fiables de l'étendue des usages OSS au sein de notre organisation.

#### Outils

Il existe beaucoup de possibilités d'établir un tel inventaire. Une méthode peut consister à classer les ressources OSS en quatre catégories :

- Ce qu'on utilise : le logiciel utilisé en production ou développement ;

- Ce qu'on intègre : exemple, les bibliothèques OSS que l'on intègre à une application maison ;
- Ce qu'on produit : exemple, une bibliothèque que l'on a publiée sur GitHub, ou un projet OSS que l'on développe ou auquel on contribue régulièrement ;
- Ce qu'on héberge : l'open source que l'on fait tourner pour fournir un service interne comme un CRM, GitLab, Nexus, etc. Un exemple de tableau pourrait ressembler à ceci :

Utilisé	Intégré	Produit	Hébergé	Compétences
Firefox, LibreOffice, Postgresql	Bibliothèque slf4j	Bibliothèque YY sur GH	GitLab, Nexus	Java, Python

La même identification doit s'appliquer aux compétences et expérience :

- Disponibles au sein des équipes
- Pouvant être développées ou acquises en interne (formation, coaching, expérimentation)
- Nécessitant d'être obtenues via le marché, ou des partenariats / contrats.

### Recommandations

- Faire les choses simplement ;
- Il s'agit d'un exercice de niveau relativement général, pas d'un inventaire détaillé pour la comptabilité ;
- Bien que cette activité soit un bon point de départ, inutile de la terminer à 100% avant de lancer d'autres activités ;
- S'occuper des problèmes, ressources et compétences liés au **développement logiciel** fait partie de l'activité « Montée en compétences » (4.2) ;
- L'inventaire doit couvrir tous les domaines de l'IT : systèmes d'exploitation, intergiciels, bases de données, administration système, outils de développement et de tests, etc. ;
- Commencer à identifier les communautés pertinentes : il est plus facile d'obtenir du support et des retours concernant un projet lorsque sa communauté vous connaît déjà.

### Ressources

- Un excellent cours sur [Le logiciel Libre et Open Source \(FOSS\)](#), par le Professeur Dirk Riehle.

## 4.2 Montée en compétences open source

Identifiant de l'activité : [GGI-A-18](#).

### Description

Cette activité vise à planifier et initier les capacités techniques et les premières expériences avec l'open source après réalisation d'un inventaire (voir plus haut). C'est aussi l'occasion de commencer à établir une feuille de route simplifiée de développement des compétences.

- Identifier les compétences et formations requises.
- Mettre en place un projet pilote pour démarrer, apprendre au fil de l'eau, définir un premier jalon de réussite.
- Capitaliser sur les leçons apprises pour construire un corpus de connaissances.
- Commencer à identifier et documenter les prochaines étapes vers une adoption plus large.
- Élaborer une stratégie pour les prochains mois ou l'année, afin d'obtenir engagements et support financier.

Domaine de l'activité :

- Linux, Apache, Debian, compétences en administration ;
- Bases de données open source MariaDB, MySQL, PostgreSQL, etc. ;
- Technologies open source de virtualisation et cloud ;
- Pile LAMP et alternatives.

## Évaluation de l'opportunité

Comme toute technologie de l'information, et sans doute encore plus, l'open source induit l'innovation. L'open source croît et évolue rapidement : il est nécessaire que l'organisation se maintienne à jour.

Cette activité aide à identifier les domaines où la formation peut rendre les gens plus efficaces et plus confiants quant à l'usage de l'open source. Elle aide à prendre des décisions concernant l'évolution des employés. Initier des compétences minimales dans l'open source permet d'évaluer l'opportunité de :

- Compléter des solutions IT avec des technologies existantes du marché, développées par l'écosystème ;
- Développer de nouveaux types de collaboration internes et externes à l'organisation ;
- S'approprier des compétences en matière de technologies nouvelles et innovantes.

## Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Une matrice de compétences est mise en place.
- Le domaine d'application des technologies OSS utilisées est défini de façon proactive, évitant ainsi leur usage incontrôlé.
- Un niveau d'expertise satisfaisant est acquis concernant ces technologies.
- Les équipes ont suivi une formation aux « bases de l'open source » pour démarrer.

## Outils

Un outil clé est la Matrice (ou Cartographie) d'Activité (ou Compétence(s)).

Cette activité peut être menée à bien par :

- l'utilisation de didacticiels en ligne (il en existe beaucoup de gratuits sur Internet) ;
- la participation à des conférences pour développeurs ;
- la formation professionnelle, etc.

## Recommandations

- Utiliser et développer des composants open source de manière sûre et efficace nécessite un état d'esprit ouvert et collectif, qu'il convient de reconnaître et propager tant de haut en bas (encadrement) que de bas en haut (développeurs).
- S'assurer que l'approche est activement supportée et promue par l'encadrement. Rien ne s'accomplira sans implication de la hiérarchie.
- Impliquer les gens (développeurs, parties prenantes) dans le processus : organiser des tables rondes et rester à l'écoute des idées.
- Laisser du temps et des ressources aux gens pour découvrir l'activité, tester et jouer avec ces nouveaux concepts. Si possible, de façon ludique : ludification et récompenses sont de bonnes incitations.

Un projet pilote avec les étapes suivantes peut servir de catalyseur :

- Identifier une technologie ou un framework pour démarrer ;
- Trouver de la formation en ligne, un didacticiel, des exemples de code pour expérimenter ;
- Construire un prototype de la solution attendue ;
- Identifier quelques experts pour challenger et guider l'implémentation.

## Ressources

- [Qu'est-ce qu'une Matrice de Compétences](#) : introduction rapide.
- [Comment construire une Matrice de Compétences pour votre équipe](#) : un modèle commenté.
- [MOOC sur la culture libre](#) (en français) : cours en 6 parties sur la culture libre, introduction au droit d'auteur, à la propriété intellectuelle, aux licences open source.



## 4.3 Supervision de l'open source

Identifiant de l'activité : [GGI-A-19](#).

### Description

Cette activité vise à contrôler l'usage de l'open source et garantir que le logiciel open source est géré de manière adéquate. Différentes perspectives sont concernées, que ce soit l'usage d'outils ou solutions métier open source, l'inclusion de composants open source dans les développements internes, la modification d'une version de logiciel pour l'adapter à ses besoins, etc. Il est aussi question d'identifier les domaines dans lesquels l'open source est devenu une solution de fait (parfois cachée), et d'y évaluer sa pertinence.

Il peut être nécessaire de clarifier les points suivants :

- La fonctionnalité requise est-elle fournie ?
- Y a-t-il des fonctionnalités complémentaires dont on n'avait pas besoin, et qui accroissent la complexité des phases d'assemblage (build) ou d'exécution ?
- Quelles sont les contraintes liées à la licence, et leurs implications légales ?
- À quel point la solution accroît-elle l'indépendance de votre organisation vis-à-vis de ses fournisseurs ?
- Existe-t-il une offre de support en phase avec vos besoins métier, et quel est son coût ?
- TCO (Coût Total de Possession) ;
- L'encadrement est-il informé des avantages de l'open source, du moins au-delà des « économies de coût de licence » ? Être à l'aise avec l'open source aide à tirer le meilleur profit de la coopération avec les communautés, projets et éditeurs ;
- Évaluer s'il est pertinent de partager les coûts de développement en reversant ses propres développements à la communauté, en tenant compte de toutes les implications, comme la conformité à la licence.
- Vérifier la disponibilité de support, communautaire ou professionnel.

### Évaluation de l'opportunité

Définir un processus de décision orienté vers l'open source est un moyen de maximiser ses profits.

- Cela évite l'émergence d'usages sans contrôle et de coûts cachés des technologies open source.
- Cela conduit à des décisions stratégiques et organisationnelles informées et conscientes quant à l'open source.

Coûts : l'activité peut remettre en question et conduire à reconsidérer l'usage de facto de l'open source comme sous-optimal, inefficace, risqué, etc.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette activité :

- L'open source est devenu un choix confortable quand il n'est plus perçu comme une exception ou un choix dangereux ;
- L'open source est devenu une option « mainstream » ;
- Les acteurs clés sont suffisamment convaincus que l'open source a des avantages stratégiques justifiant qu'on y investisse.
- Il est possible de démontrer que le TCO d'une solution basée sur l'open source est inférieur à celui d'alternatives propriétaires.
- Il est possible d'évaluer à quel degré le gain d'indépendance vis-à-vis des fournisseurs permet ou va permettre des économies ;
- Il est possible d'évaluer à quel degré l'indépendance qu'apporte la solution réduit les coûts potentiels d'un changement de solution (ouverture des formats).

### Outils

A ce stade, nous ne pensons à aucun outil pertinent ou concerné par cette activité.

## Recommandations

- Gérer de manière proactive l'usage de l'open source nécessite un niveau minimum de compréhension de ses fondamentaux, préalable à toute décision le concernant.
- Se concentrer sur les fonctionnalités nécessaires plutôt que la recherche d'une alternative à une solution propriétaire.
- S'assurer de la disponibilité de support et de la pérennité des développements futurs.
- Considérer les effets sur votre organisation de la licence de la solution.
- Convaincre les acteurs clés des avantages de l'open source, bien au-delà de l'« économie sur les coûts de licence ».
- Être honnête, en n'exagérant pas les bénéfices attendus de l'open source.
- Dans le processus de décision, il est important d'évaluer différentes solutions open source, pour éviter les désillusions liées à des attentes erronées, éclaircir ce que l'organisation devra faire, rendre perceptibles les avantages apportés par l'ouverture des solutions. L'organisation doit faire en sorte d'adapter cette évaluation à son contexte propre.

## Ressources

- [Les 5 principaux avantages de l'Open Source](#) : Blog sponsorisé mais intéressant, lecture courte.
- [Évaluer les coûts cachés de l'Open Source](#) : une analyse des coûts de support de l'OSS, sponsorisée par IBM.

## 4.4 Logiciel open source professionnel

Identifiant de l'activité : [GGI-A-20](#).

### Description

Cette activité vise au choix proactif de solutions OSS, commerciales ou communautaires, dans le domaine professionnel. Elle couvre également la définition de politiques préférentielles quant à la sélection de solutions métier open source.

Alors que le logiciel open source est le plus souvent utilisé par des professionnels de l'IT (systèmes d'exploitation, intergiciel, bases de données, administration système, outils de développement), il a encore besoin d'être reconnu dans les domaines où les professionnels métier sont prépondérants.

Alors que le logiciel open source est le plus souvent utilisé par des professionnels de l'IT (systèmes d'exploitation, intergiciel, bases de données, administration système, outils de développement), il a encore besoin d'être reconnu dans les domaines où les professionnels métier sont prépondérants.

### Évaluation de l'opportunité

À mesure que l'open source se généralise, il se répand bien au-delà des systèmes d'exploitation et outils de développement, trouvant ses usages dans les couches supérieures des systèmes d'information, jusqu'aux applications métier. Il est pertinent d'identifier quelles applications OSS sont utilisées avec succès pour répondre aux besoins de l'organisation, et comment elles peuvent y devenir un choix préférentiel en termes économiques.

L'activité peut induire des besoins en formation et des coûts de migration.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Il existe une liste des solutions OSS recommandées pour couvrir des besoins identifiés en matière d'applications métier.
- Une ébauche de politique de sélection des applications métier open source est définie.
- Les solutions métier propriétaires utilisées font l'objet d'évaluation par comparaison avec des équivalents OSS.

- Les processus d'approvisionnement et d'appel à proposition spécifient une préférence pour l'open source (si légalement possible).

### Outils

Outils pour cartographier le logiciel et les applications métier ?

Dans l'état actuel, nous ne connaissons pas d'outil pertinent lié à cette activité.

### Recommandations

- Discuter avec ses collègues, tirer les leçons de ce que font des organisations comparables à la vôtre.
- Visiter les événements industriels locaux pour y rechercher des solutions open source et du support professionnel.
- Essayer les versions et le support communautaires préalablement à la souscription de contrats de support payants.

### Ressources

- [Qu'est-ce que l'open source d'entreprise ?](#) : courte lecture à propos de l'open source pour l'entreprise.
- [101 Applications open source pour aider votre business à prospérer](#) : une liste indicative de solutions open source orientées business.

## 4.5 Gérer les compétences et les ressources en matière de développement de logiciels open source

Identifiant de l'activité : [GGI-A-42](#).

