

Collaborative Development within Open Source Communities

Javier Soriano

Universidad Politécnica de Madrid, Spain

Sonia Frutos

Universidad Politécnica de Madrid, Spain

Miguel Jiménez

Universidad Politécnica de Madrid, Spain

INTRODUCTION

Open source communities are one of the most successful--and least appreciated--examples of high-performance collaboration and community building on the Internet today. Open source communities began as loosely organized, ad-hoc communities of contributors from all over the world who shared an interest in meeting a common need. However, the organization of these communities has proven to be very flexible and capable of carrying out all kind of developments, ranging from minor projects to huge programs such as Apache (Höhn, & Herr, 2004; Mockus, Fielding, & Herbsleb, 2005).

Other collaboration-intensive communities could benefit enormously by learning what open source communities are and how they work. In fact, their motivation and objectives are not confined to software development projects. They are increasingly taking shape around non-software-related collaborative activities (Shah, 2005). Moreover, open source has come to stand for much more than software whose source code can be freely modified and redistributed subject to just a few restrictions imposed by the terms of its distribution license. Information, documentation, and other “sources” generally related to innovation, and knowledge building and sharing processes, tend to come under the open source umbrella.

A full comprehension of open source communities requires an in-depth understanding of the underlying organizational process (e.g., the software development process for software development projects). Some of the patterns underlying these organizational processes are not confined to software development and are common to other successful communities as well. One of the main goals of current research into open source

communities is to identify these patterns (Kim, 2003) and develop a pattern language that can be used to describe, build, and improve other types of successful communities.

Finally, there is a trend toward two traditionally different development styles derived from opposing assumptions about the nature of development tasks—the model of most of the commercial world vs. the model of the open source world—converging. People would be astute to try to import some of the open source community model’s virtues into a commercial context and will find it worthwhile taking a look at the conditions necessary for creative work.

Bearing these premises in mind, this article begins by defining and characterizing the term “open source community.” It then tackles the issue of how these communities work (i.e., what the *patterns of collaboration* within successful open source communities are) and describes how these patterns could be applied in other types of communities apart from software related communities, and vice versa. This is intended to further the understanding of the open source model and its implications outside the realm of software development. In examining these questions, the article discusses existing, relevant research, and presents original case studies of working open source communities. These case studies hit at how collaboration works within successful projects.

OPEN SOURCE COMMUNITIES FUNDAMENTALS

We can define an *open source community* as a loosely organized, ad-hoc community of contributors from all over the world. These contributors share an interest in

meeting a common need, ranging from minor projects to huge developments, which they do through a high-performance collaborative development environment, allowing the organizational scheme and processes to emerge over time. The term derives from the notion of *community* (i.e., an amalgamation of people with related interests), where intent, belief, resources, preferences, needs, goals, and a multitude of other conditions may be present and common, affecting the degree of adhesion within the group. Communities may meet to share information, to participate in shared projects, or to complete group tasks. What most characterizes a community is the pursuit of a common productive goal and sharing interaction in many ways.

Essentially born out of a desire for increased general access to source and binary code, open source communities have been bound to computer networks, and have evolved at the same pace as the Internet. Inexpensive access to Internet resources and source or binary code has allowed programmers to collaborate irrespective of where they are, and is one of the major factors in the growth of the number and size of communities.

The open source collaborative development carried out by open source communities has led to the use and spread of more and more sophisticated collaborative development environments (CDEs), virtual spaces where all the stakeholders of a software project, possibly distributed in time and space, can negotiate, brainstorm, discuss, share knowledge and resources, and generally labor together to carry out some task in the context of a software development process (Booch & Brown, 2003). CDEs serve as the meeting point not only for the developers of the community, but also for the users, who play an important role in open source software development.

The philosophy behind open source communities is founded on peer-to-peer collaboration and delegating tasks for other developers to provide input at will. They are based on meritocracy, where it is the more active or capable contributors who act as coordinators or leaders, since there are no project managers as in a software development company. Therefore, the contributors to a community are motivated not only by the desire to produce a certain functionality or the idea that the source code and applications should be open, but also by the recognition of their achievements by the community or the intellectual stimulus.

The collaborators in an open source community are referred to as hackers, namely people with strong

computer skills who try to reach a goal by enhancing existing code or resources. This idea of enhancing source code or programs is closely knit with the beginning of the open source movement, because open source communities grew up around an existing program or solution, either by enhancing free software or by an open software release of an application (Raymond, 1999). In any case, the community starts working with an open release that has to be attractive and promising enough to encourage new developers to join the community.

Finally, the influence of the user's opinions in the software development process is a key feature in open source communities, in which the barriers between developers and users are generally quite low (Kim, 2003) and the communication channels between both groups are easily accessed. Moreover, users can easily become developers due to the inherent organization to open source communities, which enhances the community knowledge about the users' needs.

PATTERNS OF COLLABORATION FOR SUCCESSFUL OPEN SOURCE COMMUNITIES

Some of the patterns of collaboration underlying the organizational processes performed by open source communities are not specific to software development. These patterns represent knowledge blocks for building and improving successful communities as suggested by Alexander (1979). The following are prominent examples of patterns that have been identified in the most successful communities such as those analyzed in Kim (2003):

- **Evolve the community:** Bearing in mind the ideas explained in Raymond (1999), we conclude that designing an organizational structure for what might be rather than what is is likely to hinder not boost the project. Most successful communities allow an organizational scheme and processes to emerge over time rather than attempting to impose any structure. This will help minimize the risk of creating unnecessary organizational overheads with no immediate benefits. Community processes are lightweight, and tend to emerge in response to changing conditions.
- **Co-evolution:** Co-evolution is a term coined by Engelbart (1992) to describe how tools and their

3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/collaborative-development-within-open-source/17617

Related Content

Revisiting Musings on Co-Designing Identity-Aware Realities in Virtual Learning: The Shared Experiences

Francisca Yonekura (2023). *Research Anthology on Virtual Environments and Building the Metaverse* (pp. 386-395).

www.irma-international.org/chapter/revisiting-musings-on-co-designing-identity-aware-realities-in-virtual-learning/316104

An Interactive Space as a Creature: Mechanisms of Agency Attribution and Autotelic Experience

Ulysses Bernardet, Jaume Subirats Aleixandriand Paul F.M.J. Verschure (2017). *International Journal of Virtual and Augmented Reality* (pp. 1-15).

www.irma-international.org/article/an-interactive-space-as-a-creature/169931

A Review of Augmented Reality in K-12 Education Environments

Adam C. Carreon, Sean J. Smithand Kavita Rao (2020). *International Journal of Virtual and Augmented Reality* (pp. 32-61).

www.irma-international.org/article/a-review-of-augmented-reality-in-k-12-education-environments/283064

Advanced Visual SLAM and Image Segmentation Techniques for Augmented Reality

Yirui Jiang, Trung Hieu Tranand Leon Williams (2022). *International Journal of Virtual and Augmented Reality* (pp. 1-28).

www.irma-international.org/article/advanced-visual-slam-and-image-segmentation-techniques-for-augmented-reality/307063

Can You Feel It?: Effectiveness of Anxiety Cues for the Design of Virtual Reality Exposure Therapy

Jessica Morton, Jolien De Letter, Anissa All, Tine Daeseleire, Barbara Depreeuw, Kim Haesen, Lieven De Marezand Klaas Bombeke (2021). *International Journal of Virtual and Augmented Reality* (pp. 1-17).

www.irma-international.org/article/can-you-feel-it/298983