

# Build a FLOSS community from scratch?

## Ways of building open source projects and pitfalls to avoid

**Stéphane Ribas**

Technological Development Department, INRIA, Montbonnot, France

email: [stephane.ribas@inria.fr](mailto:stephane.ribas@inria.fr)

### Paper type

Case study: Case studies describe actual interventions or experiences within organizations. They may well be subjective and will not generally report on research. A description of a legal case or a hypothetical case study used as a teaching exercise would also fit into this category.

### Abstract

Over the last decades, computing information technologies have fostered the development of new collaborative ways of working in the R&D domain, often evolving towards the creation of an organised community. The increasing connections at international level raise a need for remote collaboration around the world that emphasises the need to organise the creation and sustainability of such communities. Moreover, Free/Libre Open Source Software (FLOSS) is becoming more and more present in the IT landscape changing drastically the rules of the game (technical and business). However, many people underestimate the importance of communities over code: communities could become a booster for those OSS pieces of code if they are conducted properly. Derived from Stan Garfield and Richard McDermott academic researches on communities, this article presents an approach to build and sustain communities in the FLOSS area. The method, based on the five main steps described below, has been assessed on two European projects (AspireRFID and QualiPSo). Results are reviewed and we believe that transparency, trust, people commitment are keys that pave the way to success. Launching a community and developing it successfully is not “I code, I publish, I exist” but relies on a structured approach. By presenting such a method, we hope to foster anyone who would like to launch a community to throw himself into the adventure in a more productive manner.

### Keywords

Free/Open Source Software, Collaborative Environment, Communities, Knowledge Sharing, Methodology

## 1. Introduction

*“Our passion about Open Source Software (OSS) became quickly unmanageable and ends up in growing a lonely business into a worldwide OSS community almost overnight! At last, we did not alone believe in that OSS development born in our garage a few years ago and got rapidly help from around the world. Enthusiastics rushed to propose contributions and wanted to commit around the clock. Overwhelmed by that wave of free services, we moved on the scale to sceptical, satisfied, happy, anxious and eventually desperate. We wanted to share our ideas and hopes about that piece of codes and we succeeded well above our expectations and that unmanageable success is what killed our dream and us. Today we are back to two people around an incredibly complex piece of code which will take us centuries to understand and debug.”* by anonymous former bazaar owner.

This testimony, although fake, could illustrate some situations in the OSS landscape and among the OSS developers. That story may definitely reflect a lot of effort wasted in the OSS domain by coders who only wanted to share their passion. Indeed, OSS communities host most of the visible OSS developments and without them; growing software into an operational and mature (commercial) product is too hard a task for a group of people no matter their excellence. Of course some OSS products have been able to generate enough momentum to develop a well-established OSS community but how many have disappeared mined by problems not always technical? No surprise, the building and sustainability of a successful OSS community relies on some basic rules and processes which cannot be overlooked if you want to ensure better chances of success. Many people underestimate the importance of communities over code: communities could become a booster for those OSS



pieces of code if they are conducted properly.

Based on the Stan Garfield and Richard McDermott approach [1, 2], author experience and the creation of the different communities, this paper describes a methodology to build and sustain an OSS project.

Stan Garfield's approach is built around the definition of goals, actions and metrics and put a lot of emphasis on organizations' interests (e.g. public organizations are more interested in eLearning while companies aim at Total Cost of Ownership reduction). Richard McDermott's approach focuses on Community of Practice using surveys to gather needs and interests. As a result of this activity, Richard McDermott proposes to build initiatives and projects that would fit the community motivations and help to build the network as a side effect.

Our approach, based on the five main steps described below, has been assessed on two European projects [3, 4] and results are reviewed at the end of this document.

Definitely OSS focused; this step-by-step guide is mainly addressed to community managers and pinpoint common issues that may poison the growth of the community. Killer hints are provided as praise to "the Cathedral and the Bazaar" masterpiece [5] and are, somehow, based on the author experiences.

