



À propos

*The world is open by default ;
technology is just catching up.*
James CORBETT (*open journalist*, manager
de la communauté Corbett Report,
#fOSSa2012)

À qui s'adresse ce guide ?

Vous souhaitez diffuser largement vos résultats, pérenniser une base de code, augmenter le nombre de développeurs contribuant au code, accroître le nombre de membres de la communauté, mais vous ne savez pas comment vous y prendre ? Vous commencez à manquer de temps et de ressources pour tout gérer ? Alors vous avez probablement besoin de renforcer vos compétences dans le domaine de la gestion de communauté et ce guide est fait pour vous ! Il est centré sur la création et l'animation d'une communauté autour du développement de logiciel libre et propose à ce titre une aide méthodologique tournée vers la gestion des communautés.

Ce guide a été initialement développé pour répondre aux attentes des chercheurs et ingénieurs de développement logiciel d'organismes publics

tels que Inria, le CNRS, l'INRA, etc. Cependant les réponses génériques qu'il apporte sur les méthodes de gestion de communauté sont applicables dans d'autres contextes.

Ce que n'aborde pas ce guide

Les questions liées aux modèles économiques du libre et qui ont déjà donné lieu à une publication par les auteurs de ce guide (Ribas, Guillaud, Ubéda, 2013) ne sont pas abordées ici.

De même, la description des licences *open source*, qui est l'objet d'une abondante littérature, n'est pas notre sujet.

Exemples par la pratique

À la fin de ce guide sont présentées deux études de cas réels. Les actions menées et leur impact y sont décrits afin d'illustrer la mise en œuvre des méthodes proposées. Nous avons aussi inséré deux interviews de *community managers* afin de vous faire partager leurs expériences en la matière et donner leurs points de vue sur le sujet.



Introduction



Community over code!

Ross GARDLER, Vice-Président de Apache
Software Foundation, #FOSSa2009

Pour les chercheurs travaillant dans les sciences du numérique, le logiciel joue un double rôle en étant à la fois un objet d'étude (langages, compilateurs, systèmes d'exploitation, *frameworks*, etc.) mais également un moyen de contribuer à une production scientifique (pour renforcer la véracité, la validité et la robustesse des résultats scientifiques). Le monde de la recherche a une inclination naturelle à utiliser le logiciel libre pour le développement et la diffusion des connaissances et des technologies, l'accroissement de la visibilité des chercheurs et la constitution de masses critiques de développement. Mais pour en tirer profit, il est nécessaire d'organiser la communauté autour du code et de trouver les motivations qui réuniront les personnes autour de valeurs partagées.

Ce guide traite de la création et de l'animation d'une communauté autour du développement de logiciel libre. Le présent document n'a pas d'autre prétention que d'être une aide méthodologique abordant la question des communautés, en résumant les concepts, prodiguant des conseils et présentant un ensemble de bonnes pratiques. Il décrit des comportements et des processus permettant d'appréhender les concepts généraux



autour des communautés dans le domaine du développement de logiciel libre. Il regroupe des éléments bien connus mais les organise de manière à accompagner les équipes de recherche autour de cette question, et se veut avant tout pratique. À la fin de ce guide sont proposées des références permettant d'aller plus loin ainsi qu'une bibliographie.

Mais avant d'aller plus loin, il est important de rappeler quelques définitions autour des communautés (il est primordial de comprendre comment une communauté émerge et quelle est sa dynamique). Le prochain paragraphe propose donc une définition du terme *communauté*, puis nous abordons les grandes lignes de la méthode exposée dans ce livre.

Qu'est ce qu'une communauté ?

Une communauté est un groupe de personnes partageant des préoccupations, une passion ou un intérêt communs, qui s'organise et engage des actions concertées afin de résoudre un problème partagé.

On distingue plusieurs types de communautés : les communautés d'apprentissage, les communautés d'intérêt, les communautés de « passion », les communautés de pratique.

Une communauté d'apprentissage est un groupe de personnes qui partagent des valeurs communes et qui sont activement engagées dans un processus d'apprentissage mutuel (par exemple, le projet Dessine moi un robot ¹, ou encore la communauté Pixees ²). De telles communautés sont devenues un modèle initial pour les autres types de communautés. Les membres d'une telle structure ont un sens fort de loyauté envers le groupe qui alimente leur désir de continuer à travailler et à aider les autres, de produire un effet sur la communauté en étant non seulement réactif mais actifs. Une communauté d'apprentissage laisse une liberté suffisante pour que les membres puissent exprimer des opinions personnelles, demander de l'aide ou des informations spécifiques et commenter des événements.

