

Wildhainweg 9
Case postale 6935
3001 Berne
Téléphone 031 633 59 00
Télécopie 031 633 59 99
www.be.ch/oio
info.kaio@fin.be.ch



Plateforme Open source

Guide concernant la création de communau- tés-

Date d'édition	27. novembre 2018
Version	1.0
Statut du document	Autorisé
Classification	Non classifié
Auteur	Joos Thomas, FIN-KAIO-AP-SW
N° de document	287257

Sommaire

1	Objet	3
2	Structure du projet de communauté	4
3	Décisions de principe	5
3.1	Gestion de produit	5
3.2	Implication des fournisseurs	6
3.3	Répartition des coûts.....	7
4	Teneur du projet de communauté	8
4.1	Objectifs	9
4.2	Organisation	9
4.3	Feuille de route et processus de changement	11
4.4	Processus de développement.....	12
4.5	Partage des frais et fournisseurs	13
4.6	Commercialisation	14
4.7	Gestion des contributions externes.....	15
4.8	Exploitation de l'application.....	18
5	Liste de projets de communauté existants	19
6	Procès-verbal du document	20

1 Objet

Le présent guide s'adresse aux spécialistes TIC qui veulent distribuer une application dont ils sont responsables en rendant son code source public. Il leur fournit des informations importantes sur l'organisation et la mise en place d'une communauté.

Le genre de communauté qui convient le mieux à une application dépend étroitement de l'environnement informatique, des utilisateurs et utilisatrices possibles et de la vision de l'institution qui distribue le logiciel. Étant donné qu'il n'existe pas de méthode universelle de création de communautés, son responsable devra, sur la base du présent document, rédiger un projet concret pour son application.

Le présent guide contient également une liste de projets déjà existants, afin de favoriser la normalisation.

L'organisation de communautés pour des bibliothèques ne fait pas directement l'objet de recommandations : l'utilisation de processus standard simples définis dans le modèle de dépôt rend en général inutile l'élaboration d'un projet.

2 Structure du projet de communauté

Le document à produire doit avoir la structure présentée ci-dessous. Selon le genre de communauté, certains sous-chapitres ne seront pas nécessaires.

1. Objectifs
 - a. Application
 - b. Communauté
2. Organisation
 - a. Propriétaire du code source
 - b. Organes
 - c. Droits de vote
3. Feuille de route et processus de changement
4. Développement
 - a. Droits des *committers*
 - b. Procédure de révision
5. Répartition des frais et fournisseurs
6. Commercialisation
7. Gestion des contributions externes
 - a. Gestion des bogues signalés
 - b. Gestion des demandes de conseil
 - c. Gestion des *pull requests* (demandes de tirage)
 - d. Gestion des *forks* (embranchements)
8. Exploitation de l'application

3 Décisions de principe

Pour élaborer le projet de communauté, il faut commencer par répondre à trois questions de fond :

- a) Comment résoudre la question de la gestion de produit et qui définit la feuille de route ?
- b) Comment faut-il associer les fournisseurs à la démarche ?
- c) Comment la question de la répartition des frais est-elle réglée ?

Les réponses à ces questions déterminent le modèle de communauté.

Nous vous recommandons de mettre l'accent sur l'avenir proche pour la définition de la communauté. Si l'application n'a pas encore d'utilisatrices ou utilisateurs intéressés, il n'est guère judicieux de créer une association et de formuler des procédures complexes d'élaboration de la feuille de route pour définir la communauté.

3.1 Gestion de produit

Il existe différentes modalités de gestion de produit, en fonction du type d'application, de son importance stratégique et du stade du cycle de vie auquel elle se trouve. Pour le projet de communauté, il est particulièrement important d'une part de définir la feuille de route selon les exigences propres à l'informatique et au domaine auquel l'application est destinée et, d'autre part, d'arrêter la publication des versions.