## **2. What is a community?**

A community can be considered as a group of people who share the same interests, the same concerns or the same passion. A community can also be based on roles or specialities (e.g. network administrator, Linux user groups, modular synthesiser user group). Those people deepen their understanding by sharing their knowledge, solving problems for others, interacting regularly with other community members, asking and answering questions, and reusing good ideas [1]. They are keen to participate actively and a new virtual identity is created based upon a strong social linkage that hopefully produces a collective outcome [6].

Briefly, we can distinguish four important kinds of communities [1]: community of learning (e.g. Plume/RELIER project [7]), community of interest (e.g. Audiofanzine [8]), community of passion (e.g. System Administrators participating in the JRES 2010 Conference [9]), community of practice (e.g. Linux User Group [10]). Identifying which kind of group your project belong too, can help you to understand members motivation and draw your incentives to blossom collaboration [11, 12].

### **1.1 Community of learning**

A community of learning is a group of people who share common values and beliefs and are actively engaged in learning together from each other. Such communities have become the initial template for other types of communities. The participants in a learning community must feel some sense of loyalty to the group that drive their desire to keep working and helping others, influencing what happens in the community (active and not just reactive). A learning community has enough room to give the chance to the participants to express personal opinions, ask for help or specific information, and share stories of events [6, 13].

### **1.2 Community of interest**

A community of interest could be considered as a group of people that share topics that do not really require a formal community, but more threaded discussions for collaboration and knowledge sharing. These may be loosely connected groups of people with no strong commitment in terms of delivering something together. They stay well aware of the topics and ask questions [6].

### **1.3 Community of passion**

A community of passion is made up of a group of people with the richest and most formal set of activities, governance, and structure. Members have a particular role (e.g. Network Security Advisor), actively help others members to fit and bloom into this role, and aim at mastering the discipline [6].

### **1.4 Community of practice**

The structure of a Community of practice is less formal and is based on common work specialities. Members have a particular role or speciality (e.g. Security) and focus on developing expertise and skills in this role or speciality. An important key motivation factor is to learn about the speciality and solve problems [6]. The methodology discussed in this paper was drafted from and applied to this type of community.

## 2. Method for building and maintaining a community of practice from scratch

The method goes through several steps starting from “Dream” up to “Monitor”. Those steps could be organised into 5 main categories [2, 12, 14, 15]: Analyse, build, promote, sustain, monitor.

### 2.1 Analyse

The goal of analyse phase [1, 14, 15] is to start the process and gather all relevant information to understand what we want to do, where we want to go, why and how. As already mentioned, a community could emerge from scratch through the passion of a small, enthusiastic group but raising the chance of success needs a bit more organisation and processes.

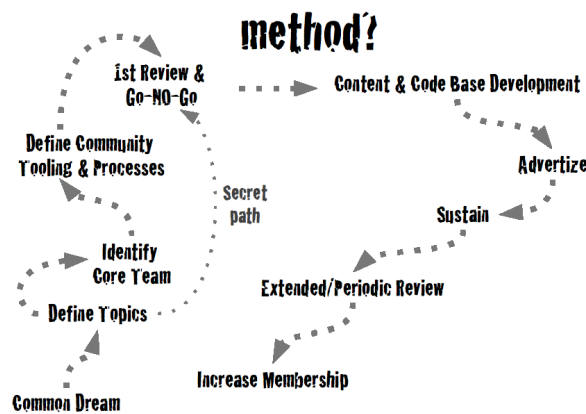


Figure 1. Method to create and sustain a community

In this phase, you should start by dreaming about: how ideally would your community work? How should it be organised? Who will lead it and make things happen? Let your imagination flows freely. It is also very important to evaluate if your project make sense (e.g. answer to a need) as this will help to increase the community membership and dissemination process.

Identifying topics and sub topics will allow clarification of your ideas as well as making sharing ideas with your colleagues easier. At this point, it is important to identify your core team to know who is committed and willing to actively contribute to the journey.

