Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2003 Proceedings

International Conference on Information Systems (ICIS)

December 2003

Survival of Open-Source Projects: A Population Ecology Perspective

Shobha Chengalur-Smith
University at Albany, SUNY

Anna Sidorova
University at Albany, SUNY

SURVIVAL OF OPEN-SOURCE PROJECTS: A POPULATION ECOLOGY PERSPECTIVE

Shobha Chengalur-Smith and Anna Sidorova
School of Business
University at Albany, SUNY
Albany, NY USA
Shobha@albany.edu **ASidorova@uamail.albany.edu**

Abstract

In the last few years, open source (OS) software development has become a viable alternative to commercial software. OS developers form virtual teams and the goal of this study is to examine the viability of the projects as organizational forms. We draw from the population ecology literature and propose that such factors as project reliability, size, age, and niche focus will be related to the survival of OS projects. Specifically, the purpose of this research is to test the applicability of some basic theorems of population ecology to open source projects. In this research we focus on short-term survival of OS projects. If the population ecology paradigm is useful, we can expand our analyses. The study uses archival project data available at SourceForge to test the theoretical propositions.

Keywords: Open source development, virtual communities, organizational ecology, computer-mediated collaboration

Introduction

Open source (OS) software products are generally distinguished from commercial software by their licensing. OS software typically allows downloading of readable source code along with the program and gives licenses to copy, distribute, and/or modify the program as long as the enhancements are distributed under the same licensing terms as the original software (Hamm et al. 2003). This concept existed long before Richard Stallman founded the Free Software Foundation in the 1980s. IBM used to provide software along with its mainframe hardware that customers could enhance and the customers then contributed their products to IBM for maintenance and further development (O'Reilly 1999).

In the last few years, OS software development has become a viable alternative to commercial software. Companies and government agencies strapped for cash during the recession are turning to OS software products that usually provide cheaper solutions for organizational IT needs Bulkeley 2003). Linux, an OS operating system is threatening Microsoft's and Sun's positions in the server market, and Apache has already achieved a dominant market position as a Web server.

While Linux, Apache, PHP, and a few others are responsible for most of the buzz related to OS development, these success stories account for just a minuscule percentage of the OS projects that are under way. SourceForge, one of the largest OS communities, hosts over 10,000 projects alone. While some of these projects enjoy high visibility, others receive relatively little attention. In addition, some once-active projects lose their popularity over time, while others manage to sustain their attractiveness.

The purpose of this research is to unearth factors that may explain the survival of certain OS projects. An understanding of these factors will allow companies to make more informed decisions about investments related to OS projects. Proponents of the OS movement claim that companies will create better products when their users influence product development. Clearly members of OS projects are not only part of a virtual team but also of a collaborative community. We hope that this research will have results that can be applied to a more general class of problems than software development. Thus instead of focusing on the OS software product or its implementation, we are interested in the viability of the projects themselves. We draw from the population ecology literature and focus on such factors as organizational age, size, reliability and niche strategy that have been proposed to influence survival of traditional organizations and examine whether they can explain the survival of OS projects.

The study uses archival project data from SorceForge.org to test the influence of the aforementioned factors on OS projects. The paper is organized as follows. First, OS projects are described and then the organizational ecology theory is summarized and its applicability to OS projects is discussed. Next research hypotheses relating to the survival of OS projects are formulated. Then measurement issues are discussed and the methodology being used to test these hypotheses is described. We conclude with limitations of this research and avenues for future research.

OS Development

OS software is developed by independent programmers from all over the world, typically working on one or more projects on a volunteer basis. OS projects involve computer-mediated virtual collaboration. OS projects normally consist of a core development team as well as a large number of contributors who test for bugs and add functionality (O'Reilly 1999).

Researchers have been intrigued by this new phenomenon, and this interest is apparent in the multiple streams of research that have recently emerged. While earlier studies focused on success stories of specific projects such as Linux and Apache (Fielding 1999; Torvalds 1999), later studies attempted to examine the OS development movement from multiple research perspectives. Table 1 briefly summarizes some important research on OS software development.