Possibilités :

- a) Canton de Berne : le canton de Berne entend rester maître de l'application, il définit lui-même la feuille de route.
- b) Fonctionnement collaboratif : en collaboration avec d'autres utilisateurs et utilisatrices.
- c) Organisation indéterminée : liens souples, pas de feuille de route active

3.1.1 Canton de Berne

Si l'application a une grande importance stratégique ou si le canton de Berne doit pouvoir mettre ses modifications rapidement en œuvre, il est tout à fait sensé de garder la maîtrise du logiciel. Le canton de Berne est alors le seul à décider des modules à ajouter au logiciel et du moment auquel il le fera.

Cette solution a pour inconvénient de contraindre les autres utilisatrices et utilisateurs potentiels à employer leur propre version (embranchement ou *fork*), faute de quoi ils n'ont guère la possibilité de réaliser les adaptations et les extensions dont ils ont besoin. Cela ne s'oppose pas à la distribution du logiciel sous code source public, les outils de gestion du code¹ proposant une assistance étendue pour ces cas de figure. Il est possible d'adapter les logiciels et de corriger les erreurs de façon sélective dans les deux sens.

Il est très difficile, dans ce modèle, de partager les frais.

3.1.2 Élaboration collaborative

Les décisions concernant la feuille de route et les priorités sont prises de concert avec les autres membres de la communauté. Membre de celle-ci, le canton de Berne participe à la prise de décisions, mais n'a pas le dernier mot.

¹ Comme GitHub.

Pour que la prise de décisions collective fonctionne, il faut que des structures et des règles aient été définies au préalable. La répartition des coûts inhérents à la mise en œuvre des adaptations décidées d'un commun accord doit aussi être réglée.

Ce modèle aboutit à une seule version de l'application, utilisée par tous les membres, de sorte que le coût total est nettement inférieur à celui des autres variantes. Toutefois, la souplesse d'utilisation diminue, car les utilisateurs et utilisatrices doivent se concerter avant toute adaptation, ce qui ralentit la prise de décisions et la rend plus laborieuse.

Ce modèle convient en particulier aux applications qui seront développées avec des partenaires clairement définis (comme d'autres cantons).

3.1.3 Organisation indéterminée

Dans ce modèle, les processus et les structures ne sont définis que de façon vague. S'il reste certes maître de l'application, le canton est toutefois ouvert aux contributions et aux adaptations.

Les membres ne définissent pas de feuille de route, ou seulement une feuille de route très sommaire, le consensus se dégageant de façon informelle des discussions tenues directement sur la plateforme de distribution. La communauté est plutôt dynamique, les membres peuvent être très actifs pendant longtemps, puis se retirer.

Ce modèle convient en particulier à des applications de petite taille, déjà parvenues au stade d'exploitation et qui sont « terminées ». Il s'agit aussi généralement de la meilleure variante pour les composants ou applications de nature technique conçus pour un groupe cible plutôt nombreux qui n'est pas défini avec exactitude à l'avance.

3.2 Implication des fournisseurs

Il y a essentiellement deux façons d'associer les fournisseurs au développement :

- a) *Rôle de mandataire* : le canton pourra en principe s'en passer à moyen terme. S'il est chargé du développement de l'application sur mandat du canton (ou de la communauté) dans le respect de critères de coût et de qualité, le fournisseur n'a toutefois aucune influence sur la feuille de route et les priorités, ou, à la rigueur, uniquement en qualité de consultant.
- b) *Rôle de partenaire stable avec droit de codécision* : le fournisseur participe à la prise de décision concernant le développement de l'application, ce qui fortifie son rôle et accroît sa disposition à investir dans l'application. Dans ce modèle, les fournisseurs distribuent ensuite le logiciel et prospectent le marché à la recherche de nouveaux clients.

Le rôle actif joué par le fournisseur comporte des avantages pour l'administration cantonale : la commercialisation de l'application accroît les probabilités que la communauté augmente à un rythme soutenu, ce qui réduit plus rapidement le coût par membre.

