UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by Novática <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by Novática

UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Cover page designed by Concha Arias-Pérez
"Falling Upwards" / © CEPIS 2009
Layout Design: François Louis Nicolet
Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <novatica@ati.es>

UPGRADE Newslist available at
<http://www.upgrade-cepis.org/pages/editinfo.html#newslist>

The opinions expressed by the authors are their exclusive responsibility

ISSN 1684-5285

Monograph of next issue (August 2009)

"20 years of CEPIS: Informatics in Europe today and tomorrow"

(The full schedule of **UP**GRADE is available at our website)

CEPIS UPGRADE

The European Journal for the Informatics Professional
http://www.upgrade-cepis.org

Vol. X, issue No. 3, June 2009

Monograph: Libre Software for Enterprises

(published jointly with Novática*)

Guest Editors: *Jesús-M. González-Barahona, Teófilo Romero-Otero, and Björn Lundell*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Community Management in Open Source Projects

Martin Michlmayr

The increasing complexity of communities established around free and open source software (FOSS) projects as well as the increasing commercial interest in such projects has led to more and more communities adopting a dedicated community manager position. The community manager acts as the liaison between company and community and ensures that there is a good relationship between the two. They also facilitate the community, help it grow and ensure that the community stays healthy. This paper will discuss community management in FOSS projects and look at the role of the community manager position.

Keywords: Collaboration, Community, Management, Open Source, Volunteers.

1 Introduction

Online communities are increasingly getting attention for a number of reasons. One of them is the enormous wealth of knowledge that users involved in these communities have created on a collaborative basis. The Internet has played a significant role in the creation of this knowledge by enabling people with similar interests to get together and share their passions and interests. According to Lessig [1], the collaborative opportunities offered by the Internet have led to a transformation from a predominantly read-only culture to a culture that is increasingly read-write. In other words, fewer people are passive consumers and more and more people actively share their ideas and create new knowledge or products. Instead of merely watching television, fans go online and discuss the show with their peers; in addition to reading books, fans write their own fan fiction and music fans remix music and share it with others.

Even on the Internet, the majority of users are passive consumers but there is a trend towards being engaged more actively. This is partly because sites are increasingly being designed with the goal of attracting input and active participation from users and customers. O'Reilly [2] uses the term *architecture of participation* to describe "*the nature of systems that are designed for user contribution*".

There are many examples of online sites that have used its architecture of participation to attract a significant user base of active contributors. Wikipedia is the famous encyclopaedia that "*anyone can edit*", Flickr is a photo sharing site with countless high quality photos (many under a Creative Commons license), Amazon has a wealth of user-contributed book reviews and who has not heard of YouTube, which allows you to "*broadcast yourself*". What makes these sites special is not the underlying technology that could easily be replicated by someone else but the community of users they have attracted who contribute to the site and who make the site what it is.

Acknowledging the importance of online communities, Seth Godin, a well-known marketing expert, identified the role of the online community organizer (or community manager) as one of the most important jobs of the future [3]. Community managers help to build a new community as well as to facilitate and grow an existing community. This paper will discuss community management and the role of the community manager in the light of FOSS projects, which rely on communities of developers and other contributors to create software. The next section will discuss the role of community management and different views of the community manager position. This is followed by a section on the roles of community managers and the skills required for their role. Finally, various activities will be covered that can be used to build a community.

Author

Martin Michlmayr has been involved in various free software projects for over 10 years. He used to be the Volunteer Coordinator for the GNUstep Project, acted as Publicity Director for Linux International and served as the leader of the Debian project. In the two years as the leader of Debian, Martin represented the project and performed important organizational and coordination tasks within the project. Martin joined Hewlett-Packard in 2007 where he works as an Open Source Community Expert and acts as the community manager of FOSSBazaar. Martin currently serves on the board of the Open Source Initiative (OSI). Martin holds Master degrees in Philosophy, Psychology and Software Engineering, and earned a PhD from the University of Cambridge. <tbm@hp.com>.

2 A Definition of Community Management

FOSS projects have a long history of community participation and development. The FOSS paradigm was started by individuals who shared the common goal of producing software that would give users more liberties than proprietary software [4]. Even though open source software can be developed by a company without any involvement from outside contributors, the majority of software is either developed by a community of volunteers (who are either unpaid or paid by a company that has an interest in the software) or in a collaborative fashion by a company that is leading the project together with a community of users who are interested in improving the software. Community is a defining characteristic of most FOSS projects along with license and development methodology. The academic literature also associates the collaborative methods in which FOSS projects are performed with a number of positive aspects, such as potentially high levels of peer review [5] and user innovation [6].

