Going Open: Does it Mean Giving Away Control?

Nadia Noori and Michael Weiss

> *"Many people think that open source projects are sort of chaotic and anarchistic. They think that developers randomly throw code at the code base and see what sticks."*
>
> Mitchell Baker
> Chairman, Mozilla Foundation

Open source software has evolved from being an effort driven by a collective of volunteers to become an integral part of commercial software. Constant demands for new features besides maintaining product quality made companies seek open source as an answer for these demands. These growing demands brought with them control of quality, architecture, contribution management, and community management.

This article explores the governance strategies adopted by open source software projects to manage the quality of complements (such as plug-ins that extend a platform's functionality) developed by community members outside the core team. The outcomes of the research contribute to our understanding of the strategies followed by different open source platform owners (the open source project initiators) to manage external innovation in the case of platform extensions in two areas: i) governance models and ii) regulatory tools.

Introduction

Conventional research on open innovation and collaboration examines how organizations open up the innovation process and how they control the collaboration with others. The research also identifies the need for governance models and architecture of participation in collectives that embrace open innovation practices to help maintain momentum and ensure continuity. Existing research on systemic innovation and platform ecosystems helps us understand the structure of platform-complement product systems (Boudreau and Hagiu, 2009: tinyurl.com/a3xc7xj; Baldwin and Woodard, 2009: tinyurl.com/aceg9ac; Dahlander and Gann, 2010: tinyurl.com/chacrs9). Studies examine the types of relationships and information transactions among the members of a platform ecosystem, the importance of platform owners as regulators of the ecosystem, and regulatory instruments available to them.

Iyer (2006; tinyurl.com/csmescv) describes how software companies are operating in a small world of interconnected networks and how innovation is increasingly taking place in such networks. The question is no longer whether or not to open the innovation process and collaborate, but how to best leverage a network of external parties (Tuomi, 2002: tinyurl.com/crbmhrv; Moore, 2006: tinyurl.com/5rtbj6u; Pisano and Verganti, 2008: tinyurl.com/67bcd3b; Vujovic and Ulhoi, 2008: tinyurl.com/b6l3jov).

In most open source projects, there is a focal organization that acts as that platform owner (or keystone in a project ecosystem) that provides the platform and facilitates contributions by other members in the community (Iansiti and Levien, 2004: tinyurl.com/7t4xgvn; Noori and Weiss, 2009; tinyurl.com/ae2n8su). A platform can be a product, service, or technology that provides a foundation for other parties to develop complementary products. The platform can be owned by a single player, as in the case of Apple's control over the iPod and iPhone application ecosystems, or it can be developed and influenced by a group of players, as in the case of the Eclipse software development platform (des Rivieres and Weigand, 2004: tinyurl.com/arwl3j3; Shuen, 2008: tinyurl.com/anvzq6y; Hagiu and Yoffie, 2009: tinyurl.com/bu7zn3n).

The literature on how platform owners manage complementary markets is focused on complements that build on the platform but not on complements that integrate

Going Open: Does it Mean Giving Away Control?

Nadia Noori and Michael Weiss

with the platform, such as extensions. Platform extensions extend the functionality of a platform beyond its core capabilities and can integrate at different levels with the platform and other existing extensions, and this integration reflects upon the integrity of the platform. So, the quality of platform extensions contributed by community members has become a concern for platform owners because it can comprise the overall quality of the platform and other complements as well (Messerschmitt and Szyperski, 2003: tinyurl.com/d2ulug3; Gawer and Cusumano, 2008: tinyurl.com/bjkhq3j; Bosch, 2009: tinyurl.com/arrlwcz).

We studied six open source projects (Eclipse, Firefox, Apache HTTP, Spring, OpenOffice, and MySQL) to examine the strategies followed by the platform owners to manage contribution by external parties and to manage the quality of the complements developed by those parties. We limited the research to one type of complement, platform extensions, because of the direct effect of the quality of extensions on platform integrity as well as to keep the study to a manageable size.

Research Method

Research on regulating the platform extension development process is still in its early stages, therefore there is little known about strategies adopted by platform owners to control the quality of extensions developed by external parties (Hagiu, 2009; tinyurl.com/adn8gb2). We used the case study research approach due to the novelty of the research area (Eisenhardt, 1989; tinyurl.com/7dfuc3z).