S'il choisit ce modèle qui accorde au fournisseur des droits de participation élargis, le canton de Berne doit veiller à respecter les règles relatives aux marchés publics. Il est recommandé de faire appel à au moins deux fournisseurs, afin d'éviter une dépendance excessive envers un seul d'entre eux. D'autres fournisseurs doivent aussi avoir la possibilité de rejoindre la communauté.

3.3 Répartition des coûts

Il existe deux modèles principaux de répartition des coûts de maintenance, de développement et de mise au point de nouvelles fonctions :

- a) *Individualisation des coûts* : chaque partenaire supporte les coûts occasionnés par les adaptations qu'il souhaite. Il est toutefois possible de décider, au cas par cas, que certains développements seront pris en charge par tous les partenaires.

- b) *Mutualisation des coûts* : les coûts sont répartis selon une clé définie, chaque partenaire prenant à sa charge une partie de tous les frais. Des exceptions ne s'appliquent qu'à des développements très concrets, à la charge de leurs utilisateurs et utilisatrices.
La clé de répartition peut être par exemple la taille du canton en question ou le nombre d'utilisateurs et utilisatrices.

En principe, la variante b (mutualisation des coûts) n'est judicieuse que si les partenaires ont aussi opté pour la gestion de produit collective (variante b, élaboration collaborative). En effet, seul le membre qui peut prendre part à la prise de décision sera disposé à assumer les coûts qui en découlent.

4 Teneur du projet de communauté

Le présent chapitre fournit des conseils pour la rédaction du projet de communauté sur la base des décisions de principe énoncées au point 3.

Gestion de produit :

- a) Canton de Berne
- b) Élaboration collective
- c) Organisation indéterminée

Implication du fournisseur :

- a) Rôle de mandataire
- b) Rôle de partenaire

Répartition des coûts :

- a) Individualisation des coûts
- b) Mutualisation des coûts

Lorsque la teneur diffère en raison des décisions de principe adoptées, nous utilisons la structure en tableau ci-dessous. Le *texte en italique* sert de modèle et peut s'intégrer sans modification au projet de communauté.

GP ²	F ³	C ⁴	Proposition
c	b	a	<p>Contenu ou suggestions lorsque les décisions de principe sont les suivantes :</p> <p>Gestion de produit : c) Organisation indéterminée Implication du fournisseur : b) Rôle de partenaire Répartition des coûts : a) Individualisation des coûts</p> <p><i>Ce texte peut être copié dans le modèle et adapté.</i></p>
b	-	b	<p>Contenu ou suggestions lorsque les décisions de principe sont les suivantes :</p> <p>Gestion de produit : b) Élaboration collaborative Implication du fournisseur : a) ou b) (ne joue aucun rôle) Répartition des coûts : b) Mutualisation des coûts</p>

Le projet de communauté doit être un document évolutif, qui peut être modifié à tout moment. Il n'est guère utile de planifier trop longtemps à l'avance et il est préférable d'adapter le projet à la situation initiale. Si d'autres membres adhèrent ultérieurement à la communauté, il est possible de modifier tant l'organisation que les processus et de mettre en place des mécanismes plus complexes.

² Gestion de produit.

³ Implication des fournisseurs.

⁴ Répartition des coûts.

La distribution en sous-chapitres s’inspire directement de la structure proposée pour le projet de communauté (voir point 2).

4.1 Objectifs

Les objectifs expriment d’une part le véritable objet, la « vision » de l’application. Cette vision doit aussi se refléter dans le fichier « README.md » du dépôt (cf. « Check-list Code source et documentation »).

Il faut par ailleurs décrire les attentes de la communauté :

- Qui doit adhérer ? Qui sont les membres potentiels ?
- Faut-il prospecter le marché à la recherche de nouveaux membres ?
- Quels sont les principaux objectifs de la distribution du logiciel en code source ouvert ?

