Chapter 11
A Dynamic Model of Adoption and Improvement for Open Source Business Applications

Michael Brydon
Simon Fraser University, Canada

Aidan R. Vining
Simon Fraser University, Canada

ABSTRACT
This chapter 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 enterprise business applications? The conventional wisdom is that open source provision works best for low-level system-oriented technologies while large, complex enterprise business applications are best provided by commercial software vendors. The authors challenge the conventional wisdom by developing a two-stage model of open source disruption in business application 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
Many commercial software firms face the possibility that free and open source software (FOSS) will disrupt their markets. A “disruptive innovation” is a new product, service, or business model that initially enters a market as a low-priced, lower-quality alternative to the products of market incumbents but which, through a process of rapid improvement, eventually satisfies mainstream consumers and supplants some or all 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 com-
commercial software markets has been variable so far. On the one hand, 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). On the other, FOSS entrants in the desktop operating system and office productivity software markets have had almost no impact on incumbents. Despite the economic significance of the software industry, there has been little formal analysis of the factors that lead to major disruption by FOSS in some markets but negligible disruption in others. This is especially true of 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.

What drives FOSS? Like all forms of open innovation, FOSS is characterized by recursive interdependence between user adoption and technological improvement (West & Gallagher, 2006). To this point, open source production has worked most effectively for software developed by hackers (software experts) for use by hackers. However, enterprise applications differ in important ways from well-known 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). Accordingly, one might expect FOSS to have less disruptive impact on the market for enterprise applications. However, an alternative scenario is possible. Under certain conditions profit-maximizing firms have clear incentives to contribute to the development of open source enterprise software, such as enterprise resource planning (ERP), customer relationship management (CRM), and supply chain management (SCM) packages. The willingness of firms to pay programmers to write code and contribute it to a FOSS project as part of their employees’ regular duties reduces or eliminates dependence on conventional hacker-level incentives in predicting who will contribute to FOSS projects. Instead, the emphasis shifts to understanding the conditions that lead profit-maximizing firms to invest in projects from which they cannot fully appropriate the benefits of their investment.

We estimate the probable impact of FOSS in enterprise software markets by developing a dynamic model of FOSS adoption and improvement. The objective is to help predict whether open source entrants will disrupt commercial incumbents in a particular software market. The model draws on both the disruptive technology and the adoption of technology literatures because neither literature alone can fully account for the high variability in the level of disruption achieved by FOSS.

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, as shown in Figure 1. Improvements in performance over time are undoubtedly critical to disruption; however, little is said about the precise mechanisms by which the improvements—or “sustaining innovations”—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 to predicting whether a technology is potentially disruptive or “merely inferior” (Adner, 2002) is the identification of plausible mechanisms for rapid and significant improvement along dimensions of performance that matter to mainstream users.

Models from the adoption of technology literature, in contrast, tend to focus on specific attributes of the innovation or environment in order to predict the innovation’s adoptability. The critical shortcoming of most adoption models for our purposes is that they are static. Disruption is
Related Content

Improving Constraints Checking in Distributed Databases with Complete, Sufficient, and Support Tests
www.igi-global.com/chapter/improving-constraints-checking-distributed-databases/20718?camid=4v1a

Data Modeling for Non-Standard Data
www.igi-global.com/article/data-modeling-non-standard-data/51091?camid=4v1a

Two Meta-Model for Object-Role Modeling
Dave Cuyler and Terry Halpin (2005). Information Modeling Methods and Methodologies: Advanced Topics in Database Research (pp. 17-42).
www.igi-global.com/chapter/two-meta-model-object-role/23007?camid=4v1a

A Systematic Literature Review on the Quality of UML Models
www.igi-global.com/chapter/systematic-literature-review-quality-uml/74398?camid=4v1a