**Killer hint #1:** Restrict enthusiasm, free expression and emerging ideas from your colleagues.

**Killer hint #2:** Among the core team, you will need to identify a community leader; choose a person who is non technical background, who does not know the subject, with poor communication and listening skills, and lack of decision making capabilities. Most of all do not legitimate his role to the rest of the community and always keep unclear his responsibilities.

Once the team defined, you should address the means of operation; basically how are you doing to work together? How are you going to work with external contributors? What is your architecture of participation? This will involve further imaginative thinking and perhaps brainstorming sessions and information exchange. These community processes will encompass governance and intellectual property management [16], communication, development life cycle, collaborative environment, collaboration guidelines, dissemination and promotion.

**Killer hint #3:** Propose non flexible and very formal procedures; establish a strong top-down privileged oriented community; use terms like ‘product’ instead of ‘project’, ‘governance rules/by laws’ instead of ‘governance principles’; stick to one business model and never change it whatever may happen; force developers to follow classic V project development life cycle; propose complex property rights policies and procedures; ensure that communication, dissemination and promotion initiative requires your despotic approval.

Now initial jottings will need to be replaced by more sophisticated tools (**Killer hint #4:** preferably non OSS tools).



Identify (and use) community tools to:

- (1) Foster collaboration using source code management systems, bug tracker, webinar, wiki, chat, polls, meetings project management facilities...
- (2) Structure your content using content management system, wiki, website portal...
- (3) Facilitate your dissemination activities (promotion and awareness) using community directory listing, mailing list, RSS feeds, newsletter, blogs...

Those tools focalised around a forge will enable daily operation and communication, being a central repository for the community and for the project.

Complementary to these tools, FLOSS social engine that integrate in one place, several interesting services (wiki, blogs, forum, rss, document repository, social network engine, etc.) are emerging. The integration into one place if well managed can leverage drastically information sharing (or knowledge sharing) among the different working groups of a community and facilitate the management of your “online-communities”. Among those initiatives, we can mention Nosferos [17], Elgg [18], Open Atrium [19] and Buddy Press [20].

Through this process, you should arrive at a better idea of what you intend to build and achieve. At this stage stop and research whether there are any existing communities or projects sharing the same concerns and topics: You may integrate those communities rather than building and your own collaboration environment.

As a consequence, a go-no-go decision (e.g. shall we create our initiative from scratch or shall we join another community?) should be made based on all the collected information and the planned roadmap. What ever decision you take, you will need to explain it carefully and identify what the main showstoppers are given the current context to avoid asking the same questions again later when the original idea is revisited (**Killer hint #5: Never take that decision formally**).

Providing you selected the “from scratch” option, you proceed now to the second main phase: **Build**.

## 2.2 Build

This phase is where you put bricks and mortar together. You start coding, create documentations (e.g. User Guide, API Development Guide, User Case, etc.), setup your portal (good project/website name, clear mission statement, FLOSS license used, development status, how to install the software, where can we download the software’s, display some screenshots and a demo, how can people contribute, etc.), release packages, publish project tasks and roadmap, manage volunteers and external contributions [21, 22, 23].

**Killer hint #6:** *Avoid any open attitude towards members; propose as often as possible to re invent the wheel. State that globalization/localisation/skin adaptation add-ons are useless. Don’t publish roadmap neither your project security threads.*

Keep in mind that relative “successful” OSS projects are often initiatives that **propose open and decentralized architecture to allow external people to participate** to your project (e.g. Eclipse and Mozilla Firefox plug-ins architecture).

**Killer hint #7:** *Impose closed monolithic architecture software and avoid participation from community members to any decision making process.*

Simple and flexible processes will drive your daily activities: community governance, intellectual property policy, project life cycle, communication rules and collaboration guidelines.

Having “completed” those tasks it is time to advertise your existence.