### Description

Cette activité est axée sur les compétences et les ressources de **développement de logiciels**. Elle comprend les technologies et les compétences de développement spécifiques des développeurs, ainsi que les processus, méthodes et outils de développement.

Une grande quantité de documentation, de forums et de discussions issus de l'écosystème, et de ressources publiques est disponible pour les technologies open source. Afin de bénéficier pleinement de leur approche open source, il faut établir une feuille de route de ses actifs actuels et des objectifs souhaités afin de mettre en place un programme cohérent pour les compétences, les méthodes et les outils de développement au sein des équipes.

**Domaines d'application** Il faut établir les domaines dans lesquels le programme sera appliqué, et comment il améliorera la qualité et l'efficacité du code et des pratiques. Par exemple, le programme n'aura pas les mêmes avantages s'il n'y a qu'un seul développeur travaillant sur des composants open source, ou si l'ensemble du cycle de vie du développement est optimisé pour inclure les meilleures pratiques open source.

Il faut définir le champ d'application du développement open source : composants techniques, applications, modernisation ou création de nouveaux développements. Voici quelques exemples de pratiques de développement qui peuvent bénéficier de l'open source :

- Administration du Cloud.
- Applications cloud-natives, comment innover avec ces technologies.
- DevOps, intégration continue / livraison continue.

### Catégories

- Compétences et ressources nécessaires au développement de logiciels open source : propriété intellectuelle, licences, pratiques.
- Compétences et ressources nécessaires pour développer des logiciels à l'aide de composants, de langages et de technologies open source.

- Compétences et ressources nécessaires pour utiliser les méthodes et processus de l'open source.

### Évaluation de l'opportunité

Les outils open source sont de plus en plus populaires parmi les développeurs. Cette Activité répond au besoin d'éviter la prolifération d'outils hétérogènes au sein d'une équipe de développement. Elle aide à définir une politique dans ce domaine. Elle permet d'optimiser la formation et l'acquisition d'expérience. Un inventaire des compétences est utilisé pour le recrutement, la formation et la planification de la succession au cas où un employé clé quitterait l'entreprise.

Nous aurions besoin d'une méthodologie pour cartographier les compétences en matière de développement de logiciels open source.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Voici une description de la chaîne de production des logiciels libres (la "chaîne d'approvisionnement en logiciels"),
- Il existe un plan (ou une liste de souhaits) pour la rationalisation des ressources de développement,
- Il existe un inventaire des compétences qui résume les compétences, la formation et l'expérience des développeurs actuels,
- Il existe une liste de souhaits en matière de formation et un programme visant à combler les lacunes en matière de compétences,
- Il existe une liste des meilleures pratiques manquantes en matière de développement de logiciels libres et un plan pour les appliquer.

### Recommandations

- Commencez simplement, développez régulièrement l'analyse et la feuille de route.
- Lors du recrutement, mettez l'accent sur les compétences et l'expérience en matière de logiciels libres. Il est toujours plus facile de former et de coacher des personnes ayant déjà un ADN open source.
- Consultez les programmes de formation des fournisseurs de logiciels et des écoles de logiciels libres.

### Ressources

Plus d'informations :

- Une introduction à [ce qu'est un inventaire des compétences](#) de Robert Tanner.
- Un article sur les compétences en matière de logiciels libres : [5 compétences en matière d'open source pour améliorer votre jeu et votre CV](#)

Cette activité peut inclure des ressources et des compétences techniques telles que :

- **Langages populaires** (comme Java, PHP, Perl, Python).
- **Frameworks open source** (Spring, AngularJS, Symfony) et outils de test.
- Méthodes de développement Agile, DevOps et open source **et meilleures pratiques**.

Activités associées :

- [GGI-A-28 perspective RH](#)

## 5 Activités de l'objectif « Confiance »

### 5.1 Gérer la conformité juridique

Activity ID: [GGI-A-21](#).

#### Description

Les organisations doivent mettre en œuvre un processus de conformité juridique pour sécuriser leur utilisation et leur participation aux projets open source.

La gestion mature et professionnelle de la conformité légale, au sein de l'organisation et tout au long de la chaîne d'approvisionnement, consiste à :

- Réalisation d'une analyse approfondie de la propriété intellectuelle qui comprend l'identification des licences et la vérification de la compatibilité.
- Garantir que l'organisation peut utiliser, intégrer, modifier et redistribuer en toute sécurité des composants open source dans le cadre de ses produits ou services.
- Fournir aux employés et aux contractants un processus transparent sur la façon de créer et de contribuer aux logiciels à code source ouvert.

*Analyse de la composition du logiciel (SCA)* : Une part importante des problèmes juridiques et de PI (propriété intellectuelle) résulte de l'utilisation de composants publiés sous des licences qui sont soit incompatibles entre elles, soit incompatibles avec la façon dont l'organisation souhaite utiliser et redistribuer les composants. Le SCA est la première étape de la résolution de ces problèmes, car « il faut connaître le problème pour le résoudre ». Le processus consiste à identifier tous les composants impliqués dans un projet dans un document de nomenclature, y compris les dépendances de construction (build) et de test.

*Vérification des licences* : Un processus de vérification des licences utilise un outil pour analyser automatiquement la base de code et identifier les licences et les droits d'auteur qu'elle contient. S'il est exécuté régulièrement et idéalement intégré aux chaînes de construction (build) et d'intégration continue, il permet de détecter rapidement les problèmes de propriété intellectuelle.

#### Évaluation de l'opportunité

Avec l'utilisation croissante des logiciels libres dans les systèmes d'information d'une organisation, il est essentiel d'évaluer et de gérer les risques juridiques potentiels.

Cependant, la vérification des licences et des droits d'auteur peut être délicate et coûteuse. Les développeurs doivent être en mesure de vérifier rapidement les questions juridiques et de PI. Disposer d'une équipe et d'un responsable dédiés à la propriété intellectuelle et aux questions juridiques garantit une gestion proactive et cohérente des questions juridiques, contribue à sécuriser l'utilisation et les contributions aux composants open source et fournit une vision stratégique claire.

#### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Un processus de vérification de licence facile à utiliser est disponible pour les projets ;
- Un processus de vérification de la PI facile à utiliser est disponible pour les projets ;
- Il existe une équipe ou une personne responsable de la conformité juridique au sein de l'organisation ;
- Des audits réguliers visant à évaluer la conformité juridique sont prévus.

Autres moyens de mettre en place des points de contrôle :

- Il existe un processus de vérification des licences facile à utiliser ;
- Il existe une équipe juridique/PI pouvant être consultée ;
- Tous les projets fournissent les informations nécessaires pour que les personnes puissent utiliser le projet et y contribuer ;
- Il y a un contact dans l'équipe pour les questions liées à la PI et aux licences ;
- Il y a un responsable de l'entreprise qui se consacre à la PI et aux licences ;

- Il existe une équipe dédiée aux questions relatives à la PI et aux licences.

### Outils

- [ScanCode](#)
- [Fossology](#)
- [SW360](#)
- [Fossa](#)
- [OSS Review Toolkit](#)

### Recommandations

- Informer les gens sur les risques associés aux licences en conflit avec les objectifs commerciaux ;
- Proposer une solution facile pour que les projets puissent mettre en place une vérification des licences sur leur base de code ;
- Communiquer sur son importance et aider les projets à l'ajouter à leurs systèmes d'intégration continue ;
- Fournir un modèle ou des directives officielles de structure d'un projet.
- Mettre en œuvre des vérifications automatisées pour garantir que chaque projet est conforme aux directives.
- Envisager de réaliser un audit interne pour identifier les licences de l'infrastructure de l'entreprise ;
- Fournir une formation de base sur la PI et les licences à au moins une personne par équipe ;
- Fournir une formation complète sur la PI et les licences au responsable ;
- Mettre en place un processus de remontée des problèmes de PI et de licence vers le responsable.

N'oubliez pas que la conformité n'est pas seulement une question de droit, mais aussi de PI (propriété intellectuelle). Voici donc quelques questions pour vous aider à comprendre les conséquences de la conformité juridique :

- Si je distribue un composant open source et que je ne respecte pas les conditions de la licence, j'enfreins la licence → incidences juridiques ;
- Si j'utilise un composant open source au sein d'un projet que je souhaite distribuer/publier, cette licence peut obliger à une visibilité sur des éléments de code que je ne souhaite pas rendre open source → Impact de la confidentialité pour l'avantage tactique de mon entreprise et avec des tiers (implications légales) ;
- Il s'agit d'une discussion ouverte sur la question de savoir si l'utilisation d'une licence open source pour un projet que je veux publier octroie une PI pertinente → incidences en matière de PI. ;
- Si je publie un projet en open source *avant* tout processus de brevet, cela exclut *probablement* la création de brevets concernant le projet → incidences en matière de PI ;
- Si je publie un projet en open source *après* tout processus de brevet, cela permet *probablement* la création de brevets (défensifs) concernant ce projet → potentiel de PI ;
- Dans les projets complexes qui font intervenir de nombreux composants avec de nombreuses dépendances, la multitude de licences open source peut présenter des incompatibilités entre les licences → incidences juridiques.

### Ressources

- Il existe une large liste d'outils sur la page [Outils de conformité OSS existants](#).
- [Pratiques recommandées de conformité des logiciels libres pour l'entreprise](#). Un livre d'Ibrahim Haddad, de la Fondation Linux, sur les pratiques de conformité des logiciels libres pour l'entreprise. [OpenChain Project](#).

## 5.2 Gérer les vulnérabilités des logiciels

Activity ID: [GGI-A-22](#).

## Description

Le code d'un développeur est autant sécurisé que la partie la moins sécurisée de son code. Des cas récents (par exemple, [heartbleed<sup>1</sup>](#), [equifax\[ ^equifax\]](#)) ont démontré l'importance de vérifier les vulnérabilités dans les parties du code qui ne sont pas directement développées en interne. Les conséquences des expositions vont des fuites de données (avec un impact considérable sur la réputation) aux attaques par des logiciels malveillants (ransomware) et à l'indisponibilité de services menaçant l'activité.

Les logiciels libres sont connus pour leur meilleure gestion des vulnérabilités que les logiciels propriétaires, principalement pour les raisons suivantes :

- Davantage de personnes cherchent à trouver et à corriger les problèmes sur le code et les processus ouverts ;
- Les projets open source corrigent les vulnérabilités et publient des correctifs et de nouvelles versions beaucoup plus rapidement.

Par exemple, une [étude de WhiteSource](#) sur les logiciels propriétaires a montré que 95 % des vulnérabilités trouvées dans leurs composants open source avaient déjà fait l'objet d'un correctif au moment de l'analyse. L'enjeu est donc de **mieux gérer les vulnérabilités à la fois dans la base de code et dans ses dépendances**, qu'elles soient fermées ou open source.

Afin d'atténuer ces risques, il faut mettre en place un programme d'évaluation de ses actifs logiciels et un processus de vérification des vulnérabilités exécuté régulièrement. Mettez en place des outils qui alertent les équipes concernées, gèrent les vulnérabilités connues et préviennent les menaces provenant des dépendances logicielles.

## Évaluation de l'opportunité

Toute entreprise qui utilise des logiciels doit surveiller ses vulnérabilités dans :

- son infrastructure (par exemple, l'infrastructure du Cloud, l'infrastructure réseau, les magasins de données) ;
- ses applications métier (RH, outils CRM, gestion des données internes et relatives aux clients) ;
- son code interne : par exemple, le site web de l'entreprise, les projets de développement internes, etc.,
- et toutes les dépendances directes et indirectes des logiciels et services.

Le retour sur investissement des vulnérabilités est peu connu jusqu'à ce que quelque chose de grave se produise. Il faut envisager les conséquences d'une violation majeure des données ou de l'indisponibilité de services pour estimer le coût réel des vulnérabilités.

De même, il faut à tout prix éviter une culture du secret et de la dissimulation des problèmes de sécurité au sein de l'entreprise. Au contraire, les informations sur l'état de la vulnérabilité doivent être partagées et discutées afin de trouver les meilleures réponses auprès des bonnes personnes, allant des développeurs aux cadres dirigeants.

Les avantages de la prévention des cyberattaques par une gestion attentive des vulnérabilités logicielles sont multiples :

- Éviter les risques pour la réputation ;
- Éviter les pertes d'exploitation (DDoS, Ransomware, temps de reconstruction d'un système informatique alternatif après une attaque) ;
- Respecter les règles de protection des données.