Une communauté d'intérêt peut être vue comme un groupe de personnes partageant des thèmes qui ne nécessitent pas de structure formelle mais qui mènent des discussions tournées vers la collaboration et le partage des

1. <https://dmlr.inria.fr/>.

2. <https://pixees.fr/>.



connaissances (comme les ingénieurs systèmes et réseaux participant aux conférences JRES¹, les ingénieurs de recherche et développement logiciels participant à la liste de diffusion DEVLOG²). Elles peuvent consister en des groupes de personnes faiblement connectées sans grande implication en terme de production commune. Elles restent cependant bien au fait de leurs thèmes et posent des questions.

Une communauté de « passion » est basée sur un groupe de personnes dont les activités, la gouvernance et la structure sont les plus riches et les plus formelles. Les membres y jouent un rôle particulier (ex. conseil en sécurité de réseau), ils aident activement les autres membres à coller à leur rôle et visent à maîtriser la discipline. Les communautés de passion sont davantage tournées vers le partage de connaissances dans un domaine donné que sur la réalisation d'un projet particulier. Leurs membres peuvent intervenir dans des communautés de pratique en tant qu'experts ou consultants.

La structure d'une communauté de pratique est moins formelle et basée sur des spécialités communes (ex. les groupes d'utilisateurs de GNU/Linux³). Les membres ont un rôle ou une spécialité spécifique (ex. la sécurité, le développement J2EE) et se focalisent sur le développement de l'expertise et des compétences qu'ils ont dans ce rôle ou cette spécialité. Un facteur essentiel de motivation est l'apprentissage de la spécialité et la résolution de problèmes.

La communauté de pratique, concept hérité des sciences humaines et sociales et antérieur au développement de l'informatique, présente trois caractéristiques spécifiques :

1. elle permet de résoudre collectivement et de manière efficace un problème partagé et, durant ce processus
2. elle permet à une proportion importante de ses membres de suivre un processus d'apprentissage
3. les activités et interactions qui se déroulent en son sein ont pour effet de produire des connaissances nouvelles. Elles sont donc productrices d'innovation.

1. <https://www.jres.org>.

2. <http://devlog.cnrs.fr/>.

3. <http://www.linux.org/groups>.



La méthodologie présentée dans ce livre a été ébauchée à partir de ce modèle et appliquée à ce type de communauté, mais pas seulement. . .

Les membres de ces communautés ont des interactions quasi quotidiennes avec des personnes extérieures au projet – ces dernières forment parfois entre elles des réseaux d’entraides, produisent des communautés. . . communautés d’apprentissage ou d’intérêt. N’oublions pas ces communautés : sans ces « utilisateurs » nous ne pourrions pas disséminer nos résultats de recherche, ni nos productions de logiciels. Ces communautés sont vitales pour la pérennité des projets de développement logiciel. Des conseils afin d’attirer leur attention, les encourager à utiliser votre logiciel, à rejoindre votre communauté et à contribuer, sont exposés tout au long de ce guide.

La communauté des développeurs dans le monde du logiciel libre

La collaboration a toujours été très présente chez les développeurs, particulièrement depuis l’arrivée d’Internet. Les communautés de développeurs ont montré leur efficacité dans de multiples domaines, et cela est tout particulièrement vrai pour le développement de logiciels libres qui se prêtent, par construction, aux effets précédemment mentionnés : résolution de problèmes, montée en compétence des développeurs et production d’innovation.

Pourquoi créer une communauté ?

L’existence d’une communauté active autour d’un code, qu’il s’agisse de développeurs ou d’utilisateurs, permet de disposer de ressources pour le faire évoluer, pour améliorer sa qualité et garantir une bonne réactivité face aux évolutions, qu’elles soient scientifiques, techniques ou d’usages. Une autre caractéristique essentielle des communautés étant de produire des connaissances nouvelles et de stimuler l’innovation, le partage d’informations et de connaissances, ainsi que les interactions au sein du groupe permettent de faire émerger des idées, de lancer de nouvelles pratiques, de faire naître des usages en rupture. Plus la communauté s’étend



et se diversifie, plus elle devient un gage de qualité et de dynamisme. Une communauté est également un puissant vecteur de diffusion permettant à un logiciel d'acquérir visibilité et réputation, ce qui peut constituer un gros atout en cas de collaboration avec le monde économique.

Par où commencer ?

Le point de départ consiste à qualifier le besoin initial afin d'évaluer la pertinence de votre projet de communauté : s'agit-il d'un besoin susceptible d'être partagé par un nombre suffisant de personnes, qu'il s'agisse de développeurs et/ou d'utilisateurs ?

