Chapter III

Ecology and Dynamics of Open Source Communities

Michael Weiss, Carleton University, Canada

Gabriella Moroiu, Carleton University, Canada

Abstract

The goal of this chapter is to document the evolution of a portfolio of open source communities. These are communities formed around a set of related projects with common governance, which often produce artifacts shared among all projects. It helps to think of a portfolio of project communities as an ecology, in which the projects are mutually dependent, and there is both cross-project collaboration and competition for resources among the communities. As a case study, we explore the ecology of communities within the Apache project, one of the largest and most visible open source projects. We infer the community structure from developer mailing lists, and study how the communities evolve and interact over time. The analysis lends support to the often-stated hypothesis that open source communities grow by a process of preferential attachment. However, we show that the influx of external developers is not the only factor affecting community growth. The structure and dynamics of a community is also impacted by inter-project information flow, and the migration of developers between projects (including the formation of spin-offs).
Introduction

The open source model has received increasing attention as an alternative to closed source development. It is characterized by the transparency of development process and artifacts produced, as well as the decentralized organizational structure through which a community of developers and users coordinate their activities. The progress of an open source project is continuously tracked in a number of archives including code repositories, mailing lists, wikis, and bug tracking lists. Community members (both developers and users) can belong to any number of organizations, and their decision making process is governed by the principle of meritocracy, whereby members are given varying levels of access to the project based on their history of contribution.

Despite the evident successes of the open source model, however, we do not fully understand how open source communities organize themselves. We would like to gain a better understanding of the mechanisms underlying the growth of communities. From these, we could then derive guidelines for the design of successful open source communities. In this chapter we look at the evolution of community structure. As we want to understand the open source development model, we focus on developer communities. One goal of the chapter is to describe community structure in terms of observable metrics such as size or degree distribution. However, this does not provide us with insight on the inner workings of community organization. Therefore, we also explore potential growth factors and mechanisms that help explain the observed evolution of community structure.

As a case study, we explore the ecology of communities within the Apache project, both because it is a highly visible group of open source communities, but also because a wealth of data is being collected on the Apache project site that allows deep insight into the dynamic project structure. What sets this study apart from, for instance, a study of SourceForge projects (which largely vary in scope, size, and maturity) is that the Apache projects have a common domain (web services), are well established in the open source community, and are relatively large. For example, the observation made about SourceForge projects that most projects have only one developer and are not interconnected is not true about Apache projects.

We have analyzed eight years (between 1997 and 2004) of project developer mailing lists to extract the structure of the associated communities. The selection of the Apache project, and the vehicle of mailing lists for recovering social interactions was also motivated by the availability of the Agora tool for visualizing community participation in the Apache project from project mailing lists, which was developed by an Apache core developer (Mazzocchi, 2006). However, as noted above, there are other ways of tracking community participation, and, ultimately, one would like to construct a model of community evolution by combining multiple views on community activity. Questions such as how contributions to code repositories and