## 2.3 Promote

You have a team in place, a code base, good documentations and a portal where people can find all the information needed to either contribute or use your code. Basically, something to show!

Your community is established so you should advertise it:

- (1) Use your own network and ask your members to spread the word,
- (2) Submit articles to conferences and forums that reach your audience,

- (3) Use a mailing list and send broadcast messages,
- (4) You can publish news in (local and international) newspapers and webzines.
- (5) You may also use Social Networks that are now widely spread (e.g. LinkedIn, Twitter) to publicise your community.

Obviously, you may start first by building a list of contacts (communities that may share same topics or interests, your own contact network, use Google alert service, participate to conferences and event to increase your own network, collaborate with other FLOSS initiatives, etc.) and then uses several communication channels to promote your existence [24] and increase your membership as a side effect (see Figure 2).

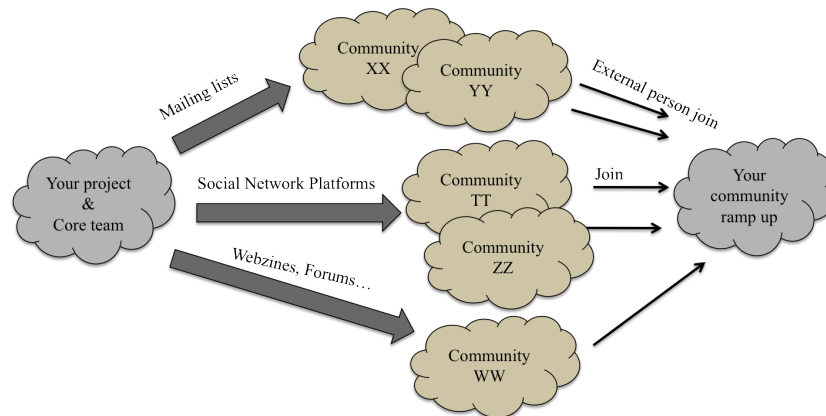


Figure 2. Use several communications channel may help your promotion.

**Killer hint #8:** *Do not promote, reputation will grow by itself thanks to product excellence.*

Once your community is up and running and has reached a certain level of recognition, you may enter the next phase **Sustain**.

## 2.4 Sustain

You need to keep the momentum and, as a consequence, community leader(s) should have devoted time and leadership to animate and stimulate the community [1]. We recommend to hold periodic events (e.g. project summit, general assembly), hold periodic phone conferences and publish minutes notes regularly, be active on forums, and sometimes even copy interesting topics coming from external sources into your news, blogs and forums. We recommend you to stay transparent in any decision you make as well as listening to people (enough room to deal with any topic/matter) [2].

You also need to reach a critical mass of members [25] that will leverage your daily operation (System Administration, Webmaster, Project Manager, Bug Trackers, Bug Fixers, Information Integrator, Facilitators, Translators...) thus you also need to increase the number of people in your community: propose initiatives (e.g. coding contest, code camp, project summit, etc.), use virtual social networking, show your community values on your portal, explain how people can participate and become a member, offer incentives to join. Do not hesitate to cross boundaries and collect good ideas from other initiatives. Identify and encourage your members to advocate and promote your project identity but also contribute to the portal content and initiatives.

Two other key factors are for external users to contribute actively in a community [11, 26, 27]: generating recognition and knowledge sharing. You could propose to show on your home page the recently registered users, and who are the top bug fixers. Next, you should devote resource(s) to organise the knowledge within the community; propose knowledge sharing facilities (e.g. webinars [28], workshops, use case, etc.). Keep in mind that FLOSS culture blossoms on the following roots: open attitude and free room for expression, transparency, trusts, listening, bottom up organisation, share and learn without boundaries.