The focus of this research is on survival of OS projects, and we rely on organizational theory to inform our research hypotheses. Analysis of several seminal definitions of an organization (Scott 1998) suggests that organizations are groups of individuals characterized by (1) cooperation and interaction among individuals, (2) a common goal, (3) a central coordinating system, and (4) a formal establishment. As a first step in applying propositions of organizational theories to explaining the survival of OS projects, we establish the commonalities between OS projects and organizations.

Clearly, an OS project involves a group of interacting and cooperating individuals that work toward a specific goal, namely development or enhancement of a software product. Most OS projects also undergo a formal establishment procedure, which frequently involves registering a project with a foundry, such as SourceForge. (Open Source Develpoment Network's SourceForge is a Web system that supports collaboration on OS projects by providing free hosting services for project developers, including mailing lists, discussion forums, and back-up facilities.) Coordination needs in OS projects are primarily related to managing concurrent versions of software and assigning tasks related to creating patches, and such coordination is usually achieved through some kind of concurrent version control systems and tracking and communication tools such as those offered by SourceForge.

Recently, Crowston and Scozzi (2002) considered open source systems development projects as virtual communities based on the following three main aspects of virtual organizations: (1) group members have a shared interest or goal—OSS community members share a common interest in the software being developed, (2) geographical distribution, and (3) use of information and communication technology to communicate and manage interdependencies—the Internet including mailing lists, bulletin boards, and source code control systems (e.g., concurrent versioning system—CVS) are used to coordinate OSS development.

Theoretical Foundation and Research Propositions

Several general theories on organizational success and survival exist, including rational organization perspective, contingency model, transaction cost economics, resource dependence theory, population ecology, and institutional theory (Scott 1998). Of these theoretical perspectives, the organizational ecology perspective appears to be the most applicable for a study of OS projects because, as we show below, its underlying assumptions are mainly satisfied in the case of the OS movement.

Organizational ecology applies Darwinian selection to populations of organizations. The theory was first proposed by Hannan and Freeman (1977) as an alternative to the dominant theory of organizational change through adaptation (cf. Mintzberg 1979). Instead, organizational ecologists espouse the view that when environmental changes occur, new organizational forms emerge through a process of selection and replacement. This theory has since been formalized by Peli et al. (1994).

Table 1. Key Findings from the Literature

Research Focus	Study	Propositions/Results/Conclusions
Economic Motvation	Von Hippel and Von Krogh 2003	Suggest that OSSD model allow for coexistence of two types of motivation: private investment and public goods.
	Lerner and Tirole 2002	Explains individual and corporate involvement in OSSD projects through labor economics, especially the literature on career concerns, and industrial organization theory.
	Myatt and Wallace 2002	Examine the problem of public good provision in case of OSSD using game theoretical framework.
Organizational/Structural/ Project Success	Crowston and Scozzi 2002	Ability to marshal developers and manage projects using ad hoc coordination mechanisms are related to project success.
	Healy and Schussman 2003	Explain project activity level by professionalism of core contributors, hierarchy between core and other developers and clear leadership.
	Ousterhout 1999	Discusses the role of corporate involvement in fostering large-scale success of open source projects.
Development Models	Raymond 1998	Discusses distinguishing characteristics and success factors of the OSSD model.
	Zhao and Elbaum 2003	Examine the issue of quality assurance in open source projects.
	Neff and Stark 2002	Examine the role of constant user involvement in the OSSD process.
	O'Reilly 1999	Examines applicability of principles from OSSD to traditional software development.

Conditions under which the evolution theory of blind innovation would apply include the existence of individual intentions that are divorced from organizational outcomes and the uncertainty or lack of understanding among actors within organizations about the connection between the means and the ends (Hannan and Freeman 1984). OS projects are often characterized by a diversity of interest among members of a development team due to contending members seeking to shape the rules to benefit their self-interests (Bezroukov 1999). Additionally, one of the major differences between commercial software development and OS sotfware development is the lack of an explicit system-level or detailed design, project plan, schedule, or list of deliverables (Mockus et al. 2000). For these reasons, we believe that evolutionary theory would be relevant to the study of OS projects.