La gestion des vulnérabilités des logiciels open source n'est qu'une partie du processus plus large de cybersécurité qui traite globalement de la sécurité des systèmes et des services de l'organisation.

## Suivi de l'avancement

Il doit y avoir une personne ou une équipe dédiée à la surveillance des vulnérabilités et des processus, facile à joindre, sur laquelle les développeurs peuvent s'appuyer. L'évaluation des

---

<sup>1</sup><https://fr.wikipedia.org/wiki/Heartbleed>

vulnérabilités est une partie standard du processus d'intégration continue, et les personnes sont en mesure de surveiller l'état actuel des risques dans un tableau de bord dédié.

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Tous les logiciels et services internes sont évalués et surveillés pour détecter les vulnérabilités connues ;
- Un outil et un processus dédiés sont mis en œuvre dans la chaîne de production des logiciels pour empêcher l'introduction de brèches de sécurité dans les activités quotidiennes de développement ;
- Une personne ou une équipe est chargée d'évaluer le risque lié aux CVE/vulnérabilités par rapport à l'exposition aux risques ;
- Une personne ou une équipe est responsable de l'envoi des CVE/vulnérabilités aux personnes concernées (SysOps, DevOps, développeurs, etc.).

### Outils

- Outils GitHub
  - GitHub fournit des directives et des outils pour sécuriser le code hébergé sur la plateforme. Voir la [documentation de GitHub](#) pour plus d'informations ;
  - GitHub fournit [Dependabot](#) pour identifier automatiquement les vulnérabilités dans les dépendances ;
- [Eclipse Steady](#) est un outil gratuit et open source qui analyse les vulnérabilités des projets Java et Python et aide les développeurs à les atténuer ;
- [OWASP dependency-check](#) : un scanner de vulnérabilités open source ;
- [OSS Review Toolkit](#) : un orchestrateur open source capable de collecter des avis de sécurité pour un projet et ses dépendances, à partir de différentes sources de vulnérabilités.

### Ressources

- La [base de données des vulnérabilités CVE du MITRE](#). Voir aussi la [base de données de sécurité des NVD du NIST](#), et les ressources satellites comme [CVE Details](#).
- Consultez également cette nouvelle initiative de Google : [les Vulnérabilités open source](#).
- Le groupe de travail OWASP publie sur son site web une [liste de scanners de vulnérabilités](#), aussi bien du monde commercial que du monde open source.
- J. Williams et A. Dabirsiaghi. La malheureuse réalité des bibliothèques non sécurisées, 2012 (« The unfortunate reality of insecure libraries »).
- [Detection, assessment and mitigation of vulnerabilities in open source dependencies](#), Serena Elisa Ponta, Henrik Plate & Antonino Sabetta, Empirical Software Engineering volume 25, pages 3175-3215 (2020).
- [A Manually-Curated Dataset of Fixes to Vulnerabilities of open source Software](#), Serena E. Ponta, Henrik Plate, Antonino Sabetta, Michele Bezzi, Cédric Dangremont. Il existe également une [boîte à outils en cours de développement pour mettre en œuvre l'ensemble de données susmentionné](#).

## 5.3 Gérer les dépendances logicielles

Activity ID: [GGI-A-23](#).

### Description

Un programme *d'identification des dépendances* recherche les dépendances réellement utilisées dans la base de code. Par conséquent, l'organisation doit établir et maintenir une liste de dépendances connues pour sa base de code et surveiller l'évolution des fournisseurs identifiés.

L'établissement et la mise à jour d'une liste de dépendances connues est un outil et un pré-requis pour :

- La vérification de la propriété intellectuelle et des licences : certaines licences ne peuvent pas être mélangées, même en tant que dépendance. Il faut connaître ses dépendances pour évaluer les risques juridiques associés.



- Gestion des vulnérabilités : l'ensemble d'un logiciel est aussi faible que sa partie la plus faible : voir l'exemple de la faille [Heartbleed](#). Il faut connaître ses dépendances pour évaluer les risques de sécurité associés ;
- Cycle de vie et durabilité : une communauté active sur un projet dont on dépend est un bon signe pour les corrections de bugs, les optimisations et les nouvelles fonctionnalités ;
- Sélection réfléchie des dépendances utilisées, selon des critères de « maturité » — l'objectif étant d'utiliser des composants open source sûrs, avec une base de code saine et bien maintenue, et une communauté vivante, active et réactive qui acceptera des contributions externes, etc.

### Évaluation de l'opportunité

L'identification et le suivi des dépendances est une étape nécessaire pour atténuer les risques associés à toute réutilisation de code. En outre, la mise en œuvre d'outils et de processus pour gérer les dépendances logicielles est une condition préalable pour gérer correctement la qualité, la conformité et la sécurité.

Considérez les questions suivantes :

- Quel est le risque pour l'entreprise (coût, réputation, etc.) si le logiciel est corrompu, attaqué ou poursuivi en justice ?
- La base de code est-elle considérée comme critique pour les personnes, l'organisation ou l'entreprise ?
- Que se passe-t-il si un composant dont dépend une application change de dépôt logiciel ?

La première étape, minimale, consiste à mettre en place un outil d'analyse de la composition du logiciel (SCA). Le soutien de sociétés de conseil spécialisées peut être nécessaire pour une analyse complète de la composition du logiciel ou une cartographie des dépendances.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Les dépendances sont identifiées dans tout le code développé en interne ;
- Les dépendances sont identifiées dans tout le code externe exécuté au sein de l'organisation ;
- Une procédure d'analyse de la composition du logiciel (SCA) ou d'identification des dépendances, facile à mettre en place, est disponible pour que les projets l'ajoutent à leur processus d'intégration continue ;
- Des outils d'analyse des dépendances sont utilisés.

### Outils

- [OWASP Dependency check](#) : outil d'analyse de composition logicielle (SCA) qui tente de détecter les vulnérabilités publiquement divulguées contenues dans les dépendances d'un projet ;
- [OSS Review Toolkit](#) : une suite d'outils pour aider à l'examen des dépendances des logiciels Open Source ;
- [Fossa](#) : analyse rapide, portable et fiable des dépendances. Supporte l'analyse des licences et des vulnérabilités. Ne dépend pas du langage, s'intègre à plus de 20 systèmes de build ;
- [Software 360](#).
- [Outil de licence Eclipse Dash](#) : prend une liste de dépendances et demande à [ClearlyDefined](#) de vérifier leurs licences.
- [Le projet FOSSology](#) : FOSSology est un projet open source dont la mission est de faire progresser la conformité aux licences open source.

### Recommandations

- Effectuer des audits réguliers sur les dépendances et les exigences de la propriété intellectuelle pour atténuer les risques juridiques ;
- Idéalement, intégrer la gestion des dépendances dans le processus d'intégration continue afin que les problèmes (nouvelle dépendance, incompatibilité de licence) soient identifiés et corrigés dès que possible ;

- Garder la trace des vulnérabilités liées aux dépendances, tenir les utilisateurs et les développeurs informés ;
- Informer les gens des risques associés à une mauvaise licence ;
- Proposer une solution facile pour que les projets puissent mettre en place une vérification des licences sur leur base de code ;
- Communiquer sur son importance et aider les projets à l'ajouter à leurs systèmes d'intégration continue ;
- Mettre en place un KPI visible pour les risques liés aux dépendances.

### Ressources

- Page du groupe des [outils de vérification de la licence des logiciels libres existants](#) ;
- [Conformité des licences des logiciels libres et open source : Outils pour l'analyse de la composition des logiciels](#), par Philippe Ombredanne, nexB Inc ;
- [Modèle de maturité de la durabilité des logiciels](#) ;
- [CHAOSS](#) : logiciel libre d'analyse de la santé communautaire ;

## 5.4 Gérer les indicateurs clés

Activity ID: [GGI-A-24](#).

### Description

Cette activité permet de collecter et de surveiller un ensemble d'indicateurs qui éclairent les décisions de gestion quotidiennes et les options stratégiques concernant les logiciels libres gérés de manière professionnelle.

Les métriques clés relatives aux logiciels libres constituent la toile de fond de l'efficacité de la mise en œuvre des programmes de gouvernance. L'activité couvre la sélection de quelques indicateurs, leur publication aux équipes et à la direction, et l'envoi de mises à jour régulières sur l'initiative, par exemple via une lettre d'information ou des actualités de l'entreprise.

Cette activité nécessite :

- que les parties prenantes discutent et définissent les objectifs du programme ;
- la mise en place d'un outil de mesure et de collecte de données connecté à l'infrastructure de développement ;
- la publication d'au moins un tableau de bord pour les parties prenantes et pour toutes les personnes impliquées dans l'initiative.

Les indicateurs sont basés sur des données qui doivent être collectées auprès de sources pertinentes. Il existe par chance de très nombreuses sources dans le domaine du génie logiciel libre. En voici quelques exemples :

- l'environnement de développement, la chaîne de production CI/CD ;
- le département des ressources humaines ;
- les outils de test et d'analyse de la composition du logiciel ;
- les dépôts logiciels.

Exemples d'indicateurs :

- Nombre de dépendances résolues, affichées par type de licence ;
- Nombre de dépendances périmées/vulnérables ;
- Nombre de problèmes de licence/PI (propriété intellectuelle) détectés ;
- Contributions apportées à des projets externes ;
- Durée d'ouverture des bugs ;
- Nombre de contributeurs sur un composant, nombre de commits, etc.

Cette activité consiste à définir ces exigences et besoins de mesure, et à mettre en place un tableau de bord qui montre de manière simple et efficace les principaux indicateurs du programme.

## Évaluation de l'opportunité

Les indicateurs clés permettent de comprendre et de mieux gérer les ressources consacrées aux logiciels libres, et de mesurer les résultats afin de communiquer efficacement et de récolter tous les bénéfices de l'investissement. En communiquant largement, davantage de personnes peuvent suivre l'initiative et se sentiront impliquées, ce qui en fait finalement une préoccupation et un objectif au niveau de l'organisation.

Si chaque activité est assortie de critères d'évaluation qui permettent de répondre aux questions sur les progrès réalisés, il est toujours nécessaire d'effectuer un suivi à l'aide de chiffres et d'indicateurs quantitatifs.

Qu'il s'agisse d'une petite start-up ou d'une grande entreprise internationale, les indicateurs clés permettent aux équipes de rester concentrées et de suivre les performances. Les indicateurs sont essentiels, car ils soutiennent la prise de décision et constituent la base du suivi des décisions déjà prises.

Avec des chiffres et des graphiques simples et pratiques, les membres de l'ensemble de l'organisation pourront suivre et synchroniser les efforts en matière d'open source, ce qui en fait une préoccupation et une action communes. Cela permet également aux différents acteurs de mieux entrer en piste, de contribuer au projet et d'en tirer les bénéfices globaux.

## Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Une liste de métriques / mesures et la manière de les collecter ont été établies ;
- Des outils pour collecter, stocker, traiter et afficher les indicateurs sont utilisés ;
- Il existe un tableau de bord généralisé, accessible à tous les participants, qui montre les progrès réalisés dans le cadre de l'initiative.

## Outils

- [GrimoireLab](#) de Bitergia ;
- Les outils génériques de BI (elasticsearch, grafana, visualisations R/Python...) conviennent également, lorsque les connecteurs appropriés sont configurés en fonction des objectifs définis.

## Recommandations

- Rédiger les objectifs et la feuille de route de la gouvernance open source ;
- Communiquer en interne sur les actions et le statut de l'initiative ;
- Impliquer les personnes dans la définition des KPI, afin de s'assurer :
  - qu'ils soient bien compris ;
  - qu'ils fournissent une vue complète des besoins ;
  - qu'ils soient pris en compte et suivis.
- Construire au moins un tableau de bord qui peut être affiché pour tout le monde (par exemple sur un écran dans la salle), avec des indicateurs essentiels pour montrer les progrès et la situation globale.

## Ressources

- La [communauté CHAOSS](#) a beaucoup de bonnes références et ressources liées aux indicateurs open source.
- Consultez les métriques pour les [attributs de projet](#) de la méthodologie OW2 [Market Readiness Levels](#).
- [Une nouvelle façon de mesurer l'ouverture : The Open Governance Index](#) par Liz Laffan est une lecture intéressante sur l'ouverture dans les projets open source.
- [Governance Indicators : A Users' Guide](#) est le guide de l'ONU sur les indicateurs de gouvernance. Bien qu'il s'applique à la démocratie, à la corruption et à la transparence des nations, les principes de base de la mesure et des indicateurs appliqués à la gouvernance valent la peine d'être lus.