4.2 Organisation

GP	F	C	Proposition
a	-	a	<p>Puisque le canton de Berne reste maître du logiciel, il n’est pas nécessaire de mettre sur pied une organisation supplémentaire. L’organe diffuseur du logiciel (un office, par ex.) en reste propriétaire et se charge de toutes les tâches de coordination.</p> <p>Le chapitre « Organisation » sera dans ce cas très concis.</p>
b	-	a	<p>Dans ce cas de figure, nous recommandons d’adopter la forme juridique de la société simple.</p> <p>Afin que la feuille de route et les exigences puissent être élaborées en même temps et harmonisées, il est recommandé de créer deux organes dans lesquels siège un représentant ou une représentante de chaque membre de la communauté :</p> <ul style="list-style-type: none"> • un comité exécutif, qui définit la stratégie et les priorités ; • une commission technique, qui formule les exigences et les contenus concrets au niveau expert. Selon le volume de travail, il est aussi possible d’instituer une commission technique pour chaque thème. <p>Le propriétaire du code est le service qui le distribue.</p> <p>Lorsque les structures deviennent plus complexes ou que la communauté compte plus de cinq membres ou utilisatrices et utilisateurs, on peut envisager la forme juridique de l’association. Observez à cet effet les commentaires de la variante qui suit.</p>
b	-	b	<p>Il est recommandé de choisir la forme juridique de l’association. Il est possible soit de créer une association ad hoc de diffuser l’application par le biais d’une association déjà existante. La création d’une association s’impose pour éviter que la ventilation des coûts devienne trop laborieuse et parce que les tâches de coordination génèrent probablement suffisamment de travail pour engager un gérant ou une gérante à temps partiel.</p> <p>Le service transmet ensuite les droits sur le code à l’association, qui se</p>

			<p>charge de la gestion de la communauté et des commandes auprès des fournisseurs.</p> <p>Le service juridique de l'OIO se fera un plaisir de vous aider à rédiger les statuts.</p>
b	a	a	<p>(complète « b_-b »)</p> <p>Dans ce cas de figure, les fournisseurs ne sont pas membres de l'association, mais uniquement mandataires. Le gérant ou la gérante ne doit pas être mis à disposition par un fournisseur.</p>
b	b	b	<p>(complète « b_-b »)</p> <p>Les fournisseurs sont membres de l'association, un d'entre eux peut aussi mettre à disposition le gérant ou la gérante.</p>
c	-	a	<p>Il n'est pas nécessaire de décrire l'organisation dans les détails, car celle-ci doit rester indéterminée. Il faut toutefois décider qui reçoit les demandes de conseil des tiers et les traite (compétence).</p> <p>Il faut aussi déterminer si des tiers doivent être associés aux travaux et de quelle façon :</p> <ul style="list-style-type: none"> • pas d'implication directe : les tiers ne peuvent que soumettre propositions et contributions (sous la forme de <i>pull requests</i>). • implication comme contributeur : les tiers qui fournissent régulièrement de bonnes contributions peuvent être associés directement aux travaux. • transmission à des tiers : si, après la distribution, des tiers contribuent davantage au développement du logiciel que le canton de Berne, l'application peut leur être transmise. <p>Le service chargé de la distribution garde la propriété du logiciel.</p> <p>Cette variante équivaut à l'organisation des projets « classiques » à code source ouvert. L'exemple https://opensource.guide/leadership-and-governance/ fournit de plus amples informations.</p>
c	a	a	<p>(complète « c_-a »)</p> <p>Le fournisseur ne se charge que de tâches de coordination purement techniques (audit du code, évaluation, etc.).</p>
c	b	a	<p>(complète « c_-a »)</p> <p>Le fournisseur peut se charger des tâches de coordination.</p>