**Killer hint #9:** *Never acknowledge contribution, recognise successes or award incentives*

As in the analyse phase, you may also have a look around and search for any existing communities or projects that may be interested by using your code base or that you may interested in co-operating with you. Identify those communities, approach their leaders and propose collaboration [2]. When project budget is a constraint, exploring potential cooperation and synergies with other projects is often a very good mean to build a wider

network of contacts that could lead to increase momentum, visibility and membership. In another words, **join forces with other projects and pay back for the help!**

***Killer hint #10:** Force other OSS identified communities to join and merge your community.*

## 2.5 Monitor

In order to assess progress and trends, and validate results of your actions, we recommend that you define your own monitoring process rather than using predefined templates: identify the goal (why and what would you like to measure), define the metrics, collect data (analyse), identify problems and trends, define accordingly the actions to address both of them, execute the actions and re-assess the project health.

For instance, if you want to monitor the expansion of your web portal visits [29], you could analyse the number of visitors, the number of page viewed, the number of posts, the number of downloads, the number of RSS feed subscribers, the total user-generated content, etc. You can also monitor the behaviour of your visitors, the number of page view per visit, the number of post views per visitors, new registrations and first time visitors, repeated visitors, newsletter subscriptions, time spent on the website, etc.

We advise you to monitor the growth of your community, and the health of your project, through metrics such as number of bugs tracked, number of downloads, number of visits, number of post in forums and email exchanges, number of events organised, etc.

Finally, bear in mind that you should also monitor the community at large [30]: Are you in a community-starting phase? Or in a community-growing phase? Maturity phase? Or are you at the point where you are declining (or reviving)? A survey may help you to discover which point your community development has reached.

Literatures on evaluating communities and OSS project management can be found in [31, 32].

***Killer hint #11:** Never look back, never learn from your errors, and never monitor progress or trends.*

## 3. AspireRFID Community Project Experience report

FP7 ASPIRE project, partially funded by the EC [3], aims to develop and promote an open-source, lightweight, standards-compliant, scalable, privacy-friendly, and integrated middleware to ease the development, the deployment and the management of RFID-based applications and sensor-based applications.

The ASPIRE consortium wanted to create a vibrant community around its Open Source RFID framework and ensure a code base living beyond the European project lifetime funding (2011). Thus, the idea arose of creating a community of practice (and passion), around (and for) the code base.

We used this project as a test bed for our method, both through project auditing and project recommendations. At the same time, we assessed the method and fine-tuned the first version of steps and tools.

Being involved in both the ASPIRE and the OW2 community [33], we quickly found a positive match in the RFID domain: on the one hand, the ASPIRE project aiming at developing an OSS RFID framework, on the other hand, the RFID initiative was promoted by the OW2 consortium. The first one was taking off while the second one was looking for new contributions. It was a perfect fit and a great mutual benefit. As a consequence our Go-No-Go decision end up with a unanimous “let’s join the OW2 community”.

We may highlight three main results achieved:

- (1) **Speed up start up phase:** using the method, we made a review of existing communities and discover that OW2 consortium could help to create the AspireRFID community, and they have a lot of excellent community tools that fit well with AspireRFID community needs. OW2 propose a forge so that the developers start very quickly to code. OW2 also propose awareness and promotion tooling.
- (2) **Improve dissemination and promotion:** we applied the method and got results very quickly thanks to specific promotion actions such as papers, code camp, coding contest, project summit, user advisory board... We found out that OW2 was a good valorisation and promotion channel through its mailing list and other initiatives.
- (3) Finally, the dissemination and promotion successes increased the **exploitation outcome:** in a RFID



benchmark made by an industrial consortium ICOM [34], the ASPIRE framework was compared to commercial products. Despite the fact that the project was not finished, and the framework was under construction, ASPIRE was ranked higher compared to its competitors.

We conclude that the method helped to strengthen the product assessment, to publicise the product roadmap, to grow the product reputation and to ensure the longevity of the project.

#### 4. QualiPSo Community Project Experience report

The goal of QualiPSo project [4] is to define and implement technologies, procedures and policies to leverage the Open Source Software development current practices to sound, well recognized and established industrial operation.