## 5.5 Faire des revues de code

Activity ID: [GGI-A-44](#).

### Description

La revue de code est une tâche de routine qui implique une revue manuelle et/ou automatisée du code source d'une application avant la sortie d'un produit ou la livraison d'un projet au client. Dans le cas des logiciels libres, la revue de code ne se limite pas à la détection d'erreurs de manière opportuniste ; il s'agit d'une approche intégrée au développement collaboratif réalisée au niveau de l'équipe.

Les revues de code devraient s'appliquer au code développé en interne ainsi qu'au code réutilisé à partir de sources externes, car elles améliorent la confiance générale dans le code et renforcent l'appropriation. C'est également un excellent moyen d'améliorer les compétences et les connaissances globales au sein de l'équipe et de favoriser la collaboration entre les membres de l'équipe.

### Évaluation de l'opportunité

Les revues de code sont utiles chaque fois que l'organisation développe un logiciel ou réutilise des éléments de logiciel externes. Bien qu'il s'agisse d'une étape standard du processus d'ingénierie logicielle, les revues de code dans le contexte de l'open source apportent des avantages spécifiques tels que :

- Lors de la publication de code source interne, vérifier que les directives de qualité adéquates sont respectées.
- Lors de la contribution à un projet open source existant, vérifier que les directives du projet ciblé sont respectées.
- La documentation disponible publiquement est mise à jour en conséquence.

C'est également une excellente occasion de partager et de faire respecter certaines règles de conformité juridique de votre organisation, telles que :

- Ne jamais supprimer les en-têtes de licence ou les droits d'auteur existants trouvés dans le code open source réutilisé.
- Ne pas copier et coller du code source de Stack Overflow sans l'autorisation préalable de l'équipe juridique.
- Inclure la ligne de copyright adéquate lorsque c'est nécessaire.

Les revues de code apportent confiance et sécurité dans le code. Si les gens ne sont pas sûrs de la qualité ou des risques potentiels de l'utilisation d'un produit logiciel, ils devraient procéder à des examens du code et collecter les commentaires des pairs.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- La révision du code open source est reconnue comme une étape nécessaire.
- Des revues de code source ouvert sont planifiées (régulièrement ou à des moments critiques).
- Un processus pour mener des revues de code open source a été défini et accepté collectivement.
- Les revues de code open source sont une partie standard du processus de développement.

### Recommandations

- La revue de code est une tâche collective qui fonctionne mieux dans un bon environnement collaboratif.
- N'hésitez pas à utiliser les outils et les modèles existants dans le monde du logiciel libre, où les revues de code sont un standard depuis des années (décennies).

**Ressources**

- [Qu'est-ce qu'une Revue de Code ?](#) : une lecture didactique sur la revue de code trouvée sur la bibliothèque de pratiques ouvertes de Red Hat.
- [Bonnes pratiques de Revue de Code](#) : une autre perspective intéressante sur ce qu'est la revue de code.

## 6 Activités de l'objectif Culture

### 6.1 Promouvoir les bonnes pratiques de développement open source

Activity ID: [GGI-A-25](#).

#### Description

Cette activité consiste à définir, promouvoir activement et mettre en œuvre les bonnes pratiques open source au sein des équipes de développement.

Comme point de départ, les sujets suivants pourraient faire l'objet d'une attention particulière :

- Documentation pour les utilisateurs et les développeurs.
- Organisation correcte du projet sur un dépôt accessible au public.
- Promouvoir et mettre en œuvre une réutilisation contrôlée.
- Fournir une documentation complète et à jour du produit.
- Gestion de la configuration : flux de travail git, modèles de collaboration.
- Gestion des versions : publier tôt et souvent, des versions stables vs des versions de développement, etc.

Les projets OSS ont un modus operandi particulier, de type [bazar](#). Afin de permettre et d'encourager cette collaboration et cet état d'esprit, certaines pratiques sont recommandées pour faciliter le développement collaboratif et décentralisé et les contributions de développeurs tiers...

**Documents communautaires** Assurez-vous que tous les projets au sein de l'entreprise proposent les documents suivants :

- README - description rapide du projet, comment interagir, liens vers des ressources.
- Contribuer (« CONTRIBUTING ») - introduction pour les personnes souhaitant contribuer.
- Code de conduite (« CODE\_OF\_CONDUCT ») - ce qui est acceptable ou non comme comportement au sein de la communauté.
- LICENSE - la licence par défaut du référentiel.

**Les bonnes pratiques de REUTILISATION** [REUSE](#) est une initiative de la [Free Software Foundation Europe](#) pour améliorer la réutilisation des logiciels et rationaliser la conformité des licences open source.

#### Évaluation de l'opportunité

Bien que cela dépende fortement de la connaissance commune du logiciel libre au sein de l'équipe, la formation des personnes et la création de processus qui consolident ces pratiques sont toujours bénéfiques. C'est encore plus important lorsque :

- les utilisateurs et contributeurs potentiels ne sont pas connus,
- les développeurs ne sont pas habitués au développement open source.

#### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Le projet établit une liste des bonnes pratiques de l'open source à respecter.
- Le projet surveille son alignement sur les bonnes pratiques.
- L'équipe de développement a été sensibilisée au respect des bonnes pratiques du logiciel libre.
- De nouvelles bonnes pratiques sont régulièrement évaluées, et un effort est fait pour les mettre en œuvre.

## Outils

- L'outil [REUSE helper tool](#) aide à rendre un dépôt conforme aux bonnes pratiques [REUSE](#). Il peut être inclus dans de nombreux processus de développement pour confirmer le statut actuel.
- [ScanCode](#) a la capacité de lister tous les documents communautaires et légaux dans le dépôt : voir la [description de ses fonctionnalités](#).
- GitHub dispose d'une fonctionnalité intéressante permettant de [vérifier les documents communautaires manquants](#). Elle se trouve sur la page du dépôt > « Insights » > « Community ».

## Recommandations

- La liste des bonnes pratiques dépend du contexte et du domaine du programme et devrait être réévaluée régulièrement dans une optique d'amélioration continue. Les pratiques doivent être surveillées et évaluées régulièrement pour suivre les progrès.
- Former les gens à la réutilisation des logiciels libres (en tant que consommateurs) et aux écosystèmes (en tant que contributeurs).
- Mettre en œuvre [REUSE.software](#).
- Mettre en place un processus de gestion des risques juridiques associés à la réutilisation et aux contributions.
- Encourager explicitement les gens à contribuer à des projets externes.
- Fournir un modèle ou des directives officielles de structure d'un projet.
- Mettre en œuvre des vérifications automatisées pour garantir que chaque projet est conforme aux directives.

## Ressources

- [Liste des bonnes pratiques open source par OW2](#), tirée de la méthodologie Market Readiness Levels.
- [Site officiel de REUSE](#) avec spécifications, tutoriel, FAQ.
- [Directives communautaires de GitHub](#).
- Un exemple de [bonnes pratiques en gestion de configuration avec GitHub](#).

## 6.2 Contribuer aux projets open source

Activity ID: [GGI-A-26](#).

### Description

Contribuer à des projets open source utilisés librement est un principe clé de la bonne gouvernance. Il s'agit d'éviter d'être un simple consommateur passif et de contribuer en retour aux projets. Lorsque les gens ajoutent une fonctionnalité ou corrigent un bug pour leur propre usage, ils doivent faire en sorte que cela soit suffisamment générique pour contribuer au projet. Les développeurs doivent avoir le temps de contribuer.

Cette activité couvre les domaines suivants :

- Travailler avec les projets open source "upstream".
- Signaler les bugs et les demandes de fonctionnalités.
- Contribuer au code et à la correction des bugs.
- Participer aux listes de diffusion de la communauté.
- Partage d'expérience.

### Évaluation de l'opportunité

Les principaux avantages de cette activité sont :

- augmenter la connaissance générale et l'engagement envers l'open source au sein de l'organisation : à mesure que les gens commencent à contribuer et à s'impliquer dans des projets open source, ils ont un sentiment d'utilité publique et améliorent leur réputation personnelle.

- développer la visibilité et la réputation de l'organisation au fur et à mesure que les contributions font leur chemin dans le projet. Cela montre que l'organisation est réellement impliquée dans l'open source, contribue en retour et promeut l'équité et la transparence.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Il existe une voie claire et officielle pour les personnes souhaitant contribuer.
- Les développeurs sont encouragés à contribuer aux projets open source qu'ils utilisent.
- Un processus est en place pour assurer la conformité juridique et la sécurité des contributions des développeurs.
- KPI : Volume des contributions externes (code, listes de diffusion, problèmes...) par individu, équipe ou entité.

### Outils

Il peut être utile de suivre les contributions, à la fois pour garder une trace de ce qui est contribué et pour pouvoir communiquer sur l'effort de l'entreprise. Des tableaux de bord et des logiciels de suivi d'activité peuvent être utilisés à cette fin. Notamment :

- [GrimoireLab](#) de Bitergia
- [ScanCode](#)

### Recommandations

Encourager les personnes au sein de l'entité à contribuer à des projets externes, en :

- Leur donnant le temps d'écrire des corrections de bugs et fonctionnalités génériques bien testées, et de les reverser à la communauté.
- Fournissant une formation aux personnes sur la contribution aux communautés open source. Cela concerne à la fois les compétences techniques (améliorer les connaissances de votre équipe) et la communauté (appartenance aux communautés open source, code de conduite, etc.).
- Offrir des formations sur les questions juridiques, de propriété intellectuelle et techniques, et mettre en place un contact en interne pour aider sur ces sujets si les gens ont des doutes.
- Prévoir des incitations pour les travaux publiés.
- Notez que les contributions de l'entreprise/entité reflèteront la qualité de son code et son implication, donc assurez vous que votre équipe de développement fournit un code suffisamment bon.

### Ressources

- L'initiative [CHAOSS](#) de la Fondation Linux propose des outils et des conseils sur la manière de suivre les contributions dans le développement.

## 6.3 Appartenir à la communauté open source

Activity ID: [GGI-A-27](#).

### Description

Cette activité vise à accroître chez les développeurs le sentiment d'appartenance à la grande communauté open source. Comme pour toute communauté, les personnes et les entités doivent participer et contribuer en retour à l'ensemble. Cela renforce les liens entre les praticiens et apporte de la durabilité et de l'activité à l'écosystème. D'un point de vue plus technique, cela permet de choisir les priorités et la feuille de route des projets, d'améliorer le niveau de connaissances générales et de compréhension technique.

Cette activité couvre les points suivants :



- **Identifier les événements** qui valent la peine d'être suivis. La mise en relation de personnes, l'apprentissage de nouvelles technologies et la création d'un réseau sont des facteurs clés pour profiter pleinement des avantages de l'open source.
- Envisagez **l'adhésion à des fondations**. Les fondations et organisations open source sont un élément clé de l'écosystème open source. Elles fournissent des ressources techniques et organisationnelles aux projets, et constituent un bon endroit neutre où les adhérents peuvent discuter de problèmes et de solutions communes, ou travailler sur des normes.
- Surveillez les **groupes de travail**. Les groupes de travail sont des espaces de collaboration neutres où les experts interagissent sur un domaine spécifique comme l'IoT, la modélisation ou la science. Ils constituent un mécanisme très efficace et rentable pour aborder ensemble des préoccupations générales, bien que spécifiques à un domaine.
- **Participation au budget**. En dernier ressort, l'argent est le catalyseur. Planifiez les dépenses nécessaires, accordez du temps rémunéré aux personnes pour ces activités, anticipez les prochaines actions, afin que le programme ne doive pas s'arrêter après quelques mois par manque de financement.

### Évaluation de l'opportunité

L'open source fonctionne mieux lorsqu'il est réalisé en relation avec la communauté open source au sens large. Cela facilite la correction des bugs, le partage des solutions, etc.