Even though community has always played an important role in FOSS projects, community management by dedicated people with a community manager or similar position is a fairly recent development. This is partly because the success of FOSS has led to an enormous growth of the user and developer base of projects and this evolution of the community can be associated with certain growing pains that need to be managed [7]. Furthermore, hybrid economic models which combine sharing and commercial economies are increasingly employed in the context of FOSS development and the relationship between the two has to be managed actively. Lessig [1] defines a hybrid as "*either a commercial entity that aims to leverage value from a sharing economy, or it is a sharing economy that builds a commercial entity to better support its sharing aims*". Mozilla, a FOSS project that develops the popular Firefox browser, is an example of the latter because its supporting organization has obtained substantial funding through partnerships. There are countless examples of companies whose business model relates to Linux and other free and open source software and that have a key interest in ensuring that the community around this software will strive and remain healthy.

There are a number of reasons why companies should establish large communities around the FOSS projects they are building. In addition to getting contributors who help develop and evolve the software, having a community gives people a place to engage with the company [8]. This in turn can lead to brand loyalty, product innovation and users who may become advocates for the company and products and may create viral marketing. Red Hat (with Fedora) and Novell (with OpenSUSE) are just two prominent examples from the Linux industry of companies that have established community-driven projects from which the company benefits in a number of different ways.

The benefits of communities and the need to facilitate and manage them have given rise to the community manager position. While traditionally the project leader ensured that there was a strong community around the project, the responsibilities are nowadays often divided among different people. The project leader is typically seen as a technical position and they have the responsibility to drive the software forward and work with developers. The community manager, on the other hand, ensures that there is a healthy community around the project, interacts with users, developers and other stakeholders, and facilitates organizational aspects of the project. When a company is the driving force behind a project, the community manager acts as the liaison between company and community and ensures that there is a good relationship between the two.

Many successful FOSS projects that are driven by corporations or that have high levels of corporate involvement have a dedicated community manager today. While many of them use the "community manager" title, there are actually substantial differences with how this role is viewed. Adam Williamson (2009), who used to work for Mandriva and is now employed by Red Hat, made the following observation: "*It's very interesting to see how differently the role is conceived; I was sort of a user community mouthpiece at Mandriva, Jono [Jono Bacon, community manager of Ubuntu] considers "community" to mean mostly the developer/contributor community, and Zonker [Joe Brockmeier, community manager of OpenSUSE] says his role works out to be mostly evangelism*".

This observation shows that the responsibilities of the community manager are very broad. They have to work with users, developers, contributors and other stakeholders, often act in an advocacy role and carry out many other tasks (the most important of which will be discussed in the next section). One of the key functions of the community manager is to ensure that everyone else can contribute effectively. Greg DeKoenigsberg [10], a Community Architect at Red Hat, remarked in an interview that "*it's a position that focuses on making the life of the individual contributor as productive, and as painless, as possible*". DeKoenigsberg elaborates on the community manager role: "*You want to grow that contributor base, by whatever means. You want your core developers to be community-oriented, of course, but you also want them to spend the vast majority of their time doing what they do best, which (one hopes) is coding. Which means that you want someone else worrying about keeping your community of contributors happy and growing*".

Ensuring that contributors are happy and get something out of the project is important because otherwise they might stop contributing to the project and might focus their energies on other activities. However, this leads to the question as to which factors and values are important to volunteers in collaborative projects. Mark Shuttleworth, the founder of the Ubuntu project, suggests that volunteers expect the following three conditions [1]:

- *Respect:* the community and its contributors must be respected and honoured properly.
- *Responsibility:* contributors from the community must have sufficient authority in the project. This is particularly important in projects that are led by companies because they have to balance leading the project with giving external contributors the authority to also influence the direction of the project.
- *Meaning*: contributors want to have the feeling that the project to which they contribute has meaning. They want to "*feel they're being part of something that's big and important and beautiful*". If a project doesn't give volunteers the feeling that their contributions as well as the project itself are meaningful, they will spend their time on a project that does.

These values are important because they are closely related to the motivation of contributors and the only way to influence the work of volunteers is through motivation. Jim Grisanzio, the former community manager of the OpenSolaris project, shared some comments about the distinction between power and leadership on his blog [11] that are important in this context: "*You can't substitute power for leadership. Leaders of volunteers elicit cooperation by tapping into the shared values of the community, and that's a much more challenging exercising than dictating orders with threats of force to back you up*".

In other words, community managers have no say over the activities of non-paid contributors and have to find better ways than power to motivate contributors to work on certain activities. They have to lead the community but at the same listen to the needs of the community and support them.