4.3 Feuille de route et processus de changement

GP	F	C	Proposition
a	a	-	<i>Le canton de Berne définit la feuille de route et les changements à mettre en œuvre. Il applique ce faisant les processus standard de l'office XY.</i>
a	b	-	<i>Le canton de Berne définit la feuille de route de concert avec la société XY. Il décide des changements à mettre en œuvre. Il applique ce faisant les processus standard de l'office XY.</i>
b	-	a	<p>Les membres de la société ou de l'association arrêtent la procédure d'élaboration de la feuille de route et d'approbation des changements. D'autres procédures peuvent s'imposer en fonction de la structure des membres (nombre, rapports de taille, etc.).</p> <p>Toutes les procédures doivent cependant prévoir des mécanismes pour résoudre les conflits et dégager des majorités.</p>
b	-	b	Comme « b_-a », à la différence qu'il faut en sus fixer une clé de répartition des coûts. Il faut aussi tenir compte de la façon de régler la répartition des coûts des modifications que certains membres ne souhaitent pas. Il peut en effet arriver, surtout dans les associations dans lesquelles la taille des membres est inégale, qu'un membre particulièrement grand (comme le canton de Berne) assume une part considérable du coût d'un développement sans en retirer aucune utilité.
c	-	-	<p>Exemple, ne convient pas à toutes les situations.</p> <p><i>Les membres renoncent à formuler une feuille de route, car l'application satisfait actuellement leurs besoins.</i></p> <p><i>Ils débattent ouvertement des changements et recherchent le consensus. Le propriétaire du code (canton de Berne) décide en cas d'égalité des voix.</i></p>

4.4 Processus de développement

GP	F	C	Proposition
a	-	-	<p><i>Les droits des committers (droits d'accès en écriture au code) sont réservés aux fournisseurs désignés par le canton de Berne. Ces droits leur sont retirés à l'échéance du mandat.</i></p> <p><i>Les fournisseurs doivent procéder à l'audit technique du code des contributions externes acceptées sur le plan du contenu par le canton de Berne. Ils doivent veiller à la qualité technique et le canton de Berne les rémunère pour ces travaux.</i></p>
b	a	a	<p><i>En principe, tous les fournisseurs mandatés par un membre de la communauté ont des droits d'accès en écriture. Ces droits leur sont retirés à l'échéance du mandat.</i></p> <p><i>Un fournisseur désigné à cet effet par la communauté doit revoir toutes les modifications qui concernent des éléments essentiels du code. Il répond de la qualité technique de l'application et doit également vérifier les modifications approuvées par la communauté sur le plan du contenu.</i></p>
b	a	b	<p><i>Les droits des committers (droits d'accès en écriture au code) sont réservés aux fournisseurs membres de la communauté. Ces fournisseurs choisissent leur mode d'organisation et répondent collectivement de la qualité du code.</i></p> <p><i>Lorsqu'un membre de la communauté confie à un fournisseur externe le soin de modifier ou de développer l'application, un fournisseur membre de la communauté doit se charger de l'audit du code. Il est rémunéré pour ce faire par le mandant.</i></p> <p><i>Les modifications touchant au noyau de l'application doivent faire l'objet d'un audit par un deuxième fournisseur, lui aussi membre de la communauté.</i></p> <p><i>Les fournisseurs membres de la communauté sont également responsables de l'audit des contributions externes et du nouveau code technique acceptés par l'association sur le plan du contenu.</i></p>
c	-	-	<p><i>Les droits d'accès en écriture appartiennent à l'origine aux développeurs et développeuses du fournisseur mandaté. Des droits d'accès en écriture sont octroyés à tous les développeurs et développeuses qui contribuent de façon soutenue au projet pendant plusieurs mois et qui possèdent les compétences sociales requises.</i></p> <p><i>Ces développeurs et développeuses gardent en principe leurs droits d'accès en écriture s'ils changent d'employeur ou si le canton de Berne choisit un autre fournisseur. Ces droits ne deviennent caducs que si leurs titulaires y renoncent de leur propre gré ou si la majorité des autres committers décident de les retirer.</i></p> <p><i>Le canton de Berne a le droit de désigner committers certains dévelop-</i></p>