While the primary focus of population ecology is on populations of organizations and organizational birth and death rates (Scott 1998), population ecologists also identify some factors that are related to organizational survival at the individual organization level. In the following section, we explore the role of these determinants of organizational survival in influencing the survival of OS projects.

Reliability. Hannan and Freeman (1984) suggest that organizations with higher reliability, defined as the ability to produce collective products of a given quality repeatedly, are more likely to be able to attract critical resources and thus are more likely to survive. The survival of OS projects depends on their ability to attract new developers, as well as users who would download software, report bugs, and offer suggestions about new functionality of the software. Such involvement on the part of developers and users requires initial time investment, and is more likely if the developers and the users believe that the project will be able to reliably deliver software solutions (for the users) or provide development opportunities (for developers).

Proposition 1: More reliable OS projects are more likely to survive.

Size. According to Peli et al. (1994) larger organizations have higher reliability than smaller organizations of the same class. In the case of OS projects, it appears reasonable that projects with more developers are more likely to be able to repeatedly release high quality software updates. This suggests that OS project size is likely to be positively related to project reliability, and consequently to project survival. This is consistent with the results of a study by Cavalier (1998) that suggests that the size of the developer pool for OS projects affects success.

Proposition 2: Size of the OS project will be positively related to project reliability and hence to project survival.

Age. Peli et al. (1994) propose that reliability increases with time. This concurs with Hannan and Freeman's (1984) claim that organizational death rates decrease with age as older organizations develop "dense webs of exchange and affiliate with centers of power." In the case of OS projects, older projects are likely to develop a larger network of contributors and therefore are more likely to repeatedly release new software updates. In a similar vein, Crowston and Scozzi (2002) found that older projects were likely to be more active and in more advanced stages of development, although they may not have high interest levels as measured by the number of user downloads and page views.

Proposition 3: Age of the OS project will be positively related to project reliability and hence to project survival.

Niche. Competition plays an important role in organizational ecology. Populations that depend on identical environmental resources are said to occupy the same niche (Hannan and Freeman 1984). If two populations of organizations occupy the same niche while differing in some organizational characteristics, then the population with the less fit environmental characteristics will be eliminated. Those that occupy a broad niche will have a more generalist structure while those in a narrow niche will have a more specialist form. Specialist organizations invest resources in maximizing their fit to the environment and accept the risk of having the environment change, whereas generalist organizations invest resources in excess capacity or slack and maintain their flexibility.

Generalism is more costly in the short term but more geared to changing environments. Like traditional organizations, OS projects differ in terms of the breadth of needs satisfied by their software products. While some projects focus on software solutions to be used only by a specific audience, or to run on only one operating system, others target a broader audience and attempt to accommodate a larger range of computing environments. Since addressing the needs of multiple users or adapting a product to multiple computing environments is resource intensive, it is likely that the effect of the niche breadth on OS projects will be similar to that in traditional organizations.

Proposition 4: OS projects that occupy a broad niche are less likely to survive in the short term.

Research Method

In order to test the theoretical propositions discussed in the previous section, we will examine archival data on selected OS projects registered with SourceForge. The SourceForge foundry was started in November 1999 and includes approximately 55,000 registered projects, of which approximately 11,000 are ranked in terms of activity level. The remaining 40,000+ projects are not ranked because they do not exhibit any activity and therefore are out of the scope of interest for our study. In order to faithfully represent the 11,000 active projects, a stratified sample will be constructed which will consist of 500 projects with different activity levels.

The study will employ longitudinal data, with individual projects as units of analysis.

Project Survival. For the purpose of performing a preliminary analysis, we define project survival in the short term (six months) as existence of some activity associated with the project. Practically, we will use two operationalizations of project survival. According to the first operationalization, a project will be considered alive at a moment in time if it is included in SourceForge activity ratings. The second operationalization will rely on the concept of organizational death. That is, the project will be considered dead if it did not exhibit any activity (no CVS commits, no bugs reported, and no comments posted) for a period of time, the exact duration of which is to be determined through a sensitivity analysis. This operationalization of project survival is similar to the operationalization of project success by Crowston and Scozzi (2002) measured using activity levels, stage of development, and user interest levels.