In summary, the community manager position is a very challenging one that comprises many different tasks and responsibilities. Community managers work with a number of stakeholders and perform many different activities. They have no power over contributors and have to find interesting incentives, such as contributing to a meaningful project, in order to motivate contributors.

3 Roles and Skills of Community Management

As can be imagined from the description of the community manager position above, a community manager has many roles and has to perform a number of different activities. They include:

- Facilitation: one of the main responsibilities of the community manager is to facilitate the community in many different ways. For example, the community manager has to help grow the community by encouraging new members to join the project and to help these contributors to get started. Furthermore, the community manager will usually keep the "big picture" of the project in mind and will connect people. If someone has or is working on the solution to a problem someone else has but they don't know of each other, the community manager can connect the two so they can collaborate. Knowing what's going on within the whole project is also important because it allows the community manager to suggest neglected activities to members who are new to the project. Communication in general is very important in small and large projects and it's an important task of the community manager to send updates and keep people informed with what's going on within the project. Finally, it is important to ensure healthy interactions within a community. Many projects, especially when they reach a certain size, will attract so called trolls who bring negative energy and take away time from contributors. Some projects have adopted a Code of Conduct that governs the communication and interactions within a project. It's the responsibility of the whole project to make sure the Code of Conduct is followed but the community manager should intervene if nobody else does.
- Delegation: as argued above, a key responsibility is to facilitate. It is not the responsibility of the community manager to perform all tasks in a project. In fact, the community manager should be cautious to be too active in some areas because otherwise it will stop potential volunteers from contributing. For example, if a project has a support forum, the community manager should not immediately respond to every question. Instead, they should ask other volunteers to jump in and help and point out questions to experienced developers who might be able to answer. The community manager should ensure that help is given on the forum and they can of course answer questions themselves as well here and there, especially if nobody else responds.
- Marketing: community managers often get to play an advocacy role. This may involve travelling to conferences to talk about the project and to get new contributors involved. It can also mean talking to the press about certain aspects of the project. Finally, community managers are also often asked to maintain a good relationship with other projects and companies.
- Listening and observation: one of the key advantages of having a large community is to gather input from a large number of people with diverse opinions. It's the responsibility of the community manager to listen to the needs and ideas of contributors and to make sure that those thoughts are passed on to other people within the community or to the company driving a project. In addition to listening to the community, the community manager has to ensure that the feedback will be taken into account and acted upon.
- Creating vision: as argued before, volunteers need a clear motivation to contribute to one project rather than another. The community manager has to help create a shared culture for the community as well as a vision for the project. It is important to take feedback from contributors into account when evolving the community in a given direction.
- Reporting: the community manager has to track and report the health and state of the community on a regular basis. This can involve a number of different metrics, such as bug statistics, figures on the growth of the membership and other factors that define the community. The community manager has to communicate the state of the community to the company but they also have to use the data in order to improve the community.

Based on these activities, a number of skills can be derived that are vital for community managers:

- Communication: the community manager is a communicator. They have to interact with users, developers, contributors and other stakeholders interested or involved in the project. They have to regularly give updates of what's going on in the project and to work with representatives from other projects, companies or the press. Good writing skills are therefore essential and public speaking skills can be very beneficial as well. Furthermore, since large projects are typically international, it's helpful when community managers know how to deal with other cultures and also how to make contributors from different cultures feel welcome in the project.
- Marketing: Getting the word out about the project is very important to attract new contributors and potentially also to attract other resources, such as funding. While the community manager shouldn't have to do marketing in the traditional sense, it's not unusual for them to be responsive for "community marketing".

- Management: organizing various activities is another key responsibility for the community manager and so they have to be good at organization. For example, they may have to organize events such as user group meetings, keep track of tasks that have to be performed in the project and who has volunteered for what.
- Technical skills: even though the community manager is not a technical position, it helps to know a few technical things. This is useful to better understand technical problems or challenges inside the project and to help new volunteers find an area they can contribute to. Having some technical skills can also be helpful when the community manager wants to set up a new forum or other infrastructure to support the community and everyone else is too busy with other things.
- Patience and helpfulness: the community manager has to listen to many contributors and interact with people who are new to the project and therefore unfamiliar with its practices and culture. Growing the community can also take a long time [12]. Because of this, patience and helpfulness are important virtues for a community manager.
- Being visible and present: as argued above, the community manager should not answer all questions on the forum or solve everyone's problems. However, they have to be present and visible so contributors know that they always have someone to go to if they have a problem. Foster [13] argues that being a community manager is not a 9-5 job, partly because the community manager has to respond immediately when there is a crisis and partly because of time zones in an international community.

4 Activities to Build Community

There are many ways to build and grow a community. A number of popular activities employed by FOSS projects are covered here to give an impression of the kind of activities that have been found to be successful.