			<p><i>peurs ou certaines développeuses des entreprises qu'il mandate. Il n'a pas le droit de retirer les droits d'accès en écriture sans motifs.</i></p> <p><i>Les audits du code sont effectués par au moins une personne qui a la qualité de committer. Les committers choisissent leur mode d'organisation. Le canton de Berne s'engage à les indemniser pour l'audit du code, pour autant que cette charge soit supportable.</i></p>
--	--	--	--

4.5 Partage des frais et fournisseurs

GP	F	C	Proposition
a	a	a	<p><i>Les fournisseurs sont choisis périodiquement dans le cadre d'un appel d'offres OMC ou d'une procédure de gré à gré (en fonction de l'importance des prestations de maintenance).</i></p>
a	b	-	<p>Important : il convient de vérifier au préalable si cette procédure est autorisée par les dispositions relatives au droit des marchés publics.</p> <p><i>Le canton de Berne fait du fournisseur XXX son partenaire stratégique et vise à établir avec lui une collaboration durable.</i></p>
b	a	a	<p><i>Chaque membre de la communauté confie sous sa propre responsabilité à un ou plusieurs fournisseurs de logiciel le soin de réaliser les adaptations et les développements qu'il souhaite.</i></p> <p><i>Si des adaptations sont demandées par plusieurs membres, ces derniers arrêtent la répartition des coûts au cas par cas. Le membre qui supporte la plus grande partie des coûts choisit et mandate le fournisseur.</i></p> <p><i>(Règles relatives à la maintenance du logiciel)</i></p>
b	a	b	<p><i>L'association choisit périodiquement au nom de ses membres les fournisseurs dans le cadre d'un appel d'offres OMC ou d'une procédure de gré à gré (en fonction de l'importance des prestations de maintenance).</i></p> <p><i>Les coûts sont répartis suivant la clé suivante : (à définir : suivant la population, les utilisateurs et utilisatrices, etc.)</i></p>
b	b	a	<p>Comme « b-a-a », complété le cas échéant par une contribution des fournisseurs impliqués. Exemple :</p> <p><i>Chaque société logicielle membre de la communauté s'engage à investir à ses propres frais au moins XY jours de travail par an dans le perfectionnement, la mise à jour technologique ou la commercialisation de l'application.</i></p>
b	b	b	Comme « b-a-b », complété le cas échéant de la même façon que pour le cas de figure « b-b-a ».
c	a	-	Comme « a-a-a » complété comme suit :

			<p><i>Les améliorations fournies par les fournisseurs potentiels et leurs activités au sein de la communauté sont prises en compte lors de l'évaluation.</i></p> <p><i>Les changements souhaités par des tiers ne sont pas financés par le canton de Berne, mais mandatés et payés par ces derniers.</i></p>
c	b	-	<p>Comme « a-b-a » complété comme suit :</p> <p><i>Il est attendu des fournisseurs qu'ils collaborent activement au sein de la communauté, même si les frais qu'ils consentent à cet effet ne sont pas directement pris en charge.</i></p> <p><i>Les changements souhaités par des tiers ne sont pas financés par le canton de Berne, mais mandatés et payés par ces derniers. Le fournisseur se déclare disposé à les réaliser pour un prix équitable, à la demande des tiers.</i></p>