C'est aussi un bon moyen pour les entreprises de montrer leur soutien aux valeurs de l'open source. Communiquer sur l'implication de l'entreprise est important à la fois pour la réputation de l'entreprise et pour l'écosystème open-source.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Une liste d'événements auxquels les gens pourraient assister est rédigée.
- Il existe un suivi des conférences publiques données par les membres de l'équipe.
- Les personnes peuvent soumettre des demandes de participation à des événements.
- Les personnes peuvent soumettre des projets de sponsoring / adhésion.

### Recommandations

- Sonder les personnes pour savoir quels événements les intéressent ou seraient les plus bénéfiques pour leur travail.
- Mettre en place une communication interne (bulletin d'information, centre de ressources, invitations...) pour que les gens soient au courant des initiatives et puissent y participer.
- Assurez-vous que ces initiatives peuvent bénéficier à différents types de personnes (développeurs, administrateurs, support...), et pas seulement aux cadres dirigeants.

### Ressources

- [Qu'est-ce qui motive un développeur à contribuer à un logiciel open source ?](#) Un article de Michael Sweeney sur clearcode.cc.
- [Pourquoi les entreprises contribuent à l'open source](#) : Un article de Velichka Atanasova de VMWare.
- [Pourquoi vos employés devraient contribuer à l'open source](#) : Une bonne lecture par Robert Kowalski de CloudBees.
- [7 façons dont votre entreprise peut soutenir l'open source](#) : Un article de Simon Phipps pour InfoWorld.
- [Les événements : force vitale de l'open source](#) : Un article de Donna Benjamin de RedHat.

## 6.4 Point de vue des RH

Activity ID: [GGI-A-28](#).

## Description

Le passage à la culture open source a de profondes répercussions sur les RH :

- **Nouveaux processus et contrats** : Les contrats doivent être adaptés pour permettre et promouvoir les contributions externes. Cela inclut les questions de propriété intellectuelle et de licence pour le travail effectué au sein de l'entreprise, mais aussi la possibilité pour l'employé ou le contractant d'avoir ses propres projets.
- **Différents types de personnes** : Les personnes travaillant dans le domaine de l'open source ont souvent des motivations et des mentalités différentes de celles des entreprises propriétaires. Les processus et les mentalités doivent s'adapter à ce paradigme axé sur la réputation de la communauté, afin d'attirer de nouveaux types de talents et de les retenir.
- **Développement de carrière** : il est nécessaire d'offrir un plan de carrière qui encourage et valorise les employés pour leurs compétences techniques et non techniques ainsi que pour les compétences attendues par votre organisation (collaboration pour diriger les efforts communautaires, communication pour agir en tant que porte-parole de votre entreprise, etc.). Les RH ont un rôle clé à jouer pour faire de l'open source un objectif culturel.

**Main d'œuvre** Pour un développeur qui travaille depuis longtemps sur la même solution propriétaire, passer à l'open source peut sembler un changement important et nécessiter une adaptation. Mais pour la plupart des développeurs, les logiciels open source n'apportent que des avantages.

Les développeurs qui sortent aujourd'hui de l'école ou de l'université ont tous travaillé sur des logiciels libres. Au sein d'une entreprise, la grande majorité des développeurs utilisent des langages open source et importent quotidiennement des bibliothèques ou des extraits de code open source. Il est en effet beaucoup plus facile de coller des lignes de code open source dans un programme que de déclencher le processus de sourcing interne, qui passe par de multiples validations à travers la ligne hiérarchique.

L'open source rend le travail du développeur plus intéressant car avec l'open source, un développeur est toujours à l'affût de ce qui est inventé par ses pairs hors de l'entreprise. Il reste donc à la pointe de la technologie.

Pour une organisation, il est nécessaire d'avoir une stratégie RH pour 1/ qualifier ou requalifier la main d'œuvre existante 2/ positionner l'entreprise sur le recrutement de nouveaux talents, en reflétant l'attractivité de l'entreprise par le biais de l'open source.

Attirer des personnes avec un bon état d'esprit open source, qui comprennent déjà le code et savent comment travailler avec les autres, est merveilleux. L'alternative d'évangélisation / formation / stage est intéressante mais plus coûteuse et plus longue.

— PDG d'un fournisseur de logiciels OSS

Cela illustre que l'embauche de personnes dotées d'un ADN open source est une voie d'accélération à prendre en compte dans la stratégie des RH.

## Processus

- Établir ou revoir les descriptions de poste (compétences techniques, compétences générales, expérience)
- Programmes de formation : auto-formation, formation formelle, coaching de la direction, lien avec les pairs, communautés
- Établir ou revoir le plan de carrière : compétences, résultats/impact et étapes clés de la carrière

## Évaluation de l'opportunité

1. Encadrer les pratiques de développement : le problème n'est probablement pas tant d'inciter les développeurs à utiliser davantage de logiciels libres, que de s'assurer qu'ils les utilisent en toute sécurité, en respectant les conditions de licence de chaque technologie libre, et sans abandonner les contrôles de sécurité traditionnels (les lignes de code libres peuvent contenir des codes malveillants),
2. Revoir les pratiques de collaboration : avec les pratiques de développement vient l'opportunité d'étendre l'agilité et la collaboration à d'autres secteurs d'activité de votre

organisation. Le sourcing interne est souvent utilisé pour favoriser ces comportements, bien qu'étant à mi-chemin de la culture open source,

3. Culture de l'organisation : en fin de compte, tout dépend de la culture de votre organisation. L'open source peut être le porte-drapeau de valeurs telles que l'ouverture, la collaboration, l'éthique et la durabilité.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Une formation est disponible pour présenter à la fois les avantages et les contraintes (conformité aux conditions de propriété intellectuelle de la licence) liés à l'open source.
- Chaque développeur, chaque architecte, chaque chef de projet (ou Product Owner/Business Owner) comprend les avantages et les contraintes (conformité aux conditions de propriété intellectuelle de la licence) liés à l'open source.
- Les développeurs sont encouragés à contribuer aux communautés open source, et à en assumer la responsabilité, et peuvent recevoir une formation adéquate pour le faire.
- Les aptitudes et les compétences sont reflétées dans les descriptions de poste et les étapes de carrière de l'organisation.
- L'expérience que les développeurs ont acquise dans le domaine de l'open source (contributions aux communautés open source, participation au processus de conformité interne, interventions externes pour l'entreprise...) est prise en compte dans le processus d'évaluation des RH.

### Outils

- Matrice de compétences.
- Programmes de formation publics (ex. open source school).
- Sourcing : GitHub, GitLab, LinkedIn, Meetups, Epitech, Epita ...
- Modèles de contrat (clause de loyauté).
- Descriptions de poste (modèles) et étapes de carrière (modèles).

### Recommandations

La plupart du temps, les développeurs connaissent déjà certains principes de l'open source et sont prêts à travailler avec et sur des logiciels open source. Cependant, il y a encore quelques actions que la direction devrait prendre :

- Privilégier l'expérience des logiciels libres à l'embauche, même si le poste pour lequel le développeur est recruté ne concerne que la technologie propriétaire. Il y a de fortes chances, avec la transformation numérique, que le développeur soit un jour amené à travailler sur de l'open source.
- Programme de formation aux logiciels libres : chaque développeur, chaque architecte, chaque chef de projet (ou Product Owner/Business Owner) devrait avoir accès à des ressources de formation (vidéos ou formations en présentiel) qui présentent les avantages de l'open source mais aussi les contraintes en termes de propriété intellectuelle et de conformité des licences.
- Des formations devraient être mises à disposition des développeurs qui souhaitent contribuer aux communautés open source et faire partie des organes de gouvernance de ces communautés (certifications Linux).
- Reconnaissance dans les processus d'évaluation personnelle des RH de la contribution de l'employé (développeur ou architecte) aux sujets liés à l'open source tels que les contributions aux communautés open source et la conformité aux conditions de propriété intellectuelle des licences. La plupart des sujets sont partagés et s'inscrivent dans les parcours de carrière techniques, tandis que certains pourraient ou devraient être spécifiques.
- Stratégie de communication et positionnement de l'entreprise : il faut aborder les aspects de communication (dans quelle mesure le sujet est essentiel pour votre organisation au point d'être reflété dans votre rapport annuel), comment cela affecte-t-il votre stratégie de communication (un contributeur open source pourrait être un porte-parole de votre entreprise, y compris pour les contacts avec la presse).

## Ressources

- En ce qui concerne la capacité des personnes à s'exprimer en dehors de l'entreprise lors d'événements, veuillez consulter l'activité « Affirmer publiquement l'utilisation de l'open source » de l'objectif « Engagement ».

## 6.5 Regarder vers l'amont (« Upstream first »)

Activity ID: [GGI-A-39](#).

### Description

Cette activité vise à faire prendre conscience des bénéfiques à contribuer en retour, donc à instaurer comme principe de regarder ou penser vers l'amont.

Dans cette optique, tout développement sur un projet open source doit être réalisé avec le niveau de qualité et d'ouverture requis pour être transmis aux développeurs principaux du projet, et rendu public par eux.

### Évaluation de l'opportunité

Écrire du code pensé pour une intégration amont a pour résultat :

- du code de meilleure qualité,
- du code prêt à être soumis au projet amont,
- du code fusionné au cœur du projet,
- du code qui sera compatible avec les versions futures,
- la reconnaissance par la communauté du projet, pour une coopération meilleure et plus profitable.

Regarder vers l'amont est bien plus qu'être simplement « bienveillant ». Ça signifie avoir son mot à dire dans le projet. Ça signifie contrôler la situation. Ça signifie agir plutôt que réagir. Ça signifie que vous comprenez l'open source. ([Maximilian Michels](#))

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression dans cette activité : la vision amont est-elle réelle ?

- Un accroissement significatif du nombre de « pull/merge requests » soumises à des projets tiers.
- L'existence d'une liste de projets tiers pour lesquels une démarche de contribution pourrait être envisagée.

### Recommandations

- Identifier les développeurs les plus expérimentés en matière d'interaction avec les développeurs amont.
- Faciliter l'interaction entre vos développeurs et les développeurs principaux des projets ciblés (événements, hackathons, etc.)

## Ressources

- Une explication claire du principe « Upstream First » et en quoi il s'intègre à l'objectif « Culture » : <https://maximilianmichels.com/2021/upstream-first/>.

« Upstream First » signifie qu'à chaque fois que vous résolvez dans votre copie locale un problème d'une manière pouvant profiter à d'autres, vous reversez vos évolutions au projet amont, par exemple en envoyant un patch ou en ouvrant une « pull request » dans le dépôt amont.

- [Que sont l'Amont et l'Aval dans le développement logiciel ?](#) : une explication limpide.
- Explications via les documents de conception de Chromium OS : [Upstream First](#).
- Red Hat, à propos des avantages de l'approche [upstream first](#).

## 7 Activités de l'objectif « Engagement »

### 7.1 S'engager dans des projets open source

Activity ID: [GGI-A-29](#).

#### Description

Cette activité consiste à engager des contributions significatives à certains projets OSS qui sont importants pour vous. Les contributions sont remontées et soumises au niveau de l'organisation (et non au niveau personnel comme dans #26). Elles peuvent prendre plusieurs formes, du financement direct à l'allocation de ressources (personnes, serveurs, infrastructure, communication, etc.), pour autant qu'elles bénéficient au projet ou à l'écosystème de manière durable et efficace.

Cette activité va au-delà de l'activité #26 pour faire émaner de l'organisation les contributions aux projets open source, les rendant plus visibles, puissantes et bénéfiques. Dans cette activité, les contributions sont censées apporter une amélioration substantielle et à long terme au projet OSS : par exemple, un développeur ou une équipe qui développe une nouvelle fonctionnalité très attendue, des actifs d'infrastructure, des serveurs pour un nouveau service, la reprise de la maintenance d'une branche largement utilisée.

L'idée est de réserver un pourcentage de ses ressources pour parrainer des développeurs de logiciels libres qui écrivent et maintiennent des bibliothèques ou des projets que l'on utilise.

Cette activité implique d'avoir une cartographie des logiciels open source utilisés, et une évaluation de leur criticité pour décider lequel soutenir.

#### Évaluation de l'opportunité

Si chaque entreprise utilisant des logiciels libres contribuait au moins un peu, nous aurions un écosystème sain. <https://news.ycombinator.com/item?id=25432248>

Soutenir des projets permet d'assurer leur pérennité et d'avoir accès à des informations, voire d'influencer et prioriser certains développements (bien que cela ne doive pas être la raison principale du soutien aux projets).