Reliability. As defined by Hannan and Freeman (1984), reliability refers to the ability of an organization to produce a product in a predictable manner. Considering that the goal and the outcome of an OS project is a software product, a project can be considered reliable if it is able to release software updates in a predictable manner. While it is difficult to directly measure the ability to release software updates, the total number of releases appears to be a good indicator of project reliability. In addition, projects that release a production version of a software product are more likely to be perceived as more reliable than projects that only release an alpha version of their software. Therefore, the highest development stage can also be used as an indicator of project reliability. Finally, projects that concurrently have several versions of software at different development stages (such as pre-alpha, alpha, and beta) are more likely to be able to have new releases in the future; therefore, the number of simultaneous development stages can be used as another indicator of project reliability.

Niche Focus. Like traditional organizations, OS projects differ in terms of the breadth of the niche they occupy. While all projects generally provide software solutions, some projects focus only on one user group (e.g., system administrators) and others address needs of multiple user groups (e.g., end users, developers, and system administrators). Similarly, while some projects focus on developing software for specific operating systems (e.g., Linux), other projects work on developing software solutions that would run on multiple operating systems. Finally projects differ in terms of application diversity of the developed software, which is indicated by the project topic. While some projects list only one topic (e.g., file sharing) to describe the software application, other projects list multiple topics. Therefore, we will use the breadth of the intended audience, breadth of operating systems supported by the project, and application diversity as indicators of the project niche breadth.

Age and Size. Although it is difficult to determine the actual age of a project, we will rely on the date that the project was registered in the SourceForge foundry as an indicator of project age. With regard to project size, we will use the number of core developers on a project team as an indicator.

Limitations and Future Research

The purpose of the research study discussed here is to examine the role of such factors as OS project size, age, reliability, and niche focus in determining project survival as measured by software popularity and project activity. In addition to shedding light on some determinants of projects survival, the study is likely to provide insights regarding the applicability of theoretical paradigms developed for traditional organizations to OS projects. If this paradigm proves effective, it would lead us to test more involved hypotheses. For instance, Baum and Singh (1994) define two complementary terms, overlap density and non-overlap density, as measures of competition and cooperation respectively. They found that overlap density increased death rates whereas non-overlap density decreased death rates. This would suggest that OS projects that occupy a niche with a high overlap density are less likely to survive.

Some limitations of the current study are related to the operationalization and measurement of dependent and independent variables. For example, using the number of downloads as a measure of survival may seriously understate the popularity of the product as the software could be copied and shared widely. Second, although we measure the size of the project by the number of developers, Cavalier (1998) distinguished between total size (number of individual participants) and effective size (number of active contributors) and defined *effective power* as the product of the effective size and the average per participant contribution measured in hours per week. We do not currently have access to such data from SourceForge.

In terms of our study sample, inclusion of only those projects that are registered with SourceForge is likely to reduce the generalizability of our results to a certain extent. However, the sheer volume of projects registered on SourceForge should reduce any nonrepresentative bias. Most empirical studies on organizational ecology suffer from sample selection bias because many emerging organizations fail before they start operations. Thus unsuccessful founding attempts are excluded (Amburgey and Rao 1996). However, a sample of OS projects from SourceForge could yield more diversity because projects can get registered on SourceForge before any product is released.

In our present study, we have limited ourselves to analyzing short-term OS project survival. It would be of interest to see if the factors that affect short-term project survival differ from those that affect longer-term survival, say two years from now. Another complication is the fact that OS projects may have long periods of inactivity (perhaps due to the volunteer nature of developer work), and so projects that we classify as dead or inactive may merely be dormant. Thus the issue of project revival needs to be considered.

Clearly companies are starting to invest in OS projects in multiple ways and so far we have ignored the distinction between projects that are purely volunteer-based and those that are company sponsored. It would be interesting to see if sponsored projects are more likely to survive. Choice of license may also play a role in the long-term survival of a project.

References

Amburgey, T. L., and Rao, H. "Organizational Ecology: Past, Present and Future Directions," *Academy of Management Journal*, (39:5), 1996, pp. 1265-1286.