To meet this goal, the QualiPSo core team is composed of 60 active contributors (European, Brazilian and Chinese ICT industry players, SMEs and academics), separated into working groups:

- (1) Development of "OSS and legal issues" methodologies and tooling,
- (2) Publication of "OSS technical interoperability" methodologies,
- (3) Development of a next generation forge based on Service Oriented Architecture (SOA),
- (4) Publication of OSS project management methodology (processes, tools, community, evaluation),
- (5) Development of an advanced integration testing tool suite for OSS projects, etc.

This project is partially funded under EU's sixth framework program (FP6), as part of the Information Society Technologies (IST) initiative and ended up in March 2011.

After a quick investigation on possible synergies with other OSS projects and organisations, the QualiPSo consortium took the decision to build their initiative from "scratch".

Thanks to our first relative "successful" experience about on-line communities, we have been invited – in January 2010 - to advice and participate to the growth on the QualiPSo visibility, membership and ensure that achievements will be exploited after the project completion. We started to analyse the QualiPSo project against the method previously described.

**"Analyse" phase:** Governance, roles and responsibilities, description of the activities, project/activities life cycle, intellectual property governance, collaboration guidelines, internal architecture of participation, face-to-face meetings and audio conference frequency policy, have been formalized and acknowledged by the European Commission. Compared to the ASPIRE project, we identified three kinds of groups; a community of practice and passion that can be associated to the QualiPSo core team members (the people who are developing new pieces of codes, new OSS methodologies, etc.) and a community of interest (that will made use of the results rather than modifying or improving them). We focused on the third kind of community rather than sustaining the momentum of the QualiPSo core team. Indeed, new project managers, with an assert leadership and long experience in OSS management, were injected in the governance as we stepped in.

**"Build" phase:** Many research achievements, publications, tools, documents, methods and piece of codes have been produced but only few team members were aware of it. **"Promote", "Sustain" and "Monitoring"** phases suffered drastically from this situation impacting the project reputation, and, slow down collaborations.

Here are the key actions we performed:

- (1) **Phase "Build":** We focused on reorganizing the achievements (and web portal) for a large public to easier assume ownership of the results and get support from it,
- (2) **Phase "Promote":** We identified a list of potential interested organisations, OSS communities, webzines, individuals, etc. We propagated news and achievements, among those contacts, on a regularly basis.
- (3) **Phase "Sustain":** The different project leaders ensured the internal momentum; we focused on building alliances with other OSS communities and organisations.
- (4) **Phase "Monitor":** We followed our own advices about web portal monitoring (see section Monitor)

As a complement to these actions, we implemented different tips described in this article (creation of a newsletter, participation to several conferences, publications in international magazines, organisation of workshops & webinars, co-organized an OSS conference [35], published surveys, offered goodies as an incentive to join, etc).

We may highlight four main results achieved:

- (1) **Grow of the community:** Membership increased from 60 members to more than 300 people today (20% of new members every month over a 10 months period monitoring),
- (2) **Building alliances:** Collaboration with 4 main OSS communities and organisations are flourishing (KDE Foundation, Linux Foundation, OPEES organisation, AfroCio – African Network in IT domain),
- (3) **Human adventure:** A sense of belonging as recently emerged within the QualiPSo core team members (most probably due to the continuum momentum, regular face to face meetings, leadership...). Secondly, new roles have emerged, champions and angels, the champions are referee to activities and achievements, where as angels tend to promote the use of QualiPSo results within their own organisations.
- (4) **Life after the completion:** Several research institutes and industrials have declared their wish to extend the achievements. Four main sub-projects will continue after the project completion; the next generation forge (QualiPSo Factory), the integration testing tool suite, the legal issues methodology, OSS project management methodology (QualiPSo Open Source Maturity Model).

## 5. Conclusion

As with every new process, the method needs adjustments and improvements, we are continuously integrating new findings; what we presented along this report is our latest update. By presenting such a framework, we hope to foster anyone who would like to launch a community to throw himself into the adventure safely and avoiding pitfalls such as authors did.