Si le besoin est identifié et semble suffisamment fédérateur, alors la première étape consiste à définir le projet puis à publier le code du logiciel (si ce n'est pas déjà fait) afin d'améliorer son développement, en faire la promotion afin d'amorcer la communauté, puis la faire croître et l'entretenir. Se lancer dans la création d'une communauté demande du temps et des ressources, aussi la décision de tenter cette aventure doit-elle être prise avec discernement et de manière non systématique car il faut s'assurer, avant de se lancer, que les contraintes engendrées n'excéderont pas les avantages espérés.

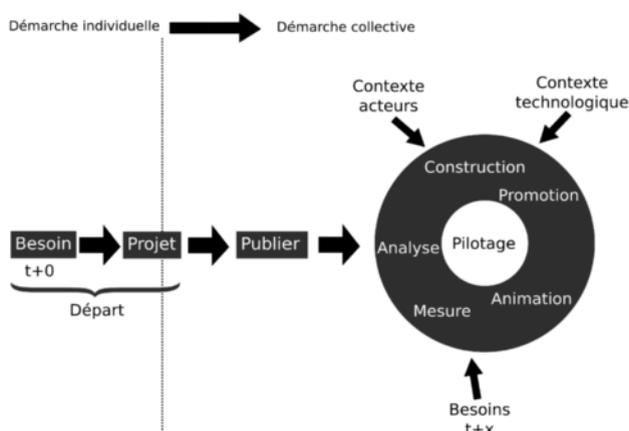
Une communauté naît rarement de façon spontanée et il faut être prêt à consacrer du temps pour la faire émerger, la faire croître puis maintenir la motivation de ses membres. La base de toute communauté est le partage, ce qui implique en contrepartie une certaine dilution du contrôle qu'il faut évaluer en fonction de ses objectifs (diffusion d'une méthode, standardisation, transfert technologique, promotion d'une école de pensée, etc.).

Il faut ensuite définir clairement ce que l'on souhaite partager et la forme que doit prendre ce partage. Cela permet de déterminer une architecture de code qui facilitera le type de partage envisagé (dans son entier, par module, ou autres) et par un choix pertinent de licence. On se focalise très souvent sur le choix de cette dernière, mais l'organisation du code et, plus tard, le mode de gouvernance autour de ce code sont tout aussi importants.

L'organisation et les outils d'animation constituent l'architecture de participation de la communauté. L'ensemble des valeurs et des objectifs partagés par cette communauté sont les ressorts qui l'animeront. Les valeurs



peuvent être scientifiques, techniques, économiques, éthiques ou sociétales. Dans la pratique on retrouve souvent une combinaison de ces valeurs qu'il faut hiérarchiser autour d'une valeur principale qui définira la communauté. Négliger cette étape de réflexion sur les objectifs et les valeurs est l'une des premières causes d'échec. En effet, la première chose que feront les participants potentiels à cette communauté sera d'évaluer l'adéquation des valeurs qu'elle affiche par rapport aux motivations qui les animent.



Les différentes étapes

Dynamique de la méthode

Ce guide propose un mode d'emploi de la gestion des communautés à base de bonnes pratiques. La méthode proposée, itérative, comprend les étapes suivantes : définition, analyse du contexte, structuration du projet, mise en œuvre de l'architecture de participation, publication du code, promotion du logiciel, animation de la communauté, suivi et évolution.

- 1 Program 'Construire une communauté.c'
- 2 /* LICENCE is CC-BY-SA */
- 3 #include "guide gestion de communauté.h"



```
4   #define rêve = undefined
5   #define projet défini = undefined
6   #define inventaire = undefined
7   #define objectifs valeurs = undefined
8   #define gouvernance = undefined
9   #define habitat numérique = undefined
10  #define répondre à un besoin partagé = true
11
12  void setup() /* Tâches à faire séquentiellement */
13  {
14      rêver(); /* optionnel */
15      définir projet();
16      faire inventaire();
17      définir objectifs et valeurs();
18      faire veille autour du projet();
19      définir habitat numérique();
20      mise en oeuvre habitat numérique();
21  }
22
23  void loop() /* Tâches pouvant être effectuées en
24             parallèle */
25  {
26      coder();
27      faire promotion projet();
28      animer communauté();
29      augmenter nombre de membres();
30      faire veille autour du projet();
31      suivi et évolution du projet();
32  }
```

Le chapitre 1 concerne la première étape, la définition du projet. En effet, il est primordial de répondre aux questions suivantes : quel est le concept ? à quel(s) besoin(s) répond-il ? quels sont les objectifs ? quelles sont les ressources disponibles ? en somme, de quoi je pars et vers où je veux aller.