Avantages potentiels de cette activité : garantir que les rapports de bogues sont traités en priorité et que les développements sont intégrés dans la version stable. Coûts potentiels associés à cette activité : consacrer du temps aux projets, engagement financier.

#### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Projet bénéficiaire identifié.
- Option de soutien décidée, telle qu'une contribution monétaire directe ou une contribution au code.
- Nomination d'un chef de projet.
- Une certaine contribution a été apportée.
- Le résultat de la contribution a été évalué.

Points de vérification empruntés au questionnaire d'OpenChain [auto-certification](#) :

- Nous avons une politique de contribution aux projets open source au nom de l'organisation.
- Nous disposons d'une procédure documentée régissant les contributions aux projets open source.
- Nous disposons d'une procédure documentée pour sensibiliser tout le personnel du logiciel à la politique de contribution aux projets open source.

#### Outils

Certaines organisations proposent des mécanismes de financement de projets à code source ouvert (cela pourrait être pratique si votre projet cible figure dans leurs portefeuilles).

- [Collectif ouvert](#).

- [Software Freedom Conservancy](#).
- [Tidelift](#).

### Recommandations

- Concentrez-vous sur les projets essentiels pour l'organisation : ce sont les projets que vous souhaitez le plus aider par vos contributions.
- Ciblez les projets communautaires.
- Cette activité nécessite une connaissance minimale d'un projet cible.

### Ressources

- [Comment soutenir les projets open source maintenant](#) : une courte page avec des idées sur le financement des projets open source.
- [Sustain OSS : un espace pour les conversations sur le soutien de l'open source](#)

## 7.2 Soutenir les communautés open source

Activity ID: [GGI-A-30](#).

### Description

Cette activité consiste à s'engager auprès des représentants institutionnels du monde de l'open source.

Elle est réalisée par :

- L'adhésion aux fondations OSS (y compris le coût financier de l'adhésion).
- Le soutien et la défense des activités des fondations.

Cette activité implique d'allouer aux équipes IT et de développement du temps et un budget pour participer aux communautés open source.

### Évaluation de l'opportunité

Les communautés open source sont à l'avant-garde de l'évolution de l'écosystème open source. S'engager dans les communautés présente plusieurs avantages :

- cela permet de rester informé et à jour,
- le profil de l'organisation est amélioré,
- l'adhésion induit des bénéfices,
- cela fournit une structure et une motivation supplémentaires à l'équipe informatique open source.

Les coûts comprennent :

- les frais d'adhésion,
- le temps de personnel et un budget de voyage alloué pour participer aux activités de la communauté,
- le suivi des engagements en matière de propriété intellectuelle.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- L'organisation est un membre officiel d'une fondation open source.
- L'organisation participe à la gouvernance d'une fondation / communauté.
- Les logiciels développés par l'organisation sont soumis à la base de code d'une fondation ou ont été ajoutés à celle-ci.
- L'adhésion est reconnue sur les sites web de l'organisation et de la communauté.
- Une évaluation des coûts/avantages de l'adhésion a été réalisée.
- Un point de contact pour la communauté a été désigné.

**Recommandations**

- Rejoignez une communauté adaptée à votre taille et vos ressources, c'est-à-dire une communauté qui peut entendre votre voix et où vous pouvez être un contributeur reconnu.

**Ressources**

- Consultez cette [page utile \(en anglais\)](#) de la Fondation Linux sur le pourquoi et le comment de l'adhésion à une communauté open source.

**7.3 Afficher publiquement l'usage de l'open source**

Activity ID: [GGI-A-31](#).

**Description**

Cette activité consiste à reconnaître l'utilisation de logiciels libres dans un système d'information, dans des applications et dans de nouveaux produits.

- Fournir des exemples de réussite.
- Faire des présentations lors d'événements.
- Financer la participation à des événements.

**Évaluation de l'opportunité**

Il est maintenant généralement admis que la plupart des systèmes d'information fonctionnent avec des logiciels libres et que les nouvelles applications sont pour la plupart réalisées en réutilisant des logiciels libres.

Le principal avantage de cette activité est de créer un terrain de jeu équitable entre le logiciel libre et les logiciels propriétaires, de s'assurer que les logiciels libres reçoivent la même attention et sont gérés de manière aussi professionnelle que les logiciels propriétaires.

Un autre avantage est qu'elle contribue grandement à améliorer le profil de l'écosystème open source et, puisque les utilisateurs d'open source sont identifiés comme des « innovateurs », renforce également l'attractivité de l'organisation.

**Suivi de l'avancement**

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Les fournisseurs commerciaux de logiciels libres ont l'autorisation d'utiliser le nom de l'organisation comme référence client.
- Les contributeurs sont autorisés à le faire et à s'exprimer sous le nom de l'organisation.
- L'utilisation des logiciels libres est ouvertement mentionnée dans le rapport annuel du département informatique.
- Il n'y a aucun obstacle à ce que l'organisation explique son utilisation des logiciels open source dans les médias (interviews, événements open source et industriels, etc.).

**Recommandations**

- L'objectif de cette activité n'est pas que l'organisation devienne un centre d'activisme OSS, mais de s'assurer qu'il n'y a aucun obstacle à ce que le public reconnaisse son utilisation de l'open source.

**Ressources**

- Exemple du [CERN](#) affirmant publiquement son utilisation d'OpenStack.

**7.4 S'engager auprès des fournisseurs de logiciels libres**

Activity ID: [GGI-A-33](#).

## Description

Passez des contrats avec les vendeurs de logiciels libres qui fournissent les logiciels essentiels pour vous. Les entreprises et entités qui produisent des logiciels open source doivent prospérer pour assurer la maintenance et le développement de nouvelles fonctionnalités. Leur expertise spécifique est requise sur le projet, et la communauté d'utilisateurs s'appuie sur leur activité et leurs contributions continues.

L'engagement avec les vendeurs de logiciels libres prend plusieurs formes :

- Souscrire à des plans d'assistance / support.
- Passer des contrats avec des sociétés de services locales.
- Sponsoriser des développements.
- Payer pour une licence commerciale.

Cette activité implique de considérer les projets open source comme des produits à part entière qui méritent d'être achetés, tout comme les produits propriétaires, bien que généralement beaucoup moins chers.

## Évaluation de l'opportunité

L'objectif de cette activité est d'assurer un soutien professionnel aux logiciels libres utilisés dans l'organisation. Elle présente plusieurs avantages :

- Continuité du service grâce à des corrections de bogues en temps voulu.
- Performance du service grâce à une installation optimisée.
- Clarification du statut légal/commercial du logiciel utilisé.
- Accès rapide aux informations.
- Stabilité des prévisions budgétaires.

Le coût est évidemment celui des plans d'assistance choisis. Un autre coût pourrait être d'abandonner l'externalisation en masse auprès de grands intégrateurs de systèmes au profit d'une contractualisation fine avec des PME expertes.

## Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Les logiciels libres utilisés dans l'organisation bénéficient d'un support commercial.
- Des plans de support pour certains logiciels open source ont été contractualisés.
- Le coût des plans de support est une entrée légitime dans le budget informatique.

## Recommandations

- Dans la mesure du possible, trouvez des PME expertes locales.
- Méfiez-vous des grands intégrateurs de systèmes qui revendent l'expertise de tiers (en revendant des plans de support que les PME expertes en logiciels libres fournissent en réalité).

## Ressources

Quelques liens illustrant la réalité commerciale des logiciels libres :

- [Le point de vue d'un investisseur sur l'évolution des projets open source de la communauté vers l'entreprise.](#)
- [Une lecture rapide pour comprendre l'open source commercial.](#)

## 7.5 Politique d'achat de logiciels libres

Activity ID: [GGI-A-43](#).



## Description

Cette activité consiste à mettre en œuvre un processus de sélection, d'acquisition, d'achat de logiciels et de services open source. Il s'agit également de prendre en compte le coût réel des logiciels libres et de le provisionner. Les logiciels libres peuvent être « gratuits » à première vue, mais ils ne sont pas sans coûts internes et externes tels que l'intégration, la formation, la maintenance et le support.

Une telle politique exige que les solutions open source et propriétaires soient considérées de manière symétrique lors de l'évaluation de la valeur de l'investissement comme la combinaison optimale entre le coût total de possession et le rapport qualité-prix. Par conséquent, le département des achats informatiques doit considérer activement et équitablement les options open source, tout en s'assurant que les solutions propriétaires sont traitées sur un pied d'égalité dans les décisions d'achat.

La préférence pour l'open source peut être explicitement exprimée sur la base de la flexibilité intrinsèque de l'option open source lorsqu'il n'y a pas de différence de coût global significative entre les solutions propriétaires et open source.

Les services achat doivent comprendre que les entreprises qui offrent un soutien aux logiciels libres ne disposent généralement pas des ressources commerciales nécessaires pour participer à des appels d'offres, et adapter leurs politiques et processus d'achat de logiciels libres en conséquence.

## Évaluation de l'opportunité

Plusieurs raisons justifient les efforts pour mettre en place des politiques spécifiques d'approvisionnement en logiciels libres :

- L'offre de logiciels et de services commerciaux open source est croissante, ne peut être ignorée, et nécessite la mise en place de politiques et de processus d'approvisionnement spécifiques.
- Il existe une offre croissante de solutions commerciales open source très compétitives pour les systèmes d'information des entreprises.
- Même après avoir adopté un composant open source gratuit et l'avoir intégré dans une application, des ressources internes ou externes doivent être fournies pour maintenir ce code source.
- Le coût total de possession (TCO) est souvent (mais pas nécessairement) plus faible pour les solutions open source : pas de frais de licence à payer lors de l'achat/la mise à niveau, marché ouvert pour les fournisseurs de services, possibilité de fournir une partie ou la totalité de la solution soi-même.

## Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Les nouveaux appels à propositions demandent de manière proactive des soumissions open source.
- Le service achats dispose d'un moyen d'évaluer les solutions open source par rapport aux solutions propriétaires.
- Un processus d'achat simplifié pour les logiciels et services open source a été mis en place et documenté.
- Un processus d'approbation faisant appel à une expertise inter fonctionnelle a été défini et documenté.

## Recommandations

- Assurez-vous de tirer parti de l'expertise de vos équipes IT, DevOps, cybersécurité, gestion des risques, et achats lors de la mise en place du processus ([5 Open Source Procurement Best Practices](#), en anglais).
- La loi sur la concurrence peut exiger que le terme « open source » ne soit pas spécifiquement mentionné.

- Sélectionnez la technologie en amont, puis passez à l'appel d'offres pour les services de personnalisation et d'assistance.

### Ressources

- [Facteurs de décision en faveur de l'achat de logiciel open source](#) : ce n'est pas nouveau, mais c'est une excellente lecture de nos collègues d'OSS-watch au Royaume-Uni. Consultez-les [diapositives](#).
- [5 bonnes pratiques d'approvisionnement Open Source](#) : un article récent sur l'approvisionnement en logiciels libres avec des conseils utiles.

## 8 Activités de l'objectif « Stratégie »

### 8.1 Vers une stratégie de gouvernance open source d'Entreprise

Activity ID: [GGI-A-16](#).

#### Description

Définir une stratégie de gouvernance des logiciels open source au sein de l'entreprise garantit des approches cohérentes et visibles à la fois pour une utilisation interne et pour des contributions et des implications externes. Une vision et une direction claires et bien établies rendent la communication plus efficace.

Le passage à l'open source s'accompagne de nombreux avantages, mais aussi de certains devoirs et d'un changement de culture de l'entreprise. Il peut avoir un impact sur les modèles commerciaux et influencer la manière dont une organisation présente sa valeur et son offre, ainsi que sa position vis-à-vis de ses clients et de ses concurrents.

Cette activité comprend les tâches suivantes :

- Mettre en place un responsable open source, avec le parrainage et le soutien de la direction générale.
- Établir et publier une feuille de route claire pour l'open source, avec des objectifs précis et des avantages attendus.
- S'assurer que tous les cadres dirigeants en prennent connaissance et agissent conformément à cette feuille de route.
- Promouvoir les logiciels open source au sein de l'entreprise : encourager leur usage, les initiatives internes et le niveau de connaissances.
- Promouvoir le logiciel libre à l'extérieur de l'entreprise : par des déclarations et une communication officielles, et par une participation visible aux initiatives OSS.