In our research, we examined 12 open source platforms and related commercialized software platforms (if any) from the time period between 2000 and 2009. The unit of analysis was a software platform that provides the ability for external parties to extend its functionalities. An example of an extension we examined is the tabbed browser extension for Firefox, which affects the core functionality of the platform.

Eisenhardt (1989; tinyurl.com/7dfuc3z) recommends theoretical sampling when selecting cases for case study research, which implies that cases may be chosen to replicate previous cases or extend emergent theory. Unfortunately, previous research did not provide a reference for selecting cases using theoretical sampling; therefore, data was collected in two initial waves of four cases each and, once preliminary results emerged, a third wave of four more cases was chosen using theoretical sampling.

Governance Models

Within the sample cases, we found three types of governance models: tight-control, loose-control, and hybrid-control. Each governance model consists of the following attributes: community structure, extension types, and governance structure and network openness. Each model was associated with non-trivial trade-offs in terms of governance, openness, quality, and flow of ideas.

Furthermore, different levels of governance and openness may be applied to different types of extensions. Internal extensions, which are more widely used and usually deployed together with the platform core, are often more tightly controlled than external extensions, which are developed to meet more specialized needs. In the case of internal extensions, there is a significant impact of low quality of those extensions on the platform and on each other; a reduced flow of new ideas is traded off against higher quality. As for the external extensions, it is more important to allow new ideas to develop than to monitor their quality. Yet, the distinction between internal and external extensions is not fixed; over its life, an extension may change its type.

The governance structure of a network can be either hierarchical, flat, or a hybrid between these extremes. In a hierarchical governance structure, the platform owner both defines problems and selects which solutions are adopted, whereas in a flat structure a community decides on both problems and solutions. The case between those two extremes is a hybrid of hierarchical and flat structures: although the community decides on problems, the platform owner selects solutions.

Openness of a platform network refers to the degree to which participation in the network is open. In an open network, any party (partners, customers, or even competitors) can contribute to the platform. Open source projects are examples of this type of network. In a closed network, the platform owner selects who can participate based on the capabilities and resources required for the innovation (Pisano and Verganti, 2008; tinyurl.com/67bcd3b).

Table 1 summarizes the governance models, their attributes, and their associated effects on the quality of extensions and flow of ideas. Table 2 summarizes the advantages and disadvantages of each of the three governance models with an emphasis on the quality of extensions and flow of ideas.

Going Open: Does it Mean Giving Away Control?

Nadia Noori and Michael Weiss

Table 1. Summary of governance models and associated extensions types, governance structure, and network openness

Governance Model Type	Community Structure	Extension Types	Governance Structure	Network Openness	Extension Quality	Flow of Ideas
Tight-control model	Core and internal	Internal only	Hierarchical	Closed/ open	Medium to high	Low to medium
Loose-control model	Core and external	Internal	Hierarchical	Closed	Medium to high	Medium
		External	Flat	Open		
Hybrid-control model	Core, internal, and external	Internal	Hierarchical / flat	Open	High	Medium to high
		External	Flat	Open		
		Commercial (internal or external)	Hierarchical	Close		

Table 2. Advantages and disadvantages of the three types of governance models

Governance Model Type	Advantages	Disadvantages
Tight-control model	• Easy to control the development process • Easy to control the quality of the product • Direct efforts towards achieving the platform owner's desire to meet certain requirements	• Limited number of ideas
Loose-control model	• Unlimited number of ideas • Unpredictability could lead to unusual use of the platform functionalities that can trigger innovation in the community	• Unpredictable community of developers • Difficult to control the community of developers • There is no way to guarantee the quality of the product
Hybrid-control model	• Platform owners have control over who develops what in the community • Platform owners can control the quality of developed product on different levels • Platform owners are the orchestrators for the flow of ideas in the community	• Requires effort from the organization to oversee the different parties involved

Going Open: Does it Mean Giving Away Control?