4.6 Commercialisation

GP	F	C	Proposition
a	a	-	<p><i>Nous n'entreprenons aucune activité pour commercialiser l'application, mais la distribuons sur les plateformes de logiciels libres. Nous informons les organes spécialisés que nous avons distribué l'application.</i></p> <p><i>Si des personnes intéressées prennent contact avec nous, nous envisageons la possibilité de fonder une communauté ou de continuer à travailler avec leur version du logiciel.</i></p>
b	b	-	<p>Pour compléter ce qui précède.</p> <p><i>Les fournisseurs ont le droit de commercialiser l'application et de chercher des personnes intéressées. Ils peuvent indiquer le canton de Berne comme référence. Nous sommes conscients que des versions différentes du logiciel apparaîtront.</i></p>
b	a	a	L'application est commercialisée par les membres ou par l'association.
b	b	a	L'application est commercialisée par les fournisseurs.
b	-	b	L'application est commercialisée par l'association et par les membres de celle-ci.
c	-	-	<p>L'application est commercialisée par les fournisseurs ou la communauté décide de ne pas la commercialiser expressément.</p> <p>Variante sans commercialisation expresse :</p> <p><i>Nous n'entreprenons aucune activité pour commercialiser l'application, mais la distribuons sur les plateformes de logiciels libres. Nous informons les organes spécialisés que nous avons distribué l'application et que nous</i></p>

			<p><i>souhaitons leur collaboration.</i></p> <p><i>Nous faisons une démonstration de notre logiciel pour les personnes intéressées et essayons de les associer à la définition de la feuille de route.</i></p>
--	--	--	--

4.7 Gestion des contributions externes

GP	F	C	Proposition
a	-	-	<p>Il faut aussi traiter les demandes de conseil et les contributions externes lorsque le canton reste entièrement maître du logiciel.</p> <p>Nous proposons la teneur suivante :</p> <p><i>Gestion des bogues signalés</i></p> <p><i>Nous essayons de reproduire les bogues signalés par des tiers et leur demandons des précisions si nécessaire. Si la reproduction aboutit, le canton de Berne charge le fournisseur de remédier à l'erreur. Si l'erreur ne peut être reproduite ou si les frais à consentir pour y remédier sont disproportionnés, nous présentons nos excuses au tiers et lui demandons s'il peut déposer une pull request pour remédier à l'erreur.</i></p> <p><i>Gestion des demandes de conseil</i></p> <p><i>Nous répondons à toutes les demandes reçues. Si la réponse à une demande exige de nous un effort déraisonnable et s'il ne semble pas utile d'y consacrer davantage de temps, nous présentons nos excuses en invoquant nos ressources limitées et prions le tiers de prendre contact avec un fournisseur afin de recevoir de plus amples conseils.</i></p> <p><i>Gestion des pull requests</i></p> <p><i>Nous examinons avec bienveillance toutes les pull requests. Afin d'éviter des efforts inutiles au contributeur, nous l'avertissons d'emblée lorsqu'un élément est incompatible avec notre feuille de route ou lorsqu'il n'est pas certain que nous acceptions sa contribution. La décision d'accepter une contribution est du seul ressort du canton de Berne, qui applique des critères techniques. Il accepte toujours les corrections des erreurs qui ont passé l'épreuve de la procédure de révision.</i></p> <p><i>Gestion des forks</i></p> <p><i>Nous donnons notre soutien aux forks de notre application. Quiconque utilise l'application doit en créer un embranchement. Nous examinons une fois par année au moins les forks créés et reprenons les développements de qualité qui nous sont utiles.</i></p>