- Face to face meetings: while the day to day work of FOSS projects are typically performed via the Internet, face to face meetings have been shown to be very useful for a number of reasons. One reason is that face to face meetings are very effective, especially when it comes to brainstorming new ideas and solving existing problems. Another reason is that real life meetings allow people who normally interact only via the Internet to get to know each other in person. This can make future collaboration much easier, motivates people and leads to a stronger sense of community.
- Booths at conferences: they can be used to recruit new volunteers and allow interaction between developers and users. Going to conferences and exhibitions can be very motivating for developers because they get to meet people who use their software and it's also rewarding for many users to meet the original developers of the software they employ. Meeting users is also a good way to obtain feedback that can be used to further improve the software.
- Online training events: a number of projects have recently started to organize special events for newcomers to the projects who are interested in learning more about the project and about ways to contribute. For example, the OpenSUSE Community Week consisted of many training sessions about various aspects of OpenSUSE. The training was conducted on Internet Relay Chat (IRC), a chat system that is often employed by FOSS projects as a key communication infrastructure.
- Bug squashing parties: another way to allow newcomers to help out and to get some volunteers to contribute on a regular basis is to hold online or real life events during which bugs in the software are analyzed and fixed. Such events are often coordinated by an experienced Quality Assurance (QA) person from the project and there are other experienced developers who help newer contributors get started and to fix their first bug. This can be a great motivation to newcomers who may then choose to increase their contribution to the project.
- Newsletter: posting regular updates of a project is a great way to keep the community at large informed with developments within the project. Newsletter can also be used to publish calls for help or promote activities that users may find interesting.

As mentioned above, these are just a number of examples of successful activities FOSS projects employ to build a community. To some extent, successful activities may depend on a particular project but in the majority of cases it makes sense to look at other projects to get good ideas of activities that can be implemented in a project.

5 Conclusions

FOSS projects have always relied on communities of contributors who improve the software. Because of the increasing size of communities as well as commercial interest in FOSS communities, the community manager role is getting of increasing importance. A community manager can act as the liaison between the company and a community, and they can ensure that there is a healthy relationship between the two. The community manager position comprises many different activities, ranging from facilitation to advocacy. This broad set of responsibilities is reflected in the skills a community manager has to bring to the table, including strong communication and organization skills as well as patience and helpfulness.

References

[1] Lawrence Lessig. "Remix". Bloomsbury Academic, London, 2008. ISBN: 978-1408113479.

[2] Tim O'Reilly. "The Architecture of Participation". 2004. <http://www.oreillynet.com/pub/a/oreilly/tim/articles/architecture_of_participation.html>.

[3] Seth Godin. "Jobs of the future, #1: Online Community Organizer". 2007. <http://sethgodin.typepad.com/seths_blog/2007/07/jobs-of-the-fut.html>.

[4] Richard Stallman. "The GNU Operating System and the Free Software Movement". In: C. DiBona, S. Ockman, M. Stone (Eds.), "Open Sources: Voices from the Open Source Revolution". pp. 53-70, 1999.

[5] Eric S. Raymond. "The Cathedral and the Bazaar". O'Reilly & Associates, 1999. ISBN-10: 1565927249.

[6] Karim R. Lakhani, Eric von Hippel. "How open source software works: free user-to-user assistance". Research Policy, 32(6), 923-943, 2003.

[7] Martin Michlmayr, Benjamin Mako Hill. "Quality and the Reliance on Individuals in Free Software Projects". Proceedings of the 3rd Workshop on Open Source Software Engineering, 105-109, 2003.

[8] Dawn Foster. "Why Companies Should Have Online Communities". 2008. <http://fastwonderblog.com/2008/04/16/why-companies-should-have-online-communities/>.

[9] Adam Williamson. "LinuxFest recap". 2009. <http://www.happyassassin.net/2009/04/27/linuxfest-recap/>.

[10] Greg DeKoenigsberg. "Interview with Greg DeKoenigsberg, Red Hat Community Architect". LinuxQuestions.org <http://www.linuxquestions.org/questions/interviews-28/interview-with-greg-dekoenigsberg-red-hat-community-architect-725426/>.

[11] Jim Grisanzio. "The Distinction Between Power and Leadership". 2009. <http://blogs.sun.com/jimgris/entry/the_distinction_between_power_and>.

[12] Matthias Stürmer. "Open Source Community Building". Master thesis, University of Bern, 2005.

[13] Dawn Foster. "What Does it Take to Manage a Community?". 2007. <http://fastwonderblog.com/2007/09/03/what-does-it-take-to-manage-a-community/>.