The main outcomes (that AspireRFID and QualiPSo experiences shares) can be summarised as:

- (1) The method should be applied right from the beginning to get the most benefits; both projects had started few months before we started to apply this method.
- (2) People commitment is the key. A community is based on people and no method or tools can replace willingness to contribute.
- (3) Devoting time for knowledge sharing is important, as well as finding people with a certain level of creativity and leadership.
- (4) Vision should be shared among the members and clearly publicized.
- (5) Methodology steps are not sequential; overlapping processes will boost the timeline.
- (6) AspireRFID and QualiPSo are FP Projects where the consortium has to deliver certain deliverables based on a roadmap already defined that can differ from the wishes of community members. Conflicts of interests must be identified and addressed openly as soon as possible.

Given what has been discussed, we can also claim how to kill a community. The recipe is pretty straightforward:

- (1) Prevent open discussion about vision and strategy
- (2) Never seek member's views and opinions,
- (3) Do not recognize contributions and do claim all achievements as yours,
- (4) Set up strict by-laws and rigid processes,
- (5) Mess up the development process and environment
- (6) Impose monolithic architecture software,
- (7) Fight with competitive OSS communities to gain market shares,
- (8) Trust the methodology "I code, I publish, I exist",
- (9) Block any community monitoring processes initiative,
- (10) Forgot about the OSS culture,
- (11) Set up an intellectual property right policy that will facilitate you making money on the back of your contributors,
- (12) Be obsessed by this unique priority: be the richest man in the world thanks to OSS model.

Without any bad intentions nor doing it on purpose, we applied those nasty rules by the past. No surprise, some of our OSS projects took longer to get successful, some others were stopped due to a lack of participation: **The building and sustainability of an OSS community relies on some basic rules and processes that cannot be overlooked if you want to ensure better chances of success.** Bear in mind that "relative" successful OSS projects are often initiatives that make sense (e.g. answer to a need); the author(s) promote the project existence to increase its development and; ensure that the community get enough interests to participate along the project life time.