b	-	<p>-</p> <p>On entend ici par contributions externes les contributions faites par des personnes n'appartenant pas à la communauté définie (qui ne sont pas membres de l'association). Solutions différentes à celle exposée ci-dessus :</p> <p><i>Gestion des forks</i></p> <p><i>Nous nous employons à éviter des forks (durables) de l'application, lui préférant l'adhésion à la communauté des personnes intéressées. Au sein de la communauté, nous œuvrons afin que tous les membres utilisent la version principale.</i></p>
c	-	<p>-</p> <p><i>Nous attachons de l'importance à la création d'une communauté performante et au développement d'une culture ouverte.</i></p> <p><i>Gestion des bogues signalés</i></p> <p><i>Nous essayons de reproduire et de corriger les bogues signalés et demandons à cette fin que les personnes qui les signalent collaborent avec nous. Si elles sont en mesure de le faire, elles doivent présenter directement une pull request permettant de corriger les erreurs.</i></p> <p><i>Gestion des demandes de conseil</i></p> <p><i>Nous traitons toutes les demandes dans un délai d'une semaine au maximum. Nous traitons en priorité et approfondissons les demandes des contributeurs actifs. Nous essayons d'associer les autres membres au traitement des demandes de conseil.</i></p> <p><i>Gestion des pull requests</i></p> <p><i>Nous essayons d'obtenir le plus grand nombre de pull requests de qualité. Nous associons tous les contributeurs actifs aux audits et à l'appréciation des pull requests sur le plan informatique et sur celui du contenu. Lorsque des contributions prêtent à controverse, nous essayons de parvenir à une solution au moyen d'une votation officielle.</i></p> <p><i>Gestion des forks</i></p> <p><i>Nous essayons d'éviter autant que faire se peut que des forks soient nécessaires, car nous voulons pratiquer une gestion de la communauté active et ouverte. Si des forks sont créés, nous les examinons attentivement et essayons de restituer les développements et améliorations qui en résultent. Dans ce cas, nous demandons aux contributeurs de déposer une pull request.</i></p> <p>Les sites suivants fournissent d'autres indications utiles à une bonne gestion de la communauté : https://opensource.guide/code-of-conduct/ et https://opensource.guide/best-practices/</p>

4.8 Exploitation de l'application

GP	F	C	Proposition
b	-	-	En option, il est aussi possible que l'exploitation soit centralisée au niveau de l'association, en tant que solution de service logiciel. Il convient d'examiner si cette solution est judicieuse.
-	b	-	Il faut décider si le fournisseur du logiciel doit aussi en assurer l'exploitation.

5 Liste de projets de communauté existants

Nous indiquons ci-dessous des projets de communauté déjà existants, ainsi que des documents comparables, considérés comme de bonnes pratiques :

- GERES-Community (<http://geres-community.ch>)
Classification :
 - Gestion de produit : b) Élaboration collaborative
 - Fournisseur : plutôt a) Mandataire
 - Répartition des coûts : b) Mutualisation des coûts

Remarque : si le registre des communes GERES n'est pas un logiciel à code source ouvert, de nombreux aspects de son organisation sont cependant pertinents pour les logiciels libres. Documents : statuts (publics) et règlement d'affaires (sur demande)

6 Procès-verbal du document

N° de document 287257

Auteur Joos Thomas, FIN-KAIO-AP-SW

Suivi des modifications

Version	Nom	Date	Remarques
1.0	Mario Siegenthaler	14.05.2018	Première version
0.2	Thomas Joos	22.05.2018	Révision
0.3.-0.4	Mario Siegenthaler	08.06.2018	Révision
0.5	Thomas Joos	13.06.2018	Finalisation
1.0	Thomas Joos	04.07.2018	Version définitive après approbation par le PB
1.1	Thomas Joos	12.09.2018	Corrections des erreurs, modifications mineures et formulation inclusive
1.2.-1.3	Stefan Schneider	13.09.2018	Révision
1.4	Thomas Joos	18.09.2018	Finalisation

Contrôle

Version	Service	Date	Visa	Remarque
0.5	Thomas Joos	13.06.2018	Tjo	---
0.6	PB	02.07.2018	PB	Approbation par le Portofolio board
1.2	État-major	13.09.2018	ssc	Contrôle linguistique

Validation

Version	Service	Date	Visa	Remarque
1.0	Service / responsable de	25.06.2018	mwe / rae	---

	domaine			
0.6	PB	02.07.2018	PB	Validation par le Portofolio board