Définir, publier et appliquer une stratégie claire et cohérente favorise également l'adhésion de toutes les personnes au sein de l'entreprise et facilite les initiatives des équipes.

#### Évaluation de l'opportunité

C'est le bon moment pour travailler sur cette activité si :

- Il n'y a pas d'effort coordonné de la part de la direction, et l'open source est toujours considéré comme une solution ad hoc.
- Il existe déjà des initiatives internes, mais elles ne remontent pas jusqu'aux niveaux supérieurs de la direction.
- L'initiative a été lancée il y a quelque temps, mais elle se heurte à de nombreux obstacles et ne donne toujours pas les résultats escomptés.

#### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Il existe une charte de gouvernance open source claire pour l'entreprise. Cette charte doit contenir :
  - les objectifs à atteindre,
  - pour qui faisons nous cela,
  - ce qui est du pouvoir du ou des stratèges et ce qui ne l'est pas.
- Une feuille de route open source est largement disponible et acceptée dans toute l'entreprise.

#### Recommandations

- Mettre en place un groupe de personnes et des processus pour définir et surveiller la gouvernance de l'open source au sein de l'entreprise. S'assurer qu'il existe bien un réel engagement du top management sur les initiatives open source.
- Communiquer sur la stratégie open source au sein de l'organisation, en faire une préoccupation majeure et un véritable engagement de l'entreprise.

- Veiller à ce que la feuille de route et la stratégie soient bien comprises par tout le monde, des équipes de développement aux gestionnaires et techniciens d'infrastructure.
- Communiquer sur ses progrès, afin que les gens sachent où en est l'organisation par rapport à son engagement. Publier régulièrement des mises à jour et des indicateurs.

### Ressources

- [Liste de vérifications et références pour une gouvernance open source.](#)
- [L'open source comme enjeu de souveraineté numérique, par Cédric Thomas, OW2 CEO, Workshop à Orange Labs, Paris, 28 janvier 2020.](#)
- [Une série de guides pour gérer les logiciels open source dans l'entreprise, par la Fondation Linux.](#)
- [Un bon exemple de document sur la stratégie open source, par le groupe LF Energy](#)

## 8.2 Sensibiliser les cadres dirigeants

Activity ID: [GGI-A-34](#).

### Description

L'initiative open source de l'organisation ne produira ses avantages stratégiques que si elle est appliquée au plus haut niveau, en intégrant l'ADN open source dans la stratégie et le fonctionnement interne de l'entreprise. Un tel engagement ne peut se produire si les cadres supérieurs et le comité de direction n'en font pas eux-mêmes partie. La formation et l'état d'esprit open source doivent également être étendus à ceux qui façonnent les politiques, les décisions et la stratégie globale, tant à l'intérieur qu'à l'extérieur de l'entreprise.

Cet engagement garantit que les améliorations pratiques, les changements d'état d'esprit et les nouvelles initiatives bénéficient d'un soutien constant, bienveillant et durable de la part de la hiérarchie, ce qui entraîne une participation plus fervente des travailleurs. Il façonne la façon dont les acteurs externes perçoivent l'organisation, ce qui apporte des avantages en termes de réputation et d'écosystème. C'est également un moyen de légitimer l'initiative et ses avantages à moyen et long terme.

### Évaluation de l'opportunité

Cette activité devient essentielle si/quand :

- L'organisation a fixé des objectifs globaux relatifs à la gestion des logiciels open source, mais peine à les atteindre. Il est peu probable que l'initiative puisse aboutir sans une bonne connaissance et un engagement clair des cadres dirigeants.
- L'initiative a déjà commencé et progresse, mais les niveaux supérieurs de la hiérarchie ne la suivent pas correctement.

Il devrait devenir évident, en toute logique, que tout ce qui n'est pas une utilisation ad-hoc de l'open source nécessite une approche cohérente et réfléchie, étant donné l'étendue de l'activité en termes d'équipes impliquées et de changement culturel.

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Il existe un bureau/responsable de la gouvernance mandaté pour définir une stratégie uniforme en matière de logiciels open source dans toute l'entreprise et s'assurer que le champ d'application est clair.
- Il y a un engagement clair et contraignant de la hiérarchie envers la stratégie de logiciels open source.
- Il y a une communication transparente de la hiérarchie sur son engagement envers le programme.
- La hiérarchie est disponible pour discuter des logiciels libres. Elle peut être sollicitée et mise au défi sur ses promesses.
- Il existe un budget et un financement appropriés pour l'initiative.

## Recommandations

Voici quelques exemples d'actions associées à cette activité :

- Organiser des formations pour démystifier l'open source auprès des cadres dirigeants.
- Obtenir une approbation explicite et pratique de l'utilisation et de la stratégie du logiciel open source.
- Mentionner et approuver explicitement le programme open source dans les communications internes.
- Mentionner et approuver explicitement le programme open source dans les communications publiques.

Open source is a *strategic enabler* that embarks *enterprise culture*. What does this mean?

- L'open source peut être utilisé comme un mécanisme pour challenger les fournisseurs et réduire les coûts d'acquisition des logiciels.
  - Should open source come under the purview of *Software Asset Managers* or *purchasing departments*?
- Open source licences enshrine the freedoms that deliver the benefits of open source, but they also carry *obligations*. If not met appropriately, responsibilities can create legal, commercial and image risks to an organisation.
  - Les conditions de la licence permettront-elles de voir des parties du code qui devraient rester confidentielles ?
  - Cela aura-t-il un impact sur le portefeuille de brevets de mon organisation ?
  - Comment les équipes projet doivent-elles être formées et supportées à ce sujet ?
- La plus grande valeur de l'open source réside dans la contribution aux projets open source externes.
  - Comment mon entreprise doit-elle encourager (et suivre) cette démarche ?
  - Comment les développeurs doivent-ils utiliser GitHub, GitLab, Slack, Discord, Telegram, ou tout autre outil que les projets open source utilisent habituellement ?
  - L'open source peut-il avoir un impact sur les politiques de ressources humaines de l'entreprise ?
- Bien sûr, il ne s'agit pas seulement de contribuer en retour, qu'en est-il de mes propres projets open source ?
  - Suis-je prêt à faire de l'innovation *ouverte* ?
  - Comment mes projets vont-ils gérer les contributions *entrantes* ?
  - Dois-je faire l'effort d'entretenir une communauté pour un projet donné ?
  - Comment dois-je diriger la communauté, quel rôle doivent jouer ses membres ?
  - Suis-je prêt à céder les décisions relatives à la feuille de route à une communauté ?
  - L'open source peut-il être un outil précieux pour réduire le cloisonnement entre les équipes de l'entreprise ?
  - Dois-je gérer le transfert de l'open source d'une entité de l'entreprise à une autre ?

## 8.3 Open source et souveraineté numérique

Activity ID: [GGI-A-35](#).

### Description

La souveraineté numérique peut être définie comme

“La capacité et la possibilité pour les individus et les institutions de jouer leur(s) rôle(s) dans le monde numérique de manière indépendante, intentionnelle et sûre.” — Centre de compétences pour l'informatique publique, Allemagne

Afin de mener ses activités correctement, toute entité doit pouvoir s'appuyer sur plusieurs partenaires, services, produits et outils. L'examen des liens et des contraintes de ces dépendances permet à l'organisation d'évaluer et de contrôler sa soumission aux facteurs externes, et par conséquent d'améliorer son autonomie et sa résilience.

Par exemple, une dépendance exclusive à un fournisseur peut entraver les processus et la valeur ajoutée de l'organisation ; en tant que telle, il faut donc l'éviter. L'open source procure un moyen

de contourner ce verrouillage. L'open source joue un rôle important dans la souveraineté numérique, en permettant un plus grand choix entre les solutions, les fournisseurs et les intégrateurs, et un meilleur contrôle des feuilles de route informatiques.

Il convient de noter que la souveraineté numérique n'est pas une question de confiance : nous devons évidemment faire confiance à nos partenaires et fournisseurs, mais la relation est encore meilleure lorsqu'elle est fondée sur la reconnaissance et le consentement mutuels, plutôt que sur des contrats forcés et sur des contraintes.

Une meilleure souveraineté numérique apporte les avantages suivants, permettant à l'organisation d'améliorer :

- sa capacité à faire ses propres choix sans contraintes,
- sa résilience face aux acteurs et aux facteurs externes,
- sa position de négociation face aux partenaires et fournisseurs de services.

### Évaluation de l'opportunité

- Dans quelle mesure est-il difficile/coûteux de se passer d'une solution ?
- Les fournisseurs de solutions peuvent-ils imposer des conditions non souhaitées à leurs services (par exemple, un changement de licence ou une mise à jour de contrat) ?
- Les fournisseurs de solutions pourraient-ils augmenter unilatéralement leurs prix, faute d'alternatives ?

### Suivi de l'avancement

Les **points de contrôle** suivants dénotent une progression de cette Activité :

- Les dépendances critiques vis-à-vis de fournisseurs et de partenaires de l'organisation sont évaluées.
- Il existe un plan de secours face à ces dépendances identifiées.
- L'exigence de souveraineté numérique est affichée à chaque nouvelle solution étudiée.

### Recommandations

- Identifier les risques de dépendances critiques vis-à-vis des fournisseurs de services et des partenaires.
- Tenir à jour une liste d'alternatives open source pour les services critiques.
- Intégrer le critère de souveraineté numérique à l'examen de tout nouvel outil ou service envisagé.

### Ressources

- [Une introduction à la souveraineté numérique et à l'Open Source : partie I](#) et [Une introduction à la souveraineté numérique et à l'Open Source : partie II](#), du site Open-Sourcerers.
- L'excellent article de superuser.openstack.org sur [Le rôle de l'Open Source dans la souveraineté numérique](#). En voici un court extrait :  
La souveraineté numérique est une préoccupation majeure du 21<sup>e</sup> siècle, en particulier pour l'Europe. L'open source a un rôle majeur à jouer dans la mise en place de la souveraineté numérique, en permettant à chacun d'accéder à la technologie nécessaire, mais aussi en fournissant la transparence de la gouvernance et l'interopérabilité nécessaires au succès de ces solutions.
- Le point de vue de l'Union Européenne sur la souveraineté numérique, selon l'[Open Source Observatory \(OSOR\)](#) : Open Source, souveraineté numérique et interopérabilité : La déclaration de Berlin.
- La position de l'UNICEF sur l'[Open Source pour la souveraineté numérique](#).

## 8.4 L'open source au service de l'innovation

Activity ID: [GGI-A-36](#).

**Description**

L'innovation est la mise en œuvre pratique d'idées qui aboutissent à l'introduction de nouveaux biens ou services ou à l'amélioration de l'offre de biens ou de services.

— Schumpeter, Joseph A.

L'open source peut être un facteur clé d'innovation grâce à la diversité, à la collaboration et à un échange fluide d'idées. Des personnes issues d'horizons et de domaines différents peuvent avoir des points de vue divers et apporter des réponses nouvelles, améliorées, voire révolutionnaires aux problèmes connus. On peut favoriser l'innovation en écoutant les différents points de vue et en encourageant activement la collaboration ouverte sur des projets et des sujets.

De même, participer à l'élaboration et à la mise en œuvre de normes ouvertes est une excellente façon de promouvoir de bonnes pratiques et idées pour améliorer le travail quotidien de l'entreprise. Cela permet également à l'entité de piloter et d'influencer l'innovation là où elle en a besoin et d'améliorer globalement sa visibilité et sa réputation.

Grâce à l'innovation, l'open source permet non seulement de transformer les biens ou les services que votre entreprise commercialise, mais aussi de créer ou de modifier l'ensemble de l'écosystème dans lequel elle souhaite prospérer.

Par exemple, en diffusant Android en open source, Google invite des centaines de milliers d'entreprises à créer leurs propres services sur la base de cette technologie open source. Google crée ainsi un écosystème dont tous les participants peuvent bénéficier. Bien sûr, très peu d'entreprises sont assez puissantes pour créer un écosystème par leur propre décision : mais il existe de nombreux exemples d'alliances entre entreprises pour créer un tel écosystème.

**Évaluation de l'opportunité**

Il est important d'évaluer la position de votre entreprise par rapport aux usages en vigueur chez les concurrents, partenaires et clients, pour ne pas trop l'éloigner des standards et technologies qu'ils utilisent. Innover, c'est évidemment se différencier, mais ce qui diffère ne doit pas représenter un périmètre trop large, sinon votre entreprise ne bénéficierait pas des développements logiciels réalisés par les autres acteurs de l'écosystème ni de l'élan commercial que celui-ci procure.