Nadia Noori and Michael Weiss

Regulatory Instruments

To support the governance models and enforce control over the development process of the platform extensions, the platform owners needed to use a number of regulatory instruments in conjunction with the governance model. In our reseach, we found that regulatory instruments enabled platform owners to manage information transactions in the platform network, support the process of developing extensions, and control the quality of the product by providing the community with tools to develop, test, and integrate extensions, and to share the experience among the platform members.

The regulatory instruments help platform owners to establish barriers of entry to the developer community such as pricing or the development process. Instruments such as platform architecture and toolkits help to create technical boundaries between the platform and the developer community.

Regulatory tools such as pricing and membership refer to a pricing schema set by the platform owner to charge third-party developers for gaining access rights to the platform, information, and service. Pricing and membership programs enable platform owners to filter the inflow of ideas as well as the quality in the network. Although using the pricing instrument enables the platform owner to control the flow of ideas, there is no guarantee of eliminating poor-quality contributions because it depends on the pricing structure and the platform owner's need to access external resource (Hagiu, 2008; tinyurl.com/bymusad).

Development toolkits are another form of regulatory tool. Toolkits are a combination of software infrastructure and development frameworks that reduce the time and effort required to develop, provision, and operate extensions; they also contribute to quality. For example, to ensure the quality of extensions, the Mozilla project offers a toolkit consisting of several tools that include a testing framework to test the performance and quality of Firefox extensions. Also, the Eclipse platform provides a plug-in development environment, a comprehensive series of build and user-interface creation tools, and tools for API maintenance.

Sandboxes are another type of regulatory tool in which extension developers are allowed to test their extensions in the actual deployment environment. For example, Firefox provides a sandbox review process on its Firefox add-on site, where extensions are available for trial and testing by the community. The sandbox review process enables the developers to test their extension before moving it to the general-availability phase.

Introducing a development process is also another regulatory tool used by platform owners to control the quality of developed extensions and filter the inflow of ideas into the platform ecosystem. An example of the development process used as a regulatory instrument is the incubation process, which is another method used by platform owners to control the quality of extensions developed by external parties. The incubation process enables the platform owners to filter the flow of ideas in the internal-extensions community of contributors (Duenas et al., 2007; tinyurl.com/aytv5x7). For some platforms, such as Mozilla, the incubator is a working directory that is considered a testing ground for experimenting with new ideas and it is a workspace where lead developers or module owners work with inexperienced developers.

Practical Implications

The results of our research are relevant to managers of both open and closed source platforms, third-party developers creating platform extensions, and researchers in innovation management. The research provides collaboration models that help platform owners understand the strategies adopted by other platform owners to manage the quality of platform extensions.

The models are a combination of collaboration governance structures and regulatory tools that helped platform owners to leverage the innovation process in their ecosystems and provide guidelines for developing third-party complements. The research also opens opportunities for future research on creating models for how platform owners can maximize user innovation in platforms, and how they can manage the platform-extension development process.

Conclusion

Open source is a living example of the viability and sustainability of the open-innovation model. The process of going open and maintaining growth and success of the open source platform is not chaotic or a set of random actions. Throughout the years, the open source community has learned how to organize itself and provide collaboration models and tools that fit within the free/libre open source software context. These communities needed such control mechanisms in place to ensure quality and maintain growth. Open source platforms had evolved from voluntarily initiatives to sustainable entities alongside commercial equivalents inside the software industry.

Going Open: Does it Mean Giving Away Control?

Nadia Noori and Michael Weiss

About the Authors

Nadia Noori holds an MASc degree in Technology Innovation Management from Carleton University in Ottawa, Canada. Her research interests includes open source platforms, governance models and collaboration frameworks, and product architecture and design. She works currently as a Production and Platforms Manager at Coral CEA, a not-for-profit organization based in Ottawa.

Michael Weiss holds a faculty appointment in the Department of Systems and Computer Engineering at Carleton University, and is a member of the Technology Innovation Management program. His research interests include open source ecosystems, mashups/Web 2.0, business process modeling, social network analysis, and product architecture and design. Michael has published on the evolution of open source communities, licensing of open services, and the innovation in the mashup ecosystem.

Citation: Noori, N. and M. Weiss. 2013. Going Open: Does it Mean Giving Away Control? *Technology Innovation Management Review.* January 2013: 27-31.