Chapter 8.11
Adoption, Improvement and Disruption:
Predicting the Impact of Open Source Applications in Enterprise Software Markets

Michael Brydon
Simon Fraser University, Canada

Aidan R. Vining
Simon Fraser University, Canada

ABSTRACT

This article develops a model of open source disruption in enterprise software markets. It addresses the question: Is free and open source software (FOSS) likely to disrupt markets for commercial enterprise software? The conventional wisdom is that open source provision works best for low-level system-oriented technologies, while large, complex enterprise business applications are best served by commercial software vendors. The authors challenge the conventional wisdom by developing a two-stage model of open source disruption in enterprise software markets that emphasizes a virtuous cycle of adoption and lead-user improvement of the software. The two stages are an initial incubation stage (the I-Stage) and a subsequent snowball stage (the S-Stage). Case studies of several FOSS projects demonstrate the model's ex post predictive value. The authors then apply the model to SugarCRM, an emerging open source CRM application, to make ex ante predictions regarding its potential to disrupt commercial CRM incumbents.

INTRODUCTION

Software firms increasingly face the possibility that free and open source software (FOSS) will disrupt their most profitable markets. A disruptive innovation is a new product, service, or business

model that enters a market as a low-priced, underperforming alternative to offerings from market leaders, but which, through a process of rapid improvement, eventually satisfies the requirements of mainstream consumers and supplants incumbents (Christensen, 1997; Markides, 2006). Prototypical examples of disruptive innovations include discount online brokerages (which won significant market share away from established full-service brokerages) and personal computers (which evolved into a viable substitute for larger, more expensive mini and mainframe computers). The disruptive effect of FOSS on commercial software markets has so far been variable. At one extreme, the Apache project has forced commercial vendors of Web servers to either exit the market (IBM, Netscape), offer their products for free (Sun), or bundle their software at zero price with other offerings (Microsoft's IIS). At the other extreme, FOSS entrants in the desktop operating system and office productivity software markets have had almost no impact on the dominance of incumbent commercial software vendors.

Despite the economic importance of the commercial software industry, there has been little formal analysis of the factors that lead to major disruption by FOSS in some markets and negligible disruption in others. This is especially true for enterprise applications—the complex software programs that support critical, cross-functional business processes, such as order management, financial reporting, inventory control, human resource planning, and forecasting. FOSS, like all forms of open innovation (West & Gallagher, 2006), is characterized by recursive interdependency between adoption and technological improvement. To this point, open source production has worked best for software developed by hackers (software experts) for use by hackers. However, enterprise applications differ in important ways from commonly cited FOSS successes, such as Apache, the Perl programming language, and the Linux operating system. The intrinsic and culture-specific motivations that drive voluntary participation in FOSS projects by software experts are likely to be weaker or non-existent for business-oriented software (Fitzgerald, 2006). We might therefore expect FOSS to have less of an impact on the provision of enterprise software.

However, an alternative theoretical perspective is possible. Under certain conditions, firms may have incentives to contribute to the open source development of enterprise applications such as enterprise resource planning (ERP), customer relationship management (CRM), and supply chain management (SCM). The willingness of firms to pay programmers to write code and contribute it to a FOSS project as part of their employees' regular duties eliminates the importance of conventional hacker-level incentives in predicting whether a FOSS project will attract developers. Instead, the emphasis shifts to understanding the conditions under which a firm is motivated to make investments in a project for which it cannot fully appropriate the benefits.

We attempt to resolve the conflicting perspectives regarding the potential impact of FOSS in enterprise software markets by developing a dynamic model of FOSS disruption. Our model draws on both the Disruptive Technology and the Adoption of Technology literatures, as neither literature alone can fully account for the high variability in the level of disruption achieved by FOSS in different software markets. The disruptive technology literature emphasizes the role of technological improvement over time in fostering disruption. For example, Christensen (1997) illustrates the disruption dynamic by plotting the historical performance improvement demanded by the market against the performance improvements supplied by the technology. Improvements in performance over time are undoubtedly critical to disruption; however, little is said about the precise mechanisms by which the improvements are achieved. As Danneels (2004) points out, an *ex post* analysis of general trends is of little use when making *ex ante* predictions about the disruptive potential of a particular technology. The key