**Suivi de l'avancement**

Les **points de contrôle** suivants dénotent une progression de cette activité :

- Les technologies - et les communautés open source qui les développent - ayant un impact sur l'entreprise ont été identifiées.
- Les progrès et les publications de ces communautés open source sont suivis - vous êtes même au courant de leur stratégie avant que les versions ne soient rendues publiques.
- Les employés de l'organisation sont membres de (certaines de) ces communautés open source et influencent leurs feuilles de route et leurs choix techniques en contribuant aux lignes de codes et en participant aux organes de gouvernance de ces communautés.

**Recommandations**

Parmi toutes les technologies nécessaires au fonctionnement de votre entreprise, vous devez identifier :

- les technologies qui pourraient être les mêmes que celles de vos concurrents,
- les technologies qui doivent être spécifiques à votre entreprise.

Restez au courant des technologies émergentes. L'open source est le moteur de l'innovation depuis une dizaine d'années, et de nombreux outils puissants d'usage quotidien en sont issus (pensez à Docker, Kubernetes, aux projets Big Data d'Apache ou à Linux). Inutile de tout savoir sur tout, mais il faut connaître suffisamment l'état de l'art pour identifier les nouvelles tendances intéressantes.

Autorisez et encouragez les personnes à soumettre des idées innovantes, et à les mettre en avant. Si possible, consacrez des ressources à ces initiatives et faites-les croître. Comptez sur la passion et la volonté de chacun pour créer et favoriser les idées et les tendances émergentes.

## Ressources

- 4 innovations que nous devons à l'Open Source.
- Les innovations de l'Open Source, par le Professeur Dirk Riehle.
- La technologie Open Source, source d'innovation.
- L'innovation Open Source peut-elle fonctionner dans l'entreprise ?.
- Europe : stratégie en matière de logiciels Open Source.
- Europe : stratégie 2020-2023 pour les logiciels Open Source.

## 8.5 L'open source au service de la transformation numérique

Activity ID: GGI-A-37.

### Description

"La transformation numérique est l'adoption de la technologie numérique pour transformer les services ou les entreprises, en remplaçant des processus non numériques ou manuels par des processus numériques ou en remplaçant une technologie numérique ancienne par une technologie numérique plus récente." (Wikipedia)

Lorsque les organisations les plus avancées dans la transformation numérique pilotent conjointement le changement de leurs activités métier, informatiques et financières pour ancrer le numérique dans leur évolution, elles reconsidèrent leur :

- Modèle économique : la chaîne de valeur via les écosystèmes, les solutions SaaS, le recours aux services.
- Finance : opex/capex, personnel, externalisation.
- Informatique : innovation, modernisation de l'héritage/des actifs.

L'open source est au cœur de la transformation numérique :

- Technologies, pratiques agiles, gestion de produits.
- Ressources humaines : collaboration, communication ouverte, cycle de développement/décision.
- Modèles économiques : essayer avant d'acheter, innovation ouverte.

En termes de compétitivité, les processus les plus visibles sont probablement ceux qui ont un impact direct sur l'expérience client. Et nous devons reconnaître que les grands acteurs, ainsi que les start-ups, ont radicalement changé les attentes des clients en offrant une expérience client inédite.

L'expérience client, comme tous les autres processus de l'entreprise, dépend entièrement de l'informatique. Chaque entreprise doit transformer son informatique, c'est l'objet de la transformation numérique. Les entreprises qui ne l'ont pas encore fait doivent maintenant réaliser leur transformation numérique le plus rapidement possible, sinon elles risquent d'être rayées du marché. La transformation numérique est une condition de survie. Les enjeux étant très importants, une entreprise ne peut pas entièrement confier sa transformation numérique à un fournisseur. Chaque entreprise doit mettre la main à la pâte avec son informatique, donc avec les logiciels libres, car il n'y a pas d'informatique sans logiciels libres.

Les avantages attendus de la transformation numérique sont les suivants :

- Simplifier, automatiser les processus de base, les rendre temps réel.
- Permettre des réponses rapides aux changements de la concurrence.
- Tirer parti de l'intelligence artificielle et du big data.

### Évaluation de l'opportunité

La transformation numérique pourrait être gérée par :

- Segments de l'IT : l'IT de production, l'IT au service d'activités (CRM, facturation, approvisionnement...), l'IT orientée métier (RH, finance, comptabilité...), le Big Data.
- Plusieurs technologies ou processus soutenant l'IT : l'infrastructure (cloud), l'IA (Intelligence Artificielle), les processus (Make-or-Buy, DevSecOps, SaaS).



Injecter de l'open source dans un segment ou une technologie particulière de votre informatique révèle que vous voulez garder la main dessus, car vous avez évalué qu'il est important pour la compétitivité de votre entreprise. Il importe d'évaluer la position de votre entreprise par rapport à vos concurrents, mais aussi par rapport à d'autres industries et aux acteurs clés en termes d'expérience client et de solutions de marché.

### Suivi de l'avancement

#### 1. Niveau 1 : Évaluation de la situation

- J'ai identifié :
  - les segments de l'informatique qui sont importants pour la compétitivité de mon entreprise, et
  - les technologies open source nécessaires pour développer des applications dans ces segments.
- Et j'ai donc décidé :
  - sur quels segments je veux gérer en interne le développement de projets, et
  - sur quelles technologies open source je dois développer une expertise en interne.

#### 1. Niveau 2 : Engagement

- Sur certaines technologies open source utilisées dans l'entreprise, plusieurs développeurs ont été formés et sont reconnus comme des contributeurs de valeur par la communauté open source.

Dans certains segments, des projets fondés sur des technologies open source ont été lancés.

#### 1. Niveau 3 : Généralisation

- Pour tous les projets, une alternative open source est systématiquement étudiée dès la phase de lancement. Pour faciliter l'étude de cette alternative open source par l'équipe projet, un budget central et une équipe d'architectes, hébergés au sein du département informatique, sont consacrés à l'assistance aux projets.

Les **KPIs** :

- KPI 1. Ratio pour lequel une alternative open source a été étudiée : (nombre de projets / nombre total de projets).
- KPI 2. Ratio pour lequel l'alternative open source a été choisie : (nombre de projets / nombre total de projets).

### Recommandations

Au-delà d'un titre, la transformation numérique est un état d'esprit qui implique des changements fondamentaux, qui devraient aussi (ou même principalement) venir des strates supérieures de l'organisation. La direction doit promouvoir les initiatives et les nouvelles idées, gérer les risques et éventuellement mettre à jour les procédures existantes pour les adapter aux nouveaux concepts.

La passion est un énorme facteur de réussite. L'un des moyens développés par les acteurs clés dans ce domaine est de créer des espaces ouverts aux nouvelles idées, où les gens peuvent soumettre des propositions et travailler librement sur leurs idées concernant la transformation numérique. La direction doit encourager de telles initiatives.

### Ressources

- [Fondation Eclipse : Favoriser la transformation numérique en Europe grâce à la collaboration mondiale en matière de logiciels libres.](#)
- [Europe : stratégie en matière de logiciels Open Source.](#)
- [Europe : stratégie 2020-2023 pour les logiciels Open Source.](#)

## 9 Conclusion

Comme nous l'avons déjà mentionné, la bonne gouvernance open source n'est pas une fin en soi mais bien un cheminement. Nous devons nous soucier de nos biens communs, des communautés et de l'écosystème qui les font prospérer, car notre réussite commune, et donc individuelle, en dépend.

En tant que praticiens du domaine des logiciels et passionnés d'open source, **nous** nous engageons à continuer d'améliorer ce mémento de bonne gouvernance et à travailler à sa diffusion et son rayonnement. Nous croyons fermement que les organisations, les individus et les communautés doivent travailler de concert pour construire un meilleur patrimoine de biens commun, accessible et bénéfique pour tous.

**Vous** êtes les bienvenus pour rejoindre l'OSPO Alliance, contribuer à notre travail, faire passer le mot et être l'ambassadeur d'une meilleure connaissance et gouvernance de l'open source au sein de votre propre écosystème. Il existe une grande variété de ressources disponibles, depuis les articles de blog et les articles de recherche jusqu'aux conférences et aux cours de formation en ligne. Nous fournissons également un ensemble de documents utiles sur [notre site web](#), et nous sommes ravis de pouvoir vous aider autant que possible.

**Définissons et construisons ensemble l'avenir de ce mémento de bonne gouvernance !**

### 9.1 Contact

Le meilleur moyen pour entrer en contact avec l'OSPO Alliance est de poster un message sur notre liste de diffusion publique à l'adresse <https://accounts.eclipse.org/mailling-list/ospo.zone>. Vous pouvez également venir discuter avec nous lors des événements open source habituels, participer à nos webinaires mensuels OSPO OnRamp, ou encore prendre contact avec l'un des membres - ils vous redirigeront volontiers vers la bonne personne.

### 9.2 Annexe : Modèle de tableau de bord personnalisé d'activité

La dernière version du modèle de tableau de bord personnalisé des activités est disponible dans la section « ressources » du [GitLab de l'initiative de bonne gouvernance](#).

OW2 OSS Good Governance initiative		<b>LA BONNE COMPAGNIE S.A.</b>		Tableau de Bord personnalisé	
<b>Objectif / Activité</b> Culture 1		<b>Promouvoir les bonnes pratiques de développement open source</b>		<b>Mis à jour le 07/28/21</b>	
<b>Description Personnalisée</b> <i>Portée du travail à réaliser</i> Description succincte ... <ul style="list-style-type: none"><li>Faits marquants...</li><li></li></ul>		<b>Évaluation de l'opportunité</b> <i>Pourquoi cette activité est pertinente</i> <ul style="list-style-type: none"><li>Points à améliorer / blocages...</li><li>Principaux axes de progression...</li></ul>			
<b>Objectifs</b> <i>Ce qu'on attend de cette itération</i> <ul style="list-style-type: none"><li>Objectif 1...</li><li>Objectif 2...</li></ul>		<b>Outils</b> <i>Technologies, outils, produits employés pour cette activité</i> <ul style="list-style-type: none"><li>Ressources...</li></ul>		<b>Notes opérationnelles</b> <i>Approches, méthodes pour faire progresser l'activité</i> <ul style="list-style-type: none"><li>Commencer par...</li><li></li></ul>	
<b>Résultats clés</b> <i>Comment mesurer le succès de cette itération</i>		<b>Progrès</b>	<b>Score</b>	<b>Évaluation personnelle</b>	
1. Résultat clé 1 (minimum un résultat clé)		<b>xx%</b>	<b>.9</b>	Commentaire personnel	
2. Résultat clé 2		<b>xx%</b>	<b>.5</b>	Commentaire personnel	
3. Résultat clé 3		<b>xx%</b>	<b>.5</b>	Commentaire personnel	
4. Résultat clé 4 (maximum quatre résultats clés)		<b>xx%</b>	<b>.0</b>	Commentaire personnel	
			<b>.475</b>		
<b>Calendrier</b> <i>Dates de début/fin, Étapes</i> <ul style="list-style-type: none"><li>Indiquer des dates/étapes</li></ul>		<b>Efforts</b> <i>Temps et budget</i> <ul style="list-style-type: none"><li>Allocation de temps pour les trois prochains mois</li><li>Budget autorisé</li></ul>		<b>Participants</b> <i>Qui participe ? Dirige ?</i> <ul style="list-style-type: none"><li>XX doit préparer une présentation interne</li></ul>	
<b>Problèmes</b> <i>Difficultés, incertitudes, blocages, points d'attention, dépendances</i> <ul style="list-style-type: none"><li>Problème 1...</li><li>Problème 2...</li></ul>		<b>État</b> <i>Comment se passe l'activité</i>  Commentaire personnel sur la santé de l'activité			
		<b>Suivi global d'avancement</b>		<b>XX%</b>	

**Notes**

**Aperçu du forum lié à l'activité sur GitLab**

<https://gitlab.ow2.org/ggi/ggi-castalia/-/issues/25>

Copiez / collez ici le contenu de la description de l'activité depuis <https://gitlab.ow2.org/ggi/ggi-castalia/>

Ce document servira de référence pour l'élaboration du tableau de bord des activités personnalisées