## References

- [1] Implementing a Successful KM Programme (2007) Stan Garfield. Ark Group Australia. <http://stangarfield.googlepages.com/home>
- [2] McDermott Consulting: Community of Practice, <http://www.mcdermottconsulting.com>
- [3] OW2 AspireRFID Project <http://wiki.aspire.objectweb.org/xwiki/bin/view/Main/WebHome>, ASPIRE FP7 research project <http://fp7-aspire.eu>
- [4] QualiPSo European Project <http://www.qualipso.org>
- [5] The Cathedral and the Bazaar, Eric Raymond, 1997, <http://catb.org/~esr/writings/cathedral-bazaar/>
- [6] Communities of practice: learning, meaning, an identity. E. Wenger, Cambridge University Press 1998. ISBN: 0-521-66363-6
- [7] PLUME/RELIER Project (Promoting economical, Useful and Maintained softwarE for the Further Education And THE Research communities) - Jean-Luc Archimbaud – <http://www.projet-plume.org/en>
- [8] Audiofanzione, French Musician Forum Community, <http://www.audiofanzine.fr>
- [9] JRES 2010, Journées réseaux, <https://2010.jres.org/>
- [10] WorldWide Linux User Groups, <http://www.linux.org/groups/>
- [11] Open source software communities: current issues by Françoise Détienne, Jean-Marie Burkhardt, Flore Barcellini, PPIG Newsletter - September 2006
- [12] A life-cycle perspective on online community success by Iriberry, A. and Leroy, G. 2009. ACM Comput. Surv. 41, 2, Article 11 (February 2009)
- [13] Cultivating Communities of Practice by E. Wenger, R. McDermott & W. M. Snyder, 1998. ISBN-1578513308
- [14] Building Communities of Practice: A summary guide, 2003, by Pioneers of Change ([www.pioneersofchange.net](http://www.pioneersofchange.net))
- [15] Best Practices: Starting and Driving Communities of Practice by Daniel W. Rasmus, 2003, Planning Assumption, <http://www.gigaweb.com>
- [16] FOSS Governance Fundamentals, 2008, <http://www.fossbazaar.org>
- [17] Nosferos, web platform for social and solidarity economy networks, <http://www.noosfero.org/>
- [18] ELGG, open source social networking engine, <http://www.elgg.org/>
- [19] Open Atrium, Intranet and team portal package based on Drupal, <http://openatrium.com>
- [20] Buddy Press, a social network engine based on WordPress, <http://buddypress.org/>
- [21] Producing Open Source Software, Karl Fogel, <http://producingoss.com/>
- [22] Open source software development methodology by OSS Watch, 2007-2011, <http://www.oss-watch.ac.uk/resources/developers.xml>
- [23] Project Management in Free Software Projects by Martin Michlmayr, 2010, <http://fossa2010.inrialpes.fr/day-1> and <http://www.cyrius.com/publications>
- [24] Exploiting Social Software to Build Open Source Communities by Imed Hammouda, Timo Aaltonen and Petri Sirkkala, Department of Software Systems, 2009, <http://www.cs.tut.fi/sosea08/papers/paper8.pdf>
- [25] Modèles économiques des logiciels open source et logiciels libres, Fabernovel Consulting, 2007, [http://www.fabernovel.com/businessmodels\\_opensource.pdf](http://www.fabernovel.com/businessmodels_opensource.pdf)
- [26] A Proposal for Recognizing Reputation within Communities of Practice, C.Cruz, M.Gouvêa, C.Motta, F. Santoro, UFRJ, UNIRIO, Brazil, 2008. ISBN 978-3-540-92718-1
- [27] Reputation, trust and the dynamics of leadership in communities of practice, P. Muller, 2006
- [28] OW2 Europe Local Chapter Webinar Platform - <http://www.ow2.org/view/Activities/EuropeLocalChapterWebinars>
- [29] Impact Interaction , B2B Communities – What works, 2009, [https://www.socialtext.net/ocu2009/index.cgi?b2b\\_communities\\_what\\_works\\_best\\_practices\\_from\\_the\\_past\\_10\\_years\\_of\\_b2b\\_community\\_management](https://www.socialtext.net/ocu2009/index.cgi?b2b_communities_what_works_best_practices_from_the_past_10_years_of_b2b_community_management)
- [30] Framework for Governance in Open Source Communities by C. Latemann, S. Stieglitz, Postdam University. 2005, ISBN 0-7695-2268-8
- [31] The community Round Table, <http://community-roundtable.com/2009/06/the-community-maturity-model/>
- [32] ROI et présence sur les médias sociaux, outils d'évaluation - <http://vansnick.isexl.com/2010/01/17/roi-et-presence-sur-les-medias-sociaux-outils-devaluation/>, translated from MITSLOAN Management Review “Can you Measure the ROI of Your Social Media Marketing”, <http://www.slideshare.net/MichaelGaspar/mit-5769540>
- [33] OW2 <http://www.ow2.org/>
- [34] ICOM, pôle des industries du commerce, Lille-France, <http://www.picom.fr/>
- [35] fOSSa, the Free Open Source Software Academia Conference, INRIA, 2009-2010, <http://fossa.inrialpes.fr/>



## Biographies

**Stéphane Ribas** (M.Sc, University of Surrey 1996) has spent 12 years in software industries & services. He has spent many years in European countries & has been involved in several important projects as a support & technical consultant for large customers. Stephane joined INRIA in 2008 to co-lead OW2 Europe Local Chapter & contribute(d) to several Open Source and European projects (Xwiki Concerto, AspireRFID, QualiPSO, Humavips, HOMES). He has developed very strong skills in building & fostering online communities. Today, Stéphane is in charge of animating the technologies and development department within INRIA institute.