Baum, J. A. C., and Singh, J. V. "Organizatoinal Niches and the Dynamics of Organizational Mortality," *American Journal of Sociology* (100:2), September 1994, pp. 346-380.

Bezroukov, N. "Open Source Software Development as a Special Type of Academic Research (A Critique of Vulgar Raymondism)," *First Monday* (4:10), 1999 (available online at **http://www.firstmonday.dk/issues/issue4_10/bezroukov**).

Bulkeley, W. M. "Free Software—Out of the Shadows," *The Wall Street Journal*, Technology (A Special Report), March 31, 2003, p. R6.

Cavalier, F. J. "Some Implications of Bazaar Size," August 11, 1998 (available online at **http://www.mibsoftware.com/bazdev/0003.htm**).

Crowston, K., and Scozzi, B. "Open Source Software Projects as Virtual Organizations: Competency Rallying for Software Development," *IEE Proceedings Software* (149:1), 2002, pp. 3-17.

Fielding, R. T. "Shared Leadership in the Apache Project," *Communications of the ACM* (42:4), April 1999, pp. 42-43.

Hamm, S., Ante, S. E., and Greene, J. "The Linux Uprising," *Business Week* (3822), March 3, 2003, pp. 78-83.

Hannan, M. T., and Freeman, SJ. "The Population Ecology of Organizations," *American Journal of Sociology* (82:5), 197, pp. 929-964.

Hannan, M. T., and Freeman, J. "Structural Inertia and Organizational Change," *American Sociological Review* (49), 1984, pp. 149-164.

Healy and Schussman. "The Ecology of Open Source Software Development," Working Paper, Department of Sociology, University of Arizona, Tucson, AZ, 2003 (available online at **http://www.kieranhealy.org/files/drafts/oss-activity.pdf**).

Lerner, J., and Tirole, J. "Some Simple Economics of Open Source," *Journal of Industrial Economics* (50:2), 2002, pp. 197-234.

Mintzberg, H. "An Emerging Strategy of Direct Research," *Administrative Science Quarterly* (24:4), 1979, pp. 582-589.

Mockus, A., Fielding, R. T., and Herbsleb, J. "A Case Study of Open Source Development: The Apache Server," in *Proceedings of the 22nd International Conference on Software Engineering*, Limerick, Ireland, 2000, pp. 263-272.

Myatt, D. P., and Wallace, C. "Equilibrum Selection and Public Good Provision: The Development of Open Source Software," *The Oxford Review of Economic Policy* (18:4), 2002, pp. 446-461.

Neff, G., and Stark, D. "Permanently Beta: Responsive Organizations in the Internet Era," Working Paper, Center on Organizational Innovation, Columbia University, New York, 2002 (available online at **http://opensource.mit.edu/papers/neff-stark.pdf**).

O'Reilly, T. "Lessons from Open Source Software Development," *Communications of the ACM* (42:4), 1999, pp. 33-37.

Ousterhout, J. "Free Software Needs Profit," *Communications of the ACM* (42:4), 1999, pp. 44-46.

Peli, G., Bruggeman, J., Masuch, M., and O'Nuallain, B. "A Logical Approach to Formalizing Organizational Ecology," *American Sociological Review* (59), 1994, pp. 571-593.

Raymond, E. S. "The Cathedral and the Bazaar," *First Monday* (3:3), 1998 (available online at **http://www.firstmonday.org/issues/issues3_3/raymond.index.html**).

Scott, R. W. *Organizations: Rational, Natural and Open Systems,* Prentice-Hall, Upper Saddle River, NJ, 1998.

Torvalds, L. The Linux Edge," *Communications of the ACM* (42:4), 1999, pp. 38-39.

Von Hippel, E., and Von Krogh, G. "Open Source Software and the Private-Collective Innovation Model: Issues for Organization Science," *Organization Science* (14:2), 2003, pp. 209-226.

Zhao, L., and Elbaum, S. "Quality Assurance under the Open Source Development Nodel," *Journal of Systems and Software* (66:1), 2003, p. 65